**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*

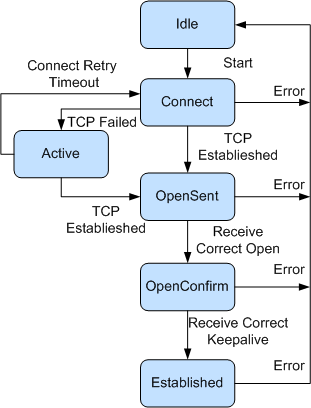
BGP използва четири типа съобщения, в зависимост от състоянието на мрежата:

* **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.

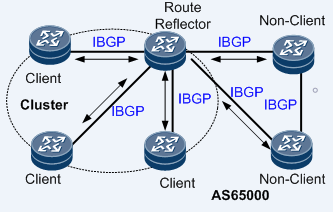
Маршрутизаторите, използващи BGP, взимат решението за това по какъв начин определена информация да достигне дестинацията си. Решението се взима на базата на параметрите от четирите категории. Даден маршрут се избра по определен ред на атрибутите на пътя. Съществуват десет стандартни атрибута, но се практикуват 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.

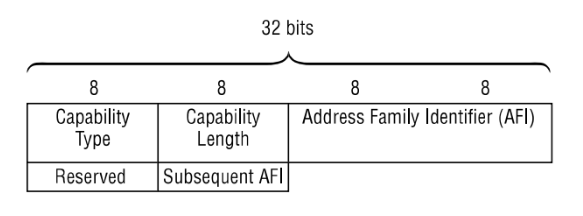
Route Reflector клиент се нарича iBGP маршрутизатор, на който Route Reflector маршрутизаторът отразява пътища. Non-client се нарича обикновен iBGP маршрутизатор, който не участва в Route Reflection процеса и не биват отразявани пътища към него, но поддържа iBGP сесия с Route Reflector маршрутизатора.

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.

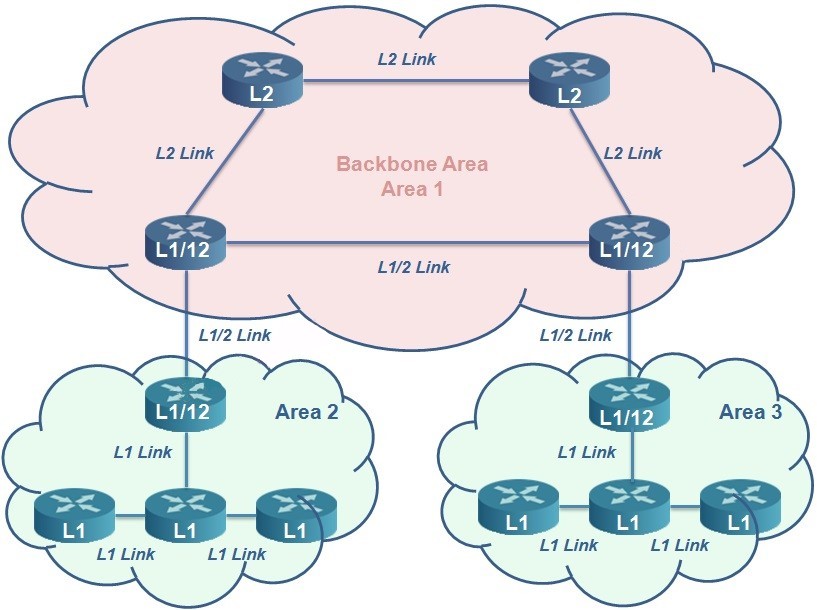
Основният недостатък на IS-IS е, че като всеки Link-State протокол, няма добро мащабиране, като повече маршрутизатори се добавят към маршрутизиращия домейн. С увеличаването на броя на маршрутизаторите се увеличава и големината и честотата на топологичните актуализиращи съобщения, както и времето за което 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.

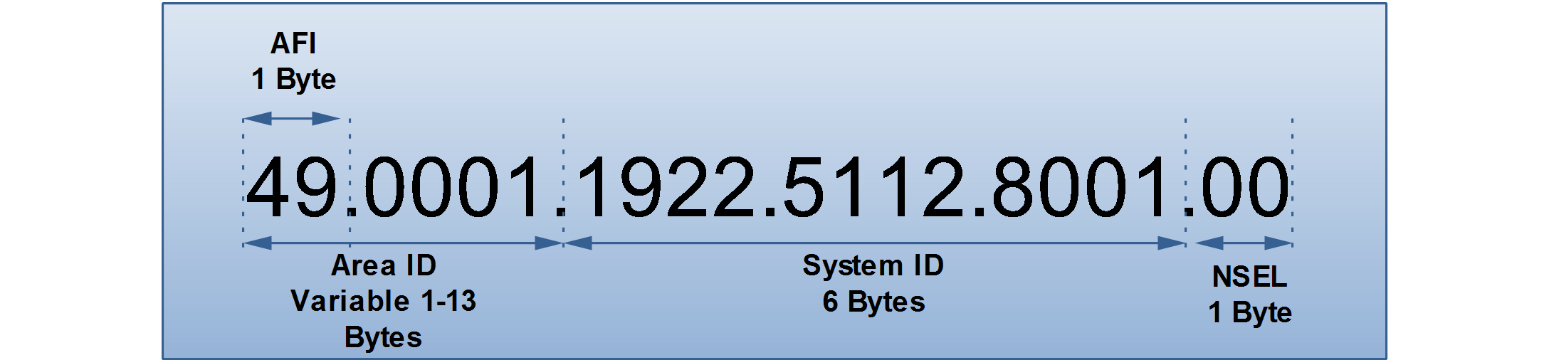
Всеки IS-IS маршрутизатор разпространява информация за своят статус (състояние на интерфейси, достижими съседи, цени на всички интерфейси) към всички други маршрутизатори използващи Link-State PDU (LSP) съобщения. Всеки маршрутизатор използва получените съобщения да изгради идентична база от данни, която 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*

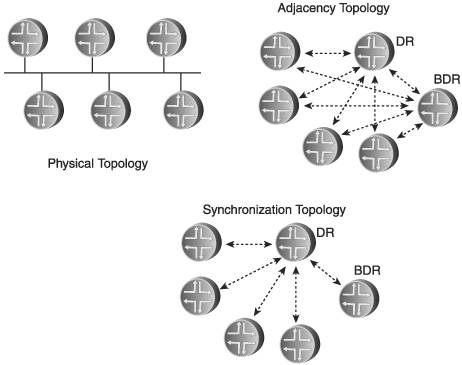
* **Area address (Area ID) –** това поле може да 1 до 13 октета дълго и бървият байт е резервиран за AFI.
* **System ID –** това поле е 6 октета дълго и служи за идентификация на един маршрутизатор. Всеки маршрутизатор трябва да има уникално System ID за своята зона и ниво на което функционира.
* **NSEL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

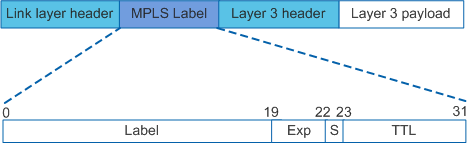


*Фиг.1.12. Видове топологии, които изискват назначаването на DIS*

**1.2.7. 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.

**1.2.7.1. 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.

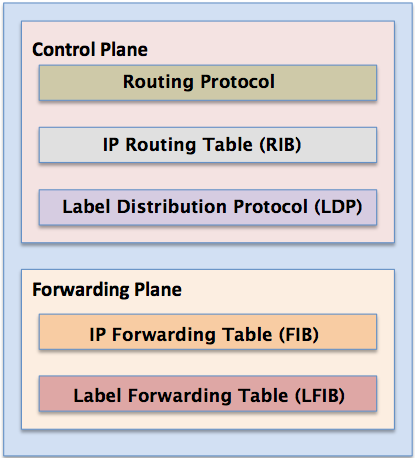
*Фиг.1.13. Формат на MPLS етикет*

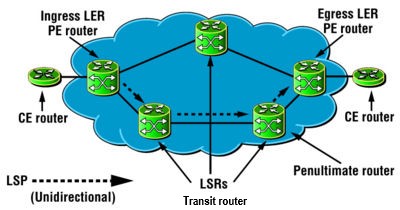
* **Label** – в това поле се записва стойността на етикета с дължина 20 бита
* **Exp** – това поле е с дължина 3 бита и се ползва за Class of Service (CoS)
* **S** – това поле указва, че етикетът е последният, използван върху пакета, с дължина 1 бит.
* **TTL** – Time to Live – полето указва времето на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

**1.2.7.2. 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. **IS-IS**
    2. **BGP**
       1. **Route Reflection**
       2. **MPBGP**

1. **MPLS**
2. **LDP**
3. **VRF**
4. **QoS**

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

* 1. **Основни изисквания към мрежовата топология**
     1. **Изграждане на MPLSVPN архитектура за две различни компании**
     2. **Използване на протокол LDP за дистрибуция на лейбъли**
     3. **Употреба на IS-IS протокол в MPLS мрежата**
     4. **Използване на BGP и MPBGP за реализация на динамична маршрутизация**
     5. **Използване на Route Reflector в MPLS мрежата**
     6. **Осигуряване на резервираност на всеки от клиентите на ниво доставчик на услугата**
     7. **Внедряване на политика за качество на услугата - Quality of Service**
     8. **Симулация на мрежовото решение на GNS3**
  2. **Описание на мрежовата топология**
  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>

**Съдържание**