**ТЕХНОЛОГИЧНО УЧИЛИЩЕ “ЕЛЕКТРОННИ СИСТЕМИ”**

**към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

**ДИПЛОМНА РАБОТА**

Тема: Изграждане на MPLSVPN мрежова архитектура

Дипломант: Научен ръководител:

*Александър Павлов Христо Войнски*

СОФИЯ

2020

**Използвани съкращения**

* IP – Internet Protocol
* OSI – Open Systems Interconnection
* LAN – Local Area Network
* WAN – Wide Area Network
* MAC – Media Access Control
* BGP – Border Gateway Protocol
* eBGP – Exterior Border Gateway Protocol
* iBGP – Interior Border Gateway Protocol
* MPLS – Multiprotocol Label Switching
* LDP – Label Distribution Protocol
* RSVP - Resource Reservation Protocol
* VRF – Virtual Routing & Forwarding
* IS-IS – Intermediate System to Intermediate System
* QoS – Quality of Service
* TCP – Transmission Control Protocol
* IPv4 – Internet Protocol version 4
* IGP – Interior Gateway Protocol
* EGP – Exterior Gateway Protocol
* IGRP – Interior Gateway Routing Protocol
* EIGRP – Enhanced Interior Gateway Routing Protocol
* OSPF – Open Shortest Path First
* NET – Network Entity Title
* RIP – Routing Information Protocol
* RIPng - Routing Information Protocol next generation
* MAC – Media Access Control
* PDU – Protocol Data Unit
* LSPs – Link State PDUs
* IIHs – IS-IS Hello PDUs
* CSNPs - Complete Sequence Number PDUs
* PSNPs - Partial sequence number PDUs
* VPN – Virtual Private Network
* RFC – Request For Comment
* AD – Administrative Distance
* AS – Autonomous System
* UDP – User Datagram Protocol
* MED – Multiple Exit Discriminator
* TTL – Time-To-Live
* NHLFE - Next Hop Label Forwarding Entry
* ILM - Incoming Label Map
* DIS – Designated Intermediate System
* PDU – Protocol Data Unit

**Увод**

В днешно време компютърните мрежи са изключително необходими и с всеки изминал ден мрежовата свързаност нараства. Хората не биха могли да комуникират помежду си, ако не съществуваха мрежовите устройства. С нарастването на компютърните технологии нараства и нуждата за сигурност, ефикасност и качество на интернет услугите. Големите компании и организации, зависими от Интернет свързаността, тези които поддържат и изпълняват изискванията от своите клиенти, се нуждаят от множеството технологии и протоколи, които предоставят ефикасност, качество, сигурност и съвременно обменят информация помежду си.

За да могат доставчиците да предоставят на клиентите си своите услуги най-ефикасно и най-качествено, се използват множество техологии и методи. Често използвана технология при корпоративните решения е Multi-Protocol Label Switching (MPLS). Една от най-често срещаните услуги на доставчиците, са виртуални частни мрежи (VPN) от втори и трети слой на OSI модела. Чрез тях се постига бързодействие, надеждност и улеснение при поддръжката на частните мрежи на клиентите, като се предоставя и възможност за пренасяне на различен трафик, спрямо изискванията на компанията.

Главната цел на дипломната работа е да се изгради мрежова архитектура, която съдържа мрежата на доставчик на интернет услуги и два негови клиента, които са представени като две копмании, всяка от които има централен офис и клон офис на различни географски разположения. Доставчикът предоставя на всеки от клиентите отделна виртуална частна мрежа, чрез която се свързват двата офиса на двете отделни компании, като едната компания няма достъп до мрежата на другата. Създава се корпоративно решение, използвайки едни от най-използваните и ефикасни технологии и протоколи – BGP, IS-IS, MPLS и се имплементират политики за контрол на качеството на услуги. Изграждането на мрежовата архитектруа е на базата на мрежовия симулатор – GNS3, а тестването и работоспособността на мрежата се осъществява чрез командите *ping, traceroute* и анализатор на трафик *Wireshark.*

**Първа глава: Технологии, протоколи и стандарти, използвани в MPLS VPN мрежовата топология**

* 1. **Интернет протокол версия 4 – IPv4**

Интернет протокол версия 4 (IPv4) е четвъртата ревизия на Интернет протокола (IP) и е широко използван протокол при комуницкациите в различните типове мрежи. Протоколът служи за индентификацията на всяко едно устройство свързано в мрежата. Той работи на Мрежовия слой на OSI модела и освен че се използва при статична маршрутизация, се използва и от протоколите за динамична маршрутизация като OSPF, EIGRP, RIP, IS-IS за избиране на най-оптималния път през който интернет пакета да премине за да достигне своята дестинация. Без IP адреси, устройствата не биха могли да комуникират помежду си и да изпращат данни един на друг. Протоколът е същественото в инфраструктурата на мрежите в целия свят.

Когато устройство се свърже към дадена мрежа, на него му се възлага уникален логически адрес, чрез който може да бъде достъпено от другите устройства в мрежата и чрез който може да комуникира с тях. Логическият адрес се дели на две части: хост част и мрежова част. Мрежовата част посочва мрежовия сегмент към който спада хоста, а хост частта - неговия адрес в тази мрежа. Адресацията се предоставя като 32 битов адрес, който е разделен на четири октети, чрез точки. Всеки октет е формиран от осем бита. За различаване на мрежовата част и хост частта на логически адрес се използва мрежова маска (subnet mask). Тя е 32 битова и е съставена от „1“ и „0“. Частта на мрежовата маска, която е съставена изцяло от „1“ определя мрежовата част, а частта съставена изцяло от „0“ – хост частта.

IP адресната класация се разделя на: Class A, B, C, D, E. Всеки клас има различен брой на възможните хостове и подмрежи и различна мрежова маска.

Разпределението, предназначението и детайлите на всеки клас са показани на Фиг.1.1.

*Фиг.1.1. Класове адреси*

От класове A, B, C освен публични адреси има и частни адреси, които са със собствени мрежови маски. Адресните пространства, резервирани за частна адресация са следните: 10.х.х.х/8 (Class А), 172.16.х.х/16 (Class B), 192.168.х.х/24 (Class C) [2]. За да се използват адресните пространства най-ефикасно и да няма излишък от неизползвани адреси, поради голямата мрежова маска, се използва метод наречен VLSM (Variable Length Subnet Mask). Чрез този метод мрежовата маска се разделя на подмаски с дължина зададена според изисквания брой хостове.

 На фиг.1.2. е показан заглавната част на IP пакета заедно с нейните полета.

*Фиг.1.2. IPv4 заглавна част (header)*

Заглавната част на IPv4 съдържа следните полета:

* **Version –** Първото поле указва коя версия на IP е пакета. Понеже само IPv4 ползва това поле, винаги стойността на това поле ще бъде 4 бита.
* **Header Length –** Това 4 битово поле указва каква е дължината на заглавната част (header) на IP пакета с 32 битови инкрементации. Минималната стойност на това поле е 5 или представено в битове „0101“, а максималната стойност е 15 или представено в битове „1111“.
* **Type of Service –** Това поле се използва за QoS (Quality of Service) или така нареченото „Качество на услугите“. Предоствени са 8 бита за това поле, чрез които може да се маркира какъв трафик пренася този пакет.
* **Total Length –** Това 16 битово поле указва целия размер на IP пакета (заглавна част и данни). Минималният размер е 20 байта, ако няма данни (Data), и максималният размер е 65,535 байта.
* **Identification –** Поле, което се използва за различаване на фрагментирани пакети от различни дейтаграми.
* **IP Flags –** Полето се състои от 3 бита, които се използват за фрагментация: Първият бит винаги е със зададена стойност „0“ ; Вторият бит се нарича DF (Don’t Fragment) бит, който показва че съответния пакет не трябва да бъде фрагментиран.
* **Fragment Offset –** В случай на фрагментирани IP пакети, това поле съдържа изместването от началото на IP пакета. Това поле се използва за сглобяване на фрагментираните IP пакети.
* **Time to Live –** Всеки път, в който IP пакет премине през маршрутизатор, стойността на това поле се намалява с 1. Когато това поле достигне стойност „0“, маршрутизаторът ще отхвърли съответния пакет и ще изпрати „ ICMP: time exceeded” съобщение към подателя, изпратил този пакет. Time to Live полето има 8 бита и се използва за предотвратяване на маршрутизиращ цикъл.
* **Protocol –** В това 8 битово поле се указва кой протокол е енкапсулиран в IP пакета, чрез стойност, например TCP (Transmission Control Protocol) има стойност 6 и UDP (User Datagram Protocol) има стойност 17.
* **Header Checksum –** В това 16 битово поле се съхранява „checksum” на заглавната част на IP пакета. Стойността се изчислява използвайки алгоритъм който преминава през всички полета на заглавната част на IP пакета. Получателят на пакета използва това поле, с цел да провери ако съществуват грешки в заглавната част.
* **Source Address –** Това поле указва 32 битов адрес на IP източника (подателя).
* **Destination Address –** Това поле указва 32 битов адрес на IP дестинацията (получателя).
* **IP Option –** Това поле не е често използвано, не е задължително и има променлива дължина според използваните опции. Когато се използва това поле, стойността на полето „Header Length“ се увеличава.
* **Data –** Това поле съдържа заглавната част и данните получени от протоколите от транспортния слой.
  1. **Статична и динамична маршрутизация**

За да може един маршрутизатор да изпраща пакети до други мрежи, той трябва да има попълнена маршрутизираща таблица, за да знае къде да изпрати съответната информация. Маршрутизиращата таблица на всеки маршрутизатор съдържа пътища до дадена дестинация, научени чрез различни начини. Освен пътищата научени като директно свързани пътища (път, който е научен от маршрутизатора чрез физическа свързаност с друго устройство), съществуват пътища, които са ръчно конфигурирани, така наречените статични пътища, и пътища които се научават с помощта на маршрутизиращи протоколи – динамични пътища.

**1.2.1. Статична маршрутизация**

Статичната маршрутизация е процес на ръчно конфигуриране, от мрежовият администратор, на пътища в маршрутизиращата таблица. Тези пътища не се променят след като се конфигурират, освен ако не се промени от мрежови администратор. Статичните пътища не се влияят от възникнали промени в мрежата, затова не изисква допълнителни ресурси за да научи промените. Ако освен статичен път има и други пътища към една дестинация, ще се предпочете статичния път, защото той е с по-голям приоритет за маршрутизатора - има AD равен 1. Може да се зададе статичен път по подразбиране (Default static route), през който минава всеки пакет, за който не е намерен път в маршрутизиращата таблица, през който да премине към своята дестинация.

**1.2.2. Динамична маршрутизация**

Динамичната маршрутизация е по-комплексен начин за маршрутизация, осъществяваща се на базата на маршрутизиращи протоколи, които променят маршрутизиращата таблица според промените настъпващи в мрежата. Преизчислението на пътищата се случва постоянно и поради това този тип маршрутизация изисква много повече ресурси отколкото статичната. Когато се появи промянa в дадена мрежа, се изпраща съобщение на маршрутизатора за да специфицира тази промяна, да си обнови и прекалкулира динамичните си пътища и да изпрати информация за променените си пътища на другите маршрутизатори в мрежата, за да попълнят промените в своята таблица. Преизчисленията на пътищата става на базата на специфичен алгоритъм, според зависи от маршрутизиращия протокол. Съответно използването на голяма част ресурси се счита за най-големия недостатък на динамичната маршрутизация.

**1.2.3. Външни и вътрешни маршрутизиращи протоколи**

* **Външни протоколи (EGP) –** Протоколи, които обменят информация и функционират между две или повече автономни системи. Този тип протоколи спада към една единствена категория – „Path-Vector“.
* **Вътрешни протоколи (IGP) –** Протоколи, които функционират в рамките на една автономна система. Те се разделят на три категории: Link-State, Distance-Vector, Hybrid.

**1.2.4. Link-State маршрутизиращи протоколи**

Този тип маршрутизиращи поддържат топологична таблица, в която се съдържа цялата информация за мрежовата топология и за всеки маршрутизатор, който използва „Link-state“ маршрутизиращ протокол. Тези маршрутизатори изпращат информация за своите директно свързани пътища и техните състояния. Тази информация се изпраща под формата на „multicast” съобщение. Когато топологичната таблица е завършена, всеки маршрутизатор изчислява най-добрите пътища за всяка мрежа. „Link-state“ протоколите са базирани на „Shortest Path First“ (SPF) алгоритъма за изчисление. Този алгоритъм е познаван също като „Dijkstra” алгоритъм. Чрез този алгоритъм, когато някое състояние на път се промени, маршрутизаторите си изпращат актуализиращо съобщение (Link-State Advertisement) помежду си и пътищата се преизчисляват отново.

Този тип протоколи са по-малко податливи на маршрутизиращи „loops“ отколкото са Distance-Vector протоколите. От друга страна, изискват повече процесорна мощ и памет. IS-IS и OSPF са част от Link-State протоколите.

**1.2.4. Distance-Vector маршрутизиращи протоколи**

Това са протоколи, които използват алгоритъм за изчисление на метрика и цена на базата на разстоянието до друга мрежа – през колко маршрутизатора ще премине един пакет или колко на брой скокове (hops) ще направи. Първоначално се изпраща „broadcast“ съобщение, което има за цел да обяви присъствието на нов маршрутизатор в мрежата. Всеки от маршрутизаторите, използващ Distance-Vector протокол, научава за нови пътища през своите съседи. Цялото действие на динамичните протоколи от този тип зависи от информацията, идваща от съседите на даден маршрутизатор. Те не знаят за състоянието на мрежата, познават само съседните маршрутизатори. Изпращат се периодични съобщения с цялата маршрутизираща таблица и времето им на сходимост е голямо. RIPv1, RIPv2, EIGRP (този протокол се смята за хибриден, защото обединява спецификациите на няколко категории протоколи), IGRP са част от протоколите от този тип.

**1.2.5. Path-Vector маршрутизиращи протоколи**

Това са протоколи, които са подобни на Distance Vector протоколите, но не се разчита на следенето на дължината на път, а по-скоро се анализира даден път (AS път). Често тези протоколи се използват за маршрутизиране в среди, където няма постоянна метрика. Устройството, през което минава пакета, се добавя към AS пътя и се праща към следващите маршрутизатори. Добавеният път се анализира, за да се провери валидността му. Border Gateway Protocol е единствения протокол, който спада към Path-Vector категорията. На Фиг.1.3 са означени категориите протоколи.

****

*Фиг.1.3. Категории протоколи за динамична маршрутизация*

**1.2.6. Border Gateway Protocol**

Border Gateway Protocol (BGP) е Path-vector маршрутизиращ протокол, позволяващ осъществяването на връзката между отделните автономни системи (AS). Автономните системи са група от маршрутизатори, които са под управлението на една или повече компании с еднакви политики за маршрутизатори, като използват вътрешни протоколи за маршрутизация за комуникация помежду си и външни за комуникация с другите автономни системи. Има два вида BGP – internal BGP (iBGP) за вътрешна маршрутизация и external BGP (eBGP) за външна маршрутизация. При имплементация на eBGP, маршрутизаторите трябва да са директно свързани, защото TTL (Time to Live) параметърът BGP съобщенията е със стойност 1, което означава че съобщението може да стигне само до един маршрутизатор, след като е изпратено. Aко e пусната функционалността „multi-hop“, TTL стойността е по-голяма от 1. Автономната система се означава с 32-битов номер (преди 2007г. – 16 битов), като този номер е от категорията „Публични автономни системи“ или „Частни автономни системи“. Публичните автономни системи се означават с номер от 1 до 64 495 и частните с номер от 64 512 до 65 534, като също така съществуват номера автономни системи специално резервирани за документация, които са в диапазона от 64 496 до 64 511. Всеки маршрутизатор, участващ в BGP процеса, обявява своите пътища със BGP съседите си. Начинът за ограничение на споделяния между BGP съседите трафик е чрез имплементацията на маршрутизираща политика, която представлява поредица от правила които се прилагат на изходящия или входящия трафик отвъд или в рамките на една автономна система. На Фиг.1.4 е означена свързаността между ****три автономни системи, посредством eBGP и iBGP.

*фиг.1.4. Свързаност между три автономни системи*

Протоколът има четири версии, като последната версия има функциолността да пренася и IPv6 трафик. Поддържа се Classless Inter Domain Routing (CIDR), което позволява да се използват мрежи с променлива маска на мрежата (subnet mask), като BGP поддържа механизъм за агрегиране на мрежите до класово разделяне на IP адресите. По този начин се спестяват изчеслителни ресурси и се разтоварва трафика от мрежата. BGP работи с Transport Control Protocol (TCP) за транспортен протокол за обмен на съобщения и работи на порт 179. Маршрутизаторите използващи BGP се наричат „BGP speakers“, а тези, установили TCP сесия, се наричат „peer” маршрутизатори. Първоначално, при стартиране на мрежата с BGP, всички маршрутизатори разменят пълните си маршрутизиращи таблици и след това започват да изпращат съобщения, ако е настъпила промяна в таблиците.

В зависимост от броя на автономните системи, има възможност BGP да не е препоръчително да се използва. Топологиите биват:

* **Single-homed** – това е топология, при която автономна система има връзка само с една друга автономна система. За предпочитане е да се използват статични пътища. Предимството на тази топология е че е изгодна откъм ценови аспект. Големият недостатък обаче е че липсва резервираност, понеже е само една връзка и ако се появи проблем и тя спре да функционира, няма да има друг път по който да се осъществи свързаност. На Фиг.1.5. е показана single-homed топология.

*Фиг.1.5. Single-homed топология*

* **Multi-homed** – това е топология, при която автономна система има връзка с повече от една автономна система. При това положение е подходящо да се използва eBGP. Основно предимство на тази топология е, че за разлика от single-homed, има по-добра резервираност поради наличието на повече от 1 връзка между автономните системи. На Фиг.1.6. е показана multi-homed топология.

*Фиг.1.6. Multi-homed топология*

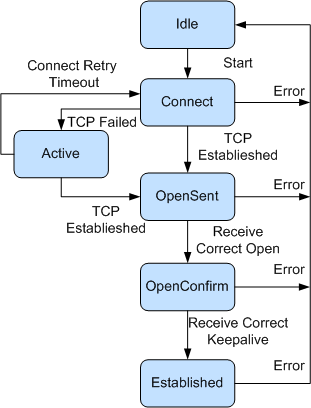
BGP използва четири типа съобщения, в зависимост от състоянието на мрежата:

* **OPEN** – първото съобщение, което се изпраща, след като се установи TCP сеисията. Чрез него се изпраща информация за версията на протокола, номера на автономната система, времето на изчакване (Hold time), в което е нужно съседния маршрутизатор да отговори, както и адреса на BGP speaker.
* **KEEPALIVE** – изпраща се от отсрещната страна, след като се приеме OPEN съобщението. Този тип съобщение се изпраща в интервал от 60 секунди по подразбиране, за да поддържат връзката.
* **UPDATE** – използва се, за да се обменя маршрутизираща информация. Изпращат се мрежовия адрес и дължината на subnet маската, информация за пътя и недостъпните пътища.
* **NOTIFICATION** – съобщение което се изпраща към всички BGP маршрутизатори в автономната система в случай на грешка, като за всеки тип грешка има номер, указан в съобщението.

При имплементация на BGP е препоръчително и най-добре да се използва „loopback“ интерфейс (логически интерфейс, който е постоянно активен, освен ако не бъде деактивиран от мрежовият администратор), с който се гарантира, че мрежовия адрес на маршрутизатора ще е постоянно достъпен.

**1.2.6.1. BGP session establishment**

В рамките на автономната система, BGP работи посредством IGP (Internal Gateway Protocol). За да се създаде завършена BGP сесия между два маршрутизатора се обменят определен брой съобщения. Броят на съобщенията варира, според това дали е успешно преминаването от едно състояние в друго или не. На Фиг.1.7 са означени състоянията през които се преминава за да се осъществи BGP сесия.

*Фиг.1.7. Състояния на BGP сесията*

* **Idle** – това е първоначалното състояние в което се намира маршрутизатора, при което не приема заявките изпратени към него за BGP съседство, от съседните маршрутизатори. След като се пусне BGP процес на маршрутизатора, се инициализира TCP сесия към съседния маршрутизатор. Вследствие на което, се преминава в състояние Connect.
* **Connect** – при това състояние, локалният маршрутизатор стартира ConnectRetry таймер (120 секунди) и изчаква да се завърши TCP сесията. Ако успешно се инициализира TCP сесията, локалният маршрутизатор изпраща OPEN съобщение към съседа си и се преминава в състояние OpenSent. В случай, че TCP сесията се инициализира неуспешно, локалният маршрутизатор нулира ConnectRetry таймера и се преминава в състояние Active. Ако ConnectRetry таймера изтече докато локалният маршрутизатор е в състояние Connect, таймера се нулира и се прави нов опит за връзка. Маршрутизаторът остава в състояние Connect.
* **Active** – при това състояние, локалният маршрутизатор опитва да инициализира TCP сесия със своят съсед. Ако сесията се установи успешно, се изпраща OPEN съобщение към съседа и локалният маршрутизатор преминава в състояние OpenSent. Ако TCP сесията не се установи успешно, локалният маршрутизатор инициализира нова сесия, нулира ConnectRetry таймера и преминава обратно в състояние Connect.
* **OpenSent** – при това състояние, TCP сесията е успешно установена и локалният маршрутизатор изпраща OPEN съобщение и изчаква да получи OPEN съобщеие от своя съсед. Когато валидно се получи валидно OPEN съобщение, локалният маршрутизатор започва да изпраща KEEPALIVE съобщения към своя съсед. BGP съседите обменят своите параметри и локалният маршрутизатор преминава в състояние OpenConfirm. В случай, че TCP сесията се провали, когато маршрутизаторът е в състояние OpenSent, BGP сесията се прекратява, нулира се ConnectRetry таймера и се преминава в състояние Active. Ако, локалният маршрутизатор получи невалидно OPEN съообщение, се преминава в състояние Idle.
* **OpenConfirm** – преминава се в това състояние, когато локалният маршрутизатор получи валидно OPEN съобщение от съседа си. Локалният маршрутизатор изпраща KEEPALIVE съобщения към своя съсед и изчаква да получи KEEPALIVE съобщение от него. В случай, че не получи KEEPALIVE съобщение от своя съсед, се преминава в състояние Idle.
* **Established** – локалният маршрутизатор достига това състояние, когато получава KEEPALIVE съобщение от своя съсед, докато е в състояние OpenConfirm. Това е последното състояние на съседство и обозначава напълно функционираща BGP връзка. В случай, че локалният маршрутизатор получи NOTIFICATION съобщение, се преминава в състояние Idle.

**1.2.6.2. BGP Path Attributes**

За избирането на най-добър път до отдалечената автономна система, при използването на BGP, се гледат атрибутите, които имат даден път до избраната дестинация. Те се разделят на четири категории:

* **Well-known mandatory** – това са параметри, които задължително трябва да се разбират от всички маршрутизатори, използващи BGP. Те трябва да се съдържат във всяко едно BGP UPDATE съобщение.
* **Well-known discretionary** – това са параметри, които задължително трябва да се разбират от всички маршрутизатори, използващи BGP, но е възможно и да не се включват в BGP UPDATE съобщенията.
* **Optional transitive** – това са параметри, за които не е задължително да се разбират от всички маршрутизатори, използващи BGP, но трябва да се приемат и да се предадат на съседните маршрутизатори.
* **Optional non-transitive** – това са параметри, за които не е задължително да се разбират от всички маршрутизатори, използващи BGP, но може да изпусне BGP UPDATE съобщението и да не го предаде на съседните маршрутизатори.

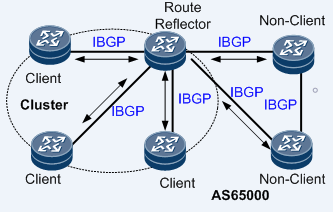
Маршрутизаторите, използващи BGP, взимат решението за това по какъв начин определена информация да достигне дестинацията си. Решението се взима на базата на параметрите от четирите категории. Даден маршрут се избра по определен ред на атрибутите на пътя. Съществуват десет стандартни атрибута, но се практикуват само седем, които са най-често използваните при имплементация на BGP:

* **Local preference (LOCAL PREF)** – използва се при пренос на информация вътре в автономната система. Указва най-добрият път за изход на автономната система. Този атрибут се конфигурира от мрежовият администратор и действа на базата на приоритет. Пътят с най-висок приоритет бива избран за основен BGP път към определената мрежа, през който да премине трафикът, а всички други за резервни пътища, които се активират, когато основният отпадне. Колкото е по-голяма стойността на приоритета, толкова по-предпочитан пътя.
* **AS\_PATH** – това е задължителен атрибут (от категория well-known mandatory), който указва през кои автономни системи преминава информацията. Този атрибут се използва, за да се предотвратят цикли в мрежата.
* **ORIGIN** – това е задъкжителен атрибут (от категория well-known mandatory), който указва по какъв начин определен път е въведен в BGP. Той указва маршрутизатора от когото идва информацията.
* **Multi-Exit Discriminator (MED)** – това е незадължителен атрибут (optional non-transitive), който се използва за информация, пренасяна между автономните системи. Указва кой път е по-добър за входен трафик в автономната система, ако има повече от един.
* **NEXT\_HOP** – това е задължителен (well-known mandatory), който дефинира следващия IP адрес по пътя за избраната дестинация. За eBGP това е винаги адресът на съседния маршрутизатор.
* **ATOMIC\_AGGREGATE** – това е задължителен атрибут, който може да се включва в UPDATE съобщението (well-known discretionary). Използва се, когато даден маршрутизатор събира няколко пътя, агрегира ги, за да се предаде на съседен маршрутизатор. Този атрибут не трябва да се премахва, ако пакета с информация се предава.
* **AGGREGATOR** – това е незадължителен атрибут (optional transitive), който указва IP адреса на последния маршрутизатор, агрегирал номерата на автономните системи.

Има четири допълнителни атрибута, които са създадени с цел да се разшири действието на протокола:

* **Community** – това е незадължителен атрибут, указващ група от дестинации. Използва се, за да групира крайни точки и определени общности, за да се прилагат различни политики.
* **Originator ID** – това е незадължителен атрибут, показващ източника на пътя, когато се използва Route Reflector. Предотвратява зацикляне в мрежата.
* **Cluster list** – това е незадължителен атрибут, който показва номера на cluster, изпратен от Route Reflector.

**1.2.6.3. Route Reflection**

**Когато се използва iBGP, топологията трябва да е full-mesh, тоест всеки маршрутизатор да е свързан към всички други. Този метод може много да усложни мрежата и затова се въвежда Route Reflector (RR). Това научава път от iBGP маршрутизатор и го препраща на друг такъв. По този начин няма нужда от допълнително окабеляване на мрежата и BGP сесиите се намаляват драстично. В една мрежа може да има повече от един RR, като те се третират като стандартни iBGP маршрутизатори, което означава, че трябва да има full-mesh топология на RR маршрутизаторите, като има възможност да се зададе предпочитан RR и да се създаде Cluster list. Един маршрутизатор може да има две роли – Route Reflector или Route Reflector Client. Когато даденият маршрутизатор е Route Reflector Client, той препраща научените пътища към Route Reflector маршрутизатора и така всеки друг Route Reflector Client знае и може да достъпи тези пътища без да има осъществена iBGP сесия между съответните Route Reflector Client маршрутизатори. По този начин използваните ресурси в мрежовата топология се намаляват и се избягва прекомерното окабеляване. На Фиг.1.8. е означена мрежова топология на автономна система с iBGP и Route Reflection метод.

*Фиг.1.8. Мрежова топология на автономна система посредством iBGP и RR*

Чрез Route Reflection методът, така наречените „клиенти“ не знаят че се използва Route Reflector метод в топологията и считат че е конфигурирана full-mesh топология. Всеки RR маршрутизатор съдържа „cluster-id“, което се счита за номер на група. Има възможност да има повече от един RR маршрутизатор в една група. Ако има само един RR маршрутизатор в групата, тогава cluster-id получава номера на идентификационния номер на маршрутизатора в BGP процеса.

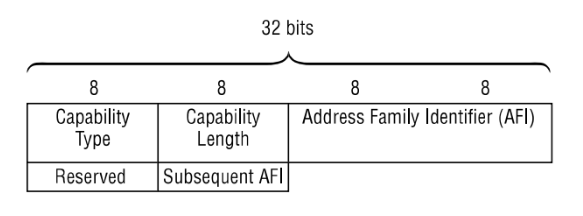
Route Reflector клиент се нарича iBGP маршрутизатор, на който Route Reflector маршрутизаторът отразява пътища. Non-client се нарича обикновен iBGP маршрутизатор, който не участва в Route Reflection процеса и не биват отразявани пътища към него, но поддържа iBGP сесия с Route Reflector маршрутизатора.

Когато се отразят пътища от Route Reflector маршрутизатор, се прилагат следните правила:

* Път научен от eBGP съсед може да бъде препратен към друг eBGP съсед, клиент и non-client.
* Път научен от клиент може да бъде препратен към друг eBGP съсед, клиент и non-client.
* Път научен от non-client, може да бъде препратен към друг eBGP съсед и клиент, но не и към друг non-client маршрутизатор.

**1.2.6.4. Multiprotocol BGP**

Развитието на BGP и неговата широкоразпространена използваемост, го правят уникална платформа за споделяне и научаване на информация помежду автономните системи, както и вътре в тях. Тази информация може да съдържа IPv6 пътища или IPv4 пътища. Информация свързана с виртуални частни мрежи (VPN) и Multiprotocol Label Switching (MPLS) също може да се обменя помежду BGP съседи. Способността на BGP да пренася подобна информация се нарича Multiprotocol BGP (MPBGP). MPBGP е функционалност, за която се обменя информация помежду BGP съседите, по време на установяването на съседство. Всеки маршрутизатор описва своята способност да поддържа различна „reachability“ информация, като изпраща „Capability advertisement“ в BGP OPEN съобщението. На Фиг.1.9 е показан формата на Capability advertisement.

*Фиг.1.9 Формат на MPBGP Capability advertisement*

* **Capability Type** – това поле указва типа на способност, която се обменя между съседите. За MPBGP, това поле има стойност равна на 1, което обозначава multiprotocol разширения.
* **Capability Length** – това поле указва дължината на останалите полета в Capability опцията. Постоянна стойност равна на 4 е използвана за всички MPBGP преговори.
* **Address Family Identifier (AFI)** – това поле кодира типа на информацията от Интернет слоя, която маршрутизаторът използва по време на сесията. AFI стойностите използвани от маршрутизаторите включват:
  + 1 – IPv4
  + 2 – IPv6
  + 25 – Layer 2 VPN
* **Reserved** – това поле не се използва и му е зададена постоянна шестнадесеттична стойност равна на 0x00.
* **Subsequent Address Family Identifier (SAFI) –** това поле предоставя допълнителна информация относно маршрутизираният трафик пренасян мрежду съседи. SAFI стойности използвани от маршрутизатори включват:
  + 1 – Unicast
  + 2 – Multicast
  + 4 – Labeled unicast
  + 128 – Labeled VPN unicast
  + 129 – Labeled VPN multicast

Всяка съвкупност от маршрутизираща информация използвана от машрутизатора е уникално описана от нейните AFI и SAFI номера. С изключение на IPv4 unicast пътищата, всяка друга Network Layer Reachability Information (NLRI) се споделя и отхвърля, чрез използването на MP\_Reach\_NLRI и MP\_Unreach\_NLRI BGP атрибути. Multiprotocol Unreachable Network Layer Reachability Information (MP\_Unreach\_NLRI) е атрибут позволяващ пренасянето и транспортирането на пътища, които са недостижими.

**1.2.7.** **Intermediate System to Intermediate System (IS-IS)**

Intermediate System to Intermediate System (IS-IS) е протокол за динамична маршрутизация, класифициран като Interior Gateway Protocol (IGP) и спада под категорията „Link-State” маршрутизиращи протоколи. Протоколът обменя маршрутизираща информация в рамките на една автономна система. Като всеки Link-State протокол, IS-IS създава топологична таблица, която съдържа информация за физичексата свързаност на мрежата. Информацията за тази таблица се изпраща на всеки маршрутизатор в автономната система. След като всеки маршрутизатор има пълната информация за физическата свързаност на мрежата, започва процес на изчисляване на най-оптималните пътища до всички дестинации в мрежата. Процесът на изчисление се базира на Dijkstra алгоритъма.

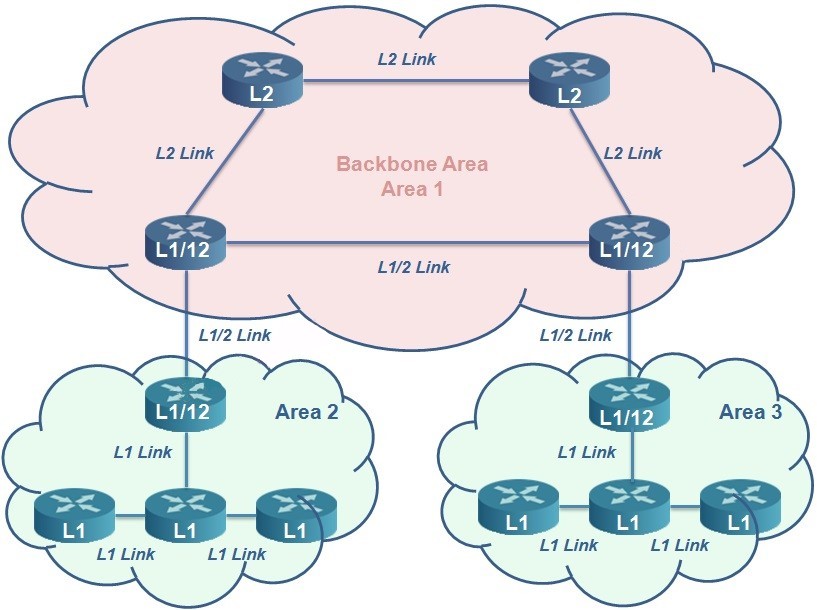
Основният недостатък на IS-IS е, че като всеки Link-State протокол, няма добро мащабиране, като повече маршрутизатори се добавят към маршрутизиращия домейн. С увеличаването на броя на маршрутизаторите се увеличава и големината и честотата на топологичните актуализиращи съобщения, както и времето за което се изчисляват всички пътища на ново. Тази липса на мащабируемост означава, че Link-State протоколите не са подходящи за марщрутизация в по-обширни и големи мрежи. Това е причината, поради която Interior Gateway протоколите маршрутизират трафик само в рамките на една автономна система.

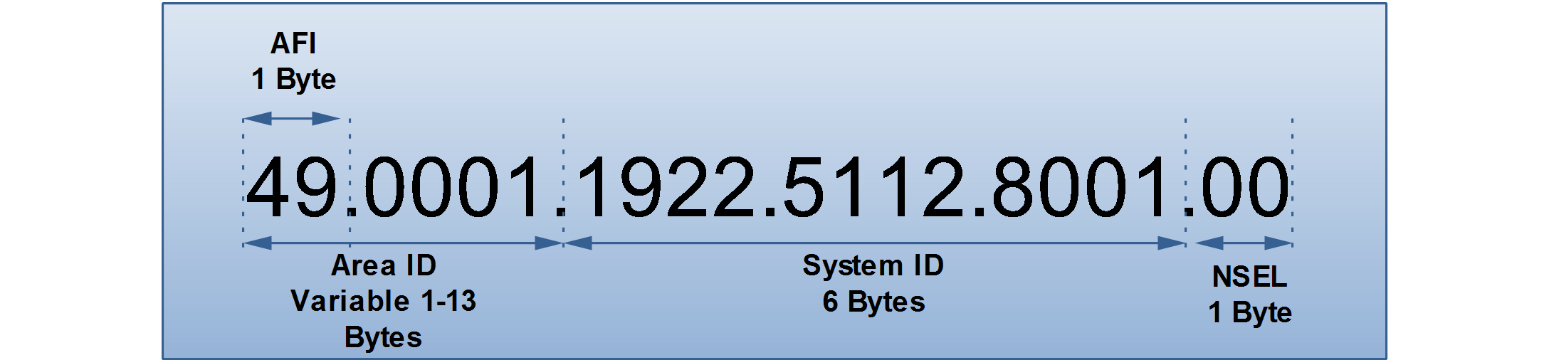
Всеки IS-IS маршрутизатор разпространява информация за своят статус (състояние на интерфейси, достижими съседи, цени на всички интерфейси) към всички други маршрутизатори използващи Link-State PDU (LSP) съобщения. Всеки маршрутизатор използва получените съобщения да изгради идентична база от данни, която описва пълната топологична информация на автономната система. От тази база данни, всеки маршрутизатор изчислява пътища, които записва в своята маршрутизираща таблица използвайки Shortest Path First (SPF) алгоритъма. Тази маршрутизираща таблица съдържа „next-hop“ адреса и изходящият интерфейс до всяка дестинация в мрежата.

Този протокол изгражда йерархична структура на мрежата, като се разделя маршрутизиращият домейн на отделни части, наречени зони (areas). Всяка зона притежава уникален за нея адрес, чрез който тя се отличава от другите зони в мрежата. Във всяка зона има Intermediate System (маршрутизатор), който функционира на различни нива или на две нива едномвременно.

* **Level 1 Intermediate System** – маршрутизатор, който обменя маршрутизираща информация само с други маршрутизатори, функциониращи на първо ниво. Тези устройства притежават информация само за пътищата в зоната в която се намират, но не и за пътища в други зони.
* **Level 2 Intermediate System** – маршрутизатор, който обменя маршрутизираща информация само с маршрутизиатори, функциониращи на второ ниво, които са от други зони или от същата зона.
* **Level 1 & Level 2 Intermediate System** – маршрутизатор, който притежава информация за пътища в зоната, в която се намира, и за пътищата в други зони. Този тип маршрутизатори работещи на това ниво се използват предимно за Area Border Routers (ABRs).

На Фиг.1.10 е показана йерархична структура изградена от IS-IS протокола.

*****Фиг.1.10. Йерархична структура изградена от IS-IS*

Всеки маршрутизатор в IS-IS процеса, се нарича Network Service Access Point (NSAP) и се индентифицира чрез Network Entity Title (NET) адрес. Неговата дължина варира от 8 до 20 октета и се съдържа от три части. На Фиг.1.11 е показан форматът на NET адреса.

*Фиг.1.11. Формат на NET адрес в IS-IS*

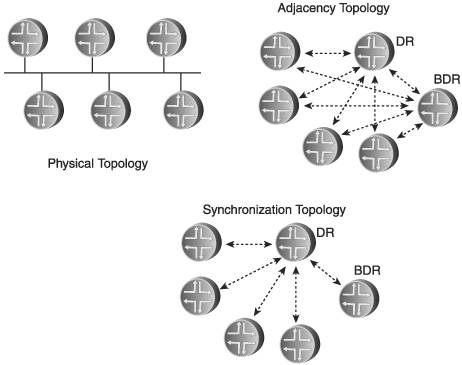
* **Area address (Area ID) –** това поле може да 1 до 13 октета дълго и бървият байт е резервиран за AFI.
* **System ID –** това поле е 6 октета дълго и служи за идентификация на един маршрутизатор. Всеки маршрутизатор трябва да има уникално System ID за своята зона и ниво на което функционира.
* **NSEL –** това поле е с постоянна стойност 00.

Междинните системи обменят маршрутизираща информация използвайки Protocol Data Units (PDUs), които биват три вида:

* **IIHs** – Intermediate System-to-Intermediate System Hello PDUs, биват обменяни между IS съседите, за поддържане на съседство. IIHs включват системното ID (System ID) на подателя, адресът на неговата зона и всичките му IS съседи. Съществуват три вида IIHs:
  + Point-to-Point IIHs – те се пращат помежду директно свързани междинни системи.
  + Level-1 LAN IIHs – те се изпращат на „multiaccess“ междинни системи, когато IS подателя функционира на първо ниво (Level 1).
  + Level-2 LAN IIHs – те се изпращат на „multiaccess“ междинни системи, когато IS подателя функционира на второ ниво (Level 2).
* **LSPs** – една междинна система генерира Link-State PDUs (LSPs) за да сподели своите IS съседи и дестинации до които има свързаност.LSP може да се идентифицира като:
  + **System ID** на междинната система, която е генерирала LSP
  + **Pseudonode ID** – тази стойност е винаги 0, освен когато LSP е pseudonode LSP.
  + **LSP number** – от 0 до 255
  + **32-битово sequence number –** 32-битово последователно число, което се увеличава, когато нова версия на LSP е генерирана.
* **SNPs** – Sequence Number PDUs, съдържат обобщено описание на едно или повече LSPs. Съществуват два типа SNPs за Level 1 и Level 2:
  + **Complete Sequence Number PDUs (CSNPs)** – Този тип съобщения са периодични и съдържат пълна информация за всички състояния на връзките, които са налични в базата от данни на протокола.
  + **Partial Sequence Number PDUs (PSNPs)** – Чрез това съобщение се изисква допълнителна информация за липсващите състояния на връзките.

В “multiaccess” мрежите всички устройства споделят една преносна среда. Ако всички междинни системи изпращат своите съобщения до всичките си съседи, служебният трафик, който бива генериран, значително нараства и намалява производителността на мрежата. Това води до необходимост от избор на обозначена междинна система (Designated Intermediate System - DIS), която е отговорна за поддържането на синхронизирана база от данни във всички останали междинни системи в тази мрежа. Тя се избира на базата на интерфейсен приоритет. В случай, че всички междинни системи имат еднакъв интерфейсен приоритет, се избира тази с най-висок физически адрес (MAC адрес).

На Фиг.1.12 са показани топологии, при които е необходимо да се избере DIS.

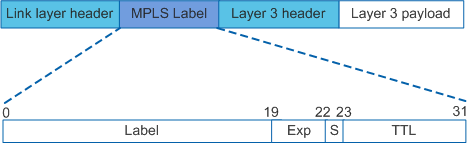


*Фиг.1.12. Видове топологии, които изискват назначаването на DIS*

**1.3. Multiprotocol Label Switching (MPLS)**

MPLS е технология за мултипротоколно етикетно комутиране (Multiprotocol Label Switching) е метод, при който информацията се пренася с помощта на етикети (labels) през мрежата. Първоначално MPLS бил замислен като технология, с която се позволявало пакетите да се маршрутизират по-бързо и ефикасно, но с годините тази технология се превръща в средство за предоставяне на различни услуги на клиентите. Една от най-известните услуги, която може да се предостави посредством MPLS, е виртуална частна мрежа на слой 3 от OSI модела (Layer 3 Virtual Private Network). При MPLS мрежите, разчитането на заглавната част става само веднъж, когато поакета навлезе в MPLS домейна.

**1.3.1. Labels**

MPLS използва етикети (labels), с които пакетите се маркират и се добавят към определен еквивалент за предаване на класове - Forwarding Equivalence Class (FEC). Класовете представляват съвкупност от пакети, притежаващи еднакви параметри, например еднакви дестинации или политики. Етикетите се задават от самите маршрутизатори и са с локално значение. Поставянето на етикети става статично или динамично, посредством протокол, а възлите (nodes) обменят информация за етикетите по определен път с комутация на етикети (Label Switched Path – LSP). На Фиг.1.13 e показан форматът на един етикет. Дължината му е 4 байта и се слага в момента, в който пакета навлезе в MPLS мрежата.

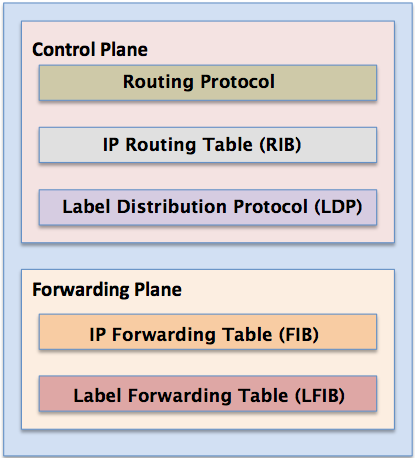
*Фиг.1.13. Формат на MPLS етикет*

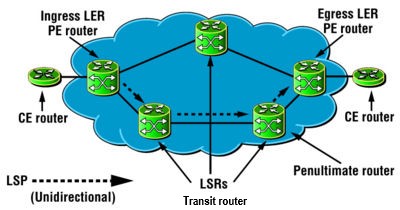
* **Label** – в това поле се записва стойността на етикета с дължина 20 бита
* **Exp** – това поле е с дължина 3 бита и се ползва за Class of Service (CoS)
* **S** – това поле указва, че етикетът е последният, използван върху пакета, с дължина 1 бит.
* **TTL** – Time to Live – полето указва времето на живот на етикета, с 8 бита дължина

Етикетът се добавя между заглавните части на слой 2 и слой 3 на пакета. Комутирането става на базата на тези етикет, като те се подреждат спрямо класа на еквивалентност, към когото са добавени. Пакетите може да се добавят към различни видове еквивалентни класове при всеки маршрутизатор от мрежата, ако ситуацията го изисква. Така се получава етикетният стек, като с най-долния, последен етикет се разбира накъде да тръгне изходящият трафик, а другите над него са за комутацията на пакетите в MPLS мрежата.

**1.3.2. Label Switching Router**

Това е маршрутизаторът, на който има конфигуриран MPLS и може да обработва етикети и техните стойности. В LSR може да се разглеждат две равнини. Процесът на маршрутизация е разделен на две части. Първата равнина е равнината за препредаване (Forwarding Plane) или също така известна като равнина на данните (Data Plane), като тя е на хардуерна основа и се управлява от втората равнина – управляваща равнина (Control Plane). В нея се намират маршрутизиращите протоколи и методите за разпространение. И в двете равнини се намира база от данни – Label Forwarding Information Base (LFIB), в която се съдържат етикетите и информацията за тях. В равнината за препредаване на информация присъства и Forwarding Information Base (FIB), което е база от данни съдържаща адресите. Освен равнината за препредаване и управляващата равнина, съществува и менажираща равнина (Management Plane). В нея се съдържат конфигурацията на маршрутизатора, въведена от мрежовият администратор, чрез която се управляват маршрутизиращите протоколи на управляващата равнина. На Фиг.1.14 е означена структурата и разположението на равнините в един Label Switching Router от MPLS домейна.

*******Фиг.1.14. Структура на равнините в LSR*

****В MPLS средата съществуват четири вида маршрутизатори, според разположението си в домейна, които пренасочват идващите пакети по различните пътища, чрез етикети. На Фиг.1.15 са означени видовете MPLS маршрутизатори.

*Фиг.1.15. Видове MPLS маршрутизатори в MPLS домейна*

* **Ingress Router** – маршрутизатор, който е входна точка в пътя на комутация на етикети (LSP). Те са първите маршрутизатори, през които минава входния трафик, когато влиза в MPLS домейна. Пакетите се енкапсулират във входящите маршрутизатори като се слага допълнителна информация в заглавната част на пакета, чрез операцията за добавяне на етикети (Label Push Operation).
* **Transit Router** – маршрутизатор, намиращ се между входният и изходният за MPLS домейна LER (Label Edge Router). Има възможност до 253 маршрутизатора в един Label Switched Path (LSP). Ограничението е поради имплементираното поле за живот в заглавната част на етикета, което е 8 бита. Това означава, че в един път може да има 255 маршрутизатора – 253 транзитни, един входен и един изходен. Функцията на транзитния маршрутизатор е да приеме етикета, да прочете заглавната му част, да провери за съответствия в MPLS таблицата за етикети, да смени входния етикет с изходен и да намали полето за живот (TTL) с 1, като го пренасочи към следващото устройство по пътя. Извършва операцията за смяна на етикети (Label Swap Operation).
* **Penultimate Router** – последният транзитен маршрутизатор преди изходният маршрутизатор. Неговото предназначение е да премахне етикета (Label Pop Operation). След което пакета се препраща към изходния маршрутизатор от LSP без MPLS информация в заглавната част на пакета. Полето за живот на пакета се намалява с 1, а препращането става с последния сложен етикет за еквивалентен клас, преди най-долния етикет на пакета.
* **Egress Router** – последният маршрутизатор в LSP. Всеки път за комутация на етикети трябва да има изходен маршрутизатор. Той е изхода от MPLS домейна и неговата задача е да приеме входния трафик, да потърси съвпадение на адреса на дестинация в своята маршрутизираща таблица и да го препрати на следващия маршрутизатор, който е отвъд MPLS домейна.

**1.3.3. Label Switched Path (LSP)**

Това е път с комутация на етикети, създаден от MPLS, който е само с една посока. За един пакет, пътят представлява последователност от маршрутизатори. Еднопосочните пътища дават пълен контрол върху преноса на информация. Всеки път задължително има входен и изходен маршрутизатор. Създаването на път за комутация на етикети става по два начина:

* **Статично създаден път** – статично създадените пътища се конфигурират от мрежовият администратор, като той определя какви етикети да се задават и на къде се препраща трафика. Тази операция наподобява на конфигурирането на статични пътища за пренос на IP пакети. Статично създадените пътища за комутация на етикети заемат по-малко хардуерни ресурси и нямат нужда от постоянна поддръжк, но съществува недостатъка, че ако мрежата е голяма, има възможност да се получат грешки при конфигурирането на статичните пътища.
* **Динамично създаден път** – създаването на път по този начин изцяло зависи от сигнализиращ протокол за изграждане на пътища. Конфигурира се само входящият маршрутизатор (Ingress router) с нужната информация за създаване на път. Другите маршрутизатори в MPLS мрежата получават съобщения от първия, с които се изгражда пътя до дестинацията. Един от най-използваните протоколи е протоколът за Label Distribution Protocol (LDP) – протокол създаден изцяло за функционалността на MPLS.

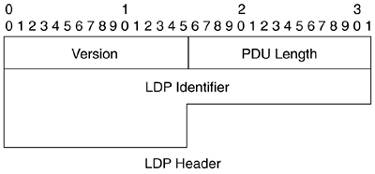
Често, когато се използва MPLS в мрежата на интернет доставчика, има възможност да се създават виртуални частни мрежи (VPN), в които могат да се маршрутизират както частни, така и публични адреси отделно от останалия трафик. При използването на частни адреси, често се стига до повтаряне на едно и също адресно пространство (address overlapping). Затова при конфигурирането на една виртуална частна мрежа се използва Virtual Routing and Forwarding Tables (VRF). С тях доставчиците разграничават трафика на клиентите и когато имат повтарящи се мрежови адреси се използва разделител на пътища Route Distinguisher (RD). Това поле е с дължина 6 байта и се поставя преди полето на частния адрес. По този начин се предоставя уникален частен адрес. Има два формата, по който се създават тези уникални адреси:

* **as-number:number** – с този формат, мрежовият администратор разполага с поле от 2 байта за номер на автономната система и 4 байта за пореден номер.
* **ip-address:number** – с този формат, мрежовият администратор разполага с поле от 4 байта за номер на автономната система и 2 байта за пореден номер.

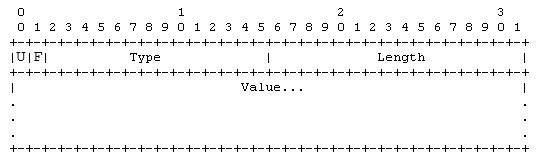
Форматът се задава от мрежовият администратор, а маршрутизаторът, който е входен за MPLS домейна (Ingress router), добавя тази информация към получения път от клиентите. Така получения IPv4 адрес се превръща във VPN-IPv4 адрес. Изходният маршрутизатор (Egress router) превръща VPN-IPv4 адресът в обикновен IPv4 и го предава на клиента.

**1.3.4. Label Distribution Protocol (LDP)**

Протоколът за разпределяне на етикети – Label Distribution Protocol (LDP) е разработен специално за нуждите на една MPLS мрежа. LDP работи върху вътрешен маршрутизиращ протокол. Създаването на път през MPLS мрежата чрез този протокол се базира на последователност от различни процеси и обмен на съобщения, като всеки създаден път се асоциира с определен еквивалент за предаване на класове (Forwarding Equivalence Class – FEC). Всеки път до определена мрежа притежава уникален етикет, който може да бъде зададен статично от мрежовият администратор или динамично, като се избира етикет на случаен принцип от съвкупност от етикети. LDP започва да функционира автоматично след като бъде конфигуриран MPLS от мрежовият администратор. Тогава LDP започва да изпраща съобщения на всички активни интерфейси с функциониращ MPLS процес, за да обмени информация за етикети с всички маршрутизатори в MPLS домейна. Всички LDP съобщения започват с LDP заглавна част, която е последвана от TLV (type, length, value) кодировка на съобщенията, като винаги има едно задължително TLV поле и има възможност за допълнителни TLV полета. На Фиг.1.16 e показан форматът на LDP PDU и на LDP съобщението.

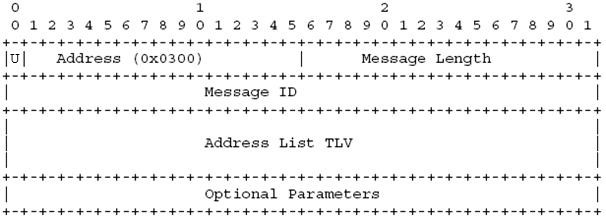
*******Фиг.1.16. Формат на LDP PDU и на LDP съобщението*

* **Version –** полето е с дължина 2 байта и обозначава версията на протокола.
* **PDU Length –** полето е с дължина 2 байта и обозначава дължината на даденото PDU.
* **LDP Identifirer –** полето е с дължина 6 байта и обозначава „пространството“ в което се обменят етикетите.

****TLV полетата представляват съобщенията на протокола. Като има задължителни и незадължителни полета, в зависимост от обменящите се съобщения. На Фиг.1.17 е показана общата структура на TLV полето.

*Фиг.1.17. Структура на TLV поле*

* **U –** поле с дължина 1 бит, което служи за разпознаване на TLV поле, тоест съобщението, което е изпратено. Ако то е непознато има възможност да се върне при уведомление (при 1), но може и да не се върне уведомление (при 0).И в двата случая цялото съобщение се игнорира.
* **F –** поле с дължина 1 бит, което служи да обозначи дали съобщението ще се препрати на следващите маршрутизатори или не. Полето се използва, ако има 1 при „U“ полето.
* **Type –** поле с дължина 14 бита и обозначава вида на съобщението.
* **TLV Length –** поле с дължина 16 бита и обозначава дължината на даденото TLV.
* **Value –** поле с дължина 32 бита и представлява стойността на LDP съобщението.

****Заглавната част на LDP заедно с TLV полетата образуват Protocol Data Unit (PDU). Преращането на съобщения се постига, като се изпращат PDU през TCP сесия на порт 646. По този начин се осигурява двупосочна връзка и сигурност при предаване на информацията. Единствено Hello съобщението се изпраща през UDP порт 646 с multicast адрес до всички маршрутизатори до тази подмрежа – 224.0.0.2. Всяко едно PDU може да пренася по няколко LDP съобщения. След като е установена сесията LDP изпраща съобщения за етикетна информаци, всякакви промени по топологията или за приключване на сесията. На Фиг.1.18 е означена цялостната структура на едно съобщение от LDP.

*Фиг.1.18. Цялостна структура на LDP съобщение*

* **U –** поле с дължина 1 бит, което служи за разпознаване на TLV поле, тоест съобщението, което е изпратено. Ако то е непознато има възможност да се върне при уведомление (при 1), но може и да не се върне уведомление (при 0).И в двата случая цялото съобщение се игнорира.
* **Message Type –** поле с дължина 15 бита и обозначава вида на съобщението.
* **Message Length –** поле с дължина 16 бита и обозначва дължината на даденото TLV
* **Message ID –** поле с дължина 32 бита и представлява номера на LDP съобщението.
* **Mandatory Parameters (TLVs) –** поле с променлива дължина, в което се записват нужните параметри на LDP съобщение. То може и да е празно, тъй като не всички съобщения имат нужда от параметри.
* **Optional Parameters (TLVs) –** поле с променлива дължина, в което се записват допълнителните параметри на LDP съобщение.

Съществуват четири категории съобщения, които определят работата на протокола:

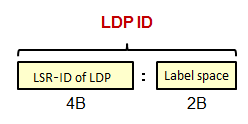
* **Discovery** – тип съобщения, с които се установява връзка със съседни маршрутизатори, на които функционира MPLS процес. Изпращат се на multicast адрес – 224.0.0.2.
* **Session** – тип съобщения, с които се установява, поддържа и затваря сесия с маршрутизатори, използващи LDP протоколa – „LDP peers“. С това съобщение се договарят параметрите и настройките за предаване на етикетна информация.
* **Advertisement** – тип съобщения, чрез които се създават, променят и изтриват етикети за отделните FECs.
* **Notification** – тип съобщения, които се използват, за да се предаде информация относно грешки. Задължителен параметър е Status (TLV), който обозначава събитието, което се е случило за да се появи грешка. Съществува възможност да се изпраща и допълнителна, разширена информация относно събитието в Optional Parameters (TLV) полето. Ако съобщеният проблем е от голямо значение, то TCP сесията между маршрутизаторите има възможност да бъде затворена, в следствие на което се отхвърля цялата информация.

В тези четири категории има общо 12 съобщения на LDP:

* **Hello message** – това е първото съобщение, което се изпраща след установяване на TCP сесията. Изпраща се на UDP порт 646. Съдържа едно задължително поле с няколко параметъра, от които най-важен е времето на задържане (Hold time). В задължителните съобщения се обозначва какъв да е видът на съобщенията – дали да са до определена дестинация или да се изпращат на multicast адреса.
* **Basic discovery** – това е обикновеният начин на намиране на съседен маршрутизатор. Изпращат се периодично Link Hello съобщения на UDP порт 646 с multicast адрес 224.0.0.2. Този, който получи съобщението се превръща в съседен маршрутизатор.
* **Extended discovery** – при този начин за намиране на съседни маршрутизатори, периодично се изпращат „Targeted hello“ съобщения с определен мрежови адрес. Маршрутизаторът, с когото се търси съседство, ако отговори на съобщението, той също изпраща Targeted hello съобщение и по този начин се установява съседство.
* **Initialization message** – с това съобщение се изпращат параметрите на връзката за предаване на етикетна информация. Съществува само едно поле със задължителни параметри. Чрез това съобщение се изпраща версията на протокола, времената на KeepAlive съобщенията. Също така се изпраща и начинът на предаване на етикетна информация.
* **KeepAlive message** – съобщението служи за периодично предаване на съобщение, което да нулира KeepAlive, с цел да не се затвори установената TCP сесия.
* **Address message** – съобщение, което служи на локалния маршутизатор за предаване на конфигурираните адреси на другите маршрутизатори.
* **Address Withdraw message** – съобщение, което служи на локалния маршрутизатор да предаде своите премахнати адреси, които са били конфигурирани на интерфейсите на другите маршрутизатори. Съдържа задължително поле с лист от адресите на интерфейсите, които са премахнати.
* **Label Mapping** – чрез това съобщение се изпращат етикетите, зададени към определен FEC. Притежава две задължителни полета – FEC TLV, в което се записва компонента на определения клас и Label TLV, в което се записва самият етикет.
* **Label Request** – съобщение, което се използва за заявяване на етикет от съседен маршрутизатор. Притежава задължително поле – FEC TLV, в което се записва компонента на определения клас.
* **Label Abort Request** – съобщение, което се използва за отказване от заявката на етикет от съседен маршрутизатор. Притежава две задължителни полета – FEC TLV, в което се записва компонента на определения клас и Label Request Message ID TLV, в което се записва номера на съобщението, чрез което се иска етикет.
* **Label Release** – съобщение, което се използва за да се укаже, че обявеният етикет ще се премахне. Притежава задължително поле – FEC TLV, в което се записва компонента на определения клас.
* **Label Withdraw** – съобщение, което се използва за да се премахне даден етикет, в случай, че класът на еквивалентност не се ползва вече. Притежава задължително поле – FEC TLV, в което се записва компонента на определения клас.

LDP функционира в два типа пространства (label spaces) и се осъществява по два начина по които може да се обмени етикетна информация:

* **Per interface label space –** обмяната на етикети се осъществява на базата на директно свързани устройства. Етикетът може да се използва само между двете свързани устройства. Приложимо е при мрежи с комутация на пакети.
* **Per platform label space –** обмяната на етикети се осъществява чрез всички видове устройства, работещи в мрежата. Интерфейсът, на който се разменя етикета, не е от значение. Приложимо е при мрежите с множествен достъп (Broadcast multiaccess).

При имплементацията на LDP се използва идентификатор, с който се обозначава пространството за обмяна на етикети в мрежата. Той се състои от две полета и е с дължина 6 байта. На Фиг.1.19 е показан форматът на LDP идентификатора.

*Фиг.1.19. Формат на LDP идентифиркатор*

* **LSR-ID –** поле с дължина 4 байта, указващо уникалният номер на маршрутизатора, на който е конфигуриран MPLS. Той трябва да е уникален за всяко етикетно пространство.
* **Label space –** поле с дължина 2 байта, което указва номера на етикетното пространство в което се намира даденият етикет.
  1. **Virtual Private Network (VPN)**

Виртуалните частни мрежи (Virtual Private Networks), в качеството си на технология, са метод, чрез който за определен клиент се „заделя“ мрежа от обща физическа мрежова инфраструктура. Но по този начин целият трафик на клиентите, използващи предоставеното от интернет доставчика оборудване, бива отделен от трафика на останалите клиенти. На клиентите им се предоставя връзка под формата на тунел (tunnel). За да се осъществи VPN тунелирането се използват различни тунелиращи протоколи. Най-голямото предимство на виртуалните частни мрежи е сигурността, защото трафикът в тунела е криптиран. Съществуват два вида виртуални частни мрежи:

* **Site-to-Site VPN** – при този тип VPN, от двете страни на тунела са разположени устройства, на които има конфигурация за частната мрежа. Крайните устройства, свързани към този тунел, всъщност нямат налична информация, за това че се намират във виртуална частна мрежа. Те изпращат и получават нормален трафик през шлюз на виртуалната частна мрежа (VPN Gateway), който е отговорен за енкапсулацията и декапсулацията на стандартния трафик. Този тип VPN позволява да се свързват отделни мрежи.
* **Remote Access VPN** – този тип VPN се базира на клиент/сървър архитектурата. Често е използвано в корпоративните среди. Позволява се на единични устройства да се свързват сигурно към дадена мрежа. На всяко крайно устройство трябва да има инсталиран софтуер, чрез който потребителят да изпраща своята информация. Тя бива енкапсулирана или декапсулирана от приложението преди да се изпрати към VPN Gateway, който изпраща информацията на съседен VPN Gateway, както при Site-to-Site VPN.

Когато доставчик на услуги отговаря за предоставянето на VPN (Provider Provisioned VPN), обикновено посредством MPLS, тогава той може да предостави VPN на два слоя на OSI модела – Каналния слой и Мрежовия слой. В тези две категории VPN попадат няколко VPN технологии:

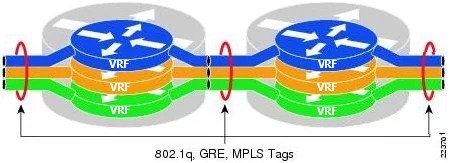
* **Layer 2:**
  + **Point-to-point (P2P)** – в тази категория се включва Virtual Private Wire Service (VPWS). Това е услуга, която е базирана на point-to-point архитектурата. Два маршрутизатора на потребителя са свързани посредством виртуална връзка, направена през мрежа с комутация на пакети. В тази категория попада и „псевдо“ връзката – Pseudo wire (PW), което представлява виртуална point-to-point връзка през мрежа с комутация на пакети, като през псевдо връзката може да премине всякаква технология от слой 2 на OSI модела.
  + **Point-to-multipoint (P2M) –** в тази категория се включват Virtual Private LAN Service (VPLS) и IP-Only Private LAN Service (IPLS). Първата VPN технология представлява създаването на функционалността на локалната мрежа (LAN) върху мрежа с комутация на пакети. Има възможност за свързване на няколко LAN сегмента с цел да работят като един. Изпращането на информация става на базата на физически адреси (MAC адреси). IPLS технологията се използват маршрутизатори и крайни устройства, като в създадената виртуална частна мрежа се пренасят единствено IP пакети.
* **Layer 3:**
  + **PE-Based –** при създаването на виртуални частни мрежи от този тип, оборудването на доставчика създава и поддържа виртуалната частна мрежа. Потребителите, свързани към този тип мрежи биват изолирани един от друг. Мрежата на доставчика знае за VPN трафика, който се пренася през тунели. Към тази категория се включват Virtual Router (VR) и BGP/MPLS IP VPNs технологиите. И при двете технологии се поддържа отделен виртуален метод за пренасяне на VPN информация.
  + **CE-Based –** при създаването на виртуални частни мрежи от този тип, оборудването, предоставено от доставчика на услуги, не съдържа информация относно създаденият VPN. Цялата информация се обработва от оборудването на клиентите, а мрежата на доставчика служи само за пренос, като тя няма представа какъв е пренасяният трафик. IPSec е технология, която може да се включи към тази категория.

Независимо от използваната технология за предоставяне на VPN, има два типа устройства, които са задължителни за създаване на една виртуална частна мрежа:

* **Provider Edge –** това са устройствата, намиращи се накрая на мрежата на доставчика. Те имат възможността да обменят информация с устройствата на клиента. Това са или маршрутизатор или комутатор.
* **Customer Edge –** това са устройствата, намиращи се накрая на мрежата на потребителя. Те имат възможността да обменят информация с устройствата на доставчика на услуги. Това са или маршрутизатор или комутатор.
  1. **Virtual Routing and Forwarding (VRF)**

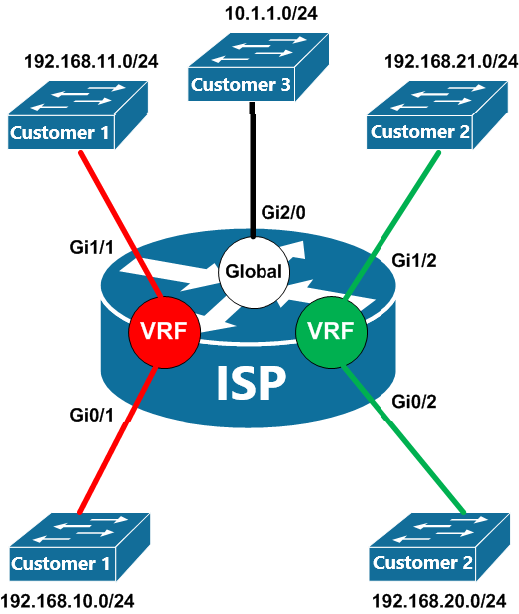
При нуждата от IP мрежи, които са изолирани и отделни, като че ли са използвани от различни компании, организации или отдели, ще се използват няколко IP мрежи изградени от отделни маршрутизатори, които не са свързани помежду си. Те ще имат осъществена Layer 2 или Layer 1 свързаност, но не са свързани по Layer 3 и не образуват мрежа.

Мрежовата виртуализация (Network virtualization) позволява на един маршрутизатор да притежава множество маршрутизиращи таблици. Това подобрява функционалността, чрез разрешаването на пътищата да бъдат сегментирани без използването на множество устройства. Понеже трафикът е автоматично изолиран от останалия, VRF подобрява нивото на сигурност и успява да елиминира нуждата от криптиране и автентикация. Интернет доставчиците често се възползват от предимствата на VRF, за да създадат отделни виртуални частни мрежи (VPNs) за своите клиенти, следователно тази технология също бива наричана „VPN Routing and Forwarding“. Глобалната маршрутизираща таблица съдържа всички IP интерфейси, които не са част от виртуална мрежа. На Фиг.1.20 е показана свързаност на два маршрутизатора, ползващи VRF, чрез който се пренася 802.1q, GRE и МPLS трафик.

****

*Фиг.1.20. Свръзаност на два маршрутизатора ползващи VRF*

На Фиг.1.21. е указана мрежова топология с функционираща VRF технология, предоставяща изолирана от глобалния трафик свързаност мрежду клиентите посредством отделни VRF таблици за всеки клиент.

*****Фиг.1.21. Мрежова топология свързваща два отделни клиента посредством VRF*

* 1. **Quality of Service (QoS)**

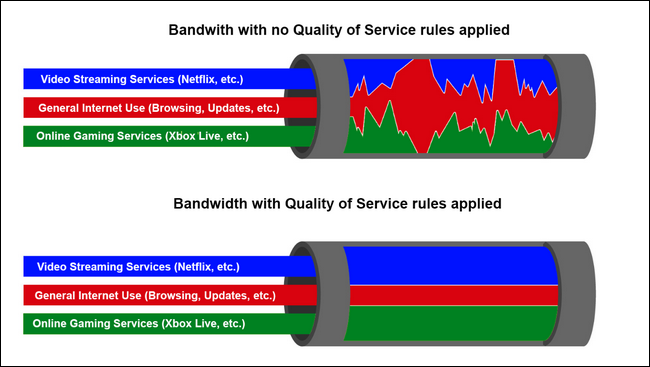
Контролът върху качеството на услугите (Quality of Service) гарантира, че даден трафик ще достигне до дестинацията си с минимални или без загуби на пакети. QoS позволява да се приоритизира трафик правилнно, за да може видеопредаването във видеоразговорите през Интернет пространството, както и интернет телефонията, да функционират правилно, тъй като те изискват най-много от ресурсите на мрежата. Трафикът от различните услуги се добавя в опашка (queue), в която се предават повече пакети за приоритетните приложения. Големината на трафика изцяло зависи от скоростта на вързката. QoS се прилага при устройства, където има вероятност от задръстване на мрежата (congestion). Това са случаите на агрегиране на трафик от множество крайни устройства, преминаване от връзка с по-голяма скорост на предаване към такава с по-малка или преминаването от локална мрежа към глобална мрежа. С контрола върху качеството на услугите, пакетите се предават без да има загуба от потока с данни и се намалява закъснението (delay) и трептенията (jitter), тъй като пакетите са подредени според приоритета им.

За използването на QoS, мрежовата архитектура се нуждае от опашки, в които да подрежда трафика според приоритета им. С опашките трафикът се приоритизира, буферира и дори се пренарежда. Има няколко вида алгоритми, с които се създават опашките, но най-използваните са:

* **First-In First-Out (FIFO) –** това е най-лесният алгоритъм за създаване на опашка. Пакетите се предават така, както са пристигнали при устройството. Няма подреждане или приоритизиране на трафика. Има само една опашка и всички пакети вътре в нея се обслужват еднакво. FIFO е най-бързият метод за създаване на опашка и се използва при връзки с голяма скорост, където има минимално времезакъснение и количеството на задръствания е минимално.
* **Weighted Fair Queuing (WFQ) –** това е метод, при който трафикът се разпределя автоматично по равно спрямо скоростта на връзката. Трафикът, който е най-важен и зависим от времезакъснения се слага отпред на опашката, с цел намаляване на времезакъснението. Приоритизира се потокът от данни с по-малка големина пред този с по-голяма. С този метод трафикът се класифицира, спрямо информацията от заглавните части на пакетите. Недостатък е, че използването на този метод ограничава тунелирането и криптирането на данните, тъй като те променят стойностите, нужни на алгоритъма.
* **Class-Based Weighted Fair Queuing (CBWFQ) –** това е метод, при който трафикът се разпределя по равно, както при WFQ, но тук администраторът определя класовете на потоците от данни. Задават се критетрии, спрямо които пакетите се обработват. Всеки клас има FIFO опашка, която се подрежда спрямо приоритета в обща опашка.
* **Low Latency Queuing (LLQ) –** надгражда CBWFQ, като се въвежда стриктна приоритизация на опашката (Strict priority queuing). Това използва трафик, който има нужда от минимални времезакъснения като видео трафик или аудио трафик, да бъде приоритизиран и сложен на първо място в опашката. Останалите FIFO опашки се подреждат спрямо отредената им скорост от връзката.

Когато се имплементира QoS върху една мрежа, то контролът се прави на базата на модели за политики. Те биват два типа:

* **Integrated Services (IntServ) –** при този модел, качеството на услугите се контролира от протокол, който съобщава на маршрутизатори пакетите, които имат нужда от специално обслужване. Използва се Resource Reservation Protocol (RSVP), за да се сигнализира за нуждите на мрежата да приложи различни методи за контрол върху качеството на трафика. Моделът се използва предимно при мрежи, където има трафик в реално време, тъй като този тип трафик се приоритизира и доставя с най-малко загуби. Недостатък е, че използва много от ресурсите на устройствата, поради постоянно сигнализиране за състоянието на връзките. Моделът не е мащабируем.
* **Differentiated Services (DiffServ) –** при този метод трафикът се класифицира спрямо полето „Type of Service“ от заглавната част на интернет протокола. В сравнение с IntServ, методът е сравнително по-лесен за имплементация и е по-мащабируем. Чрез него се гарантира за аудио и видео трафикът, че ще има минимални времезакъснения. Не се сигнализира от приложенията за нуждите от качество на услугата. Използва се класификация, за да се раздели трафикът спрямо нуждите на мрежата.

При използването на QoS, трафикът често може да се ограничи до определена скорост или да се оформи (shaped) до такава. Това се постига с policer/shaper модел. Ограничението (policing) на трафика до определена скорост на предаване може да доведе до големи пикове от информация по време на нейнто предаване, защото трафикът от различните приложения, преминаващ през мрежата, е с различна големина. Ако се надвиши ограничението, което е поставено на връзката, се стига до изпускането на трафик или неговото приоритизиране (remarking) до по-крайни места в редиците. Оформянето (shaping) до определена скорост на предаване позволява трафикът да се разпределя на равни по големина редици и да се изпраща през равномерни интервали от време. В случай, че съществува пакет с информация надвишаваща максималната допустима за връзката, то той се добавя към следващата редица, приготвена за изпращане. Ако не се използва метод за структуриране на редиците, тогава информацията, изискваща определени времеви параметри (например VoIP приложения), става негодна за използване. За да работи всичко нормално, без загуба на пакети по трасето, се използва CBWFQ или LLQ, в зависимост от пренасяния трафик. На Фиг.1.22 е указана функционалността на методите за ****структуриране в сравнение с работата без тях.

*Фиг.1.22. Сравнение на функционалността с методи за структуриране и без методи за структуриране при Quality of Service*

**Втора** **глава: Проектиране на физическата реализация на MPLS VPN мрежова архитектура**

**2.1. Основни изисквания към мрежовата топология**

**2.1.1. Изграждане на MPLSVPN архитектура за две различни компании**

* Изграждане на мрежовата топология на базата на Cisco 7200 маршрутизатори.
* Всяка компания трябва да има офис и клон с по един маршрутизатор, който участва като „Customer Edge” маршрутизатор в MPLSVPN мрежовата архитектура.
* Мрежата на интернет доставчика трябва да се състои от 10 маршрутизатора:
  + 2 служещи като Route Reflector.
  + 4 служещи като „Provider Edge” маршрутизатори.
  + 4 служещи като „Core” (Provider) маршрутизатори, които установяват свързаността в мрежата на интернет доставчика.
* Всички адреси на интерфейси в мрежата на интернет доставчика трябва да са публични.

**2.1.2. Използване на протокол LDP за дистрибуция на етикети**

* Конфигуриране на Loopback адрес на маршрутизаторите в мрежата на интернет доставчика.
* Стартиране на LDP протокола чрез командата **mpls label protocol ldp**.
* Конфигуриране на **router-id** на LDP протокола. За **router-id** се използва конфигурирания Loopback адрес на маршрутизатора (Loopback 1).

**2.1.3. Употреба на IS-IS протокол в MPLS мрежата**

* Стартиране на IS-IS процес в мрежата на интернет доставчика, чрез използване на командата **router isis** в глобален конфигурационен режим.
* Конфигуриране на **router-id** за IS-IS процеса.
* Конфигуриране на „net” атрибут, който указва зоната в която функционира IS-IS процеса.
* Задаване на пасивен интерфейс Loopback 1.
* Конфигуриране на тип ниво на което функционира IS-IS процеса, в случая – ниво 2.
* Стартиране на IS-IS процеса на интерфейсите, които са свързани в MPLS мрежата.

**2.1.4. Използване на BGP и MPBGP за реализация на динамична маршрутизация**

* Стартиране на BGP процес с номер на автономна система:
  + За клиентите се използват частни номера на автономни системи.
  + За интернет доставчика се използва публичен номер на автономна система.
* Конфигуриране на **router-id** на BGP протокола.
* Конфигуриране на eBGP свързаност между СЕ (маршрутизаторите на клиентите) и РЕ (маршрутизаторите на интернет доставчика).
* Обявяване на директно свързаните пътища на СЕ маршрутизаторите, в случая техните Loopback адреси.
* Конфигуриране на iBGP свързнаост между PE и Route Reflector маршрутизаторите в мрежата на интернет доставчика.
* Създаване на **vpnv4** адресно семейство в BGP процеса на маршрутизаторите.
* Задаване на РЕ маршрутизаторите ролята на Route Reflector клиенти.
* Асоцииране на конфигурираните VRF за отделните клиенти към процеса на BGP протокола.
* Обявяване на мрежите РЕ – СЕ на всички РЕ маршрутизатори.
* Конфигуриране на **route-map** за приоритизиране на път към автономните системи на двата отделни клиента.

**2.1.5. Използване на Route Reflector в MPLS мрежата**

* Създаване на два маршрутизатора в мрежата на интернет доставчика, които играят ролята на Route Relfector.
* Конфигуриране на Loopback адрес на двата маршрутизатора.
* Конфигуриране на iBGP свързнаост между всички PE маршрутизатори и тези два маршрутизатора.
* Създаване на **vpnv4** адресно семейство в BGP процеса на маршрутизаторите.
* Задаване на всички РЕ маршрутизатори, свързани към тези два маршрутизатора, чрез iBGP, ролята Route Reflector клиент.

**2.1.6. Осигуряване на резервираност на всеки от клиентите на ниво доставчик на услугата**

**2.1.7. Внедряване на политика за качество на услугата - Quality of Service**

**2.1.8. Симулация на мрежовото решение на GNS3**

**2.2. Описание на мрежовата топология**

**2.3. Адресация на мрежовата топология**

**Трета** **глава: Симулация на MPLS VPN мрежовата архитектура на GNS3**

* 1. **Основна конфигурация на мрежовите устройства**
  2. **Конфигуриране на протокол за динамична маршрутизация IS-IS**
  3. **Конфигуриране на протокол за динамична маршрутизация BGP**
     1. **Конфигуриране на тип iBGP топология Route Reflector**
     2. **Конфигуриране на протокол за мултифункционална динамична маршрутизация MPBGP**
  4. **Конфигуриране на протокол за дистрибуция на етикети LDP**
  5. **Конфигуриране на протокол за комутация на етикети MPLS**
  6. **Конфигуриране на VRF таблица**
  7. **Конфигуриране на политики за качество на услугите QoS**

**Четвърта глава: Тестване на работоспособността на мрежовата архитектура**

* 1. **Тестване на свързаността между отдалечено свързаните компании**
  2. **Тестване на функционалността на протокола BGP**
  3. **Тестване на функционалността на протокола IS-IS**
  4. **Тестване на функционалността на протокола MPLS**
  5. **Тестване на функционалността на протокола LDP**
  6. **Тестване на функционалността на VRF**
  7. **Тестване на функционалността на QoS**

**Заключение**

**Използвана литература**

1. <http://www.steves-internet-guide.com/ipv4-basics/>
2. <https://protechgurus.com/how-does-isis-protocol-work-explained/>
3. <https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_isis/configuration/15-mt/irs-15-mt-book/irs-ovrw-cf.html#GUID-64960B8C-24DB-4929-A7BD-7C308120A650>
4. <https://www.networkworld.com/article/2297171/network-security-mpls-explained.html>
5. <https://www.metaswitch.com/knowledge-center/reference/what-is-label-distribution-protocol-ldp>
6. <https://www.cisco.com/c/en/us/td/docs/ios/mpls/configuration/guide/12_2sr/mp_12_2sr_book/mp_ldp_overview.html#wp1354663>
7. <https://www.imperva.com/blog/bgp-routing-explained/>
8. <https://www.cloudflare.com/learning/security/glossary/what-is-bgp/>
9. <https://www.fir3net.com/Networking/Protocols/what-is-a-bgp-route-reflector.html>
10. <http://www.networkers-online.com/blog/2009/02/bgp-route-reflector-basics/>
11. <https://searchunifiedcommunications.techtarget.com/definition/QoS-Quality-of-Service>
12. <https://www.plixer.com/blog/what-is-vrf-virtual-routing-and-forwarding/>
13. <https://tools.ietf.org/html/rfc4271>
14. <https://tools.ietf.org/html/rfc3031>
15. <https://tools.ietf.org/html/rfc3813>
16. <https://tools.ietf.org/html/rfc5036>
17. <https://tools.ietf.org/html/rfc2547>
18. <https://tools.ietf.org/html/rfc5777>
19. <https://www.certkiller.com/guide-explain-bgp-attributes-and-best-path-selection.htm>
20. <http://www.networkers-online.com/blog/2009/02/bgp-route-reflector-basics/>
21. <https://networklessons.com/bgp/bgp-route-reflector>
22. <http://www.iana.org/assignments/address-family-numbers/address-family-numbers.xhtml>
23. <https://networklessons.com/tag/mp-bgp/multiprotocol-bgp-mp-bgp-configuration>
24. <https://www.iana.org/assignments/safi-namespace/safi-namespace.xhtml>
25. <https://flylib.com/books/en/4.280.1.44/1/>

**Съдържание**