Рачунарске мреже 2

Павле Вулетић

Београд, 2018.

Павле Вулетић

Рачунарске мреже 2

електронски уџбеник

Рецензенти

др Зоран Јовановић, редовни професор

др Славко Гајин, доцент

Наставно научно већа Електротехничког факултета одобрило је објављивање овог електронског уџбеника одлуком број 356/3 од 21.3.2018.

Издаје и штампа

Електротехнички факултет

Универзитет у Београду

тираж: 50 примерака

Година издања: 2018.

ISBN: 978-86-7225-067-1

Садржај

1. Предговор

Намена овог уџбеника је да пре свега послужи студентима треће године студијског програма Софтверско инжењерство и четврте године модула Рачунарска техника и информатика као основна литература за изучавање напредних технологија рачунарских мрежа које се обрађују у оквиру предмета Рачунарске мреже 2. Уџбеник није основна литература из области рачунарских мрежа, већ се подразумева да је читалац упознат са основним концептима рада рачунарских мрежа, а посебно рада локалних рачунарских мрежа и интерних протокола рутирања, што се на поменутом студијском програму и модулу изучава у оквиру предмета Рачунарске мреже 1. Уџбеником су обухваћени како теоријски концепти рада одабраних, данас актуелних механизама у рачунарским мрежама, тако и детаљан опис практичних вежби и примене ових механизама у симулационом окружењу које је у смислу начина рада потпуно једнако раду у реалним условима, што омогућава како квалитетније разумевање описаних механизама, тако и бољу припрему за рад и примену стеченог знања. На крају, израда овог уџбеника представља и покушај да се неке теме из ове широке области по први пут представе на нашем језику уз преглед најактуелнијих технолошких новости и праваца истраживања и развоја, те се аутор нада да би овај уџбеник могао да нађе читаоце и у широј стручној заједници.

Београд, јануар 2018 Аутор

2. Архитектура интернета и BGP протокол

Данас више није потребно много простора посвећивати томе који је значај који је интернет добио у последњих двадесетак година и на који начин је променио многе аспекте свакодневног живота и пословања. Приступ интернету и коришћење услуга преко њега је једноставно нешто што се подразумева. За стручњаке информационо-комуникационих технологија од кључног значаја је, без обзира на стручно усмерење, разумевање тога како интернет функционише и шта од такве мреже могу да очекују. Софтверски инжењери ће пројектовати и развијати различите апликације које ће бити доступне преко рачунарских мрежа у неком затвореном компанијском окружењу или потпуно доступне преко интернета, пружање услуга у „облаку“ (енг. *cloud*) је постало уобичајено, а развијају се нове технологије сензора и уређаја доступних преко интернета (интернет ствари – енг. *Internet of Things*). Све ово захтева разумевање основних механизама функционисања интернета како би се искористиле његове предности, али и како би се нови системи заштитили од могућих опасности и проблема. Стога, главна тема овог поглавља ће бити објашњавање неких фундаменталних механизама и протокола на којима се заснива рад интернета, а пре свега BGP протокола као основног протокола који повезује различите мреже у данашњи интернет.

Изучавање организације интернета је интересантно и због тога што је интернет највећи потпуно дистрибуирани рачунарско-комуникациони систем који и данас убрзано расте. Изузев неколико функција које морају да буду централизоване да не би дошло до конфликтних ситуација (нпр. додела IP адреса, бројева портова, аутономних система, *top-level* домена и слично), остале функције, а пре свега начин на који су организовани рутирање између и повезивање ентитета (тзв. аутономних система) представљају један потпуно дистрибуиран систем у којем су сви ентитети у смислу повезивања и имплементације права да регулишу политику рутирања своје мреже потпуно једнаки. Разумевање начина организације и рада интернета може у многоме да помогне у разумевању проблема који се јављају у развоју оваквих не-хијерархијских, потпуно дистрибуираних система у којима у

значајном делу не постоје јединствене тачке у случају чијег отказа долази до отказа целог система (енг. *single point of failure*).

Још један битан аспект изучавања начина рада интернета је тај што је интернет систем у којем су сви аутономни системи који га чине потпуно независни у смислу права да дефинишу начин на који ће бити повезани у ову мрежу и креирају политику повезивања (нпр. од кога желе, а од кога не желе да приме неки садржај, где да се пошаљу одређени пакети итд.). Ово чини интернет уједно и највећим системом који поседује аутоматизован начин на који ентитети могу да остваре жељену политику повезивања и рутирања и тиме заштиту сопствених интереса. Аутоматизација овог процеса је значајна јер омогућава брзу реакцију у случајевима отказа уз поштовање интереса аутономних система, али без утицаја оператера – људи.

Експлозиван развој интернета је учинио да је у тренутку након што је достигнута критична маса повезаних мрежа постало јако тешко мењати основне механизме на којима се заснива. Због тога су поједини механизми који се и данас користе настали још у време пре настанка интернета, у време када вероватно нико од пројектаната тих механизама није могао да претпостави колико ће сложена и велика та мрежа постати. Због овога је значајно је разумети и кораке у развоју интернета како би се разумели разлози за поједине техничке одлуке које су донесене, али и последице које ово носи на каснији развој технологија рачунарских мрежа.

2.1. Преглед развоја интернета

Од настанка дигиталног начина преноса информација развој рачунарских комуникација је непрекидан. Као што ће бити показано, упркос томе што су кључни протоколи на којима су засноване рачунарске комуникације стари између 25 и 35 година, значајне промене у начину на који се користе рачунарске мреже се стално дешавају. Неколико кључних етапа у развоју рачунарских мрежа могу да се јасно уоче и оне ће бити приказане у наредним поглављима као и у остатку ове књиге.

2.1.1. Време до дефинисања TCP и IP протокола

Шездесетих година 20. века, убрзо након настанка дигиталног начина преноса информација и првих радова о пакетском начину комуникација, појавиле су се идеје да се тај начин преноса искористи за повезивање рачунара, остваривање њиховог заједничког рада и дељења информација. Први експерименти повезивања рачунара везама остваренim путем телефонских линија су били успешни, али су и показали ограничења дотаташњег начина комуницирања стварањем фиксних веза између две тачке. Ово је довело до реализације првог великог пројекта повезивања повезивања универзитетских рачунарских центара – Арпанет. Арпанет пројекат је био финансиран од стране америчке агенције за напредне

пројекте под Министарством одбране - АРПА[1] (*Advanced Research Projects Agency*) од 1969. до 1989. године и њиме су повезивани универзитети, истраживачки институти и владине институције у САД, на некомерцијалној основи. Прве године пројекта су биле време када је започео рад на архитектури и кључним протоколима потребним за реализацију мреже. Уређаји који су служили за прослеђивање пакета (данас рутери и свичеви) су се звали *Interface Message Processor* (IMP), а први мрежни протокол је био NCP – *Network Control Protocol* (1970)[2.1]. Формирана је радна група (*Network Working Group*) која је доносила спецификације протокола и механизама који су названи *Request or Comments* (RFC). У прво време RFC документи нису имали формалну тежину стандарда, већ су служили између осталог и за дискусију учесника на дизајну и имплементацији мреже (одатле и назив докумената). Временом ови документи су постали еквивалент стандардима у другим стандардизационим телима, а и данас користе и свакодневно доносе за ту сврху, само у оквиру *Internet Engineering Task Force*-а (IETF) уз процедуру доношења која је много формалнија него у доба Арпанета.

У првим годинама Арпанета је настала и прва кључна апликација која је допринела популарности концепта рачунарских комуникација – електронска пошта – мејл (1971) [2.1]. Могућност брзе комуникације текстуалним порукама, далеко брже од класичне поште учинила је да овакав начин комуницирања постане интересантан и корисницима у великим корпоративним окружењима, па је концепт рачунарских комуникација почео да се шири ван универзитетско-истраживачког окружења – стварале су се рачунарске мреже у различитим организацијама, али ове мреже по правилу нису биле међусобно повезане. Ипак кључно место развоја техничких концепата рада мреже била је Арпанет мрежа. Арпанет мрежа је брзо расла – од првих 9 IMP уређаја 1969. године, преко 40 1973. до преко 200 1981. године. Већ тада мрежа је обухватала и локације ван САД: везу према Лондону или према Норвешкој инсталацији за детекцију земљотреса – NORSAR, али ове везе не могу да се сматрају за претече данашњег интернета јер су припадале једној истој мрежи - Арпанет. Арпанет мрежа је била прављена на принципима отворене архитектуре, која би омогућила и другим мрежама да се повежу на њу, без обзира на технологију и начин на који раде мреже.

Током седамдесетих година нису постојали многи механизми који су данас уобичајени у рачунарским мрежама. Тако је на пример DNS (*Domain Name System*) систем чинио један фајл (hosts.txt) који се налазио на SRI - *Stanford Research Institute* у којем су се налазиле адресе свих уређаја на Арпанет мрежи. Рачунари су преузимали овај фајл како би могли да добију адресу жељене дестинације и са растом мреже овакав начин рада је постао нескалабилан. NCP протокол није имао многе функционалности које су данас уобичајене: адресирање рачунара, провера да ли је дошло до грешака у преносу итд. Све ово је довело до потребе да се развију нови протоколи.

Један од кључних дана у историји рачунарских комуникација је 1.1.1983. када је Арпанет мрежа прешла на коришћење TCP и IP протокола. Те исте године су донесени RFC документи којима је дефинисан DNS систем, а прва имплементација DNS софтвера је дошла

1 Агенција је неколико пута мењала име из АРПА у ДАРПА и обрнуто. Слово Д означава Defense.

наредне године. То је време када Арпанет почиње да добија обрисе данашњих рачунарских мрежа. Услед појаве персоналних рачунара, убрзаног пораста броја рачунара и развоја других рачунарских мрежа независних од Арпанета и потребе да се различите мреже међусобно повежу, 1982. године је донесена прва верзија екстерног протокола рутирања – *Exterior Gateway Protocol* (EGP). Ово су били кључни догађаји за настанак интернета.

2.1.2. Настанак глобалног интернета

Успех и раст Арпанет мреже утицао је на настанак сличних мрежа како у САД, тако и у другим земљама (највеће мреже сличног типа су биле Cyclades у Француској, SERCnet мрежа у Великој Британији итд.). Комерцијалне компаније, као и владине организације у Америци су имале своје мреже. Почетком осамдесетих се из Арпанет мреже издвојила војна Милнет мрежа, а финансирање мреже универзитета и института је прешло у руке фондације за науку (NSF – *National Science Foundation*) па је формиран нови пројекат NSFnet који је био финансиран од 1985 до 1995. Ова појава великог броја различитих и одвојених мрежа изазвала је потребу да се ова раздвојена комуникациона острва споје, како би корисници могли да међусобно комуницирају, али и како би на ефикаснији начин могли да се користе рачунарски ресурси у различитим мрежама. У то исто време мреже академских институција су биле уобичејене широм света.

Формиране су прве тачке за размену саобраћаја (*Internet Exchange Points*) у којима је више мрежа било спојено и било је омогућено да уређаји из њих могу да међусобно комуницирају. Владине организације су имале две тачке за размену саобраћаја, по једну на источној и западној обали Америке (FIX-E и FIX-W) од 1989. године. Комерцијалне компаније су прву тачку за размену саобраћаја направила 1991. - CIX. NSFnet је доделила посао телекомуникационој компанији Sprint да је повеже са сличним – академским и истраживачким мрежама у Европи и Азији [2.2]. У ово време касних осамдесетих и почетком деведесетих година се искристалисала архитектура протокола рутирања у рачунарским мрежама – донети су RIP (1988), OSPF (1989) и ISIS (1991) као кључни интерни протоколи рутирања, а у завршној фази је било доношење протокола BGP-v4 који је настао из претходно споменутог EGP протокола и данас једини екстерни протокол за размену рута између мрежа различитих институција. То је и време настанка веба који је утицао на експлозиван раст популарности интернет услуга и појаве великог броја нових апликација рачунарских мрежа. Све ово је утицало на то да се значајно развије посао фирми које су пружале услугу повезивања рачунарских мрежа – интернет провајдера (ISP). Са гашењем пројекта NSFnet практично је престала фаза у којој је била кључна улога академских и истраживачких мрежа у развоју интернета, већ је он пресељен ка интернет провајдерима и произвођачима мрежне опреме.

Ни у време рада Арпанет и NSFnet мрежа, а ни касније када су главну улогу у развоју интернета преузели провајдери, архитектура интернета није подразумевала постојање неког посебног дела интернета који има већи значај од других или кроз који би саобраћај морао да

прође да би се дошло до других мрежа (не постоји мрежа која би могла да се третира као кичма интернета). Интернет је био и остао скуп аутономних система[2] који су међусобно повезани и који су у смислу начина рада (протокола које користе) по свему једнаки: архитектура је потпуно дистрибуирана. Аутономни системи са истим правима су једнако како мреже највећих светских провајдера тако и мреже малих локалних кабловских или АДСЛ провајдера који пружају услуге повезивања на интернет. Разлика је у величини мреже и у томе са колико других аутономних система се повезују (неки највећи аутономни системи повезују и више хиљада других аутономних система) [2.3]. Разлика у броју аутономних система са којима је повезан неки аутономни систем, географском подручју који покрива, као и капацитет саме мреже учинили су да у економском смислу нису сви аутономни системи једнаки што је објашњено у наредном поглављу.

2.1.2.1. Врсте интернет провајдера

Као што је већ наглашено, архитектура интернета је потпуно дистрибуирана и у техничком смислу (у смислу протокола које користе и могућности које имају) сви аутономни системи су једнаки. Међутим, постоји једна неформална подела провајдера по слојевима која је заснована на економском утицају који ови провајдери имају:

- Провајдери првог слоја су провајдери који до свих тачака на интернету могу да дођу а да не купе приступ интернету од других аутономних система. Ови провајдери приступ интернету обезбеђују тако што бесплатно размењују саобраћај са другим провајдерима првог слоја или продају приступ осталим провајдерима.
- Провајдери другог слоја су провајдери који са појединим провајдерима размењују саобраћај, али морају да од неког провајдера првог слоја да купе транзит како би имали доступност до свих тачака на интернету.
- Провајдери трећег слоја који искључиво купују транзит до свих тачака на интернету.

Очигледно је да је позиција провајдера првог слоја таква да постоји могућност да се одлучује о ценама приступа интернету. Због тога постоје посебни услови за учешће у овој затвореној групи провајдера који су врло технички захтевни (мрежа провајдера првог слоја мора да буде присутна на свим континентима, да буде великог капацитета, да има велики број суседних аутономних система са којима размењује саобраћај[3]), али постоје и економска ограничења – ни један провајдер не сме да одреди цену интернета која је нижа од договорене у овој групи.

Група провајдера првог слоја се мењала током развоја интернета. На то је утицало како то што су неки провајдери мењали власника, то да је долазило до спајања и укрупњавања провајдера, али и промене у развоју кључних интернет услуга и њихов утицај на начин комуницирања што је објашњено у наредном поглављу: позиција провајдера првог слоја у

2 Аутономни систем је скуп уређаја и веза (мрежа) под јединственом административном контролом – власништво једне фирме. Више о аутономним системима ће бити речи у поглављу 2.2.1.

3 Нпр. провајдер првог слоја Level3 за националну размену саобраћаја захтева да мрежа са којом ради размену без надокнаде има најмање 500 суседних аутономних система [2.4]

време писања ове књиге је много мање привилегована и значајна него пре 10 или 15 година. Док је 2006. године ова група имала 9 чланова (7 из САД, 1 из Холандије и 1 из Јапана) [2.5], данас их има 16 са већим уделом компанија ван САД (укључени су и немачки, италијански, шведски, шпански, индијски провајдери) и са све већом тендецијом да се интернет саобраћај не рутира кроз америчке провајдере [2.6][4].

2.1.3. Време доминације пружалаца садржаја

У периоду од настанка интернета до средине прве деценије 2000-их кључна зарада од интернета је било пружање услуге повезивања на интернет којим је бројним корисницима омогућено да: приступе жељеном садржају, да обаве куповину, провере рачун у банци и слично. Са друге стране компаније које пружају услуге преко интернета (продаја, банкарске услуге, услуге интерактивних комуникација и слично) су имале интерес да буду на квалитетан начин повезане на интернет и сви они су за приступ интернету плаћали провајдерима.

Од средине 2000-их структура кључних економских интереса на интернету се потпуно променила. Почеле су да доминирају компаније које пружају садржај различите врсте и то пре свих:

- Google са читавим низом различитих услуга попут мејла, видео садржаја на Youtube-у, претрага, простора на диску, пословних канцеларијских апликација и других које се корисницима обично дају бесплатно, да би се кроз приступ садржају компаније пласирале рекламе од којих Google остварује највећу зараду,
- Facebook, који се кроз бесплатне услуге друштвене мреже такође бави оглашавањем,
- Microsoft који је почео да пласира свој софтвер у облику тзв. услуга у облаку које подразумевају рад преко интернета.

Интерес ових компанија је да садржај који дају корисницима буде што ефикасније доступан широм света како би могли да остваре зараду од својих кључних производа. Уместо да приступ интернету купују од великих провајдера, ови пружаоци садржаја (енг. *content providers*) су почели да граде своје приватне интерконтиненталне мреже које повезују њихове велике дата центре широм света и које су по топологији и распрострањености почеле да изгледају исто као мреже провајдера првог слоја [2.7][2.8]. Мреже пружалаца садржаја су почеле да буду присутне у све већем броју тачака за размену саобраћаја широм света, уз политику компанија да раде бесплатну размену саобраћаја са свим аутономним системима који су у тим тачкама за размену саобраћаја и који то желе. Оваква политика пружалаца садржаја је довела до тога да је количина саобраћаја према провајдерима почела да опада, а

4 Тачан списак провајдера првог слоја не може да се зна јер су међусобни уговори између комерцијалних компанија пословна тајна, па не могу да се знају одредбе под којима су неке мреже повезане. Такође, на конфузију утиче и намера неких провајдера да се у маркетиншке сврхе прогласе провајдерима првог слоја или регионалним провајдерима првог слоја. Међутим на основу утицаја неких аутономних система, величина њихових мрежа и броја повезаних аутономних система и утицаја који имају на развој интернета, постоји процена које мреже су у неком тренутку провајдери првог слоја.

тиме и њихови приходи и значај, док постоји јасан континуирани тренд пораста значаја пружалаца садржаја [2.9]. Последњих година кључни финансијери великих инфраструктурних комуникационих пројеката су управо ове фирме које су изградиле највеће интерконтиненталне везе [2.10][2.11], а такође су интерес исказале у правцу реструктурирања начина на који раде рачунарске мреже и производње мрежних уређаја [2.12].

Други ефекат који је утицао на промену начина коришћења интернета и економске односе у вези са њим је повећан значај дистрибуције видео садржаја. Данас видео садржај (дистрибуција снимљеног садржаја са Youtube-a, Netflix и сличне компаније) чини више од 70% укупног саобраћаја на интернету са тенденцијом даљег пораста [2.13]. Како све већи број корисника широм света овај садржај гледа преко својих мобилних уређаја и телевизора, појавила се потреба да се он ефикасно дистрибуира. За ту сврху се користе тзв. мреже за дистрибуцију садржаја (*Content Delivery Networks* – CDN)[5] које садржај доносе много ближе кориснику тако што га реплицирају на сервере који су на многим тачкама на интернету. Ова локализација садржаја је довела до додатног опадања саобраћаја ког класичних провајдера првог слоја тако да је већ неколико година већи удео интернет садржаја у мрежама пружалаца садржаја него у мрежама класичних провајдера са даљом тенденцијом у корист пружалаца садржаја [2.13].

Све ове је довело до тога да је начин на који се користи интернет значајно промењен у последњих десетак година:

- Комуникационе сесије су много више локалне него раније јер се много чешће комуницира са серверима пружалаца садржаја или мрежа за дистрибуцију садржаја који су тополошки близу корисника него са серверима компанија које су власници садржаја.
- Саобраћај брзо одлази до аутономних система пружалаца садржаја који данас недвосмислено имају кључни утицај на развој интернет технологија и услуга и заобилази велике провајдере првог слоја чији је значај опао.

Ипак, кључни протоколи на којима се заснива интернет су данас готово исти као у време настанка интернета.

2.1.4. Управљање интернетом – организације

Технички и технолошки развој мрежних технологија је у првој половини Арпанет пројекта био у оквиру Network Working Group неформалне групе која је доносила RFC документе са спецификацијама протокола и механизама који се користе у рачунарским мрежама. Из ове групе је 1986. године створен IETF (*Internet Engineering Task Force*), данас вероватно најпознатије интернет тело у којем се дефинишу нови механизми који ће се користити на интернету и рачунарским системима и које и данас доноси RFC документе. IETF је

5 Мреже за дистрибуцију садржаја су детаљније описане у поглављу 5.1.

организован као скуп неформалних тематских радних група (*working groups*) људи заинтересованих за унапређење појединих интернет механизама, мада данас најчешће велике компаније стратешки укључују запослене у својим развојним тимовима да доприносе доношењу стандарда који одражавају њихове правце развоја. Није основан као посебно правно лице и нема свој буџет и трошкове. Рад у оквиру IETF је потпуно отворен – свако може да се пријави на мејлинг листу било које радне групе и допринесе њеном раду или само пасивно прати догађања. IETF организује три годишња састанка на којима се састају радне групе и разматрају свој рад. Рад радних група је отворен и сви заинтересовани појединци могу да учествују. У прво време IETF је постојао као самостална група људи, док је касније по оснивању ISOC (*Internet SOCiety*) 1992. године IETF прикључен овој организацији.

Поред IETF техничким развојем се бави и IRTF (*Internet Research Task Force*), основан 1989. у којем се не доносе стандарди већ се бави дугорочнијим истраживањем појединих проблема који по успешном окончању могу да пређу у IETF како би било формално стандардизовани. И IRTF је укључен у ISOC по његовом оснивању. У време када је престало финансирање Арпанет пројекта и када је постало јасно да и NSFnet неће још дуго трајати, основан је ISOC са циљем да се обезбеди даље финансирање развоја интернета. ISOC је основан као приватна непрофитна организација са седиштем близу Вашингтона у САД и неколико регионалних канцеларија широм света. ISOC се финансира из чланарина преко 100.000 индивидуалних и корпоративних чланова.

Данас се у оквиру ISOC поред IETF и IRTF налази и *Internet Architecture Board* (IAB), саветодавно тело које надгледа правац развоја технологија и стандарда које се разматрају у оквиру IETF и IRTF и именује руководиоце радних група у оквиру ових тела. IAB је настао од *Internet Configuration Control Board* – тела у оквиру ДАРПА агенције у време док је водила Арпанет пројекат.

Друга кључна организација која данас управља неким аспектима интернета је ICANN (*Internet Corporation for Assigned Names and Numbers*), основан 1998. такође као приватна непрофитна организација са седиштем у Лос Анђелесу. Ово тело је задужено за доделу кључних идентификатора на интернету (IP адреса, бројева портова, бројева аутономних система, *top level* домена итд.) и одржавање root DNS система. Наведени послови се обављају у оквиру IANA (*Internet Assigned Numbers Authority*) тела која је основано од стране Владе САД 1988., а које је данас једно одељење у оквиру ICANN. IANA је додељивање IP адреса, бројева аутономних система и управљање деловима root DNS система дистрибуирала на тзв. регионалне интернет регистре (*Regional Internet Registry* – RIR) непрофитне организације које су задужене за поједине географске области на свету: RIPE[6] за Европу, ARIN[7] за северну Америку, APNIC[8] за Азију, Аустралију и земље Пацифика, LACNIC[9] за јужну и централну Америку и AFRINIC[10] за Африку. Мреже које желе да добију своје IP

6 https://www.ripe.net/
7 https://www.arin.net/
8 https://www.apnic.net
9 http://www.lacnic.net/
10 https://www.afrinic.net

адресе и бројеве аутономних система и на тај начин буду повезане на интернет се региструју код одговарајућег регионалног интернет регистра, и постају локални интернет регистри (*Local Internet Registry* – LIR). RIR се финансирају од годишње чланарине које плаћају LIR.

Организација и управљање интернетом су значајно другачији од начина регулисања класичних телекомуникација где постоје чврста национална регулатива и регулаторна тела која управљају ресурсима и бројевима (у Србији РАТЕЛ – Регулаторна агенција за електронске комуникације и поштанске услуге), док су међународни аспекти телекомуникација регулисани у оквиру Међународне телекомуникационе уније ITU (*International Telecommunication Union*) – агенције Уједињених нација. Како је интернет преузео водећу улогу у остваривању међународних комуникација и обављања свих врста пословних делатности, а многи аспекти пословања преко интернета нису адекватно дефинисани (нпр. правна заштита у случају напада или крађа када подаци пролазе преко већег броја земаља, аспекти цензуре и заштите осетљивих група на интернету итд.), постоје сталне иницијативе да се развој интернета формализује на сличан начин као класичне телекомуникације. 2006. године је основан форум за управљање интернетом (*Internet Governance Forum*) у оквиру Уједињених нација у оквиру ког се расправља о начинима како би се управљање неким аспектима интернета побољшало. За сада још увек нема адекватних решења за управљање интернетом која би ишла у правцу веће регулације попут оне у оквиру ITU.

2.2. BGP протокол

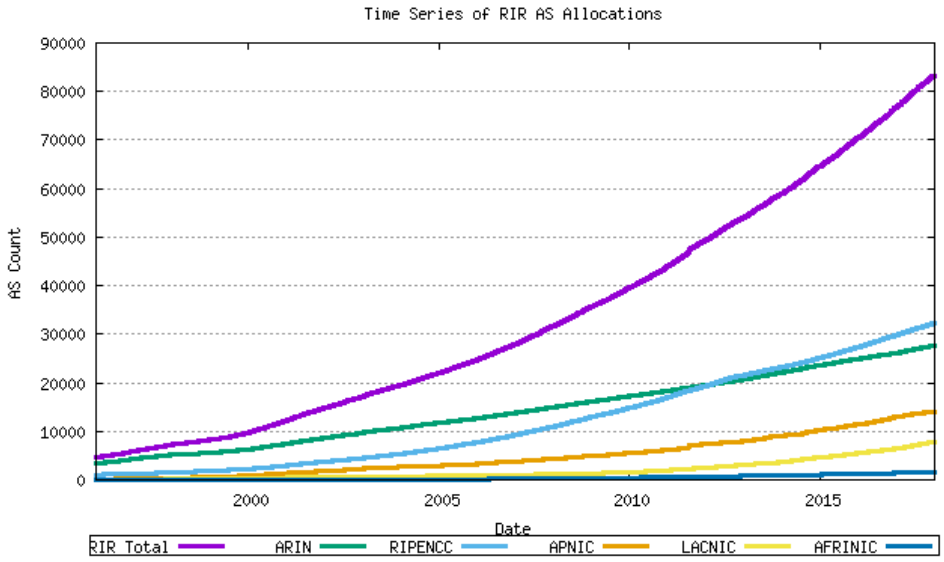
BGP (*Border Gateway Protocol*) је једини екстерни протокол рутирања који се данас користи за размену рута између аутономних система. BGP је уз DNS систем једини механизам који има глобални домет на интернету и омогућава јединствено функционисање ове огромне дистрибуиране инфраструктуре. Сви аутономни системи који су повезани на интернет користе BGP протокол да би размењивали руте тако да се информација о промени неке руте у једном аутономном систему пропагира до свих осталих аутономних система. Верзија BGP протокола која се данас користи је 4 [2.14] и она је усвојена 1995. године када је замењен претходни протокол – EGP који се користио у NSFnet.

Једна од кључних особина протокола рутирања је метрика. Оно што је специфично код BGP протокола је то што за њега није једноставно рећи која је његова метрика. Као шо ће бити показано, BGP користи више различитих критеријума за одређивање најбоље путање које власник аутономног сиситема може по својој жељи да промени, како би могао да одреди политику рутирања која њему највише одговара. Подешавање произвољне политике рутирања је једна од кључних особина која је потребна да би се очувала аутономија аутономних система који на тај начин могу да остваре своје пословне интересе.

2.2.1. Аутономни системи

Аутономни систем је скуп уређаја и веза које су под контролом једне организације и који представља основну јединицу на интернету која са другим аутономним системима размењује руте путем BGP протокола. Аутономни систем није мрежа било које организације, већ само оних које су регистроване као LIR у регионалним интернет регистрима и које су од RIR добиле број аутономног система и поседују своје IP адресе. Услов да би нека мрежа добила број аутономног система је да у тренутку пријављивања за добијање броја још најмање два постојећа аутономна система гарантују да ће по добијању броја аутономног система размењивати саобраћај са новим аутономним системом. Ово значи да мрежа неке институције која је повезана на само једног провајдера не може да постане аутономни систем, већ ће она добити адресе провајдера на ког је повезана и бити део његовог адресног простора.

Број аутономног система је до 2007. године био 16-битна вредност. Бројеви мањи од 64.512 су могли да се додељују аутономним системима који су повезани на интернет, док су бројеви већи или једнаки 64.512 тзв. приватни бројеви аутономних система који могу да се користе у затвореним окружењима, али не и да се појаве у BGP табелама на интернету. Број аутономних система непрекидно расте брже од линеарног раста и данас има преко 80.000 аутономних система који чине интернет, што је показано на слици 2.1 где се види и расподела броја аутономних система по регионалним интернет регистрима. Због исцрпљивања простора бројева аутономних система 2007. године је број аутономног система промењен тако да буде 32-битна вредност, што оставља довољно простора за будући раст.



Слика 2.1 Број аутономних система на интернету. Преузето са http://www.potaroo.net/tools/asn32/ (3.1.2018.)

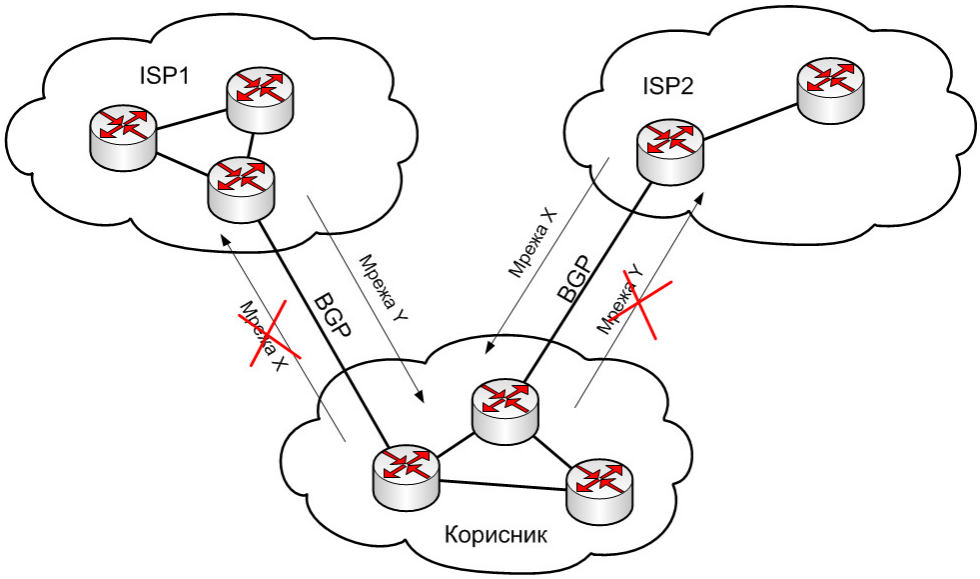
У време настанка интернета аутономни сиситеми су типично биле провајдерске мреже и све оне мреже које су омогућавале већем броју корисника повезивање на интернет. Временом се ово променило и данас све већи број институција које кључне пословне операције обављају

преко интернета (нпр. банке, фирме за е-трговину, компаније које пружају услуге у облаку итд.) имају потребу за редундантним везама ка интернету преко већег броја провајдера па постају аутономни системи. Управо ове организације чине да број аутономних система на интернету и даље убрзано расте.

Постоји неколико врста аутономних система према улози коју имају на интернету:

- Аутономни системи са једним излазом (тзв. *stub AS*) који имају само теоријски значај због претходно изнетог начина на који се додељују бројеви аутономних сиситема
- Аутономни системи са више излаза без транзита саобраћаја (енг. *multihomed nontransit*).
- Аутономни системи са више излаза и транзитом саобраћаја (енг. *multihomed transit*)

Пример за аутономни систем са више излаза и без транзита саобраћаја би била мрежа неке банке (корисник интернет услуга) која је због редундансе и повећања доступности услуга повезана на два провајдера (Слика 2.2).

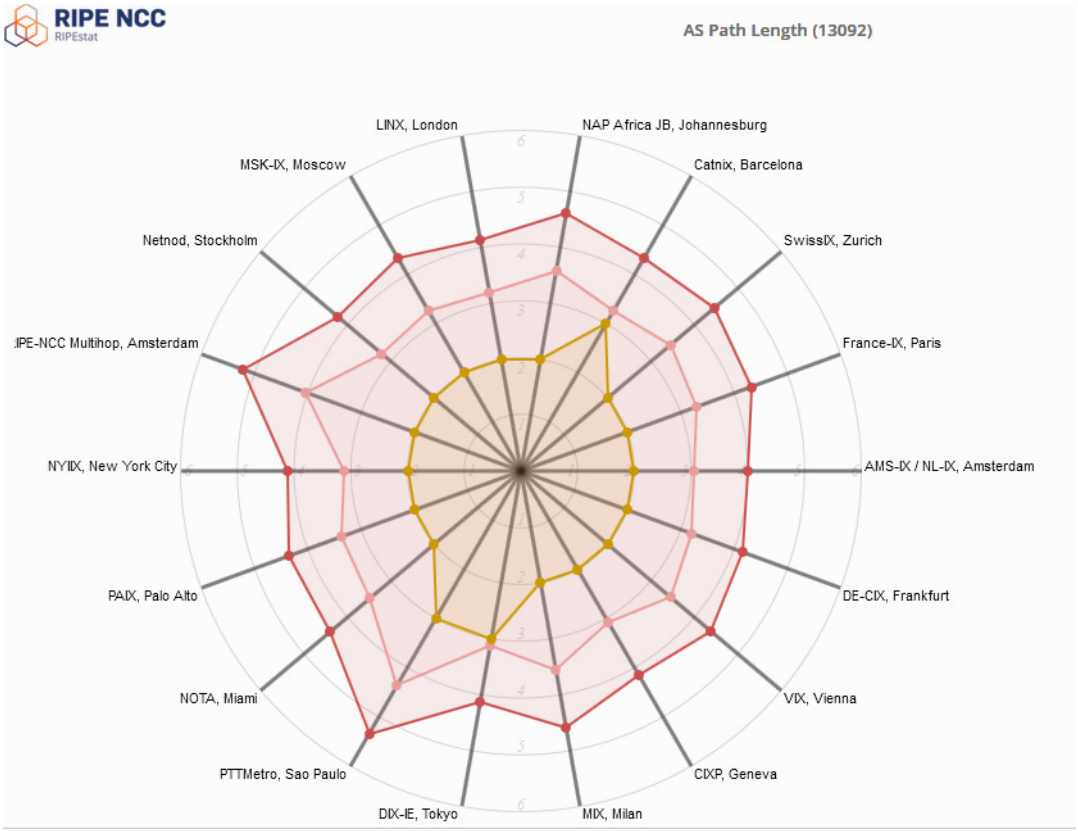


Слика 2.2 Аутономни систем са више излаза и без транзита саобраћаја

На овој слици провајдер 1 поседује скуп адреса Y, а провајдер 2 скуп адреса X. Ако би се од аутономног система корисника и BGP протокола очекивало да се понаша аналогно интерним протоколима рутирања, онда би корисник по добијању руте Y од провајдера 1 ову руту проследио провајдеру 2, и обрнуто руту X од провајдера 2 према провајдеру 1. То значи да би на овај начин било омогућено рутирање саобраћаја између два провајдера (између уређаја на мрежама X и Y) кроз мрежу корисника. Како је корисник купио везе од оба провајдера за потребе свог посла, у његовом интересу није да пропушта туђи саобраћај кроз своје плаћене везе. Због овога ће корисник да филтрира руту ка мрежи Y како се не би огласила према провајдеру 2 и руту ка мрежи X како се не би огласила према провајдеру 1. На тај начин рутирање између мрежа X и Y неће ићи кроз мрежу корисника, већ ће везе према

провајдерима служити искључиво за потребе корисника. Овај пример управо илуструје кључну разлику BGP протокола у односу на све остале протоколе рутирања – једна од кључних особина BGP је могућност да манупилацијом рутама сваки аутономни систем дефинише своју политику рутирања како би остварио свој економски интерес од коришћења интернета.

Пример аутономног система са више излаза и дозвољеним транзитом би била мрежа неког провајдера. У његовом интересу је управо да омогући повезивање више мрежа кроз своју инфраструктуру јер на тај начин остварује зараду – продајом веза према интернету корисницима. И ове мреже имају потребу за регулисањем рутирања, јер им је у интересу на пример да што више саобраћаја добију од оних аутономних система са којима имају бесплатну размену саобраћаја, а мање од оних аутономних система којима по капацитету везе плаћају за приступ неким деловима интернета.



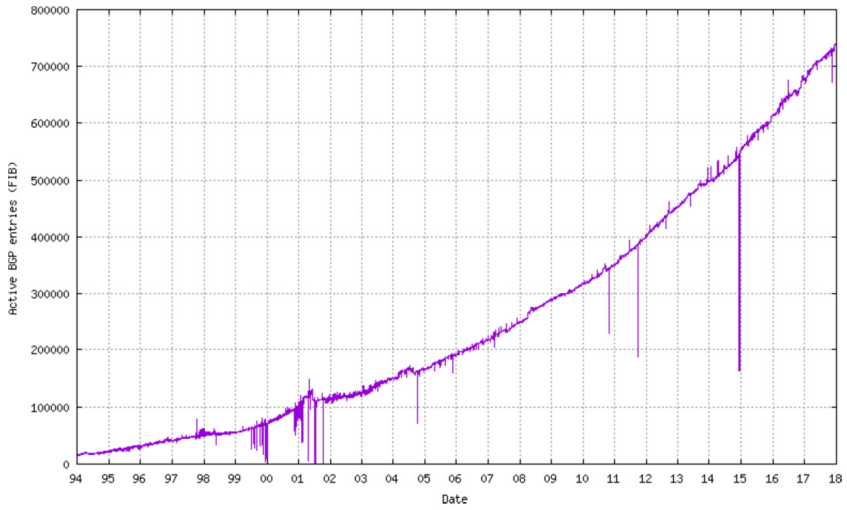
Слика 2.3 Дужина путање у броју аутономних система од AMPEC мреже до тачака за размену саобраћаја. Преузето са https://stat.ripe.net/

Као што је показано, на интернету у време писања ове књиге има преко 80.000 аутономних система. Упркос овако великом броју, аутономни системи су густо повезани са великим бројем међусобних веза тако да путање пакета на интернету ретко прелазе преко више од 5-6 аутономних система. Слика 2.3 показује број аутономних система (тзв. дужину AS путање) од аутономног система Академске мреже Србије (AMPEC – AC 13092) до тачака за размену саобраћаја широм интернета. Као што може да се види, ова дужина варира између 2 и 5 аутономних система у зависности од рутирања између ових тачака и тога да ли се користи

тзв. *AS-path prepending* (спољашња линија на слици - објашњено у поглављу 2.2.4.3) - до 4 аутономна система без *prepending-a* (средња линија на слици). Ове путање су нешто краће за аутономне системе у Америци јер је ту највећи број великих аутономних система првог слоја, а нешто дуже за аутономне системе из региона Аустралије и Пацифика.

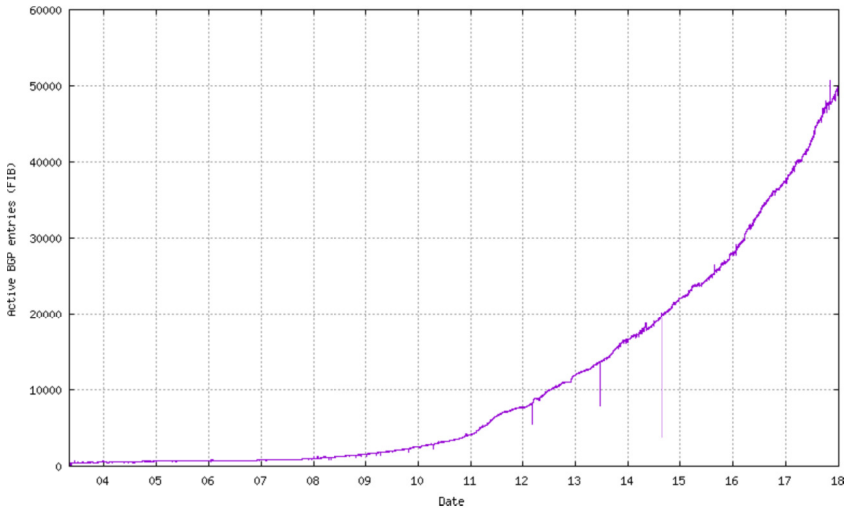
2.2.2. BGP начин рада

BGP протокол успоставља сесије између рутера који се зову BGP суседи (енг. *neighbours*). BGP сесије се успостављају преко TCP протокола, порт 179. За разлику од интерних протокола рутирања, BGP протокол се ослања на TCP зато што су интернет табеле рутирања које се њиме преносе далеко веће од количине информација које размењују интерни протоколи рутирања и размена се врши преко већих растојања, те је поузданост преноса једна од битних особина. Такође, за разлику од интерних протокола рутирања, BGP суседи не морају да буду директно повезани, што повећава вероватноћу да на некој од веза између BGP суседа дође до губитка пакета. Поруке послате BGP протоколом могу да буду заштићене од промена коришћењем MD5 алгоритма.



Слика 2.4 Број IPv4 рута у пуној интернет табели рутирања (преузето са http://bgp.potaroo.net/as6447/ 3.1.2018.)

BGP протокол је по начину рада блажи *distance-vector* протоколима рутирања. Њиме се преносе све руте између два рутера, а то може да буде и пуна интернет табела рутирања (енг. *Full Internet Routing Table* – FITR). За разлику од интерних *distance-vector* протокола попут RIP-а, BGP не преноси целу табелу рутирања периодично, већ се комплетна табела шаље иницијално приликом успостављања суседског односа, а касније се шаљу само оне руте за које постоје промене. Слике 2.4 и 2.5 показују величину пуне интернет табеле рутирања за IPv4 и IPv6 руте респективно.



Слика 2.5 Број IPv6 рута у глуној интернет табели рутирања (преузето са http://bgp.potaroo.net/v6/as6447/ 3.1.2018.)

Као што може да се види, број рута и даље расте брже од линеарног, чак и за IPv4 руте које су дошле до готово потпуног исцрпљивања. Очекује се да ће у скорој будућности број ових рута ипак стагнирати, а раст броја IPv6 рута и даље убрзано расти.

```
RS_AS3303>sh bgp summa
BGP router identifier 217.192.89.52, local AS number 65097
BGP table version is 13297967, main routing table version 13297967
665446 network entries using 98486008 bytes of memory
665446 path entries using 42588544 bytes of memory
131614/14 BGP path/bestpath attribute entries using 17899504 bytes of memory
97580 BGP AS-PATH entries using 3697340 bytes of memory
22341 BGP community entries using 2059194 bytes of memory
304 BGP extended community entries using 9166 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 164739756 total bytes of memory
BGP activity 3985490/3278862 prefixes, 5094653/4388039 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
193.247.171.25  4        65000 9565688  101418 13297967    0    0 4w4d       665446
```

Слика 2.6 Сумарни преглед BGP конфигурације сервера рута провајдера Swisscom (AS3303)

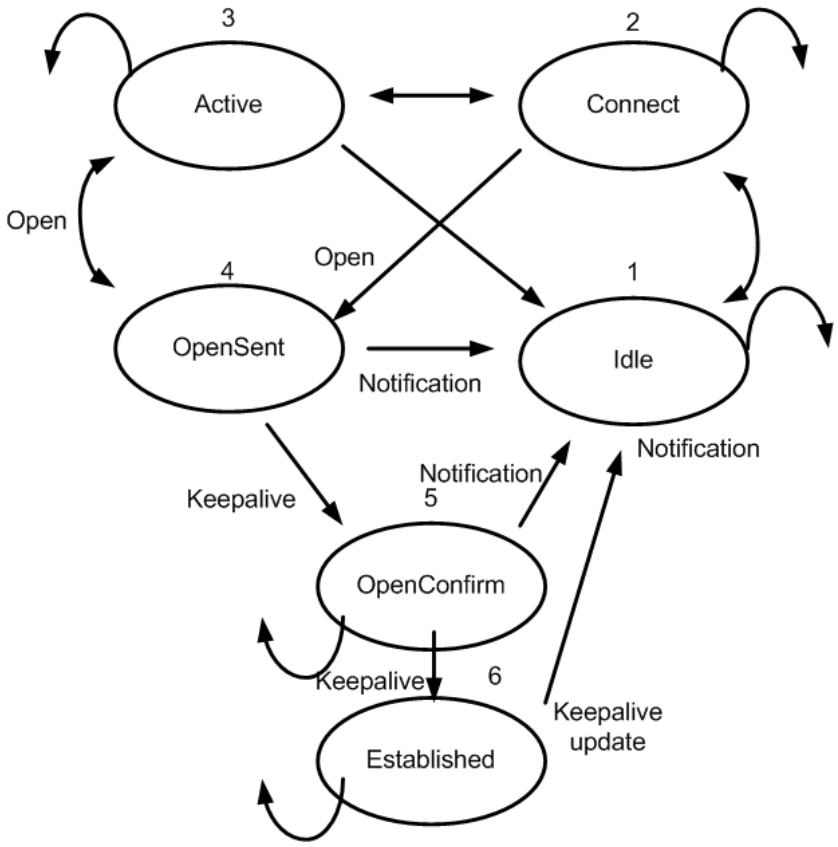
Руте добијене од свих суседа се смештају у BGP табелу која због величине пуне интернет табеле рутирања заузима значајне меморијске ресурсе, а једна најбоља рута се из BGP табеле преbацује у табелу рутирања. Слика 2.6 показује да само једна инстанца пуне интернет табеле рутирања на серверу рута једног провајдера првог слоја која има 665.446 рута заузима готово 100MB меморије. Рутери са великим бројем суседских односа морају да чувају вишеструке копије ових рута и потребни су им значајни меморијски ресурси.

2.2.2.1. BGP поруке и стања сесија

Функционисање BGP протокола је релативно једноставно, што је и разумљиво јер су у сваком тренутку стотине хиљада инстанци овог протокола укључене у свим аутономним

системима на интернету и размењују информације о стотинама хиљада рута. BGP протокол има само 4 врсте порука:

- *Open* порука којом се успоставља BGP сесија. Да би се успоставила сесија, у *Open* порукама које размењују BGP суседи морају да се поклопе бројеви аутономних система и времена слања *Keepalive* порука.
- *Keepalive* порука која представља само BGP заглавље и шаље се периодично како би се проверило да ли су BGP сусед и сесија са њиме и даље активни.
- *Notification* порука којом се суседи обавештавају о евентуалним грешкама у порукама које су добили.
- *Update* порука којом се шаљу информације о рутама. У жаргону BGP протокола који је установљен RFC документима руте се зову NLRI – *Network Layer Reachability Information.*



Слика 2.7 Машина стања BGP протокола

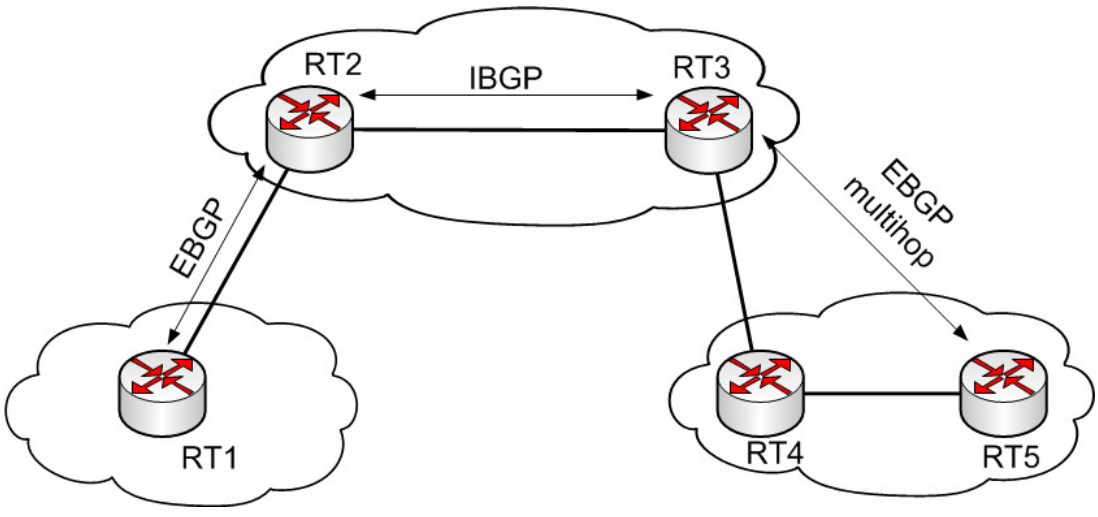
Свака BGP сесија има неколико стања кроз која пролази (Слика 2.7). *Idle* је прво стање BGP сесије непосредно након што је конфигурисана на уређајима. BGP почиње да ослушкује BGP поруке како би се отпочело успостављање сесије ако стигне нека порука од суседа. Када се почне успостављање TCP сесије за дату BGP сесију прелази се у стање *Connect*. Уколико се TCP сесија успешно успоставила шаље се BGP *Open* порука и прелази се у стање *OpenSent*. Ако се TCP сесија није успоставила прелази се у стање *Active*. У *Active* стању се чека

истицање тајмера након кога се поново покушава успостављање TCP сесије за дату BGP сесију. Уколико је сесија са неким суседом дуго у *Active* стању значи да постоји или проблем у повезаности између два рутера или у конфигурацији параметара сесије (IP адреса).

Након што је послата *Open* порука, ако се од суседа добије коректна повратна *Open* порука, шаље се *Keepalive* порука и прелази у стање *OpenConfirm*. Ако је добијена *Open* порука са неком грешком, суседу се одговара *Notification* поруком и прелази се поново у стање *Idle*. Када се у стању *OpenConfirm* добије *Keepalive* порука, прелази се у стање *Established* и почиње размена *Update* порука у којима су информације о рутама које поседује дати сусед. *Notification* порука добијена у било ком од ова два последња стања враћа сесију у стање *Idle.*

2.2.3. Интерни и екстерни BGP

BGP протокол је специфичан по томе што постоје две његове варијанте – једна која се користи за размену рута између аутономних система – екстерни BGP, односно eBGP и једна која се користи за размену рута између рутера унутар једног аутономног система – интерни BGP, односно iBGP. Улога iBGP није да замени интерни протокол рутирања већ да обезбеди континуитет рутирања аутономног система онда када постоји више различитих улаза у аутономни систем (на пример велики аутономни системи провајдера првог слоја који имају улазе на различитим континентима и треба да обезбеде рутирање између мрежа повезаних на њих). Слика 2.8 показује врсте BGP сесија. Као што је већ речено, BGP не захтева да BGP суседи буду директно повезани (важи и за интерне и за екстерне суседе). Ако је то случај, такве сесије се називају *multihop* BGP сесије.

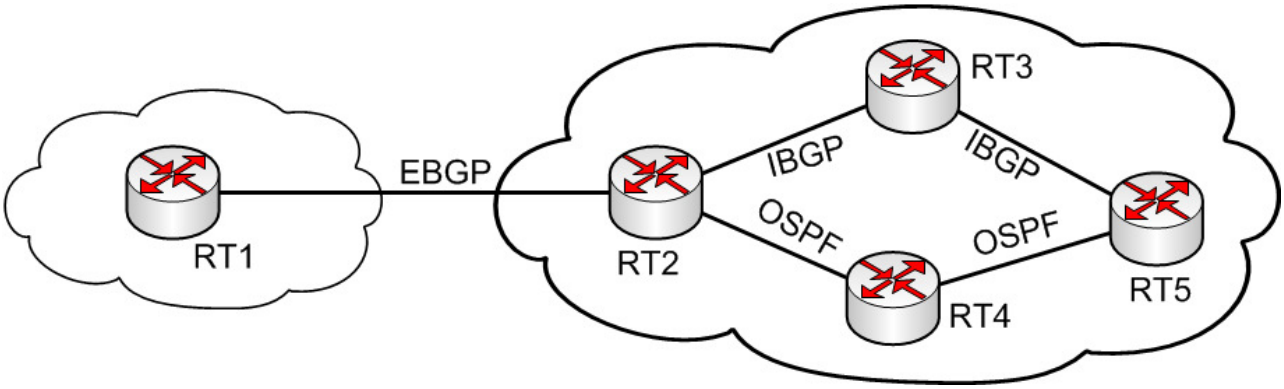


Слика 2.8 Врсте BGP сесија

2.2.3.1. Специфичности iBGP протокола

iBGP има једно специфично ограничење: рута која је добијена од једног iBGP суседа не може да буде прослеђена другим iBGP суседима. Ово значи да ако би се успоставило рутирање као на слици 2.9, руте које би рутер RT3 добио од рутера RT2 не би смеле да се проследе рутеру

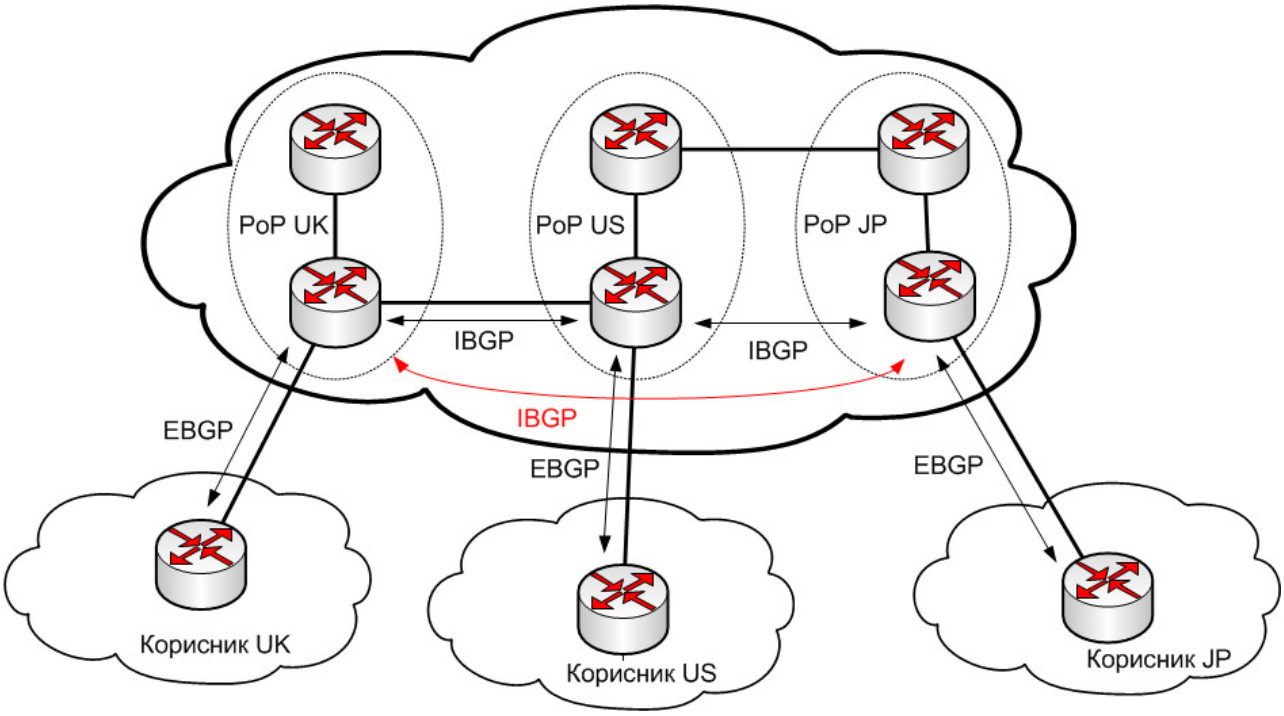
RT4 путем iBGP. То је понашање протокола које је у потпуној супротности са начином на који раде интерни протоколи рутирања чија је основна функција да прослеђују информације о рутама унутар домена у којем су конфигурисани. Међутим, треба имати на уму да намена iBGP протокола није да обезбеди рутирање унутар домена, већ да помогне у транспорту BGP рута кроз аутономни систем. Такође, када iBGP не би имао ово ограничење, постојала би опасност од стварања петљи у рутирању унутар аутономног система. Као што ће бити показано у поглављу 2.2.4.3 механизам који BGP користи за спречавање петљи у рутирању је бележење информација о аутономним системима кроз који је прошла рута. Ово је у случају интерног BGP неизводљиво јер су све размене рута унутар истог аутономног система, те поменуте заштите нема. Због тога је предвиђено да интерни протоколи рутирања постоје у сваком аутономном систему паралелно са iBGP протоколом. Са друге стране не постоје ограничења у прослеђивању рута између iBGP и eBGP суседа.



Слика 2.9 iBGP рутирање - пример

2.2.3.2. Континуитет BGP унутар аутономног система

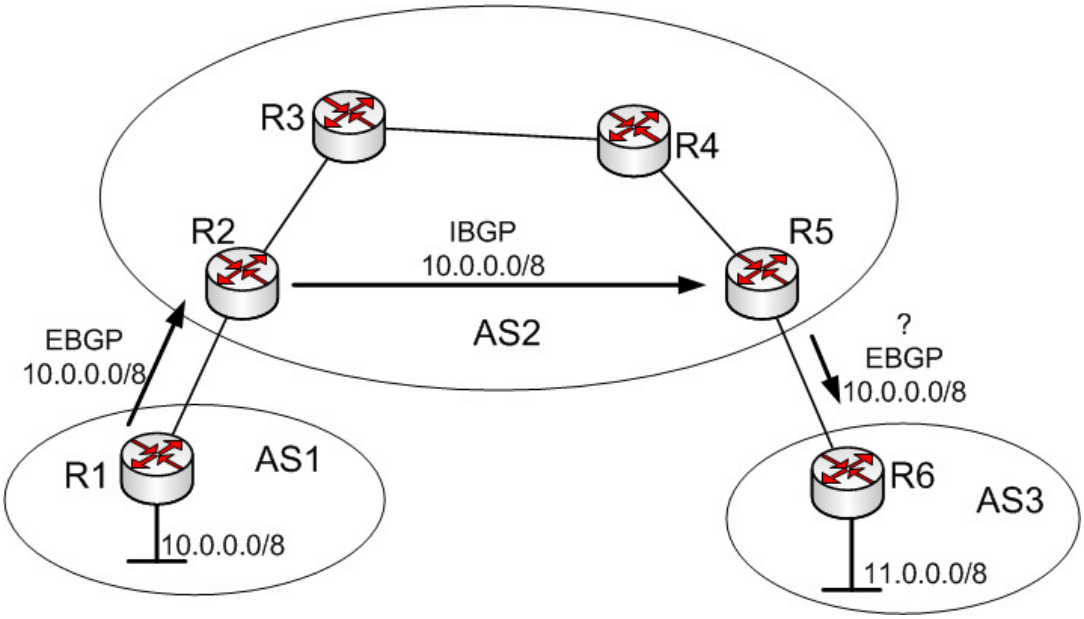
Поменута специфичност iBGP протокола повлачи неке последице у организацији рутирања унутар аутономних система. На слици 2.10 је дат пример једне мреже великог провајдера који повезује три корисника у Великој Британији, САД и Јапану. Уколико би се успоставиле iBGP сесије између рутера провајдера на следећи начин: UK-US и US-JP, корисници из САД би могли да комуницирају са корисницима у Великој Британији и Јапану, али не би могли да међусобно комуницирају корисници из Велике Британије и Јапана зато што руте из њихових мрежа не би биле размењене (рутер у САД не би смео да руте добијене из Велике Британије проследи ка тачки у Јапану и обрнуто). Због овога је потребно да се успостави iBGP сесија између рутера у Великој Британији и Јапану. Односно, да би било коректно рутирање у аутономном систему провајдера потребно је да постоји потпун граф iBGP сесија између свих ивичних рутера аутономног система који размењују руте са другим аутономним системима. Ово није проблем, јер као шо је већ наведено, BGP сесије могу да се успостављају између рутера који нису директно повезани (*multihop*).



Слика 2.10 Пример обезбеђивања континуитета рутирања у аутономном систему

2.2.3.3. Сихронизација BGP и интерног протокола рутирања

Уколико би се применили принципи описани у претходном поглављу, могло би да дође до ситуације описане на слици 2.11. Аутономни систем 1 оглашава руту 10.0.0.0/8 према аутономном систему 2 и рутеру 2 који путем iBGP прослеђује ту руту ка рутеру 5, који даље руту прослеђује према аутономном систему 3. Што се тиче контролне равни, односно нивоа прослеђивања рута све је у реду, јер не постоје ограничења да рута пропагира на овај начин (рута добијена путем iBGP може да буде оглашена eBGP суседима). Ипак, ако би се покушало слање пакета од мреже 11.0.0.0/8 ка мрежи 10.0.0.0/8, пакети не би пролазили, а разлог за то је чињеница да рутери 3 и 4 унутар аутономног система 2 који су на физичкој путањи пакета немају информацију о томе како се пакети прослеђују према мрежама 10.0.0.0/8 и 11.0.0.0/8. Оваква ситуација у којој неки аутономни систем исправно оглашава руте, али није у могућности да проследи пакете ка оглашеним рутама се назива „црна рупа“ у рутирању (енг. *black hole*). Да би се избегле црне рупе у рутирању уведено је правило да BGP унутар неког аутономног система мора да буде синхронизован са интерним протоколом рутирања да би нека рута могла да буде оглашена екстерним BGP суседима. Ово значи да нека рута може да буде оглашена у други аутономни систем само ако је већ постојала у табели рутирања пре него што је добијена путем iBGP. Тиме се обезбеђује да сви рутери на путањи пакета имају информацију о томе где су све потенцијалне дестинације и спречавају се црне рупе у рутирању.



Слика 2.11 Проблем стварања „црних рупа“ у рутирању

Постоји више начина на који може да се оствари синхронизација BGP и интерног протокола рутирања, али нису сви практично могући:

- Редистрибуција свих BGP рута у интерни протокол рутирања: Ако би се на рутеру 2 редистрибуирала рута ка мрежи 10.0.0.0/8 у интерни протокол рутирања који постоји у аутономном систему 2, онда не би било проблема да рута ка овој мрежи дође до рутера 3, 4 и 5. На тај начин не би било проблема да се рута даље огласи ка аутономном систему 3 и да се оствари регуларно рутирање. Међутим овај приступ је могуће реализовати само у затвореним (нпр. лабораторијским) окружењима али не и на интернету. Један разлог је број рута у пуној интернет табели рутирања који би направио велике проблеме интерном протоколу рутирања ако би се све руте редистрибуирале у њега – интерни протоколи рутирања нису пројектовани тако да раде са стотинама хиљада рута. Други разлог ће бити објашњен у поглављу 2.2.8 - на овај начин би се нарушио један од битних принципа да аутономни систем може да оглашава само оне руте које поседује.
- Пошто редистрибуција није практично могућа између осталог због скалабилности, једно потенцијално решење да се смањи број рута би било да се на рутере 3 и 4 поставе *default* руте које указују у правцу аутономног система 1. Међутим, док би овакво решење решило проблем протока пакета у једном смеру, ка аутономном систему 1, повратни пакети не би могли да прођу, јер би за њих постојале петље у рутирању, па ни ово није решење.
- Решење које се стандардно користи у великим аутономним системима је да се успостави потпун граф iBGP сесија између свих рутера у аутономном систему који су на путањи пакета и искључи синхронизација. Тако ће сви рутери у аутономном систему сигурно имати информацију о томе где се налазе све BGP дестинације

добијене од суседних аутономних система, а неће постојати проблем са радом интерног протокола са пуном интернет табелом рутирања. Како је број потенцијалних iBGP сесија које се на овај начин креирају сразмеран квадрату броја рутера у аутономном систему и може да буде јако велики, те како додавање једног новог рутера значи реконфигурацију свих осталих и како такође ово значи да велики број рутера у аутономном систему треба да има хардвер за прослеђивање пакета који може да подржи пуну интернет табелу рутирања, реализовани су механизми описани у поглављу 2.2.7 којима број iBGP сесија може да се смањи.

2.2.4. Атрибути рута

Пример аутономног система без транзита из поглавља 2.2.1 је показао једну кључну особину BGP протокола, а то је могућност власника аутономног система да утиче на то на који начин ће руте које добија или шаље пропагирати кроз његов аутономни систем. Као што ће бити показано, утицај на руте не мора да буде само филтрирање (одбацивање) рута као у примеру из датог поглавља, већ је могуће утицати и на путање којима ће пакети пролазити. Могућност да се врше манипулације рутама тако да се одреде оптимални токови пакета у складу са интересима сваког аутономног система је кључни предуслов за постојање интернета као највеће дистрибуиране инфраструктуре са потпуно аутоматизованим системом размене рута, без икаквог мануелног уплитања приликом одлучивања, у којој сваки ентитет може да заштити своје интересе.

Да би све ово било могуће, свака рута која се шаље BGP протоколом има придружен низ атрибута, а сам процес обраде рута је знатно сложенији него код интерних протокола рутирања.

Атрибути рута могу да се поделе у 4 основне групе [2.14]:

- Добро познати обавезни атрибути (енг. *Well Known Mandatory*). Ово су атрибути који су описани BGP RFC документима (морају да их подржавају све стандардне BGP имплементације) и морају да буду придружени свакој рути. То су: *Next Hop, AS-Path* и *Origin*.
- Добро познати необавезни атрибути (енг. *Well Known Discretionary*). Ово су атрибути који су описани BGP RFC документима (морају да их подржавају све BGP имплементације), али не морају да буду придружени свакој рути. То су: *Local Preference* и атрибути који служе за агрегацију адреса као што је *Atomic Aggregate*.
- Опциони атрибути који не морају бити подржани у свакој имплементацији BGP протокола. Они се деле на:
  - транзитивне, када неки рутер прослеђује руту коју је добио са овим атрибутом без обзира на то да ли ће искористити атрибут у процесирању руте или не. Овакав атрибут је *Community*.

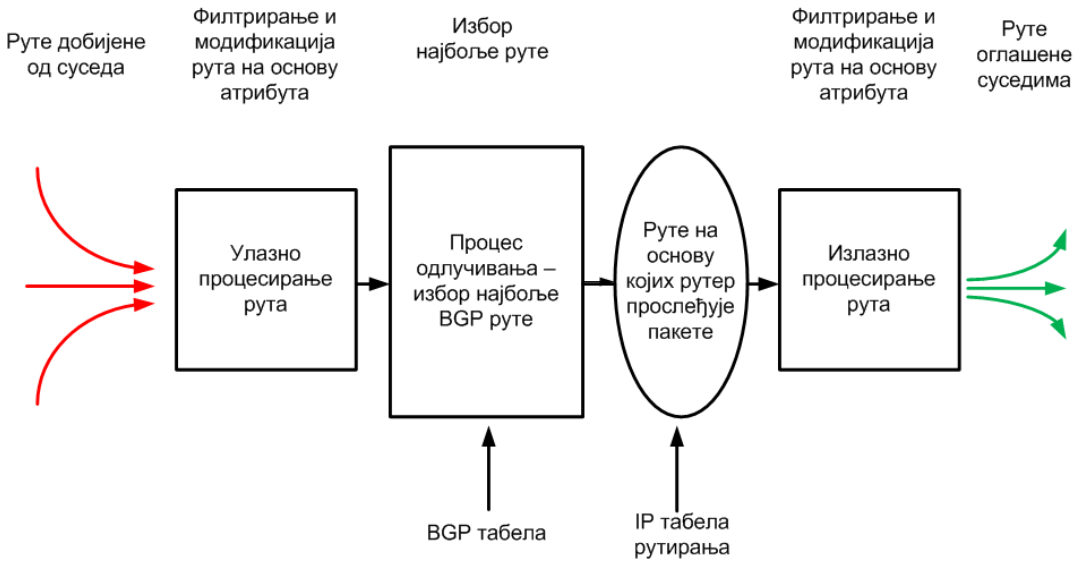
- нетранзитивне, када неки рутер прима руту са овим атрибутом, али је не прослеђује даље. Овакав атрибут је MED.

Сви кључни атрибути ће бити описани у наставку овог поглавља, али пре тога је прво потребно да се објасни на који начин BGP протокол процесира руте.

2.2.4.1. Начин процесирања рута у рутерима

Слика 2.12 шематски показује све активности на процесирању BGP рута. Руте се добијају од суседа са обавезно добро познатим обавезним атрибутима, а могу да буду постављени и други атрибути. Приликом уласка рута у рутер, могуће је поставити улазне филтре и правила којима се:

- руте прослеђују неизмењене у BGP таблу.
- рутама мењају неки атрибути па се са измењеним атрибутима прослеђују у BGP таблу.
- руте филтрирају и не пролазе у BGP таблу.



Слика 2.12 Процес обраде BGP рута у рутеру [2.2]

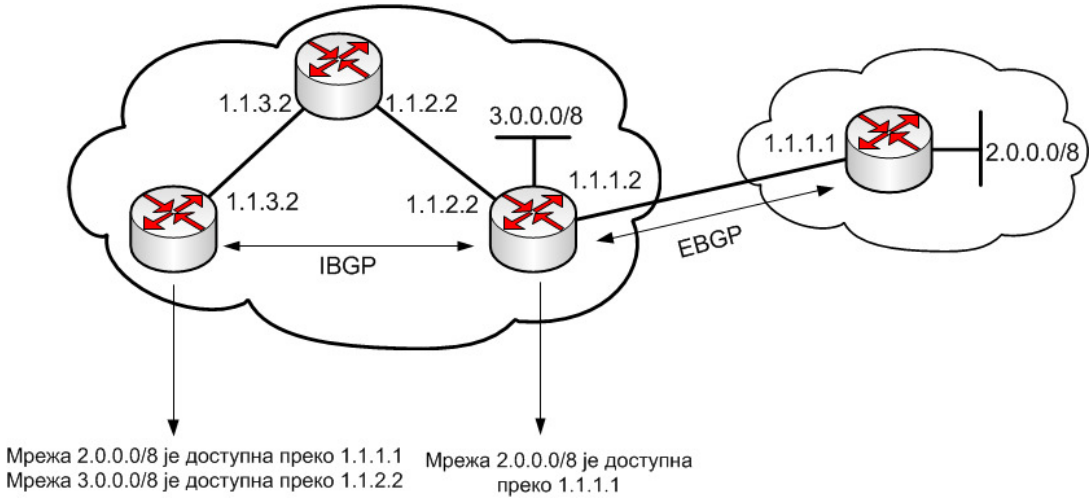
Руте добијене од свих суседа након улазног процесирања улазе у BGP таблу која се чува у меморији рутера. Од свих рута ка истој дестинационој мрежи које су добијене од различитих суседа увек се бира тачно једна најбоља рута која улази у таблу рутирања и на основу које се врши прослеђивање пакета. Механизам избора најбоље руте је описан у поглављу 2.2.5. Рутер даље оглашава своје BGP руте ка суседима и то тако да се оглашавају само оне руте које се налазе у табели рутирања, односно оне које су изабране као најбоље. Те руте се шаљу са одговарајућим атрибутима које су имале након улазног процесирања, а пре слања је могуће да се врши и излазно процесирање рута којим је могуће спровести исте акције као приликом улазног процесирања: проследити руте са неизмењеним атрибутима, проследити

их са измењеним атрибутима или их потпуно филтрирати као у примеру нетранзитног аутономног система.

2.2.4.2. Next Hop атрибут

Next Hop је један од три добро позната обавезна атрибута који морају да буду придружени свакој рути. Њиме се преноси информација о томе који је уређај огласио руту којој је придружен. За овај атрибут важе следећа правила:

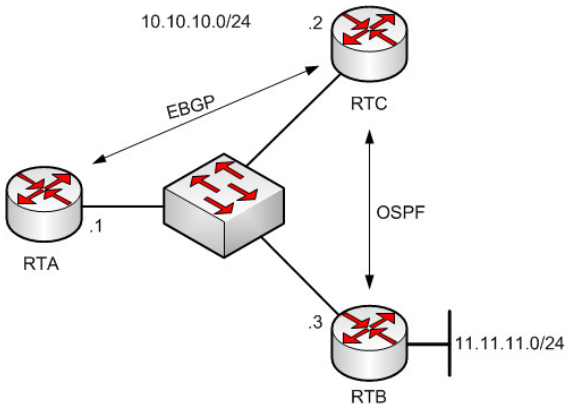
- Ако је рута добијена путем eBGP протокола, *Next Hop* атрибут ће бити адреса рутера из суседног аутономног система који је огласио дату руту (адреса 1.1.1.1 за мрежу 2.0.0.0/8 из примера са Слике 2.13).
- Ако је рута добијена путем iBGP протокола, а у питању је рута која је дошла из суседног аутономног система, *Next Hop* атрибут се неће променити, односно биће адреса рутера из суседног аутономног система који је огласио дату руту.
- Ако је рута добијена путем iBGP протокола, а у питању је рута која је оглашена унутар датог аутономног система, онда ће *Next Hop* атрибут бити адреса рутера унутар датог аутономног система који је огласио дату руту (адреса 1.1.2.2 за мрежу 3.0.0.0/8 из примера са Слике 2.13).



Слика 2.13 Вредности Next Hop атрибута

Први предуслов да би нека BGP рута била убачена у табелу рутирања је да у табели рутирања постоји рута ка мрежи на којој је *Next Hop* адреса (што је још једна од заштита од тзв. црних рупа у рутирању, јер ако рутер нема руту ка *Next Hop* адреси, тешко је очекивати да ће моћи да проследи пакете према дестинацији). Ако је у питању *Next Hop* атрибут за руту која је оглашена унутар датог аутономног система (рута 3.0.0.0/8 из примера), онда се *Next Hop* атрибут природно оглашава интерним протоколом рутирања. Ако је у питању *Next Hop* атрибут за руту која је оглашена из другог аутономног система (рута 2.0.0.0/8 из примера), онда је уобичајена пракса да се мрежа на којој је *Next Hop* адреса (мрежни сегмент између два аутономна система) дода у интерни протокол рутирања и на тај начин оглашава.

На тзв. *multiaccess* сегментима (на којима има више од два уређаја на истој мрежи, на истом LAN-у – ово су све етернет мреже) постоји једна модификација претходно датих правила.



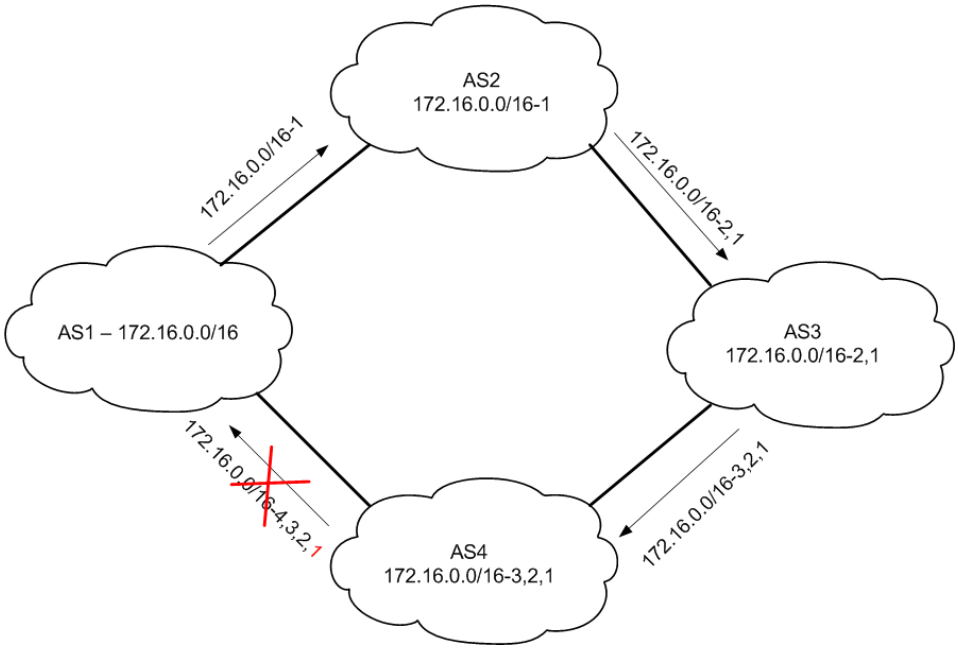
Слика 2.14 Next Hop атрибут на етернет мрежама

На слици 2.14 је дат пример такве мреже. Рутери A и C су eBGP суседи. Рутер B оглашава мрежу 11.11.11.0/24 путем интерног протокола рутирања ка рутеру C, који ову мрежу оглашава даље ка рутеру A путем BGP протокола. Када би се *Next Hop* атрибут за мрежу 11.11.11.0/24 одредио као у претходно изнетим правилима, његова вредност би била 10.10.10.2, односно адреса рутера C. Међутим, ово би довело до неоптималног прослеђивања пакета од рутера A према мрежи 11.11.11.0/24: пакети би од рутера A били послати ка рутеру C, одакле би требало да се пошаљу ка рутеру B, односно, требало би да буду послати на интерфејс рутера C преко кога су и дошли, што би изазвало активирање механизма *ICMP Redirect* којим би се захтевало слање пакета директно од рутера A ка рутеру B за дестинацију 11.11.11.0/24. Да би се овај корак избегао, начин доделе *Next Hop* атрибута на *multiaccess* сегментима је да се додели адреса оног рутера који је огласио дату руту, а у овом примеру је то адреса 10.10.10.3, односно адреса рутера B.

2.2.4.3. AS-Path

Други добро познати обавезни атрибут је *AS-Path*. Овај атрибут представља листу свих аутономних система кроз који је прошла нека рута и представља основни механизам заштите BGP протокола од стварања петљи у рутирању.

На слици 2.15 је приказан начин пропагације руте 172.16.0.0/16 од аутономног система 1 који је оглашава према аутономним системима 2, 3 и 4. Сваки аутономни систем ће приликом оглашавања руте дописати свој број аутономног система у *AS-Path* листу (дописивање броја аутономног система се зове на енглеском *prepending*) и на слици може да се види како се ова листа продужава сваким наредним оглашавањем. Уколико би аутономни систем приликом добијања руте у *AS-Path* листи препознао свој број аутономног система (ситуација у којој би аутономни систем 4 послао руту 172.16.0.0/16 ка аутономном систему 1), та рута би била одбачена да не би дошло до петљи у рутирању. У *AS-Path* листи у рутерима на интернету могу да се нађу само јавни бројеви аутономних система.

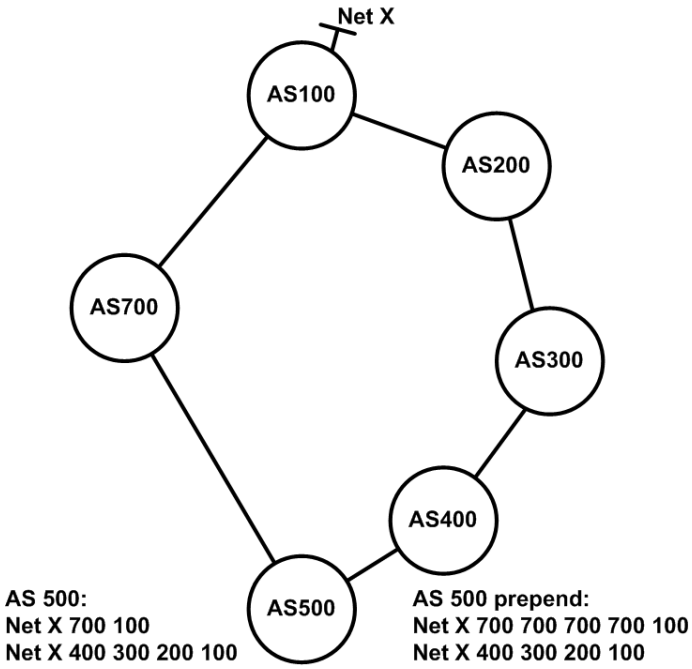


Слика 2.15 Заштита BGP протокола од стварања петљи у рутирању помоћу AS-Path атрибута

AS-Path је један од атрибута који учествују у избору најбоље руте. Уколико се добије иста рута од два суседа, као боља би била изабрана она рута која има краћи *AS-Path*, што је и природно очекивана метрика – мањи број аутономних система до дестинације. Пошто је ово један од атрибута који учествује у избору најбоље руте, могуће је манипулисати овим атрибутом како би се административно одредила најбоља путања. *AS-Path* је могуће мењати како у улазном, тако и у излазном процесирању BGP рута. Ако се *AS-Path* мења приликом долазног процесирања, онда се у *AS-Path* листу додаје један или више додатних бројева аутономног система из ког је добијена дата рута, а ако се мења приликом излазног процесирања, онда се у *AS-Path* додаје један или више бројева аутономног система који оглашава дату руту. У оба случаја се додавањем додатних аутономних система у *AS-Path* листу датој рути „квари“ метрика, односно чини се да у избору најбоље руте на основу *AS-Path* атрибута рута има дужу путању и буде мање пожељна за избор. Ово је приказано у примеру са слике 2.16.

Аутономни систем 500 добија руту ка мрежи X преко две путање: (700, 100) и (400, 300, 200, 100). Ако би се користио *AS-Path* као критеријум за избор најбоље руте за рутирање пакета према мрежи X, била би одабрана путања преко аутономних система 700 и 100 као она са краћим *AS-Path*-ом дужине 2 аутономна система. Међутим, овакав избор можда не одговара аутономном систему 500 (нпр. веза 500-700 је загушена или малог капацитета). Ако би желео да се пакети према мрежи X рутирају десном путањом, преко аутономних система 400, 300, 200 и 100, аутономни систем 500 би требало да приликом улазног процесирања руте X коју добија од аутономног система 700 дода још три ознаке аутономног система из ког је добио руту (дакле 700). Тиме би *AS-Path* ове руте био (700, 700, 700, 700, 100), односно његова

дужина би била 5, што је веће од друге путање која је дужине 4. Овим би се учинило да пакети од аутономног система 500 ка мрежи X иду десном путањом.



Слика 2.16 Пример промене AS-Path атрибута у циљу избора најбоље путање

Ако се жели да се утиче на супротни смер пакета - како улазе у аутономни систем, онда би требало додавати у *AS-Path* додатне ознаке оног аутономног система који оглашава своје руте онолико пута колико је потребно да се оствари жељени ефекат. Ако би аутономни систем 500 желео да утиче на то како ће пакети од мреже X долазити ка њему и то тако да и у долазном смеру иду путањом 100, 200, 300, 400, 500, требало би да своје руте које потичу из аутономног система 500 оглашава са три додатне вредности аутономног система 500, тако да када дођу до аутономног система 100 ове руте имају следећи *AS-Path*: (500, 500, 500, 500, 100) и буду „лошије“ од рута које долазе са друге стране. Практични пример манипулације *AS-Path* атрибутом и реализације додавања бројева аутономних система је показан у поглављу 7.4.2.

2.2.4.4. Origin

Последњи добро познати обавезни атрибут је *Origin*. Овај атрибут означава порекло руте и поставља га онај рутер који оглашава дату руту. BGP спецификација наводи да ни један рутер на путањи оглашавања руте не би требало да мења овај атрибут, али ово није експлицитно забрањено. *Origin* је у BGP *Update* порукама један бајт који може да има следеће три вредности:

- 0 која се назива IGP и означава да је рута добијена експлицитним конфигурисањем на рутеру (помоћу *network* команде, што је показано и у поглављу 7.4.2).

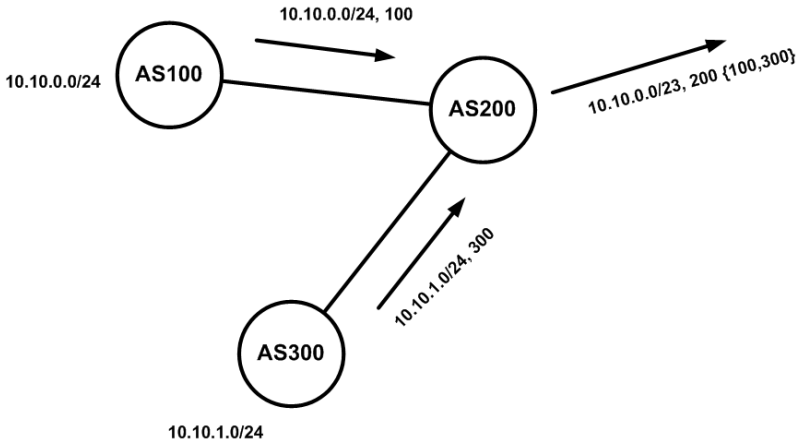
- 1 која се назива EGP и према BGP спецификацији означава да је рута добијена из EGP протокола. Пошто се овај протокол не користи већ више од 20 година, ни једна рута у интернет табелама рутирања не би требало да има ову вредност *Origin* атрибута. Међутим, пракса је показала да се у пуним интернет табелама рутирања и данас налазе руте са овим пореклом, а то је због тога што су неки произвођачи мрежне опреме оставили могућност да се *Origin* произвољно сетује или мења [2.15].
- *2* која се назива *Incomplete* и која означава да је рута добијена редистрибуцијом из неког другог протокола рутирања.

И *Origin* атрибут се користи у избору најбоље руте. Боље су оне руте које имају нижу вредност *Origin* атрибута.

2.2.4.5. Атрибути који подржавају агрегацију рута

Агрегација рута је ситуација у којој се више рута са краћом маском замењује једном рутом која обухвата адресне просторе мањих рута. Тиме се постижу смањење броја рута које треба оглашавати и мање и ефикасније табеле рутирања. На пример, уместо оглашавања посебних рута 192.168.0.0/24, 192.168.1.0/24, 192.168.2.0/24 и 192.168.3.0/24 ефикасније је огласити само руту 192.168.0.0/22 која обухвата адресне просторе све ове четири руте, наравно, ако положај ових рута у мрежи и оглашавање агрегиране руте то омогућавају. BGP подржава агрегацију рута помоћу следећих атрибута:

- ATTOMIC_AGGREGTE – који може да има две вредности: *True* или *False*. Ако је вредност атрибута *True*, то означава да је дата рута агрегирана.
- AGGREGATOR који означава адресу BGP рутера који је извршио агрегацију.
- AS_SET – који представља скуп аутономних система из којих потичу компоненте дате агрегиране руте.



Слика 2.17 Агрегација рута и BGP протокол

Приликом агрегације губе се неке информације о рутама које су компоненте агрегиране руте (нпр. две компоненте могу да имају различите вредности *Origin* атрибута, а агрегирана рута

само једну од те две вредности), као и информација о пореклу компоненти руте. Како би се избегле петље у рутирању потребно је да се агрегиране руте оглашавају са AS_SET атрибутом. У примеру са слике 2.17 ако би се агрегирана рута ка мрежи 10.10.0.0/23 вратила неком путањом ка аутономним системима 100 или 300, они би дату руту могли да одбаце јер би препознали свој број аутономног система у оквиру AS_SET атрибута.

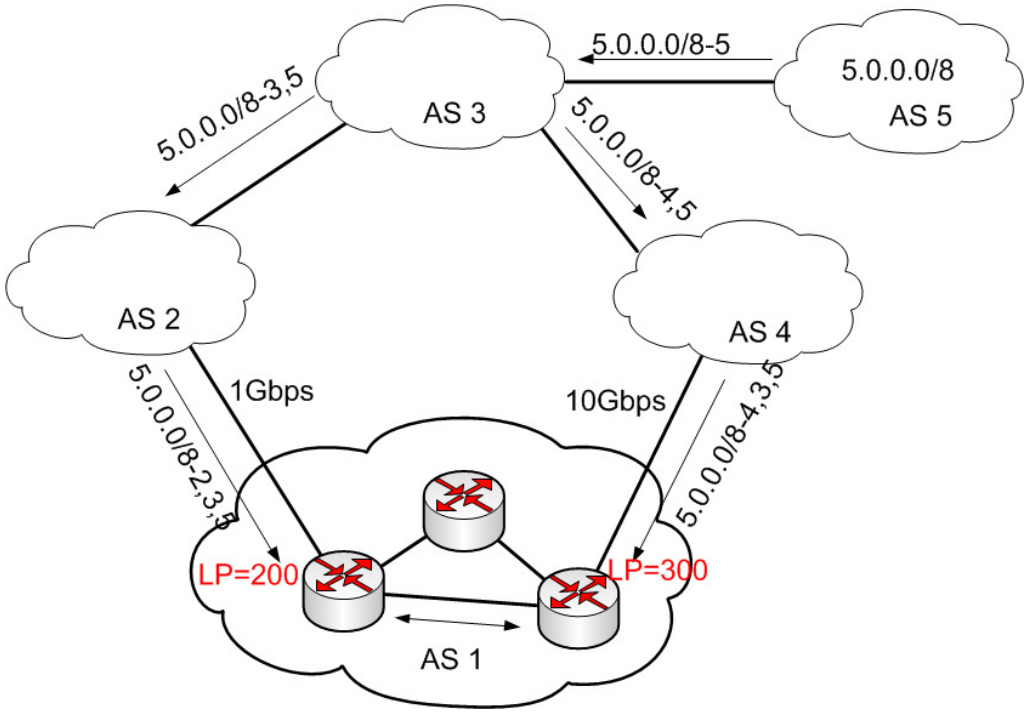
Агрегација рута у којој се руте из више аутономних система агрегирају у једну се данас практично не користи на интернету. Разлога је више, а кључни су ти да се агрегацијом мења порекло руте, што је у супротности са начином на који се штити стабилност интернета од лажног оглашавања рута (поглавље 2.2.8) и тај што је повезаност аутономних система на интернету тако густа да је тешко наћи место на којем би два суседна аутономна система имала и суседне адресне префиксе који би омогућавали агрегацију. Много је чешћа управо супротна ситуација, да аутономни системи оглашавају парцијалне делове својих адресних простора како би се остварило оптималније рутирање што и доприноси сталном порасту броја рута на интернету.

2.2.4.6. Local Preference

Local Preference је добро познати али необавезни атрибут. Он представља број који се додељује некој рути који служи касније у процесу избора најбоље руте. Ако две руте имају различиту вредност *Local Preference* атрибута, као боља ће бити узета она која има већу вредност. *Local Preference* атрибут има следеће особине:

- поставља се приликом уласка руте у неки рутер
- размењује се искључиво путем iBGP протокола. Ово значи да је овај атрибут локалан за један аутономни систем. Вредности *Local Preference* атрибута постављене у једном аутономном систему неће бити оглашене у други аутономни систем.

Пример коришћења *Local Preference* атрибута је дат на слици 2.18. У овом примеру рута 5.0.0.0/8 се оглашава према аутономном систему 1 и у њега долази преко две путање: (5, 3, 2) и (5, 3, 4). Дужине *AS-Path* путања су једнаке у овом случају, те овај атрибут не може да одлучи у избору најбоље руте. Међутим аутономни систем 1 жели да пакети ка мрежи 5.0.0.0/8 излазе преко путање (4, 3, 5) јер је веза ка аутономном систему 4 већег капацитета од везе ка аутономном систему 2. Док би једно решење за ово било да се *AS-Path prepending*-ом поквари рута која долази преко аутономног система 2, исто може да се изведе постављањем *Local Preference* атрибута за руту 5.0.0.0/8 када улази у аутономни систем 1. Ако се постави *Local Preference* за руту 5.0.0.0/8 када долази из аутономног система 2 на 200, а када долази из аутономног система 4 на 300, руте са овако постављеним *Local Preference* вредностима ће се разменити унутар аутономног система 1 путем iBGP и сви рутери унутар овог аутономног система ће као бољу руту изабрати ону са вишом вредношћу *Local Preference* атрибута, односно ону која је дошла из аутономног система 4. Начин на који се поставља *Local Preference* атрибут и механизам његове размене су показани у практичним примерима у поглављу 7.4.2.

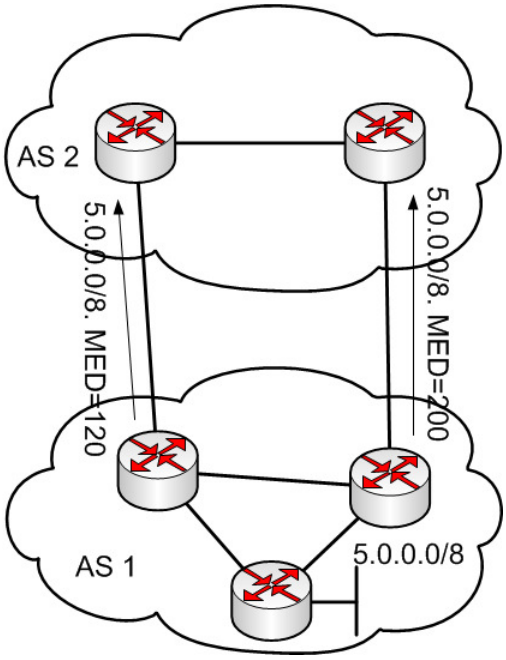


Слика 2.18 Начин коришћења Local Preference атрибута у циљу избора најбоље путање

2.2.4.7. Multi-Exit Discriminator – MED

Док се *Local Preference* атрибут додељује рутама које улазе у аутономни систем и утиче на начин на који ће пакети одлазити из аутономног система, MED атрибут се додељује рутама које излазе из аутномног система са циљем да утиче на то како ће пакети улазити у аутономни систем. MED атрибут је опциони нетранзитивни атрибут, што значи да када се пошаље суседном аутономном систему, он га неће проследити даље, наредним аутономним системима. Ово такође значи и да је основна намена MED атрибута да се користи у оним ситуацијама када један аутономни систем има више веза према суседном и да тада допринесе у одлучивању о томе која ће рута бити одабрана као најбоља.

Начин коришћења MED атрибута је показан на слици 2.19. Аутономни систем 1 оглашава према аутономном систему 2 руту ка мрежи 5.0.0.0/8 преко две везе које поседује. Ако жели да утиче на то да пакети према мрежи 5.0.0.0/8 улазе преко леве везе, аутономни систем 1 треба да постави нижу вредност MED атрибута када оглашава руту ка мрежи 5.0.0.0/8 преко те везе (у примеру постављена вредност 120) него када је оглашава преко десне везе (у примеру постављена је вредност 200). Приликом избора боље руте аутономни систем 2 ће одабрати ону руту која има нижу вредност MED атрибута. Уколико би аутономни систем 2 даље оглашавао руту ка мрежи 5.0.0.0/8, он је не би оглашавао са вредношћу MED атрибута коју је поставио аутономни систем 1 јер је реч о нетранзитивном атрибуту. Ово даље значи да MED атрибут не може да се користи за избор боље руте ако пакети долазе из два различита аутономна система, јер ови аутономни системи неће моћи да размене MED атрибут који је поставио аутономни који оглашава руту.



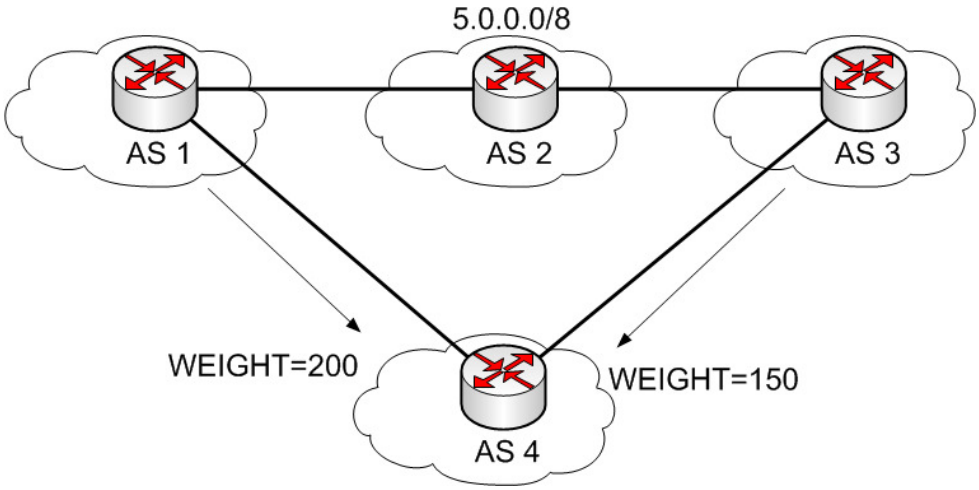
Слика 2.19 Начин коришћења MED атрибута за избор најбоље путање

MED атрибут се још назива и метрика (енг. *metric*) јер је једна од идеја била да се у њега унесе вредност метрике интерног протокола рутирања из аутономног система који оглашава неку руту, тако да у аутономни систем пакети улазе на улазе који су тополошки најближи дестинационим мрежама.

2.2.4.8. Weight

Weight атрибут такође може да се користи у избору најбоље руте, али није прави BGP атрибут јер се не преноси BGP порукама, већ је локалан за један рутер. Слично *Local Preference* атрибуту, рута са вишом вредношћу *Weight* атрибута је она која ће бити убачена у табелу рутирања, што је и показано на примеру са слике 2.20. На овом примеру ће путања пакета од аутономног система 4 ка мрежи 5.0.0.0/8 бити преко аутономних система 1 и 2, јер је постављена *Weight* вредност већа за руту добијену од аутномног система 1.

Weight атрибут је карактеристичан за уређаје компаније Cisco. Ово није стандардни атрибут BGP протокола.



Слика 2.20 Начин коришћења Weight атрибута за избор најбоље путање

2.2.5. Начин одређивања најбоље путање

У претходним поглављима је показан низ атрибута који утичу на начин одређивања најбоље путање. Руте са свим овим атрибутима се смештају у BGP табелу чији је пример приказан на слици 2.21. За руту 3.0.0.0 се види да је добијена од 3 суседа чије су *Next Hop* адресе дате, да нису постављане посебне вредности за *Weight* и *Local Preference* (*LocPrf*) атрибуте и да је за руту добијену од суседа 195.178.34.57 постављена MED вредност 150. Такође за све руте се виде *AS-Path* листе и може да се види да је за руте добијене од аутономног система 8400 примењено једно додатно *prepend*-овање када су руте ушле у аутономни систем. На крају сваког реда стоји слово *i* које означава да су дате руте са пореклом IGP.

```
cisco6509#sh ip bgp
BGP table version is 5011434, local router ID is 147.91.0.112
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  0.0.0.0          195.178.34.57            150             0 8400 8400 i
*>                  195.178.35.17                            0 8400 8400 i
*> 3.0.0.0          195.178.35.17                            0 8400 8400 702 703 80 i
*                   195.178.34.57            150             0 8400 8400 702 703 80 i
*                   195.251.4.44                             0 34771 5408 20965 3356 701 703 80 i
*  4.0.0.0          195.178.34.57            150             0 8400 8400 5400 3356 i
*                   195.178.35.17                            0 8400 8400 5400 3356 i
*>                  195.251.4.44                             0 34771 5408 20965 3356 i
*  4.23.84.0/22     195.178.34.57            150             0 8400 8400 5400 6461 20171 i
*>                  195.178.35.17                            0 8400 8400 5400 6461 20171 i
*                   195.251.4.44                             0 34771 5408 20965 1299 6461 20171 i
```

Слика 2.21 Извод BGP табеле рутера

Ови атрибути који су постављени за добијене руте утичу на то које ће од њих бити одабране као најбоље (оне са ознаком >). Процес избора најбоље руте је јасно дефинисан, посебно зато што је очигледно је да је могуће направити ситуацију у којој неки од ових атрибута имају супротстављене критеријуме: на пример рута која је добијена од два суседа може да има са једне стране краћи *AS-Path* и мањи *Local Preference* од руте добијене од другог суседа. По

критеријуму краћег *AS-Path*-а требало би узети прву руту, а по критеријуму веће *Local Preference* вредности другу руту.

Процес избора најбоље BGP руте се састоји из низа критеријума који се разматрају редом. Уколико један критеријум не може да одлучи која је рута боља, прелази се на следећи. Чим се пронађе критеријум који може да одлучи у избору за најбољу руту, процес се прекида. Критеријуми за избор рута су следећи:

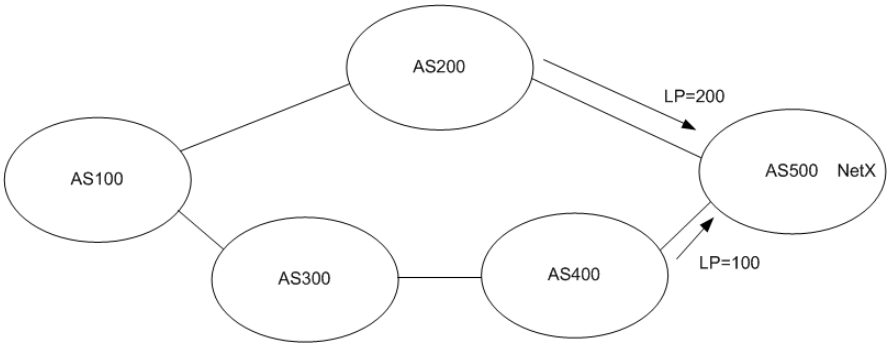
1. Ако *Next Hop* атрибут за дату руту не постоји у табели рутирања, рута се игнорише, односно не може да буде убачена у табелу рутирања.
2. (ако постоји *Weight* атрибут за дату руту и ако је дефинисан, у табелу рутирања ће ући рута са највећом *Weight* вредношћу)[11]
3. Ако су *Weight* вредности исте, у табелу рутирања ће ући рута са највећом вредношћу *Local Preference* атрибута.
4. Ако су и *Local Preference* вредности исте, у табелу рутирања улази рута са најкраћим *AS-Path*-ом.
5. Ако претходни критеријуми не могу да одлуче која је рута најбоља, рутер ће одабрати руту са нижом вредношћу *Origin* атрибута.
6. Ако су и вредности *Origin* атрибута исте, рутер ће одабрати руту са нижом MED вредношћу.
7. Ако су и MED вредности исте, рутер ће да одабере руту која је добијена од eBGP суседа пре него руту добијену од iBGP суседа.
8. Ако и претходни критеријум не може да одлучи најбољу руту, бира се она чија је метрика интерног протокола рутирања до BGP *Next hop* адресе нижа.
9. Ако су и претходни критеријуми исти, бира се рута која је добијена раније (која је прва стигла у рутер).
10. Ако су и претходни критеријуми исти бира се рута која је добијена од суседа са нижим Router ID-ем (једна од адреса рутера).
11. Ако су и претходни критеријуми исти бира се рута која има мању дужину кластера (шта је кластер ће бити објашњено у поглављу 2.2.7.1).
12. Ако су и претходни критеријуми исти бира се рута која бира се рута добијена од суседа са нижом IP адресом.

Свакако на крају процеса избора најбоље руте, мора да буде изабрана тачно једна рута која ће бити пребачена у табелу рутирања и која ће моћи да буде даље оглашена ка осталим суседима.

11 Овај атрибут као што је речено је карактеристичан за рутере компаније Cisco. Други произвођачи не подржавају овај атрибут те он није део њиховог механизма одлучивања, због тога је приказан у загради. На практичним примерима показаним у поглављу 7 може да се утиче на овај параметар, па је стога овде и показан.

2.2.6. Атрибут Community

Атрибут *Community* [2.16] је такође атрибут који се шаље уз BGP руте, али као што се види у претходном поглављу, он се не користи непосредно у избору најбоље руте. Атрибут *Community* се користи за комуникацију са удаљеним аутономним системима како би се посредно утицало на избор најбоље руте.



Слика 2.22 Пример ситуације у којој је потребан Community атрибут

Пример потребе за коришћењем овог атрибута је дат на слици 2.22. Ако власник аутономног система 100 жели да утиче на то да пакети од мреже X ка њему долазе преко аутономних система 400 и 300, једно очекивано решење би било да када AS100 оглашава своје руте ка аутономном систему 200 да их огласи са два додатна *prepending*-а свог броја аутономног система, тако да *AS-Path* атрибут на излазу из аутономног система 100 на овој вези буде (100, 100, 100). На овај начин се очекује да ће за руте из аутономног система 100 када улазе у аутономни систем 500 преко горње путање *AS-Path* бити дужи него преко доње путање и да ће то утицати на жељени долазак пакета у аутономни систем 100 преко аутономних система 300 и 400. Међутим, ако је аутономни систем 500 поставио *Local Preference* вредности као на слици 2.22, тако да је вредност већа за оне руте добијене од аутономног система 200, овај *prepending* који врши аутономни систем 100 неће имати никакав ефекат јер је критеријум за избор најбоље руте на основу вредности *Local Preference* атрибута већег приоритета од критеријума на основу дужине *AS-Path.* Ово заправо значи да аутономни систем 100 манипулацијом свим претходно описаним атрибутима не може никако да саобраћај од мреже X ка свом аутономном систему усмери преко аутономних система 300 и 400 осим у случају гашења везе преко аутономног система 200.

Да би се овај проблем некако решио осмишљен је концепт *Community* атрибута. Овај атрибут је 32-битна вредност коју користе транзитни аутономни системи да би омогућили да се утиче на друге атрибуте које они постављају. Постоје неке предефинисане вредности *Community* атрибута:

- 0xFFFFFF01 – *No Export* која означава да рута са овом вредношћу *Community* атрибута не треба да буде оглашена eBGP суседима и,
- 0xFFFFFF02 – *No Advertise* која означава да рута са овом вредношћу *Community* атрибута не треба да буде оглашена никоме.

За остале вредности *Community* атрибута уобичајен начин представљања (за 16-битне бројеве аутономног система) је *AS:Community*, а аутономни системи су слободни да сами доделе вредности *Community* атрибута и акције које оне означавају. На пример велики провајдер првог слоја, компанија Level3 (аутономни систем 3356) има дефинисане следеће вредности Community атрибута које утичу на Local Preference атрибут[12]:

```
3356:70- set local preference to 70
3356:80- set local preference to 80
3356:90- set local preference to 90
```

Слика 2.23 Пример Community атрибута провајдера

Сличне вредности атрибута постоје и за друге атрибуте, као што је додавање аутономних система у *AS-Path* или промена MED вредности. Вредности *Community* атрибута аутономни системи постављају да буду јавне, најчешће на сајту провајдера, како би други аутономни системи могли да их искористе за утицај на рутирање. У примеру са слике 2.23, аутономни систем 100 би када оглашава своје руте према аутономном систему 500 преко аутономног система 200 требало да уз ове руте пошаље атрибут *Community* који има такву вредност да поставља *Local preference* на вредност која је мања или једнака вредности коју поставља аутономни систем 500 за руте које долазе дољом путањом. Тада би или било непотребно да се користи *AS-Path prepending* у аутономном систему 100, ако је нова вредност *Local preference* мања од доње или би *AS-Path prepending* извршио очекивани утицај ако су вредности *Local preference* једнаке.

2.2.7. Скалабилност iBGP

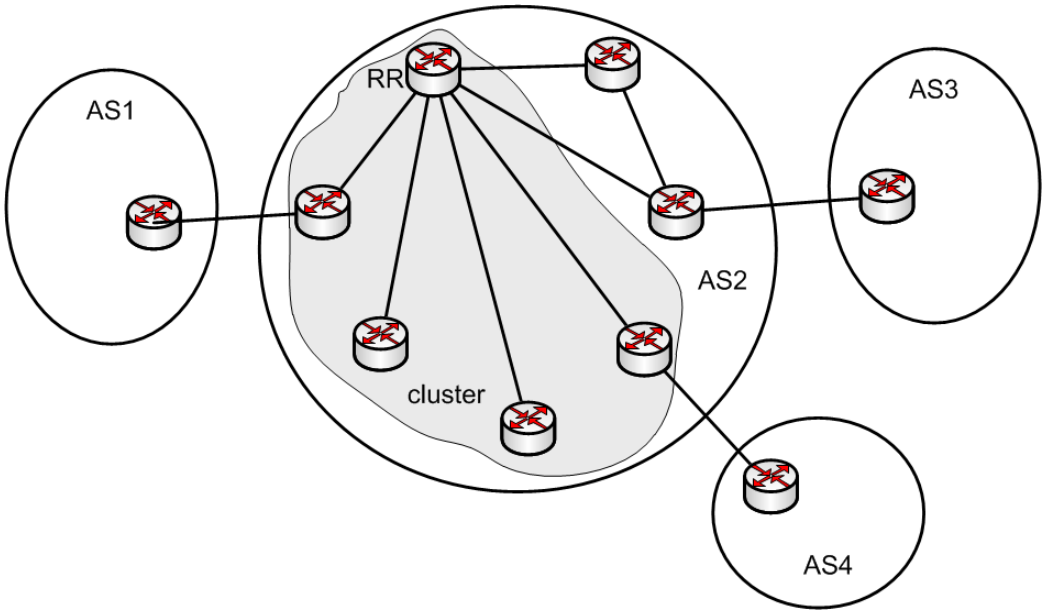
Као што је показано у поглављу 2.2.3.3, једно од најчешћих решења којим се обезбеђује континуитет рутирања у аутономном систему је успостављање потпуног графа iBGP сесија између свих рутера на путањама транзитних пакета у једном аутономном систему. Ово значи да ће број iBGP сесија бити *n(n-1)/2*, где је *n* број рутера и да на сваком рутеру треба конфигурисати сесије са свим осталим рутерима. Такође, додавање новог рутера значи његово конфигурисање, али и реконфигурацију свих осталих како би се остварио потпун граф iBGP сесија, што чини да је одржавање рутирања унутар аутономног система комплексно и нескалабилно. Иако код уређаја неких произвођача постоје механизми којима се смањује посао на одржавању конфигурација (груписање већег броја суседа у тзв. *peer* групе), све iBGP сесије морају да постоје. Да би се проблем скалабилности решио, предвиђена су два механизма која су описана у наставку овог поглавља:

- Рефлектори рута
- Конфедерације

12 Преузето са странице: http://showipbgp.com/bgp-tools/bgp-community-list/91-level3-as3356.html 4.1.2018. На овој страници могу да се виде и друге *Community* вредности које је дефинисао овај провајдер.

2.2.7.1. Рефлектори рута

Смањење броја iBGP сесија се у овој методи остварује тако што се унутар аутономног система неким рутерима даје улога рефлектора рута (енг. *route reflector*) [2.17]. Сваки рефлектор рута има своје клијенте са којима има успостављене iBGP сесије. Рефлектор рута са скупом својих клијената чине кластер.



Слика 2.24 Аутономни систем у којем је конфигурисан један кластер са једним рефлектором рута и четири клијента у кластеру

Рефлектор рута је рутер за који у неким ситуацијама не важи ограничење да ако добије руту путем iBGP не сме да је проследи другим iBGP суседима. Што се тиче осталих правила понашања iBGP рутера, ту нема промена код рефлектора рута. Они неће мењати Next Hop приликом преношења рута и преносиће све атрибуте рута који су карактеристични за iBGP (нпр. *Local Preference*). За рефлекторе рута важе следећа специфична правила:

- Ако је рута добијена од iBGP суседа рефлектора рута (неклијент), она може да буде прослеђена клијентима тог рефлектора рута.
- Ако је рута добијена од једног клијента, рефлектор рута ће је проследити осталим клијентима, као и осталим iBGP суседима рефлектора рута (неклијенти).
- Ако је рута добијена од eBGP суседа, рефлектор рута ће је проследити како својим клијентима, тако и другим iBGP суседима рефлектора рута (неклијентима).

Из претходног се види да је рефлектор рута који унутар аутномног система прослеђује руте у име својих клијената и за њих. Ефекат који је постигнут оваквим механизмом рада рефлектора рута је тај да уместо да постоји потпун граф iBGP сесија у аутономном систему, у кластеру је довољно да постоји само по једна iBGP сесија између сваког клијента и рефлектора рута (Слика 2.24), чиме се очигледно смањује број iBGP сесија у аутономном систему. У примеру са слике унутар кластера има само 4 iBGP сесије, док би без рефлектора

рута само у том делу мреже са 5 рутера морало да постоји 10 iBGP сесија, а морале би да постоје и остале iBGP сесије између клијената овог кластера и осталих рутера неклијената како би се повезали сви рутери у режиму „свако са сваким“.

У једном аутономном систему не морају да сви рутери да буду у неком кластеру, већ могу да постоје и „обични“ iBGP рутери. Између рефлектора рута који су у истм или различитим кластерима, ако има више кластера, или између рефлектора рута и осталих iBGP рутера неклијената и даље мора да постоји потпун граф iBGP сесија.

Рефлектор рута је рутер који у свом кластеру представља тачку чијим отказом долази до дељења аутономног система на делове који губе међусобну повезаност (енг. *Single Point of Failure* – SPOF) јер клијенти немају конфигурисане iBGP сесије према неклијентима. Због тога је уобичајено да се кластери организују тако што у сваком постоје најмање два рефлектора рута, а сваки клијент у том кластеру има по једну iBGP сесију према сваком рефлектору рута. Такође, рефлектори рута су рутери који постају више оптерећени од осталих рутера, те је потребно да се води рачуна о перформансама приликом избора уређаја, али и о томе каква им је локација у мрежи у односу на рутере њихове клијенте. Такође, могуће су конфигурације у којима један рефлектор рута може да буде задужен за више кластера истовремено.

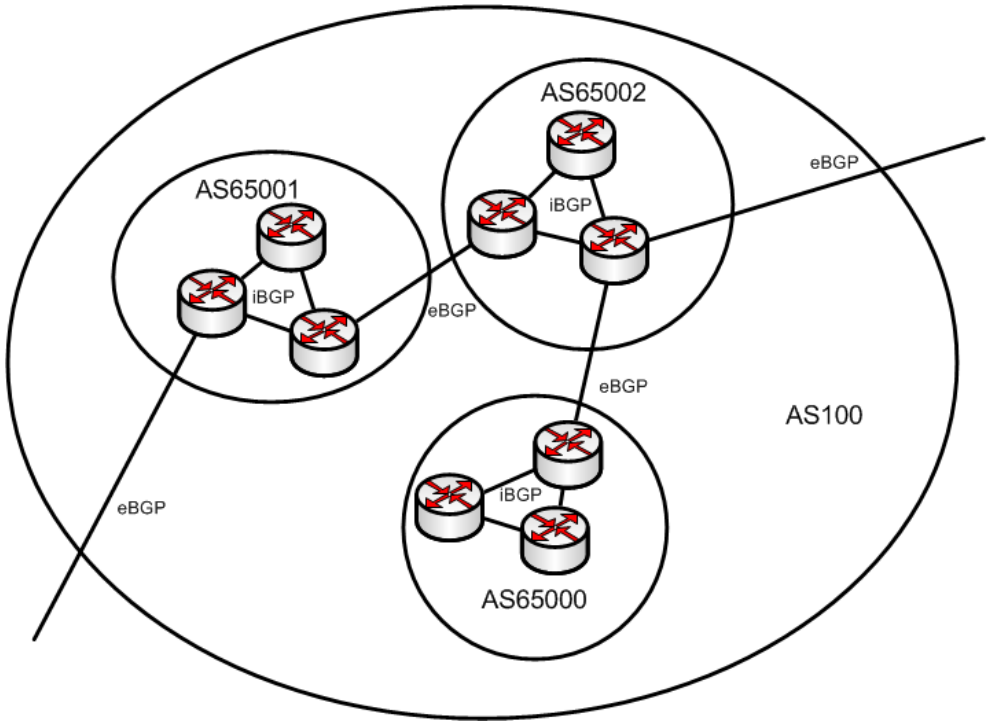
Механизам којим BGP спречава петље у рутирању је провера *AS-Path*-а руте која је добијена и уколико се у путањи види број аутономног система који добио руту, рута се одбацује. Пошто се механизмом рефлектора рута мења класичан начин рада BGP протокола и омогућава iBGP рутерима да крше правило које је уведено да би се спречиле петље у рутирању, отварају се могућности да неисправним конфигурацијама дође до услова за стварање петљи. Да би се ово спречило уведена су два нова атрибута:

- ORIGINATOR_ID - опциони нетранзитивни атрибут у који се уписује идентификатор (адреса) оног рутера који је огласио дату руту.
- CLUSTER_LIST – такође опциони нетранзитивни атрибут у који се уписује листа кластера кроз који је прошла рута којој је придружен овај атрибут. Аналогно *AS-Path* атрибуту када рефлектор рута из једног кластера проследи руту коју је добио од свог клијента, он ће тој рути додати идентификатор кластера из ког је потекла рута, а сваки следећи рефлектор рута ако рута пролази кроз већи број кластера ће додавати нове ознаке кластера. Ако би нека рута требало да дође у кластер кроз који је већ прошла, она ће бити одбачена.

2.2.7.2. Конфедерације

Други начин за решавање проблема броја iBGP сесија је механизам конфедерација [2.18]. Код овог механизма се велики аутономни систем дели на низ мањих (под)аутономних система (енг. *Sub-AS*). Ови подаутономни системи добијају бројеве аутономних система из скупа приватних бројева. Унутар подаутономних система и даље важе правила да мора да постоји потпун граф iBGP сесија, али између подаутономних система се успостављају eBGP

сесије код којих не постоји било какво ограничење у смислу прослеђивања BGP рута (Слика 2.25). На тај начин, дељењем великог аутономног система у мање јединице се спречава велики пораст броја iBGP сесија унутар великог, правог, аутономног система.



Слика 2.25 Пример аутономног система подељеног у конфедерацију три под-аутономна система

Стратегија са дељењем аутономног система и њеним ефектом на број iBGP сесија је интуитивно јасна. Такође јасно је и то да постојање подаутономних система чини да код коннфедерација не може да дође до стварања петљи у рутирању јер ће бројеви под-аутономних система моћи да буду уписани у *AS-Path*, па не постоји потреба за новим атрибутима као код рефлектора рута. Ипак, увођење eBGP сесија унутар аутономног система није тривијално.

Наиме, пошто главни аутономни систем мора и када је подељен у конфедерацију да задржи понашање класичног аутономног система, то значи да преношење атрибута мора да буде непромењено. Проблем који настаје увођењем конфедерација је последица тога што се неки атрибути који се размењују унутар аутономног система преносе само iBGP сесијама (нпр. *Local Preference*), док су неки атрибути (нпр. MED) нетранзитивни. Ово значи да у примеру са слике ако се *Local Preference* постави у оквиру подаутономног система 65001, он не би могао да буде послат према другим подаутономним системима јер су везе између њих eBGP. Такође, ако би подаутономни систем 65001 добио руту са постављеним MED атрибутом, он не би смео да проследи даље овај нетранзитивни атрибут према подаутономном систему 65002. Из овога је јасно да понашање eBGP сесија између подаутономних система не може да буде исто као понашање eBGP сесија између класичних аутономних система јер би се на

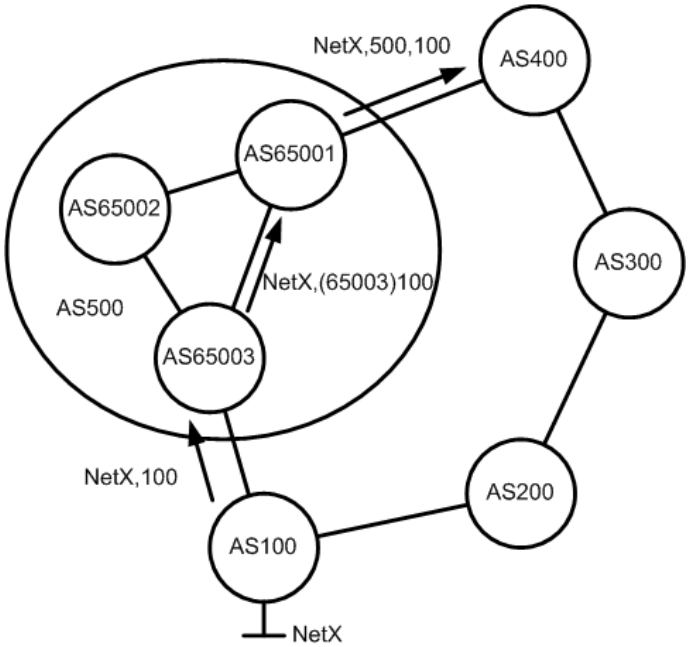
тај начин нарушило понашање правог, главног аутономног система у смислу BGP протокола. Дакле, увођењем конфедерација постојаће практично три врсте BGP сесија:

- класичне eBGP сесије између правих аутономних система
- класичне iBGP сесије унутар подаутономних система
- „интерне“ eBGP сесије између подаутономних система које се делимично понашају као eBGP сесије (бележе у *AS-Path* подаутономне системе кроз који је прошла рута), али делимично и као iBGP сесије (преносе типичне iBGP атрибуте попут *Local Preference*, и нетранзитивне атрибуте попут MED)

Ово даље повлачи и промене у механизму избора најбоље руте. Критеријум 7 из поглавља 2.2.5 ће бити модификован на следећи начин:

7. Уколико на неком рутеру постоји рута ка некој мрежи добијена из суседног подаутономног система („интерна“ eBGP рута) и рута добијена од суседног правог аутономног система (класична eBGP рута) и сви претходни критеријуми су једнаки, биће одабрана она рута која је добијена од класичног eBGP суседа.

Уколико на неком рутеру постоји рута ка некој мрежи добијена од iBGP (унутар подаутономног система) и рута добијена од суседног подаутономног система („интерна“ eBGP рута) и сви претходни критеријуми су једнаки, одабраће се она рута која је добијена од суседног подаутономног система („интерна“ eBGP рута).



Слика 2.26 AS-Path унутар конфедерације

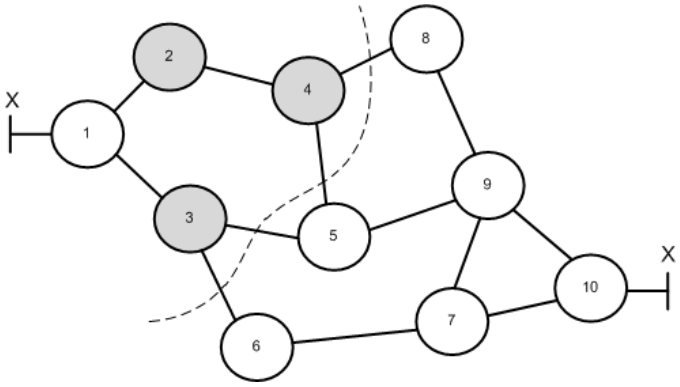
Када руте пролазе кроз аутономни систем са конфедерцијом, у *AS-Path* се додају бројеви подаутономних система, што представља заштиту од стварања петљи унутар аутономног система, али пошто су у питању приватни бројеви аутономних система, ови бројеви морају

да буду уклоњени када се рута оглашава ка следећем правом аутономном систему (Слика 2.26). Такође, приликом избора најбоље руте унутар аутономног система са конфедерацијом не узима се у обзир краћа путања подаутономних система, већ искључиво метрика интерног протокола рутирања (дакле, на слици 2.26 није бољa путања која иде преко подаутономних система 65003 и 65001 од оне која иде путањом 65003-65002-65001).

2.2.8. Очување стабилности интернета

Интернет је највећа дистрибуирана инфраструктура на свету коју данас свакодневно користи више милијарди корисника, а у којем активно учествује преко 80.000 аутономних система. Сви ови аутономни системи учествују у функционисању интернета кроз конфигурацију BGP протокола и оглашавање рута. С обзиром на тако велику групу активних учесника, значај који интернет има у свакодневном животу и све веће проблеме са сигурношћу рачунарско-комуникационих система, природно је очекивати да из неких аутономних система долазе потенцијални напади на стабилност целог система, било намерно, било као последица отказа на уређајима или софтверу које користе [2.19]. У овом поглављу ће бити објашњена два проблема и начин на који они се они отклоњају или умањују. То су:

- Лажно оглашавање BGP рута или BGP *Hijacking* – појава да неки аутономни систем оглашава намерно или конфигурационом грешком руте које им не припадају. Шта се дешава у овој ситуацији показује слика 2.27. На њој је аутономни систем 10 власник адресног простора X и оглашава га ка суседним аутономним системима. Уколико би аутономни систем 1 почео да оглашава исте руте X, а оне му не припадају, онда би они аутономни системи који су у његовој непосредној близини (до којих је *AS-Path* краћи, ако је то критеријум за избор најбоље руте), а у овом примеру сигурно аутономни системи 2, 3, и 4 као најбољу путању ка мрежи X поставили путању ка аутономном систему 1. Како аутономни систем 1 не поседује заиста уређаје који су на адресама из скупа адреса X, пакети из аутономних система 2, 3 и 4 би дошли до аутономног система 1 и ту били одбачени. Тиме би сви корисници из аутономних система 2, 3 и 4 били ускраћени за могућност да користе услуге које су на адресама мреже X или би се створила могућност да неки активан нападач из аутономног система 1 подметне лажну услугу аутономног система 10 која се регуларно налази на адресама X. Упркос постојању механизма који ово спречава, а који је објашњен даље у тексту, овакве ситуације се и даље дешавају [2.20][2.21][2.22].



Слика 2.27 Пример за BGP Hijacking

- нестабилност рута – Највећи број рутера на интернету прима пуну интернет табелу рутирања у којој се налазе све руте свих аутономних система. Промена руте из неког аутономног система (нпр. повлачење или оглашавање) потенцијално може да пропагира до свих ових рутера. Постоје ситуације када због квара на мрежном уређају или вези долази до честих промена стања интерфејса – може да много пута у минуту пређе из активног у неактивно стање и обрнуто (енг. *route flapping*). Ако је то интерфејс преко кога се оглашава нека BGP рута, свако деактивирање значи повлачење те руте из BGP табела, а свако активирање значи поновно успостављање BGP сесије и оглашавање ове руте. Како се оваква активност одражава на функционисање свих рутера до којих дата рута долази, пречесте промене статуса рута могу да утичу на оптерећење великог броја рутера на интернету што није пожељно. Због тога је реализован механизам привременог одбијања оглашавања рута којим се овај проблем решава, а који је описан у наставку текста.

2.2.8.1. Објекти и филтери рута

Да би се спречило лажно оглашавање рута које аутономни системи не поседују и немају право да оглашавају, регионални интернет регистри су у оквиру својих база направили могућност да се креирају базе тзв. објеката рута (енг. *route objects*), како за IPv4 тако и за IPv6 руте. Како регионални интернет регистри додељују и бројеве аутономних система и IP адресни простор LIR-евима, они могу и да изврше проверу тога да ли неки аутономни систем поседује или има право да огласи одређени адресни опсег[13]. Текстуални оквири ниже у тексту показују пример IPv4 и IPv6 објеката рута Академске мреже Србије креираних у RIPE. Сви ови објекти се скупљају и смештају у базе интернет регистра рута[14], одакле могу да их преузму сви аутономни системи. Добра пракса је да аутономни системи примају само оне руте за које су креирани одговарајући објекти тако што ће направити улазне BGP филтре који ће примати само оне руте за које постоје објекти рута које су преузели од интернет регистра рута. Како је креирање објеката рута динамично, уобичајена пракса је да велики

13 Постоје ситуације када неки аутономни систем може да оглашава адресе другог аутономног система, али то је могуће само уз сагласност оба аутономна система и узајамну потврду регионалном интернет регистру.

14 http://www.irr.net/index.html

интернет провајдери једном дневно ажурирају своје филтре формиране на основу објеката рута. Ипак, упркос постојању овог механизма, као што је показано, и даље се спорадично у неким деловима интернета дешавају инциденти са лажним оглашавањем рута, јер не постоји механизам којим аутономни системи могу да се натерају да овај механизам користе.

```
route: 147.91.0.0/16
descr: UNIVERSITY OF BELGRADE
origin: AS13092
mnt-by: UB-MNT
source: RIPE # Filtered
```

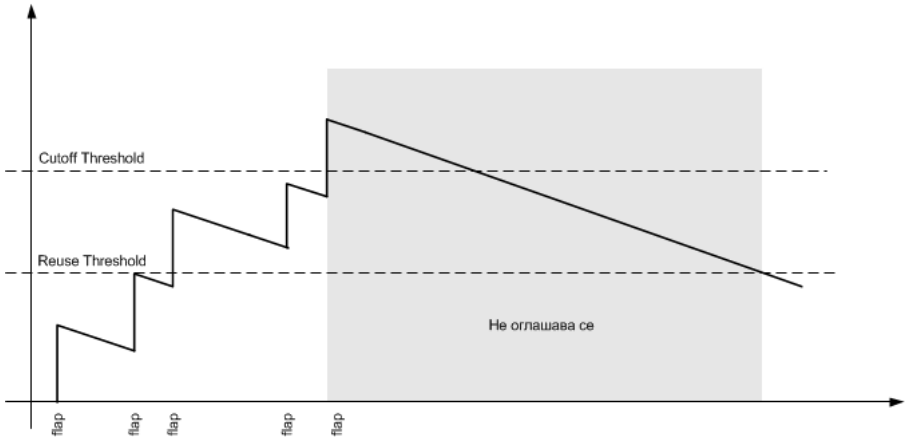
```
route6: 2001:4170::/32
descr: UNIVERSITY OF BELGRADE
origin: AS13092
mnt-by: UB-MNT
source: RIPE # Filtered
```

У поглављу 2.2.3.3 је дат један аргумент зашто редистрибуција BGP рута у интерни протокол рутирања није реално решење за мреже које се налазе на интернету (скалабилност и стабилност интерног протокола рутирања су угрожени великим бројем рута). Овде ће бити показан још један пример који указује на то да редистрибуцијом рута може да се дође до ситуације која може да представља BGP *Hijacking* ситуацију. Претпоставимо да аутономни систем 2 са слике 2.11 добија путем BGP неку руту X од аутономног система 1 и по добијању је редистрибуира у интерни протокол рутирања. Ова рута даље пропагира кроз аутономни систем 2 интерним протоколом рутирања, а затим се на другом излазу аутономног система 2 редистрибуира поново у BGP и шаље ка аутономном систему 3. Како интерни протоколи рутирања немају подршку за транспорт BGP атрибута, оног тренутка када се рута редистрибуира у интерни протокол рутирања, изгубиће се сви атрибути руте X са којима је дошла у аутономни систем 2. Приликом друге редистрибуције назад у BGP и оглашавања ка аутономном систему 3, ова рута ће бити оглашена као да потиче из аутономног система 2, што је могуће реализовати у затвореном окружењу, али је из претходно изнетог нерегуларно оглашавање на интернету.

2.2.8.2. Привремено одбијање оглашавања рута које често мењају стање

Руте које често мењају стање могу привремено да престану да буду оглашаване механизмом *Route Flap Damping* [2.23]. Овај механизам функционише тако што се свакој рути придружи један број – *Penalty*. Овај број има иницијалну вредност 0, а приликом сваке промене статуса руте (енг. *flap*) *Penalty* се повећа за неки предефинисани износ. Временом *Penalty* опада до нуле уколико нема нових промена статуса руте. Ако има пуно промена статуса руте и *Penalty* порасте преко *Cutoff Threshold* границе[15], рута ће у том тренутку престати да се даље оглашава. Престанак оглашавања руте се види као губитак свих интернет сервиса и обично мотивише аутономни систем чија је то рута да проблем који је довео до нестабилности рута реши. Када након решења проблема престану да се дешавају промене статуса руте, *Penalty* ће временом опасти на вредност која је мања од *Reuse Threshold* границе и рута ће поново моћи да се оглашава (Слика 2.28).

15 Овај назив за границу потиче из RFC документа. Произвођачи опреме ову границу називају и *Supress Limit*.



Слика 2.28 Начин рада механизма одбијања оглашавања рута

Брзина опадања вредности *Penalty* када нема промена статуса руте је дефинисана помоћу параметара *half-life* који представљају време за које *Penalty* треба да падне на половину почетне вредности. [2.23] предвиђа постојање различитих вредности за *half-life* за ситуације када се рута оглашава и када је рута повучена. Овако дефинисан механизам опадања вредности *Penalty* значи да ће крива опадања овог параметра бити сегментно линеарна (а не линеарна као што је приказано на слици 2.28), односно да ће за веће вредности *Penalty* брже опадати него за мање.

2.2.9. Мултипротоколарна проширења за BGP

Актуелна верзија BGP протокола која је настала средином 90-их година користи се и данас. Од тада се десио велики број промена у технологијама које чине или користе интернет (на пример појава IPv6, MPLS-а, мултикаста итд.). Да би се омогућило коришћење нових технологија и протокола, а IPv6 је ту најочигледнији пример, јер је то протокол за који не постоји подршка у BGP-4 протоколу, постојале су две могућности: да се поново, као 1.1.1983. када су у једном тренутку уведени TCP/IP протоколи изврши транзиција на неку нову верзију протокола која би имала све потребне измене, или да се направе модификације постојећих протокола којима би се омогућиле нове функционалности. Како је транзиција која се десила 1983. године била транзиција у оквиру само једне мреже – Арпанет, а потреба за поменутим променама се јавила у тренутку постојања најмање више хиљада аутономних система, са више десетина милиона активних кориника, са опремом различитих произвођача и старости и са све већим финансијским интересима које је интернет свакодневно омогућавао, оваква врста тренутне промене је носила превише ризика и више није била прихватљива или чак ни могућа. Додатна дискусија о ове две стратегије промена на интернету је дата у наредном поглављу.

Одабрана стратегија је била да се направе проширења основне верзије протокола која ће омогућити транспорт различитих информација кроз BGP сесије до свих дестинација на интернету и реализацију нових функционалности. Направљена проширења се називају

мултипротоколарна проширења BGP протокола (енг. *Multiprotocol Extensions to BGP-4* – MP-BGP) [2.24]. MPBGP омогућава пренос различитих информација (IPv6 рута, мултикаст рута, VPN рута итд.) кроз два нова атрибута:

- *Multiprotocol Reachable* NLRI којим се специфицирају руте које се оглашавају, њихов тип и зависне информације које су потребне да би се остварило рутирање других протокола. Тип руте означава да ли је у питању IPv6 рута, мултикаст рута, VPN рута или нешто друго (чиме је одређена и дужина руте), а зависне информације су додатне информације потребне за пуну реализацију функционалности – на пример, IPv6 захтева да и неки други параметри као што је NextHop атрибут буду такође IPv6 адресе, тако да је морало да се обезбеди да се и те информације упишу у овај атрибут.
- *Multiprotocol Unreachable* NLRI којим се повлаче претходно споменуте руте.

Неке примене коришћења мултипротоколарних екстензија за BGP ће бити детаљније описане у поглављима 3 и 5.

2.3. Литература

[2.1] Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, Stephen Wolff, Brief History of the Internet, Internet Society, 1997, https://www.internetsociety.org/wp-content/uploads/2017/09/ISOC-History-of-the-Internet_1997.pdf

[2.2] Sam Halabi, Danny McPherson, Internet Routing Architectures, Second Edition, Cisco Press, Second Edition August 23, 2000, ISBN: 1-57870-233-X

[2.3] CAIDA AS Rank, http://as-rank.caida.org/

[2.4] IP Traffic Exchange Policy for U.S. Interconnection, http://www.level3.com/en/legal/ip-traffic-exchange-policy/

[2.5] M. Winther, Tier 1 ISPs: What They Are and Why They Are Important, NTT Whitepaper, 2006, http://www.us.ntt.net/downloads/papers/IDC_Tier1_ISPs.pdf

[2.6] John Markoff, Internet Traffic Begins to Bypass the U.S., NY Times, August 2008. http://www.nytimes.com/2008/08/30/business/30pipes.html?pagewanted=all (приступљено 2.1.2018.)

[2.7] Gill P., Arlitt M., Li Z., Mahanti A. “The Flattening Internet Topology: Natural Evolution, Unsightly Barnacles or Contrived Collapse?”, Passive and Active Measurement Conference, 2008, Cleveland OH, USA

[2.8] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, A. Vahdat. 2013. B4: experience with a globally-deployed software defined wan. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM* (SIGCOMM '13). ACM, New York, NY, USA, 3-14. DOI: http://dx.doi.org/10.1145/2486001.2486019

[2.9] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, F. Jahanian. 2010. Internet inter-domain traffic. In *Proceedings of the ACM SIGCOMM 2010 conference* (SIGCOMM '10). ACM, New York, NY, USA, 75-86. DOI=http://dx.doi.org/10.1145/1851182.1851194

[2.10] D. Bach, Microsoft, Facebook and Telxius complete the highest-capacity subsea cable to cross the Atlantic 21 September, 2017, https://news.microsoft.com/features/microsoft-facebook-telxius-complete-highest-capacity-subsea-cable-cross-atlantic/ (приступљено 2.1.2018.)

[2.11] J. Hecht, Submarine Cable Goes for Record: 144,000 Gigabits From Hong Kong to L.A. in 1 Second, https://spectrum.ieee.org/telecom/internet/submarine-cable-goes-for-record-144000-gigabits-from-hong-kong-to-la-in-1-second (приступљено 11.1.2018.)

[2.12] Y. Bachar, Introducing “6-pack”: the first open hardware modular switch, https://code.facebook.com/posts/717010588413497/introducing-6-pack-the-first-open-hardware-modular-switch/ (приступљено 2.1.2018.)

[2.13] Y. Wong, The internet has been quietly rewired, and video is the reason why, https://qz.com/742474/how-streaming-video-changed-the-shape-of-the-internet/ (приступљено 2.1.2018.)

[2.14] Y. Rekhter,T. Li, S. Hares, A Border Gateway Protocol 4 (BGP-4), IETF RFC 4271, January 2006., https://tools.ietf.org/html/rfc4271

[2.15] How to Influence BGP Routes with Origin and MED Metrics, https://live.paloaltonetworks.com/t5/Learning-Articles/How-to-Influence-BGP-Routes-with-Origin-and-MED-Metrics/ta-p/54627 (приступљено 4.1.2018.)

[2.16] R. Chandra, P. Traina, T. Li, BGP Communities Attribute, IETF RFC 1997, August 1996., https://tools.ietf.org/html/rfc1997

[2.17] T. Bates, E. Chen, R. Chandra, BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP), IETF RFC 4456, April 2006., https://tools.ietf.org/html/rfc4456

[2.18] P. Traina, D. McPherson, J. Scudder, Autonomous System Confederations for BGP, IETF RFC 5065, August 2007, https://tools.ietf.org/html/rfc5065

[2.19] O. Nordström, C. Dovrolis, Beware of BGP attacks, Computer Communication Review (ACM SIGCOMM), Volume 34, Number 2, April 2004pp1-8

[2.20] T. Wan, P. C. van Oorschot, "Analysis of BGP prefix origins during Google's May 2005 outage," *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*, 2006, pp. 8 pp.-.doi: 10.1109/IPDPS.2006.1639679

[2.21] A. Toonk, BGP Stream and The Curious Case of AS12389, April 2017. https://bgpmon.net/bgpstream-and-the-curious-case-of-as12389/ (преузето 5.1.2018.)

[2.22] D.Godin, Russian-controlled telecom hijacks financial services’ Internet traffic, https://arstechnica.com/information-technology/2017/04/russian-controlled-telecom-hijacks-financial-services-internet-traffic/ (преузето 5.1.2018.)

[2.23] C. Villamizar, R. Chandra, R. Govindan, BGP Route Flap Damping, IETF RFC 2439, November 1998. https://tools.ietf.org/html/rfc2439

[2.24] T. Bates, R. Chandra, D. Katz, Y. Rekhter, Multiprotocol Extensions for BGP-4, IETF RFC 4760, January 2007, https://tools.ietf.org/html/rfc4760

3. Виртуелне приватне мреже

Архитектура, основни принципи рада и кључни протоколи данашњих рачунарских мрежа (IP и TCP) су заокружени и стандардизовани у првој половини осамдесетих година 20. века. У то време најважнија апликација која се користила посредством рачунарских мрежа је био мејл, који се покретао из неког програма из командне линије. Први оперативни системи са графичким корисничким интерфејсом за персоналне рачунаре су почели да се појављују на тржишту баш у време када је усвојен IPv4 RFC. Са таквим стањем развоја рачунарских система у време стандардизације данашњих интернет протокола, у тренутку пре настанка глобалног интернета било је тешко замислити које ће све примене добити рачунарске мреже и за које све потребе ће се користити, те сама архитектура мрежа и нека техничка решења у кључним протоколима поседују бројна касније откривена ограничења.

За било какву врсту финансијских трансакција и трговине преко мреже, које су данас уобичајене, нужно је да постоји подршка за заштиту података како не би дошло до злоупотреба и крађе. За пренос слике и звука (видеоконференције, телевизија на захтев и слично) потребно је да постоји подршка за обезбеђивање гарантованог протока од извора до дестинације како би квалитет услуге био задовољавајући. Мобилне мреже које су се развијале паралелно и у исто време са интернетом и мобилност корисничких уређаја који данас постоје захтевају да се на неки начин обезбеди и подршка за непрекидно комуницирање када се уређаји крећу или премештају унутар рачунарских мрежа. Све ове примене, које су данас подразумеване, нису подржане основним верзијама мрежних протокола, што ће у наставку текста и бити детаљније показано, и било је потребно да се постојећи протоколски стек и начин рада мрежа некако унапреди.

Истраживачи су на различите начине приступали овим проблемима. Док су неки заговарали тзв. *clean slate* приступ код кога се подразумева да би у једном тренутку морао да се уведе потпуно нов принцип рада мрежа „од нуле“, са потпуно новим протоколима који би заменили постојеће, други су били за еволутивни приступ – кроз константна мања унапређења постојећих механизама [3.1]. Како још од времена настанка рачунарских мрежа

интернет веома брзо расте и данас га свакодневно користе милијарде људи за најразноврсније примене, *clean slate* приступ се сматра за неприхватљив и ризичан јер би подразумевао отказ мреже у тренутку миграције на нови начин рада, са неизвесном поузданошћу у фази уходавања рада нових механизама и потенцијалним проблемима због скалирања механизама на глобалном нивоу који не могу да се предвиде лабораторијским тестирањем. Због тога је историја развоја рачунарских мрежа и интернета заправо историја бројних мањих еволутивних помака и унапређења, док су неки од кључних механизама који се и данас користе стари између 20 и 30 година.

Један од начина да се реше неки од претходно наведених изазова било је увођење различитих технологија виртуелних приватних мрежа и тунеловања пакета. Виртуелне приватне мреже (енг. *Virtual Private Network* – уобичајено и VPN) су мреже чија се топологија не састоји од скупа физичких уређаја и веза (те стога назив виртуелне) већ се реализују преко неке већ постојеће инфраструктуре (интернета или мреже неког пружаоца услуга - провајдера). Обично се реализују за потребе једне организације или групе корисника (те стога назив приватне), тако што се пакети који припадају кориснику виртуелне приватне мреже посебно означавају и логички одвајају од пакета других корисника, а могуће је и потпуно остваривање приватности комуникације кроз енкрипцију пакета у некој виртуелној приватној мрежи. У зависности од типа могу да се реализују преко интернета или преко мрежа пружалаца комуникационих услуга. Виртуелне приватне мреже технички најчешће подразумевају тунеловање пакета: додавање нових заглавља у пакет на основу којих се врши прослеђивање кроз мрежу (а не више на основу оригиналних IP адреса у пакету), на начин који заобилази класичан начин рутирања у IP мрежама, а како би се добиле нове функционалности мреже.

У новије време са порастом популарности технологија виртуелизације и услуга пружаних „у облаку“, развио се концепт виртуелних мрежних уређаја: виртуелних свичева који повезују више виртуелних машина на неком серверу, виртуелних рутера и мрежних баријера (енг. *firewall*) имплементираних на виртуелној машини неког сервера у дата центру. Овај концепт се зове и виртуелизација мрежних функција (енг. *Network Function Virtualization* – NFV) и њиме је омогућена једноставнија реализација неких мрежних функционалности без потребе за набавком физичких уређаја, као и пружање нових услуга у облаку. Ове технике мрежне виртуелизације нису тема овог поглавља.

3.1. Прослеђивање на основу лабеле – MPLS

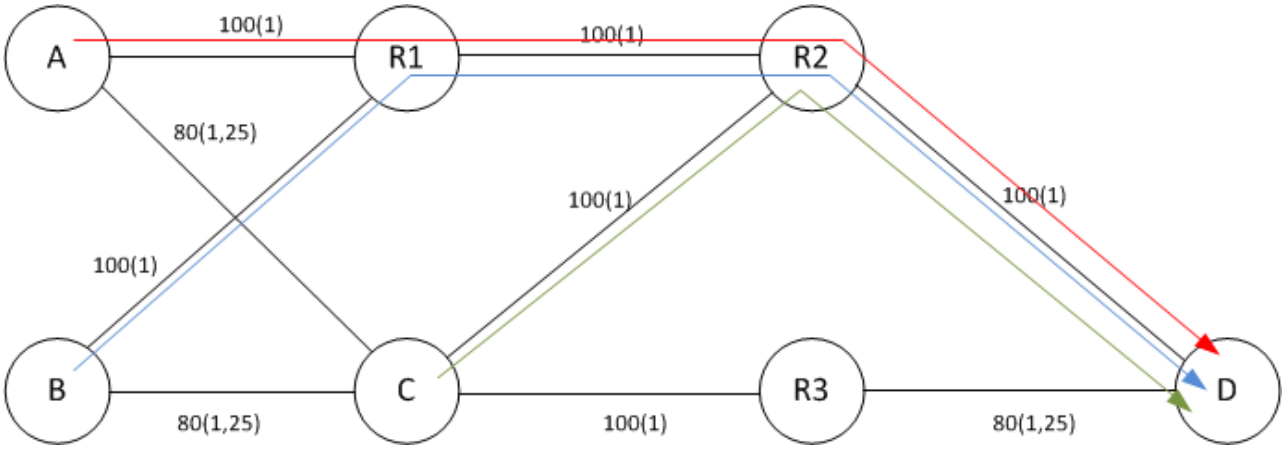
Пораст брзина веза између комуникационих уређаја је довео и до потребе да се убрза и поједностави обрада пакета приликом проласка кроз мрежне уређаје. Ово је даље довело до развоја потпуно новог начина прослеђивања пакета у мрежним уређајима: не више на основу IP адресе, већ на основу новог поља у пакетима – тзв. лабеле. Развој на технологији

прослеђивања пакета на основу лабела (енг. *MultiProtocol Label Switching* - MPLS) је завршен почетком двехиљадитих [3.2], а од половине прве деценије 21. века MPLS се наметнуо као кључна технологија за пружање услуга виртуелних приватних мрежа пословним корисницима. Поред убрзања процесирања пакета, MPLS технологија је уведена и како би се омогућило раздвајање саобраћаја различитих корисника преко неке дељене мрежне инфраструктуре, али и како би се решили проблеми који су се појавили у рачунарским мрежама попут неоптималног искоришћења ресурса мреже што је описано у наредном поглављу.

3.1.1. Проблеми класичних рачунарских мрежа

3.1.1.1. Неоптимално искоришћење ресурса мреже

Одлука о рутирању у рачунарским мрежама се врши упитом у табелу рутирања на основу дестинационе IP адресе пакета. Руте у табели рутирања одређују стандардни протоколи рутирања на основу тополошких карактеристика мреже (нпр. број рутера на путањи, капацитети веза и слично). Овакав начин рада у којем се врши оптимизација без узимања у обзир стварно расположивих ресурса у мрежи може да доведе до неоптималног искоришћења мреже. Мрежа са слике 3.1 илуструје овај проблем. У датој мрежи наведени су капацитети веза и метрика протокола рутирања која је дата у загради (претпостављено је да се метрика рачуна на начин као код OSPF протокола као реципрочна вредност капацитета помножена неком константом, у овом случају 100). Ако се посматра путања пакета од изворишта A, B и C ка дестинацији D, види се да путање са најмањом метриком воде у сва три случаја преко рутера R2, везом R2-D. Овако организовано рутирање може да доведе до загушења везе R2-D иако у мрежи постоје везе које су потпуно неискоришћене (везе A-C, B-C и C-R3-D).



Слика 3.1 Проблем неоптималног избора путања у мрежама у којима се рутирање врши на основу дестионационе адресе

Наведени проблем би могао да се реши када би:

1. метрика протокола рутирања узела у обзир и неке променљиве параметре веза (нпр. тренутно оптерећење/заузеће веза) или,
2. када би рутери могли да рутирају пакете на основу неке додатне информације (нпр. долазни интерфејс пакета, врста транспортног протокола или број порта транспортног протокола), чиме би се омогућило да се један део пакета од неког извора ка одређеној дестинацији пошаље једном, а други део другом путањом.

Први приступ решењу проблема постоји у неким мање популарним протоколима рутирања, какав је Cisco-специфичан протокол EIGRP који има могућност да у калкулацију метрике укључи и оптерећење појединачних веза. Међутим овај протокол се налази на рутерима само једног произвођача, а и ретко се примењује у пракси на овај начин, јер се у подразумеваној конфигурацији протокола параметар оптерећења веза не узима у обзир. Даља дискусија о последицама динамичког одређивања метрике на основу променљивих параметара веза ће бити дата у поглављу 3.1.8.3. Други приступ решењу проблема је у модерним рутерима могуће извести постављањем посебних полиса на основу који се прослеђују пакети (тзв. *Policy Based Routing*) којима је могуће дефинисати правила за рутирање пакета која нису одређена само дестинационом адресом и заобићи табелу рутирања у одлучивању. Проблем рутирања на основу полиса је тај што се процесирање пакета у том случају увек врши софтверски, у оквиру прекидних рутина, без могућности коришћења специјализованих чипова који имају велику проточност пакета и могу да омогуће велике капацитете веза. Такође, рутирање на основу полиса се изводи независним конфигурисањем појединачних рутера, без могућности аутоматског обавештвања суседа о догађајима у мрежи и промени топологије. Због тога имплементација рутирања на основу полиса може да доведе до петљи у рутирању целог или дела саобраћаја и ретко се препоручује. Из ових разлога је једна од мотивација увођења MPLS технологије била да се омогући аутоматизовано и брзо прослеђивање пакета на основу неког другог критеријума, а не на основу дестинационе адресе. Ово је на крају реализовано механизмима MPLS *traffic engineering* који су детаљно описани у поглављу 3.1.8.

3.1.1.2. Комплексно процесирање пакета

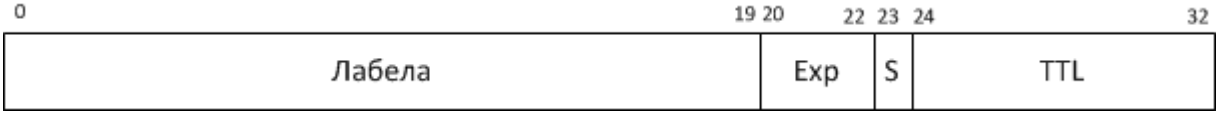
У време када су настали етернет, TCP и IP протоколи везе су биле релативно непоуздане са већим процентом губитка пакета него данас када се оптичке везе, отпорне на сметње уобичајено користе и унутар локалних мрежа. То је довело до тога да су и у етернет[16] и у IP протокол уведени механизми провере грешака (*Ethernet Frame check sequence* и IP *header checksum*), што уводи у једном делу пакета редундантну проверу. Такође, процесирање сваког појединачног пакета на рутерима подразумева поред ових провера и декрементацију и верификацију TTL и формирање новог етернет заглавља што чини процесирање IP пакета у рутерима комплексним и утиче на брзину прослеђивања и капацитете уређаја. Друга мотивација за увођење MPLS је било поједностављивање процесирања пакета приликом

16 Раније су се често користили на серијским везама други протоколи слоја везе попут PPP или HDLC, али данас се етернет користи готово универално, те се овде сматра да је то практично једини протокол слоја везе у употреби данас.

проласка кроз рутере како би се убрзало прослеђивање, али и смањиле величине табела на основу којих се врши прослеђивање пакета.

3.1.2. MPLS лабела

Кључни елемент MPLS технологије је MPLS заглавље које се додаје у пакет између заглавља протокола слоја везе (данас готово искључиво Етернет) и IP протокола. У мрежама у којима је активиран MPLS, рутирање се врши на основу овог, а не више IP заглавља. MPLS заглавље чија је дужина свега 32 бита је приказано на слици 3.2.



Слика 3.2 Елементи MPLS заглавља

Кључни елементи заглавља су MPLS лабела дужине 20 бита, 3 експериментална бита која се користе за дефинисање 8 нивоа приоритета пакета и организацију редова за чекање у рутерима (који су детаљно објашњени у глави 6), један бит „*Bottom of stack*“ - S и поље TTL. MPLS технологија дозвољава додавање више лабела у пакет, једна до друге. „*Bottom of stack*“ бит има вредност 0 ако иза дате лабеле постоји још једна лабела, а вредност 1, ако је у питању последња (унутрашња) лабела (она до заглавља IP протокола). У тренутку када се додаје MPLS заглавље у пакет, вредност TTL се преписује из IP заглавља и убацује у MPLS заглавље, како би се приликом проласка пакета кроз рутере даље декрементирала вредност TTL у MPLS заглавље. Пакети се рутирају на основу MPLS лабеле, којих може да буде укупно 2^{20}. Првих 15 лабела су резервисане, док остале могу да се слободно користе. Као што може да се види једно од поједностављивања рутирања је постигнуто тиме што приликом прослеђивања пакета више нема провере грешке у преносу IP заглавља. Такође, MPLS заглавље не садржи поље којим је означено који је протокол енкапсулиран у њега. Да би парсирање пакета могло да се правилно изведе, ова информација је остављена протоколу слоја везе, па тако етернет заглавље у пољу *EtherType* има вредности 0x8847 и 0x8848 за уникаст и мултикаст пакете респективно.

За све пакете који имају исту лабелу се каже да припадају истој класи прослеђивања (енг. *Forwarding Equivalence Class* – FEC). У зависности од начина како се пакети разврставају у FEC класе, зависи и ефекат рутирања који ће бити постигнут. У наставку текста ће бити обрађена три случаја: *Frame mode* MPLS, MPLS VPN и MPLS *Traffic engineering*.

У називу MPLS постоји реч *multiprotocol*, којом је истакнуто да је лабела могла да су уметне између произвољних протокола слоја везе (нпр. Frame Relay, ATM,…) и IP протокола. Међутим, данас се искристалисао етернет као практично једини протокол, те се MPLS среће готово искључиво у оваквом окружењу.

3.1.3. Терминологија

Рутери у мрежама у којима се користи MPLS могу да буду:

- ивични: они који имају део суседа са којима размењују пакете путем IP протокола и део суседа са којима размењују пакете у којима су лабеле. Они додају MPLS лабелу класичном IP пакету када улази у MPLS домен или је скидају када излази, дакле налазе се на ивици MPLS домена. Најчешће се означавају са PE (*Provider Edge*).
- унутрашњи: они рутери који са свим својим суседима размењују пакете у којима су лабеле. Означавају се са P (*Provider*).
- кориснички – они рутери који су изван MPLS домена. Могу да буду ивични - CE (*Customer Edge*) уколико могу да размењују пакете са PE рутерима или C (*Customer*), уколико су у унутрашњости корисничке мреже.

Сви P и PE рутери се још називају и LSR (*Label Switching Router*) – они који прослеђују пакете на основу лабела.

3.1.4. Frame mode MPLS

Frame mode MPLS је режим рада код кога се MPLS лабеле додељују на основу руте у табели рутирања рутера. Принцип и редослед доделе лабела је следећи:

1. У MPLS мрежи постоји класичан интерни протокол рутирања и сви рутери размењују руте као у класичној мрежи.
2. За сваку мрежу која се налази у табели рутирања неког рутера, сваки рутер аутономно и произвољно додељује посебну лабелу.
3. Након што додели лабеле рутама из своје табеле рутирања, рутер шаље свим својим суседима парове (рута, лабела) које је доделио.
4. Када рутери размене парове (рута, лабела) са свим суседима могу да донесу одлуку о томе које ће се лабеле користити у пакетима на сваком од мрежних сегмената. Важе следећи принципи:
   а) Лабела има локалан карактер: када пакет пролази кроз MPLS мрежу дуж MPLS путање, лабела се мења на сваком мрежном сегменту.
   б) На одређеном мрежном сегменту ће се користити она лабела која је добијена од суседа који је следећи на путањи пакета ка дестинацији (тзв. низводни, енг. *downstream* рутер).

Овај процес ће бити детаљно показан у поглављу 3.1.4.3.

3.1.4.1. Протокол за размену лабела - LDP

Протокол за размену парова (рута, лабела) између рутера у *Frame Mode* MPLS је LDP (*Label Distribution Protocol*) [3.3]. LDP користи TCP протокол на транспортном слоју. Након успостављања TCP сесије, успоставља се LDP суседски однос између рутера, након чега почиње размена парова (лабела, рута) које су рутери доделили. LDP има могућност конфигурације различитих стратегија по неким аспектима управљања лабелама:

- Начин слања лабела: Рутер може да пошаље суседима парове (лабела, рута) по захтеву од суседа (енг. *On demand*) или без захтева, самостално (енг. *Unsolicited*). Механизмом слања парова (лабела, рута) без захтева се врши бржа пропагација информација него када се користи механизам по захтеву.
- Начин пропагације лабела кроз мрежу: Рутер може да пошаље парове (лабела, рута) само ако је добио одговарајући пар (лабела, рута) за дату руту од низводног рутера за дату руту (енг. *Ordered control*) или може да пошаље парове (лабела, рута) без обзира на то да ли је добио пар од низводног рутера (енг. *Independent control*). У првом случају се обезбеђује то да ако је нека лабела стигла на одређени рутер да сви рутери од тог рутера до излазне тачке из MPLS мреже имају парове (лабела, рута) и да пакет сигурно може да буде прослеђен коришћењем MPLS лабела. У другом случају нема те гаранције, али је пропагација парове (лабела, рута) кроз мрежу бржа.
- Начин чувања лабела: Рутер може да чува све лабеле за дату руту које је добио од свих суседа (енг. *Liberal retention*) или само оне лабеле које су добијене од низводних рутера и које се користе на датом мрежном сегменту у пакету (енг. *Conservative retention*). У првом случају се троши више меморије рутера, али се приликом промене топологије брже реагује и врши промена топологије него у другом случају.

Време конвергенције мреже је време од тренутка када је дошло до промене у топологији (услед нпр. неког квара), до тренутка када сви уређаји у мрежи имају конзистентне информације о промењеној топологији. Код *Frame Mode* MPLS време конвергенције укључује време конвергенције протокола рутирања и време које је потребно паровима (лабела, рута) да пропагирају до свих рутера. Дакле, време конвергенције мреже која користи *Frame Mode* MPLS је веће од времена конвергенције класичне рачунарске мреже и то је непожељна особина. Да би се овај други део времена конвергенције који је потребан паровима (лабела, рута) да пропагирају до свих рутера максимално скратио, најчешћи режим рада LDP протокола је *Unsolicited, Independent control* и *Liberal Retention*.

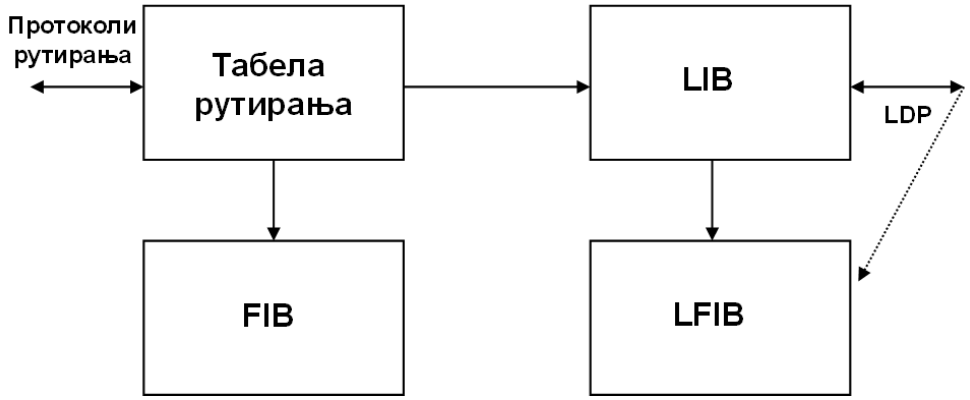
3.1.4.2. Структуре и табеле MPLS рутера

Модерни рутери који раде само са IP пакетима имају две основне структуре потребне за размену и организацију контролних информација и прослеђивање пакета:

1. табелу рутирања која се попуњава информацијама добијеним конфигурацијом рутера и од суседа путем протокола рутирања. У модерним великим рутерским шасијама, табела рутирања се налази у RAM меморији на тзв. *routing and switching engine*-у

(RSE), делу уређаја који је најчешће реализован као сервер PC архитектуре односно сервер смештен у шасију рутера, са оперативним системом рутера на себи.

2. FIB (енг. *Forwarding Information Base*) табелу која се налази у специјализованим чиповима на линијским картицама са интерфејсима. У овој табели се налазе све информације које су потребне да се IP пакет у једном пролазу кроз чип рутира на одговарајући излазни интерфејс. Те информације укључују податке о излазним интерфејсима за дату дестинацију, али и податке о новом етернет заглављу које је потребно формирати за пакет на излазном интерфејсу. Ове информације се добијају из табеле рутирања и ARP табеле. Модерне велике рутерске шасије које имају више линијских картица имају више инстанци FIB које се налазе у чиповима на свакој од картица.



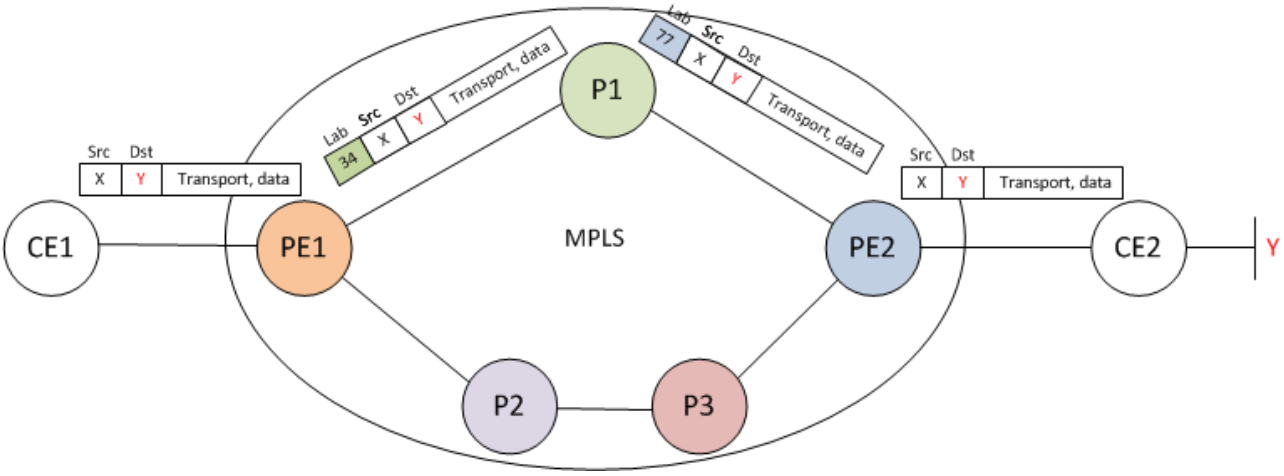
Слика 3.3 Основне табеле MPLS рутера

Јасно је да MPLS рутери који додељују рутама лабеле имају додатне табеле како би све информације могле да буду организоване. Додатно у односу на претходно наведене табеле, MPLS рутери имају (Слика 3.3):

1. LIB (енг. *Label Information Base*) табелу која се налази у RAM меморији RSE и која се попуњава на основу информација о локално додељеним лабелама рутама које су у табели рутирања датог рутера и на основу информација добијеним од LDP суседа о паровима (лабела, рута) које су они доделили.
2. LFIB (енг. *Label Forwarding Information Base*) табелу која представља аналогију FIB табеле само за пакете са лабелама – то је структура у специјализованим чиповима за прослеђивање пакета у којима су садржане све оне информације којима се омогућава прослеђивање пакета са лабелама (излазни интерфејс, излазна лабела, подаци о етернет заглављу или излазно IP заглавље ако пакет излази из MPLS мреже).

3.1.4.3. Пример успостављања путање код Frame Mode MPLS

На слици 3.4 дат је пример једне MPLS мреже која се састоји од пет MPLS рутера (два ивична: PE1 и PE2 и три унутрашња: P1, P2 и P3). Посматраће се начин на који се додељују лабеле пакетима када иду ка дестинацији Y, као и садржаји основних табела.



Слика 3.4 Начин енкапсулације пакета приликом проласка кроз MPLS мрежу

Како је наведено у поглављу 3.1.4, први корак је размена информација о рутама путем неког од интерних протокола рутирања. Ако се претпостави да су све везе истих капацитета и да се користи OSPF протокол, делови табела рутирања који указују на дестинацију Y су дати на слици 3.5. Оптимална путања пакета од CE1 ка Y води преко рутера P1 и PE2.

PE1 табела рутирања

Дест.	next hop
Y	P1

P1 табела рутирања

Дест.	next hop
Y	PE2

PE2 табела рутирања

Дест.	next hop
Y	CE2

P2 табела рутирања

Дест.	next hop
Y	P3

P3 табела рутирања

Дест.	next hop
Y	PE2

Слика 3.5 Поједностављени приказ дела табела рутирања који се односе на мрежу Y

Након што се рута нађе у табели рутирања, сваки рутер јој независно од осталих рутера додељује произвољан број лабеле. Ове лабеле су у LIB табелама означене као „*Local*“, а након тога рутери путем LDP протокола шаљу својим суседима мапирања која су направили. На крају на свим рутерима ће бити попуњене LIB табеле за дестинацију Y на начин који је показан на слици 3.6.

PE1 LIB табела

Дест.	Лабела	LSR
Y	23	Local
	34	P1
	99	P2

P1 LIB табела

Дест.	Лабела	LSR
Y	34	Local
	23	PE1
	77	PE2

PE2 LIB табела

Дест.	Лабела	LSR
Y	77	Local
	34	P1
	19	P3

P2 LIB табела

Дест.	Лабела	LSR
Y	99	Local
	23	PE1
	19	P3

P3 LIB табела

Дест.	Лабела	LSR
Y	19	Local
	99	P2
	77	PE2

Слика 3.6 Поједностављени приказ дела LIB табела који се односе на мрежу Y

Последњи корак је попуњавање LFIB и FIB табела како би рутери могли да прослеђују пакете. У ове табеле улазе само оне информације из LIB табела на основу којих ће се извршити процесирање и рутирање пакета. Рутер PE1 ће имати попуњен улаз у FIB табелу зато што је он улазни рутер за MPLS пакете и на њему ће класични IP пакети добити лабелу

и то ону лабелу која је добијена од „низводног рутера“, а то је у овом случају лабела 34 коју је доделио рутер P1. Остали рутери ће имати попуњене улазе у LFIB табелама зато што у случају дестинације Y треба да процесирају пакете који имају лабелу када долазе до њих. У LFIB табелама свих рутера се види број долазне лабеле и акција. Акција је у случају рутера P1, P2 и P3 промена лабела и наведена је лабела коју пакети треба да имају на наредној вези. Једино је у случају рутера PE1 акција POP, што значи да треба скинути лабелу и пакет проследити као класичан IP пакет ка рутеру CE1.

PE1 FIB табела

Дест.	next hop	Лабела
Y	P1	34

P1 LFIB табела

Лабела	Акција	next hop
34	77	PE2

PE2 LFIB табела

Лабела	Акција	next hop
77	POP	CE2

P2 LFIB табела

Лабела	Акција	next hop
99	19	P3

P3 LFIB табела

Лабела	Акција	next hop
19	77	PE2

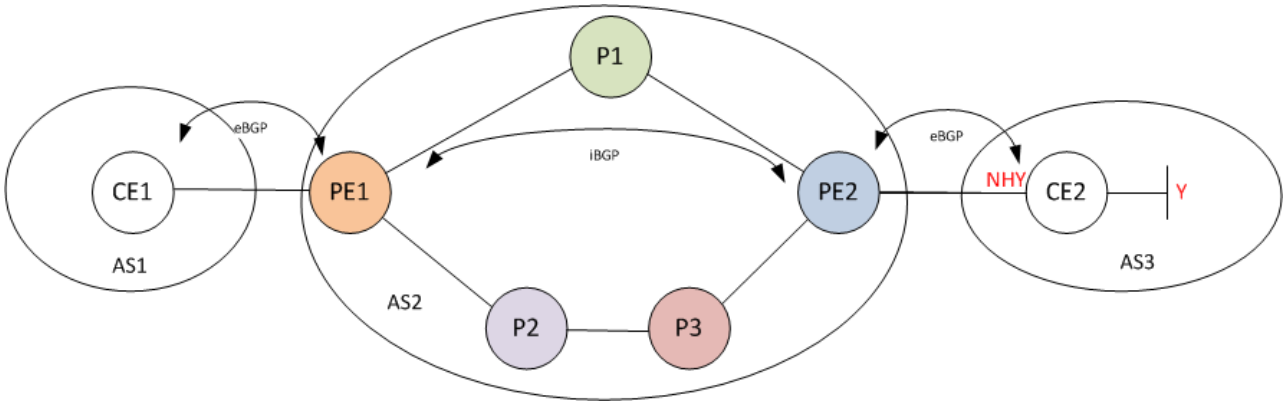
Слика 3.7 Поједностављени приказ дела LFIB табела који се односи на мрежу Y

На слици 3.4 показане су лабеле које ће пакет имати када пролази путањом PE1-P1-PE2. Ово је пример када се у мрежи не користи механизам PHP који је објашњен у поглављу 3.1.4.5. Путања пакета од уласка у MPLS мрежу до изласка из ње се зове LSP (енг. *Label Switched Path*). На овом LSP се користе лабеле 34 и 77. Такође, из претходног излагања је јасно да за двосмерну комуникацију између два уређаја морају да постоје два независна LSP са различитим лабелама у пакетима који иду по истом мрежном сегменту у супротним смеровима, формираним у оба случаја на основу дестинационих адреса.

3.1.4.4. Frame mode MPLS u BGP

Када се у табели рутирања налазе руте које су добијене из BGP протокола, постоји један изузетак у односу на правила доделе лабела дата у поглављу 3.1.4: руте које су добијене из BGP протокола неће имати све различите лабеле како је наведено у тачки 2, већ ће све руте добијене из BGP имати исту лабелу као рута ка њиховим *NextHop* адресама. Једна оваква ситуација приказана је на слици 3.8.

На овој слици MPLS мрежа је уједно и посебан аутономни систем - AS2. Између PE и CE рутера је успостављен екстерни BGP, а између PE рутера унутар аутономног система 2 је успостављен интерни BGP. Као што је показано у глави 2, уобичајено је да се руте мрежних сегмената који су на граници између аутономних система убаце у интерни протокол рутирања аутономних система, како се не би дошло у ситуацију да на рутерима унутар аутономног сиситема за руте добијене путем BGP не постоји рута ка *NextHop* мрежи, чиме оне не би могле да уђу у табелу рутирања.



Слика 3.8 Начин рада BGP протокола у MPLS мрежама

Према правилима додељивања *NextHop* атрибута за мрежу Y из аутономног система 3 овај атрибут ће бити адреса NHY на рутеру CE2. Узимајући све ово у обзир, табеле рутирања свих рутера унутар аутономног система 2 ће бити као на слици 3.9.

PE1 табела рутирања

Дест.	next hop
NHY	P1
Y	NHY

P1 табела рутирања

Дест.	next hop
NHY	PE2

PE2 табела рутирања

Дест.	next hop
NHY	CE2
Y	NHY

P2 табела рутирања

Дест.	next hop
NHY	P3

P3 табела рутирања

Дест.	next hop
NHY	PE2

Слика 3.9 Поједностављени приказ дела табела рутирања који се односе на мреже Y и NHY

Разлика у односу на пример из претходног поглавља је ту томе што ће рутери PE2 и PE1 имати у табели рутирања и руте ка мрежи Y које су добили путем екстерног, односно интерног BGP-а, док ће P рутери имати у табелама рутирања само руте ка мрежи NHY коју су добили путем интерног протокола рутирања (ка P рутерима није успостављен BGP протокол). У складу са наведеним на почетку овог поглавља, у LIB табелама PE рутера ће мрежа Y имати исту локалну лабелу која је додељена мрежи на којој се налази адреса NHY, док ће у LIB табелама P рутера бити само лабеле за мрежу NHY коју имају у табелама рутирања, што је приказано на слици 3.10.

PE1 LIB табела

Дест.	Лабела	LSR
NHY	23	Local
	34	P1
	99	P2
Y	23	Local

PE2 LIB табела

Дест.	Лабела	LSR
NHY	77	Local
	34	P1
	19	P3
Y	77	Local

P1 LIB табела

Дест.	Лабела	LSR
NHY	34	Local
	23	PE1
	77	PE2

P2 LIB табела

Дест.	Лабела	LSR
NHY	99	Local
	23	PE1
	19	P3

P3 LIB табела

Дест.	Лабела	LSR
NHY	19	Local
	99	P2
	77	PE2

Слика 3.10 Поједностављени приказ дела LIB табела који се односе на мреже Y и NHY

Из овога могу да се пребаце подаци у FIB и LFIB табеле свих рутера и оне ће изгледати као на слици 3.11.

PE1 FIB табела

Дест.	next hop	Лабела
Y	P1	34
NHY	P1	34

P1 LFIB табела

Лабела	Акција	next hop
34	77	PE2

PE2 LFIB табела

Лабела	Акција	next hop
77	POP	CE2

P2 LFIB табела

Лабела	Акција	next hop
99	19	P3

P3 LFIB табела

Лабела	Акција	next hop
19	77	PE2

Слика 3.11 Поједностављени приказ дела LIB табела који се односе на мреже Y и NHY

Из свега претходно наведеног може да се види неколико важних позитивних последица овог посебног начина доделе лабела рутама добијеним из BGP протокола:

- пакети који долазе из AS1 и иду ка дестинацији Y на рутеру PE1 добијају исту лабелу као да су у питању пакети ка NHY.
- Тако означени пакети ка дестинацији Y могу да се рутирају кроз MPLS мрежу на основу лабела иако P рутери немају у својим табелама рутирања руте ка мрежи Y, на исти начин како се рутирају пакети ка NHY. Овиме се значајно смањује величина табела рутирања на P рутерима и она уместо да буде величине пуне интернет табеле рутирања постаје сразмерна броју интерних рута датог аутономног система увећано за број BGP рутера (*NextHop* адреса) суседних датом аутономном систему. Самим тим чипови који врше прослеђивање пакета на P рутерима могу да буду са мањим табелама, једноставнији и краћег времена претраживања табела и прослеђивања пакета.
- Број лабела који је потребан једном аутономном систему је сразмеран броју интерних рута датог аутономног система увећано за број BGP рутера суседних датом аутономном систему. Захваљујући оваквом начину доделе лабела, никада није било планирано да све руте пуне интернет табеле рутирања добију посебне лабеле те је 20 бита предвиђених за MPLS лабелу је довољно за садашње и будуће потребе.
- Број iBGP сесија у датом аутономном систему је мањи него код класичних аутономних система заснованих на IP и BGP протоколима. У аутономним системима у којима се користи MPLS је довољно да постоји потпун граф iBGP веза између свих PE рутера.

3.1.4.5. Растерећење ивичних рутера - PHP

У MPLS мрежама највеће оптерећење је на ивичним рутерима који имају задатак да додају или да скидају лабелу приликом уласка односно изласка пакета из MPLS домена. Приликом изласка пакета из MPLS мреже, ивични рутер мора да уради следеће:

1. да прими MPLS пакет,
2. да на основу вредности лабеле коју је примио закључи да је потребно скинути лабелу,

3. да изврши рутирање на основу IP заглавља након што је скинута лабела.

Кораци 2 и 3 значе да је потребно да се донесу две независне одлуке о томе шта се ради са датим пакетом. Док кораци 2 и 3 могу у неким имплементацијама да се реализују у једном пролазу кроз табелу, у ситуацијама када се користи стек лабела, број одлука се повећава и тиме процесирање пакета успорава. Да би се ивични рутери растеретили вишеструког упита у табеле, реализован је механизам PHP (енг. *Penultimate Hop Popping*) код којег се (спољашња) лабела скида на претпоследњем рутеру. Тиме је последњи рутер растерећен бар једног процесирања лабеле.

PHP механизам се реализује тако што онај рутер који је излазни из MPLS мреже за дату дестинацију уместо пара (лабела, рута) својим суседима шаље пар (3, рута). Специјална вредност лабеле 3 има назив *implicit-null*[17] и означава да је у мрежи за дате дестинације активиран механизам PHP, односно тиме излазни рутер за дате пакете сигнализира претпоследњем рутеру да скине спољашње лабеле из пакета који припадају LSP коме је он излазни. Практични примери мрежа у којима је активиран механизам PHP су дати у поглављу 7.5 за *Frame Mode* MPLS и у поглављу 7.6 за MPLS L3VPN.

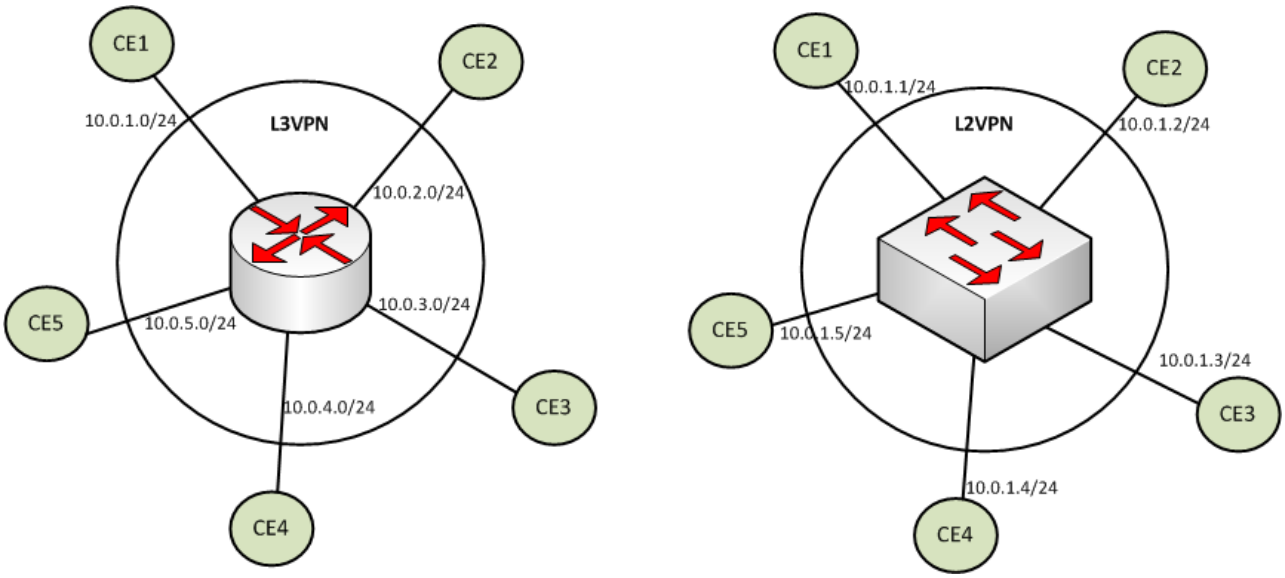
3.1.5. MPLS виртуелне приватне мреже

Један од кључних циљева увођења прослеђивања пакета на основу MPLS лабела је стварање могућности да се пакети различитих корисника мреже логички раздвоје и да може да се створи окружење које из перспективе корисника мреже изгледа као да је направљена јединствена комуникациона инфраструктура додељена само том кориснику, раздвојена од инфраструктура других корисника. За ту сврху је осмишљена и реализована технологија виртуелних приватних мрежа (VPN) у MPLS технологији. MPLS VPN су мреже које реализују пружаоци комуникационих услуга својим корисницима. Један од најчешћих примера коришћења MPLS VPN су корпоративне интранет мреже дистрибуираних организација (на пример пословна рачунарска мрежа неке банке која повезује све њене филијале и банкомате), или везе између приватних мрежа различитих организација које имају потребу да размењују податке посебним везама (екстранет тип виртуелне приватне мреже).

У зависности од начина на који се остварује веза корисничких уређаја у виртуелној приватној мрежи постоје две врсте MPLS VPN: L3VPN и L2VPN. Код L3VPN се MPLS мрежа понаша као виртуелни рутер на који су повезани кориснички CE рутери. Сваки CE рутер размењује руте са рутером у MPLS мрежи и при томе свака мрежа којом се CE рутер повезује са PE рутером је у посебном мрежном сегменту. Такође, кроз овакву мрежу не пролазе *broadcast* пакети. Код L2VPN се MPLS мрежа понаша као виртуелни свич на који су повезани CE уређаји. Сви CE рутери се налазе у истом мрежном сегменту и једном *broadcast*

17 Постоји могућност да се у овом механизму користи и тзв. e*xplicit-null* лабела којом последњи рутер јавља претпоследњем да у пакету остави лабелу, али са вредношћу 0. Овим се постиже то да се сачувају вредности осталих поља у лабели на последњој вези у MPLS мрежи, пре свега EXP бита који служе за одређивање приоритета пакета и могу да се користе за остваривање квалитета сервиса.

домену и имају адресе које припадају истој мрежи. Ово је показано на слици 3.12 где се види пример адресирања мрежних сегмената између CE рутера и MPLS мреже код L3VPN и адресирање CE уређаја у случају L2VPN. Избор врсте виртуелне приватне мреже зависи од потреба корисника.



Слика 3.12 L3 и L2 виртуелне приватне мреже – разлика у начину повезивања CE уређаја

Без обзира на врсту, MPLS VPN не обезбеђују приватност корисничког саобраћаја. Пакети се не енкриптују, а приватност пакета зависи од добре воље (или уговорне обавезе) пружаоца услуга да не открије садржај комуникације. Ако корисник жели, може сам додатно да заштити своје пакете.

3.1.6. MPLS L3VPN

Као што је претходно наведено, MPLS L3VPN се понашају као виртуелни рутери и обезбеђују транспорт корисничких рута између CE рутера. При томе је дозвољено да различити корисници мреже могу да користе преклопљене адресне опсеге. Које су последице ове две особине ће бити детаљније објашњену у наставку текста.

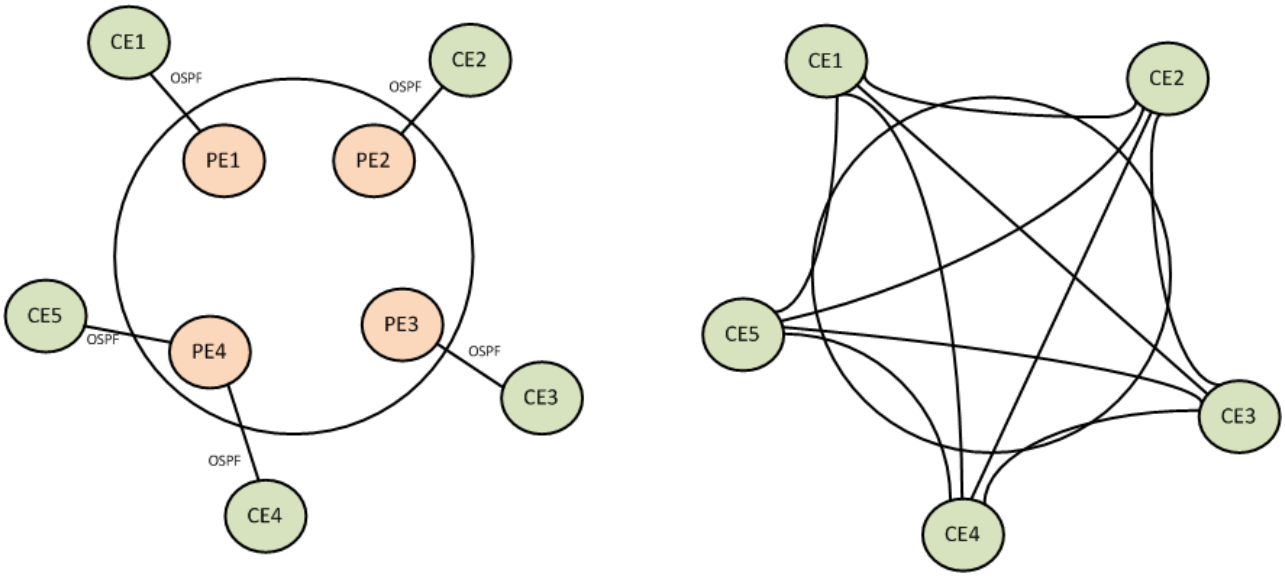
3.1.6.1. Пренос корисничких рута

Постоје две стратегије преноса рута корисника у виртуелним приватним мрежама:

- да у том преносу учествује пружалац услуга или
- да сам корисник обезбеђује пренос рута.

На слици 3.13 су приказана ова два случаја. У првом, кориснички рутери (CE) успостављају суседски однос неким протоколом рутирања са уређајима пружаоца услуга (PE). Све руте са неке од локација корисника се прослеђују PE рутеру, а онда мрежа пружаоца услуга

обезбеђује транспорт тих рута између PE рутера и пренос до свих осталих CE рутера. Овакав начин организације рутирања се користи код MPLS VPN, а модел виртуелних приватних мрежа *peer-to-peer*, јер CE и PE рутери успостављају суседски однос [3.4].



Слика 3.13 Peer-to-peer и overlay виртуелне приватне мреже

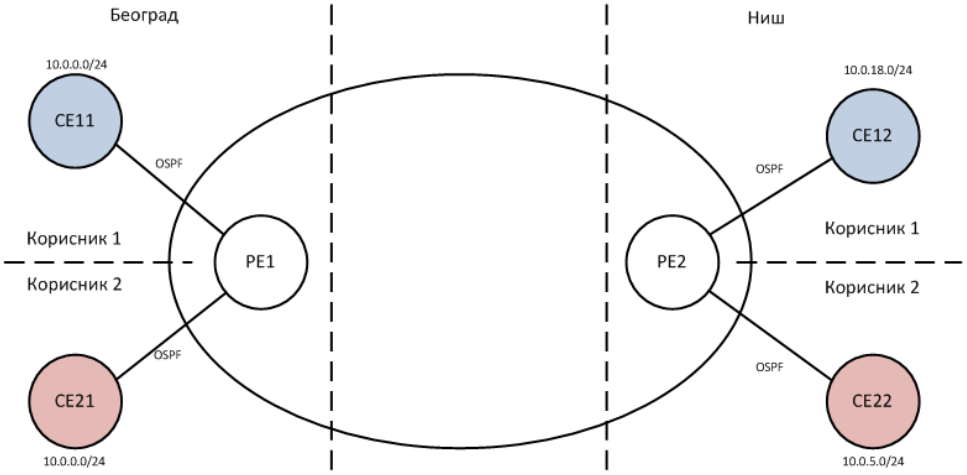
У другом случају корисник виртуелне приватне мреже сам формира мрежу креирајући мрежне тунеле између својих локација. Топологија виртуелне приватне мреже може да буде звездаста (енг. *hub and spoke*) када се формирају тунели од свих удаљених локација према једној централној тачки мреже или потпун граф (енг. *full mesh*), што је приказано на слици 3.13 или нека топологија између ове две. Каква год да је топологија, да би се омогућила комуникација између рачунара који се налазе на CE локацијама, потребно је да оне размене руте ка мрежама које су на тим локацијама и то се ради тако што корисник сам успоставља рутирање кроз успостављене тунеле. Овакав начин организације рутирања се користи код нпр. IPsec VPN, а модел виртуелних приватних мрежа *overlay*.

Кључна предност *peer-to-peer* модела је скалабилност конфигурације рутирања. У случају додавања нове локације у виртуелну приватну мрежу, код *peer-to-peer* модела потребно је прилагодити конфигурације само на одговарајућем CE рутеру и на PE рутеру на који је повезан. Са друге стране, у *overlay* моделу додавање нове локације значи конфигурацију новог CE уређаја, али и свих оних са којим је потребно да размењује руте, што у случају виртуелне приватне мреже са потпуним графом тунела између CE локација значи реконфигурацију свих CE рутера.

3.1.6.2. Преклопљени адресни опсези

На слици 3.14 је приказан пример преклопљених адресних опсега код два различита корисника виртуелних приватних мрежа код једног пружаоца услуга. Оба корисника имају потребу да повежу своје инфраструктуре у Београду и Нишу у јединствену мрежу и при томе

на локацијама у Београду које су повезане на исти рутер PE1 пружаоца услуга имају исте адресне опсеге: 10.0.0.0/24.



Слика 3.14 Пример виртуелних приватних мрежа које имају преклопљене адресне опсеге на локацији Београд

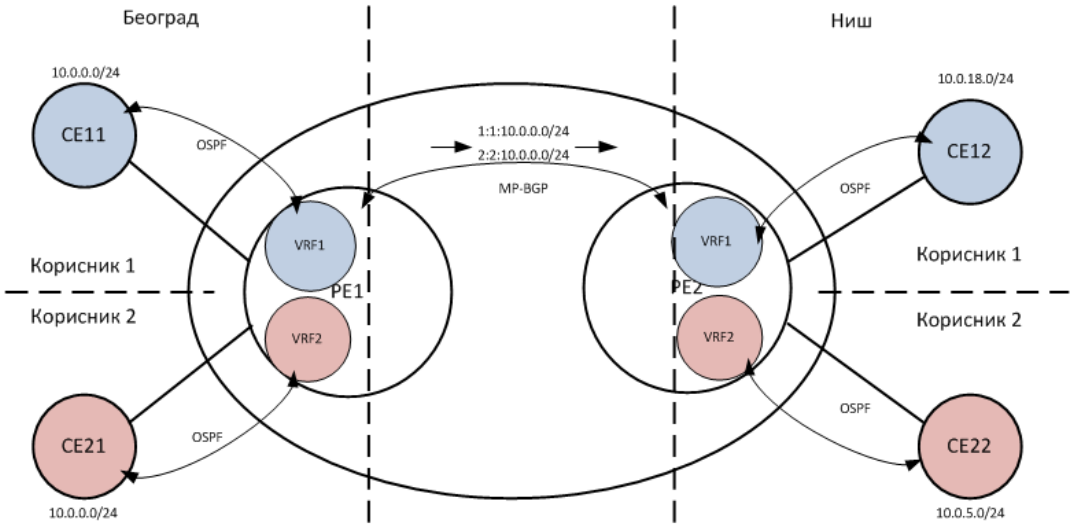
У класичним IP мрежама, рутер PE1 би у оваквој конфигурацији добио две руте ка мрежи 10.0.0.0/24 од рутера CE11 и CE21. У зависности од метрике протокола рутирања на PE-CE везама PE1 би или све пакете намењене дестинацијама у мрежи 10.0.0.0/24 рутирао ка једном од два CE рутера (чије је метрика нижа), чиме би мрежа једног од корисника била недоступна или би у случају истих метрика балансирао саобраћај према обе локације, чиме би било немогуће успостављање био какве мрежне сесије на било којој од локација корисника. MPLS VPN технологија омогућава да и у оваквим конфигурацијама са преклопљеним адресним опсезима сваки од корисника има потпуно функционално рутирање ка свим својим уређајима, што повећава флексибилност организације корисничких мрежа.

3.1.6.3. MPLS L3VPN контролна раван – организација рутирања

Да би се решио претходно наведени проблем исправног рутирања у ситуацијама када постоје преклопљени адресни опсези, MPLS VPN доноси неколико нових решења. Прво решење које је уведено је то да на PE рутерима постоји механизам за раздвајање рута које долазе од различитих корисника. То је реализовано тако што се за сваког корисника на PE рутеру формира посебна виртуелна инстанца за рутирање и прослеђивање (енг. *Virtual Routing and Forwarding* – VRF) – слика 3.15. VRF представља посебну структуру у меморији рутера, посебну табелу рутирања, у коју могу да уђу руте само једног од корисника и из које се руте без експлицитног захтева и одобрења не размењују са рутама из других VRF инстанци или са главном табелом рутирања PE рутера. Овим је обезбеђено то да се не мешају у истој табели рутирања руте различитих корисника.

CE рутери се повезују на неки од интерфејса рутера који се додељује одређеној VRF инстанци и успостављају рутирање са том VRF инстанцом, а не са PE рутером глобално. Пошто су CE рутери повезани на различите PE рутере у једној виртуелној приватној мрежи,

да би се руте у потпуности преносиле између CE рутера, потребно је да се руте из VRF инстанци транспортују између одговарајућих PE рутера. За транспорт корисничких рута између PE рутера се користи мултипротоколарни BGP, у који се и из којег се руте из VRF инстанци редистрибуирају.



Слика 3.15 Размена рута код MPLS L3VPN

Да би у примеру са слике рутер PE2 могао да разликује руте које долазе од рутера PE1 од две различите VRF инстанце и проследи их у тачно одговарајуће своје VRF инстанце, потребно је да се руте обележе на неки начин како би се означила њихова припрадност одговарајућој инстанци. Ово се ради помоћу тзв. *route distinguisher*-а (RD), који су 64-битне вредности додате свакој рути [3.5]. IPv4 рута заједно са додатим RD чини 96-битну VPN-IPv4 руту. Пошто не постоји ни један протокол рутирања који је прилагођен за рад са 96-битним рутама, јасно је зашто је за ову сврху одабран мултипротоколарни BGP, који има могућност лаког проширивања и додавања посебних типова рута. Други разлог је могућност транспорта лабела између PE рутера, што ће бити објашњено касније.

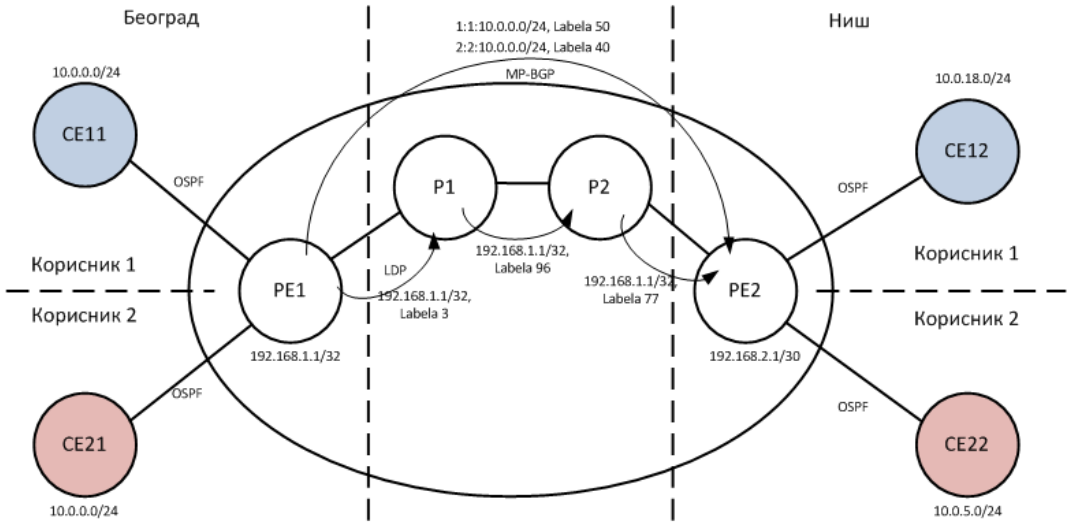
RD имају стандардом одређену структуру и у њих је могуће уписати број аутономног система или јавне адресе из тог аутономног система. Уобичајени начин записивања RD је у облику *aa:bb*, где *aa* означава тип RD и број аутономног система или адресу тог аутономног система, а *bb* неки произвољан додељени број. У примеру са слике, претпоставићемо због лакшег разликовања да су корисничким мрежама 1 и 2 додељени RD следећих вредности: 1:1 и 2:2 респективно. Руте које рутер PE1 шаље ка PE2 би у том случају биле 1:1:10.0.0.0/24 и 2:2:10.0.0.0/24.

Као што је наведено, подразумевано понашање је да све руте које припадају једној VRF инстанци остају само унутар те инстанце. Међутим, могуће је конфигурисати пребацивање рута из једне VRF инстанце у другу, када се жели да се две или више виртуелних приватних мрежа споје (на пример екстранет повезивање мрежа две различите фирме или повезивање VPN на интернет). Ово се чини помоћу тзв. *route target*-а (RT), правила којима се дефинише које ће руте бити пребачене из једне VRF инстанце у другу (импорт и експорт рута).

Важно је напоменути да се унутар MPLS мреже користи и даље неки интерни протокол рутирања и *frame-mode* MPLS за транспорт пакета између PE рутера, што ће бити објашњено у наставку текста.

3.1.6.4. MPLS L3VPN раван података – пренос пакета

Претходно је показано да је увођењем VRF инстанци и RD омогућено да на контролној равни, равни протокола рутирања, руте које припадају једној виртуелној приватној мрежи транспортују тачно између одговарајућих CE рутера и тиме је испуњена једна од предвиђених функција MPLS L3VPN. Оно што је и даље потребно објаснити је то како се пакет који је упућен од рутера PE1 и намењен дестинацији унутар мреже 10.0.0.0/24 из претходног примера рутира тачно до одговарајућег CE рутера. На PE1 рутеру постоје две VRF инстанце у којима се налази иста рута 10.0.0.0/24, само добијена од различитих CE рутера.



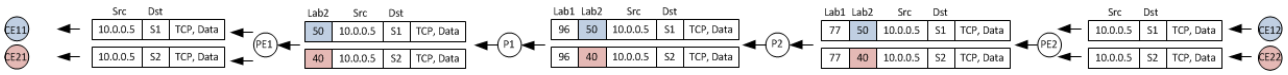
Слика 3.16 Начин успостављања лабела код MPLS L3VPN

Очигледно је да мора да постоји неки механизам којим ће се разликовати и сами пакети који припадају различитим виртуелним приватним мрежама. Ово је код MPLS L3VPN решено тако што је уведено да сви пакети имају две лабеле. Унутрашња лабела (она ближа IP заглављу) представља лабелу која означава виртуелну приватну мрежу, а спољашња лабела, ближа протоколу слоја везе представља лабелу која служи за транспорт пакета кроз MPLS мрежу. Унутрашња лабела остаје непромењена током целе путање пакета кроз MPLS мрежу, док се спољашња мења по правилима *frame-mode* MPLS.

Целокупан процес транспорта рута и пакета у MPLS L3VPN је објашњен помоћу примера са слике 3.16. Претпоставља се да рутери PE1 и PE2 имају неки од интерфејса са адресама 192.18.1.1/32 и 192.168.2.1/32 респективно. Ове руте се размењују унутар MPLS мреже неким интерним протоколом рутирања како би могла да се оствари комуникација између свих рутера у мрежи. Такође, по правилима *frame-mode* MPLS, сви рутери у MPLS мрежи алоцирају лабеле за те руте и путем LDP протокола размењују информације о њима, што је

приказано на слици за адресу рутера PE1. Ове лабеле ће бити спољашње лабеле у пакетима. У примеру са слике претпоставља се да се користи механизам PHP, јер је ово подразумевани начин рада код MPLS VPN због двоструких лабела у пакетима.

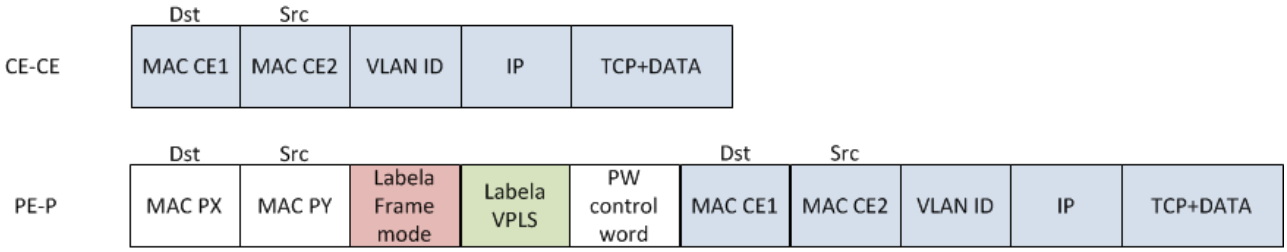
Са друге стране, мултипротоколарни BGP преноси између PE рутера VPN руте обележене помоћу RD и за те руте предлаже лабеле које ће се користити у пакетима. Ове лабеле су унутрашње лабеле у пакетима, а као и у *frame-mode* MPLS додељују се од стране низводног рутера, односно излазног PE рутера за неку дестинацију. Слика 3.17 показује кључна заглавља у пакетима када пакети пролазе кроз мрежу од рутера CE 12 и CE 22 до рутера CE11 и CE21. Спољашње лабеле су додељене по правилима *frame-mode* MPLS и мењају се на свакој вези између рутера, док унутрашње лабеле остају непромењене током проласка пакета кроз мрежу и служе излазном PE рутеру да може да раздвоји пакете који припадају различитим виртуелним приватним мрежама и проследи их до одговарајућих CE рутера. Читаоцу се препоручује да уради практичан пример L3VPN показан у поглављу 7.6., у којем може да се детаљно види начин рада на контролном нивоу и нивоу пакета.



Слика 3.17 Енкапсулација пакета приликом проласка кроз MPLS L3VPN

3.1.7. MPLS L2VPN

Код MPLS L2VPN се између CE рутера успоставља веза на слоју везе (*data link*), као да су повезани виртуелним свичем, као што је показано на слици 3.12. Постоје две врсте L2VPN веза: VPLS (енг. *Virtual Private LAN Segment*) код којих је више уређаја повезано у VPN – симулација локалне рачунарске мреже преко MPLS и VPWS (енг. *Virtual Private Wire Segment*) код који су два уређаја повезана – симулација виртуелне везе између два уређаја. Начин рада стандардизованих верзија MPLS L2VPN на контролној равни и равни података је сличан начину рада MPLS L3VPN. За разлику од L3VPN, нема размене рута између CE и PE уређаја, јер MPLS мрежа не представља виртуелни рутер, али за транспорт пакета се користи *frame-mode* MPLS, где се спољашње лабеле додељују на исти начин као код L3VPN. Што се тиче додељивања унутрашње лабеле, постоје две варијанте које су потекле из различитих компанија које производе мрежну опрему, а добиле су имена према ауторима: Kompella [3.6] и Martini, касније стандардизован као [3.7]. Код прве се за размену унутрашњих лабела између PE рутера користи BGP протокол, док се код друге користи LDP протокол. У обе варијанте се CE уређаји повезују на посебне софтверске инстанце на PE рутерима које припадају посебним L2 мрежама и којима се раздвајају информације о њима. Постоје и још неке варијанте L2VPN које су специфичне за поједине произвођаче (нпр. Juniper-ов CCC – *Circuit Cross Connect*), али оне нису добиле већу примену због лошије скалабилности.



Слика 3.18 Енкапсулација L2 датаграма који се размењују између CE рутера у MPLS L2VPN

На слици 3.18 је приказан начин енкапсулације L2 датаграма који се размењује између CE рутера и на везама између P и PE рутера. Цео датаграм, укључујући и оригинално L2 заглавље се енкапсулира у две лабеле (или једну на последњој MPLS вези због PHP), унутрашњу која означава инстанцу L2 мреже и спољашњу која служи за транспорт кроз мрежу. Овиме се постиже то да је за CE рутере повезивање кроз MPLS мрежу потпуно транспарентно, као да су директно повезани. У пакетима постоји и контролно поље које служи за исправно слање L2 датаграма и неке функције попут фрагментације, очување редоследа пакета и слање неких контролних информација [3.8].

3.1.8. Оптимизација искоришћења ресурса мреже - MPLS TE

У поглављу 3.1.1.1 наведен је проблем неоптималног искоришћења ресурса рачунарских мрежа када се користе стандардни протоколи рутирања и прослеђивање засновано на дестинационим адресама. Да би се ресурси у мрежи оптимално користили потребно је да постоји информација о томе колико је заузеће ресурса у сваком тренутку. То значи да је потребно или мерити тренутно заузеће свих веза, па ту информацију користити за одређивање оптималне путање или увести концепт резервације дела ресурса мреже (појединачних веза) за одређени мрежни ток или корисника.

Механизми MPLS TE (енг. *Traffic Engineering*) представљају скуп метода којима се обезбеђује оптимално искоришћење ресурса мреже на основу резервација дела капацитета веза на путањи од извора до дестинације за потребе различитих корисника. Да би се остварио довољно флексибилан модел резервације ресурса, MPLS TE уводи додатне критеријуме који могу да се користе у избору најбоље путање пакета. Ови критеријуми су:

- Захтевани капацитет: корисник мреже има могућност да захтева и добије такву путању пакета кроз мрежу која ће гарантовано имати расположив захтевани капацитет на свим везама на путањи. Механизми како се обезбеђује гарантовани проток за одређену класу саобраћаја су описани у глави 6.
- Афинитет (енг. *affinity*): афинитет је нека особина сваке појединачне везе која није обухваћена метриком протокола рутирања, која може да буде од значаја за избор оптималне путање, а која је приказана као нека бројчана вредност. У MPLS TE жаргону афинитет се зове и боја. На пример, оптичким везама између рутера може да се додели број 1, а онима реализованим сателитским везама број 2. Како ове две

наведене врсте веза и онда када су истих капацитета имају очигледно различите особине попут времена пропагације сигнала између рутера или поузданости приликом временских непогода, корисник приликом захтева да му се успостави путања кроз мрежу може да постави ограничење да жели да се путања успостави само по везама афинитета 1.

- Друга метрика: у MPLS TE мрежама се користи неки уобичајени *link-state* интерни протокол рутирања (нпр. OSPF или IS-IS) који има своју стандардну метрику. Међутим, постоји могућност да се дефинише и нека друга метрика другачија од метрике датог протокола (на пример вредност одређена на основу кашњења на линку). Подразумевано понашање рутера у MPLS TE је да користе управо ту другу метрику коју је могуће подесити за сваку од веза рутера. Ако ова метрика није експлицитно дефинисана, онда је њена вредност једнака вредности метрике интерног протокола рутирања. Рутери неће никада проналазити путању која је оптимална по два критеријума (две метрике) ако су дефинисани, јер је тај проблем НП-потпун [3.9] [3.10], те би процесорско оптерећење рутера било велико сваки пут када би морао да ради ову врсту оптимизације.
- Приоритет/Прече право (енг. *preemption*): Механизам приоритета односно пречег права је механизам којим се омогућава рутерима да одлуче о значају (приоритету) сваког захтева за резервацијом путање одређеног капацитета и да у ситуацијама када дође захтев већег приоритета, он има право, уколико је потребно, и да раскине постојеће везе. Постоји 8 нивоа приоритета које може да има сваки захтев, при чему најнижа вредност (0) има највећи приоритет. Захтев који има нижу вредност приоритета има право да раскине постојећу резервацију чија је вредност приоритета виша уколико му је потребан одговарајући капацитет. Ово је приказано на примеру на слици 3.19. На овој слици, са леве стране је дат пример вредности приоритета на једној вези капацитета 100 на којој постоје активне резервације капацитета 20 чија је вредност приоритета 2 и капацитета 20 чија је вредност приоритета 5. Показано је да ако би у тој ситуацији дошли захтеви са вредностима приоритета 0 или 1, имали би право да добију сав капацитет везе, иако постоје резервације на вези и у тој ситуацији, ако је потребно раскину једну или обе постојеће резервације, ако им је потребан толики капацитет јер су приоритетнији. Са друге стране ако би дошли захтеви приоритета 2 до 4, они би могли да добију максимални капацитет 80, јер не би могли да раскину постојећу резервацију приоритета 2, али би могли да раскину резервацију приоритета 5. Захтеви приоритета 5 до 7 не могу да раскидају постојеће резервације које имају мањи или једнак приоритет, те могу да добију максимални капацитет 60.

Прече право	Приоритет
0	100
1	100
2	80
3	80
4	80
5	60
6	60
7	60

Прече право	Приоритет
0	100
1	100
2	80
3	60
4	60
5	40
6	40
7	40

Слика 3.19 Пример резервације капацитета на основу приоритета

На овој слици са десне стране је показана ситуација у којој је након прве две резервације дошла трећа, капацитета 20 и приоритета 3. Приликом сваке резервације ресурса у мрежи оваква табела која се чува за сваку појединачну везу се ажурира новим вредностима.

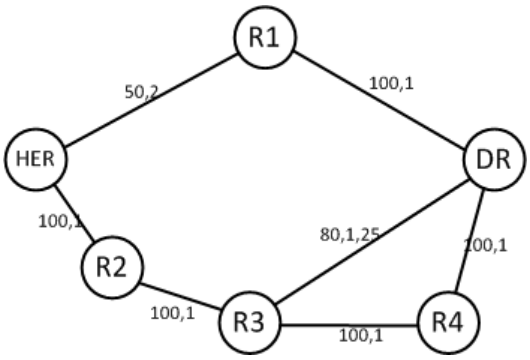
3.1.8.1. Одређивање оптималне путање

У претходном излагању показани су додатни критеријуми који у MPLS TE служе за оптимизацију одређивања путања кроз раунарске мреже. Када дође нови захтев за успостављањем путање, неки уређај у мрежи мора да на основу података добијених из мреже и постојећих резервација израчуна оптималну путању за нови захтев. Тај уређај може да буде неки од рутера MPLS TE мреже или неки екстерни централни уређај (на пример сервер). Ова два начина одређивања оптималне путање имају и неке предности и мане, па не може да се каже да је неки приступ бољи. MPLS TE користи први начин одређивања оптималне путање, код ког онај рутер који се налази на почетку жељене резервације (тзв. *headend* рутер) одређује оптималну путању. Предност овог начина одређивања оптималне путање је тај што рутери свакако имају све информације о статусу мреже кроз информације о статусу интерфејса и протоколе рутирања, па могу да лако аутоматски реагују на евентуалне промене у топологији. Проблем са овим начином одређивања путања је тај што у неким ситуацијама, које ће бити објашњене у поглављу 3.1.8.3 путање ипак могу да буду неоптималне. Са друге стране код екстерног одређивања оптималне путање, какав се користи у новијим технологијама попут софтверски дефинисаних мрежа [3.11], проблем је организација слања контролних информација о статусу веза према централној локацији, као и чињеница да један такав уређај представља критичну тачку чијим отказом цела мрежа постаје нефункционална.

3.1.8.2. Рутирање са ограничењима

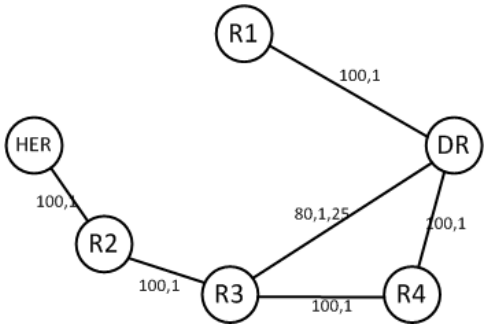
Да би се пронашла оптимална путања која задовољава све захтеве корисника користи се тзв. рутирање са ограничењима (енг. *Constraint-Based Routing* – CBR). Овакав начин рутирања подразумева да се из физичке топологије мреже избацују све оне везе које не задовољавају неки постављени критеријум, па да се на тако скраћену топологију примењује алгоритам за

израчунавање оптималне путање (*Дајкстра* алгоритам у случају *link-state* потокола). Ово ће бити објашњено на примеру са слике 3.20.



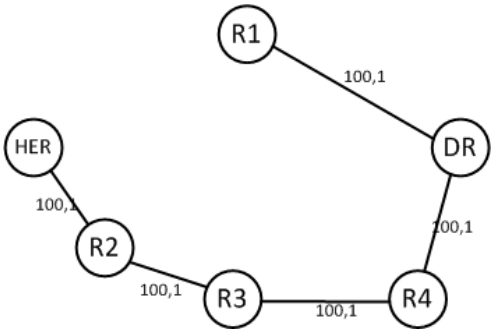
Слика 3.20 Пример мреже у којој је примењено рутирање са ограничењима

На овој слици је дата топологија мреже са капацитетима веза и метрикама које су рачунате на начин као код OSPF протокола (100/капацитет). Посматраће се успостављање путања од *headend* рутера (HER) до дестинационог рутера (DR). У овој мрежи постоје три алтернативне путање различитих метрика: HER-R1-DR (метрика 3), HER-R2-R3-DR (метрика 3,25) и HER-R2-R3-R4-DR (метрика 4). Класична IP мрежа би одабрала прву путању. Ако би се уз захтев да се пакети прослеђују до дестинационог рутера поставило и ограничење да за дате пакете треба да се обезбеди најмањи капацитет путање 60, онда би применом методе рутирања са ограничењима била из топологије избачена веза HER-R1 јер не задовољава постављени критеријум и добила би се топологија као на слици 3.21.



Слика 3.21 Везе које су остале у топологији након примене критеријума да је минимални капацитет путање 60

У оваквој скраћеној топологији преостале су две алтернативне путање, а била би одабрана она која води везом HER-R2-R3-DR јер има мању метрику. Уколико би постојао захтев да се успостави путања која захтева капацитет 90, онда би из топологије била избачена и веза R3-DR јер је њен капацитет 80 и добила би се топологија као на слици 3.22, на којој је преостала једна једина путања кроз мрежу HER-R2-R3-R4-DR.



Слика 3.22 Везе које су остале у топологији након примене критеријума да је минимални капацитет путање 90

На сличан начин би могла да се поставе ограничења у погледу афинитета, па би онда из топологије биле избачене све оне везе које не задовољавају одређени захтевани афинитет.

3.1.8.3. Проширање протокола рутирања за MPLS TE

Претходно описани алгоритам рада рутирања са ограничењима је имплементиран у пракси кроз проширења интерних *link-state* протокола рутирања (OSPF-TE[3.12] и ISIS-TE[3.13]). У овом поглављу ће бити описан начин на који је то реализовано у OSPF-TE протоколу.

Седам година након усвајања прве верзије OSPF протокола, направљен је додатак овог протокола којим су предвиђена три нова LSA (енг. *Link-State Advertisment*)[18] у које је могуће уписати произвољне податке за будућа проширења (тзв. *Opaque* LSA) [3.14]. То су LSA број 9, 10 и 11, који се прослеђују на вези, у једној OSPF области (енг. *area*) или по целом аутономном систему респективно.

OSPF-TE користи ове додатне LSA поруке којима рутери оглашавају све посебне атрибуте и ограничења који постоје за све везе које су део TE мреже, а описани су претходно. Ови атрибути су:

- *TE метрика* (4 бајта) у који је уписана вредност друге метрике на датој вези, ако је друга метрика конфигурисана
- *Maximum Bandwidth* (4 бајта) који представља капацитет дате везе
- *Maximum Reservable Bandwidth* (4 бајта) који представља део капацитета везе коју може да искористи TE алгоритам за доделу ресурса. Ова вредност може да буде мања од капацитета везе, када један део капацитета може да се искористи за TE резервације, а оставља се део капацитета везе за неке друге потребе, али може да буде и већа од капацитета везе. У овом другом случају се ради пребукирање (енг.

18 Подсетник: У стандардном OSPF протоколу LSA порукама рутери оглашавају статусе својих веза и у њиховом опису стоје информације о врсти везе (*point-to-point* или *multipoint*), IP адресама на оба краја везе и идентификатору везе. Након пријема свих LSA порука, рутери могу да израчунају оптималне путање кроз мрежу применом Дајкстра алгоритма.

oversubscription) ресурса мреже, када се оператер мреже узда у то да је таква статистичка природа расподеле коришћења ресурса различитих корисника да неће доћи до загушења (нпр. део корисника је из САД, а други део из Јапана и периоди максималног коришћења када су потребни максимални капацитети су различити у току дана).

- *Unreserved Bandwidth* (32 бајта) представља 8 вредности расположивог капацитета за резервације различитог приоритета, на начин који је описан претходно.
- *Administrative Group* (4 бајта) у који се уписују вредности афинитета.

Помоћу ових атрибута могуће је детаљније описивање свих веза и реализација оптималних путања у мрежи на основу различитих критеријума, а не само на основу дестинације пакета. Након што сви рутери размене LSA поруке са свим претходно описаним атрибутима, *headend* рутер може да одреди оптималну путању.

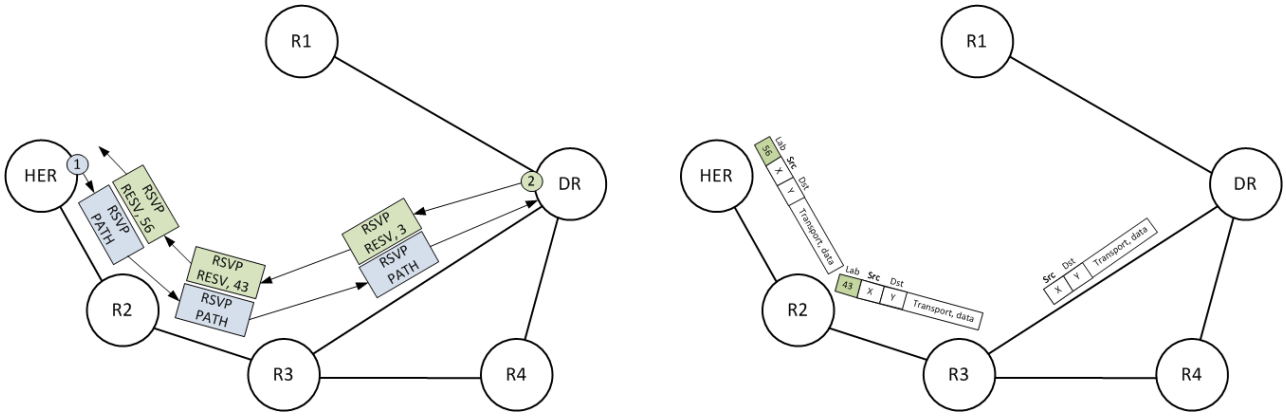
Важно је напоменути једну значајну последицу оваквог рада протокола рутирања на перформансе рутера. За разлику од класичног OSPF који шаље LSA поруке након промене конфигурације или статуса неког од линкова (промена топологије мреже) што захтева рекалкулацију рута на свим рутерима у OSPF области, код OSPF-TE се шаљу LSA поруке и приликом сваке нове резервације, зато што се том приликом мења статус везе кроз промену садржаја атрибута *Unreserved Bandwidth*. Ово значи да ће MPLS TE мрежа у просеку имати више промена статуса веза у јединици времена од класичне IP мреже, а то значи и потребу за чешћим израчунавањем оптималних путања Дајкстра алгоритмом. Како комплексност овог алгоритма расте са квадратом броја чворова у мрежи, тежи се минимизацији броја ситуација када треба да се овај алгоритам примени, па је на рутерима често имплементирана могућност да се ограничи број ситуација када ће се слати LSA поруке, тако што се постави неки праг испод кога се неће реаговати на промене. На пример: за све резервације које су мање од 10% капацитета везе неће се слати нови LSA, већ ће се он слати само онда када нове резервације кумулативно пређу тај праг. Ово објашњава тврдњу из поглавља 3.1.8.1 да ако се израчунавање оптималних путања врши на рутерима, да се неће нужно пронаћи оптимална путања, јер рутери у неком тренутку можда немају потпуно тачан скуп информација о свим резервацијама у мрежи.

3.1.8.4. Успостављање MPLS TE путање

Након што *headend* рутер одреди оптималну путању у складу са постављеним скупом ограничења, потребно је да се ова путања за пакете (LSP) формира и одреде лабеле које ће се користити на њој. За ово је искоришћена екстензија протокола RSVP[19] (*Resource Reservation Protocol*) – RSVP-TE [3.15]. *Headend* рутер шаље RSVP-TE PATH поруке дуж одређене путање. Ове поруке се разликују од RSVP PATH порука по томе што имају додатне објекте. Кључни додатни објекти су LABEL_REQUEST којом се захтева успостављање нове LSP путање и EXPLICIT_ROUTE, у који се уписују адресе свих рутера дуж оптималне путање

19 RSVP протокол је предвиђен за обезбеђење посебног третмана за класичне IP пакете у оквиру архитектуре интегрисаних сервиса за обезбеђење квалитета сервиса о чему ће бити више речи у поглављу 6.3.

коју је одредио *headend* рутер, чиме се задаје путања PATH порука тако да иду управо одређеном оптималном путањом, а не онако како би их рутирао обичан интерни протокол рутирања. У оквиру LABEL_REQUEST објекта постоји могућност да се упишу атрибути сесије, односно карактеристике тражене путање која се формира. Уколико сви рутери дуж путање могу да обезбеде тражену путању, почев од последњег рутера на путањи шаљу се RSVP RESV поруке у оквиру којих се алоцира лабела која ће се користити на свакој од веза на траженој путањи. Ово је показано на слици 3.23 за пример када се тражи резервација путање капацитета 70. На десном делу слике су показани пакети са одговарајућим лабелама када се користи и механизам PHP.



Слика 3.23 Начин успостављања лабела помоћу RSVP-TE

Уколико неки од рутера дуж одређене путање није у могућности да обезбеди путању захтеваних карактеристика, послаће *headend* рутеру поруку са грешком. До овакве ситуације може да дође онда када *headend* рутер нема потпуно тачну информацију о статусу свих резервација у мрежи и може да захтева резервацију која превазилази расположиве ресурсе.

3.2. Мобилност у рачунарским мрежама

Концепт мобилности уређаја који су повезани на неку комуникациону инфраструктуру је данас лако разумљив. Са мобилним телефоном је могуће кретати се кроз мрежу свог мобилног оператера, али и прелазити у суседну земљу, у мрежу другог мобилног оператера, а да при томе услуга остаје стално доступна. Такође, број телефона, идентификатор уређаја на кога је могуће позвати његовог корисника се том приликом не мења, а онај ко позива мобилни уређај није свестан где се овај уређај налази – веза се свакако успоставља.

Аналогија овоме би била ситуација у којој би се неки корисник рачунарске мреже кретао из једне у другу мрежу (на пример носио свој лаптоп), при томе задржаво фиксну IP адресу са своје матичне локације (нпр. из организације у којој ради и чији је лаптоп), а да све комуникационе везе које има у неком тренутку остану непрекинуте приликом преласка у другу мрежу. Ако се изузму технички проблеми у једноставном преласку из мреже у мрежу када се користе фиксне, жичане везе (данас се много више користе бежичне везе него раније), ово у рачунарским мрежама са стандардним коришћењем IP протокола није могуће.

Кључни проблем је тај што је семантика IP адресе таква да се у њој заједно налазе како информација о идентификатору уређаја (*host* део адресе), тако и информација о локацији уређаја (*network* део адресе – идентификатор мреже). Уколико би уређај са фиксном IP адресом која потиче из једне мреже дошао у другу мрежу, престао би да буде доступан у новој мрежи јер нова, гостујућа мрежа не би оглашавала руту ка адреси мреже мобилног уређаја. Да би се овај проблем превазишао осмишљен је концепт Мобилног IP (енг. *Mobile IP*) који користи двоструку IP енкапсулацију (IP-in-IP).

3.2.1. Мобилни IP - IP-in-IP енкапсулација

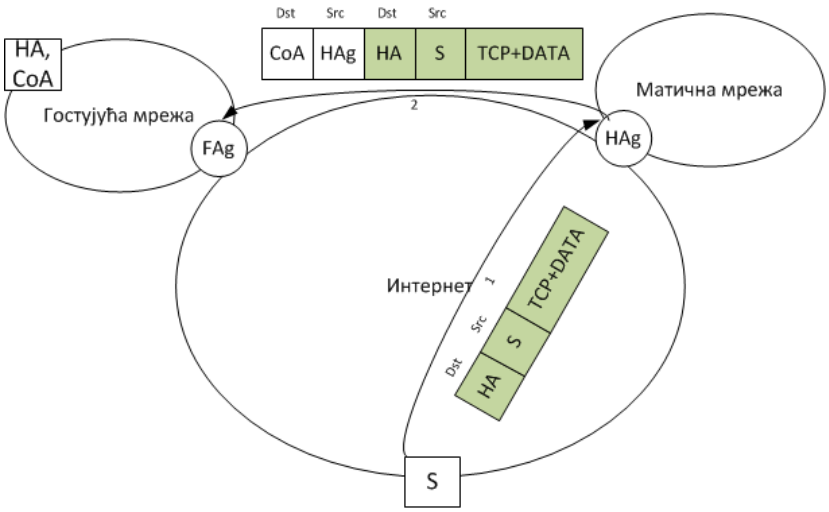
У оквиру концепта мобилног IP уведена је следећа терминологија [3.16]:

- Мобилни уређај чије се кретање посматра у својој матичној мрежи из које потиче има IP адресу која се зове *Home Address* (HA). Ова адреса је јавна IP адреса и припада адресном простору матичне мреже коју она оглашава да би била доступна преко Интернета. Циљ увођења Мобилног IP је да се омогући да мобилни уређај може да буде доступан преко своје HA и када јесте и када није у матичној мрежи, а да при томе неки уређај S који комуницира са њим није свестан локације на којој се налази мобилни уређај.
- Рутери који се налазе обично на излазу из мреже, који подржавају Мобилни IP се зову агенти. Они могу бити агент у матичној мрежи датог уређаја (енг. *Home Agent* – HAg) и агент у гостујућој мрежи у коју је мобилни уређај прешао (енг. *Foreign Agent* – FAg). Агенти воде евиденцију о томе да ли се уређаји којима је мрежа у којој су они агенти матична налазе у тој мрежи или су прешли у неку другу мрежу (енг. *Mobility binding table*) и такође воде рачуна о свим мобилним уређајима из других мрежа који су дошли привремено у њихову мрежу (енг. *Visitors table*).
- Да би мобилни уређај могао да буде доступан и када се налази у гостујућој мрежи, он у њој мора да добије привремену IP адресу која се зове *Care-of Address* (CoA) која припада адресном простору гостујуће мреже. CoA може да се додељује на два начина:
  - тако што ће сваки гостујући уређај добити посебну CoA из одређеног скупа адреса за гостујуће уређаје. То се зове *Collocated* CoA.
  - тако што сви гостујући уређаји добијају једну исту CoA која се налази на агенту. Ово се зове *Foreign Agent* CoA.

Агенти периодично шаљу поруке којима оглашавају своје присуство и којима траже од мобилних агената да им се пријаве. У овим порукама се налази једна или више CoA на основу којих (и своје HA) мобилни уређај зна да ли се налази у матичној или гостујућој мрежи. Уколико је мобилни уређај у матичној мрежи, његов HAg ће имати информацију о томе да је уређај са адресом HA у матичној мрежи и сваки пакет који је упућен њему ће бити и прослеђен на стандардан начин.

Када мобилни уређај пређе у гостујућу мрежу, након пријема огласа агента у тој мрежи, моћи ће да инсталира CoA адресу коју је добио и да захтева регистрацију код агента. FAg ће у таβелу гостију уписати HA адресу госта и IP адресу његовог матичног агента - HAg. FAg прослеђује ове регистрационе параметре матичном агенту госта и новододељену CoA, чиме и HAg бива обавештен о томе да је један од уређаја из његове мреже прешао у другу мрежу и ажурира одговарајућу таβелу. HAg потврђује пријем регистрационих порука чиме су размењене све контролне информације потребне да би се остварила комуникација са HA.

Када неки уређај S шаље пакет ка HA, пакети ће бити рутирани према матичној мрежи мобилног уређаја. На уласку у матичну мрежу ће их пресрести HAg и на основу информација у *Mobility binding table* моћи ће да одреди да се уређај налази у некој удаљеној мрежи и која је његова CoA у тој мрежи. HAg ће добијени пакет додатно енкапсулирати у још једно IP заглавље чије ће IP адресе бити IP HAg као адреса извора пакета и IP адреса CoA као дестинација – ово је IP-in-IP енкапсулација у којој постоје два IP заглавља. У овој енкапсулацији се два IP заглавља налазе директно једно до другог. Спољашње IP заглавље ће у пољу *Protocol* којим се означава енкапсулирано заглавље у IP заглављу имати вредност 4 које означава још једно IP заглавље. Тако формиран пакет ће се рутирати према CoA и доћи ће до мобилног уређаја (у зависности од врсте CoA или директно до мобилног уређаја или до FAg). Ово је приказано на слици 3.24.



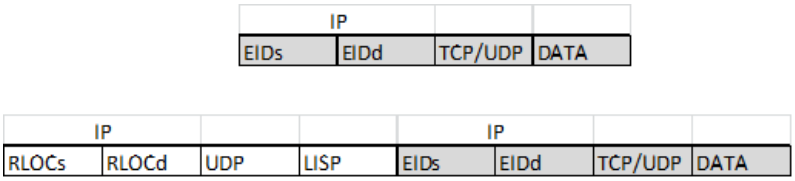
Слика 3.24 Енкапсулација код Mobile IP

Овакво рутирање преко HAg очигледно није оптимално јер пакети не иду најкраћом путањом (и у жаргону рачунарских мрежа се зове „троугласто“ рутирање – енг. *triangular routing*). Могу да се замисле ситуације у којој су S и HAg тополошки близу (у истој земљи), а матична мрежа мобилног уређаја на другом крају света, па би пакети од извора до дестинације прелазили вишеструко већи пут од оптимално потребног. Да не би долазило до троугластог рутирања, било би потребно да чворови у мрежи (рутери) бележе локацију сваког мобилног уређаја, што у данашњој ситуацији са милијардама уређаја на интернету и све већим бројем преносивих уређаја не би био скалабилан приступ, те је троугласто рутирање једини прихватљив компромис.

У повратном смеру би мобилни уређај теоретски могао да направи IP пакет са адресом извора HA и адресом дестинације S, директно га пошаље према S и тиме избегне враћање пакета према HAg и троугласто рутирање. Међутим у пракси је ово мало вероватно јер већина мрежа на Интернету има имплементиране филтре којим се спречава слање пакета са лажних адреса (*IP Spoofing*) тако што се допушта да из неке мреже изађу само они пакети који имају изворишну адресу која припада скупу адреса те мреже. Како HA не припада скупу адреса гостујуће мреже, овакви пакети би били филтрирани, па је једино могуће решење онда троугласто рутирање и у повратном смеру коришћењем IP-in-IP енкапсулације ка HAg.

3.2.2. Location/ID Separation протокол – LISP

Једна од новијих иницијатива да се глобално реши проблем мобилности, али и проблеми скалабилности табела рутирања на интернету и ефикаснијег повезивања мрежа на више пружалаца услуга је *Location/ID Separation* протокол – LISP [3.17]. Као што му и име каже, једна од кључних идеја овог протокола је да се раздвоји локација од идентификатора уређаја на интернету, која је претходно наведена као проблем у реализацији мобилности у рачунарским мрежама. LISP припада групи инкременталних инцијатива за унапређење интернета која може да се постепено уводи, без икаквих промена за крајње уређаје.



Слика 3.25 Изглед пакета неосредно након слања са EID уређаја и након енкапсулације на RLOC рутерима.

Предвиђено је раздвајања IP адреса у две независне подврсте: кориснички уређаји би добијали IP адресе које се називају *Endpoint identifier* – EID, а рутери IP адресе које се називају *Routing LOCator* – RLOC. Интернет табела рутирања садржи само RLOC адресе, док EID адресе не могу да буду у табелама рутирања. Крајњи уређаји шаљу пакете са EID адресама у IP заглављу. Ти пакети када дођу до неког од рутера са RLOC адресама се додатно енкапсулирају на начин приказан на слици 3.25. како би могли да буду пренесени до одговарајуће локације дестинационог EID. Један од кључних проблема код LISP је мапирање EID у одговарајуће локације на интернету – иза ког RLOC се налази сваки EID.

3.3. Заштита података послатих преко мреже – IPsec и SSL

Један од кључних недостатака класичних рачунарских мрежа заснованих само на IP протоколу је недостатак подршке за пружање заштите података који се шаљу мрежом. Ако се

зна каква је организација интернета и то да учесник у комуникацији не може никада да буде сигуран којим путем и преко којих мрежа ће пакети са његовим подацима проћи, као и ко све може да има увид у њих, те да пакети пролазе кроз различите мреже у различитим земљама са различитим правним системима, да различито мотивисани нападачи могу да дођију приступ на практично свакој тачки на интернету, јасно је да је за било какве озбиљне пословне или финансијске примене нужно обезбеђивање заштите података који се преносе преко рачунарских мрежа.

Заштита података обухвата следеће основне активности:

- обезбеђивање поверљивости порука које се шаљу преко мреже. Поверљивост се обезбеђује енкриптовањем, на тај начин да само пошиљалац и прималац поруке могу да дођу до њеног садржаја, док евентуални нападач који има могућност да пресретне и сними пакете док пролазе кроз мрежу не може да дође до тог садржаја под претпоставком да није на неки начин дошао до кључева за енкрипцију.
- заштиту интегритета података пренетих преко мреже која подразумева гаранцију да садржај порука није мењан током проласка кроз мрежу. Постоје неке ситуације када информације које се преносе мрежом нису тајне, те их није нужно заштитити криптовањем, али је битно да се обезбеди да није дошло до промене података, јер би промена садржаја могла да доведе до дезинформације и наведе некога на погрешну акцију.
- проверу аутентичности порекла порука. Под провером аутентичности порекла порука се подразумевају најмање две различите ствари: потврда да је порука стигла тачно од одређеног лица са којим се комуницира и потврда да је порука стигла са тачно одређене IP адресе са које је пакет послат. У оба случаја провера аутентичности није тривијална јер неке традиционалне шеме које се користе за проверу идентитета и порекла (нпр. потпис или провера чулима фотографије у личној карти и саме израде личне карте која пружа сигурност да је особа са којом се комуницира заиста она чија се лична карту посматра) нису применљиве у свету комуникације уређаја који између себе размењују лако измењиве дигиталне артефакте.

Заштита података у рачунарским мрежама може да се обавља на различите начине:

- заштита на апликативном слоју која подразумева да свака апликација има имплементиране механизме којима се штити садржај њене комуникације. Тако на пример постоји PGP[20] скуп механизама, алата и додатака за мејл клијенте којима се штите мејлови. Уколико се заштита успоставља на апликативном слоју, онда се штити само и искључиво апликација у којој су активирани механизми заштите, док су све остале апликације које немају ове механизма незаштићене.
- заштита на транспортном слоју која подразумева заштиту на нивоу TCP или UDP порта. Често је функционално слична заштити на апликативном слоју, јер када се

20 https://www.openpgp.org/

користи заштита на пример HTTP саобраћаја коришћењем SSL протокола, тако да се користи HTTPS, то јесте пример заштите на транспортном слоју, али заправо значи заштиту комуникације између веб прегледача и неког веб сервера који је конфигурисан да ради као HTTPS сервер.

- заштита на мрежном слоју која подразумева заштиту целокупног саобраћаја између:
  1. два рачунара или
  2. две локалне рачунарске мреже или
  3. рачунара и локалне рачунарске мреже, између којих се успоставља заштићена мрежна веза.

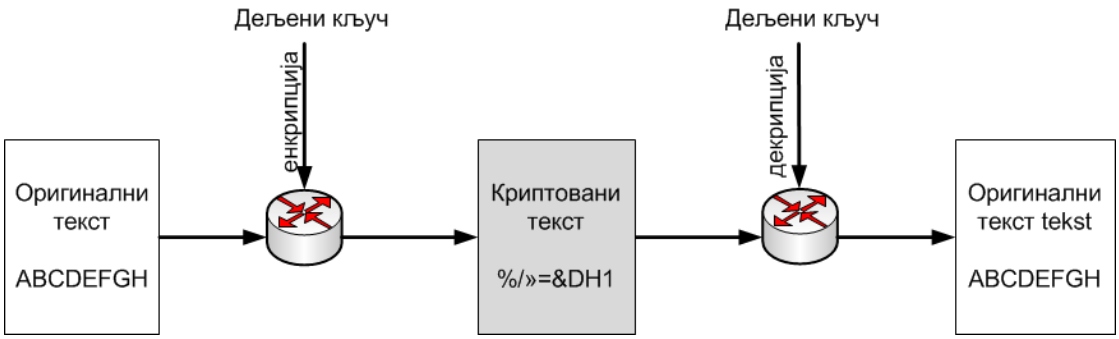
  Рачунари и локалне рачунарске мреже том приликом могу да буду на потпуно различитим деловима интернета. Уколико је успостављена заштита на мрежном слоју између нека два ентитета, онда ће комуникација свих апликација између тих ентитета бити заштићена, без обзира на механизме заштите самих апликација.
- заштита на слоју везе значи заштиту целокупног саобраћаја на сегменту између два директно повезана уређаја. Овакав начин заштите није практичан за заштиту комуникације ентитета који нису директно повезани и када се захтева заштита с краја на крај, јер би подразумевао да заштита мора да се обавља на сваком појединачном сегменту од којих се састоји цела путања.

3.3.1. Преглед механизама заштите

У овом поглављу се даје кратак преглед начина и ефеката рада алгоритама и протокола којима се врши заштита података. Подразумева се да ће читалац детаљна сазнања о овим механизмима добити или је добио на посебном курсу.

3.3.1.1. Симетрични криптографски алгоритми

Симетрични криптографски алгоритми врше енкрипцију (шифровање) и декрипцију (дешифровање) коришћењем истог кључа, што је приказано на слици 3.26. Ови алгоритми имају најчешће блоковску структуру: оригинална порука која се криптује се подели у блокове једнаке величине (нпр. 128, 192 или 256 бита), па се онда алгоритам примењује на те блокове, сваки посебно или уланчано.

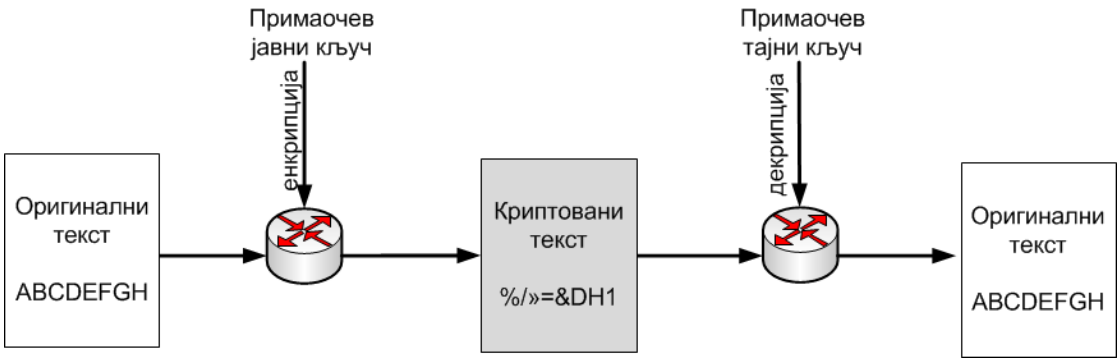


Слика 3.26 Симетрични криптографски алгоритми

Операције криптовања и декриптовања се састоје из вишеструког понављања операција XOR, шифтовања и премештања бита у блоковима. Најпознатији алгоритми који се данас користе за симетричну енкрипцију су AES (енг. *Advanced Encryption Standard*), IDEA, Blowfish и други који користе кључеве дужине 128 бита и веће. Неки раније коришћени алгоритми као што је DES се данас сматрају недовољно сигурним због кратких кључева. Један од кључних проблема, а посебно у великим системима са великим бројем учесника који имају потребу да криптују комуникацију је начин расподеле симетричних кључева, о чему ће бити више речи у наставку.

3.3.1.2. Асиметрични криптографски алгоритми

Асиметрични криптографски алгоритми подразумевају постојање два кључа која су међусобно повезана: приватни и јавни кључ. Порука криптована једним од ова два кључа може да се декриптује само другим кључем, али не и истим (на пример: порука криптована јавним кључем може да буде декриптована само приватним кључем и обрнуто – слика 3.27).



Слика 3.27 Асиметрични криптографски алгоритми

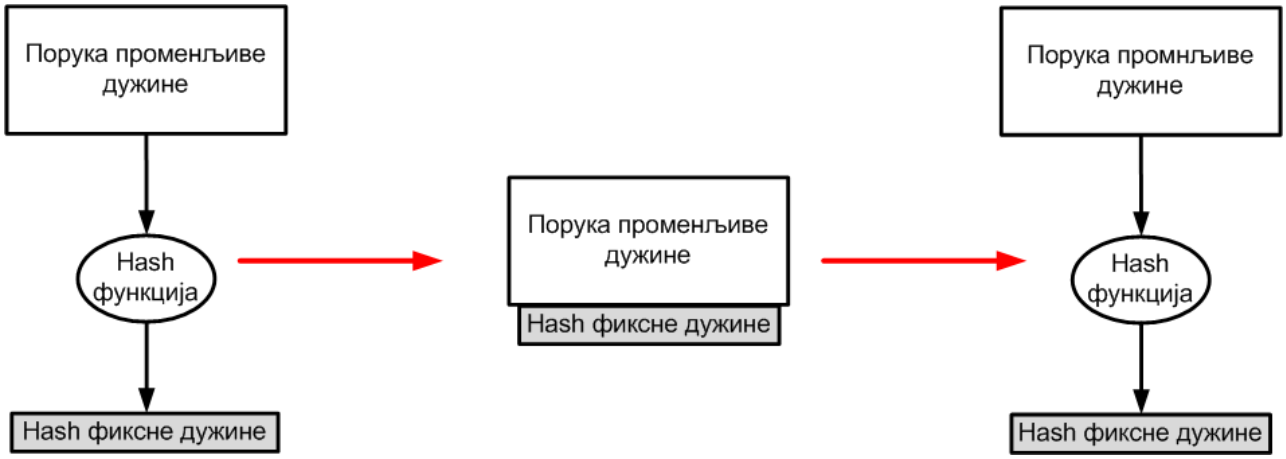
Власник кључева не открива свој приватни кључ, док јавни кључ може да дâ другим учесницима у сигурној комункацији. Уколико је нека информација криптована приватним кључем и ако тај кључ није компромитован, примаоци поруке који је декриптују јавним кључем могу да буду сигурни које је порекло дате поруке јер је само власник приватног кључа могао да криптује поруку која се исправно декриптује јавним кључем. Асиметрични

алгоритми користе математичку операцију експоненцијације по модулу (слично Дифи-Хелман алгоритму који је показан у поглављу 3.3.1.4) и то са јако великим бројевима (кључеви код ових алгоритама имају 1024 бита и више), те је њихова рачунска комплексност много већа од комплексности симетричних алгоритама. Због тога се у применама у рачунарским мрежама за заштиту пакета користе симетрични алгоритми, а за проверу идентитета постоје бројне шеме које користе асиметричну енкрипцију. Најпознатији асиметрични криптографски алгоритам је RSA (*Rivest-Shamir-Adelman*).

3.3.1.3. Хеш функције

Хеш (енг. *hash*) функције служе за проверу интегритета поруке. Одлике хеш функција су следеће:

- За поруку произвољне дужине дају излаз фиксне дужине – хеш
- Иреверзибилне су (није могуће из хеша добити оригиналну поруку)
- За сваку и најмању промену поруке, хеш ће се променити
- Тешко је (у смислу процесорског времена) пронаћи две поруке које дају исти хеш, као и направити измену постојеће поруке тако да даје исти хеш као оригинална порука



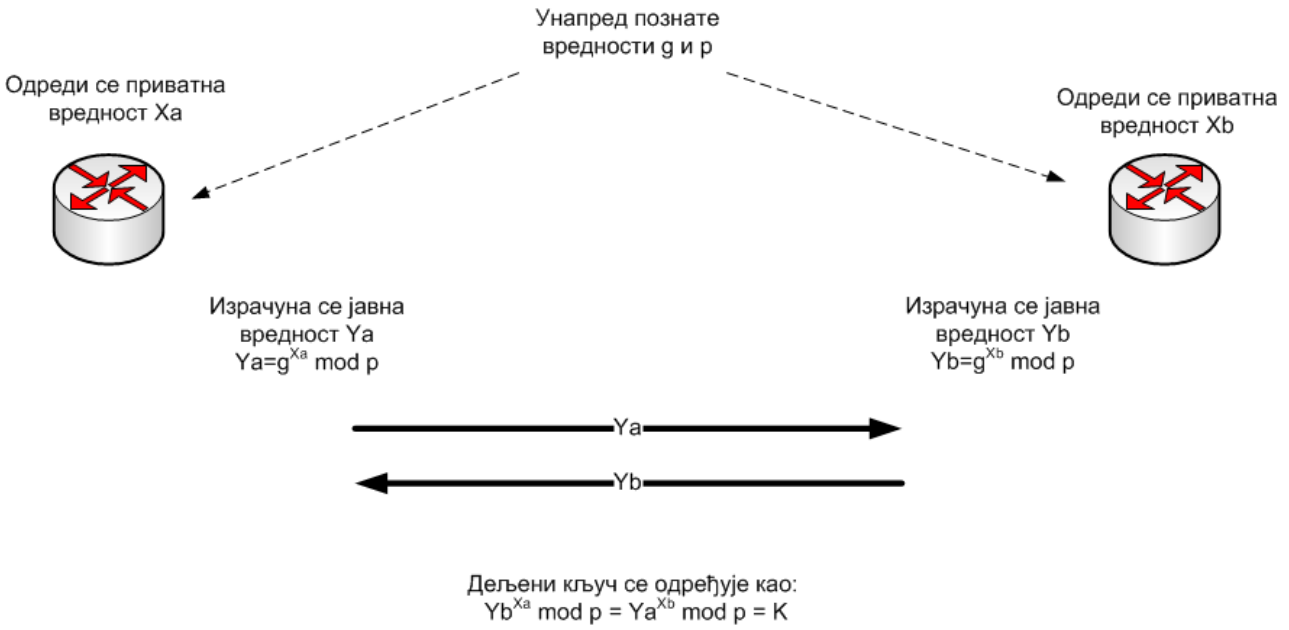
Слика 3.28 Хеш функције

Уколико се са поруком чији се интегритет штити пошаље хеш, пријемна страна израчунавањем хеша од поруке која је добијена и поређењем са добијеним хешом може да утврди да ли је дошло до промене поруке приликом преноса (слика 3.28). Шема приказана на слици 3.28 има туману што потенцијални нападач може да пресретне поруку, да је промени и уколико познаје алгоритам за израчунавање хеша, израчуна нови хеш на основу промењене поруке што чини да пријемна страна не може да детектује промену. Због тога се у рачунарским мрежама користе тзв. HMAC (енг. *Hashed Message Authentication Code*) верзије ових алгоритама код којих се на улаз хеш функције доводи и тајни кључ који деле предајна и пријемна страна. Без познавања овог кључа нападач не може да измени поруку и креира хеш

који ће преварити пријемну страну да прихвати измењену поруку. Најпознатији хеш алгоритми су MD5 и SHA и то данас њихове верзије које дају хешеве од најмање 256 бита.

3.3.1.4. Механизми размене кључева

Да би симетрични алгоритми могли да се користе мора да постоји исти кључ на обе стране комуникације. У мањим и статичким окружењима са неколико учесника ово је могуће извести и неком методом која не користи рачунарске комуникације (на пример договором, телефоном, куриром итд.). Међутим, у већим системима где се учесници налазе широм света овакав начин поделе кључева може да буде непрактичан, а посебно ако се зна да кључеви морају да се мењају с времена не време (било периодично било након што је неким кључем криптована одређена количина података). За ту потребу су развијени механизми размене кључева (енг. *Key Exchange*). Један од најпознатијих алгоритама који се користи за ову сврху је Дифи-Хелман (*Diffie-Hellman*) алгоритам чији је принцип рада приказан на слици 3.29.



Слика 3.29 Дифи-Хелман размена кључева

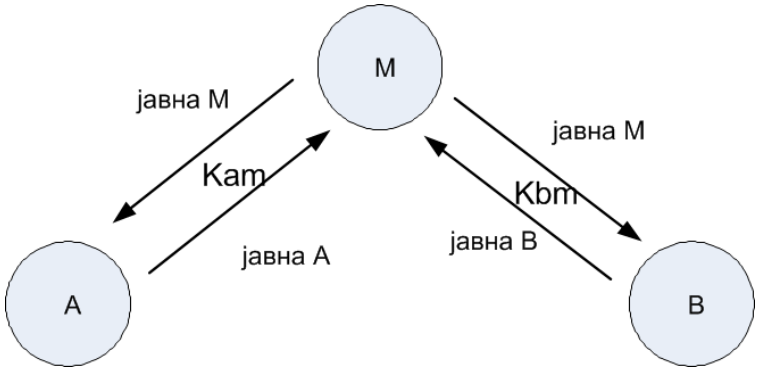
Овај алгоритам који користи математичку операцију експоненцијације по модулу је заправо алгоритам којим се генерише (а не размењује) исти кључ на две стране на начин који онемогућава нападача да из порука које може да пресетне и из вредности g и p које су унапред познате дође до тог кључа. Упркос томе, ова метода се уобичајено назива размена кључева. Сложеност и сигурност алгоритма је одређена величином броја p који је велики псеудо-прост број (најмање 1024 бита, а пожељно и много више).

Постоје и други начини за размену кључева који користе енкрипцију јавним кључем неког асиметричног алгоритма, што ће бити показано на примеру SSL VPN у наставку текста.

3.3.1.5. Неки напади на механизме размене кључева

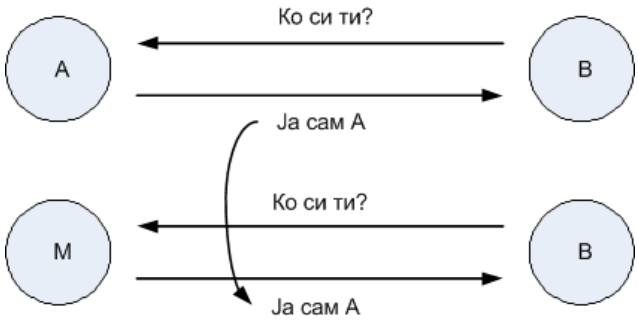
Основна верзија Дифи-Хелман алгоритма је подложна неким нападима: „човек у средини“ (енг. *Man in the Middle* – MITM) и нападима понављањем. Због тога се у пракси никада не користи Дифи-Хелман алгоритам на начин како је показано у претходном поглављу већ је размена сложенија, што је показано у поклављу о IKE протоколу.

До MITM напада може да дође у ситуацијама када је слаба (или је нема) провера идентитета учесника у размени кључева. На слици 3.30 је приказана овај случај. Уколико нападач M пресретне јавну Дифи-Хелман вредност коју је једна од страна у размени (A) послала другој (B) и одговори му својом јавном вредношћу, A и M ће моћи да креирају симетричан кључ Кам из ових јавних вредности. Ако M у исто време пошаље своју јавну вредност према B и B одговори, M и B ће моћи да креирају кључ Квм. Уколико претпоставимо да су овом приликом A и B покушали да направе заштићену комуникациону сесију у којој су успоставили видео везу (нпр. *skype*), A ће моћи да шаље видео ка B криптујући га кључем Кам, M ће декриптовати тај видео садржај кључем Кам, моћи ће да сними декриптован садржај и да га даље енкриптује кључем Квм и пошаље ка B. A и B ће имати успостављену видео везу у којој ће се видети и комуницирати, знаће да и један и други криптују сав саобраћај и вероваће да је њихова веза сигурна. Међутим, сав садржај комуникације ће бити доступан нападачу M. Разлог због ког може да дође до овог напада је тај што A и B нису приликом размене кључева проверили с ким се размена заиста обавља. Битно је напоменути да тривијални механизми провере идентитета који користе IP адресе, мејл адресе или сличне лако променљиве идентификаторе нису адекватан начин провере идентитета. У поглављу 3.3.2.3 ће бити објашњено детаљније на који начин се остварује провера идентитета у протоколима који се користе у рачунарским мрежама.



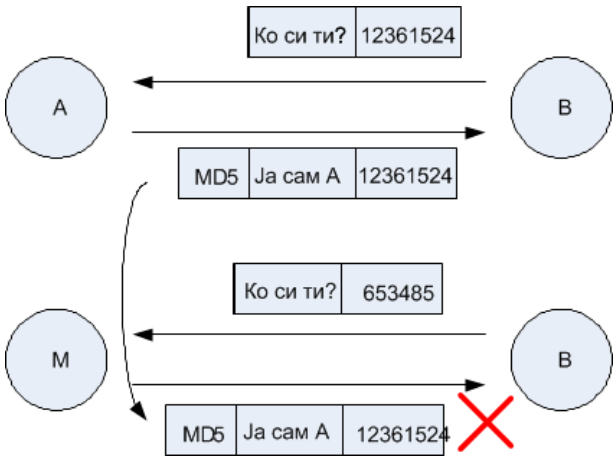
Слика 3.30 Man-in-the-middle напад

Једна од класичних шема за проверу идентитета је та да онај ко проверава нечији идентитет пошаље упит другој страни која треба да пошаље креденцијале којима доказује свој идентитет. Чак и ако је ова размена креденцијала криптована и ако су креденцијали невидљиви за нападача, у овако једноставној размени је могуће извести напад понављањем (енг. *replay attack*). Ако је нападач M забележио пакет са одговором учесника A који је послао валидне креденцијале, могао би да на упит одговори копијом одговора A и да добије приступ систему или да превари B да је он A.



Слика 3.31 Replay напад

Да би се спречили напади на проверу идентитета понављањем, уводе се јединствени идентификатори сесије који се мењају за сваку сесију и који се идеално генеришу као псеудослучајни бројеви које нападач не може да предвиди. Оваква шема је приказана на слици 3.32. B када тражи проверу идентитета A шаље уз упит (криптовано, невидљиво за нападача) и идентификатор сесије. A у свом одговору шаље своје креденцијале и хеш израчунат од креденцијала и идентификатора сесије. Ако би M у својој новој сесији покушао да пошаље пакет којим је A добио исправно потврдио свој идентитет, B би одбио такав одговор јер вредност хеша не би одговарала очекиваној пошто се идентификатори сесије различити. Из овог разлога се у разменама кључева на интернету користе случајни бројеви (nonce – енг. *number used only once*), што ће такође бити показано у поглављу 3.3.2.3.



Слика 3.32 Заштита од Replay напада

3.3.1.6. Препоруке за снагу коришћених криптографских алгоритама у рачунарским мрежама

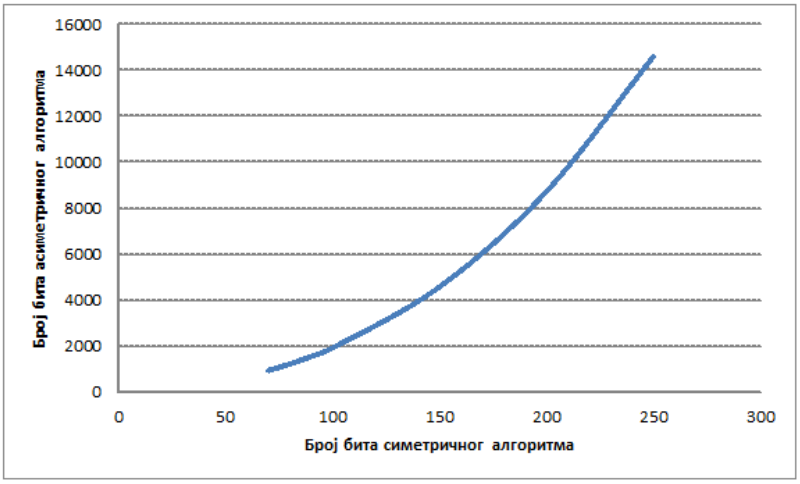
Као што је показано претходно, симетрични алгоритми користе релативно једноставне логичке операције и операције прерасподељивања бита и кључеви за њих су данас најчешће дужине између 128 и 256 бита. Са друге стране асиметрични алгоритми користе математичку операцију експоненцијације јако великих бројева који имају минимално 1024 бита и значајно су процесорски захтевнији. Због тога се искристалисало да се симетрични алгоритми користе за заштиту пакета који пролазе мрежом, док се асиметрични алгоритми користе за

проверу идентитета и у размени кључева којом приликом се врши криптовање мале количине података.

Да би неки систем био адекватно обезбеђен, потребно је да сви његови делови буду приближно једнаке сигурности. Нема много смисла користити симетричан алгоритам за који је потребно 200 година да се дође до оригиналне поруке претрагом свих могућих кључева ако се за размену кључева за овај алгоритам користи асиметрични алгоритам који је могуће разбити за 10 дана. Мора се очекивати да ће интелигентан нападач увек напасти најслабију карику у систему. Због овога су направљене препоруке о величини кључева алгоритама за енкрипцију пакета и алгоритама за размену кључева [3.18]. За симетричне алгоритме је релативно једноставно утврдити време које је потребно за њихово извршење (број операција процесора потребних за енкрипцију или декрипцију једног блока оригиналне поруке), па тиме и време потребно за сирову претрагу свих кључева. Према овој препоруци данас (2017. година) би за сигурност од 20 година[21] за податке послате симетричним алгоритмом било довољно да се користи кључ од 114 бита (што значи да је било који алгоритам с кључем величине 128 бита довољан за обично, цивилно коришћења услуга на интернету), с тим да се сваке следеће године дода 2/3 бита како би се узео у обзир развој рачунара и брзине процесирања. За асиметричне алгоритме ово није тако једноставно јер се користи математичка операција у којој само неки бројеви (псеудо-прости) могу да буду кандидати у процесу креирања RSA кључева, што значајно смањује њихов број. Стога је истим документом је дефинисна формула којом се процењује еквивалентност снаге симетричних и асиметричних алгоритама анализом времена потребног за претрагу свих могућих кључева и једних и других. На основу ње је урађен график на слици 3.33. На слици се види да је за симетричне алгоритме са кључем величине 128 бита потребно да кључ RSA алгоритма или Дифи-Хелман вредности експонента имају око 3000 бита (идеално 3072, као број који је степен броја 2). Ове препоруке одговарају и најновијим препорукама NIST – америчког Националног института за стандардизацију[22]. На жалост данас је честа пракса произвођача опреме и софтвера да алгоритми за размену кључева и проверу идентитета не поштују ове препоруке и обично је алгоритам за размену кључева слабија карика у систему.

21 Овакве процене треба узети увек са резервом, јер у рачуницу колико дуго су безбедни неки подаци криптовани одређеним алгоритмом улази већи број параметара попут тога: колико је процесора нападач спреман да употреби (колико му вреди информација до које хоће да дође), да ли ће бити пронађена нека рупа у криптографском алгоритму, да ли ће да развој рачунарске технике буде као до сада и да ли може да се једноставно екстраполира данашњи темпо развоја или ће за коју годину или месец да дође до неког значајног продора (нпр. нови алгоритам за ефикасну експоненцијацију великих бројева) који може да значајно скрати ово време итд.

22 https://www.keylength.com/en/4/



Слика 3.33 Крива еквивалентне снаге симетричних и асиметричних криптографских алгоритама

3.3.2. IPsec

IPsec је скуп протокола и механизама који се користе за заштиту података на мрежном слоју. [3.19]. Настао је у време између стандардизације IPv4 и IPv6 протокола, тако да је имплементиран као додатак на IPv4, док чини стандардни део IPv6.

Два ентитета која успостављају сигурну комуникацију на мрежном слоју успостављају IPsec сигурносну асоцијацију (енг. *Security Association*). Сигурносна асоцијација представља скуп правила и метода којима је одређен начин на који ће се заштитити неки скуп пакета и означена је јединствено на сваком уређају SPI индексом (енг. *Security Parameter Index*). Уређаји који успостављају сигурносне асоцијације имају типично две базе: SAD (енг. *Security Association Database*) и SPD (енг. *Security Policy Database*). У првој се налазе описи свих сигурносних асоцијација и механизама којима ће се штитити пакети (нпр. за сигурносну асоцијацију чији је SPI 12345678 користиће се ESP протокол[23] у тунел режиму рада са AES-256 и SHA-256-HMAC алгоритмима са одговарајућим кључевима итд.). У другој бази се налази скуп правила којим је одређено за које пакете се примењује одговарајућа сигурносна асоцијација слично приступним листама и пакетским филтерима (нпр. SPI 12345678 се користи за све пакете који иду са изворишних адреса 192.168.1.0/24 ка дестинационим адресама 10.0.5.0/24).

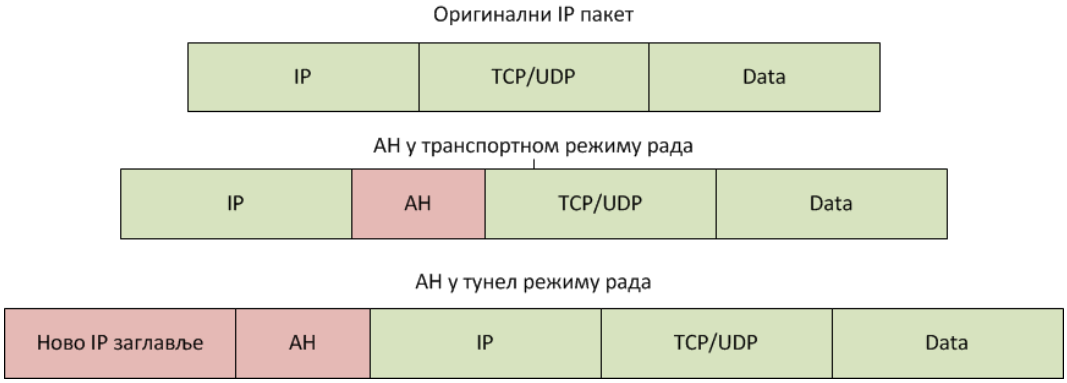
Сигурносне асоцијације су унидирекционе што значи да је за успостављање двосмерне комуникације између ентитета потребно да постоје две сигурносне асоцијације, свака у по једном смеру.

Основне компоненте IPsec скупа протокола су: *Authentication Header* (AH), *Encapsulating Security Payload* (ESP) и *Internet Key Exchange* (IKE) који ће бити описани у наставку текста.

23 Описан нешто даље у тексту

3.3.2.1. Authentication Header (AH)

Authentication Header (AH) је механизам који пре свега служи за заштиту интегритета послатих података и проверу аутентичности порекла пакета [3.20]. Може да се користи у два режима рада: транспортни и тунел режим рада, од чега зависи и који делови пакета и на који начин су тачно заштићени. Коришћење AH подразумева и додавање једног додатног заглавља. У транспортном режиму рада AH заглавље се умеће између оригиналног IP заглавља и заглавља протокола вишег слоја, док се у тунел режиму рада AH заглавље ставља испред оригиналног IP заглавља, док се испред њега додаје ново IP заглавље (Слика 3.34).



Слика 3.34 AH енкапсулација у транспортном и тунел режиму рада

Елементи AH заглавља су приказани на слици 3.35. Поље *NextHeader* означава које је заглавље које долази иза AH заглавља (у случају транспортног режима рада на пример TCP или UDP, у случају тунел режима рада – IP). Поље *Payload Length* означава дужину променљивог последњег поља (*Integrity Check Value - ICV*) у заглављу. У *ICV* пољу се налази излаз неке хеш функције која се користи за верификацију интегритета и аутентичности порекла пакета. Како ове функције имају различите дужине излаза у зависности од врсте (нпр. HMAC верзије MD5 или SHA алгоритама), потребно је да се зна дужина последњег поља да би се пакет исправно парсирао на пријемној страни.

Next Header	Payload length	RESERVED
Security Parameter Index (SPI)		
Sequence number		
Payload: Integrity Check Value - ICV (Променљива дужина)		

Слика 3.35 AH заглавље

Остала поља у заглављу су SPI индекс који омогућава пријемној страни да зна који алгоритам да користи за верификацију пакета и секвенцијални број пакета који је за први пакет у сигурносној асоцијацији 1 и монотоно се инкрементира за сваки следећи пакет. Секвенцијални број служи за спречавање напада понављањем истог пакета, јер сваки

следећи пакет мора да има инкрементирану вредност секвенцијалног броја у односу на претходни.

Улазни подаци у хеш функцију чији се излаз преноси пољем *ICV* су:

- У транспорт режиму рада: сви подаци који се налазе у AH заглављу осим ICV + сви подаци иза AH заглавља + непроменљива поља у оригиналном IP заглављу[24].
- У тунел режиму рада: сви подаци који се налазе у AH заглављу осим ICV + сви подаци који се налазе иза AH заглавља + непроменљива поља у новом IP заглављу.

Као што се из описа основних елемената заглавља види, AH коришћењем хеш функције у коју улазе сви непроменљиви делови оригиналног пакета се постижу следеће функције:

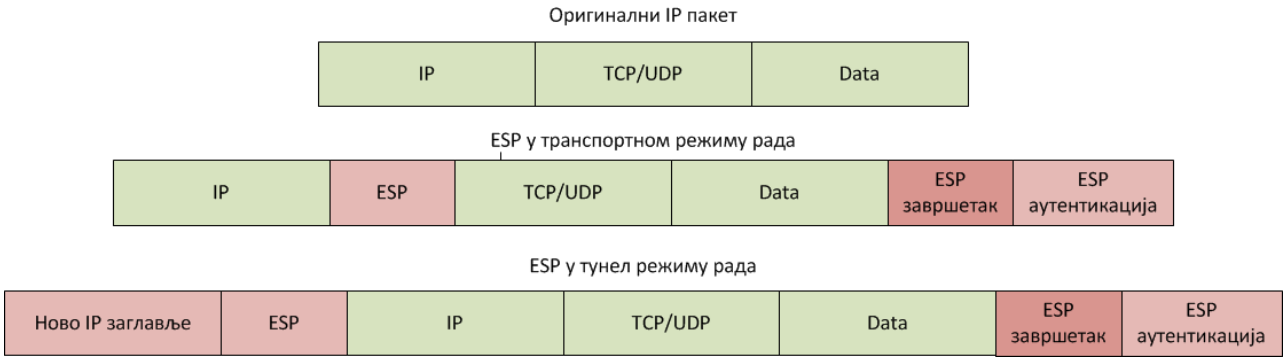
- Обезбеђивање интегритета пакета и података у њему (у хеш функцију улазе подаци у пакету).
- Обезбеђивање аутентичности порекла пакета (у хеш функцију улазе IP адресе).
- Заштита од напада понављањем (у хеш функцију улазе секвенцијални бројева пакета).

3.3.2.2. Encapsulating Security Payload (ESP)

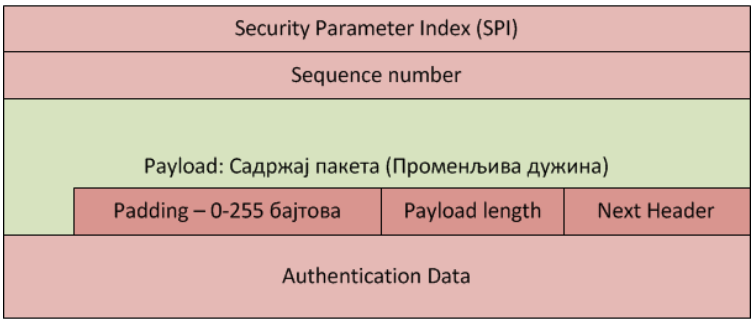
Encapsulating Security Payload (ESP) је механизам који пре свега служи за заштиту поверљивости података који се шаљу мрежом њиховом енкрипцијом, али може да пружи и заштиту интегритета и аутентичности порекла, као и још неке сигурносне функције, што ће бити објашњено у наставку текста [3.21]. ESP као и AH може да се користи у транспорт и тунел режиму рада и ESP заглавље се смешта на исти начин као и AH у овим режимима. Са друге стране ESP има нешто сложенију структуру јер поред ESP заглавља постоји и ESP завршетак (енг. *trailer*) који се додаје на крај пакета, што је показано на слици 3.36. У ESP пакетима је енкриптовано све између ESP заглавља и ESP аутентикације.

Елементи ESP заглавља су приказани на слици 3.37. Као и у AH, постоје поља за SPI и секвенцијални број пакета са истом улогом. Такође, и поља *NextHeader* и *Payload Length* постоје и у ESP, с тим да су у овом протоколу смештена у ESP завршетак. Поље које је специфично за ESP је *Padding*. Ово поље служи за то да се садржај пакета, који је променљиве дужине допуни до дужине која је једнака целом броју блокова симетричног алгоритма којим се криптује садржај пакета. На крају пакета се налази ESP аутентикација која представља излаз хеш функције на чији улаз долази све од ESP заглавља до краја ESP завршетка и у тунел и у транспортном режиму рада.

24 Нека поља у IP заглављу се мењају током проласка кроз мрежу, попут TTL, поља за фрагментацију ако до ње дође, или DSCP док су друга поља попут IP адреса, поља за протокол енкапсулиран у IP непроменљива. Пошто IP адресе улазе у хеш функцију, то значи да AH протокол пружа верификацију порекла пакета, односно тога да IP адресе нису мењане током проласка кроз мрежу.



Слика 3.36 ESP енкапсулација у транспортном и тунел режиму рада



Слика 3.37 ESP заглавље

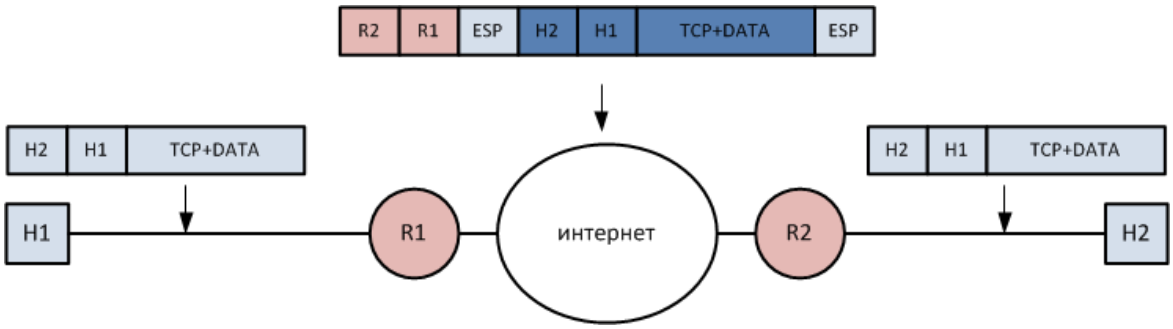
Овако формирани пакети пружају следеће сигурносне механизме:

- заштиту поверљивости података енкрипцијом,
- заштиту интегритета података аутентикационом хеш функцијом,
- аутентикацију порекла, али само када се користи тунел режим рада, када на улаз хеш функције долазе и IP адресе оригиналног пакета,
- Заштиту од напада понављањем као код AH,
- Заштиту од анализе токова у мрежи, што ће бити детаљније објашњено у наставку.

IPsec спецификација дефинише да је обавезно коришћење тунел режима рада када се сигурносне асоцијације успостављају између LAN мрежа. Један пример овога је показан на слици 3.38.

У овој мрежи два рачунара са адресама H1 и H2 размењују пакете (а на свакој од тих локација у локалним мрежама може да буде још рачунара који би на исти начин међусобно комуницирали). Између рутера R1 и R2 који су на излазу локалних мрежа на обе стране су успостављене сигурносне асоцијације са ESP протоколом у тунел режиму рада. На слици су приказане поједностављене слике заглавља у пакетима са назначеним IP адресама на различитим тачкама у мрежи. Оно што је очигледно је то да посматрач који се налази у некој тачки на интернету и који би снимао саобраћај између ове две мреже може да види пакете у

чијем заглављу су видљиве само IP адресе рутера R1 и R2, али не може да види IP адресе рачунара који заиста међусобно комуницирају јер су оне криптоване. Који год рачунари да комуницирају на ове две локације, на интернету ће видљиве бити само адресе рутера R1 и R2. Посматрач не може да процени каква је интерна структура локалних мрежа, какви су профили комуникације између појединачних рачунара нити колико их је што значајно отежава почетне фазе у нападу на информатичке ресурсе ових мрежа. Пример реализације IPsec VPN мреже у којој се користи ESP у тунел режиму рада је дат у поглављу 7.7.



Слика 3.38 Енкапсулација пакета кроз ESP SA у тунел режиму рада

3.3.2.3. Internet Key Exchange (IKE)

И за AH који користи HMAC верзије хеш алгоритама и за ESP који користи симетричне алгоритме за криптовање је потребно да се на обе стране сигурносне асоцијације нађе исти кључ. За ову потребу је предвиђен протокол за размену кључева IKE (енг. *Internet Key Exchange*). У време настанка IPsec поред IKE протокола постојао је и протокол ISAKMP (енг. *Internet Security Association and Key Management Protocol*). ISAKMP је био замишљен као протокол који дефинише формате основних порука чијом комбинацијом је могуће креирати различите шеме размена кључева, од којих је IKE био први. Временом се показало да се нове шеме за размену кључева нису развијале, тако да је у другој ревизији пројекта ISAKMP укинут и за потребе IPsec је остао само један протокол за размену кључева: IKE.

IKE је заснован на Дифи-Хелман механизму за креирање кључева, уз додатну заштиту од напада понављањем и *man-in-the-middle* напада. Постоје две основне верзије IKE размене кључева (v1[3.22] и v2[3.23]), а у оквиру сваке од верзија постоје различите варијанте провере идентитета удаљене стране са којом се кључеви размењују:

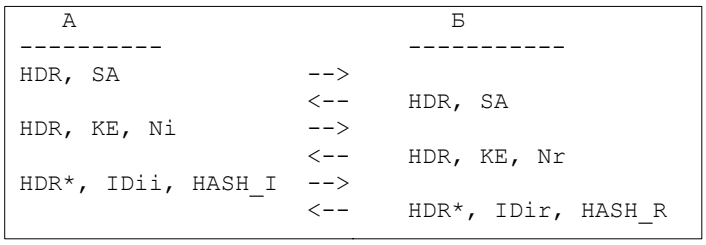
- провера идентитета помоћу унапред подељеног симетричног кључа. Овим кључем се криптује неки идентификатор који служи за проверу идентитета (IP адреса, мејл адреса, корисничко име и слично). Само уколико учесници у размени кључева поседују исти кључ (и не поседује га нападач) моћи ће на исправан начин да енкриптују и декруптују тај идентификатор и провере аутентичност друге стране у размени. Овакав начин размене је погодан за мањи и статичан (слабо променљив)

δрој учесника у виртуелној приватној мрежи. За веће и динамичне групе постоји проδлем ефикасне промене овог кључа.

- провера идентитета помоћу дигиталних потписа.
- провера идентитета помоћу дигиталних сертификата и PKI инфраструктуре.

Упркос томе што је формално доношењем друге верзије IKE протокола прва верзија стављена ван употреδе, још увек постоји много активних имплементација IKEv1. У наставку текста ће δити описан начин размене кључева са провером идентитета помоћу унапред подељеног симетричног кључа. Размена кључева се врши у две фазе:

- у првој фази се креирају кључеви којима ће се заштитити размена кључева у другој фази размене. Може да се реализује на два начина: *Main mode* и *Aggressive mode*. *Main mode* који се врши постепено у три размене (6 порука) ће δити приказан у наставку текста, док се у *Aggressive mode* начину исти криптографски материјал размењује у три поруке.
- у другој фази се креирају кључеви којима ће се вршити заштита IPsec пакета. Ова фаза се зове *QuickMode*.



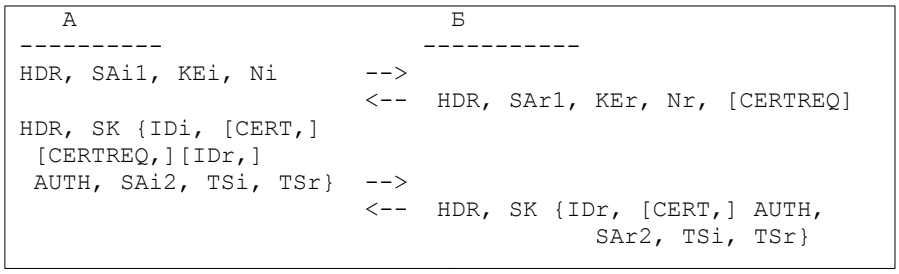
Слика 3.39: IKEv1 Main Mode са унапред подељеним кључевима

На слици 3.39 је приказан IKEv1 *Main Mode* са унапред подељеним кључевима (начин поделе ових кључева није предмет стандарда, једноставно се подразумева да су ти кључеви на неки сигуран начин подељени). У заглављу свих порука (HDR) су SPI вредности које ће имати сигурносна асоцијација када δуде креирана. У првом пару порука се размењују SA делови порука у којима ентитети који успостављају сигурносне асоцијације шаљу скупове протокола и алгоритама које подржавају и које предлажу за заштиту друге фазе (на многим уређајима се ова сигурносна асоцијација из историјских разлога назива и даље ISAKMP SA иако је ISAKMP протокол поништен каснијим ревизијама IKE протокола). Уколико постоји δар један пар предложених алгоритама које подржавају оδе стране, сигурносна асоцијација ће моћи да се успостави и прећи ће се на размену криптографског материјала за ове алгоритме. У трећој и четвртој поруци се размењују јавне Дифи-Хелман вредности (KE делови поруке) и случајне вредности Ni и Nr (сваки учесник креира своју случајну вредност) које служе за спречавање напада понављањем порука. Након ове две поруке оδе стране имају размењене јавне Дифи-Хелман вредности и могу да креирају исти кључ који ће се користити за заштиту наредних порука. Пета и шеста порука у првој фази размене су означене знаком * што значи да су те поруке криптоване кључем који је креиран након четврте поруке. У овим

порукама се шаљу подаци о идентитету (могу да буду различити: IP адресе, FQDN имена, корисничка имена итд.) криптовани унапред подељеним кључем чиме се верификује идентитет страна које размењују кључеве, као и хеш вредности које се рачунају на основу идентитета и претходно размењених случајних вредности чиме се спречавају напади понављањем. У другим шемама са провером идентитета помоћу дигиталних потписа или сертификата се у овим порукама налази тај други криптографски материјал који је потребан за проверу идентитета.

Слика 3.40: IKEv1 Quick Mode

У *Quick* моду (слика 3.40) се понавља слична шема размене кључева, једино што се врши у свега три поруке. SA делови поруке носе овај пут предлоге скупа алгоритама које учесници могу да користе за заштиту пакета, а KE делови поруке нове Дифи-Хелман вредности из којих ће се креирати нови кључ којим ће се криптовати IPsec пакети. IKE стандард дефинише псеудо-просте бројеве величине 768 и 1024 бита који могу да се користе у Дифи-Хелман алгоритму, а каснији стандард доноси псеудо-просте бројеве до 8192 бита [3.24] чиме може да се постигне адекватна заштита размене кључева данас.



Слика 3.41: Прва фаза IKEv2

IKEv2 размена доноси неке разлике у терминологији и садржају порука (Слика 3.41). Задржан је принцип у којем постоје две фазе у размени у којима се креирају два различита кључа, али се оне сада називају иницијална размена којом се дефинише заштита друге фазе и размена сигурносне асоцијације детета (енг. *Child SA*) којом се креирају кључеви за заштиту IPsec пакета. У прве две поруке иницијалне размене се врши иницијализација IKE сигурносне асоцијације (раније ISAKMP SA) којом се договарају параметри друге размене и врши размена јавних Дифи-Хелман вредности и случајних бројева. Већ након првог пара порука могу да се креирају сесијски кључеви којима се криптују наредне поруке размене. Трећа и четврта порука су криптоване сесисјким кључем (SK), а у њима се поред информација које омогућавају сигурну проверу идентитета се налазе и селектори саобраћаја

(TS – енг. *Traffic Selectors*) којима се размењују информације о томе на које пакете ће се примењивати сигурносне асоцијације (информације из SPD базе).

Друга фаза у којој се креира сигурносна асоцијација – дете је приказана на слици 3.42 и врло је слична првој верзији IKE уз додатак селектора саобраћаја.

Слика 3.42: Друга фаза IKEv2

3.3.2.4. IPsec виртуелне приватне мреже са више чворова

Једна од најчешћих примена IPsec је за креирање скупа тунела којима преко неке дељене инфраструктуре (нпр. интернета) једна организација жели да повеже своје испоставе у једну јединствену мрежну инфраструктуру (као на слици 3.13 - десно). IPsec стандард прописује да у овим ситуацијама мора да се користи у тунел режиму рада, а ти тунели се успостављају између рутера који су на излазу свих појединачних локација. Ово је вероватно најекономичнији начин за повезивање испостава организације зато што је на свакој локацији потребно обезбедити само везу ка интернету која је типично значајно јефтинија од посебних веза између локација које обезбеђују провајдери. Да би у таквој мрежи били доступни сви рачунарски ресурси на свим локацијама, а посебно у топологијама које нису потпун граф тунела (много чешћа је топологија звезде јер је једноставнија за одржавање и прати организацију фирме), потребно је да се руте ка мрежама које су на свакој од локација међусобно размене. Као што је у поглављу 3.1.6.1 показано у оваквој ситуацији је корисник сам тај ко треба да обезбеди рутирање кроз успостављене тунеле.

IPsec спецификација није до краја прецизна када је реч о подршци за рутирање кроз тунеле. Док је један део учесника у IETF IPsec радној групи сматрао да протоколи рутирања попут OSPF не могу да се успоставе кроз IPsec тунеле зато што захтевају пренос OSPF порука директно енкапсулираних у IP и то путем мултикаста, а IPsec сигурносне асоцијације по својој природи уникаст, други су сматрали да не постоји ни једна формална препрека да се кроз IPsec тунел пошаљу пакети протокола рутирања. На крају, имплементације IPsec на рутерима су биле такве да нису подржавале директно рутирање пакета протокола рутирања кроз IPsec тунеле. Да би се решио овај проблем прибегло се коришћењу GRE (енг. *Generic Routing Encapsulation*) протокола који служи за пренос информација протокола рутирања преко тунела остварених преко интернета. Између рутера који успостављају сигурносне асоцијације мора да се успостави још један - GRE тунел како би се омогућио и транспорт порука протокола рутирања. Како увођење још једног тунела и коришћења две енкапсулације значајно погоршава однос корисних и контролних информација у пакету (Слика 3.43), у овој ситуацији је препоручено да се користе транспортне сигурносне асоцијације које су у смислу заштите пакета због постојања GRE протокола еквивалентне IPsec тунел режиму рада (уз додато GRE заглавље).

Слика 3.43 Енкапсулација пакета протокола рутирања коришћењем IPsec и GRE тунела у тунел транспортном режиму рада

3.3.2.5. IPsec виртуелне приватне мреже за појединачне кориснике

Друга најчешћа примена IPsec је креирање тунела за појединачне кориснике који треба да се повежу на рачунарску мрежу своје фирме када се налазе ван ње (на службеном путу, код куће, итд.). У тој ситуацији се IPsec тунел успоставља од мобилног уређаја (лаптоп, таблет, телефон) појединачног корисника до неког сервера концентратора IPsec тунела у фирми (уређај на који се повезују сви мобилни корисници). На мобилном уређају се по стварању сигурносне асоцијације креира још један, виртуелни интерфејс са адресом коју додељује IPsec концентратор. Такође, концентратор треба да мобилном уређају пошаље и скуп рута из мреже фирме за које ће мобилни уређај слати пакете у креирану сигурносну асоцијацију.

```
user@ThinkPad-T420:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.1.1     0.0.0.0         UG    600    0        0 wlp3s0
link-local      *               255.255.0.0     U     1000   0        0 wlp3s0
192.168.1.0     *               255.255.255.0   U     600    0        0 wlp3s0
user@ThinkPad-T420:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.1.1     0.0.0.0         UG    600    0        0 wlp3s0
10.8.0.0        *               255.255.0.0     U     50     0        0 tun0
91.187.128.0    10.8.0.1        255.255.224.0   UG    50     0        0 tun0
147.91.0.0      10.8.0.1        255.255.0.0     UG    50     0        0 tun0
160.99.0.0      10.8.0.1        255.255.0.0     UG    50     0        0 tun0
link-local      *               255.255.0.0     U     1000   0        0 wlp3s0
192.168.1.0     *               255.255.255.0   U     600    0        0 wlp3s0
```

Слика 3.44: Интерна табела рутирања мобилног уређаја на којем је успостављена IPsec сигурносна асоцијација – пре подизања (горе), после подизања (доле)

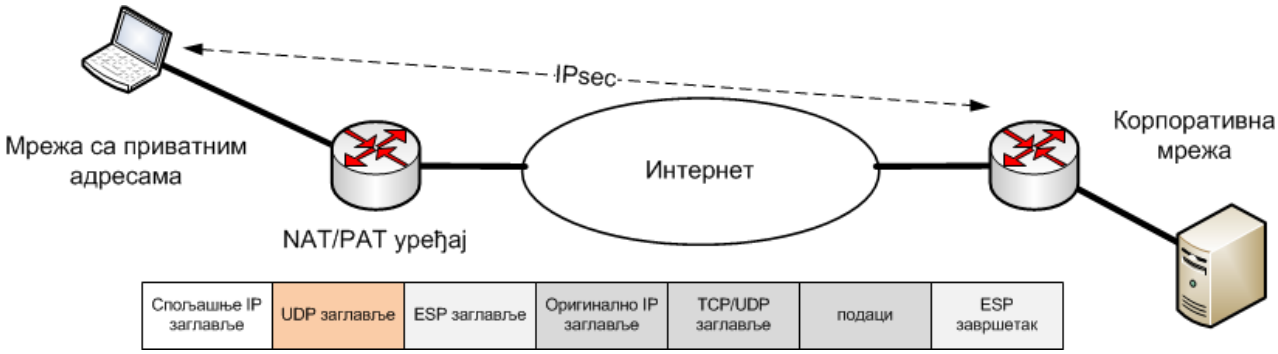
Ово је приказано на слици 3.44 где се виде табеле рутирања пре и после успостављања IPsec сигурносне асоцијације на једном рачунару. Након успостављања сигурносне асоцијације се види да је креиран нови интерфејс *tun0*, и да су додате четири нове руте за које ће пакети бити послати у криптовани тунел. У примеру на слици је конфигурисан тзв. *split tunneling* код ког се део пакета рутира кроз криптовани тунел, а сав остали ка интернету: *default* рута указује на физички интерфејс рачунара и ти пакети ће ићи мимо криптованог тунела. Оваква конфигурација се не препоручује јер би компромитацијом мобилног уређаја преко интернета нападач могао да добије кроз мобилни уређај приступ заштићеној корпоративној мрежи. У ситуацијама где се захтева строга заштита ресурса, IPsec концентратор шаље *default* руту

уместо појединачних рута, како би сав саобраћај од мобилног уређаја био послат у сигурносну асоцијацију.

Да би се ово остварило направљени су следећи додаци за IKE:

- XAuth којим је у IKE додата и провера идентитета појединачног корисника који се повезује на корпоративну мрежу и логовање његовог начина коришћења мреже за евентуално касније откривање понашања корисника (од када до када је био повезан путем сигурносне асоцијације, коју је адресу добио итд.).
- *Mode* конфигурација којом се кроз IKE протокол врши пренос рута и конфигурације тунела за мобилни уређај.
- Детекција постојања транслације адреса (NAT - енг. *Network Address Translation*) на путањи од мобилног уређаја до концентратора како би се реализовао механизам *NAT Traversal.*

Постојање транслације адреса на путањи од мобилног уређаја до концентратора, а посебно што транслација адреса најчешће подразумева и транслацију портова (енг. *Port Address Translation* - PAT) представља проблем за IPsec зато што када се користи ESP и у тунел и транспорт режиму рада заглавља протокола транспортног слоја су криптована, тиме и бројеви портова и нема могућности да се транслација адреса обави. Да би се овај проблем решио осмишљен је механизам NAT *Traversal* којим се у пакет додаје још једно додатно UDP заглавље пре ESP заглавља које није криптовано, те је тиме транслација адреса омогућена (Слика 3.45). У овом пакету када га шаље мобилни уређај у спољашњем IP заглављу су IP адресе концентратора и мобилног уређаја (приватна адреса добијена у локалној мрежи), а на интернету је уместо адресе мобилног уређаја транслирана адреса. У унутрашњем IP заглављу су адресе добијене *Mode* конфигурацијом.



Слика 3.45 NAT Traversal заглавље

Детекција транслације адреса се обавља кроз IKE тако што се у размену додаје и порука са резултатом хеш функције која се рачуна од IP адреса и бројева портова послатог пакета. Уколико је дошло до транслације адреса, пријемна страна неће израчунати исту хеш вредност која је послата и онда ће морати да се примењује NAT *Traversal* – да се дода UDP заглавље у пакет како је описано.

3.3.3. Заштита на транспортном слоју

Протоколи за заштиту на транспортном слоју су настали пре протокола за заштиту на мрежном слоју (IPsec). Прву верзију SSL протокола (енг. *Secure Socket Layer*) је направила компанија Netscape за потребе заштите HTTP саобраћаја и имплементацију HTTPS. Касније је овај начин заштите примењен на друге протоколе попут FTP, SMTP, POP3 итд. SSL је имао три верзије које су све формално повучене из употребе (верзија 1.0 никада није ни објављена, а верзије 2.0 и 3.0 су повучене 2011. и 2015. [3.25][3.26] и сматрају се за несигурне). Последња верзија SSL 3.0 је функционално готово једнака протоколу TLS 1.0 (енг. *Transport Layer Security*), а данас актуелне верзије TLS протокола су 1.1 и 1.2 које доносе нека унапређења у одбрани од напада и нове верзије криптографских алгоритама (дуже кључеве и дуже хеш функције) у односу на прву верзију.

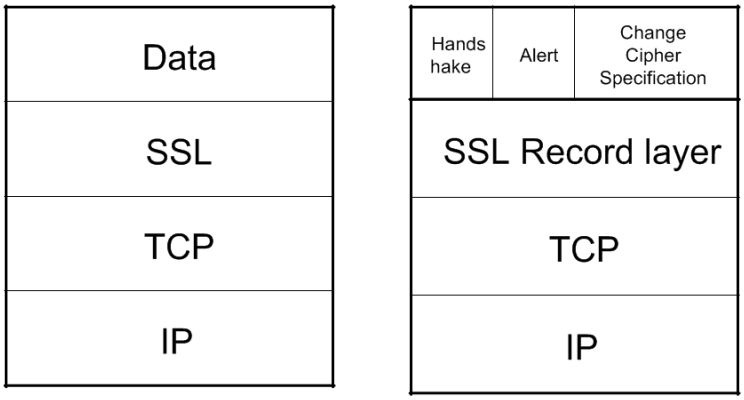
Данас (децембар 2017) је више од 50% садржаја који се преноси путем интернета криптовано помоћу HTTPS са тенденцијом сталног пораста, а око 63% свих сајтова користи HTTPS. Од тога око 90% сајтова користи неку од верзија TLS протокола, највише 1.0 док и даље има око 13,7% сајтова који користе SSL3.0 који се сматра за небезбедан и повучен је из употребе [3.27]. Управо ова широка распрострањеност TLS протокола у клијентским уређајима који имају подршку за HTTPS у прегледачима интернета и инсталиране све потребне софтверске библиотеке, као и релативно компликовано коришћење IPsec за појединачне кориснике је био мотив да се коришћење SSL прошири и на креирање виртуелних приватних мрежа за појединачне кориснике. Ово ће и бити описано у наставку текста.

3.3.3.1. TLS/SSL протоколи

TLS/SLL протоколи су протоколи транспортног слоја, који су енкапсулирани у TCP (Слика 3.46), а састоје се из два слоја:

- SSL *Record Layer* који представља слој енкапсулиран у TCP и који омогућава компресију, енкрипцију, фрагментацију, и заштиту интегритета порука апликативног слоја
- SSL *Handshake* слој који се састоји од:
  - Протокола за размену кључева (*Handshake*)
  - *Alert* протокола којим се сигнализирају грешке у раду протокола и
  - *Change Cipher Specification* протокола којим се сигнализира прелазак на криптовану комуникацију.

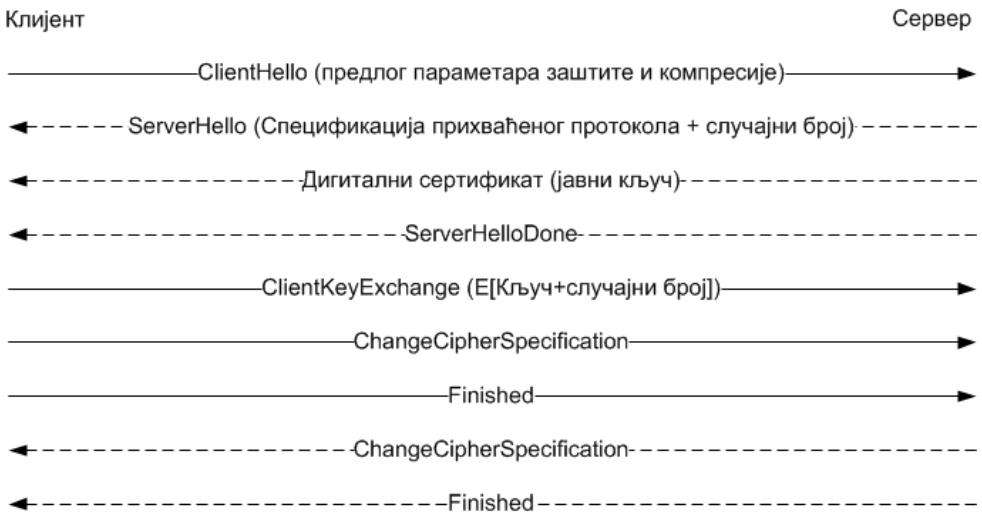
Протоколи за енкрипцију пакета могу да буду сви данас често коришћени алгоритми попут AES (до кључева дужине 256 бита), IDEA, DES итд.



Слика 3.46 Позиција SSL у протоколском стеку

3.3.3.2. Размена кључева код TLS/SSL

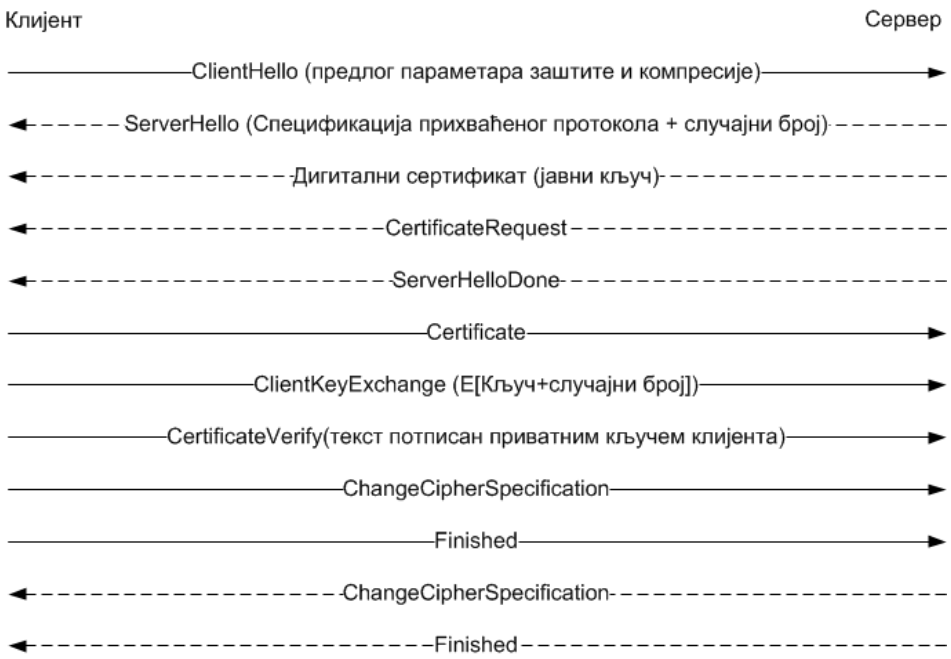
Постоји велики број различитих шема размене кључева код SSL/TLS, од различитих варијанти Дифи-Хелман размене, до размена које користе RSA алгоритам и дигиталне сертификате за проверу идентитета. У наставку ће бити показана два начина за размену који користе асиметричне криптографске алгоритме и дигиталне сертификате. Први начин подразумева коришћење дигиталног сертификата само на серверској страни (Слика 3.47), а други начин коришћење дигиталног сертификата на обе стране комуникације (Слика 3.48). Први се данас користи за приступ највећем броју сајтова који користе HTTPS (попут www.google.com), а други за приступ оним сајтовима који захтевају стриктну проверу идентитета клијента (нпр. приступ сајтовима банака када постоји могућност плаћања преко интернета).



Слика 3.47 SSL размена кључева када се користе само серверски сертификати

Размена кључева када се користе само серверски сертификати почиње *ClientHello* поруком којом клијент серверу шаље скуп свих криптографских механизама које подржава. Од тог скупа сервер одабира једну групу механизама и *ServerHello* поруком их шаље клијенту заједно са случајним бројем који служи за спречавање напада понављањем. Сервер такође

шаље свој дигитални сертификат у којем је јавни кључ сервера, а може да се пошаље и цео ланац сертификата до дигиталног сертификата неког од добро познатих сертификационих тела која су уписана у прегледаче или оперативне системе. Клијент по провери сертификата сервера криптује јавним кључем сервера тајни сесијски кључ којим ће се криптовати пакети у оквиру TLS/SSL сесије и шаље га серверу заједно са случајним бројем. Након тога све следеће поруке клијента ће бити криптоване овим сесијским кључем. Сервер декриптује сесијски кључ, проверава декриптовану случајну вредност и уколико нема грешака све своје следеће поруке криптује сесијским кључем. Овим је размена кључа завршена.



Слика 3.48 SSL размена кључева када се користе и серверски и клијентски сертификати

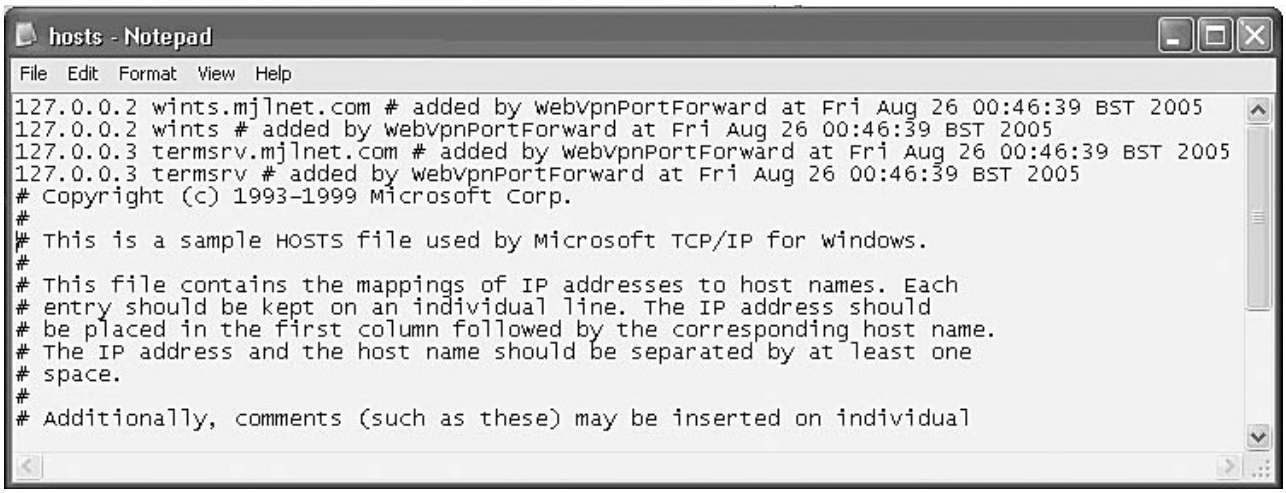
Размена кључева када се користе сертификати на обе стране комуникације је слична претходној. Разлика је у провери идентитета клијента која се врши тако што уз *ServerHello* поруку сервер шаље и захтев за клијентским сертификатом. На тај захтев клијент шаље свој дигитални сертификат у којем је клијентов јавни кључ, а да би доказао власништво над сертификатом додатно мора да криптује све претходно послате поруке у размени својим приватним кључем (*CertificateVerify* порука). Уколико сервер декриптовањем ове поруке закључи да је клијент аутентичан, размена кључа је завршена и успоставља се енкриптовани ток пакета.

3.3.3.3. TLS/SSL виртуелне приватне мреже

TLS/SSL протоколи су релативно скоро постали популарни као решење за креирање виртуелних приватних мрежа за појединачне мобилне уређаје и кориснике. Неколико је мотива за увођење овог начина за креирање криптованих веза за мобилне кориснике [3.28]:

- TLS/SSL не криптују бројеве портова у заглављу транспортног слоја, те нема проблема са проласком пакета кроз NAT.
- Све библиотеке и готово сав софтвер потребан за креирање криптованих тунела већ постоји на клијентским уређајима јер га стандардно користе прегледачи интернета, мејл клијенти и друге апликације.
- Могуће је успостављање VPN везе без посебне клијентске апликације, коришћењем прегледача интернета, што је значајно лакше за коришћење појединцима који нису технички образовани, а имају потребу за заштићеном комуникацијом. Постоји генерално мишљење да је коришћење IPsec клијентских апликација компликовано са сложеним подешавањем.

Креирање VPN тунела коришћењем прегледача интернета подразумева нешто сложенију конфигурацију на страни концентратора где је потребно одредити којим све ресурсима (рачунарима и серверима и апликацијама на њима) сваки појединачни корисник може да приступи. На пример, ако се жели да се мобилном кориснику дозволи да користи приступ рачунару у корпоративној мрежи путем удаљеног десктопа (енг. *Remote Desktop*), то је потребно експлицитно конфигурисати на концентратору за сваког појединачног корисника тако што се у оквиру конфигурације наводи IP адреса уређаја у корпоративној мрежи коме тај корисник може да приступи преко удаљеног десктопа (нпр. 10.1.2.3), као и број порта (стандардно 3389).

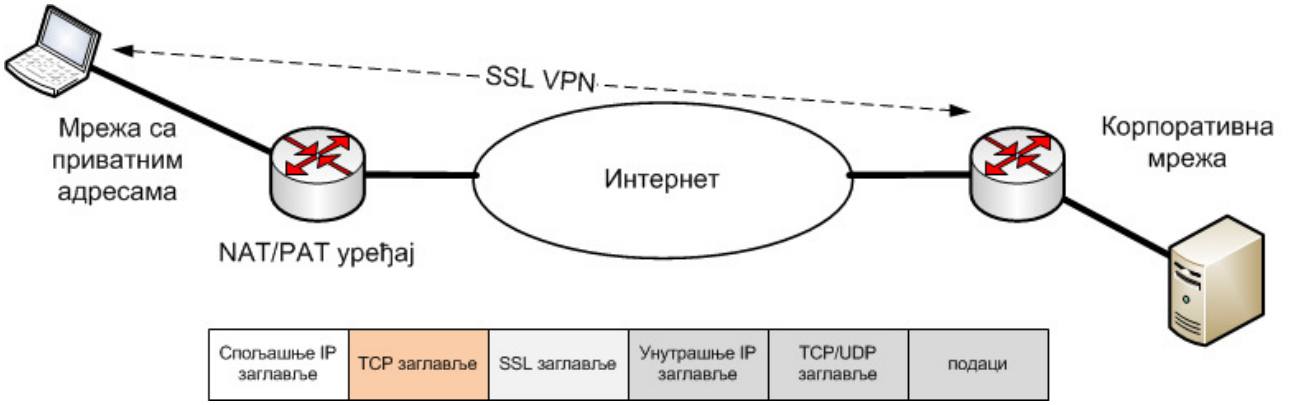


Слика 3.49 Пример модификованоī hosts фајла након успостављања HTTPS сесије [3.28]

Корисник који је путем свог прегледача остварио сигурну HTTPS везу до концентратора може да (обично кроз Java апликацију на веб страни) стартује клијентску апликацију за удаљени десктоп и то тако што затражи повезивање на адресу 127.0.0.1 (*loopback* адреса, адреса самог уређаја), по порту који је дефинисан за апликацију (3389). За другу апликацију би се користила адреса 127.0.0.2, за трећу 127.0.0.3, итд. Када то уради, пакети који су упућени на адресу самог мобилног уређаја (127.0.0.1) на порт 3389 бивају преусмерени у HTTPS сесију, и то тако што за дестинациону адресу добијају адресу 10.1.2.3 чиме пролазе

до жељене дестинације. Овакав механизам преусмеравања се зове *port forwarding*. *Port forwarding* се реализује тако што се приликом успостављања HTTPS сесије на клијентском рачунару невидљиво за корисика модификује *hosts*[25] фајл тако да се за сваку *loopback* адресу направи мапирање у стварну адресу рачунара у корпоративној мрежи коме треба приступити као што је показано на слици 3.49.

Изглед пакета у оквиру овако реализоване виртуелне приватне мреже приказан је на слици 3.50. У спољашњем IP заглављу се налазе јавна адреса концентратора видљива преко интернета и адреса мобилног уређаја коју је добио у локалној мрежи, а која се приликом проласка транслира без проблема на NAT/PAT уређајима у јавне адресе зато што је криптовани део пакета енкапсулиран у SSL, а бројеви портова су некриптовани. Унутрашње IP заглавље садржи адресу дестинационог уређаја у корпоративној мрежи коме се жели приступ (нпр. 10.1.2.3 из горњег примера) и *loopback* адресу за тај уређај коју концентратор транслира у адресу из локалног опсега корпоративне мреже.



Слика 3.50 Пролазак SSL VPN пакета кроз уређаје који раде NAT/PAT

Уколико се жели транспарентан приступ свим ресурсима мреже преко заштићене SSL везе на начин као код IPsec, а не само унапред одређеним рачунарима и сервисима као што је горе описано, онда мора да се користи клијентска апликација готово идентичне функционалности као код IPsec. Том приликом се креирају тунел интерфејси на мобилним уређајима и дефинишу се руте за које се пакети шаљу у криптовани мрежни тунел.

25 *hosts* фајл је фајл који постоји у свим стандардним оперативним системима и који служи за статичко мапирање IP адреса у симболичка имена (попут DNS сервера). У њему се стандардно налази свега неколико редова са мапирањима локалних имена, као и *loopback* адреса.

3.4. Литература

[3.1] J. Rexford, C. Dovrolis, Future Internet Architecture: Clean-Slate Versus Evolutionary Research, Communications of the ACM, September 2010, Vol. 53 No. 9, Pages 36-40, DOI: 10.1145/1810891.1810906

[3.2] E. Rosen, A. Viswanathan, R. Callon, Multiprotocol Label Switching Architecture, IETF RFC 3031, January 2001., https://tools.ietf.org/html/rfc3031

[3.3] L. Andersson, I. Minei, B. Thomas, LDP Specification, IETF RFC 5036, October 2007, https://tools.ietf.org/html/rfc5036

[3.4] L. De Ghein, MPLS Fundamentals, Cisco Press, 2007, Indianapolis, ISBN: 1-58705-197-4

[3.5] E. Rosen, Y. Rekhter, BGP/MPLS IP Virtual Private Networks (VPNs), IETF RFC 4364, February 2006, https://tools.ietf.org/html/rfc4364

[3.6] K. Kompella, Y. Rekhter, Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling, IETF RFC 4761, January 2007, https://tools.ietf.org/html/rfc4761

[3.7] M. Lasserre, V. Kompella, Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling, IETF RFC 4762, January 2007, https://tools.ietf.org/html/rfc4762

[3.8] S. Bryant, G. Swallow, L. Martini, D. McPherson, Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN, IETF RFC 4385, February 2006, https://tools.ietf.org/html/rfc4385

[3.9] E. M. Arkin, J. S. Mitchell, and C. D. Piatko. Bicriteria shortest path problems in the plane. In Proc. 3rd Canad. Conf. Comput. Geom, pages 153–156, 1991.

[3.10] https://tools.ietf.org/html/rfc3785

[3.11] Kreutz, D., Ramos, F. M. V., Verissimo, P., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2014). Software-Defined Networking: A Comprehensive Survey. Retrieved from http://arxiv.org/abs/1406.0440

[3.12] D. Katz, K. Kompella, D. Yeung, Traffic Engineering (TE) Extensions to OSPF version 2, IETF RFC 3630, September 2003, https://tools.ietf.org/html/rfc3630

[3.13] T. Li, H. Smit, IS-IS Extensions for Traffic Engineering, IETF RFC 5305, October 2008, https://tools.ietf.org/html/rfc5305

[3.14] R. Coltun, The OSPF Opaque LSA Option, IETF RFC 2370, July 1998, https://tools.ietf.org/html/rfc2370

[3.15] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, RSVP-TE: Extensions to RSVP for LSP Tunnels, IETF RFC 3209, December 2001, https://tools.ietf.org/html/rfc3209

[3.16] C. Perkins, IP Mobility Support for IPv4, IETF RFC 3344, August 2002, https://tools.ietf.org/html/rfc3344

[3.17] Iannone, L., Saucez, D., & Bonaventure, O. (2010). Implementing the Locator/ID Separation Protocol: Design and experience. *Computer Networks*, *55*(4), 948–958. doi:10.1016/j.comnet.2010.12.017

[3.18] H. Orman, P. Hoffman, Determining Strengths For Public Key Used for Exchanging Symmetric keys, IETF RFC 3766, April 2004. https://tools.ietf.org/html/rfc3766

[3.19] S. Kent, K. Seo, Security Architecture for the Internet Protocol, IETF RFC 4301, December 2005, http://tools.ietf.org/html/rfc4301

[3.20] S. Kent, IP Authentication Header, IETF RFC 4302, December 2005., https://tools.ietf.org/html/rfc4302

[3.21] S. Kent, R. Atkinson, IP Encapsulating Security Payload (ESP), IETF RFC 2406, November 1998., https://tools.ietf.org/html/rfc2406

[3.22] D. Harkins, D. Carrel, The Internet Key Exchange (IKE), IETF RFC 2409, November 1998, https://tools.ietf.org/html/rfc2409

[3.23] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, Internet Key Exchange Protocol Version 2 (IKEv2), IETF RFC 5996, https://tools.ietf.org/search/rfc5996

[3.24] T. Kivinen, M. Kojo, More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE), IETF RFC 3526, May 2003, https://www.ietf.org/rfc/rfc3526.txt

[3.25] S. Turner, T. Polk, Prohibiting Secure Sockets Layer (SSL) Version 2.0, IETF RFC 6176, March 2011, https://tools.ietf.org/html/rfc6176

[3.26] R. Barnes, M. Thomson, A. Pironti, A. Langley, Deprecating Secure Sockets Layer Version 3.0, IETF RFC 7568, June 2015, https://tools.ietf.org/html/rfc7568

[3.27] SSL Pulse, https://www.ssllabs.com/ssl-pulse/, приступљено 7.12.2017.

[3.28] M. Lewis, Comparing, Designing, and Deploying VPNs, Cisco Press, April 2006, ISBN-10: 1-58705-179-6

4. Управљање рачунарским мрежама

Управљање рачунарским мрежама је скуп активности којима се остварују жељени начин и жељени циљеви рада мрежне инфраструктуре, а то је пре свега пружање поузданих услуга корисницима мреже. Постоји више модела који теже да систематично обухвате и опишу скуп свих активности које се подразумевају под управљањем рачунарским мрежама. Један од најстаријих таквих модела, али који се и даље користи као референтан је тзв. FCAPS модел [4.1] који је почетком осамдесетих година 20. века донела организација ISO[26]. Овај модел дефинише 5 основних категорија које је потребно испунити за потпуно управљање мрежном инфраструктуром, а чија имена чине акроним модела:

- *Fault* – активности на обради и отклањању грешака које настану током рада мреже. Најчешће укључују активности на прикупљању података о раду мреже како би се препознале и открите грешке, активности на конфигурисању како би се извршиле исправке и касније бележење тога шта је урађено како би се формирала база знања и искустава.
- *Configuration* – активности на конфигурисању мреже, мрежних и других елемената како би остваривала постављене циљеве.
- *Accounting* – активности на прикуљању информација о раду мреже или услуга које се пружају помоћу мреже.
- *Performance* – активности на обезбеђивању рада мреже који задовољава очекивања корисника (нпр. брзина одзива, квалитет примљених информација). Ове активности као и оне на отклањању грешака често укључују активности на прикупљању података о раду мреже како би се препознали и открили проблеми перформанси пружених услуга и активности на конфигурисању у циљу извршавања потребних исправки.

26 ISO – *International Organization for Standardization* – Међународна организација за стандардизацију

- *Security* – активности на обезбеђењу информационе безбедности саме инфраструктуре и података који се преносе мрежом

FCAPS модел даје преглед врста активности на управљању мрежом, али не даје детаљније смернице о томе на који начин би требало да се било која од ових активности спроведе. Такође, у време настанка овог модела главна пажња је била на активностима које треба вршити пре свега над мрежним уређајима како би мрежа исправно функционисала. Временом се ова пажња померила са управљања мрежним уређајима на управљање услугама које се пружају посредством мрежа и аутоматизацију процеса управљања мрежама која подразумева израду великог броја апликација које подржавају пружање услуга посредством мреже [4.2] [4.3]. Ова промена се препознаје највише кроз активности *TeleManagement Foruma*[27] (TMF) који тежи да олакша и стандардизује различите аспекте израде и имплементација апликација за управљање рачунарским мрежама, што ће и бити описано у наставку текста.

4.1. Класична архитектура система за управљање рачунарским мрежама

На слици 4.1 је приказана класична архитектура софтверских система који се користе за управљање рачунарским мрежама, која је данас још увек најзаступљенија, а код које су мрежни уређаји затворени системи којима исти произвођач обезбеђује хардвер, оперативни систем и апликативни софтвер (софтвер за различите протоколе и механизме)[28].

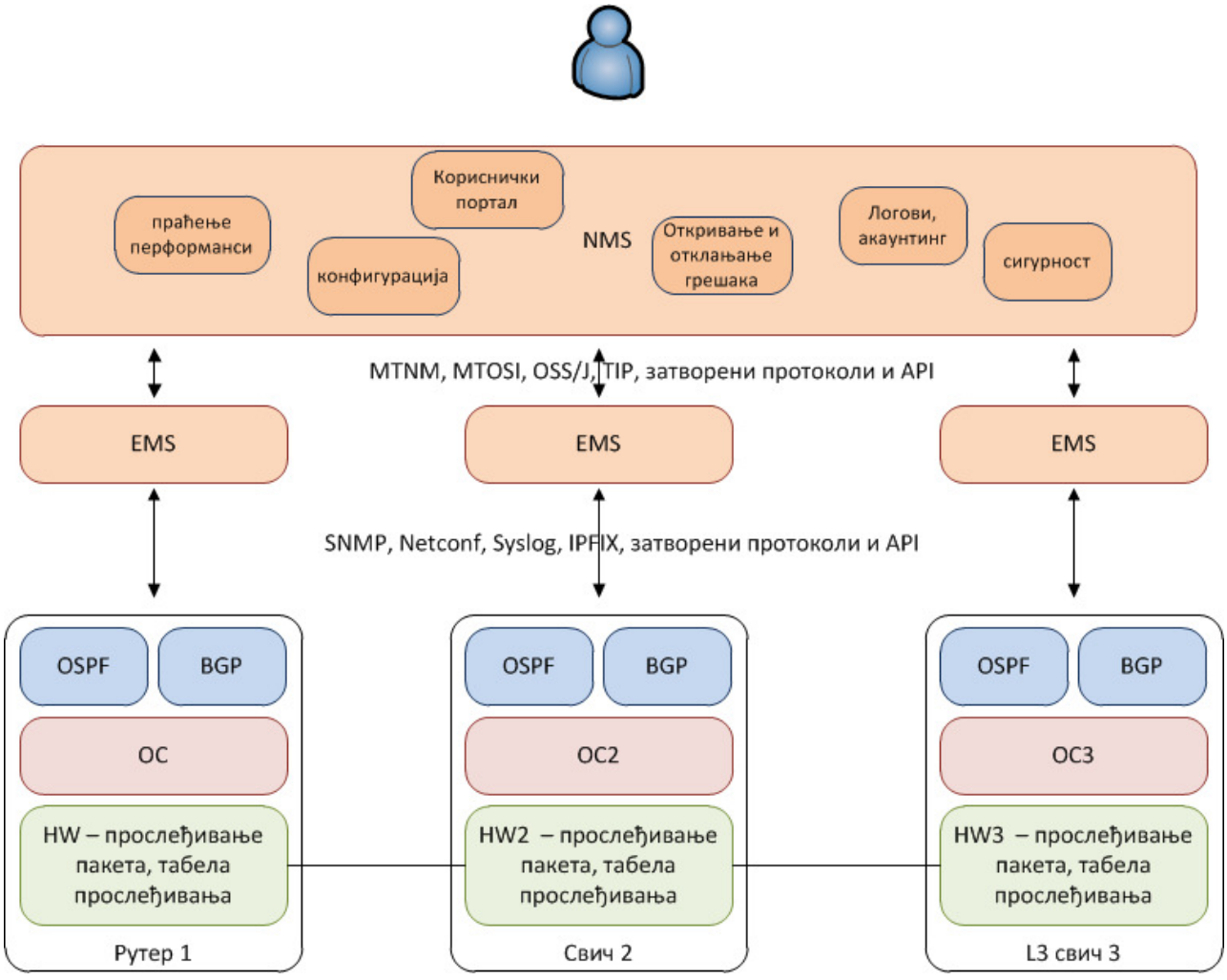
У овој архитектури произвођач мрежних уређаја (елемената) најчешће производи и системе за управљање мрежним уређајима (тзв. *Element Management System* – EMS). Ови системи обично садрже инвентаре свих уређаја, могу да аутоматски одреде топологију мреже и имају механизме који аутоматизују операције конфигурисања и надгледња мреже, али ограничено само на производе тог произвођача. EMS системи преко тзв. *southbound* интерфејса комуницирају са мрежним уређајима. Протоколи који се користе су најчешће затворени протоколи произвођача (тзв. *proprietary* протоколи), али се користе и стандардни протоколи попут SNMP, Netonf, Syslog, IPFIX и других који ће бити описани у овом поглављу.

Са друге стране EMS системи преко тзв. *northbound* интерфејса и API-ја комуницирају са апликацијама за управљање рачунарским мрежама и услугама. Управо у стандардизацији ових интерфејса и интерфејса између апликација управљачког слоја је највећи допринос дао *TeleManagement Forum*.

27 Организација која окупља највеће произвођаче мрежне опреме, софтвера за управљање мрежама и кориснике мрежне опреме (провајдере) - https://www.tmforum.org/

28 У последњих неколико година се појавио концепт Софтверски дефинисаних мрежа код којих хардвер са једне и оперативни систем и апликативни софтвер мрежних уређаја више не потичу нужно од истог произвођача, а протокол за комуникацију између њих је OpenFlow.

У наставку текста највећа пажња ће бити посвећена стандардним и често коришћенмим протоколима и механизмима који се користе за комуникацију са мрежним уређајима. Они најгрубље могу да се поделе у протоколе чија је главна улога прикупљање информација (енг. a*ccounting*) и протоколе чија је главна улога конфигурисање уређаја (енг. *configuration*), док се преостале управљачке активности: управљања грешкама и перформансама, најчешће своде на рад оператера – људи који тумаче прикупљене резултате и реагују.



Слика 4.1 Класична архитектура система за управљање рачунарским мрежама

У последње време и у те две области почињу да се појављују алати који аутоматизују процесе детекције и отклањања проблема и деградације перформансами. У Табели 4.1 дат је преглед данас најчешће коришћених протокола и механизама који ће бити обрађени у наставку текста, а према улози у процесима управљања мрежама. Сви наведени протоколи се најчешће користе заједно јер се међусобно допуњују што ће и бити показано.

Протокол/механизам	Прикупљање података и провера статуса	Конфигурисање уређаја
Приступ преко командне линије - CLI	+	+
SNMP	+	**(+)**[29]
NetFlow/IPFIX	+	
Syslog	+	
Netconf		+
YANG		+

Таβела 4.1. Најчешће коришћени ӣроӣоколи и механизми уӣрављања рачунарским мрежама

4.2. Управљање из командне линије

Управљање мрежним уређајима из командне линије (енг. *Command Line Interface* - CLI) је један од класичних начина за конфигурисање и увид у статус уређаја. Приступ уређајима се врши преко посеβних портова за управљање ако постоје (конзолни порт) или преко мреже *telnet* или SSH протоколима. Управљање из командне линије је често оβавезан начин за почетну конфигурацију јер се уређаји оβично испоручују βез икаквих конфигурација и самим тим βез могућности приступа преко мреже. Преко командне линије је могуће како конфигурисати β ило какво понашање уређаја, тако и прочитати статусе свих његових елемената или протокола. Начин рада са мрежним уређајима преко командне линије је описан у практичном делу овог курса кроз примере конфигурисања различитих технологија.

Кључни проβлеми управљања из командне линије су:

- некомпатиβилност синтакси команди и конфигурационих фајлова код различитих произвођача, који не показују спремност да унификују интерфејсе, јер навикнутост и специјализација корисника за једну врсту интерфејса представља један од начина везивања корисника опреме за произвођача (тзв. *vendor lock-in*). Ово је посеβно проβлем у мрежама у којима има опреме различитих произвођача.
- проβлем β рзине реакције, синхронизације и аутоматизације процеса надгледања и конфигурисања у мрежама са великим βројем уређаја пошто се уређаји конфигуришу појединачно.
- тешко континуирано праћење перформанси рада уређаја. Команде којима се доβија увид у статус појединих делова уређаја дају тренутне вредности параметара, али је тешко испратити њихове промене или трендове у дужим временским интервалима.

29 SNMP протокол је оригинално β ио замишљен тако да може да оβавља и прикупљање података са уређаја и њихово конфигурисање, али се временом показало у пракси да се претежно користи само за пасивно прикупљање података. Разлози за ово су дати ниже у тексту.

4.3. SNMP протокол

Један од најчешће коришћених протокола за управљање рачунарским мрежама је SNMP (енг. *Simple Network Management Protocol*) [4.4]. Овај протокол, настао 1990. године, у време финализације данашњег начина рада рачунарских мрежа и интернета је и данас актуелан. Иако је замишљен као протокол који би подржао и конфигурацију и надгледање рада мрежних уређаја, временом се искристалисало да се овај протокол користи готово искључиво за надгледање рада мрежних уређаја. Такође, временом је његово коришћење генерализовано и није се ограничило само на мрежне уређаје, већ се проширило на готово комплетну информатичку инфраструктуру: сервере, радне станице, различите софтверске производе (веб, мејл и други сервери) и елементе дата центара као што су уређаји за непрекидно напајање.

SNMP је комуникациони протокол који се успоставља између два ентитета: менаџмент станице са које се врши управљање мрежним уређајима и менаџмент агента који представља софтвер који је покренут на уређају којим се управља. Типично једна менаџмент станица комуницира са великим бројем агената на уређајима које прати.

SNMP користи UDP као транспортни протокол. Подразумеване вредности портова за SNMP су 161 и 162 за слање команди и примање тзв. трепова[30] респективно о којима ће бити више речи у наставку текста.

4.3.1. Организација MIB базе

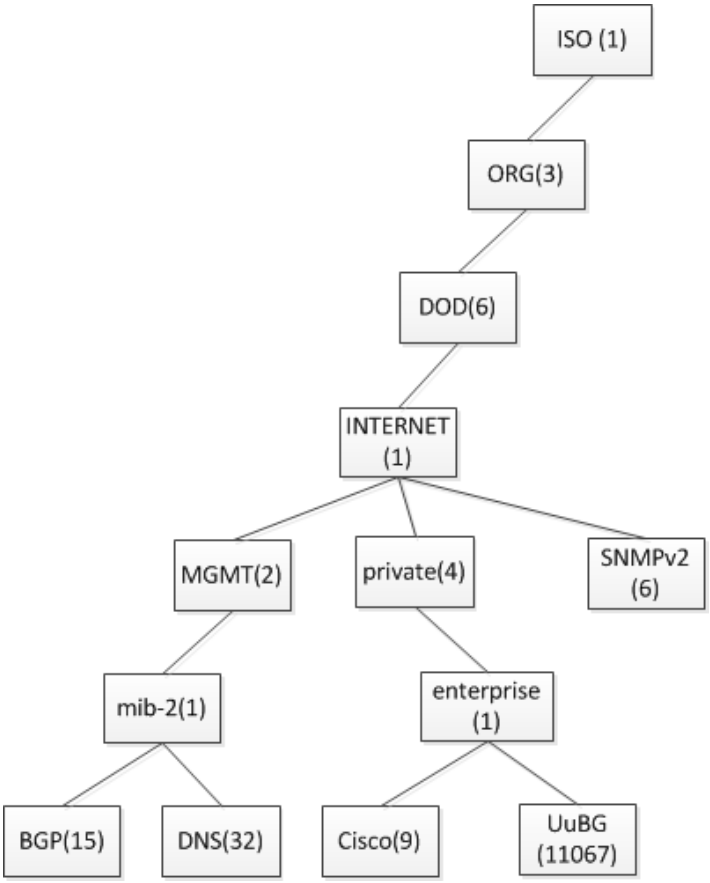
Кључни део SNMP архитектуре је MIB (енг. *Management Information Base*), база у коју су уписани сви објекти (променљиве) који могу да се очитавају са уређаја чији се рад прати и из којих агент чита податке или их у њих уписује.

MIB база је организована на хијерархијски начин, у форми графа стабла. Сваки објекат у MIB бази има свој идентификатор који се зове OID (енг. *Object Identifier*). Сваки чвор у графу MIB базе има свој број и текстуално име, а сваки OID је представљен као низ бројева (односно имена) одвојених тачком којим се од корена стабла долази до датог објекта. У време настанка SNMP протокола било је претпостављено да ће различита стандардизациона тела имати своја посебна стабла, па су одвојена стабла са коренима број 0 за Међународну телекомуникациону унију (ITU-T) и 1 за ISO. Како је SNMP протокол потекао из домена Интернета који је иницијално стандардизован у стаблу ISO, данас све променљиве до којих се долази путем SNMP протокола имају OIDe који почињу са неколико истих првих бројева: . 1.3.6.1, односно у текстуалној репрезентацији: *iso.identified-organization.dod.internet* (или краће *iso.org.dod.internet*).

MIB база је направљена тако да може да буде лако проширива, те је због тога SNMP протокол данас коришћен и популаран једнако као и у време када је направљен. Постоје неки обавезни

30 Буквални превод за треп (енг. *Trap*) је замка. Међутим, много је уобичајеније да се овај механизам назива управо овако – треп без превода.

делови стабла као што је подстабло *mib-2* (које почиње OID-ом .1.3.6.1.2.1) које треба да постоји у свакој имплементацији SNMP, а постоје посебна подстабла за различите технологије и протоколе (нпр. стабло за BGP које почиње OID-ом .1.3.6.1.2.1.15, стабло за DNS које почиње OID-ом .1.3.6.1.2.1.32) или произвођаче[31] (нпр. Cisco има свој чвор са OID-ом .1.3.6.1.4.1.9) који на тај начин могу да омогуће коришћење SNMP протокола за оне механизме који су јединствени код њихових производа. Појава било које нове технологије бива једноставно испраћена прављењем MIB спецификације и проширењем базе променљивих новим делом стабла. На слици 4.2 дат је пример организације MIB стабла са неким кључним тачкама у којима почињу подстабла поменута у овом поглављу.



Слика 4.2 Орīанизација MIB сшабла

4.3.1.1. Формаш ūодашака у MIB бази

Објекти у MIB бази су описани подскупом ASN.1 нотације, која се за SNMP протокол зове SMI (енг. *Structure of Management Information*) [4.5], а која је у текстуалном облику и лако читљива и разумљива људима. Слика 4.3 показује почетни део описа MIB базе за BGP-4 протокол и то ревизију описану RFC 4273 документом. У почетном делу описа се види да се подстабло MIB базе за BGP-4 протокол надовезује на *mib-2* тачку у стаблу (.1.3.6.1.2.1), а да је почетна тачка овог подстабла број 15.

31 Интересантно је да и Универзитет у Београду има регистровано своје подстабло. Оно почиње OIDом 1.3.6.1.4.1.11067

```
BGP4-MIB DEFINITIONS ::= BEGIN

    IMPORTS
        MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
        IpAddress, Integer32, Counter32, Gauge32, mib-2
            FROM SNMPv2-SMI
        MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
            FROM SNMPv2-CONF;

    bgp MODULE-IDENTITY
        LAST-UPDATED "200601110000Z"
        ORGANIZATION "IETF IDR Working Group"
        CONTACT-INFO "E-mail:  idr@ietf.org"
        DESCRIPTION
                "The MIB module for the BGP-4 protocol.
                 Copyright (C) The Internet Society (2006).  This
                 version of this MIB module is part of RFC 4273;
                 see the RFC itself for full legal notices."
        REVISION "200601110000Z"
        DESCRIPTION
               "Changes from RFC 1657:
                ...
                Published as RFC 4273."
        REVISION "1991102618392"
        DESCRIPTION
                "Initial version, published as RFC 1269."
        ::= { mib-2 15 }
```

Слика 4.3 Почетак дефиниције BGP-4 MIB-а

Наредна Слика 4.4 показује дефиницију прва два објекта овог дела базе: *bgpVersion* и *bgpLocalAs*, у којима је уписана верзија BGP протокола која се користи и број аутономног система који је конфигурисан на уређајима, који имају бројеве 1 и 2 респективно у модулу *bgp*, те су њихови потпуни OID-и: 1.3.6.1.2.1.15.1 и 1.3.6.1.2.1.15.2 респективно. За све објекте може да се види тип података, начин приступа и ближи текстуални опис.

MIB модули за различите протоколе, попут овог горе наведеног за BGP, те модули које произвођачи праве за своје приватне гране MIB стабла су најчешће слободно доступни за преузимање, или из RFC докумената или са сајтова произвођача и могу се учитати у апликације које се користе за преглед објеката MIB базе на неком уређају (тзв. MIB прегледачи – *browser*-и).

4.3.2. SNMP верзија 1

Прва варијанта прве верзије SNMP протокола описана је RFC документима 1065-1067 1988. године. Две године касније протокол је мало коригован RFC документима 1155-1157 и та верзија се данас сматра за прву верзију овог протокола која је описана у наставку поглавља.

```
    bgpVersion OBJECT-TYPE
        SYNTAX      OCTET STRING (SIZE (1..255))
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "Vector of supported BGP protocol version
                 numbers.  Each peer negotiates the version
                 from this vector.  Versions are identified
                 via the string of bits contained within this
                 object.  The first octet contains bits 0 to
                 7, the second octet contains bits 8 to 15,
                 and so on, with the most significant bit
                 referring to the lowest bit number in the
                 octet (e.g., the MSB of the first octet
                 refers to bit 0).  If a bit, i, is present
                 and set, then the version (i+1) of the BGP
                 is supported."
        REFERENCE
                "RFC 4271, Section 4.2."
        ::= { bgp 1 }
    bgpLocalAs OBJECT-TYPE
        SYNTAX      Integer32 (0..65535)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The local autonomous system number."
        REFERENCE
                 "RFC 4271, Section 4.2, 'My Autonomous System'."
        ::= { bgp 2 }
```

Слика 4.4 Прве две променљиве BGP-4 MIB-а

4.3.2.1. Типови података

Типови података дефинисани првом верзијом SNMP протокола су:

- *Integer* – цео број којим се описују неке променљиве попут статуса интерфејса које могу да имају различите вредности за различита стања (нпр. 1-није активан, 2-активан).
- *Octet string* – низ октета којима се описују текстуалне променљиве (нпр. име интерфејса).
- *Counter* – 32-битни бројач, користи се за бројање бајтова или пакета на интерфејсима.
- OID – идентификатор објекта који је претходно објашњен.
- *IpAddress* – IP адреса.
- *Gauge* – 32-битни број којим се описују неки параметри попут капацитета интерфејса.
- *TimeTicks* – 32-битни цео ненегативан број којим се описује време од неког догађаја, нпр. *system uptime* - време колико је уређај активан. Изражено је у стотим деловима секунде.

4.3.2.2. Команде

Основне команде прве верзије SNMP протокола су: *get* и *set* којима се врши дохватање односно упис неког објекта. У оквиру прве верзије SNMP постоји и команда *getnext* којом се дохвата следећа променљива у стаблу након специфицираног OIDa према *depth-first* претрази. Одговор на *get* команде је *getresponse*, а постоје и тзв. трепови описани у следећем поглављу.

4.3.2.3. Обавештавање о ванредним догађајима - trap

SNMP поседује уграђен механизам којим агент може да обавести менаџмент станицу о ванредним догађајима на уређајима који се прате. Механизам се у жаргону SNMP протокола зове у буквалном преводу замка (*trap*), док је у говору стручњака код нас уобичајено да се зове треп, што ће се и користити у наставку текста. Догађаји који могу да генеришу треп су дефинисани у оквиру MIB базе и није могуће дефинисати нове ван онога што је подржано инсталираним MIB базама. На уређају који се прати могуће је одабрати за које догађаје је потребно послати треп (на пример: пад неког интерфејса, промена статуса BGP суседа и слично). За послати треп менаџмент станица агенту не шаље никакву потврду. Пример једног трепа дефинисаног за BGP протокол је дат на слици 4.5. Реч је о трепу *bgpBackwardTransition* који се генерише када суседски однос између два BGP суседа пређе из вишег у ниже стање (нпр. из стања *Established* у неко претходно стање), што обично значи пад BGP сесије са тим суседом.

```
bgpBackwardTransNotification NOTIFICATION-TYPE
    OBJECTS { bgpPeerRemoteAddr,
              bgpPeerLastError,
              bgpPeerState      }
    STATUS  current
    DESCRIPTION
            "The bgpBackwardTransNotification event is
             generated when the BGP FSM moves from a higher
             numbered state to a lower numbered state.

             This Notification replaces the
             bgpBackwardsTransition Notification."
    ::= { bgpNotification 2 }
```

Слика 4.5 Дефиниција BGP трепа 1.3.6.1.2.1.15.0.2 – bgpBackwardTransition

4.3.2.4. Сигурност SNMP верзије 1

SNMP верзија 1 пружа проверу идентитета менаџмент станице која је и у време настанка протокола сматрана за слабу. За аутентикацију и ауторизацију се користи једноставна шема у којој се између менаџмент станице и агента шаље тзв. *community* стринг. *Community* стринг се шаље у незаштићеном (некриптованом) облику у оквиру SNMP пакета што га чини потпуно рањивим на нападе којима се снима садржај пакета на мрежи. Агент и менаџмент станица морају да имају конфигурисане исте вредности *community* стринга како би могла да

се оствари комуникација између њих. Пренос објеката, њихових имена и садржаја је такође незаштићен (у облику читљивог ASCII текста).

Протоколом су дефинисана два нивоа приступа агенту: „само читање“ (eng. *read only* - RO) који омогућава само очитавање променљивих из MIB базе и „читање и уписивање“ (eng. *read-write* – RW) који дозвољава и уписивање. За ова два нивоа приступа су предвиђене посебне вредности *community* стринга, а такође, ако постоје, трепови могу да имају своју посебну *community* стринг вредност. Управо овај слаб уграђени ниво сигурности је један од разлога зашто се, упркос постојању могућности да се уређаји конфигуришу путем овог протокола, SNMP користи готово искључиво за очитавање променљивих са уређаја и за остваривање функције праћења рада мреже - мања је штета уколико потенцијални нападач има могућност читања статуса уређаја од тога када може да добије и приступ да мења његов начин рада постављањем вредности неких објеката.

4.3.3. SNMP верзија 2

SNMP верзија 2 је уведена свега неколико година након верзије 1. Мотивација је била да се исправи неколико кључних недостатака који су уочени у првој верзији протокола: сигурност, мана неких типова података (нпр. 32-битни бројач), додавање нових команди и побољшање рада механизма трепова. Током рада на новој верзији настало је неколико различитих варијанти протокола:

- прва варијанта описана документима RFC 1441-1447 којом су уведене промене наведене у наставку овог поглавља, али и промене у моделу сигурности којим је предвиђена енкрипција SNMP порука и пренос аутентикационих креденцијала помоћу хеш алгоритама. Због сложености предложеног модела сигурности, ова варијанта протокола никада није прихваћена и брзо је замењена следећом.
- друга варијанта тзв. *community* варијанта SNMPv2c (RFC 1901-1908) која је имала исте промене у командама и типовима података као претходна варијанта, али је задржала стари начин аутентикације из верзије 1 помоћу *community* стрингова.
- трећа варијанта – тзв. *user-based* варијанта SNMPv2u (RFC 1909-1910) која је имала унапређење сигурности у односу на верзију 1, али на начин који је био прихватљивији за коришћење од онога предложеног првом варијантом.

У наставку текста под SNMP верзијом 2 ће се сматрати *community* варијанта која је постала готово искључиво коришћена као верзија 2 овог протокола. Сигурносни механизми предложени варијантом 3 су прихваћени тек као део SNMP верзије 3.

4.3.3.1. Промене уведене SNMPv2 протоколом

Максимална вредност коју 32-битни бројач може да има: 2^{32} (приближно $4 \cdot 10^9$) је са порастом брзина линкова почела да представља проблем у неким применама. Једна од најчешћих примена SNMP је праћење броја бита или бајтова који прођу кроз неки интерфејс

(проток кроз интерфејс). Ово се реализује тако што се периодично (најчешће сваких 5 минута) очитавају бројачи бајтова на интерфејсима (променљиве *ifInOctets*: 1.3.6.1.2.1.2.2.1.10 и *ifOutOctets*: 1.3.6.1.2.1.2.2.1.16), па се на основу разлике добијених вредности и временског интервала добија просечни проток током тог интервала. Лако је показати да кроз потпуно заузет линк од 100Mbps током петоминутног интервала прође у једном смеру око $3{,}75 \cdot 10^9$ бајтова, што значи да ће у таквој ситуацији 32-битни бројач обрнути цео свој циклус у готово сваком интервалу што ће генерисати нетачне резултате за прорачун протока. Са сталним порастом брзина линкова и преласком на гигабитске овај ефекат је постао још израженији јер би се дешавало да се циклус обрне и више пута током једног интервала. Због овога је уведен 64-битни бројач као нови тип података чиме се циклус бројача повећао на $1{,}8 \cdot 10^{19}$ што је довољно и за будуће примене.

У SNMPv2 је уведена нова команда *getbulk*, изведена из команде *get* којом може да се захтева пренос већег броја променљивих одједном по *depth-first* претрази почев од специфицираног OIDa. Ова команда је посебно згодна за преузимање различитих табела из уређаја (нпр. табела рутирања, табеле различитих протокола итд.)

SNMPv2 је променио и начин функционисања механизма трепова. За разлику од верзије 1 у којој се трепови шаљу без потврда о пријему од стране менаџмент станице, у верзији 2 је промењено име трепова у обавештења (енг. *notification*), док је уведена и порука о потврди пријема трепа (*inform*). Обе ове поруке су добиле исту синтаксу као команде *get* и *set*.

4.3.4. SNMP верзија 3

Поред увођења нове поруке (*report*) која служи за слање информација о проблемима у процесирању SNMP порука, кључна промена у верзији 3 протокола (RFC 3411-3418) је нови модел сигурности. Ова верзија уводи и неке промене у архитектури и именовању компоненти архитектуре, али се томе неће посветити више пажње.

4.3.4.1. User-based Security Model (USM)

Први део сигурносних унапређења се односи на побољшање аутентикације и ауторизације приступа агентима, а који се у SNMPv3 терминологији назива USM (енг. *User-based Security Model*) [4.6]. Овим моделом уведена су три нивоа сигурности:

- *noAuthNoPriv* којим се не обезбеђују ни провера идентитета ни заштита послатих података (слично као у старијим верзијама протокола).
- *authNoPriv* којим се обезбеђује сигурна провера идентитета, али без заштите послатих података. Провера идентитета се обезбеђује тако што се корисничко име не шаље преко мреже у незаштићеном облику (као раније *community* вредност) већ се шаљу резултати хеш функција израчунатих од корисничког имена, SNMP поруке и тајног кључа који мора да буде преконфигурисан на обе стране комуникације (тзв. HMAC верзија хеш алгоритма [4.7]). Овиме се онемогућава да потенцијални нападач

може да види креденцијале којима се добија приступ уређајима. Протокол пружа делимичну заштиту од напада понављањем (*replay*) тиме што уређаји треба да буду временски синхронизовани и тиме што се води рачуна о томе да ли је порука стигла од уређаја у очекиваном временском интервалу.

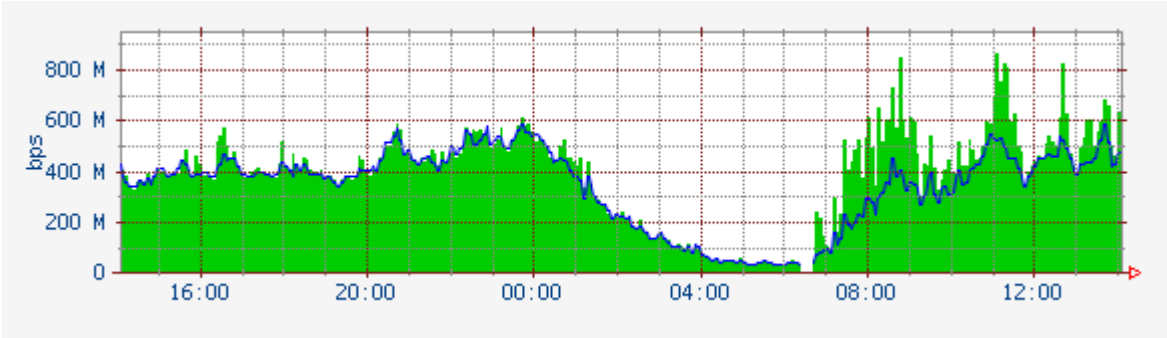
- *authPriv* којим се обезбеђује и сигурна провера идентитета и заштита послатих података енкрипцијом. Енкрипција се обавља неким од стандардних симетричних криптографских алгоритама, а као кључ се користи посебан кључ, различит од оног за аутентикацију.

4.3.4.2. View Access Control Model (VACM)

Други део сигурносних унапређења односи се на дефинисање боље грануларности нивоа приступа MIB бази кроз могућност да се различитим корисницима дозволи приступ до одређених делова MIB базе (на пример, један корисник може да приступа само објектима који описују статусе интерфејса, а други корисник само подацима о рутама). VACM модел омогућава да се дефинише OID који представља корен подстабла којем неки корисник или група корисника имају право да приступе.

4.3.5. Алати за рад са SNMP протоколом

Постоји велики број јавно доступних и бесплатних алата који користе SNMP протокол, а којима могу да се са одабраних уређаја/агената очитају појединачни објекти. Ти уређаји се обично зову MIB прегледачи (браузери – енг. *browser*). Они имају графички интерфејс у који је могуће унети основне конфигурационе параметре (адресу агента, креденцијале за приступ и слично), жељени OID објекта коме се жели приступ и SNMP команда. Пример једног таквог прегледача (*iReasoning MIB Browser*) је дат у поглављу 7.9. Сви објекти имају своје описе који су добијени из стандардно форматираних MIB фајлова који се учитавају у прегледач.



Слика 4.6 График зависности протока саобраћаја кроз неки интерфејс од времена у току једног дана

Овакви алати иако корисни за проверу рада SNMP протокола и проналажење променљивих које дају жељене информације немају већи значај у процесима континуираног праћења рада

мреже јер је тешко испратити трендове промене појединих променљивих у времену (било би потребно да оператер кроз прегледач периодично позива читање појединих објеката, да их бележи и пореди вредности). Због тога је најчешћи начин на који се користи SNMP протокол његова уградња у апликације за праћење рада мреже (мониторинг), које периодично очитавају статусе објеката и приказују их у графичком режиму рада. Пример оваквог графика дат је на слици 4.6 на којој се види график протока саобраћаја кроз неки интерфејс у времену, добијен петоминутним очитавањем бројача октета на датом интерфејсу. За реализацију апликација које користе SNMP постоји велики број програмских интерфејса и библиотека за најразличитије програмске језике.

4.3.6. Новија унапређења SNMP протокола

Ни стандардизација верзије 3 SNMP протокола није донела крај у настојањима да се на квалитетнији начин обезбеди сигурност самог протокола. RFC документи 5590-5592 и 6353 предлажу коришћење неких од постојећих стандардних екстерних механизама заштите који би се користили за SNMP протокол. Ови механизми укључују коришћење добро познатог стандардног TLS (енг. *Transport Layer Security*) или DTLS (енг. *Datagram Transport Layer Security*) протокола, те SSH протокола који се користе за заштиту на транспортном слоју код HTTP, SMTP и других протокола. Међутим, ти механизми у време писања ове књиге нису били још увек подржани од стране највећих произвођача мрене опреме, те је најквалитетнији механизам сигурности и даље онај који доноси верзија 3 SNMP протокола.

4.4. Прикупљање логова - syslog

Још у време пре настанка рачунарских мрежа које раде на данашњем принципу, пракса Unix оперативних система и апликација за њих је била да се кључни догађаји бележе у форми логова (*log* – енг. дневник) кратко записаних информација о тим догађајима које се додају у текстуални фајл. Логови су се бележили локално у неком унапред одређеном директоријуму, а тај принцип су преузели и други оперативни системи, као и мрежни уређаји чији су оперативни системи најчешће и настајали као модификација неке од верзија Unix оперативног система. Нешто касније уведена је могућност да се логови са неког уређаја експортују на централну локацију за прикупљање логова *syslog* протоколом. Ово личи на механизам трепова код SNMP протокола, али за разлику од трепова које је потребно експлицитно укључити, генерисање логова није потребно посебно конфигурисати.

Syslog протокол најчешће користи UDP као транспортни протокол за поруке и не обезбеђује потврду да је лог стигао до централне станице. Иако су стандардима [4.8] дефинисани неки обавезни делови лог порука, попут ознаке система који је генерисао лог (тзв. *Facility code*) или ознаке озбиљности (енг. *severity*) односно значаја забележеног догађаја за

функционисање система (нпр. 0 – *Emergency*, 1 - *Alert*, 3 – *Error*, 6 - *Informational*), формати и садржај порука нису нису јединствени у логовима различитих произвођача што чини аутоматизовану анализу логова релативно тешком. Могуће је направити филтрирање логова на основу ниво озбиљности тако да се нпр. не бележе поруке одређених вредности озбиљности. Слика 4.7 показује један пример логова прикупљених на рутеру у којем се виде поруке озбиљности 3 и 5, време када су забележене и њихиви кодови и кратки описи.

```
00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed
```

Слика 4.7 Пример лог порука прикупљених на рутеру

Извоз лог порука на једну централну локацију омогућио је бољу корелацију и детекцију догађаја који су забележени на више уређаја истовремено, што може да укаже на неке проблеме са перформансама али је и од највећег значаја за детекцију сигурносних упада. Ово је довело до појаве посебне класе тзв. SIEM (енг. *Security Incident and Event Management*) алата који користе у последње време и механизме машинског учења и вештачке интелигенције за откривање специфичних сигурносних упада на основу великог броја информација забележених на различитим уређајима у мрежи. Да би детекција корелисаних догађаја била квалитетна, од великог је значаја добра времеска синхронизација свих уређаја који генеришу логове.

4.5. Прикупљање детаљних информација о саобраћају

Као што је могло да се види у претходним поглављима SNMP и syslog протоколи су оријентисани пре свега на прикупљање информација о статусу уређаја и догађајима у вези са њиховим радом или радом различитих апликација или протокола на њима. Информације о начину коришћења мреже, апликацијама и протоколима које користе корисници мреже, мрежама са којим којима комуницирају и услугама које користе не могу да се добију из ових механизама. Како су детаљне информације неопходне за анализу трендова у мрежи и посебно детекцију различитих врста напада, искристалисала су се два начина на који може да се дође до детаљних информација: коришћењем протокола којима се бележе информације о свим мрежним токовима који пролазе кроз неки уређај и снимањем целокупног саобраћаја на некој вези у мрежи (обично на улазу у мрежу). Ове методе ће бити описане у наставку текста.

4.5.1. Анализа мрежних токова - Netflow/IPFIX

Мрежни ток (енг. *flow*) је одређен као скуп свих пакета заδележених у једном смеру који имају заједничке поједине контролне параметре као што су: долазни интерфејс, IP адресе, δројеви портова протокола транспортног слоја и поље *Type of service*/DSCP у IP заглављу. Мрежни токови могу да δуду врло различитих времена трајања и количина података које су пренете њима. На пример, један мрежни ток чини једна SSH или *telnet* сесија која може да траје и сатима (колико год да је корисник повезан на уређај којем приступа) – ово је пример дуготрајног тока са малом количином пренетих података. Са друге стране приликом преузимања једне веδ стране отвара се типично више мрежних токова (нпр. посеδни токови за дохватање текста, слика, видеа, реклама итд.) који се завршавају оног тренутка када се пренос заврши. То су примери краткотрајних мрежних токова, а постоје и дуготрајни токови са великим количинама пренетих података када се нпр. FTP сесијом преноси неки јако велики фајл.

Док је за TCP токове јасно њихово трајање (од првог пакета са SYN флегом до последњег пакета са FIN или RST флегом), за UDP протокол код кога нема праћења стања сесије ово није могуће, па се οδично одреди временски интервал током кога ако нема пакета који припадају датом току.

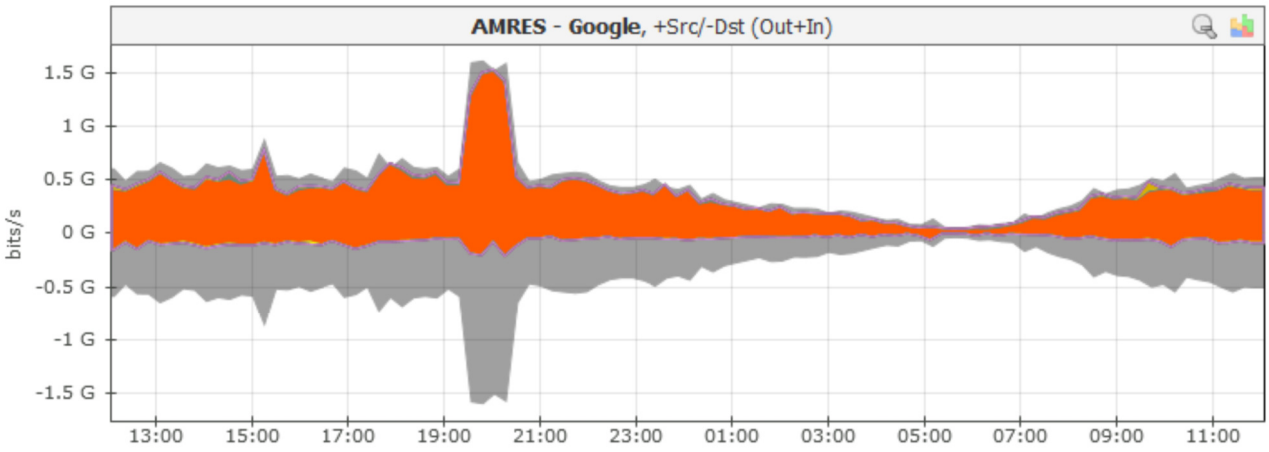
Start Time	End Time	Duration	Src IP	Src Port	Dst IP	Dst Port	Protocol
15-11-2017 19:36:47.468	15-11-2017 19:37:11.100	23.632 sec	160.99.40.41	389	99.103.97.111	34889	17
15-11-2017 19:33:56.400	15-11-2017 19:37:13.703	0.0 sec	160.99.1.1	52786	84.53.139.193	53	17
15-11-2017 19:37:11.598	15-11-2017 19:37:11.598	0.0 sec	160.99.21.5	59707	91.228.166.45	53	17
15-11-2017 19:36:19.439	15-11-2017 19:37:13.930	54.491 sec	160.99.40.41	0	95.172.92.218	0	17

TOS	TCP Flags	Flows	Packets	Bytes	Throughput	Exporter	Interface In
0	none	1	4	6,000	2.0 Kbps	172.17.12.1	619
0	none	1	1	73	-	172.17.12.1	619
0	none	1	1	179	-	172.17.12.1	619
0	none	1	287	411,558	60.4 Kbps	172.17.12.1	619

Слика 4.8: Сирове информације о мрежним токовима

Мрежни уређаји са могућношћу праћења токова воде евиденцију о свим токовима који пролазе кроз њих. У тренутку када се мрежни ток заврши рутер δележи кључне информације о њему попут: времена почетка тока, времена трајања тока, δроја пакета и δајтова који су пренети тим током (тачан скуп информација зависи од врсте протокола који се користи). Слика 4.8 показује пример информације које су заδележене за 4 мрежна тока. Зδог δољег визуелног приказа информације су на слици раздвојене у два дела. Сваки запис о мрежном току се извози (шаље) са мрежног уређаја (па се у овој архитектури мрежни уређај зове експортер) до колектора токова и ту се δележи у фајловима који се генеришу најчешће сваких пет минута са временском ознаком у имену сваког фајла. Посеδан уређај – анализатор анализира прикупљене токове и приказује их кориснику најчешће у графичкој форми, а постоје могућности генерисања аларма ако се детектује одређена врста догађаја. Један такав пример је дат на слици 4.9. на којој се види расподела по протоколима снимљена између

AMPEC мреже и Google-ових мрежа добијена анализом токова помоћу алата NetVizura[32]. Слика показује један пример у којем се види да је могуће детектовати да се у неком периоду дана (између 19:00 и 21:00 часова) десила активност која одступа од уобичајеног понашања саобраћаја што може да буде индикација за неку малициозну активност. Додатном анализом мрежних токова у том периоду могуће је добити прецизне информације о томе који тачно уређаји су учествовали у генерисању аномалије у саобраћају и поверити да ли је заиста било речи о нападу или не.



	Src (Out+In)		Dst (Out+In)	
Services	Avg	Max	Avg	Max
HTTPS [443]	341.9 Mbps (83.2 %)	1.5 Gbps	53.9 Mbps (13.1 %)	217.3 Mbps
HTTP [80, 8081]	8.8 Mbps (2.1 %)	60.9 Mbps	560.2 kbps (0.1 %)	2.9 Mbps
SMTP [25]	134.7 kbps (<0.1 %)	413.6 kbps	2.0 Mbps (0.5 %)	23.6 Mbps
IMAP-SSL [993]	1.3 Mbps (0.3 %)	73.0 Mbps	118.3 kbps (<0.1 %)	3.5 Mbps
DNS [53]	318.5 kbps (<0.1 %)	566.2 kbps	105.6 kbps (<0.1 %)	165.8 kbps
POP3 [110]	380.0 kbps (<0.1 %)	2.0 Mbps	16.7 kbps (<0.1 %)	97.4 kbps
Port [5228]	203.4 kbps (<0.1 %)	578.2 kbps	62.6 kbps (<0.1 %)	129.6 kbps

Слика 4.9 Пример анализа токова према сервису који се користи

Први протокол којим је била омогућена анализа мрежних токова настао је у компанији Cisco systems и имао је назив NetFlow који има неколико верзија. Најуобичајенија верзија овог протокола на уређајима је v5, а касније је настала и верзија 9 која експортује неке додатне информације о токовима у односу на верзију 5 и која је објављена као нестандардни (информативни) RFC документ [4.9]. Концепт прикупљања и анализе токова постао је брзо популаран и један од кључних алата у анализи понашања мрежа, трендова у њима и откривања малициозног понашања. Ово, а и чињеница да се и у релативно малој мрежи генерише велики број мрежних токова у јединици времена[33] је довело и до развоја великог

32 https://www.netvizura.com/

33 Према подацима из AMPEC мреже, у тренуцима када је проток пакета око 4Gb/s има око 18.000 мрежних токова у секунди, величина једног петоминутног фајла је око 160MB

броја софтверских алата за анализу токова и проналажења скривених информација о понашању мреже коришћењем модерних софтверских метода. Стога су многи произвођачи развили своје сличне, али међусобно некомпатибилне системе, попут Jflow фирме Juniper Networks, Cflowd Alcatel Lucent-a и други. Да би се некако стандардизовали ови системи, на основу NetFlow v9 је предложен IETF стандард IPFIX чија је актуелна верзија описана RFC документом 7011.

4.5.2. Мрежни акцелератори – прикупљање садржаја свих пакета

Претходно описана анализа мрежних токова даје агрегиране податке о мрежним токовима, али из њих не може да се види детаљан садржај комуникације и не могу да се добију информације о појединачним пакетима и њиховом садржају. Уколико је овакав ниво детаља и увид у садржај пакета потребан, неопходно је да се врши снимање целокупног саобраћаја на неком мрежном сегменту. Снимање саобраћаја је могуће вршити и рачунарима опште намене, коришћењем неког од алата који мрежни интерфејс стављају у тзв. промискуитетни режим рада када може да прими све пакете који се појаве на датом мрежном сегменту без обзира на дестинацију којој су намењени. Пакети са неке везе која се посматра могу да се доведу до рачунара који их снимају на два начина: коришћењем тзв. *mirror* функције на мрежним уређајима којима се сав саобраћај са одређеног порта копира на други жељени порт на којем је рачунар који врши анаизу или уметањем тзв. оптичких каплера (енг. *coupler*) на везу између два мрежна уређаја чиме се скреће део светлосног сигнала до рачунара који врши анализу који на тај начин може да добије потпуну копију свих пакета са посматраног линка.

Најпопуларнији алат за снимање пакета је Wireshark[34] који има графички кориснички интерфејс и уређен начин за преглед садржаја пакета и свих заглавља. Начин коришћења овог алата је показан у више примера у глави 7. Међутим перформансе овог алата нису адекватне за коришћење на везама већег капацитета. Наиме, како би обезбедио комфоран кориснички интерфејс Wireshark има имплементиране сложене филтере за разврставање пакета и анализу разних заглавља, те снима све забележене пакете на диск што у многоме ограничава број пакета који могу да буду снимљени у јединици времена, посебно у конфигурацијама са споријим дисковима, тако да се често дешава да део пакета буде одбачен. За прихватање већег броја пакета могу да се користе или слични програми који раде из командне линије (*tshark*) или директно библиотеке писане у језику C (нпр. *libpcap* или *dumpcap*) на којима се заснива рад претходно поменутих алата за снимање пакета. Један од новијих алата за снимање саобраћаја *netsniff-ng*, који користи „*zero-copy*“ принцип[35] приликом прихватања снимљених пакета има могућност снимања до 1Gb/s помоћу рачунара опште намене.

34 https://www.wireshark.org/

35 Zero-copy начин рада подразумева минимизацију броја пребацивања контекста између корисничког и кернел мода коришћењем посебних библиотека и метода које подржавају пренос поинтера на податке уместо комплетних података између корисничког и кернел мода, чиме се повећава ефикасност операција и неколико пута [4.10]

Међутим, данас су у рачунарским мрежама уобичајени капацитети веза од 10 или 100 Gb/s. За везе тих капацитета није могуће снимити сав саобраћај коришћењем рачунара и мрежних картица опште намене јер све активности које је потребно да се изврше над пакетима (примање, филтрирање у складу са неким правилом, пренос у меморију) не могу да се обаве у периоду између доласка два суседна пакета[36]. Стога за ову сврху постоје специјализовани мрежни акцелератори, односно мрежне картице које имају могућност снимања целокупног саобраћаја на линковима капацитета до 100Gb/s[37]. Ово је постигнуто тиме што мрежни акцелератори поседују FPGA чипове које је могуће програмирати тако да делимично растерете процесор рачунара тиме што ће преузимање пакета и њихово иницијално филтрирање бити извршено на акцелератору, а одабрани пакети пребачени процесору.

4.6. Аутматизација конфигурисања - NETCONF и YANG

Током дужег временског периода за комуникацију са уређајима су се претежно користили SNMP протокол за проверу статуса и приступ из командне линије за конфигурисање. SNMP због добро познатих сигурносних слабости никада није постао протокол који се користи за конфигурисање. Са друге стране, пораст броја уређаја и сложености конфигурација (посебно у виртуелизованим окружењима) довео је до јаке потребе да се процес конфигурисања аутоматизује на стандардан начин. IETF је за ту сврху своје активности усмерио у правцу доношења два протокола: NETCONF који служи за управљање конфигурацијама, њихово слање ка уређајима и преузимање и YANG, језик за моделовање конфигурација које се шаљу путем NETCONF.

4.6.1. NETCONF

NETCONF протокол служи за управљање конфигурацијама на мрежним уређајима. У терминологији протокола клијент је софтвер који се налази на менаџмент уређају, а сервер се налази на мрежном уређају којим се управља. NETCONF користи неки од добро познатих протокола за сигуран транспорт између клијента и сервера (SSH, TLS и други) којим се обезбеђује аутентикација, интегритет и поверљивост порука. Протокол користи механизам удаљених позива процедура (RPC – енг. *Remote Procedure Call*) за инцирање операција над конфигурацијама. Све поруке су у XML формату, што омогућава да се и најсложеније хијерархијске конфигурације мрежних уређаја прикажу у лако читљивом облику. Основне операције NETCONF протокола [4.11] су:

- *get-config* којом се преузима цела или специфицирани део конфгурације,

36 Ако се претпостави да је просечна величина пакета 400 бајтова и ако је линк капацитета 1Gb/s потпуно загушен, просечно време између два пакета је 3,2μs, а за линкове капацитета 10Gb/s односно 100Gb/s су ове времена 0,32μs односно 32ns респективно, а за најмање пакете и краћа.

37 На пример: https://www.napatech.com/products/napatech-smartnics/

- *edit-config* учитава/мења део или целу конфигурацију,
- *copy-config* којим се копира део или цела конфигурација,
- *delete-config* којом се брише део или цела конфигурација (подразумева се да може да се обрише само она конфигурација која није тренутно активна),
- операције *lock* и *unlock* којим се конфигураицција закључава и откључава за промене како би се боље синхронизовао рад већег броја актера,
- *get* којом се преузима део конфигурације али и статуса уређаја у вези са тим делом конфигурације
- *close-session* и *kill session* којима се захтева завршетак NETCONF сесије

Слика 4.10 показује пример *edit-config* операције којом клијент захтева да се на серверу у делу конфигурације интерфејса дода параметар MTU вредности 1500 на интерфејс Ethernet0/0.

```
<rpc message-id="101"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <top xmlns="http://example.com/schema/1.2/config">
        <interface>
          <name>Ethernet0/0</name>
          <mtu>1500</mtu>
        </interface>
      </top>
    </config>
  </edit-config>
</rpc>
```

Слика 4.10 edit-config операција догеле MTU вредности

Овакав начин форматирања и слања конфигурација омогућава израду апликација које би аутоматизовале конфигурацију мрежних уређаја, а тиме и процесе реакција на проблеме у мрежи (тзв. *Autonomic network* концепт [4.13]).

Оно што не представља део NETCONF протокола су модели података елемената конфигурација којима се управља у оквиру NETCONF операција, тако да се често среће ситуација у којој су произвођачи имплементирали NETCONF користећи своје специфичне формате конфигурација описане у поглављу 4.2. Унификацију синтакси конфигурација уређаја и протокола на уређајима различитих произвођача треба да изврши YANG језик.

4.6.2. YANG

YANG је језик за моделовање података којим се дефинишу и документују формати конфигурационих података, као и подаци о стању мрежних уређаја који се шаљу путем

NETCONF протокола, без обзира на синтаксу конфигурационих команди на уређајима. Слично организацији MIB базе, подаци су у оквиру YANG спецификације организовани у форми стабла, где сваки чвор има име и садржи или вредност или под-чворове. Приликом дизајна YANG модела тежило се да се задржи компатибилност променљивих које су описане овим моделом са MIB форматом и типовима података. У наставку је дат само врло кратак преглед особина YANG језика.

Постоје четири основна типа чворова у YANG стаблу који служе за моделовање података:

- лист – *leaf*, који садржи неку променљиву која је неког основног типа
- листа листова – *leaf-list*, који садржи листу листова истог типа
- контејнер – *container* којим се групише више чворова у подстабла
- листа – *list* којим се дефинише листа чворова које је могуће претражити на основу дефинисаног кључа.

Слично као у другим спецификацијама језика или схема којима може да се врши дефиниција модела података (нпр. JSON или XML схема), за све чворове је могуће дефинисати текстуалне описе и поставити ограничења вредности и правила коришћења за променљиве, како би могла да се изврши валидација инстанци модела података. Основни типови података су преузети иницијално из SNMP, уз додатак неких специфичних попут физичких адреса, бројева портова, аутономних система итд. [4.14], а могуће је извођење и изведених типова.

YANG омогућава груписање чворова у логичке целине - модуле и под-модуле који могу да се увезу из других фајлова (*include*). Део примера једног модула којим је описан DHCP протокол показан је на слици 4.11 у YANG синтакси[38]. Саме инстанце модела података се најчешће записују у XML или JSON формату. На примеру се види начин на који је описан лист чвор у којем се записује максимално време додељивања адресе, које има дефинисану и подразумевану вредност (*default*), као и контејнер у којем су подаци о статусу додељивања адреса: у контејнеру је листа *leases* која даје почетно и крајње време додаљивања адресе за дати кључ – IP адресу која је издата. Такође дефинисан је и тип мреже који је дао адресу и хардверска адреса уређаја.

OpenConfig[39] група коју чине велики пружаоци садржаја на интернету (Google, Facebook, Microsoft, Apple и други) и провајдери (Verizon, Level3, AT&T и други), а чији је интерес једноставнија аутоматизација процеса конфигурисања и избегавање везивања за поједине произвођаче мрежне опреме је прихватила YANG као језик којим треба да се моделују конфигурациони елементи мрежних уређаја и направила је највећи продор у стандардизацији модела конфигурација различитих протокола [4.12]. Такође у оквиру ове групе развијен је низ алата за рад са YANG моделима попут YANG парсера и преводиоца, те додатака за софтверска развојна окружења и библиотека.

38 Уобичајено је да се YANG модели приказују и у XML формату, који се зове *YANG Independent Notation* (YIN)

39 http://openconfig.net/

```
module dhcp {
  namespace "http://yangcentral.org/ns/example/dhcp";
  prefix dhcp;

  import ietfyangtypes { prefix yang; }
  import ietfinettypes { prefix inet; }

  organization
    "yangcentral.org";
  description
    "Partial data model for DHCP, based on the config of the ISC
       DHCP reference implementation.";
  container dhcp {
    description
      "configuration and operational parameters for a DHCP server.";
    leaf maxleasetime{
      type uint32;
      units seconds;
      default 7200;
    }
  }
    …
  container status {
    config false;
    list leases {
      key address;
      leaf address {
        type inet:ipaddress;
      }
      leaf starts {
        type yang:dateandtime;
      }
      leaf ends {
        type yang:dateandtime;
      }
      container hardware {
        leaf type {
          type enumeration {
            enum "ethernet";
            enum "tokenring";
            enum "fddi";
          }
        }
        leaf address {
          type yang:physaddress;
        }
      }
    }
…
}
```

Слика 4.11 Пример модула којим је описан DHCP протокол у YANG синтакси

4.7. TMF модел

Као што је наведено раније, у последњих десетак година направљена је велика промена приоритета у управљању мрежама у правцу аутоматизације управљачких процеса, те је цела област из дисциплине оријентисане на управљање мрежним уређајима прешла у правцу дисциплина пројектовања, израде и оркестрације бројних апликација које управљају различитим процесима. Све ове апликације се налазе у управљачком слоју, повезане тзв. *northbound* интерфејсима са EMS системима за управљање елементима. Подразумевана

архитектура менаџмент апликација је дистрибуирана, слабо спрегнута и сервис-оријентисана (енг. *Service-Oriented Architecture* – SOA), где свака апликација обавља једну јасно дефинисану функцију и комуницира са другим апликацијама преко магистрале сервиса (енг. *Enterprise Service Bus*) и дефинисаних интерфејса.

TMF је своје активности усмерио у неколико праваца, а све је сумирано у оквиру тзв. Frameworx скупа препорука добре праксе [4.15]:

- анализа свих пословних процеса у управљању мрежом. Израђен је генерални модел процеса (eTOM или *Business Framework*) [4.16] који је прихваћен као стандард од стране више стандардизационих тела (ITU-T, ETSI). Овај модел описује све процесе који могу да се појаве у активностима управљања мрежом, али и шире (управљање организацијом која управља мрежом, планирањем мрежних услуга, комерцијалном страном мрежних услуга и наплатом итд.). Модел може да се користи како за оптимизацију организације провајдера, процеса и процесних токова у њима, тако и као заједнички речник и референца за то шта одређени пословни процеси подразумевају и као референца процеса које треба да апликације за управљање подрже и аутоматизују.
- спецификација карактеристика и функционалности апликација за управљање мрежом и услугама. Направљен је модел „свих“ апликација које могу да се користе у управљању мрежом, од слоја обраде података добијених из мреже до највишег слоја интеракције са корисницима услуга и дати су генерички описи њихових функционаности (TAM – *Telecom Applications Map* или *Application Framework*) [4.17]. Овим се остварује боље разумевање купаца и продаваца апликација за управљање у погледу тога шта одређена апликација треба да испуни као и јасна дефиниција међусобних односа са другим апликацијама са којима треба да се изврши интеграција. Последња верзија TAM из времена настанка ове књиге садржи описе око 200 различитих апликација управљачког слоја.
- спецификација модела података кључних ентитета који се појављују у оквиру апликација: SID модел [4.18]. Модели података су дати у различитим форматима (XMI, UML, RSA) које је могуће преузети, по потреби модификовати и искористити. Овим моделом се доста скраћује процес пројектовања апликација јер постоје готови модели података за бројне ентитете који се описују управљачким апликацијама (модел података мрежних уређаја, услуга, корисника итд.).
- спецификација стандарних интерфејса између апликација. TMF током своје историје стандардизовао различите апликативне интерфејсе у облику MTNM, MTOSI или OSS/J стандарда које су имплементирали произвођачи управљачких апликација и тиме омогућили једноставну интеграцију са компонентама других произвођача које подржавају исте интерфејсе. У последњим годинама TMF је покренуо TIP (*TMF Interface Program*) програм којим је требало да се сви ови интерфејси унификују. Данас има преко 50 различитих API-ја који имају статус стандарда, а поред

спецификација модела података и операција интерфејса постоје и готове swagger[40] имплементације интерфејса и примери у JSON формату.

40 Платформа отвореног кода за прављење APIја: https://swagger.io/

4.8. Литература

[4.1] ISO/IEC 10040:1998(en) Information technology — Open Systems Interconnection — Systems management overview

[4.2] Pavlou, G. (2007). On the Evolution of Management Approaches, Frameworks and Protocols: A Historical Perspective. *Journal of Network and Systems Management*, *15*(4), 425–445. https://doi.org/10.1007/s10922-007-9082-9

[4.3] Edwards, R. (2007). History and Status of Operations Support Systems. *Journal of Network and Systems Management*, *15*(4), 555–567. https://doi.org/10.1007/s10922-007-9077-6

[4.4] J. Case, M. Fedor, M. Schoffstall, J. Davin, A simple network management protocol, RFC 1157, May 1990, https://www.ietf.org/rfc/rfc1157.txt

[4.5] K. McCloghrie, D. Perkins, J. Schoenwaelder, Structure of Management Information Version 2 (SMIv2), IETF RFC 2578, April 1999, https://tools.ietf.org/html/rfc2578

[4.6] U. Blumenthal, B.Wijnen, User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3), IETF RFC 3414, December 2002, https://tools.ietf.org/html/rfc3414

[4.7] J. Merkle, M. Lochter, HMAC-SHA-2 Authentication Protocols in User-Based Security Model (USM) for SNMPv3, IETF RFC 7860, April 2016, https://tools.ietf.org/html/rfc7860

[4.8] R. Gerhards, The Syslog protocol, IETF RFC 5424, March 2009, https://tools.ietf.org/html/rfc5424

[4.9] B. Claise, Cisco Systems NetFlow Services Export Version 9, IETF RFC 3954, October 2004., https://tools.ietf.org/html/rfc3954

[4.10] S. Palaniappan, P. Nagaraja, Efficient data transfer through zero copy, IBM developer works, https://www.ibm.com/developerworks/library/j-zerocopy/j-zerocopy-pdf.pdf, Приступљено 22.11.2017.

[4.11] https://tools.ietf.org/html/rfc6241

[4.12] OpenConfig data models and APIs, http://openconfig.net/projects/models/, приступљено 22.11.2017.

[4.13] Dobson, S., Zambonelli, F., Denazis, S., Fernández, A., Gaïti, D., Gelenbe, E., … Schmidt, N. (2006). A survey of autonomic communications. *ACM Transactions on Autonomous and Adaptive Systems*, *1*(2), 223–259. https://doi.org/10.1145/1186778.1186782

[4.14] J. Schoenwaelder, Common YANG Data Types, IETF RFC 6021, October 2010., https://tools.ietf.org/html/rfc6021

[4.15] Frameworx Reference, Core Frameworks Concepts and Principles Business Process, Information and Application Frameworks, GB991, Release 16.5.1, February 2017.

[4.16] Frameworx Standard, Business Process Framework (eTOM) Process Decompositions and Descriptions, Business Process Framework, GB921 Addendum D, Release 17.0.1, November 2017.

[4.17] Frameworx Standard, Application Framework, The Digital Services Systems Landscape, Application Framework Suite, GB929 Addendum D, Release 17.0.1, November 2017.

[4.18] Frameworx How-To Guide, Information Framework (SID), User’s Guide, Information Framework Suite, GB922 User’s Guide, Release 17.0.1, November 2017.

5. Испорука садржаја преко рачунарских мрежа

Са развојем интернет услуга, а пре свега веба, већ средином 1990-их година почео је да се јавља један нови феномен: у случају појављивања на интернету неког садржаја који је јако интересантан великом броју корисника, долазило би до успостављања великог броја конекција ка веб серверу са тим садржајем у кратком временском интервалу – долазило је до нечега што се у интернет жаргону зове *flash crowd* или *slashdot* ефекат. Привремено велико интересовање за неким садржајем је доводило до привремених загушења и недоступности сервера и морали су да се пронађу неки начини за ефикаснију дистрибуцију садржаја. Данас су ови проблеми још више изражени: у 2016. години 73% целокупног IP саобраћаја је био видео саобраћај (дистрибуција снимљеног видео садржаја као на youtube, netflix и слично, али и преноси уживо) са трендом већег повећања од осталог саобраћаја [5.1]. Ово се види и у промени начина коришћења рачунарских мрежа. Док је пре десетак година време највећег оптерећења било у радно време (између 8 и 17 часова), данас је највећи проток у вечерњим часовима, од 19 до 23, где већина корисника гледа различите видео садржаје од куће[41]. Дакле, потребно је било да се пронађе решење за проблем дистрибуције огромне количине садржаја до практично свих корисника на интернету.

Једна од најједноставнијих стратегија за повећање капацитета система за дистрибуцију садржаја је додавање додатних сервера на којима се налази копија интересантног садржаја и балансирање оптерећења између њих, што може да се врши на различите начине:

- помоћу динамичког NAT-а [5.2], тако што више сервера са истим садржајем има приватне адресе и налази иза NAT уређаја (обично рутер). Корисник који жели да приступи садржају добија приликом разрешавања симболичког имена сервера једну јавну адресу NAT уређаја, а NAT уређај по добијању захтева за приступом садржају

41 Већина великих тачака за размену интернет саобраћаја објављује статистику количине пренетих података. Читалац може да погледа како то изгледа на Амстердамској тачки за размену саобраћаја: https://ams-ix.net/technical/statistics

дистрибуира захтеве овим серверима по одређеној уобичајеној схеми (кружно, по приоритетима итд.).

- помоћу DNS-а, тако што се једно симболичко име сервера упари са више IP адреса које имају сервери са копијом истог садржаја. DNS сервер по захтеву за разрешавање симболичког имена сервера враћа увек различиту адресу, најчешће по кружном редоследу, мада постоји могућност дефинисања и тежинских фактора оптерећења сервера (да неки сервери имају веће оптерећење од других, ако су бољих хардверских перформанси).

Ипак, ове стратегије су погодне за сервере који имају релативно мало и локално генерисано оптерећење. За сервере који поседују садржаје који су интересантни милионима корисника широм света (попут нпр. youtube) повећање броја захтева ка серверима који би били смештени у једном дата центру би значило и значајно повећање мрежног оптерећења на везама ка дата центру, па су потребни другачији начини за дистрибуцију садржаја којима се много ефикасније користе мрежни ресурси. У наставку текста ће бити објашњени:

- начин рада мрежа за испоруку садржаја (енг. *Content Delivery Networks - CDN*) код којих се исти садржај реплицира на сервере широм интернета тако да буде много ближе корисницима. CDN представљају данас кључни начин за обезбеђивање брзе и ефикасне глобалне доступности садржаја.
- мултикаст начин преноса пакета којим се садржај испоручује до свих корисника који желе да га приме на начин који минимално троши ресурсе мреже.

5.1. Мреже за испоруку садржаја – CDN

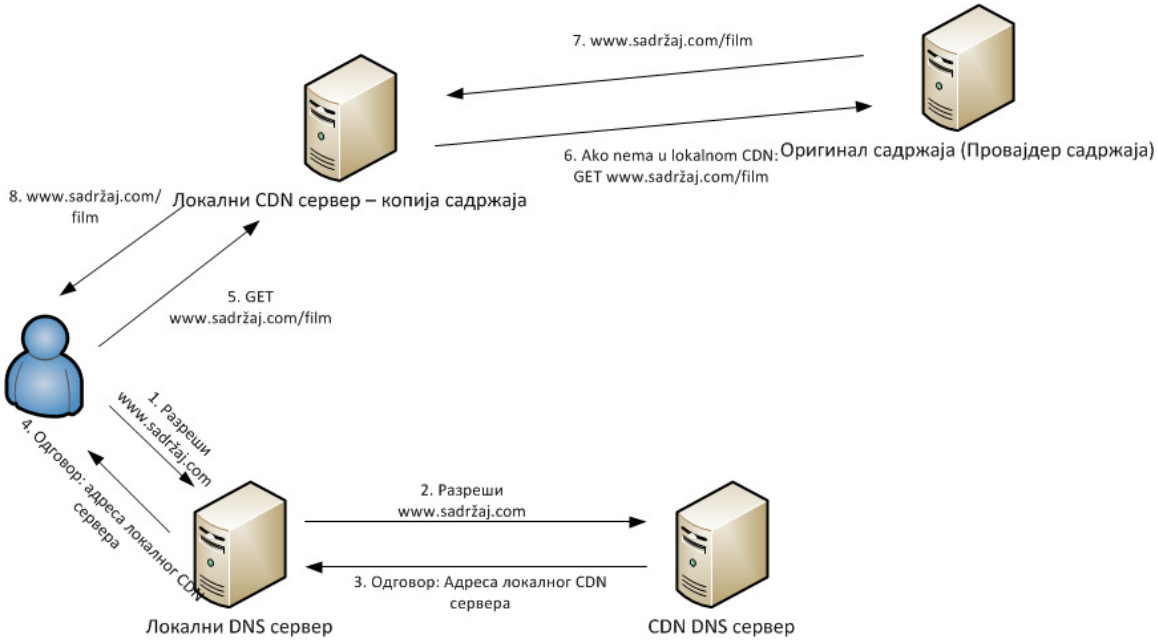
Данас постоји велики број компанија које пружају услугу испоруке садржаја до корисника на интернету (најпознатија и највећа је Akamai[42]), а у последњим годинама и све већи број пружалаца интернет услуга и класичних телекомуникационих компанија улази на ово тржиште. Мреже за испоруку садржаја нису посебне рачунарске мреже које се састоје од мрежних уређаја и веза. Оне се састоје од великог броја сервера који се налазе у дата центрима повезаним на интернет широм света (према подацима из 2017. године, Akamai поседује преко 200.000 сервера у преко 120 земаља света). Фирме власници садржаја (нпр. телевизије, новинске компаније, компаније за испоруку аудио садржаја, итд.) које желе ефикасну дистрибуцију и испоруку сопственог садржаја склапају уговор са CDN мрежама којим се регулише копирање оригиналног садржаја на сервере CDN. Овиме се постиже дистрибуција истог садржаја до различитих тачака на интернету, тако да корисници у некој земљи приступају копији садржаја који се физички налази најчешће на серверима у тој земљи или у региону, а не оригиналу садржаја који је у фирми која је његов власник. Тиме се скраћује пут који садржај треба да пређе до корисника и његово кашњење у преносу. У

42 https://www.akamai.com/

ситуацијама када велики број корисника из неког региона жели да приступи истом садржају и то чини са сервера који је у том региону, биће смањен и проток саобраћаја по међународним комуникационим везама ка оригиналу садржаја. У наставку поглавља је објашњен један од најчешћих начина на који раде CDN – тзв. *non-cooperative pull-based* [5.3].

Кључни елемент овог начина испоруке садржаја је посебан режим рада DNS сервиса, што је показано на слици 5.1. Процес испоруке садржаја почиње када корисник у некој апликацији (нпр. веб прегледач) покрене процес (у овом случају жели да види филм са сајта www.sadrzaj.com). Ако се претпостави да клијентски рачунар нема сачувану IP адресу сајта, кориснички уређај мора да се обрати DNS серверу да разреши име веб сервера домена на којем је садржај sadrzaj.com. Уколико ни локални DNS сервер нема IP адресу сајта, обратиће се DNS серверу који је задужен за домен sadrzaj.com. У овом случају се поставља да CDN мрежа буде задужена за одржавање домена sadrzaj.com. DNS сервер CDN мреже ће одговорити IP адресом локалног сервера CDN мреже који се налази близу (у смислу мрежног растојања) корисника. Корисник ће свој захтев упутити ка CDN серверу чију је IP адресу добио. Уколико на серверу не постоји захтевани садржај, CDN сервер ће се обратити серверу власника садржаја да му тај садржај испоручи и садржај ће кроз CDN сервер доћи до корисника. У овом случају нема никакве уштеде мрежних ресурса и повећања ефикасности приступа садржају јер се садржај испоручује са сервера власника садржаја. Међутим, CDN сервер ће копију садржаја чувати и сваки следећи корисник из истог региона који захтева тај исти садржај ће га добити директно са CDN сервера, чиме ће сервер власника садржаја бити растерећен додатних сесија и неће бити мрежног саобраћаја од власника садржаја ка кориснику. Овај начин рада CDN се назива *non-cooperative pull-based* зато што CDN сервери не комуницирају међусобно и не обавештавају се о томе који сервер поседује реплику ког садржаја, а садржај се повлачи (енг. *pull*) са сервера пружаоца садржаја по исказаној потреби. Постоје и предлози неких *cooperative* шема у којима CDN сервери међусобно размењују информације о томе који сервер има одређене садржаје и *push-based* механизама код којих се садржај испоручује на CDN сервере и пре пристиглих захтева, али су ове шеме више од академског него од реалног практичног значаја.

Један од кључних елемената CDN мреже је алгоритам којим DNS сервер динамички одлучује адресу ког CDN сервера ће вратити кориснику. На ову одлуку утичу: (мрежна) раздаљина између корисника и сервера, тренутно оптерећење сервера, загушење мреже на линковима ка серверима, кашњења до сервера итд. У случајевима великог регионалног оптерећења неких сервера, може да се деси да два корисника који се налазе у мрежи истог провајдера добију садржај са различитих CDN сервера, како би се боље избалансирало оптерећење. Због свега овога велике CDN мреже имају врло сложене механизме мониторинга рада своје инфраструктуре и веза између сервера, па и објављују врло квалитетне извештаје о статусу интернета [5.4].



Слика 5.1 non-cooperative pull-based начин рада CDN

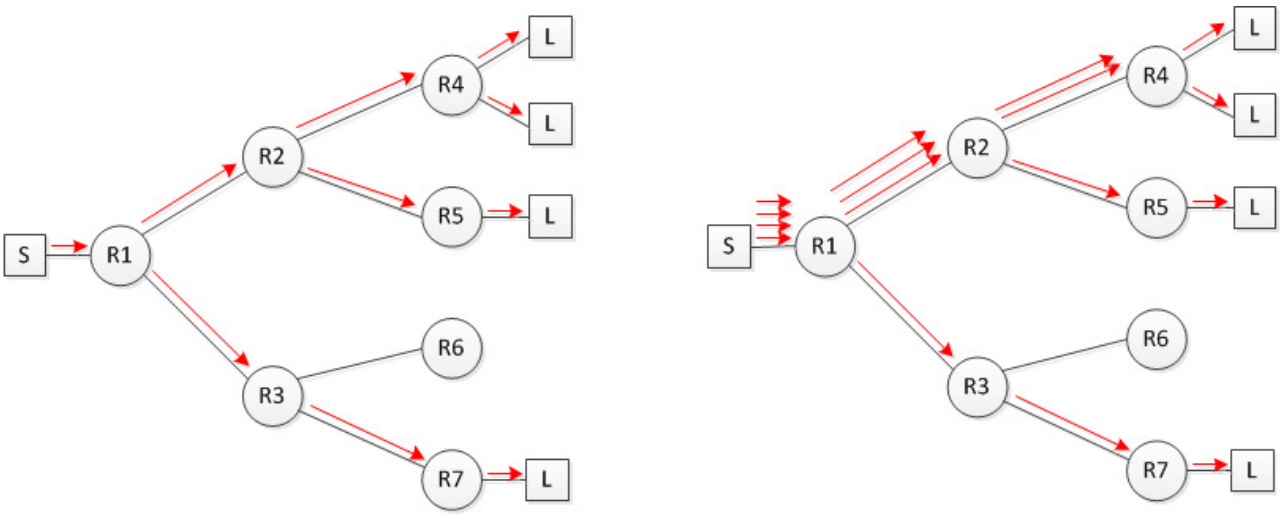
CDN мреже су најпогодније за дистрибуцију статичког *front-end* садржаја попут слика, и снимљеног (*offline*) видео или аудио садржаја. Уколико уско грло у некој вишеслојној апликацији којој се приступа преко интернета није у *front-end* слоју, онда може да се врши репликација апликација или *back-end* слоја методама *edge computing-a* и репликација база. Ово најчешће не раде CDN провајдери.

5.2. Мултикаст

Мултикаст (енг. *multicast*) је начин преноса садржаја од једног извора ка већем броју слушалаца који је изумео Стивен Диринг (*Stephen Deering*) током истраживачког рада на својој докторској дисертацији [5.5]. При томе, за разлику од бродкаста (енг. *broadcast*) код ког се садржај преноси до свих уређаја у некој мрежи, код мултикаста слушаоци могу да одаберу који ће мултикаст садржај примати[43], у ком временском периоду и могу да се налазе у више различитих мрежа. Мултикаст обезбеђује оптимално (минимално) искоришћење мрежних ресурса приликом преноса садржаја до већег броја слушалаца тако да се на свакој мрежној вези налази највише један низ пакета са садржајем који се шаљу слушаоцима, што је показано на слици 5.2. На левој страни слике је приказан начин на који се пакети дистрибуирају када један извор шаље мултикаст садржај ка четири слушаоца који се налазе у три различите мреже. Неки рутери у мрежи (R1, R2, R4) шаљу приспеле пакете на више излазних интерфејса истовремено, али тако да на свакој вези постоји највише једна копија пакета. За разлику од тога на десној страни слике је приказан начин на који се садржај

43 Ово се често каже и да „слушалац припада одређеној мултикаст групи“

дистибуира до истих корисника када се користи класичан уникаст који се користи у свим методама дистрибуције садржаја поменутим у претходном поглављу. Том приликом сваки слушалац има своју посебну сесију ка извору садржаја, што доводи до тога да се на неким везама налази истовремено више копија истог садржаја. Очигледно је да је мултикаст начин дистрибуције скалабилнији јер додавање нових корисника не производи повећање протока пакета на везама у мрежи.



Слика 5.2 Мултикаст (лево) и уникаст (десно) дистрибуција пакета до истог скупа слушалаца

Поред очигледно боље скалабилности као кључне мотивације за увођење мултикаста, овакав начин дистрибуције пакета повлачи неке последице:

- Као транспортни протокол се користи искључиво UDP. Пошто се пакети шаљу од једног извора ка већем броју слушалаца, нема могућности да се формира једна јединствена TCP сесија (која је по својој природи уникаст). Такође, и ако би постојала таква TCP сесија која би могла да се успостави у више тачака истовремено, детекција губитака пакета ка неком од слушалаца и евентуална ретрансмисија пакета би изазвала непотребно слање дупликата пакета ка осталим слушаоцима који нису искусили губитке.
- Коришћење искључиво UDP протокола даље значи да мултикаст не може да се ослања на механизме који обезбеђују поуздани пренос и детекцију загушења. То не значи да је поузданији пренос од онога што омогућава UDP немогућ. Постоје предлози како може да се повећа поузданост мултикаст преноса:
  - кроз слање уникастом негативне потврде за оне пакете који нису стигли на дестинацију, на које би одговорили неки рутери између извора мултикаст саобраћаја и слушалаца [5.6],
  - хијерархијску поделу слушалаца мултикаста на групе где код сваке групе постоји један (*designated receiver*) задужен за слање потврда о поузданом пријему [5.7] или

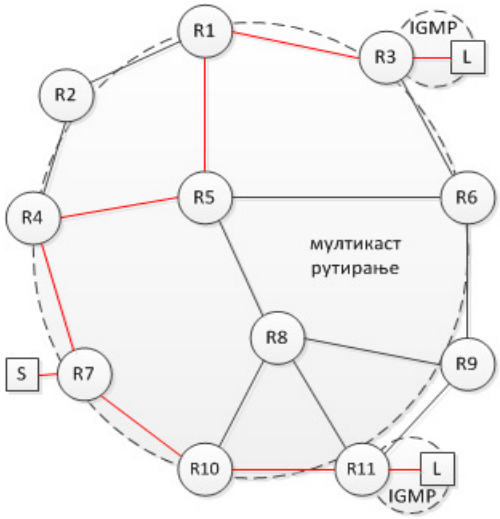
- методе код којих слушалац који не добије неки пакет може да га тражи од других слушалаца истог мултикаст садржаја [5.8].

- С обзиром на то да је начин комуникације у мултикасту од једне до више тачака истовремено, механизми заштите података описани у поглављу 3.3.2 су тешко применљиви. Наиме, један од основних елемената сигурне комуникације је размена кључева која се најчешће ослања на Дифи-Хелман размену која је по својој природи таква да може да се спроводи само између две тачке. Ипак, постоји јасна (и комерцијална) потреба за заштитом мултикаст саобраћаја: нпр. енкрипција видео садржаја који би могли да декриптују само они слушаоци који су платили приступ. Због тога су направљени предлози екстензија ових протокола у правцу пружања заштите пакета које треба да прими динамична група слушалаца којој могу да се додају нови слушаоци или из које могу да се одјављују слушаоци и то тако да могу да виде само онај садржај у време када су били чланови групе. Постоји више варијанти размене кључева, на пример:

  - алгоритми размене кључева коришћењем графова кључева и симетричних алгоритама који омогућавају ефикасну размену кључева у ситуацијама честих одлазака и долазака слушалаца [5.9] или

  - стандардизована метода која подразумева повезивање на сервер кључева/контролер групе који дистрибуира кључеве свим слушаоцима мултикаста коришћењем механизама асиметричне криптографије [5.10] [5.11].

Циљ мултикаст начина преноса пакета је да се кроз рачунарску мрежу успостави ток мултикаст пакета између извора S и свих слушалаца L по оптималним путањама како је показано на слици 5.3.



Слика 5.3 Кључни механизми за остваривање мултикаста

За ово је потребно је да буду дефинисани следећи кључни елементи мултикаст архитектуре:

- адресе мултикаст група на које се слушаоци пријављују,
- протокол којим слушаоци јављају непосредно повезаним рутерима да желе да примају (или да престану да примају) мултикаст пакете одређене мултикаст групе (IGMP протокол) и
- протокол којим рутери међусобно размењују информације о томе где је потребно проследити мултикаст пакете како би се омогућио транспорт пакета до свих слушалаца мултикаста по оптималним путањама (мултикаст протоколи рутирања PIM DM и SM)

Ови елементи су детаљно објашњени у наставку текста.

5.2.1. Врсте мултикаст дистрибуције пакета

Постоје две врсте мултикаст дистрибуције пакета:

- мултикаст код кога није специфициран извор мултикаст пакета - *Any-Source Multicast* (ASM). Код ове врсте мултикаста слушалац мултикаста се пријављује да прима пакете намењене мултикаст групи G и том приликом не одређује извор од ког жели да прима пакете. Код ове врсте мултикаста може да се деси да пакети слушаоцима стижу од више различитих извора истовремено.
- мултикаст код кога је тачно специфициран извор мултикаст пакета - *Source-Specific Multicast* (SSM). Код ове врсте мултикаста слушалац мултикаста се пријављује да прима пакете намењене мултикаст групи G и том приликом тачно одређује извор S од ког жели да прима пакете.

5.2.2. Мултикаст адресе

5.2.2.1. IPv4

У оквиру IPv4 одвојена је посебна класа адреса за мултикаст. То су адресе класе D, које све почињу са 1110 бинарно, односно све адресе од 224.0.0.0 до 239.255.255.255. Сви бити у мултикаст адреси након почетна 4 представљају адресу мултикаст групе на коју се шаљу мултикаст пакети и за које се пријављују слушаоци, што значи да овај адресни опсег дозвољава креирање 2^{28} односно приближно 268 милиона мултикаст група. Мултикаст адресним простором, као и уникаст адресама управља IANA, а у време када је дефинисана класа D адреса још увек није постојао концепт мултикаста са тачно специфицираним извором мултикаст пакета (SSM). Мултикаст адресни простор није униформан, већ је подељен на неколико сегмената од којих су најзначајнији:

- Резервисане адресе са дометом од једног мрежног сегмента (енг. *Reserved Link-Local Addresses*). Ово су адресе у опсегу 224.0.0.0-224.0.0.255 и користе се највише за неке

добро познате протоколе (нпр. OSPF – 224.0.0.5 и 224.0.0.6, RIP, итд.) или за слање пакета свим уређајима (224.0.0.1) или свим рутерима (224.0.0.2) на некој локалној мрежи. Домет од једног мрежног сегмента значи да пакети којима је дестинациона адреса у овом опсегу не пролазе кроз рутере и да им је TTL подешен на вредност 1 када се шаљу.

- Мултикаст адресе са глобалним дометом (енг. *Globally Scoped Multicast Addresses*) су адресе у опсегу 224.0.1.0-238.255.255.255. Ове адресе су потпуно аналогне јавним уникаст IP адресама, дакле то су адресе за које када се у пакетима налазе као дестинационе адресу моге да се рутирају и могу да прелазе између аутономних система и да имају произвољну TTL вредност. Да би се добила нека од ових адреса, односно право да се емитује мултикаст саобраћај по овим адресама, оне треба да се затраже од IANA. У оквиру овог адресног опсега одвојене су две посебне групе:
  - Адресе које се користе за неке добро познате протоколе на интернету (*Internetwork Control Block*) које имају опсег од 224.0.1.0 – 224.0.1.255. Један од ових протокола је NTP (енг. *Network Time Protocol*) 224.0.1.1, који се користи за дистрибуцију тачног времена преко рачунарских мрежа.
  - Адресе за SSM које имају опсег 232.0.0.0-232.255.255.255. Пошто је SSM дефинисан након што је извршена подела адреса на класе, накнадно је одвојен овај опсег.
  - GLOP адресе које имају опсег 233.0.0.0-233.255.255.255. Ове адресе су намењене за слободно коришћење од стране аутономних система без тражења дозволе од IANA како би се олакшала могућност коришћења мултикаста. GLOP адресе се формирају тако што се у средња два октета IP адресе упише број аутономног система, док је последњи октет остављен аутономном систему да може да креира 256 различитих мултикаст група, што може да се покаже на следећем примеру: За аутономни систем 62010, односно F23A хексадецимално се у други октет GLOP адресе уписује вредност F2 што је 242, а за трећи октет вредност 3A односно 58. Тако овај аутономни систем добија свој опсег мултикаст адреса које може да користи без тражења дозволе од IANA: 233.242.58.0/24.
- Адресе ограниченог домета (енг. *Limited Scope Addresses*) су адресе у опсегу 239.0.0.0-239.255.255.255. Ове адресе су аналогне приватним уникаст адресама – могу слободно да се користе унутар неког аутономног система и да се рутирају унутар њега, али не могу да изађу ван аутономног система. Аутономни системи имају пуну слободу у избору и коришћењу ових адреса.

5.2.2.2. IPv6

IPv6 мултикаст адресе почињу са FF хексадецимално. Након овог првог бајта у другом бајту прва 4 бита су флегови који ближе одређују врсту мултикаст адресе, а након њих долазе 4 бита који одређују домет (енг. *scope*) мултикаст адресе.

Од 4 бита за флегове, најзначајнији је за сада резервисан, а наредни бити су R, P и T. Бит R када је сетован, то значи да је адреса *Rendezvous-Pointa*[44] уписана у мултикаст IPv6 адресу. Када је сетован бит P, то значи да је у мултикаст адреси уписана адреса префикса дате мреже, а када је сетован бит T, то значи да је у питању динамички додељена мултикаст адреса.

Домет мултикаст адресе може да буде:

- 1 – локалан за дати интерфејс – пакет са овом адресом не може да напусти уређај и ове адресе се користе као *loopback* адресе – означавају сам уређај
- 2 - локалан за дату везу између два уређаја (попут резервисаних адресе са дометом од једног мрежног сегмента за IPv4) – пакети са овим адресама се не рутирају
- 4 – *Admin-local* – најмањи домет који може да дефинише администратор. Мора да буде мањи од осталих дефинисаних домета.
- 5 – *Site local* – домет једне локалне мреже једне организације
- 8 – *Organization local* – пакети чији домет може да буде унутар мреже једне организације која обухвата више локација. Одговара адресама ограниченог домета код IPv4.
- E – глобални домет – пакети са овим адресама могу да се рутирају без икаквих ограничења.

Неки примери IPv6 мултикаст адреса:

- *unicast-based multicast address*: ff3e:40:2001:db8:cafe:1:11ff:11ee. Ова адреса има глобални домет (E) и флегове са вредношћу 0011 (3) што значи да је у ову мултикаст адресу уписан уникаст префикс (сетован бит P). Овај префикс је у датом примеру 2001:db8:cafe:1/64, док је ознака мултикаст групе у доња 32 бита адресе: 11ff:11ee. Убацивање уникаст префикса у мултикаст адресу је еквивалентан убацивању броја аутономног система код GLOP адреса. Тиме се обезбеђује да организација која је власник одређеног адресног простора може да слободно креира мултикаст групе без опасности да се преклопе са мултикаст групама других аутономних система и без обавезе да се тражи дозвола за коришћењем од IANA.
- *embedded RP multicast address*: ff78:540:2001:db8:cafe:1::645. Ова адреса у односу на претходну има смањен домет на домет организације (8) и додатно сетован R флег што значи да се у њој налази адреса *Rendez-vous Point*-а и она је: 2001:db8:cafe:1::5. Ознака мултикаст групе је 645. У наставку текста ће бити објашњен значај слања адресе *Rendez-vous Point*-а.

44 Rendezvous Point је елемент архитектуре PIM SM протокола рутирања који је описан у поглављу 5.2.5.4.

5.2.3. Мултикаст адресе на слоју везе

Да би се формирао комплетан мултикаст пакет који ће бити послат, потребно је да се формира и заглавље слоја везе. То је данас најчешће етернет. Док је јасно да је изворишна MAC адреса, адреса уређаја који на дати мрежни сегмент шаље пакет, морало је да се некако реши питање дестинационе MAC адресе. У време када је Стивен Диринг у оквиру своје докторске дисертације осмислио концепт и прве механизме и протоколе потребне за мултикаст дистрибуцију, желео је да се свих 28 бита групне адресе преслика у MAC адресу. Како MAC адреса у прва 24 бита има идентификатор организације која је произвела мрежни картицу, а преостала 24 бита су слободна за доделу, да би постојао простор од 28 бита за мултикаст групу требало је да се купи 16 суседних индентификатора организације како би и доња 4 бита идентификатора организације могла да се искористе за упис 4 највиша бита мултикаст групе. Међутим, како је у то време мултикаст био само истраживачки пројекат једног студента докторских студија, његов Стенфорд универзитет је био спреман да купи само један идентификатор организације: 01:00:5E [5.12]. Такође, први бит у преостала 24 бита је дефинисано да је фиксиран на 0, тако да је преостало 23 бита која могу да се пресликавају из IP мултикаст адресе. Правило које је уведено је да се у доња 23 бита MAC адреса пресликвају доња 23 бита IP адресе. Ово значи да ће неке мултикаст групе имати исту MAC адресу (на пример групе 225.0.0.1 и 226.0.0.1 имају исту MAC адресу 01:00:5E:00:00:01). У ситуацијама када се у истој мрежи налазе слушаоци мултикаст група са идентичним мултикаст MAC адресама може да дође до нешто већег оптерећења оних рачунара који приме пакет са том MAC адресом и ураде његову почетну обраду на слоју везе, а који није заправо намењен њима.

Код IPv6 је предвиђено да се користе MAC адресе које почињу са 33:33 хексадецимално, а преостала 32 бита се попуњавају тако што се у њих пресликавају доња 32 бита IPv6 мултикаст адресе.

5.2.4. Пријављивање слушалаца мултикаст групе

5.2.4.1. IGMPv1

Протокол који се користи за пријављивање слушалаца на неку мултикаст групу је *Internet Group Management Protocol* (IGMP). Прва верзија IGMP протокола је настала у оквиру докторске дисертације Стивена Диринга којом је осмишљен мултикаст. IGMP поруке се енкапсулирају директно у IP заглавље и дужине су свега 64 бита (Слика 5.4). Једина два поља од значаја за рад протокола су адреса мултикаст групе за коју се врши пријављивање и поље које означава тип поруке. IGMPv1 има само два типа поруке:

- Упит (енг. *Membership Query*) којом рутери врше упит о томе да ли постоје слушаоци неке мултикаст групе на одређеном интерфејсу. Ове поруке се шаљу сваких 60s на адресу 224.0.0.1 (сви хостови на некој мрежи) и њима се поставља генерално питање:

„Да ли на овом мрежном сегменту постоји било који слушалац било које мултикаст групе?“

- Одговор (енг. *Membership Report*) којом слушаоци јављају да желе да примају пакете неке мултикаст групе. Одговори се шаљу по пријему Упита, али могу да се пошаљу и без Упита, у тренутку када се укључи апликација која треба да прими мултикаст пакете неке групе, како се не би чекао следеће слање Упита и како би мултикаст пакети почели што брже да долазе по укључењу апликације.

Слика 5.4 IGMPv1 заглавље

Регуларан процес пријављивања на неку мултикаст групу је следећи:

1. Рутери шаљу регуларни генерални Упит на 224.0.0.1 који долази до свих рачунара на неком интерфејсу рутера.
2. Рачунар који жели да прима пакете неке групе M шаље Одговор на дестинациону мултикаст адресу M, тако да ће одговор поред рутера који је послао упит добити и други рачунари који желе да примају пакете те исте мултикаст групе. Исто важи за слушаоце свих других мултикаст група. Одговор се шаље у тренутку који се одређује као случајан број између 0 и 10s. Ово је урађено да би се извело потискивање Одговора већег броја рачунара који желе да примају пакете исте мултикаст групе, зато што ће случајним одређивањем тренутка одговора сви осим првог рачунара који одговара одустати од слања одговора. Ово има једну значајну последицу – рутери неће знати колико има слушалаца неке мултикаст групе на одређеној локалној мрежи. То за њихово функционисање није ни битно, јер они морају да испоручују мултикаст пакете ка некој групи M без обзира на то колико има слушалаца.

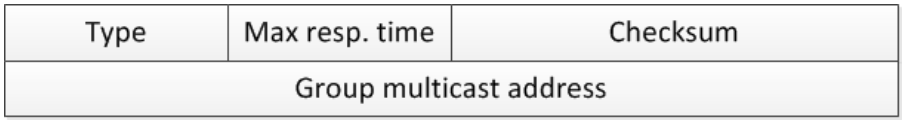
IGMPv1 нема дефинисан процес којим рачунари желе да сигнализирају да напуштају одређену мултикаст групу, већ се закључује да треба да се престане са слањем пакета на неки интерфејс уколико након три интервала слања Упита (180s) нема одговора за ту мултикаст групу. Овако дугачко време које је потребно да се престане са слањем мултикаст пакета је велика мана прве верзије IGMP протокола, посебно ако мултикаст ток пакета има велики проток информација – може да се деси да пуна три минута по престанку потребе за пријемом мултикаст пакета рутер их и даље прослеђује и оптерећује ресурсе мреже.

5.2.4.2. IGMPv2

Друга верзија протокола је пре свега требало да реши управо претходно наведене проблеме споре детекције одласка слушалаца мултикаст групе. Иако је IGMPv2 унапређена верзија прве, и очекивано је да је потпуно замени, није неуобичајено да се још увек у мрежи нађе

софтвер који прима мултикаст садржај који подржава или једну или другу или обе верзије протокола. Друга верзија протокола има више типова порука:

- Генерални упит (енг. *General query*) где се у поље са адресом уписују све нуле и којим се као и у првој верзији протокола пита: „Да ли на овом мрежном сегменту постоји било који слушалац било које мултикаст групе?“.
- Упит за специфичну групу (енг. *Group-specific query*) где се у поље са адресом уписује адреса групе M за коју се шаље упит и пита: „Да ли на овом мрежном сегменту постоји било који слушалац мултикаст групе M?“.
- Одговор који је као и у претходној верзији протокола. Одговори могу да буду типа верзије 1 или верзије 2, што је остављено због компатибилности са претходном верзијом протокола.
- *Leave* порука којом слушаоци сигнализирају да желе да напусте одређену мултикаст групу.



Слика 5.5 IGMPv2 заглавље

Такође, уведено је ново поље којим је дефинисано максимално време за одговор на упит (енг. *Maximum Response Time*) (Слика 5.5). Ово време је 10s за генерални, а 1s за упит за специфичну групу. Процес пријављивања на неку мултикаст групу је исти као у претходној верзији протокола, али је процес одјављивања из групе другачији:

1. Уколико неки рачунар жели да напусти одређену мултикаст групу M, послаће *Leave* поруку на адресу свих рутера: 224.0.0.2.
2. Пошто рутер не води рачуна о томе колико има слушалаца мултикаст групе на неком интерфејсу, он не може да зна да ли је приспела *Leave* порука дошла од последњег слушаоца групе M или не. Стога он мора да пошаље упит за специфичну групу M којим проверава да ли је преостао неки слушалац групе M.
3. Ако се на овај упит добије одговор, значи да је остало још слушалаца те мултикаст групе и мултикаст рутирање се наставља
4. Ако се на овај упит не добије одговор после два интервала максималног времена за одговор на упит, рутер закључује да је ово био последњи слушалац мултикаст групе и престаје да рутира мултикаст пакете ка групи M на тај интерфејс.

Ако на неком мрежном сегменту постоји више рутера, онда они морају да се договоре који ће бити задужен за слање упита за ту мрежу (који рутер је тзв. *Querier*). Ово је рутер са најнижом IP адресом на датом сегменту.

5.2.4.3. Остале верзије протокола

Постоји и трећа верзија IGMP протокола [5.13], али она не доноси побољшања у односу на IGMPv2 већ је направљена за подршку за пријављивање на мултикаст са специфичним извором мултикаст саобраћаја (SSM). Као што из заглавља IGMPv3 може да се види, протокол је проширен могућношћу да се специфицира један или више извора мултикаст саобраћаја на који слушалац може да се пријави. Такође, додати су још неки елементи којима се врши боља синхронизација рутера задужених за слање упита за ту мрежу (енг. *Querier*): QRV (*Querier's Robustness Variable*) и QQIC (*Querier's Query Interval Code*).

<table>
<tr><td colspan="3">Type</td><td>Max resp. code</td><td>Checksum</td></tr>
<tr><td colspan="5">Group multicast address</td></tr>
<tr><td>RESV</td><td>S</td><td>QRV</td><td>QCIC</td><td>Број извора</td></tr>
<tr><td colspan="5">Адреса извора 1</td></tr>
<tr><td colspan="5">Адреса извора 2</td></tr>
<tr><td colspan="5">...</td></tr>
<tr><td colspan="5">Адреса извора н</td></tr>
</table>

Слика 5.6 IGMPv3 заглавље

Протоколи који се користе за пријављивање на IPv6 мултикаст групе су MLDv1 и MLDv2 (енг. *Multicast Listener Discovery*) који су аналогни IGMPv2 и IGMPv3, дакле за ASM и SSM.

5.2.4.4. Мултикаст на локалним мрежама - IGMP snooping

Један од проблема који се јавља приликом слања мултикаст пакета по локалној рачунарској мрежи која је имплементирана стандардним свичевима јесте како обезбедити да мултикаст пакети долазе само до рачунара који желе да примају пакете дате мултикаст групе. Наиме, свичеви који раде на стандардан начин користећи механизме транспарентног бриџинга уче где се налазе рачунари са одређеним MAC адресама тако што приликом доласка етернет оквира преко неког порта памте изворишну MAC адресу и у MAC табелу бележе пар (порт, MAC адреса). Даље, када стигне пакет који је намењен MAC адреси која је у MAC табели, пакет бива послат само на порт који је у табели, а не на све портове, као што се ради приликом иницијалног рада свича који нема информацију о томе на ком порту се налази која MAC адреса (иницијални *flooding*). Међутим, мултикаст MAC адресе никада не могу да се нађу на позицији изворишне MAC адресе, што значи да свич неће моћи да научи где се налазе слушаоци неке мултикаст групе, а што даље значи да би сваки мултикаст пакет требало да буде послат на све портове свича. Овакав начин рада свичева је нерационалан посебно у ситуацијама када мултикаст пакети носе велике количине података (нпр. видео токови).

Да би се овакво понашање избегло, осмишљен је механизам IGMP *snooping* [5.14] којим свичеви ослушкивањем и праћењем IGMP порука могу да добију информацију о томе где се налазе слушаоци појединих мултикаст група. На основу IGMP упита могу да добију информацију о томе где се налазе рутери, а на основу IGMP одговора где су слушаоци појединих мултикаст група. Такође, пошто би механизам потискивања IGMP одговора спречио свич да добије информацију о томе где су сви слушаоци неке мултикаст групе, свич ће селективно пропуштати IGMP одговоре само према рутерима, али не и према другим слушаоцима мултикаст група, чиме ће се учинити да сви слушаоци неке мултикаст групе пошаљу своје одговоре.

5.2.5. Мултикаст протоколи рутирања

Подразумевано понашање рутера је да не пропуштају мултикаст пакете, већ је потребно да се ово експлицитно конфигурише. Након настанка концепта мултикаста предложено је више различитих протокола рутирања којима се омогућава прослеђивање пакета од извора до свих слушалаца унутар једног домена. То су биле екстензије дотаташњих уникаст протокола рутирања - екстензија RIP протокола рутирања на мултикаст - DVMRP (*Distance-Vector Multicast Routing Protocol*) или проширење OSPF за мултикаст (MOSPF), као и посебни нови протоколи као што је CBT (*Core Based Trees*). Међутим, на крају су се искристалисале две верзије једног протокола – PIM (*Protocol Independent Multicast*) као протокол који се искључиво користи за омогућавање рутирања мултикаст пакета и ове верзије ће детаљно бити описане у наставку текста.

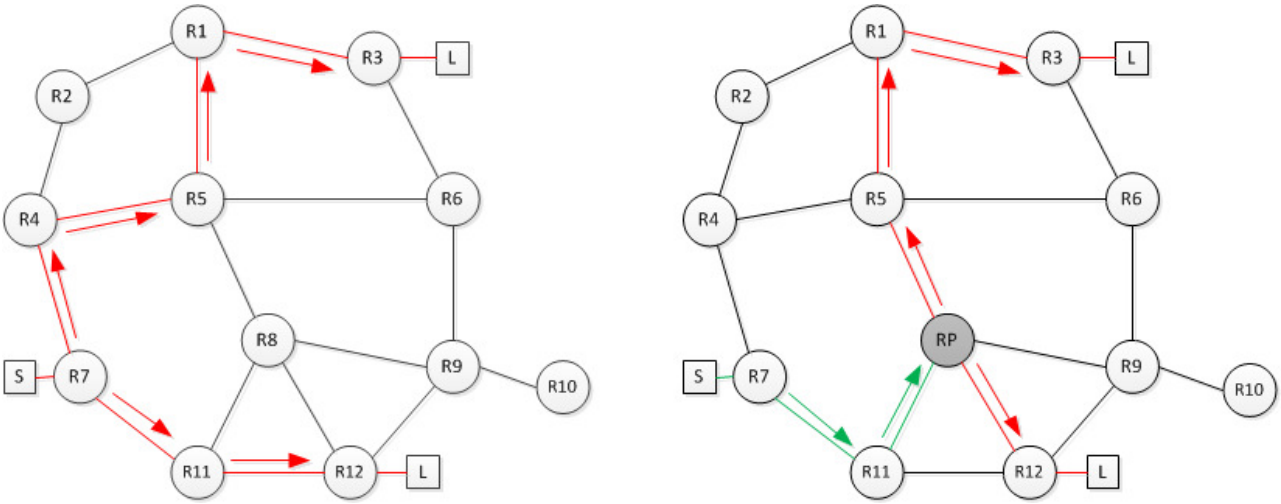
5.2.5.1. Мултикаст дистрибутивна стабла

Постоје две стратегије дистрибуције мултикаст пакета до свих слушалаца:

- по стаблу са кореном у извору односно по најкраћем стаблу (енг. *shortest-path tree*)
- по дељеном стаблу (енг. *shared-tree*)

Разлика између ове две стратегије је показана на слици 5.7. Код стратегије дистрибуције мултикаст пакета најкраћим стаблом, формирају се најкраће путање од извора мултикаст пакета до сваког слушаоца, као на левој страни Слике 5.7. Оваква стабла се означавају двојком (S,G) зато што је корен стабла у извору мултикаст пакета S, а гране су одређене најкраћим путањама у мрежи до сваког појединачног слушаоца групе G.

Са друге стране, код стратегије са дељеним стаблом одређује се једна тачка у мрежи која се назива *Rendez-vous Point (RP)* од које се формира стабло до свих слушалаца неке групе без обзира на локацију извора мултикаст саобраћаја. Оваква стабла се означавају двојком (*,G), зато што се стабло формира од RP до свих слушалаца групе G најкраћим путањама без обзира на локацију извора мултикаст пакета. Извор мултикаст саобраћаја шаље мултикаст пакете ка RP. Начин на који се то ради биће објашњен у оквиру објашњења PIM протокола.



Слика 5.7 Најкраће стабло (лево) и дељено стабло (десно)

У примеру са слике је очигледна предност стратегије са најкраћим стаблом: добиће се увек оптималне путање мултикаст пакета од извора до свих слушалаца. То код дељеног стабла није случај јер не постоји начин да се RP смести на оптималан начин тако да у свим комбинацијама распореда извора и слушалаца постоји оптимална путања. Са друге стране, дељена стабла захтевају мање рачунарских ресурса (пре свега меморије) од рутера – формираће се увек једно дељено стабло без обзира на број извора мултикаст пакета за дату мултикаст групу.

5.2.5.2. Прослеђивање на основу повратне путање (RPF)

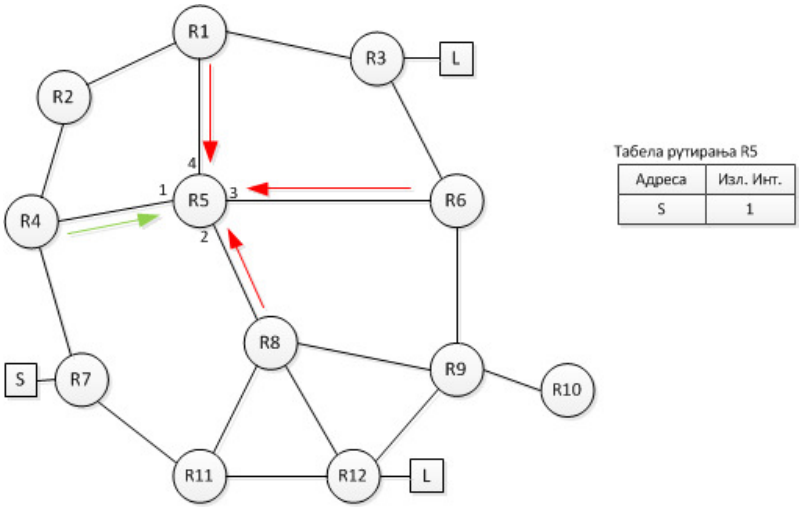
Уколико би се следила логика начина рада уникаст протокола рутирања, онда би неки мултикаст протокол рутирања требало да оглашава позиције слушалаца (дестинације мултикаст пакета) и да на основу тога и топологије мреже формира или најкраће или дељено стабло до свих слушалаца неке групе G. То и јесте начин на који раде неки старији мултикаст протоколи рутирања попут MOSPF који је специфицирао нове LSA пакете којима оглашава присуство слушалаца групе G на неком од интерфејса рутера. На основу ових LSA рутери могу да израчунају топологију мреже и најкраће путање до свих слушалаца. Међутим, овакав приступ није скалабилан: свака промена на неком интерфејсу којом се пријављује први или одлази последњи слушалац неке мултикаст групе G захтева слање LSA пакета, што даље повлачи ново израчунавање најкраћих путања по Дајкстра алгоритму који је рачунски интензиван. Такође, израчунавање једног стабла подразумева израчунавање *n* путања где је *n* број рутера који имају слушаоце групе G.

Са друге стране, битно је приметити следеће: у рачунарским мрежама везе између рутера су готово увек симетричних капацитета[45]. Ово значи да ако је одређена путања оптимална по

45 Асиметрични капацитети веза се срећу у приступном слоју мрежа, нпр. код кабловских или АДСЛ провајдера, али то није од значаја за ову дискусију, јер такве везе немају никакав утицај на одређивање путање кретања пакета (то су лист-везе у мрежи). Такође, овде се говори о рутирању унутар једног домена, тако да асиметричност путања услед административних конфигурација које омогућава BGP нису од значаја за ову дискусију.

некој метрици (нпр. капацитет број хопова или цена) од тачке А до тачке Б, одређена је и оптимална путања од тачке Б до тачке А. Ова чињеница је послужила за креирање посебног механизма који се користи приликом рутирања мултикаст пакета и одређивања оптималних стабала који се назива „прослеђивање на основу повратне путање“ (енг. *Reverse Path Forwarding* – RPF). Овај механизам функционише тако што се приликом приспећа мултикаст пакета у рутер не узима у обзир дестинациона адреса (адреса мултикаст групе) како би се донела одлука о прослеђивању, већ изворишна адреса пакета (повратна путања). На основу изворишне адресе (адреса извора мултикаст пакета) и на основу уникаст табеле рутирања се процењује да ли је дати пакет дошао са интерфејса који је на најкраћој путањи према извору мултикаст пакета.

- Ако је мултикаст пакет дошао преко интерфејса који је на најкраћој путањи према извору мултикаст саобраћаја, овај пакет је кандидат да буде прослеђен даље.
- Ако мултикаст пакет није дошао преко интерфејса који је на најкраћој путањи према извору мултикаст саобраћаја, овај пакет ће бити одбачен.



Слика 5.8 Прослеђивање на основу повратне путање

Ово је приказано на примеру на слици 5.8. За рутер R5 је дат скраћени приказ табеле рутирања за руту ка мрежи на којој се налази извор S. Најкраћа путања ка мрежи на којој је извор мултикаст пакета води преко интерфејса 1. Ово значи да ће рутер R5 моћи да пропусти само оне мултикаст пакете од извора S који су дошли преко интерфејса 1, јер ти пакети пролазе RPF критеријум, док ако би пакети дошли преко интерфејса 2, 3 и 4, морали би да буду одбачени. Овим механизмом се спречавају петље у дистрибутивном стаблу и формирање оптималних стабала што ће бити показано у поглављима у којима се описује PIM протокол.

RPF провера и чињеница да рутери већ имају израчунате уникаст табеле рутирања послужиле су за креирање PIM (енг. *Protocol Independent Multicast*) протокола за мултикаст рутирање. Сам назив протокола сугерише чињеницу да се мултикаст рутирање ослања на

уникаст таβелу рутирања и то βез оβзира на то из ког уникаст протокола рутирања су дате руте доβијене (*Protocol Independent*). PIM протокол има две варијанте:

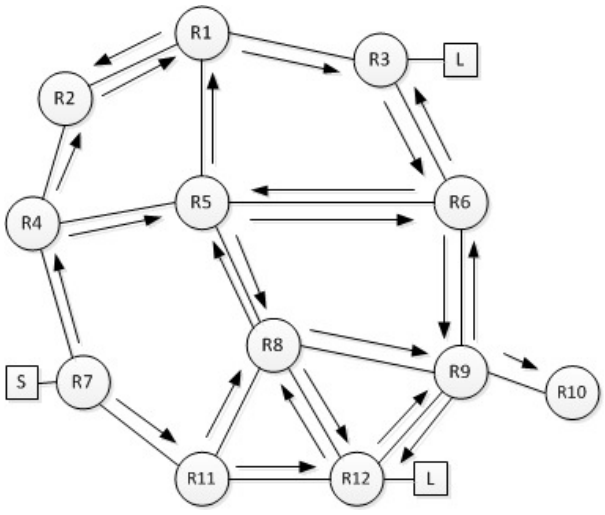
- PIM *Dense Mode* (DM) који креира најкраће стаβло са кореном у извору мултикаст пакета и
- PIM *Sparse Mode* (SM) који креира дељено стаβло са кореном у RP.

PIM мора да β уде конфигурисан на рутерима на оним интерфејсима преко којих су повезани остали рутери који учествују у дистриβуцији мултикаст пакета. Рутери остварују PIM суседске односе и размењују поруке којима се формирају дистриβутивна стаβла што ће β ити описано у наставку текста, а комуникација се остварује преко мултикаст адресе 224.0.0.13.

5.2.5.3. PIM DM

PIM *Dense Mode* формира најкраће стаβло са кореном у извору мултикаст саоβраћаја користећи методу која се зове *„flood and prune”* (β уквално преведено „преплави и поткреши“). Ово значи да се периодично (нпр. свака 3 минута) мултикаст пакети неке групе прослеђују по свим везама и до свих рутера βез оβзира на то да ли постоје слушаоци неке мултикаст групе на њима (*flood* фаза), а након тога се мултикаст саоβраћај уклања са оних веза на којима није потреβан.

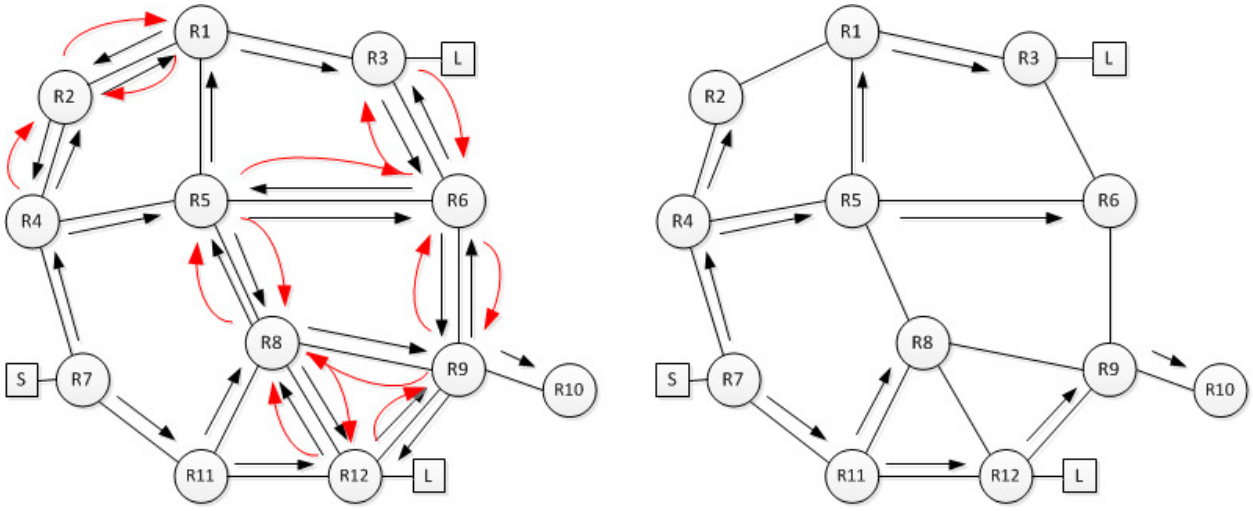
Фаза преплављивања је приказана на слици 5.9: пакети који дођу на интерфејс рутера који испуњава RPF проверу за дати извор S ће се проследити на све остале интерфејсе рутера на којима су PIM суседи. Ово омогућава да мултикаст пакети од извора S намењени групи G стигну сигурно до свих слушалаца ове групе. Међутим, као што се види на слици, овакво понашање рутера доводи до неоптималног искоришћења мреже: мултикаст пакети на некој вези могу да пролазе у оβа смера, могу да постоје дупликати пакета и мултикаст долази и у оне делове мреже у којима нема слушалаца.



Слика 5.9 Фаза преплављивања мреже мултикаст пакетима

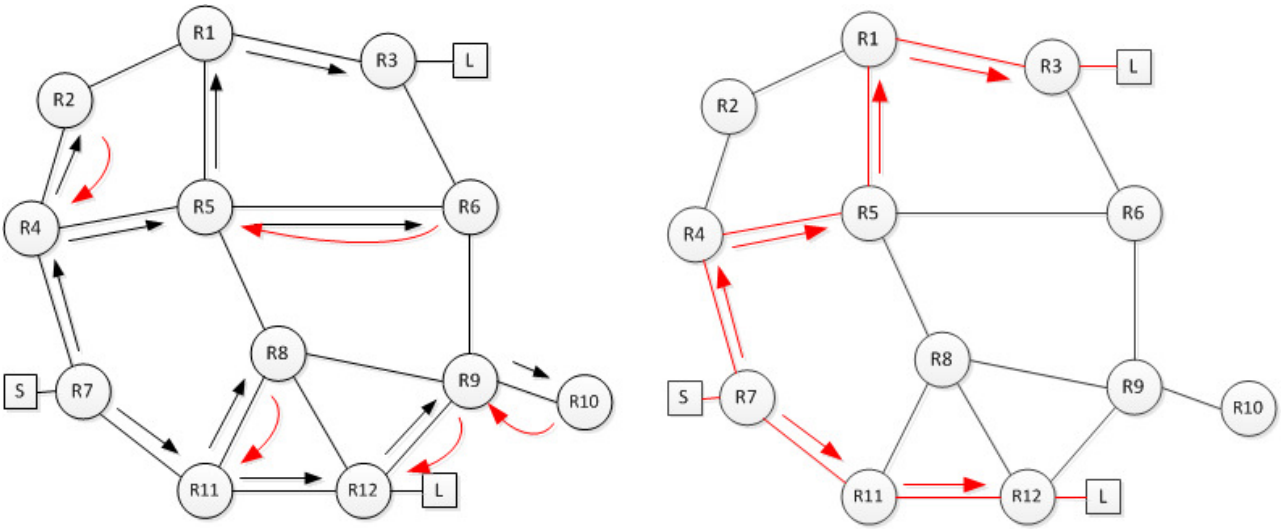
Након што су пакети дошли до свих слушалаца и свих рутера, у фази поткресивања (енг. *prune*) се уклања мултикаст саобраћај са свих линкова на којима није потребан. Ово се врши PIM *Prune* порукама које рутери шаљу једни другима чиме захтвају престанак слања мултикаст пакета преко тог интерфејса, а постоји неколико критеријума за поткресивање:

1. Уколико је пакет дошао преко интерфејса који није задовољио RPF проверу, што је показано на слици 5.10. Црвене стрелице означавају PIM *Prune* поруке за оне ситуације када се реагује на приспеле мултикаст пакете који нису прошли RPF проверу. Са десне стране је показан преостали пут мултикаст пакета који сада долазе до свих рутера, али што и даље није оптималан начин дистрибуције јер мултикаст пакети и даље долазе до оних рутера којима не требају.



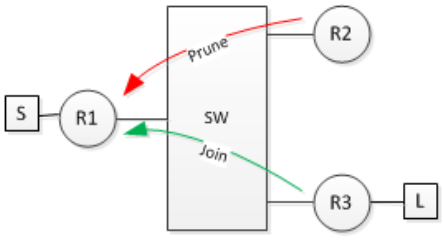
Слика 5.10 Поткресивање путања пакета које не пролазе RPF проверу

2. Ако лист-рутер нема слушаоце дате групе (ни један рачунар се IGMP поруком није пријавио на групу G). Лист-рутер који нема слушаоце у овој ситуацији је R10 и он ће такође послати PIM Prune ка R9, што је показано на слици 5.11.
3. Ако је не-лист рутер добио PIM Prune поруке од свих својих PIM суседа и нема слушаоце мултикаст група повезане на себе. Ово је случај рутера R2, R6, R8 и R9, што је приказано исто на слици 5.11. На тој слици са десне стране је коначна топологија дистрибуције мултикаст пакета. Као што може да се види, применом ових принципа одсецање мултикаст саобраћаја са оних грана на којима није више потребан може да се добије оптимална топологија мултикаст дистрибуције са кореном стабла у извору S.



Слика 5.11 Поткресивање веза којима није више потребан мултикаст

4. Ако је не-лист рутер добио PIM Prune поруке од суседа на LAN интерфејсу и није добио поруку којом се поништава овај захтев (тзв. *Prune Override* који је реализован PIM *Join* поруком). Овај специфичан случај који се јавља у ситуацији када је више мултикаст рутера повезано на један LAN сегмент (свич) приказан је на слици 5.12. Рутер R2 нема слушаоце мултикаст групе G и нема потребу да прима више мултикаст пакете. Он шаље PIM *Prune* поруку ка рутеру R1 који прослеђује мултикаст пакете од извора S. Рутер R1 не сме у овој ситуацији да безусловно престане да шаље мултикаст пакете ка мрежи на којој су R2 и R3, јер би тиме слушалац који је повезан на R3 остао без мултикаст пакета. Због тога се по примању *Prune* поруке стартује на R1 тајмер дужине 3s током кога се чека да ако постоји други рутер коме су и даље потребни мултикаст пакети, да пошаље PIM *Join* поруку којом ће поништити претходни PIM *Prune* захтев. Рутер R3 ће знати да треба да поништи *Prune* захтев зато што је и он добио овај захтев пошто се емитује преко адресе 224.0.0.13 те стиже и до R3.



Слика 5.12 Prune Override

На слици 5.13 приказан је пример једног улаза у мултикаст табелу рутирања на коме се види да постоји један долазни интерфејс који је на путањи ка извору мултикаст саобраћаја (GigabitEthernet0/3), као и два интерфејса на које рутер може да пошаље пакете (GigabitEthernet0/1 и GigabitEthernet0/2). Од та два интерфејса један је добио Prune поруку (GigabitEthernet0/2) и на њега се неће слати мултикаст пакети који иду ка групи 239.1.1.1.

```
(192.168.1.1, 239.1.1.1), 00:01:16/00:01:43, flags: T
  Incoming interface: GigabitEthernet0/3, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet0/1, Forward/Dense, 00:01:16/stopped
    GigabitEthernet0/2, Prune/Dense, 00:01:15/00:01:43
```

Слика 5.13 Пример улаза у мултикаст табелу рутирања код PIM DM

Приликом процеса одсецања мултикаст саобраћаја на некој од веза неће су избрисати улаз у табели рутирања на том рутеру ни онда када више нема интерфејса на које могу да се пошаљу мултикаст пакети. Ово је урађено како би слушаоци могли накнадно, након што је веза одсечена да се пријаве за слушање мултикаста на истој групи и како би могло да се прошири мултикаст стабло и за њих, а да не мора да се чека следећи интервал плављења мреже мултикаст пакетима. То се ради PIM *Graft* порукама. У мрежи са слике 5.11, уколико би се појавио слушалац повезан на рутер R2, рутер R2 би по добијању IGMP *Report* поруке послао PIM *Graft* поруку ка рутеру који је на путањи ка извору мултикаст саобраћаја (R4) што би изазвало проширење мултикаст стабла за нову грану (R2-R4) и слање пакета ка новом слушаоцу.

Уколико постоји више рутера који могу равноправно да испоручују мултикаст пакете на одређени мрежни сегмент (имају исту метрику до извора мултикаста), онда се одлучује који ће рутер бити тај који шаље мултикаст пакете на дати сегмент на основу IP адресе рутера на том сегменту – онај ко има већу IP адресу ће прослеђивати пакете.

PIM DM је погодан за коришћење у оним мрежама у којима су слушаоци мултикаст група релативно густо распоређени (одатле и име ове верзије протокола), тако да на готово сваком рутеру има слушалаца мултикаст групе. У таквим ситуацијама периодично плављење мреже мултикаст пакетима за дату групу не би изазвало непотребно слање велике количине пакета ка рутерима и по везама којима иначе ти пакети нису потребни.

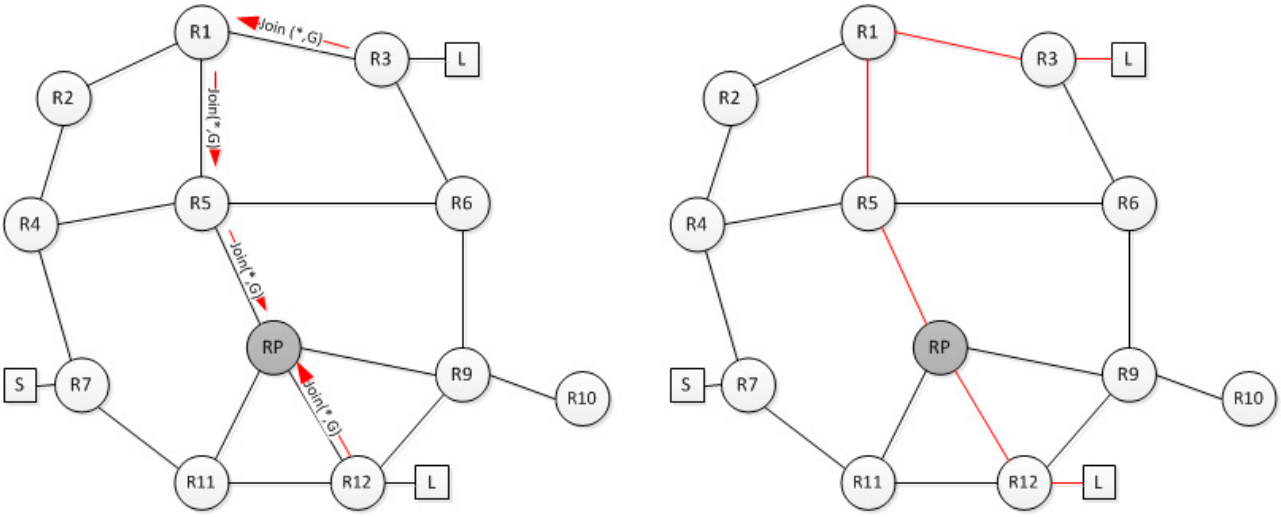
5.2.5.4. PIM SM

PIM *Sparse Mode* формира дељено стабло са кореном у RP користећи тзв. *pull* начин рада којим се мултикаст пакети доводе до рутера само онда када за њима постоји потреба. Код ове варијанте протокола рутирања постоје две независне активности:

1. Пријављивање слушалаца
2. Слање мултикаст пакета

Пријављивање слушалаца се обавља PIM *Join* порукама које се шаљу од рутера који је добио IGMP *Report* поруку од неког од рачунара који жели да прима мултикаст саобраћај G. Рутери ове поруке прослеђују према RP рутеру, након чега се формира дељено (*,G) стабло што је приказано на слици 5.14. Уколико би се неки од слушалаца одјавио од примања пакета одређене мултикаст групе, онда би рутери слали према RP PIM *Prune* поруке чиме би се

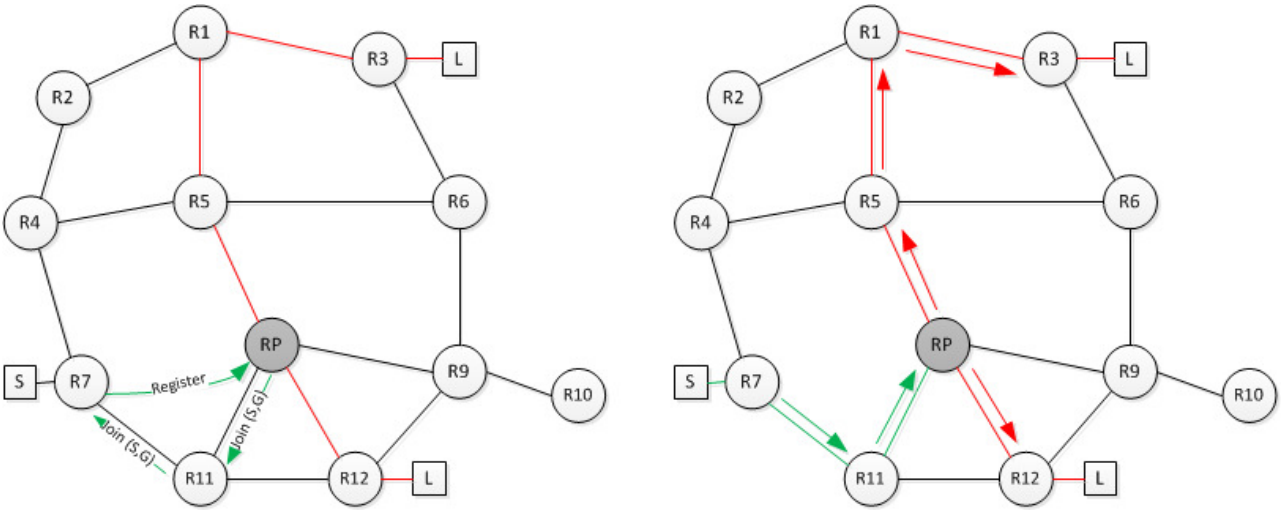
евентуално одсекле неке гране у дељеном стаблу. Дељено стабло мора да одржава своје стање тако што се захтева да се *Join* поруке шаљу периодично сваког минута. Уколико после 3 минута нема више *Join* порука, та грана стабла ће бити избрисана.



Слика 5.14 Пријављивање слушалаца и формирање дељеног стабла

Извор мултикаст пакета почиње да шаље пакете, без икакве интеракције са рутерима у мрежи. Рутер на који је повезан извор (у овом случају R7) прима мултикаст пакете и шаље их према RP енкапсулиране у уникаст (са IP адресама извора R7 и дестинације RP) PIM *Register* поруке. Уколико за мултикаст групу на коју се шаљу пакети RP нема регистрованих слушалаца у тренутку када добије PIM *Register* поруку, RP враћа ка R7 *Register Stop* поруку којом се рутеру R7 јавља да не постоје слушаоци дате мултикаст групе и захтева да престане да шаље пакете за ту групу. R7 ће то и учинити, при чему извор мултикаст саобраћаја неће имати информацију о томе. Уколико за мултикаст групу на коју се шаљу пакети RP има регистрованих слушалаца у тренутку када добије PIM *Register* поруку, RP ће започети успостављање најкраћег стабла од RP ка R7, што је показано на слици 5.15.

У том тренутку ће бити формирано мултикаст дистрибутивно стабло које се састоји из два дела: од извора до RP и од RP до свих слушалаца. Ово стабло очигледно није оптимално те је стога у оквиру PIM SM имплементиран и механизам пребацивања на најкраће стабло (SPT *Switchover*). Рутери на које су повезани слушаоци (у овом случају R3 и R12) немају информацију о томе који је извор мултикаст пакета ка групи G све док не добију прве пакете од тог извора. Од тог тренутка могу да иницирају процес пребацивања на најкраће стабло тако што ће почети да шаљу PIM *Join* поруке према извору S којима желе да се пријављиве да примају мултикаст пакете по стаблу (S,G). Када ове поруке пропагирају до рутера R7, формираће се најкраће стабло као на слици 5.7 (лево).



Слика 5.15 Слање мултикаст пакета: Енкапсулација у Register поруке (лево) и формирано стабло (десно)

Овим процесом пребацивања на најкраће стабло ће се добити исто дистрибутивно стабло као код PIM DM, али уз избегавање периодичног плављења мреже мултикаст пакетима које PIM DM носи са собом. Због овога, као и због тога што је пракса показала да су слушаоци мултикаст група релативно ретко распоређени, данас се у рачунарским мрежама доминантно користи PIM SM као основни протокол рутирања. На слици 5.16 је приказан садржај мултикаст табеле рутирања за PIM SM и групу 233.233.233.233 после пребацивања на најкраће стабло.

```
(*, 233.233.233.233), 00:00:39/stopped, RP 172.16.11.1, flags: SPF
  Incoming interface: FastEthernet0/0, RPF nbr 192.168.45.4
  Outgoing interface list: Null

(192.168.56.6, 233.233.233.233), 00:00:39/00:03:08, flags: FT
  Incoming interface: FastEthernet1/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/1, Forward/Sparse, 00:00:39/00:02:50
    FastEthernet0/0, Forward/Sparse, 00:00:41/00:02:48
```

Слика 5.16 PIM SM дељено и дистрибутивно стабло после пребацивања на најкраће стабло

5.2.5.5. Аутоматска детекција RP

Претходно описани процес формирања PIM SM стабла указује на то да сви рутери у PIM SM мрежи морају да знају адресу RP рутера пре него што почне успостављање мултикаст стабала и дистрибуција пакета. Најједноставнији начин да се ово оствари је статичко конфигурисање IP адресе RP на свим рутерима у мрежи. Међутим, пошто је RP по својој улози у мрежи уређај чијим отказом се угрожава целокупно функционисање мреже (SPoF), статичким конфигурисањем RP рутери не би могли да се адаптирају на такве околности, те је потребно да се користи неки аутоматизовани начин за конфигурацију RP на рутерима. Постоји неколико начина да се ово уради.

Auto-RP

Auto-RP је техника коју је развила компанија Cisco и која је карактеристична само за њихове уређаје. У мрежи је могуће да више рутера има улогу RP. Такође, могуће је да сваки од тих RP подржава неки подскуп мултикаст група како би се растеретило оптерећење (нпр. RP1 је задужен за све групе од 224.0.1.0 до 230.0.0.0, а RP2 за све групе од 230.0.0.0 до 239.255.255.255). Предвиђено је да RP рутери шаљу информацију о томе за које су мултикаст групе задужени слањем RP-*Announce* порука на адресу 224.0.1.39. Ове поруке долазе до рутера у мрежи са улогом MA (енг. *Mapping Agent*) који скупљају огласе RP и шаљу их свим рутерима RP-*Discovery* порукама на мултикаст адресу 224.0.1.40. По добијању ових порука рутери могу да аутоматски конфигуришу RP.

Проблем са овом методом је то што она захтева да се пакети са RP-*Announce* и RP-*Discovery* порукама дистрибуирају кроз мрежу мултикастом пре него што су на рутерима конфигурисани RP рутери, а ове мултикаст групе спадају у оне које се регуларно рутирају. Због овога је Cisco осмислио посебан режим рада, тзв. PIM SDM (*Sparse-Dense Mode*) код кога је могуће да се за неке посебне групе користи PIM DM, а то би у овом случају биле групе које су потребне за дистрибуцију огласа о адресама RP рутера.

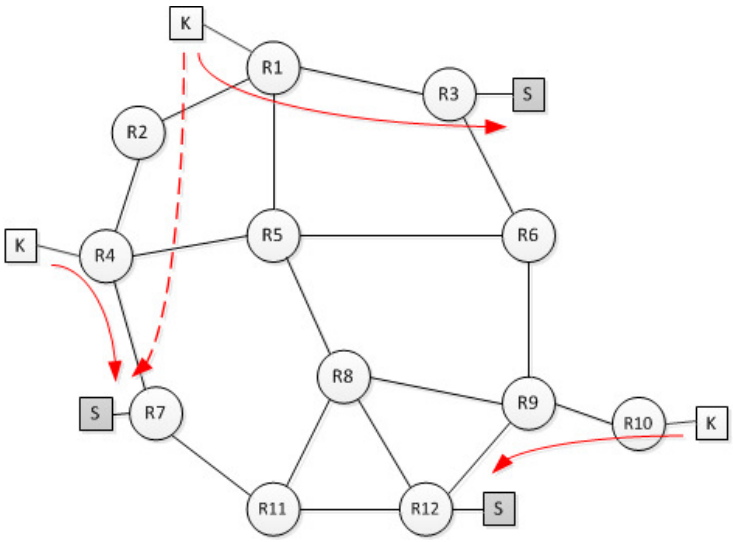
Bootstrap Router

Bootstrap Router [5.15] је стаднардизовани механизам за дистрибуцију информације о RP рутерима. Неки рутери у мултикаст домену се конфигуришу тако да буду кандидати за RP, од којих ће подскуп бити RP рутери у мрежи (c-RP). Такође неки рутери у домену се конфигуришу тако да буду кандидати за *Bootstrap* рутере (c-BSR), а један од њих ће бити *Bootstrap* рутер задужен за домен – онај који има највиши конфигурисан приоритет. Слично као у претходној методи, c-RP рутери шаљу информацију о томе за које су мултикаст групе задужени, као и конфигурисане приоритете. Ове поруке се шаљу ка BSR уникастом. Након што је добио све огласе кандидата RP, BSR рутер оглашава информацију о RP рутерима и мапирање у мултикаст групе на сваком од њих на адресу 224.0.0.13. Ова адреса спада у адресе које се не рутирају, па је подразумевано понашање рутера када добију ове поруке да раде дистрибуцију поруке ка свим суседима (*flooding*) на начин како се то ради код нпр. OSPF протокола. Пошто се за дистрибуцију оглашавања информација о RP користе уникаст и адресе са дометом од једног мрежног сегмента, не постоји проблем дистрибуције ових информација пре добијања информације о томе где су RP рутери.

Anycast са MSDP

Еникаст (енг. *Anycast*) је механизам слања пакета код ког постоји више уређаја са идентичном функционалношћу, а који су смештени на различитим тачкама у мрежи и имају исту IP адресу. Како се иста адреса оглашава са више различитих тачака у мрежи, протокол рутирања ће на основу метрике одредити који је уређај са датом адресом најближи из сваке посматране тачке у мрежи (Слика 5.17). У примеру са слике постоје три сервера S са истом IP адресом и три корисника, који сви комуницирају са оним сервером који им је по метрици најближи. Еникаст је механизам којим је природно обезбеђена отпорност на отказе и то са статичком конфигурацијом уређаја, јер би се отказом једног од уређаја (у примеру са слике је

то сервер повезан на R3) и престанком његовог оглашавања пакети прерутирали са уређаја са којим је до тада δила остваривана комуникација на први следећи најδлижи по метрици (у примеру сервер повезан на R7). Зδог могућности преδацивања на други уређај у сред комуникције, еникаст је погоднији за коришћење у ситуацијама када се користи UDP него TCP, јер контекст TCP сесије не δи могао да δуде запамћен и настављен у случају преδацивања.



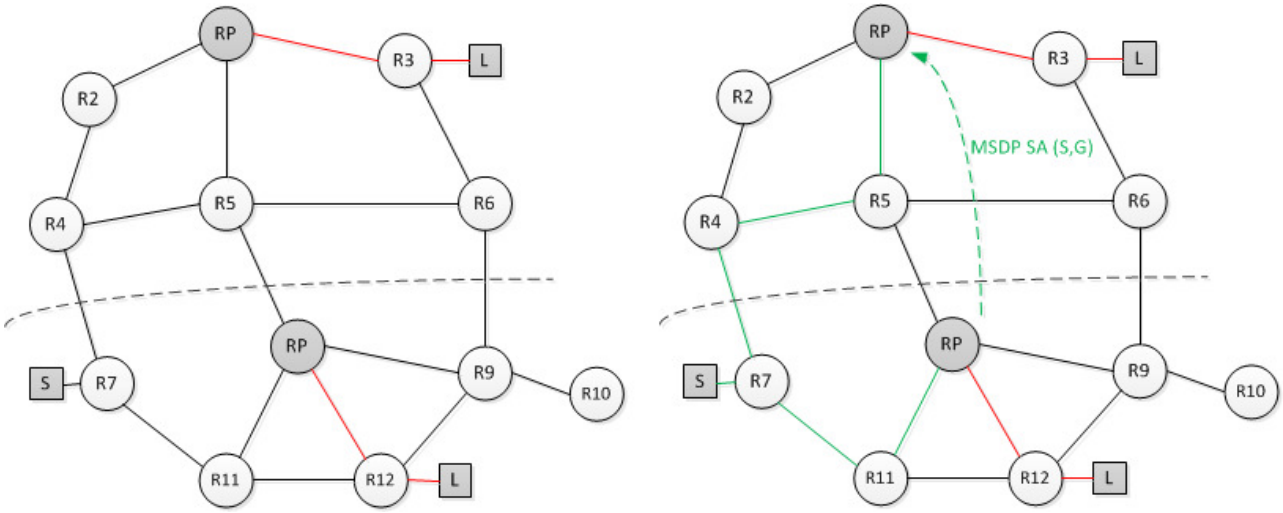
Слика 5.17 Anycast начин комуникације

Овај механизам се данас користи нпр. у оквиру DNS система где root DNS сервери имају исте IP адресе. K-root сервери којима управља европски RIR RIPE имају адресу 193.0.14.129 и налазе се на 59 локација широм света (податак из децемδра 2017 [5.16]). Све ове адресе оглашава посеδан аутономни систем направљен за ову сврху – AS25152, како δи се изδегла ситуација која није дозвољена на интернету да неки аутономни систем оглашава адресе које му не припадају. Унутар аутономних система нема никаквог ограничења у коришћењу принципа еникаста.

Примена овог механизма за конфигурацију RP отпорну на отказе је приказана примером са Слике 5.18.

На левој страни слике је показана мрежа у којој постоје два RP са истом IP адресом. Претпоставићемо да су горњем RP по метрици δлижи рутери R2, R3, R4, R5 и R6, а доњем рутери R7, R9, R10, R11 и R12. Скупови рутера који су δлиски једном или другом RP су на слици раздвојени испрекиданом линијом. На свим рутерима је конфигурисана иста статичка адреса RP рутера, али ће зδог различите δлизине RP рутерима δити формирана дељена стаδла ка два RP рутера у ситуацији када се слушаоци налазе повезани на R3 и R12. Када извор мултикаст пакета почне да шаље пакете према рутеру R7, он ће према правилима описаним у претходном поглављу ове пакете послати према доњем RP (који му је δлижи по метрици), те ће мултикаст саоδраћај моћи да се оствари δез проδлема према слушаоцу повезаном на R12. Овако организована мрежа је подељена на два независна домена и горњи RP према до сада описаним механизмима нема могућност да доδије информацију о томе да

постоји извор мултикаста за групу на коју он има регистроване слушаоце. Због тог је уведен нови протокол – MSDP (енг. *Multicast Source Discovery Protocol*) којим RP рутери могу да се обавештавају о постојању извора мултикаст саобраћаја. Када доњи RP добије *Register* поруку од рутера R7, поред успостављања најкраћег стабла према извору послаће ка горњем RP MSDP *Source Active* (SA) поруку којом ће га обавестити да је примио мултикаст пакете од извора S ка групи G. По добијању информације о постојању извора мултикаста и горњи RP рутер ће моћи да иницира успостављање најкраћег стабла до извора мултикаста чиме ће моћи да се успостави ток пакета и до слушаоца повезаног на R3. MSDP је један од кључних протокола у успостављању међудоменског мултикаста (мултикаста између аутономних система) и биће мало детаљније описан у следећем поглављу.



Слика 5.18 Редунданtни RP уз коришћење Anycast механизма

5.2.6. Мултикаст између аутономних система

У првим годинама примене мултикаста формирана је MBone мултикаст мрежа. Ова мрежа се састојала од низа уникаст тунела између тачака (сервера) који подржавају мултикаст, углавном у истраживачким организацијама и на универзитетима. Како није било ограничења у томе где могу да се завршавају тунели, MBone мрежа је имала тачке широм света. Тада није био развијен концепт међу-доменског мултикаста нити хијерархијске организације рутирања, јер се сав мултикаст развијао унутар једне јединствене MBone мреже. Тек накнадно, када је MBone мрежа почела да се значајно шири се дошло до тога да треба да се направи хијерархијска организација рутирања налик оној која постоји за уникаст: интерни протоколи (у које спадају PIM протоколи) и међу-доменски протоколи. [5.17] Пошто се до тада искристалисао PIM-SM као најбољи протокол који пружа могућност креирања оптималних стабала уз минимално оптерећење рутера, пошло се од претпоставке да ће се у сваком домену користити овај протокол, те да ће домени имати своје RP рутере.

У претходном поглављу смо видели ситуацију када постоји више RP у једном домену и показано је да је један од кључних проблема које треба решити обавештавање једног RP рутера о постојању извора мултикаста у домену другог RP. Прва идеја је била да се направи један централни RP за цео интернет, који би омогућио размену ових информација, али се

δрзо одустало јер је такав концепт у потпуној супротсности са потпуно децентрализованом архитектуром интернета. Друга идеја је δила да постоји тачка за размену мултикаст информација – MIX. NASA је резервисала посеδан δрој аутономног система за ову потреδу (AS10888) са идејом да у њој постоји PIM DM мрежа и да се у њој налазе RP свих аутономних система. Међутим и ова идеја је напуштена зδог централизације управљања.

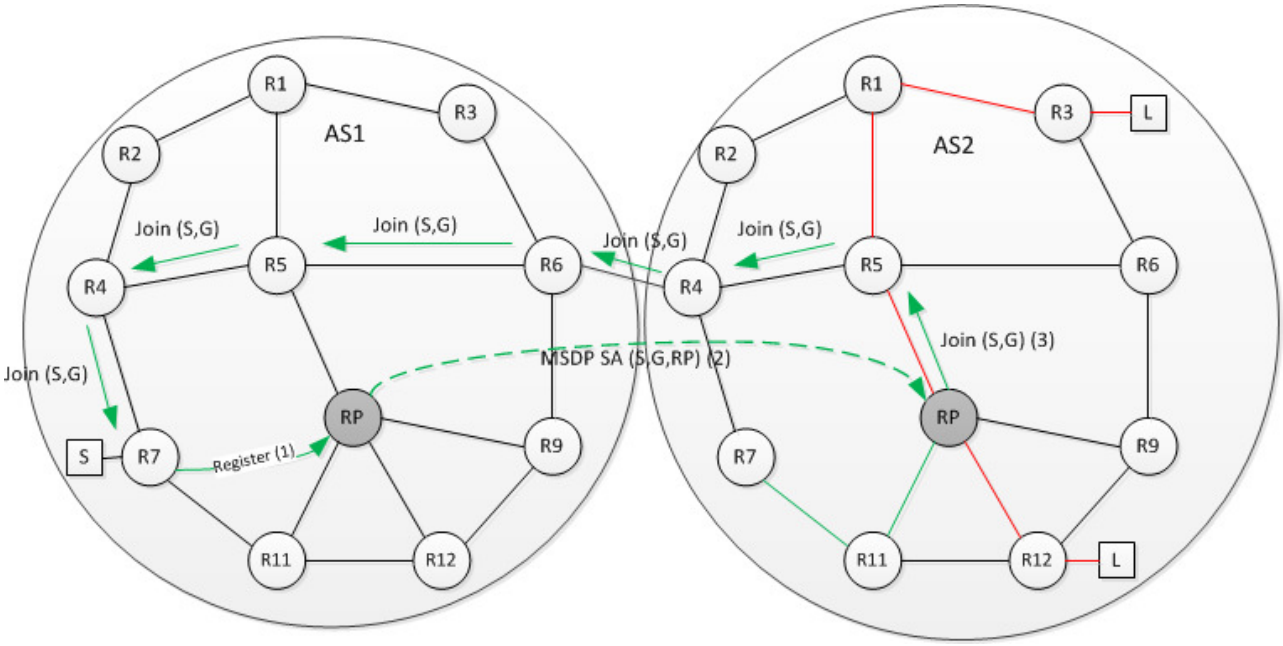
На крају, решење које је прихваћено је да се међу-доменски мултикаст реализује користећи концепт сличан ономе који је описан у претходном поглављу са MSDP протоколом. Наиме, пример са Слике 5.19 показује мрежу која је подељена постојањем два RP и начин на који може да се оδједини тако да мултикаст пакети долазе до свих слушалаца у оδа дела коришћењем MSDP протокола. Оно што је различито у односу на тај случај је то што код међу-доменског мултикаста сви RP имају различите адресе из опсега који припадају аутономним системима у којима се налазе, а такође, постоје и проδлеми када уникаст и мултикаст топологија нису конзистентне када је потреδно укључити и мултипротоколарни BGP.

5.2.6.1. MSDP протокол за међу-доменски мултикаст

MSDP протокол се успоставља између рутера у суседним аутономним системима. То су најчешће RP рутери, али није неопходно да то δуде тако. Чак и они аутономни системи који не подржавају мултикаст могу да имају рутере који комуницирају MSDP протоколом ако желе да омогуће трансфер информација о изворима мултикаста. MSDP сесије се успостављају преко TCP протокола између уређаја који не морају да δуду директно повезани.

Када се у неком домену појави извор мултикаст пакета, пакети се по правилима описаним у поглављу 5.2.5.4 шаљу до најδлижег RP рутера у његовом домену. RP шаље свим својим MSDP суседима (са којима има успостављене сесије) *Source Active* (SA) поруке у којима се налазе следеће информације: адреса извора мултикаст саоδраћаја S, адреса мултикаст групе на коју се шаље мултикаст G и адреса RP који је генерисао поруку. MSDP уређаји у суседним аутономним системима по пријему SA поруке врше њено прослеђивање свим осталим MSDP суседима осим оног од којег је доδијена SA порука – врши се плављење (енг. *flooding*) SA порукама како δи дошле до свих аутономних система на сличан начин као што се то ради мултикаст пакетима код PIM-DM. Такође, механизам прихватања SA порука је сличан као код пријема мултикаст пакета – врши се RPF провера. MSDP уређај ће примити SA поруку само ако је дошла путањом која води према адреси RP који је огласио дату SA поруку, а која је енкапсулирана у њој.

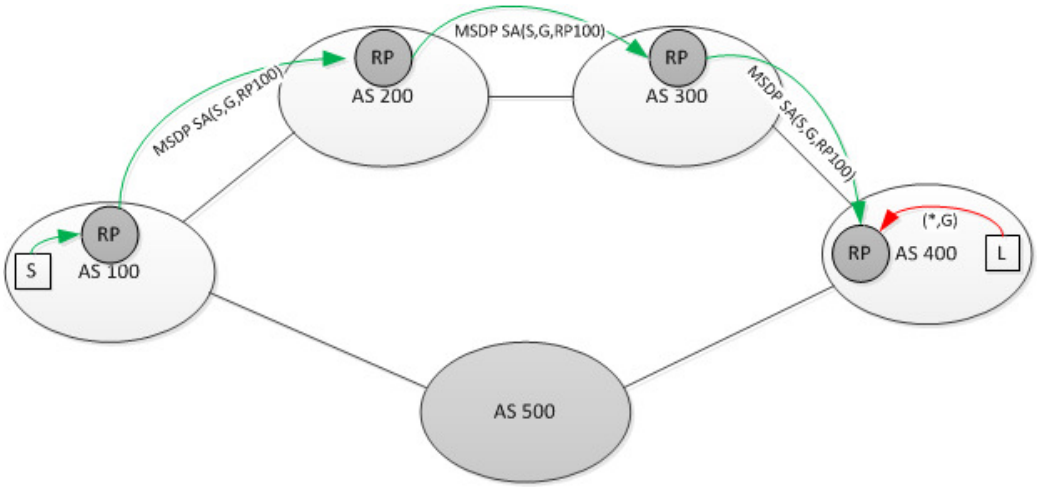
RP који прими MSDP поруку којом је оглашено емитовање мултикаст пакета за групу G уколико има регистрованих слушалац те групе иницира креирање најкраћег стаδла до извора S. Овај процес је приказан на слици 5.19, након чега може да почне проток мултикаст пакета између домена.



Слика 5.19 Успостављање мултикаст стабала између домена

5.2.6.2. Проблем неконзистентности мултикаст и уникаст рутирања

Да би неки аутономни систем био повезан на интернет, довољно је да је регистрован као LIR и да поседује број аутономног система и IPv4 адресе. Подршка за мултикаст, као и подршка за IPv6 није обавезна, те је данас ситуација таква да не подржавају сви аутономни системи мултикаст рутирање. Када би сви аутономни системи подржавали мултикаст, онда би био довољан MSDP механизам описан у претходном поглављу. Међутим, пошто то није случај, могуће је да дође до сценарија какав је приказан на слици 5.20.



Слика 5.20 Неконзистентне уникаст и мултикаст топологије

У примеру са слике аутономни системи 100, 200, 300 и 400 подржавају мултикаст и унутар аутономног система и међу-доменски и имају исправно инсталиране и конфигурисане RP рутере и MSDP протокол. Аутономни систем 500 не подржава мултикаст. Такође, ако

претпоставимо да ни један аутономни систем није вршио никакве манипулације BGP атрибутима, онда најкраћа путања између аутономних система 100 где је извор и 400 где је слушалац мултикаста води преко аутономног система 500. Ово је ситуација у којој уникаст и мултикаст топологија нису једнаке. Док је за мултикаст једина путања преко аутономних система 200 и 300, уникаст пакети могу да иду преко обе алтернативне путање, а краћа је друга путања, она преко аутономног система 500.

Оваква ситуација ствара проблем приликом успостављања мултикаст тока пакета. Наиме, SA порука када дође до RP рутера у аутономном систему 400 неће проћи RPF проверу, зато што најкраћа повратна путања ка RP у 100 који је огласио дату SA поруку води преко аутономног система 500, а сама порука је стигла преко аутономног система 400. Такође и када би пакети некако стигли до рутера у аутономном систему 400, он би морао да их одбаци јер не пролазе RPF проверу. Оваква ситуација у којој долази до стварања тзв. црне рупе у мултикаст рутирању може да се превазиђе коришћењем мултипротоколарних екстензија за BGP протокол којим се омогућава попуњавање посебне табеле на рутерима која ће служити само за RPF проверу.

Мултипротоколарне екстензије за BGP не служе за формирање мултикаст стабала, већ у овом случају оглашавају уникаст руте из аутономних система у којима је у функцији мултикаст. Ове руте могу да буду оглашене са идентификаторима мреже (SAFI – *Subnetwork Address Family Identifier*) 1 (уникаст), 2 (мултикаст) или 3 (оба), где 1 значи да се рута оглашава као класична BGP уникаст рута, 2 да се оглашава за потребе међу-доменског мултикаста и 3 да дата рута се оглашава и као уникаст и као мултикаст рута. Рута која је оглашена са SAFI 1 ће бити смештена у стандардну BGP табелу. Рута која је оглашена са SAFI 2 ће бити смештена у посебну табелу у којој су уникаст руте, а која служи за потребе RPF провере за мултикаст (*multicast* RIB). Рута која је оглашена са SAFI 3 ће бити истовремено смештена у обе табеле. Ако постоји *multicast* RIB табела, рутери ће RPF проверу вршити на основу података из ње. У примеру са слике, пошто би се руте са SAFI ознаком 2 оглашавале само преко аутономних система који подржавају мултикаст (грана 100-200-300-400), то значи да не би било проблема са RPF провером, јер би у тој табели била информација да је путања према RP који је оглсио дату SA поруку преко аутономних система који подржавају мултикаст.

5.2.7. Обавештавање слушалаца о мултикаст садржају

Да би клијентске апликације и сами корисници добили информацију о томе која мултикаст група носи одређени садржај, постоје два додатна протокола:

- *Session Announcement Protocol* (SAP) [5.18] којим се оглашава садржај који се емутије преко одређене мултикаст групе (нпр. на адреси G је пренос фудбалске утакмице). SAP оглашава садржај периодичним слањем инфорамција мултикастом на адресу 224.2.127.254.

- *Session Description Protocol* (SDP) технички описује послати садржај (начин кодовања гласа, слике итд.) како би клијентска апликација могла да употреби одговарајуће кодеке за пријем сигнала. Овај протокол се користи и у IP телефонији са SIP протоколом за размену истих информација између IP телефонских агената.

Клијентске апликације попут VLC[46] које подржавају слање и пријем мултикаст саобраћаја такође подржавају и ова два протокола, те је могуће добити листу свих распложивих мултикаст група и њихових техничких карактеристика.

5.2.8. Раширеност мултикаст преноса данас

Упркос неспорној супериорности у економичности потрошње мрежних ресурса за дистрибуцију садржаја преко интернета у поређењу са методама заснованим на уникасту укључујући и CDN, мултикаст није доживео очекивано прихватање од стране великих провајдера и данас не може да се говори о великој заступљености мултикаста на интернету – глобални мултикаст интернет и даље највише одржава истраживачка и универзитетска заједница. Подршка за међу-доменски мултикаст је у провајдерским мрежама више изузетак него правило, па нема озбиљних сервиса заснованих на мултикасту који су успостављени глобално иако потреба за тако нечим јасно постоји. Који су разлози за то није до краја јасно. Неки аутори сматрају да је то због додатне комплексности која се уноси у конфигурацију рачунарске мреже и сложеног начина рада [5.19], док други сматрају још увек не постоји добар модел којим би провајдери могли да наплаћују транспорт мултикаст саобраћаја [5.20]. Како год, данас је апсолутно доминантан начин за дистрибуцију садржаја на интернету коришћење CDN.

Ипак, мултикаст има своју примену која је ограничена углавном на појединачне мреже – унутар једног домена. Поред протокола рутирања у мрежним уређајима који стандардно користе мултикаст, користи се за дистрибуцију ТВ програма за IPTV услугу, за дистрибуцију посебних садржаја у IP телефонским мрежама, за *screencasting* помоћу UPnP (*Universal Plug and Play*) протокола и друге примене.

46 https://www.videolan.org/vlc/

5.3. Литература

[5.1] Cisco Visual Networking Index (VNI) Forecast, https://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html приступљено 16.12.2017.

[5.2] https://www.cisco.com/c/en/us/support/docs/ip/network-address-translation-nat/200608-Server-Load-Balancing-Using-Dynamic-NAT.html

[5.3] R. Buyya, M. Pathan, A. Vakali, Content Delivery Networks, Lecture Notes in Electrical Engineering, Springer-Verlag Berlin Heidelberg, 2008, ISBN 978-3-540-77886-8.

[5.4] https://www.akamai.com/uk/en/about/our-thinking/state-of-the-internet-report/

[5.5] S. Deering, Multicast Routing in Internetworks and Extended LANs, Stanford report, No. STAN-CS-88-1214, July 1988, http://i.stanford.edu/pub/cstr/reports/cs/tr/88/1214/CS-TR-88-1214.pdf

[5.6] T. Speakman, J. Crowcroft, J. Gemmell, D. Farinacci, S. Lin, D. Leshchiner, M. Luby, T. Montgomery, L. Rizzo, A. Tweedly, R. Edmonstone, R. Sumanasekera, L. Vicisano, PGM Reliable Transport Protocol Specification, IETF RFC 3208, December 2001., https://tools.ietf.org/html/rfc3208

[5.7] J. C. Lin, S. Paul, "RMTP: A Reliable Multicast Transport Protocol", ACM SIGCOMM August 1996.

[5.8] Floyd, S., Jacobson, V., Liu, C., McCanne, S., and Zhang, L. “A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing.” ACM *Transactions on Networking* , November 1996

[5.9] Chung Kei Wong, M. Gouda and S. S. Lam, "Secure group communications using key graphs," in *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 16-30, Feb 2000. doi: 10.1109/90.836475

[5.10] M. Baugher, R. Canetti, L. Dondeti, F. Lindholm, Multicast Security (MSEC) Group Key Management Architecture, IETF RFC 4046, April 2005, https://tools.ietf.org/html/rfc4046

[5.11] B. Weis, S. Rowles, T. Hardjono, The Group Domain of Interpretation, IETF RFC 6407, October 2011, https://tools.ietf.org/html/rfc6407

[5.12] B. Williamson, Developing IP Multicast Networks, Cisco Press, October 1999, ISBN-10: 1-58714-289-9

[5.13] B. Cain, S. Deering, I. Kouvelas, B. Fenner, A. Thyagarajan, Internet Group Management Protocol, Version 3, IETF RFC 3376, October 2002., https://tools.ietf.org/html/rfc3376

[5.14] M. Christensen, K. Kimball, F. Solensky, Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches, IETF RFC 4541, May 2006, https://tools.ietf.org/html/rfc4541

[5.15] N. Bhaskar, A. Gall, J. Lingard, S. Venaas, Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM), IETF RFC 5059, January 2008, https://tools.ietf.org/html/rfc5059

[5.16] K-root, RIPE NCC, https://www.ripe.net/analyse/dns/k-root/ (приступљено 17.12.2017.)

[5.17] K.C. Almeroth, The Evolution of Multicast: From the Mbone to Interdomain Multicast to Internet2 Deployment, IEEE Network, January/February 2000.

[5.18] M. Handley, C. Perkins, E. Whelan, Session Announcement Protocol, IETF RFC 2974, October 2000., https://tools.ietf.org/html/rfc2974

[5.19] S. Ratnasamy, A. Ermolinskiy, S. Shenker, Revisiting IP multicast, In Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '06). ACM, New York, NY, USA, 15-26. DOI=http://dx.doi.org/10.1145/1159913.1159917

[5.20] C. Diot, B.N. Levine, B. Lyles, H. Kassem, D. Balensiefen, Deployment issues for the IP multicast service and architecture, IEEE Network, Volume: 14, Issue: 1, Jan/Feb 2000.

6. Квалитет сервиса

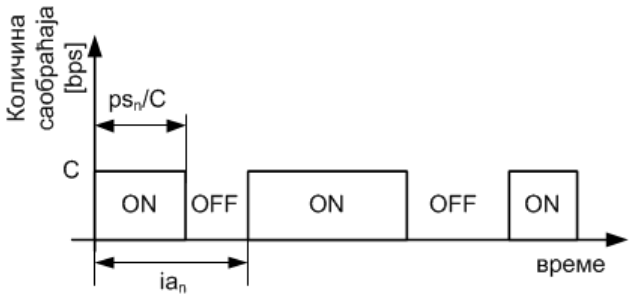
За разлику од времена настанка, када су се користиле за комуникацију између великих рачунарских центара и за пренос фајлова и текстуалних порука, рачунарске мреже данас преносе пакете који носе врло различит садржај: веб странице, текст, слике, аудио и видео садржај како снимљен, тако и онај који се користи директну за комуникацију људи, податке пословних софтверских апликација, контролне поруке рачунарских мрежа (протоколи рутирања, DNS) итд. Јасно је да различите примене мрежа и садржај на њима имају различите захтеве у погледу ефикасности преношења пакета: док за корисника који преузима неку веб страницу није претерано значајно да ли ће та страница доћи за 500ms или за 1s, повећање кашњења пакета од 500ms приликом директног разговора два лица створиће привремену деградацију квалитета разговора у виду прекида тока говора или замрзавања слике или ако траје дуже учиниће да квалитет разговора буде потпуно незадовољавајући. Како се пакети различитих апликација равноправно преносе везама рачунарских мрежа, појавио се проблем како обезбедити да они пакети за које постоје специфични захтеви у погледу ефикасности преношења кроз мрежу добију одговарајући третман. Под термином квалитет сервиса (енг. *Quality of Service* – QoS) подразумева се низ механизама, алгоритама и протокола којима се обезбеђују адекватне перформансе преношења пакета појединих одабраних апликација.

6.1. Перформансе преноса пакета кроз рачунарске мреже

6.1.1. Капацитет веза и проток података кроз мрежу

Корисник рачунарске мреже најчешће купује капацитет неке везе која му је потребна (веза ка интернету, везе између локација фирме). Капацитет везе (енг. *Link Capacity* - C_l) је

максимални број бита које је могуће послати на дати линк у јединици времена. При овоме се мисли на број бита који је могуће остварити на физичком слоју. Капацитет везе је ограничен физичким карактеристикама преносног медијума и одређен хардверским карактеристикама интерфејса.



Слика 6.1 Заузеће везе у рачунарској мрежи

На некој вези у δило ком тренутку или постоји или не постоји пакет. У том смислу, када се говори о тренутном заузећу везе, оно је једнако или капацитету везе (када има пакета на линку – ON период) или нула (када нема пакета на линку – OFF период) – Слика 6.1. Пакети се шаљу на везу брзином која је једнака њеном капацитету. На слици су дате и неке од основних мера којима се најчешће описују карактеристике саобраћаја: величина *n*-тог пакета - ps_n, време између доласка *n*-тог и *n+1*-ог пакета (енг. *interarrival time* ia_n). Време серијализације (енг. *serialization delay)* је време слања пакета величине ps_n на линк (време од тренутка изласка првог до тренутка изласка последњег δита тог пакета на везу) и једнако је ps_n/C, где је *C* капацитет линка.

Чињеница да је тренутно заузеће везе једнако или капацитету везе или нули је тривијална и не даје довољно информација о условима на које ће наићи пакети у мрежи. Зато се уводи појам количине саοбраћаја која постоји на неком линку као δрој послатих δајтова саοбраћаја усредњен у одређеном временском интервалу *τ*. Количина саобраћаја на линку *i* усредњена у *τ* је:

$$\overline{CT}_i(\tau) = \int_{t-\tau}^{t} CT_i(t)dt$$

Уοδичајени временски интервал у којем се посматра количина саοбраћаја у алатима за праћење рада рачунарских мрежа је 5 минута, а пример праћења овог параметра је показан у поглављу 4.3.5.

Потреδно је такође разликовати расположиве капацитете везе на различитим слојевима. Наиме, сваки пакет се састоји од дела који је користан за корисника (и његову апликацију) – део са подацима и дела пакета у којем су контролне информације - заглавља различитих слојева који заузимају део капацитета везе. Ово ће δити показано на примеру. Данас су најуοδичајеније етернет везе и пакети који имају IP и TCP заглавља. Такви пакети имају заглавља следеће величине:

- Етернет заглавље које укључује: *Start Frame Delimiter*, преамбулу, MAC адресе, тип и верификацију пакета укупне величине 38 бајтова.
- IPv4 заглавље величине 20 бајтова.
- TCP заглавље величине 20 бајтова.

Дакле, минимална дужина различитих заглавља (када нема додатних опција у заглављима) је 78 бајтова. То значи да код пакета максималне величине (MTU 1500 бајтова[47]) се око 5% изгуби на контролне податке (заглавља) који нису корисни за корисника мреже, али и значи да ће апликација корисника моћи да добије најмање 5% мање капацитета од максимално декларисаног. Како је просечна величина пакета у рачунарским мрежама и мања, овај проценат је већи и креће се просечно од 10-15% [6.1].

6.1.2. Мере квалитета преноса пакета

Капацитет везе и количина саобраћаја у неком временском интервалу нису једине мере из којих може да се закључи какав је квалитет преноса пакета нити га довољно детаљно описују. За поједине апликације и то пре свега оне које обезбеђују интерактивну аудио-визуелну комуникацију о којима ће бити дискутовано даље у тексту много значајније су следеће основне мере:

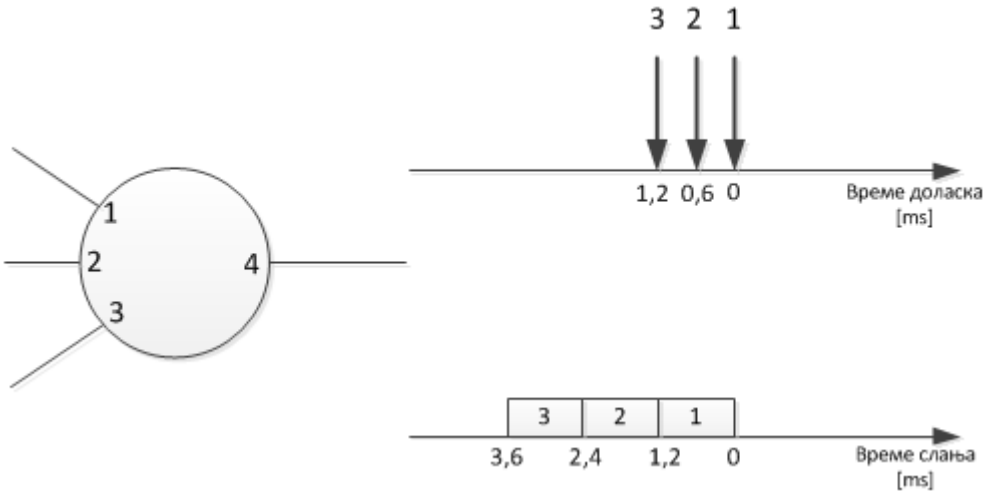
- број/проценат изгубљених пакета у преносу
- кашњење пакета – време од тренутка када је пакет послат са извориште до тренутка када је пакет комплетно стигао до дестинације
- варијација времена кашњења пакета – џитер (енг. *jitter*)

Ове мере се често налазе у тзв. *Service Level Agreement* - SLA уговорима о нивоу услуге где се пружалац услуга обавезује да ће те мере бити унутар одређених лимита, а уколико нису корисник може да наплати пенале или добије умањење цене услуге. Постоје још неке мере попут доступности или поузданости које су изведене из ове три основне, а које описују стандарди који регулишу питања SLA[6.2][6.3].

6.1.2.1. Губитак пакета

Мрежни уређаји имају већи број интерфејса преко којих могу истовремено да долазе и одлазе пакети. На слици 6.2 је приказан пример једног хипотетичког уређаја који има четири интерфејса. На интерфејсе 1, 2 и 3 долазе пакети максималне величине 1500 бајтова у тренуцима 0, 0,6ms и 1,2ms и сва три пакета су намењена излазном интерфејсу 4. Подразумевани начин рада рутера је да се пакети прослеђују по FIFO (енг. *First In First Out*) принципу. Како је излазни интерфејс 4 је капацитета 10Mbit/s, време серијализације ових пакета на линку 4 је 1,2ms.

47 На новијим уређајима, од увођења гигабитског етернета је могуће да се подеси MTU и на вредност од 9000 бајтова. Ово су тзв. *jumbo* оквири



Слика 6.2 Серијализација пакета који наилазе на заγушење

Ако се претпостави да је време процесирања пакета у рутеру занемарљиво, први пакет ће почети да излази на интерфејс 4 одмах по пристизању, дакле у тренутку 0. Како у тренутку када дође пакет 2 први пакет још увек излази на интерфејс 4, други пакет мора да сачека да се заврши слање првог пакета да би почело његово слање. Том приликом се пакет смешта у бафере који су организовани или као део централне меморије за сваки од интерфејса или као хардверски бафери на интерфејсима. Слично се дешава са пакетом 3 који мора да сачека комплетно слање пакета 2 да би почео излази на интерфејс 4.

Оваква ситуација у којој су на интерфејсе рутера дошла три пакета величине 1500 бајтова у року од 1,2ms је ситуација у којој долази до привременог загушења интерфејса 4 јер је интензитет долазног саобраћаја већи од капацитета излазног интерфејса[48]. Дакле, једна од последица недовољног капацитета излазног линка је привремено смештање пакета у бафере. Како су бафери увек коначне величине, у ситуацијама када је загушење дуготрајно, може да дође до тога да се бафери интерфејса потпуно попуне. Сваки пакет који је намењен загушеном излазном интерфејсу у ситуацији када су бафери попуњени бива одбачен, односно долази до губитка пакета.

Ако се изузму ситуације у којима је дошло до квара интерфејса или преносног медијума, главни разлог губитака пакета у рачунарским мрежама су ситуације у којима је дошло до дуготрајнијих загушења. Губици пакета код TCP сесија изазивају смањење величине прозора, односно количине послатих података без потврде што значи смањење протока TCP сесија у којима је дошло до губитка пакета. Код UDP токова који су карактеристични за пренос аудио или видео садржаја у реалном времену, губитак пакета се не надокнађује механизмом ретрансмисије, већ изазива деградацију квалитета послатог садржаја и због тога је изразито негативна појава.

48 Лако је проверити да три пакета величине 1500 бајтова који долазе за 1,2ms на различите интерфејсе рутера, а намењени су једном излазном интерфејсу представљају привремени проток од 30Mbit/s, што је веће од капацитета интерфејса 4.

6.1.2.2. Кашњење

Кашњење пакета је време које протекне од тренутка када је почето слање пакета са извора до тренутка када је последњи бит пакета стигао до дестинације. Кашњење се састоји из:

- фиксног кашњења – које је непроменљиво на путањи пакета од извора до дестинације и укључује: време пропагације сигнала по преносном медијуму, време процесирања пакета у мрежним уређајима на путањи и време серијализације пакета на свим везама од извора до дестинације.
- променљивог кашњења које је последица загушења на везама. Као што је показано у претходном поглављу загушења изазивају смештање пакета у бафере, што производи повећано кашњење пакета

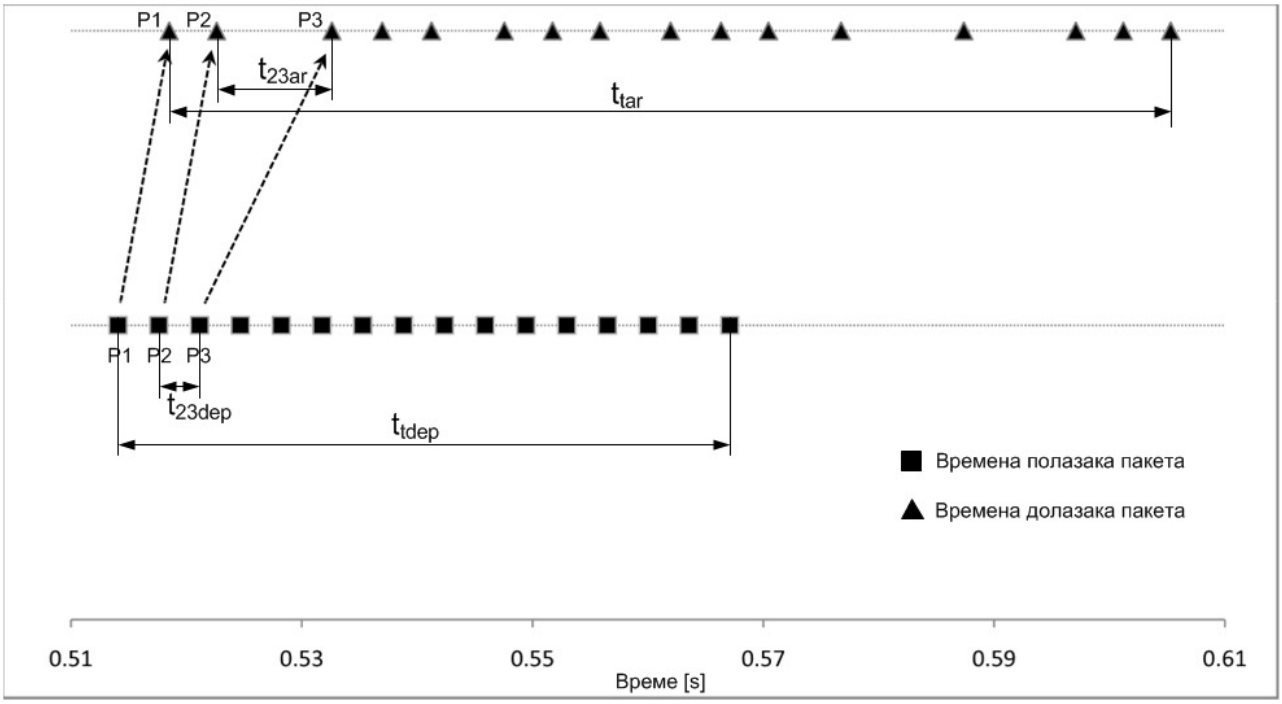
Повећано кашњење пакета није од претераног значаја у применама када се путем рачунарске мреже преузимају фајлови или веб стране, али јако негативно утиче на апликације интерактивне аудио-визуелне комуникације зато што превелико кашњење значајно отежава разговор и квари субјективни осећај квалитета. Због тога ITU препорука G.114 ограничава максимално кашњење на 150ms односно 250-300ms у једном смеру за земаљске и сателитске везе респективно за сигнал којим се шаље говор.

6.1.2.3. Варијација кашњења – џитер

Путање пакета од изворишта до дестинације се састоје од већег броја веза и на свакој од њих може да дође до нагомилавања пакета у баферима услед привремених или трајних загушења. Такође, на сваком рутеру постоји више интерфејса преко којих долазе токови пакета различитих статистичких карактеристика чије путање делом могу да се поклапају са путањама пакета који су од интереса. Овакво комплексно окружење чини то да чак и у ситуацијама када не постоји загушење није лако предвидљиво какво ће бити кашњење сваког појединачног пакета. Кашњења ће варирати између неких вредности. На слици 6.3 је показана ситуација која је снимљена у реалном мрежном окружењу у којој се у мрежу шаље периодични низ пакета са једнаким временским размаком између њих (означени квадратима) и времена долазака тог истог низа пакета након проласка кроз неколико веза од којих је на једној и загушење (означени троугловима). Очигледно је да низ пакета на долазној страни није више периодичан и да су кашњења пакета различита.

Као и претходне мере квалитета преноса, варијација кашњења пакета је посебно неповољна карактеристика за пренос аудио-визуелних података. Наиме, дигитализација гласа или покретне слике захтева периодично одабирање аналогног сигнала и затим његово дигитално кодовање. Да би се на долазној страни реконструисала реплика оригиналног сигнала, потребно је да се примљене дигиталне информације претворе у аналогну форму тако што ће се одбирци оригиналног сигнала довести на конвертор са истом периодом са којом су креирани. Варијација кашњења пакета негативно утиче на реконструкцију оригиналног сигнала јер нарушава периодичност одбирака. Да би се ублажио ефекат варијације кашњења, на долазној страни се користе тзв. дежитеризациони бафери у које се смештају пакети по

доласку да би се одбирци оригиналног сигнала вадили из бафера периодично са истим ритмом у којем су генерисани. Како убацивање новог бафера на долазној страни повећава укупно кашњење пакета, битно је да задржавање пакета у дејитеризационим баферима буде релативно мало, а то је могуће када је варијација кашњења мала.



Слика 6.3 Варијација времена пропагације пакета приликом проласка кроз мрежу

6.2. Архитектуре обезбеђења квалитета сервиса

У претходном поглављу је показано да на све три основне мере квалитета сервиса пре свега негативно утичу загушења која се јављају у мрежи. Овај негативан ефекат може да се избегне на два различита начина:

1. повећањем капацитета веза како би се смањила вероватноћа загушења без примене неких посебних мера у оквиру мрежних уређаја, или
2. применом различитих алгоритама којима се загушења у мрежи контролишу тако да пакети за које је потребно да се мере квалитета сервиса држе унутар одређених граница добију посебан третман у мрежи. Ове методе подразумевају дискриминацију и различит третман пакета који припадају различитим апликацијама или корисницима.

Први приступ код којег се оставља подразумевани начин рада мрежних уређаја (обично FIFO прослеђивање) и не врши никаква дискриминација пакета се назива *best effort*. Ритам којим су се повећавали капацитети етернет интерфејса је био врло брз у последњих 20 година (од 10-мегабитног до 100-гигабитног етернета) и одговарао је временским циклусима редовне промене мрежне опреме. Такође приступачност мултиплексирања већег броја паралелних веза по једном физичком медијуму учинили су да стратегија са сталним повећањем

капацитета веза, без дискриминације пакета буде једна од често примењиваних. Ова стратегија је популарна и код оних који заговарају принципе неутралности мреже односно интернета (енг. *NetNeutrality*) који подразумевају да се сав саобраћај на интернету третира равноправно, без дискриминације и који се противе пракси провајдера да фаворизују одређену врсту пакета и да додатно наплаћују побољшан квалитет за приступ неким веб сајтовима [6.4].

Са друге стране постоје две архитектуре побољшања квалитета сервиса код којих се врши дискриминација и различит третман пакета: интегрисани и диференцирани сервиси, које су описане у наставку овог поглавља.

6.3. Архитектура интегрисаних сервиса

Први модел пружања посебног третмана неким пакетима у рачунарским мрежама је архитектура интегрисаних сервиса (енг. *Integrated Services* – IntServ)[6.5]. Овим моделом је предвиђено да свака појединачна апликација на крајњем уређају може да тражи и да добије од мреже резервацију ресурса или гаранцију третмана пакета за своје потребе. Протокол који се користи за исказивање жеље за резервацијом и саму резервацију ресурса је RSVP (енг. *Resource Reservation Protocol*). Апликација која има потребу да се пакетима које шаље и прима дâ неки другачији третман од остатка саобраћаја шаље према дестинацији пакета RSVP PATH поруке у којима је опис захтева. Рутери на путањи примају PATH поруке, памте стање захтева и на сваком сегменту путање проверавају да ли постоје услови да се захтевани третман пружи, те прослеђују поруке даље према дестинацији. Ако на неком сегменту постоје услови да се захтевани третман омогући, од рутера који је примио PATH поруку према ономе који је послао ће се слати RSVP RESV поруке са потврдом резервације, а у супротном RSVP ERROR поруке. Како би се обезбедило да у случају асиметричних путања у мрежи PATH и RESV поруке иду истом путањом, у PATH порукама рутери приликом слања ових порука у изворишну IP адресу уписују своју адресу, што се памти на наредном рутеру на путањи. Уколико на свим сегментима путање постоји могућност да се испуни тражени захтев и пошаљу се RESV поруке дуж целе путање, на рутерима се резервише и конфигурише тражени третман.

Архитектура интегрисаних сервиса је предвидела две врсте сервиса: гарантовани проток (енг. *Guaranteed Rate*) којим се резервише део капацитета дуж путање за потребе апликације и контролисани проток (енг. *Controlled Load*) који треба да обезбеди минимално кашњење. Начини на који рутери обезбеђују овакав третман пакетима и конкретни алгоритми су описани у каснијим деловима овог поглавља у оквиру описа архитектуре диференцираних сервиса.

Архитектура интегрисаних сервиса је једини модел који омогућава да појединачна апликација добије посебан третман сопственог саобраћаја од мреже. Међутим, ова

могућност је уједно и главна мана архитектуре зато што је грануларност модела на нивоу појединачне апликације једног корисника која може да има своје резервације у мрежи сувише ситна. Већ у мрежи са више стотина рачунара где на сваком више апликација може да тражи захтеве од мреже, број потенцијалних резервација и стања које би морали да памте рутери би био огроман. Такође, постоје проблеми у вези са тиме како одредити који захтев има виши приоритет и шта радити у ситуацијама када број и врста захтева превазилазе капацитете на неком мрежном сегменту. Све ово чини да је архитектура интегрисаних сервиса нескалабилна и релативно је брзо напуштена. По усвајању архитектуре Microsoft је направио подршку за RSVP протокол и АПИ у Windows 2000 оперативном систему [6.6], али је ова подршка повучена из каснијих верзија. Са друге стране RSVP протокол је доживео своју другу примену у оквиру MPLS механизама за оптимизацију искоришћења ресурса мреже – MPLS TE што је и показано претходно у тексту.

6.4. Архитектура диференцираних сервиса

Као што је претходно показано, кључна мана архитектуре интегрисаних сервиса је превише ситна грануларност модела доделе ресурса и резервација у мрежи што је учинило да је цео модел нескалабилан. Стога је свега неколико година након овог модела донета нова архитектура - диференцираних сервиса (енг. *Differentiated Services - DiffServ*) [6.7]. Основна разлика у односу на претходни модел је та што се у архитектури диференцираних сервиса сав саобраћај класификује у коначан број класа без обзира на то од колико појединачних извора долази, а тим класама се затим додељује одређени третман. Са коначним бројем класа се решава проблем скалабилности и комплексности конфигурација и броја стања на рутерима, али се губи могућност да појединачне апликације добију фиксне гаранције за третман пакета. Ипак, ово је модел пружања квалитета сервиса који је широко прихваћен, данас актуелан и користи се у рачунарским мрежама.

Механизми диференцираних сервиса се састоје из неколико различитих активности:

- класификација и означавање (енг. *classification and marking*) – пакети се на уласку у мрежу или део мреже у којем се примењују механизми квалитета сервиса класификују у коначан број класа и означавају, како би на основу ознака у пакетима касније могле да се примењују наредне активности.
- ограничавање и поравнавање (енг. *policing and shaping*) којима се на некој вези ограничава количина пренетих података у јединици времена за неки одабрани мрежни ток.
- механизми контроле загушења (енг. *congestion management*) којима се у ситуацији када постоји нагомилавање пакета у баферима врше различите процедуре опслуживања бафера како би неки пакети имали посебан третман.

- механизми избегавања загушења (енг. *congestion avoidance*) којима се спречава тзв. глобална синхронизација TCP сесија.

Сви ови механизми ће бити детаљно описани у наставку текста овог поглавља.

6.4.1. Класификација и означавање

Класификација данас може у мрежним уређајима да се врши практично на основу било ког податка који се налази у пакету или је у вези са пакетом. Класификација може да се врши:

- на основу долазних интерфејса,
- на основу MAC адресе, VLAN ознака или неког другог податка на слоју везе,
- на основу података у IP заглављу попут адреса, поља ToS или DSCP,
- на основу TCP или UDP портова чиме се имплицитно одређује и апликација која прима или шаље дате пакете (нпр. добро је познато да се телнет саобраћај шаље по TCP порту 23, DNS по порту 53, веб по порту 80 за HTTP или 443 за HTTPS итд.),
- на основу неког податка у делу пакета где су смештени подаци (дубока инспекција садржаја пакета – енг. *deep packet inspection* - DPI). Ово је потребно у оним ситуацијама када неке апликације користе добро познате портове за слање својих пакета (нпр. торент клијенти могу да се подесе да комуницирају по порту 80 што чини претходно описану класификацију на основу броја порта неефикасном),
- на основу произвољне комбинације претходних критеријума.

Класификација пакета доноси додатно процесирање и кашњење пакета, а у ситуацијама дубоке инспекције садржаја пакета и значајну потрошњу процесорског времена. Стога најефикаснија стратегија није да се класификација обавља на сваком рутеру посебно, већ да се класификација обави на уласку пакета у мрежу или део мреже у којем се примењује архитектура диференцираних сервиса и да се на том месту пакети означе тако да рутери унутар мреже могу да коришћењем ознака одреде на који начин треба да поступају са појединим пакетима.

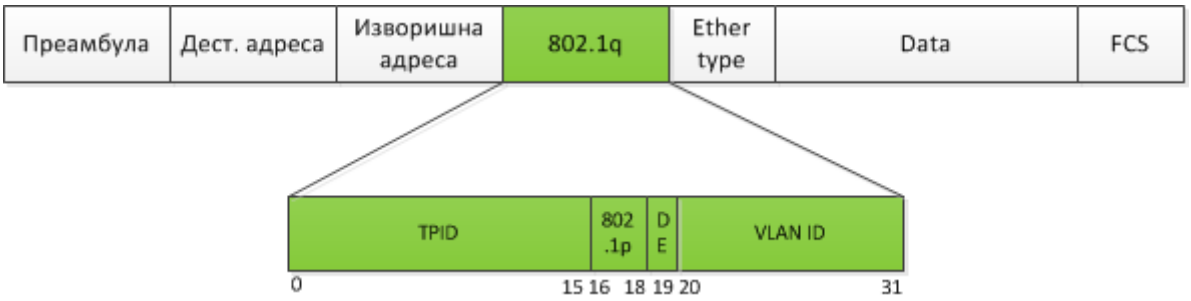
Не постоји ни пропис ни препорука у колико класа ће пакети бити разврстани. Ово је одређено како потребама конкретне мреже тако и техником обележавања која се користи у мрежи (следеће поглавље). Један пример за класификацију у рачунарској мрежи неке компаније према опадајућем значају би могао да буде:

- 1. класа – контролне инфомације, DNS и протоколи рутирања
- 2. класа – подаци које размењује кључни пословни софтвер компаније
- 3. класа – подаци за интерактивну комуникацију (пренос гласа и видеа, видеоконференције, итд.)
- 4. класа – веб и мејл саобраћај

- 5. класа – сав остали саобраћај

6.4.1.1. Означавање на слоју везе

Данас се на највећем броју веза користи етернет технологија на слоју везе. У етернет заглављу не постоји поље у које би могла да се упише информација о одређеној класи након извршене класификације. Међутим, 802.1q заглавље у које се уписује информација о VLANовима има одвојена 3 бита (бити 16-18 – 802.1p, слика 6.4) за уписивање класе сервиса које тај пакет треба да има.



Слика 6.4 Поља заглавља слоја везе

С обзиром на то да постоје 3 бита за означавање, могућа је класификација на максимално 8 различитих класа. Важно је уочити да се заглавља слоја везе мењају на свакој вези између два рутера на некој путањи која се састоји од више повезаних сегмената, тако да обележавање на слоју везе има важење само у оквиру једног мрежног сегмента. Уколико се жели обележавање с краја на крај преноса пакета, потребно је да се обележавање врши у неко поље заглавља пакета које се не мења током проласка пакета кроз мрежу (нпр. IP заглавље) или да се на сваком рутеру врши поновно обележавање пакета што није ефикасно. У рутерима постоје механизми пресликавања обележавања са слоја везе у обележавање на мрежном слоју чиме се остварује продужавање домета обележавања.

6.4.1.2. Означавање на мрежном слоју

У оригиналном IP протоколу било је дефинисано поље ToS – *Type of Service* (слика 6.5) у коме су била дефинисана поља за одређивање приоритета пакета (енг. *Precedence* – 3 бита) и посебни бити D, T и R за које је било предвиђено да се користе приликом одлучивања како ће бити рутиран пакет: по путањи минималног кашњења (енг. *Delay*), макималног протока (енг. *Throughput*) или максималне поузданости (енг. *Reliability*).

Поље *Precedence* је давало могућност да се пакети разврстају у максимално 8 класа. Са друге стране, никада у рутерима није имплементирана могућност да чињеница да је сетован неки од бита D, T или R утиче на то како ће пакет бити рутиран. Рутирање се вршило и врши се на основу дестинационе адресе и табеле рутирања. Стога је у пољу *Type of Service* практично све осим прва три бита било неискоришћено. Ово је утицало на то да приликом дефинисања архитектуре диференцираних сервиса цело поље *Type of Service* потпуно редефинише [6.8].

Слика 6.5 Type of Service поље

Редефиницијом овог поља, првих 6 бита су постали DSCP (енг. *Differentiated Services Code Point*) поље у које је могуће уписати ознаку једне од максимално 64 класа пакета. У време доношења документа којим је дефинисано DSCP поље, преостала два бита су била у том тренутку недефинисана (CU – *currently unused*), али су касније добила своју намену која ће бити описана у поглављу 6.4.4.3.

Слика 6.6 DSCP поље

Иако је корисник слободан да класама пакета додели произвољне DSCP вредности унутар своје мреже, у оквиру RFC документа којим је дефинисано ово поље су дефинисане неке препоручене вредности[49]:

- 0 – *Best Effort* (BE) која је намењена за сав саобраћај који није посебно класификован и за који није битно да се реализује неки посебан механизам квалитета сервиса
- 46 – *Expedited forwarding* (EF) класа која је намењена за оне пакете који захтевају минимално кашњење кроз мрежу. Уобичајено је да су то пакети којима се преноси интерактивна комуникација и ти пакети се обично смештају у приоритетни ред за чекање (Поглавље 6.4.3.2).
- 4 *Assured Forwarding* (AF) класе које су описане на слици 6.7 и чије су вредности дате служе за класификацију пакета без јасног критеријума какве ће услове добити нека од ових класа (на кориснику је да одреди услове за пакете према својим потребама). Унутар сваке од класа постоје три различите вероватноће одбацивања пакета. Објашњење тога шта значе ове вероватноће одбацивања пакета и како је имплементиран тај механизам ће бити дато у поглављу 6.4.4.2.

Као што је речено, не постоји обавеза да се било која мрежа придржава ове класификације, али обележавање препорученим вредностима може да буде корисно у ситуацијама када две различите мреже желе да омогуће да механизми квалитета сервиса у једној мрежи важе и за пакете класификоване и обележене у другој мрежи.

49 Ове препоручене вредности се у RFC документу називају *Per-Hop Behaviour* (PHB) зато што је њима имплицитно дефинисано какво ће бити процесирање датог пакета на сваком сегменту у мрежи.

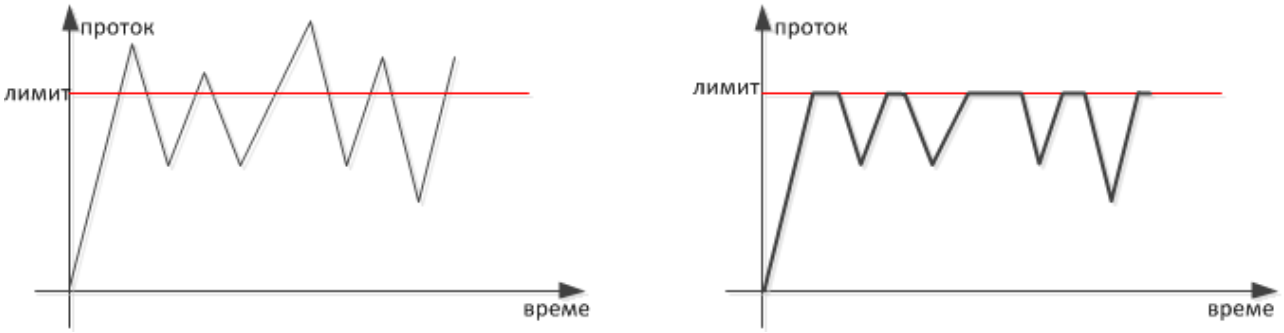
	Класа 1	Класа 2	Класа 3	Класа 4
Ниска Вероватноћа одбацивања	AF11 DSCP10	AF21 DSCP18	AF31 DSCP26	AF41 DSCP34
Средња Вероватноћа одбацивања	AF12 DSCP12	AF22 DSCP20	AF32 DSCP28	AF42 DSCP36
Висока Вероватноћа одбацивања	AF13 DSCP14	AF23 DSCP22	AF33 DSCP30	AF43 DSCP38

Слика 6.7 Assured Forwarding класе

6.4.2. Ограничавање и поравнавање (уобличавање)

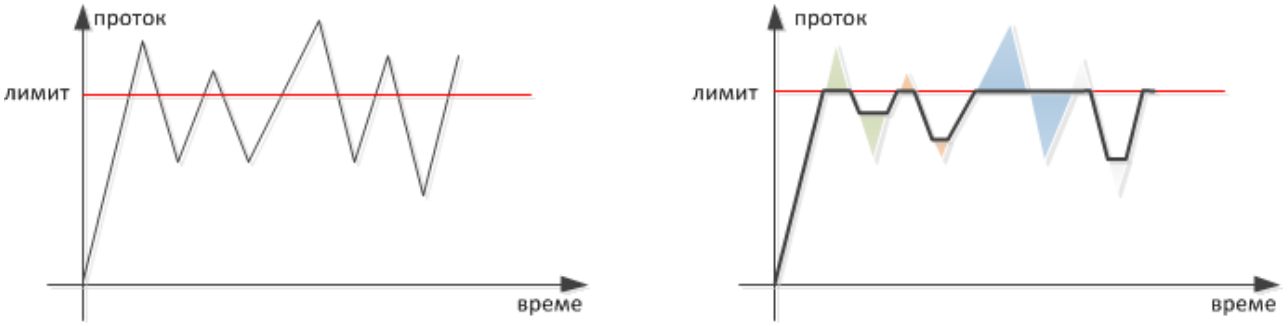
Механизми ограничавања (енг. *policing*) и поравнавања/уобличавања (енг. *shaping*) служе за то да се ограничи количина саобраћаја која долази преко неког интерфејса на одређену договорену просечну вредност како се не би изазивала загушења у мрежи или како би се кориснику дао онај капацитет који је платио. Разлика између ова два механизма је у томе како третирају пакете који излазе изван договореног просечног протока.

Док механизам ограничавања одбацује све пакете који су изван договореног профила (Слика 6.8), механизам поравнавања може да оне пакете који су изван профила смести привремено у бафере, до тренутка када долазни проток падне испод договореног нивоа и да га онда пусти кроз мрежу (Слика 6.9). Очигледно је да се механизмом поравнавања за исти улазни профил саобраћаја добија график протока који је мање варијабилан око вредности договореног лимита него у случају ограничавања. Одатле и име механизма.



Слика 6.8 Ограничавање протока

Механизми ограничавања и поравнавања се типично примењују на улазима у мрежу и служе за ограничавање количине саобраћаја који може да у мрежу пошаље неки корисник како би се држало под контролом оптерећење мреже.



Слика 6.9 Поравнавање/уобличавање протока

6.4.2.1. Ограничавање

За реализацију механизма ограничавања се користи алгоритам који се у литератури често назива *Token bucket* (кофа са жетонима). Ово име алгоритма треба да на сликовит начин прикаже како се одређује који се пакети понашају у складу са договореним лимитом и који могу да буду пропуштени, а који се одбацују[50]. Ипак у овом поглављу ће објашњење бити формалније и то на примеру три варијанте овог алгоритма.

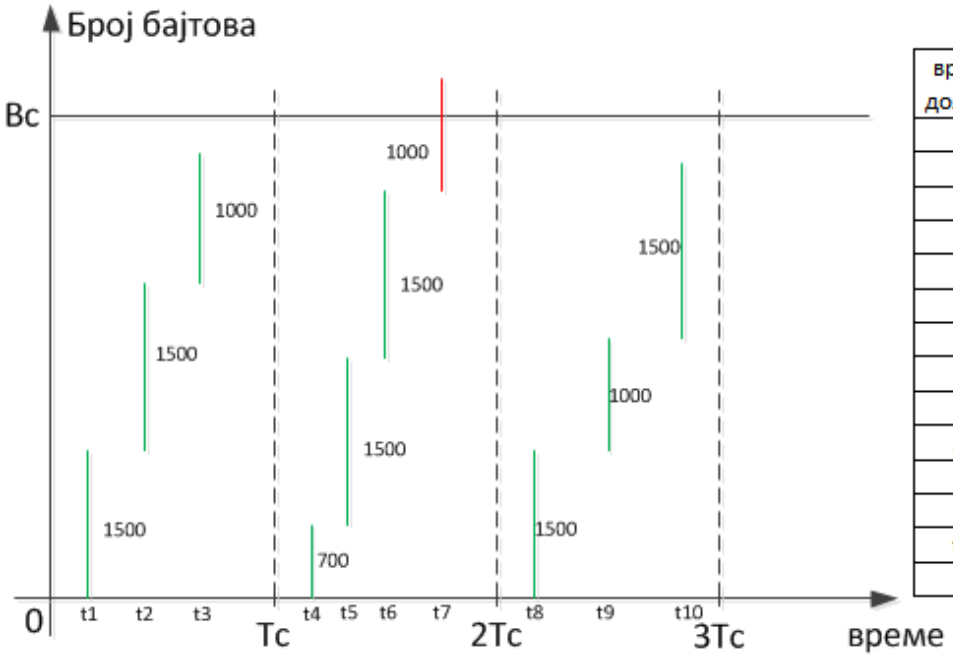
Ограничавање са једним лимитом и две класе пакета

Основне променљиве којима се дефинише понашање алгоритма су:

- Договорени проток – проток на који се провајдер обавезао да ће да сигурно пропустити (енг. *Commited Information Rate* – CIR) и исказује се као број бита/бајтова у јединици времена. Ово је вредност лимита са Слике 6.8.
- Временски интервал усредњавања/мерења – *Tc*. Већ је у поглављу 6.1 дискутовано да исказивање тренутног протока на некој вези нема много смисла и да проток мора да се мери усредњен у неком временском интервалу. Овај интервал је конфигурабилан и један је од параметара подешавања ограничавача на мрежним уређајима.
- Број бита/бајтова које је могуће пропустити у временском интервалу мерења – *Bc* (енг. *commited burst*). Однос између ових параметара је: *CIR=Bc/Tc*.

На пример, ако је договорени лимит *CIR=64Kbit/s*, и ако је интервал мерења протока *Tc=0,125ms* онда је вредност променљиве *Bc=8000* бита, односно у сваком интервалу је могуће пропустити највише 8000 бита. За линкове малих капацитета или мале капацитете ограничавања је битно одредити временски интервал мерења тако да не буде превише кратак као у овом примеру. Наиме, у овом примеру у једном интервалу је могуће пропустити максимално 8000 бита. Кроз овако конфигурисан ограничавач никада не би могли да прођу пакети максималне величине од 1500 бајтова (12000 бита), што би у потпуности обесмислило коришћење овако конфигурисане мреже.

50 Претпоставља се да се на почетку сваког интервала мерења протока додељује одређен број жетона који представљају број бита или бајтова који могу да се пренесу. Сваки пакет који прође узима из кофе онолико жетона колика је његова величина. Ако у кофи има довољно жетона, пакет може да прође, ако нема, пакет неће проћи.



време доласка	величина пакета	Bc (max 4500)
0		4500
t1	1500	3000
t2	1500	1500
t3	1000	500
Tc		4500
t4	700	3800
t5	1500	2300
t6	1500	800
t7	1000	800
2Tc		4500
t8	1500	3000
t9	1000	2000
t10	1500	500
3T		4500

Слика 6.10 Ограничавање са једним лимитом и две класе пакета

Алгоритам ограничавања са једним лимитом и две класе пакета функционише на следећи начин:

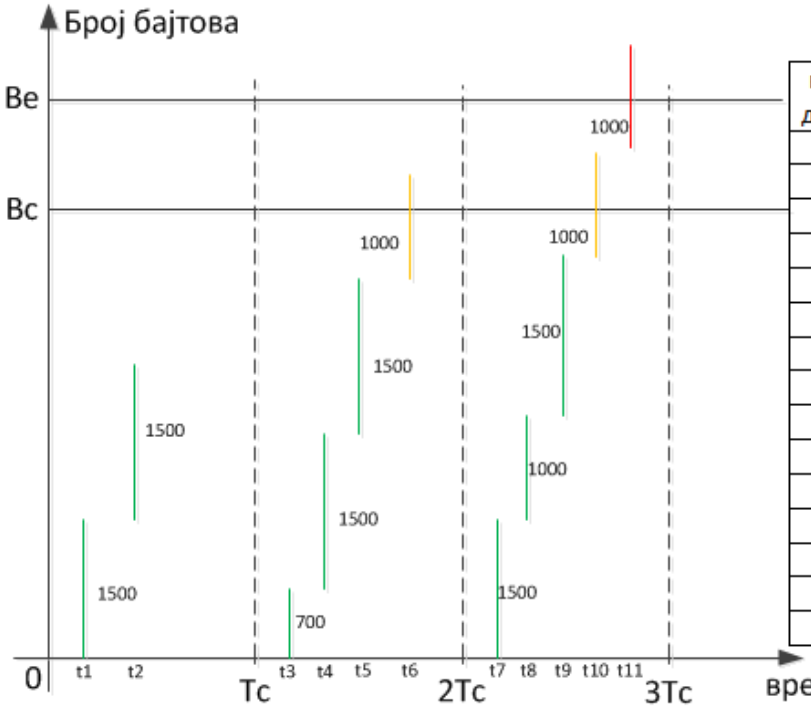
- На почетку сваког интервала мерења се променљива *Bc* поставља на максималну вредност (8000 у претходном примеру).
- Приликом доласка сваког пакета проверава се да ли је величина пакета у битима већа од броја у променљивој *Bc*:
  - Ако је величина пакета мања од тренутне вредности *Bc*, променљива *Bc* се декрементира за величину пакета и пакет се пропушта (прва класа пакета),
  - Ако је величина пакета већа од вредности *Bc*, променљива *Bc* не мења вредност и пакет се одбацује (друга класа пакета).

Ово је приказано на примеру са Слике 6.10 где су показана три узастопна интервала мерења. У првом интервалу мерења су дошла у тренуцима *t1*, *t2* и *t3* на улаз три пакета величине 1500, 1500 и 1000 бита респективно. Како је њихова укупна величина од 4000 бита мања од договорене вредности *Bc=4500*, сва три пакета могу да прођу кроз рутер јер је проток у датом интервалу мањи од договореног лимита.

У следећем интервалу се вредност *Bc* на почетку поставља поново на максималну вредност. Како у овом интервалу након три пакета чија је укупна величина 3700 бита (што оставља вредност 800 у променљивој *Bc*), четврти пакет величине 1000 бита који долази у тренутку *t7* неће бити пропуштен зато што је његова величина мања од тренутне вредности *Bc*.

Ограничавање са два лимита и три класе пакета – варијанта 1

Честа је ситуација да се не дефинише један лимит за проток као у претходном примеру, већ да провајдер поред овог лимита дефинише још један који је већи од претходног и за који се пакети пропуштају условно.



време доласка	величина пакета	Bc (max 4500)	Be (max 1000)
0		4500	0
t1	1500	3000	0
t2	1500	1500	0
Tc		4500	1000
t3	700	3800	1000
t4	1500	2300	1000
t5	1500	800	1000
t6	1000	0	800
2Tc		4500	800
t7	1500	3000	800
t8	1000	2000	800
t9	1500	500	800
t10	1000	0	300
t11	1000	0	300
3T		4500	300

Слика 6.11 Ограничавање са два лимита и три класе пакета – варијанта 1

Код овог алгоритма постоје две додатне променљиве које су потребне за његово описивање:

- Проширени проток – енг. *Extended Information Rate – EIR* је проток за који провајдер жели да пропусти пакете под условом да ови додатни пакети не производе загушење у мрежи и исказује се као број бита у јединици времена
- Број додатних бита које је могуће пропустити у временском интервалу мерења – Be (енг. *extended burst*). Однос између ових параметара је: *EIR=Be/Tc*

Ова варијанта алгоритма ограничавања са два лимитом и три класе пакета функционише на следећи начин:

- На почетку сваког интервала мерења се променљива *Bc* поставља на максималну вредност. На почетку првог интервала се променљива *Be* поставља на вредност 0.
- Приликом доласка сваког пакета се проверава да ли је величина пакета у битима већа од броја у променљивој *Bc*:
  - Ако је величина пакета мања од тренутне вредности *Bc*, променљива *Bc* се декрементира за величину пакета и пакет се пропушта (прва класа пакета – пропуштени без промене),

- Ако је величина пакета већа од тренутне вредности *Bc*, постоје две варијанте:
  - Ако је величина пакета мања од збира тренутних вредности *Bc* и *Be* променљива *Bc* се смањује на 0, а променљива *Be* се смањује за вредност (величина пакета – вредност за колико је смањена *Bc*). Пакет се пропушта, али може да добије другачију ознаку. (друга класа пакета – условно пропуштени)
  - Ако је величина пакета већа од збира вредности *Bc* и *Be* променљиве *Bc* и *Be* се не мењају, а пакет се одбацује. (трећа класа пакета - одбачени)

- Уколико на крају временског интервала мерења остане нека вредност у променљивој *Be*, пренеће се у следећи интервал.
- Уколико на крају временског интервала мерења остане нека вредност у променљивој *Bc*, биће пребачена у *Be*, до максималне вредности *Be*.

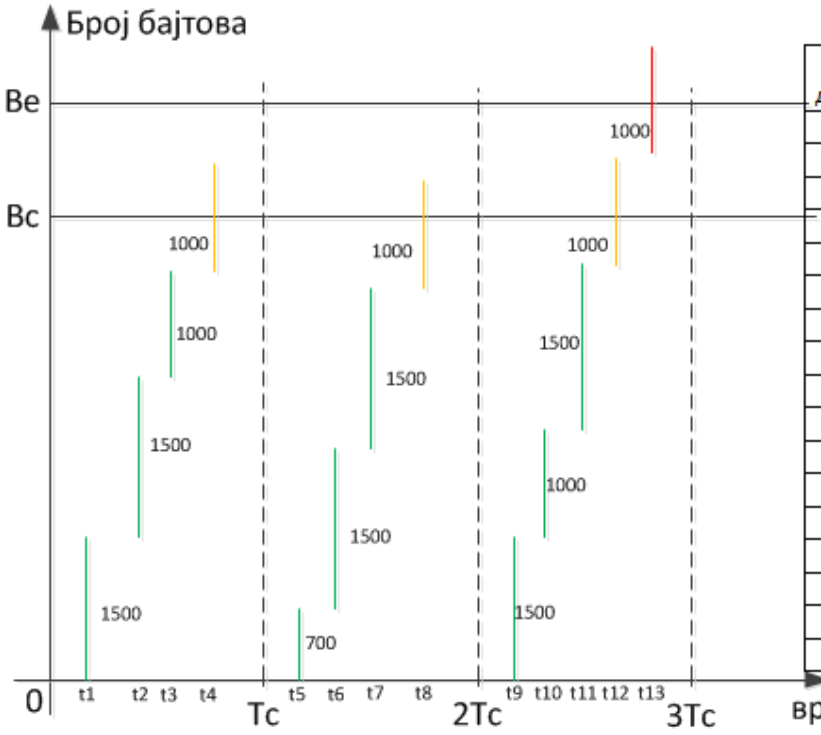
Ово је показано примером са Слике 6.11. У првом интервалу су била само два пакета укупне величине 3000. Пошто је почетна вредност *Bc=4500*, оба пакета су прослеђена без икакве промене. На крају овог интервала променљива *Bc* је имала вредност 1500. Пошто је максимална вредност *Be=1000*, на крају првог интервала у ову променљиву може да буде пребачено максимално 1000 из променљиве *Bc*. На почетку другог интервала, *Bc* ће поново бити постављена на почетну вредност 4500.

У другом временском интервалу прва три пакета укупне величине 3700 се прослеђују без промене и спадају у прву класу пакета јер се за њих декрементира *Bc*. Након проласка трећег пакета у другом интервалу вредност променљиве *Bc* је 800. Наредни пакет који долази у тренутку *t6* је величине 1000 и за њега вредност *Bc* није довољно велика да би био пропуштен. Међутим како је вредност *Be* у том тренутку 1000, променљиве *Bc* и *Be* у збиру су веће од величине пакета и овај пакет може да се пропусти условно – друга класа пакета. Променљива *Bc* ће бити декрементирана на 0 (за 800), а *Be* за разлику величине пакета и вредности за колико је декрементирана *Bc* и имаће вредност за 200 мању од почетне.

У трећем временском интервалу је показан случај пакета који долази у тренутку *t11* за који су вредности *Bc* и *Be* у збиру су мање од величине пакета и тај пакет мора да буде одбачен – трећа класа пакета.

Условно пропуштање пакета значи да се пакет прослеђује кроз дати рутер, али да се обележава неком ознаком која омогућава мрежи да управо ове пакете прве одбаци уколико на некој вези дође до загушења. Ако пакети који долазе на ограничавач припадају на пример *Assured Forwarding* класи 2 и имају ознаку AF21, пакети прве класе ће бити пропуштени без промене ознаке, пакети друге класе ће добити нову ознаку (AF22 или AF23) чиме ће прећи у класу за коју је већа вероватноћа одбацивања него пакета прве класе, а пакети треће класе ће бити одбачени. Како се тачно реализује раније одбацивање пакета друге класе биће објашњено у поглављу 6.4.4.2.

Ограничавање са два лимита и три класе пакета – варијанта 2



време доласка	величина пакета	Bc (max 4500)	Be (max 1000)
0		4500	1000
t1	1500	3000	1000
t2	1500	1500	1000
t3	1000	500	1000
t4	1000	0	500
Tc		4500	1000
t5	700	3800	1000
t6	1500	2300	1000
t7	1500	800	1000
t8	1000	0	800
2Tc		4500	1000
t9	1500	3000	1000
t10	1000	2000	1000
t11	1500	500	1000
t12	1000	0	500
t13	1000	0	500
3T		4500	1000

Слика 6.12 Ограничавање са два лимита и три класе пакета – варијанта 2

Друга варијанта ограничавања са два лимита и три класе пакета је врло слична претходној. Кључна разлика је у томе што се вредност променљиве *Be* поставља на максималну вредност на почетку сваког интервала, без обзира на то да ли је нешто преостало у променљивој *Bc*. Ово значи да у овој варијанти алгоритма у сваком интервалу, а не само у онима у којима је нешто пренесено из *Bc*, може да се пошаље укупно *Bc+Be* бита, с тим да ће они пакети који декрементирају променљиву *Be* бити условно пропуштени, као и у претходној варијанти. Пример рада ове варијанте алгоритма је показан на слици 6.12.

6.4.2.2. Поравнавање

За реализацију механизма поравнавања се користи алгоритам који се у литератури често назива *Leaky bucket* (кофа која цури). Понашање рутера приликом прослеђивања пакета на основу овог алгоритама је аналогно кофи која има рупу на дну и из које вода отиче увек константним протоком. Ако се у кофу сипа вода протоком који је мањи или једнак протоку отицања воде, вода се неће нагомилавати у кофи. Ако се у кофу сипа вода протоком који је већи од протока отицања воде, део воде ће отицати константним протоком отицања, али ће део почети да се накупља у кофи. Ако је кофа пуна и и даље вода дотиче већим протоком од протока отицања, вода ће почети да прелива и да се просипа из кофе. Ако долазни проток падне испод протока отицања, кофа ће се полако празнити.

Реализација овог алгоритма у рутерима је таква да се пакети доводе у бафер (који симулира кофу), а из бафера се ваде константним протоком тако што се у неком временском интервалу

мерења на излазни линк изδацују пакети укупне величине која одговара протоку за тај изаδрани интервал мерења.

6.4.3. Контрола загушења

Када у рачунарској мрежи на некој вези дође до загушења у једном смеру, на рутеру који шаље пакете на ту везу ће доћи до нагомилавања пакета у δаферима. Нагомилавање пакета у δаферима повећава кашњење свих пакета који се смештају у δафере и као што је раније дискутовано негативно се одражава на перформансе неких апликација, пре свега оних које οδезδеђују интерактивну аудио-визуелну комуникацију. Да δи се умањио ефекат загушења на перформансе апликација које користе мрежу, развијене су различите технике којима се опслужују δафери којима редослед којим се пакети из δафера ваде и испоручују на излазни интерфејс није подразумевани FIFO него другачији како δи се остварила један од следећих циљева:

- фер расподела између различитих мрежних токова који конкуришу за прослеђивање. Циљ фер расподеле је да се не дође до ситуације да неки мрежни ток заузме већи део капацитета док други мрежни токови доδијају знатно мањи део капацитета.
- минимално кашњење пакета кроз мрежни уређај
- гарантовани проток/део капацитета за неке мрежне токове

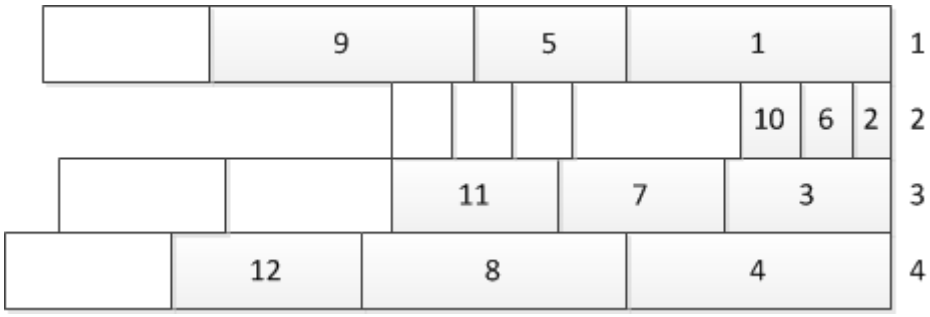
У наставку поглавља су описани неки алгоритми који се срећу у мрежним уређајима којима се тежи остваривању ових циљева.

6.4.3.1. Кружно опслуживање – Round Robin – RR

Ако се жели фер расподела капацитета између већег δроја мрежних токова у случају загушења, један од најједноставнијих алгоритама δи δило кружно опслуживање (енг. *Round Robin* - RR) у којем δи сваки мрежни ток доδио свој тренутак да проследи пакете по цикличној шеми. Ово је приказано на примеру са слике 6.13. У примеру је претпостављено да је δафер подељен у 4 логичка реда за чекање (а δрој логичких редова за чекање може да δуде већи и конфигураδилан). Сви пакети који долазе на дати излазни интерфејс су разврстани у четири класе које се смештају у посеδне делове δафера (пакети се типично разврставају на основу DSCP ознаке). Пакети се прослеђују на излазни интерфејс по цикличној шеми: прво пакет из првог логичког реда за чекање, па пакет из другог, и тако даље редом до последњег логичког реда, након чега се редослед враћа на први логички ред. На слици је δројевима означен редослед изласка пакета из δафера када се користи кружно опслуживање.

Иако је кружно опслуживање на први поглед фер јер сваки ред за чекање доδија своју једнаку прилику да пошаље пакет, зδог тога што на излазни интерфејс могу да се пошаљу само цели пакети, ова шема није фер јер фаворизује оне мрежне токове и редове за чекање који имају веће пакете. Као што може да се види на слици, други логички ред за чекање ће

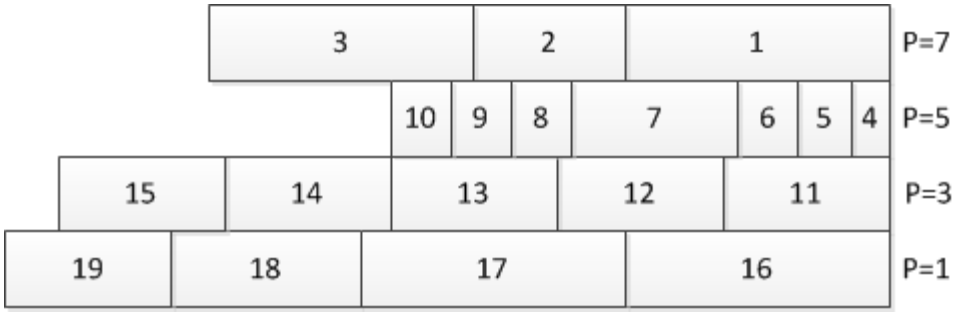
после три циклуса слања пакета послати значајно мање бита на интерфејс зато што су пакети у том реду за чекање мали. Такође, пошто остварени проток логичког реда за чекање зависи од величине пакета, овај алгоритам не може да пружи гарантовани проток логичким реовима за чекање. Због овога су развијене друге шеме (FQ, WFQ) које дају фер расподелу пакета јер узимају у обзир и величину пакета. Такође, постоје и варијанте кружног опслуживања са тежинским факторима (енг. *Weighted RR*) код којих је предвиђено да онај логички ред за чекање који има *n* пута већи тежински фактор од другог логичког реда за чекање добије *n* пута више циклуса за слање пакета.



Слика 6.13 Кружно опслуживање редова за чекање

6.4.3.2. Приоритетно опслуживање – Priority Queueing - PQ

Ако се жели да се омогући минимално кашњење неких пакета (на пример пакета који преносе податке интерактивне комуникације), онда је најједноставнији начин који то пружа приоритетно опслуживање. Код овог алгоритма бафер се такође дели на више логичких редова за чекање који имају додељене различите приоритете (слика 6.14). Пакети могу да се смештају у логичке редове за чекање на пример на основу вредности IP *Precedence* поља. Пакети се код приоритетног опслуживања прослеђују тако да ће прво на излазни интерфејс бити прослеђивани пакети из логичког реда за чекање највишег приоритета (у примеру *P=7*). Тек када сви пакети из логичког реда за чекање највишег приоритета буду прослеђени, прећи ће се на логички ред за чекање следећег мањег приоритета и тако даље.



Слика 6.14 Приоритетно опслуживање редова за чекање

Овај алгоритам прослеђивања пакета гарантује минимално кашњење пакетима у логичком реду за чекање највишег приоритета, јер како дође неки пакет у овај ред за чекање, он ће бити упућен на прослеђивање пре свих осталих пакета нижег приоритета који су у баферу. Проблем оваквог опслуживања реда за чекање је то што у ситуацији када је проток пакета

високог приоритета велики може да дође до ситуације да пакети нижег приоритета не добију своју прилику да избаце пакете (енг. *starvation*).

6.4.3.3. Фер опслуживање – Fair Queueing - FQ

У рачунарским мрежама уз FIFO дисциплину опслуживања редова за чекање која се најчешће среће на линковима великих брзина, често се користи фер опслуживање редова за чекање (енг. *fair queuing - FQ)* [6.9] или нека врста опслуживања изведена из ове врсте (*weighted fair queueing - WFQ, class-based WFQ – CBWFQ*) и то типично на споријим или загушеним линковима. Овај начин опслуживања редова за чекање је у теоријској литератури познат и као генерализовано дељење процесора (енг. *Generalized Processor Sharing – GPS)* [6.10]. Циљ *GPS* начина опслуживања редова за чекање је да у ситуацијама када постоји загушење, сви мрежни токови који треба да прођу датим линком добију приближно једнак део пропусног опсега. Мрежни ток чине пакети са истим изворишним и одредишним IP адресама и портовима транспортног слоја – енг. *flow*. Систем са фер опслуживањем реда за чекање може да буде математички формално описан на следећи начин:

Долазни низ пакета је подељен на m класа и свакој класи се придружује посебан ред за чекање. Систем може да опслужи просечно c захтева у јединици времена (капацитет система, односно у пракси капацитет излазног линка). Свакој класи i долазног процеса се придружује ненегативан тежински фактор w_i којим се одређује део капацитета система који ће припасти свакој класи. Стога може да се напише: $\sum_{i=1}^{m} w_i=1$. Када је $w_i=1/m$ за свако i онда је дисциплина опслуживања реда за чекање фер (FQ) и све класе ће имати исте услове у ситуацијама када дође до загушења. Уколико су тежински фактори различити, онда постоји нека врста тежинског фер опслуживања (WFQ, CBWFQ). Уколико све класе шаљу више саобраћаја од онога што им припада по фер алгоритму расподеле (тежински или не), свака класа ће добити следећи део капацитета система:

$$w_i c / \sum_{i=1}^{m} w_i \qquad (6.1)$$

Уколико постоје класе у којима има мање саобраћаја од ове вредности, вишак саобраћаја датих класа ће се расподелити равномерно на остале класе. У том случају до нагомилавања пакета у редовима за чекање ће доћи само за оне класе које шаљу више од фер дела капацитета који им припада. Ако је Z скуп класа код којих долази до нагомилавања пакета у редовима за чекање, онда ће све класе $i \notin Z$ моћи да пропусте сав свој саобраћај кроз излазни линк, а остале класе ће добити следећи део капацитета:

$$w_i c / \sum_{i \in Z} w_i \geq w_i c \ . \qquad (6.2)$$

У пракси се класе у фер опслуживању (механизми FQ, WFQ) реализују типично као токови у рачунарским мрежама који су одређени n-торкама (MAC адресе, IP адресе, бројеви портова

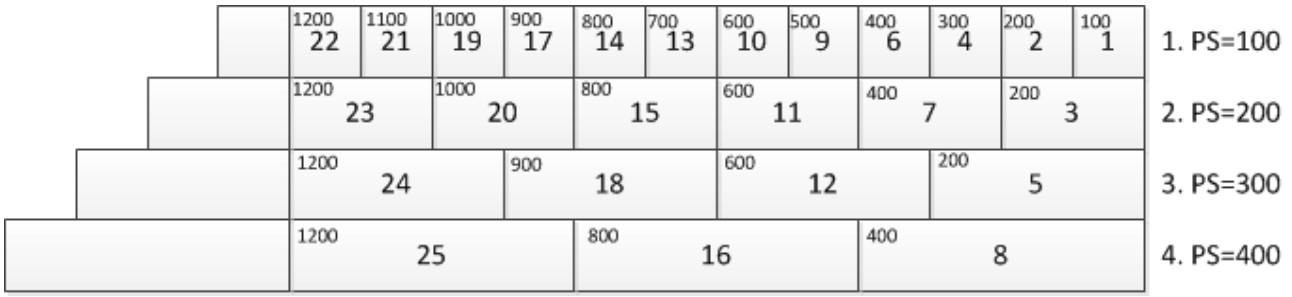
транспортног слоја). Како број токова може да буде релативно велики и већи од броја дефинисаних класа, могуће је да у једној класи буде саобраћај из више токова, а припадност појединог тока некој класи се добија одређивањем неке хеш функције поменуте *n*-торке којом се она мапира у одређену класу. Ако постоји више токова унутар једне класе, дисциплина опслуживања саобраћаја унутар те класе је обично FIFO и не постоји начин за дискриминацију саобраћаја унутар класе.

GPS је идеалан механизам у којем се долазни процес посматра као флуидан, те да самим тим постоји могућност узимања бесконачно малих временских интервала посматрања. Овај механизам се теоретски реализује као кружни алгоритам по коме би се из сваке класе у сваком кругу алгоритма вадио по један бит и прослеђивао на излазни линк. Како је практично немогуће процесирати бит из једног пакета па онда бит из другог, већ се пакети увек процесирају цели, примена кружног алгоритма би због различитих величина пакета нарушила фер однос између класа. Оне класе са већим пакетима би добиле већи део од онога што им припада по фер механизму, што је показано у поглављу 6.4.3.1. Стога је у [6.9] дефинисан алгоритам за FQ начин процесирања пакета којим се симулира GPS механизам у мрежном окружењу и који се данас стандардно користи у мрежним уређајима.

Да би се описао алгоритам потребно је дефинисати следеће променљиве:

- Број рунди - RN – енг. *Round Number* – број циклуса који је до тада добио сваки логички ред за чекање када би се из сваког реда за чекање подаци вадили бит по бит (што је реално немогуће јер пакети морају цели да изађу на излазну везу).
- Секвенцијални број - SN – енг. *Sequence Number* – број који се придружује пакету када дође у логички ред за чекање. Израчунава се на следећи начин:
  - Ако је *n*-ти пакет дошао на празан логички ред за чекање: *SN(n)=RN+size(n)*
  - Ако је *n*-ти пакет дошао на логички ред за чекање у којем већ има пакета: *SN(n)=SN(n-1)+size(n)*

Пакети из реда за чекање излазе по растућој вредности SN.



Слика 6.15 Фер опслуживање редова за чекање

Пример рада овог алгоритма је показан на слици 6.15. Претпостављено је да је бафер подељен на четири логичка реда за чекање. У примеру је подешено да у први логички ред за чекање долазе пакети величине 100 бита, други величине 200, трећи величине 300 и четврти

величине 400 бита, како би се видео утицај величине пакета на фер расподелу. Такође, претпоставља се да је у почетном тренутку вредност *RN=0* (није било опслуживања реда за чекање до тада) и то је тренутак када су дошли први пакети на сва четири логичка реда за чекање. У горњем левом углу сваког пакета су приказане израчунате вредности секвенцијалног броја сваког пакета, а у средини пакета је редослед којим пакети излазе из бафера. Може да се види да је у сваком тренутку број пренетих бита приближно једнак за све логичке редове за чекање, а у тренуцима када је на све логичке редове за чекање дошао једнак број бита у целом броју пакета (тренутак 1200), да је расподела апсолутно фер. Ово је зато што се израчунавањем секвенцијалног броја који узима у обзир величину пакета симулира број *round-robin* циклуса који би добио сваки ред за чекање када би се подаци вадили бит-по-бит.

6.4.3.4. Фер опслуживање са тежинским фактором – WFQ

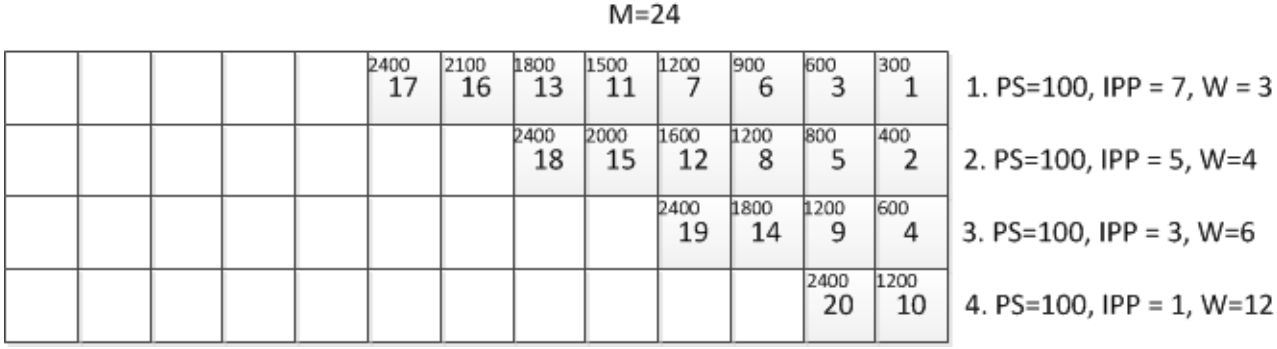
Претходно поглавље је показало како може да се реализује шема у којој сви логички редови за чекање добијају једнак део излазног капацитета. Међутим, као што је показано претходно, у неким применама фер расподела није од интереса за корисника, већ постоји потреба да се поједини мрежни токови фаворизују у односу на остале и то на основу ознака у пакетима. За такву примену је предвиђено фер опслуживање са тежинским фактором. У односу на претходни алгоритам додате су следеће променљиве:

- мултипликатор *M* – број који се користи за израчунавање тежинског фактора и
- тежински фактор *W=M/(1+IPP)*, где је IPP вредност поља IP *Precedence*

FQ алгоритам је модификован тако што се секвенцијални број израчунава на следећи начин:

- Ако је *n*-ти пакет дошао на празан логички ред за чекање: *SN(n)=RN+W*size(n)*
- Ако је *n*-ти пакет дошао на логички ред за чекање у којем већ има пакета: *SN(n)=SN(n-1)+W*size(n)*

Пакети из реда за чекање и код овог алгоритма излазе по растућој вредности SN.



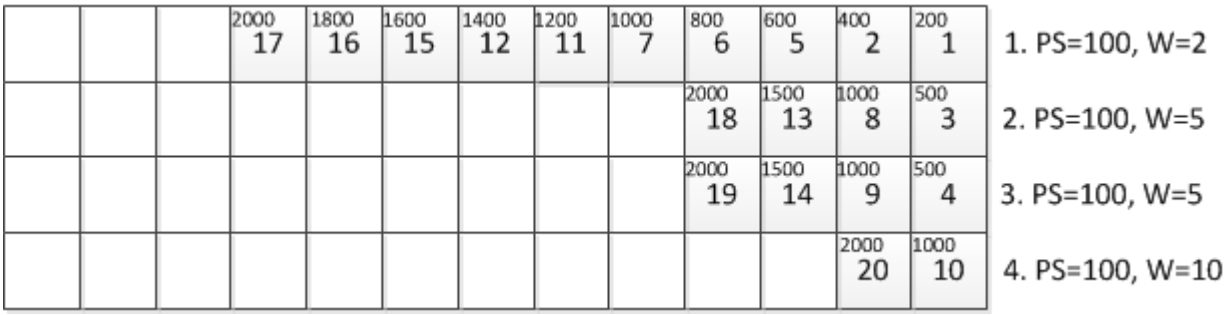
Слика 6.16 Фер опслуживање редова за чекање са тежинским фактором

На слици 6.16 је дат пример функционисања овог алгоритма. Узета је вредност мултипликатора *M=24*, а за претпостављена четири логичка реда за чекање којима је вредност поља IP *Precedence*: 7, 5, 3 и 1 се добијају тежински фактори 3, 4, 6 и 12 респективно. Претпостављено је да на логичке редове за чекање долазе пакети исте величине 100 бита. На основу претходно изнете модификације FQ алгоритма су израчунате секвенцијалне вредности пакета и приказане су у горњем левом углу. Као што може да се види код WFQ алгоритма се добија да је проток по логичком реду обрнуто пропорционалан тежинским факторима: Логички ред са *IPP=7*, чија је тежинска вредност 3 ће добити 4 пута већи проток од логичког реда са *IPP=1*, чија је тежинска вредност 12. Дакле код WFQ алгоритма може да се зна однос протока у логичким редовима за чекање за редове различитих IPP ведности. Међутим, због чињенице да се пакети разврставају у логичке редове за чекање према мрежним токовима чији је број променљив, WFQ не омогућава да се зна колике апсолутне делове капацитета ће добити који логички ред за чекање што ће бити показано на следећем примеру.

Ако се претпостави да на загушени излазни линк долази 8 мрежних токова са 8 различитих IPP вредности, онда ће према једначини (6.2) ток са *IPP=0* добити 1/36 део пропусног опсега, док ће ток са *IPP=7* добити 8/36 делова пропусног опсега. Како имплементације WFQ немају фиксан број логичких редова за чекање, ако би на загушени излазни линк дошло 8 токова са 8 различитих IPP вредности и још 17 токова са *IPP=1*, може лако да се покаже према једначини (6.2), да ће ток са *IPP=0* добити 1/70 део пропусног опсега, док ће ток са *IPP=7* добити 8/70 делова пропусног опсега. Дакле, у релативни однос ће остати исти, док ће апсолутни износи добијеног капацитета бити различити. Ако се жели стриктна контрола добијеног дела капацитета, онда мора да се користи алгоритам код ког је број класа у које се разврставају пакети коначан – *Class Based* WFQ.

6.4.3.5. Фер опслуживање са тежинским фактором засновано на класама – CBWFQ

Овај алгоритам за опслуживање је сличан претходном, уз ту разлику да је број логичких редова фиксно одређен и да се тежински фактори не одређују према вредности поља IP *Precedence* већ на основу жељеног дела капацитета излазне везе.



Слика 6.17 Фер опслуживање са тежинским фактором засновано на клацама

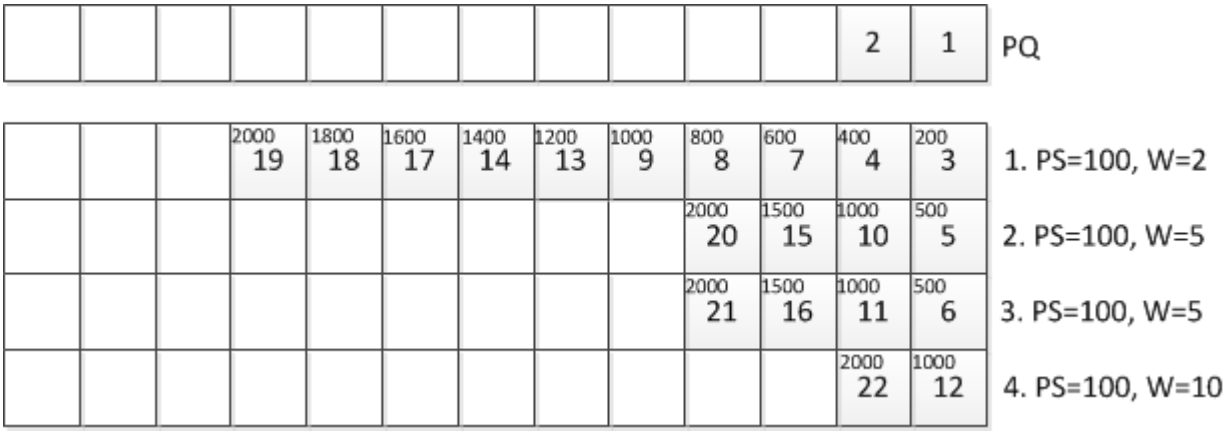
На слици 6.17 је дат пример CBWFQ алгоритма. Претпостављено је да је сав саобраћај који долази на загушени излазни линк подељен у четири класе (нпр. AF11, AF21, AF31 и AF41). Потребе су такве да саобраћај са ознаком AF11 треба да добије 50% капацитета излазног линка (односно у апсолутном износу ако је капацитет излазног линка 1Gbit/s, овај саобраћај ће добити 500Mbit/s), саобраћај са ознакама AF21 и AF31 по 20% капацитета излазног линка и саобраћај са ознаком AF41 10%. Као код WFQ, да би се одржао однос протока различитих класа, потребно је да тежински фактори буду обрнуто пропорционални деловима капацитета који треба да добије свака класа, па су одређени тежински фактори 2, 5, 5 и 10 за AF11, AF21, AF31 и AF41 респективно. У примеру са слике у бафер долазе пакети исте величине како би се видео ефекат овако одабраних тежинских фактора. Као што може да се види после довољног броја пакета који су доспели у бафер, односи добијених протока ће заиста бити онакви како је жељено. Овај алгоритам омогућава апсолутну контролу дела протока који ће добити свака класа у случају када постоји загушење. CBWFQ је прављен да ради са класама диференцираних сервиса и стога може да изврши класификацију у већи број класа од WFQ који све пакете разврстава у максимално 8 класа (на основу вредности поља IP *Precedence*).

У имплементацијама овог протокола у мрежним уређајима често је подразумевано понашање такво да је CBWFQ алгоритму на располагању максимално 75% излазног линка, док се 25% оставља за друге потребе попут контролних протокола и протокола рутирања, мада ово ограничење може да се промени.

6.4.3.6. Опслуживање са малим кашњењем - LLQ

CBWFQ као што је показано пружа апсолутну контролу дела капацитета који ће бити додељен класама пакета у случају загушења. Међутим, CBWFQ не омогућава обезбеђивање малог кашњења за пакете којима је то потребно. У случају преноса говора преко рачунарских мрежа, он се обавља пакетима који су релативно мали (укупно до 100 бајтова) и где је проток једног таквог тока релативно мали (до 30Kbit/s). За такве токове није оправдано да се додељује велики проток који у случају CBWFQ омогућава бржи излазак пакета из бафера. Стога је направљена једна модификација CBWFQ алгоритма која се зове опслуживање са малим кашњењем – енг. *Low Latency Queueing* – LLQ.

LLQ алгоритам је исти као CBWFQ коме је додат још један логички ред за чекање који је вишег приоритета од осталих и ради по принципу приоритетног опслуживања. Дакле, пакети који долазе у приоритетни ред за чекање ће бити одмах опслужени, док ће када се он испразни остали редови за чекање бити опслужени по CBWFQ принципу након пражњења приоритетног реда (Слика 6.18). Да не би дошло до потпуног занемаривања CBWFQ дела реда за чекање, код LLQ постоји могућност ограничавања протока приоритетног реда за чекање механизмом ограничавања.



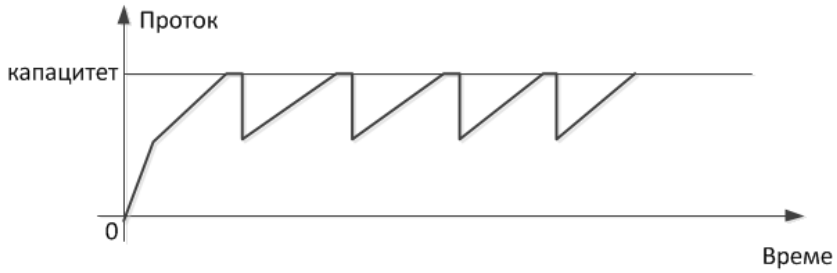
Слика 6.18 Опслуживање са малом кашњењем

Овај алгоритам под називом LLQ је имплементиран на рутерима фирме Cisco Systems. На неким рутерским платформама ове фирме, и на платформама других произвођача сличан алгоритам може да се нађе под називом MDRR (енг. *Modified Deficit Round Robin*) са приоритетним редом за чекање [6.11].

6.4.4. Избегавање загушења

Загушења производе губитке пакета, а код TCP сесија то изазива активирање механизма којима се реагује на загушење: промена величине TCP прозора којом се смањује количина послатих података без потврде пријемне стране и утиче на ублажавање загушења. Како TCP данас чини апсолутну већину саобраћаја на интернету (удео UDP је мањи од 10% удела TCP саобраћаја [6.12]), анализа понашања овог протокола приликом загушења је од великог значаја за оптимално искоришћење мреже.

Иако је TCP стандард донесен још осамдесетих година 20. века и није се мењао од тада, од доношења стандарда до данас је направљено двадесетак различитих имплементација TCP протокола које су се користиле или се данас користе у оперативним системима: [6.13] се користио у оперативним системима са Linux кернелима до верзије 3.2, [6.14] у новијим Linux кернелима, најновији [6.15] у верзијама кернела од 4.9, док Windows оперативни системи од Windows Vista користе [6.16]. Док су заглавље протокола и основни механизми функционисања остали исти, разлика између ових верзија TCP протокола је управо у начину реакције на загушења. Главна мотивација за настанак нових верзија је покушај да се у условима све већих протока на интернет везама омогући брзо слање великих количина информација између уређаја на интернету, а да се том приликом не наруши фер расподела протока између различитих TCP сесија. Старе верзије протокола са третирањем сваког губитка као загушења, спорим стартом и спорим повећањем величине прозора које зависи од времена пропагације пакета кроз мрежу су почеле да представљају ограничење у остваривању већих брзина.



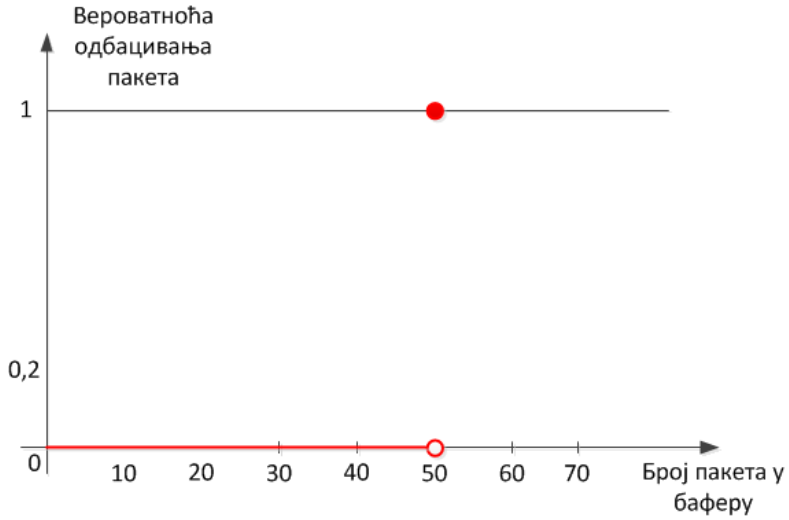
Слика 6.19 Глобална синхронизација

Оно што је ипак приближно слична карактеристика свих TCP верзија је принцип на који се мења величина прозора. Када нема загушења TCP прозор ће се лагано повећавати, чиме ће се повећавати проток TCP сесије, док ће се приликом губитка неког пакета величина прозора брзо смањити (на половину претходне вредности). Овакав начин рада се назива AIMD (енг. *Additive Increase Multiplicative Decrease*).

У ситуацијама када постоји више TCP сесија на загушеном линку, у тренутку када је бафер рутера који шаље на тај линк попуњен, десиће се то да пакети већег броја TCP сесија приближно истовремено дођу на попуњен бафер и да буду одбачени. Ово ће резултовати приближно истовременим, синхронизованим смањењем величине прозора и протока на свим тим сесијама, што ће даље изазвати пад количине саобраћаја на загушеном линку испод капацитета загушеног линка. Дакле у ситуацији када постоји потреба за коришћењем целог капацитета загушеног линка, биће периода када ће искоришћење линка бити мање од капацитета због ефекта синхронизованог смањења прозора у већем броју TCP сесија [6.17]. Све ово ће се понављати након смањења протока на TCP сесијама погођеним губитком пакета, јер ће проток свих TCP сесија поново почети да расте до максималног капацитета и добиће се профил протока који изгледа као тестераста линија (Слика 6.19). Овај ефекат се зове глобална синхронизација и није пожељан јер је очигледно да је загушени линк неоптимално оптерећен.

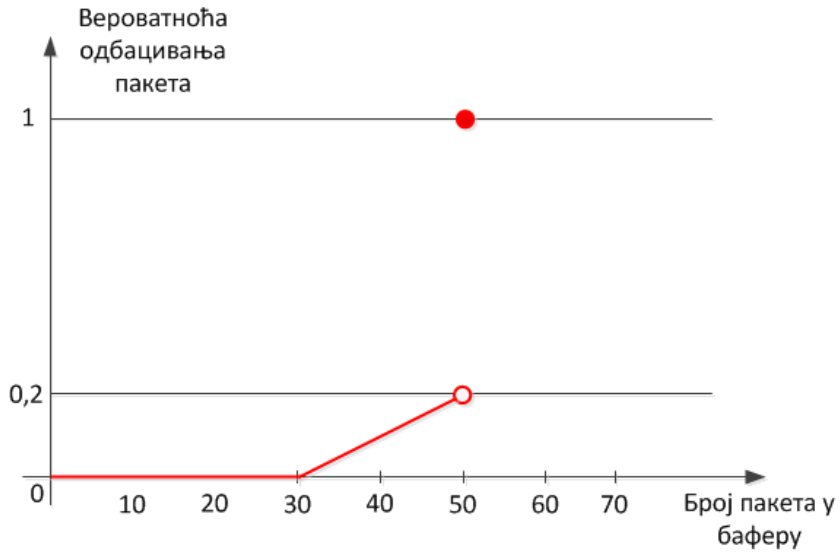
Показано је да до глобалне синхронизације долази у ситуацијама када постоји мањи број истовремених TCP сесија на линку (нпр. 100 до 500) [6.18], док у ситуацијама са великим бројем TCP сесија овај ефекат није изражен јер је број сесија које ће изгубити пакете релативно мали у поређењу са укупним бројем сесија, а и динамика сесија може да буде врло различита.

До глобалне синхронизације долази услед тога што у тренутку када је бафер на улазу на загушени линк сви пакети бивају одбачени што погађа више TCP сесија од једном. График зависности одбацивања пакета од величине бафера за бафер величине 50 пакета је показан на слици 6.20. Док бафер није потпуно попуњен, пакети неће бити одбацивани, када се напуни, сви пакети ће бити одбачени.



Слика 6.20 Вероватноћа одбацивања пакета у баферу величине 50 пакета

6.4.4.1. Случајно одбацивање пакета – RED

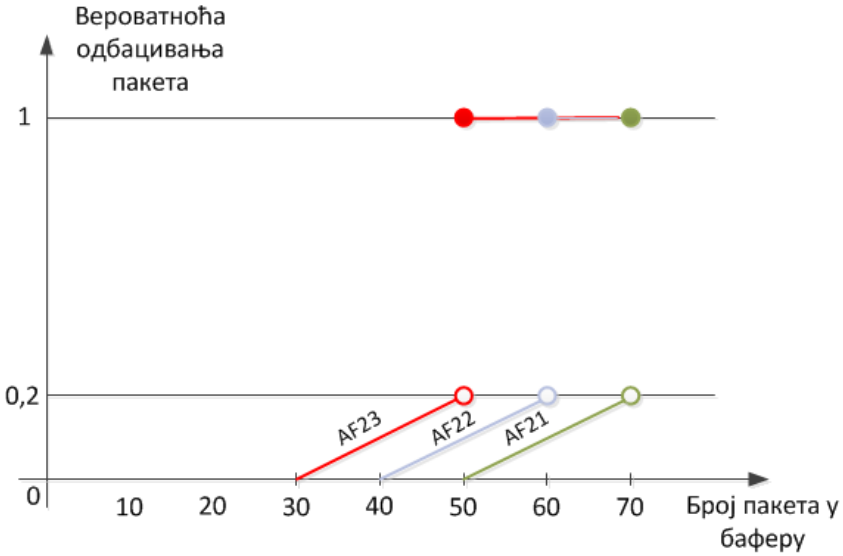


Слика 6.21 Вероватноћа одбацивања пакета у баферу величине 50 пакета са случајним одбацивањем (RED)

Механизам којим се спречава ефекат глобалне синхронизације је механизам раног откривања загушења (енг. *Random Early Detection – RED*) и случајног превентивног одбацивања пакета [6.19]. Код овог механизма је уведено да пакети могу да буду одбачени и пре него што је бафер потпуно попуњен. Ово је показано на слици 6.21. Од неке вредности попуњености бафера (у примеру 30 пакета) вероватноћа одбацивања пакета је различита од нуле. Ово значи да ће пакети који тада долазе на загушени излазни линк имати неку ненулту вероватноћу да буду одбачени. То даље значи да ће неке, а не све TCP сесије почети да губе пакете и пре него што је бафер потпуно попуњен, а тиме се постиже то да се неће десити да више TCP сесија синхронизовано изгуби пакете чиме долази до наглог пада протока на загушеном линку.

6.4.4.2. Случајно одбацивање пакета са тежинским фактором - WRED

Претходни механизам једнако третира све пакете. Како је цео смисао механизама квалитета сервиса да се на неки начин дискриминишу пакети и да им се да различит третман у мрежи, уведена је модификација RED механизма који може да дефинише различите вероватноће одбацивања пакета за различите класе пакета.



Слика 6.22 Вероватноћа одбацивања пакета у баферу величине 50 пакета са случајним одбацивањем и тежинским фактором (WRED)

На слици 6.22 је показан један такав пример у коме су дефинисана три различита профила губитака пакета за пакете са три различите DSCP ознаке: AF21, AF22 и AF23. Ови пакети припадају другој AF класи пакета. Са графика се види да ће пакети са ознакама AF23 моћи да буду одбацивани са неком ненултом вероватноћом већ када број пакета у баферу пређе 30 пакета. Када је у баферу 50 пакета, сви пакети са ознаком AF23 ће бити одбачени, приближно 10% пакета са ознаком AF22 ће бити одбачено, док се пакети са ознаком AF21 неће уопште одбацивати. Ово управо објашњава на који начин се третирају пакети у алгоритмима ограничавања који се условно пропуштају кроз мрежу – променом ознаке пакета са AF21 на AF23 ће се учинити да пакети са промењеном ознаком буду одбачени у ситуацијама када постоји загушење у мрежи пре оних пакета који у прослеђени као пакети који су у складу са договореним протоком.

6.4.4.3. Обавештавање о загушењу - ECN

Приликом стандардизовања DSCP поља у IP заглављу два најмање значајна бита осмобитног поља *Type of Service* су остала у том тренутку недефинисана (Слика 6.6). Неколико година након тога ова два бита су добила своје значење – то су два бита која се користе за сигнализацију загушења – енг. *Explicit Congestion Notification* – ECN [6.20]. Као што је претходно показано, TCP протокол посредно закључује да је дошло до загушења кроз

недостатак потврде за пакете који су изгубљени у преносу. За разлику од тога ECN бити служе за експлицитно обавештавање о томе да је дошло до загушења и пре губитка пакета. ECN механизам подразумева сарадњу мрежног слоја, јер се означавање пакета који су наишли на загушење врши сетовањем ових бита у IP заглављу и транспортног слоја – TCP протокола који треба да коригује количину послатих података у случају када дође до загушења.

Ако обе стране подржавају ECN механизам, у пакетима које шаљу ће бити вредности 01 и 10 бинарно, а ако не подржавају, вредност 00. Уколико дође до загушења, рутер на којем је детектовано загушење може да сетује ECN бите на вредност 11 бинарно (енг. *Congestion Encountered* - CE) и на тај начин да обавести дестинацију да је на проласку пакета дошло до заушења. Након тога дестинација истим битима може да обавести пошиљаоца да су пакети које шаље наишли на загушење и да је потребно да предузме неку меру на транспортном слоју како би се избегло потпуно заузеће бафера и губитак пакета.

ECN механизам у оквиру TCP протокола је подржан помоћу три нова флега у TCP заглављу:

- *Nonce Sum* (NS) је додатни бит којим се проверава функционисање ECN механизма и спречавају лажна обавештења о загушењу или сакривање ових информација
- *ECN-Echo* (ECE) којим дестинација по добијању вредности 11 - CE за ECN у IP заглављу обавештава пошиљаоца да је дошло до загушења на путу пакета које је послао.
- *Congestion Window Reduced* (CWR) којим пошиљалац пакета који су наишли на загушење потврђује дестинацији да је примљен ECE и да је смањио величину прозора.

Употреба ECN у оквиру TCP се успоставља приликом иницијалног *handshake*-а. Данас практично сви оперативни системи подржавају ECN механизам, као и рутери свих значајнијих произвођача мрежне опреме. Изузев у Apple оперативним системима (од iOS 9.3.5 и OSX 10.12.15) где је укључен у оквиру подразумеваних подешавања, у свим осталим оперативним системима мора да се посебно активира. Стога је у анализама пакета на интернету пронађен сразмерно мали број пакета са ECN битима: око 3,5% TCP сесија је захтевало ECN механизам приликом иницијалног *handshake*-а, а мање од 1% IP пакета је имало ове бите сетоване иако је процена је да преко 70% најзначајнијих веб сајтова данас подржава овај протокол [6.21].

6.5. Друге технике побољшања квалитета сервиса

Све претходно наведене технике спадају у механизме обезбеђивања квалитета сервиса у оквиру архитектуре диференцираних сервиса. Постоје још неки механизми којима могу да се

побољшају перформансе неких параметара преноса пакета који не спадају стриктно у ову архитектуру. Они су описани у наставку поглавља.

6.5.1. Фрагментација пакета

На врло спорим везама време серијализације великих пакета може да створи ситуацију да и онда када нема загушења неки пакети добију неприхватљиво велико кашњење. Ово може да буде показано на следећем примеру.

Ако на рутер долазе два мрежна тока, један који носи сигнал гласа (нпр. IP телефонија) кодован неким савременим кодеком попут G.729, и други који носи пакете неке веб сесије. Први пакети се генеришу сваких 20ms и величине су око 100 бајтова (релативно мали), док су други пакети макималне величине (MTU=1500 бајтова). Ако је излазни линк малог капацитета, нпр. 128Kbit/s, време серијализације великог пакета ће бити око 100ms. Ово значи да и у ситуацији када нема загушења на излазном линку долазак само једног великог пакета који припада другом току може да изазове додатно кашњење пакета са сигналом гласа од 80ms и значајно нарушавање периодичности пакета. Овако велико додатно кашњење је веће од препоручених лимита за џитер и требало би да се некако смањи.

За решавање овог проблема неки значајни произвођачи мрежне опреме (нпр. Cisco, Juniper) су предвидели механизам фрагментације пакета (тачан назив на енглеском овог механизма је *Link Fragmentation and Interleaving* - LFI). Ова фрагментација није иста као фрагментација на нивоу IP протокола, већ је ово механизам који се реализује само на крајевима једне везе између два рутера. Применом овог механизма велики пакет од 1500 бајтова би могао да се фрагментира на уласку на спори линк на 5 делова од по 300 бајтова и тако пошаље. Пакети са сигналом гласа могу да се распореде између фрагмената великог пакета што не би произвело негативан ефекат по њих јер је време серијализације фрагмента око 20ms. На долазној страни спорог линка се велики пакет спаја у једну целину.

6.5.2. Компресија података и заглавља

Једно логично решење за смањење појаве загушења је компресија података који се шаљу мрежом. Постоје две стратегије компресије које могу да се примене: компресија података који се преносе пакетима и компресија заглавља. Иако је из осталих области рачунарства очекивано да компресија података може да да добре резултате у смањењу обима послатих података, у наредним поглављима ће бити показано зашто то није тако и зашто су некада технике компресије заглавља много ефикасније од компресије података.

6.5.2.1. Компресија података

Алгоритми за компресију података користе неке статистичке особине оригиналних података који омогућавају да се за одређену врсту података (нпр. текст, неки фајлови) без губитка

информација подаци компримују на неколико пута мању величину од оригиналне, док за друге податке (звук, слика, видео) компресија може да се изврши или без губитка квалитета у мањем степену компресије (нпр. *flac* формат за компресију звука) или у већој мери уз прихватљиву деградацију квалитета оригиналног фајла (нпр. mp3 формат за компресију звука, jpg за слике итд.).

Мрежни уређаји могу да буду опремљени софтвером који може да компримује део пакета у којем су подаци. Алгоритми који се користе за ову сврху су алгоритми за компресију без губитка података, који раде на принципима који су најчешће исти или слични алгоритмима који се користе за стандардну компресију фајлова (*Predictor*, *Lempel-Ziv-Stac*). Природно, додавање функције компресије приликом процесирања пакета повећава комплексност обраде, додатно оптерећује процесор мрежног уређаја и уводи додатно кашњење. Међутим, и ово може да буде прихватљиво ако би се добила нека уштеда у протоку информација компримовањем садржаја пакета. На жалост, данас су ови механизми потпуно неупотребљиви из два разлога: данас је више од половине целокупног саобраћаја на интернету криптовано (користи се HTTPS) и данас је више од 70% целокупног интернет саобраћаја видео садржај. Како криптовани садржај има потпуно другачију статистичку структуру од оригиналног формата, компримовањем криптованог садржаја поменутим алгоритмима се не постиже никакав степен компресије, а пакети бивају дуже процесирани. Слично томе, видео садржај се кроз мреже преноси у већ компримованом формату (данас најчешће нека варијанта H.264 – mpeg4 формата), тако да додатни алгоритам компресије такође неће имати никакав ефекат осим негативног – дужег процесирања пакета.

6.5.2.2. Компресија заглавља

У поглављу 6.1.1 је већ дата анализа дела протока који одлази на за корисника мреже некорисне, али ипак нужне контролне инормације. За мале пакете овај однос је изразито неповољан. На пример, пакети гласа који се шаљу кодовани G.729 стандардом се шаљу сваких 20ms и у тим пакетима је део са подацима величине 20 бајтова. Ако се пакет шаље преко етернет мреже, укупна величина свих заглавља у таквом пакету је (етернет+IP+UDP+RTP) је 78 бајтова. То значи да је у пакету укупне величине 98 бајтова, само 20,4% корисних информација за корисника. У овим ситуацијама компресија заглавља може да оствари много веће уштеде него компресија података, која као што је показано није ефикасна у савременим мрежама.

За претходно поменуту примену је најбоље користити механизам којим се компримују IP, UDP и RTP заглавља са почетних 40 на свега два или четири бајта [6.22], чиме би се укупна величина пакета из претходног примера смањила са 98 на 60 бајтова, односно направила би се уштеда од 40%. Сличан механизам постоји за компресију TCP и IP заглавља где се са 40 заглавље компримује на 2 или 4 бајта. Оба ова механизма могу да постоје само на пакетима на једној вези између два рутера (веза која је малог капацитета или склона загушењу), а не могу с краја на крај на интернету јер се компресијом IP заглавља губи инормација на основу које би рутери прослеђивали пакете. Механизам компресије заглавља функционише тако што

рутер који шаље пакете на везу на којој је успостављена компресија заглавља шаље другом рутеру на вези информацију којим бројем (величине 2 или 4 бајта) који се ставља уместо заглавља су означена одређена заглавља. Рутери треба да чувају таβеле у којима су са једне стране комплетна заглавља пакета, а са друге стране бројеви који их замењују. По пријему пакета са компримованим заглављем, рутер мора да рестаурира оригинално заглавље и да пакет проследи даље.

Упркос јасним добицима компресијом заглавља малих пакета ни овај механизам се не користи често јер је у рачунарским мрежама ситуација таква да има много више великих пакета (1500 бајтова) за које је уштеда остварена на овај начин занемарљива.

6.6. Литература

[6.1] „Packet size distribution comparison between Internet links in 1998 and 2008". http://www.caida.org/research/traffic-analysis/pkt_size_distribution/graphs.xml

[6.2] MEF 10.3 Technical specification – Ethernet Services Attributes Phase 3, October 2013 - https://www.mef.net/Assets/Technical_Specifications/PDF/MEF_10.3.pdf

[6.3] Y.1540 - Internet protocol aspects – Quality of service and network performance – July 2016 - https://www.itu.int/rec/T-REC-Y.1540/en

[6.4] Testimony of Gary R. Bachula, Vice President, Internet2 Before the United States Senate Committee on Commerce, Science and Transportation, Hearing on Net Neutrality,February 7, 2006, https://www.commerce.senate.gov/pdf/bachula-020706.pdf

[6.5] R. Braden, D. Clark, S. Shenker, Integrated Services in the Internet Architecture: an Overview, IETF RFC 1633, June 1994, https://tools.ietf.org/html/rfc1633

[6.6] https://msdn.microsoft.com/en-us/library/windows/desktop/aa374050(v=vs.85).aspx

[6.7] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An Architecture for Differentiated Services, IETF RFC 2475, December 1998., https://tools.ietf.org/html/rfc2475

[6.8] K. Nichols, S. Blake, F. Baker, D. Black, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, IETF RFC 2474, December 1998., https://tools.ietf.org/html/rfc2474

[6.9] A. Demers, S. Keshav, S. Shenker, „Analysis and simulation of a Fair Queueing Algorithm", Proceedings of SIGCOMM '89, CCR Vol 19. No. 4 Austin TX USA, September 1989 pp. 1-12.

[6.10] Parekh, A. K.; Gallager, R. G. (1993). "A generalized processor sharing approach to flow control in integrated services networks: The single-node case". IEEE/ACM Transactions on Networking. 1 (3): 344. doi:10.1109/90.234856

[6.11] Understand and Configure MDRR/WRED on the Cisco 12000 Series Internet Router, March 2008, https://www.cisco.com/c/en/us/support/docs/routers/12000-series-routers/18841-mdrr-wred-18841.html

[6.12] Analyzing UDP usage in Internet traffic, CAIDA report, https://www.caida.org/research/traffic-analysis/tcpudpratio/ приступљено 27.12.2017.

[6.13] S. Ha, I. Rhee, L. Xu, CUBIC: a new TCP-friendly high-speed TCP variant. *SIGOPS Oper. Syst. Rev.* 42, 5 (July 2008), 64-74. DOI=http://dx.doi.org/10.1145/1400097.1400105

[6.14] M. Mathis, N. Dukkipati, Y. Cheng, Proportional Rate Reduction for TCP, IETF RFC 6937, May 2013, https://tools.ietf.org/html/rfc6937

[6.15] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, V. Jacobson. 2016. BBR: Congestion-Based Congestion Control. *Queue* 14, 5, pages 50 (October 2016), 34 pages. DOI: https://doi.org/10.1145/3012426.3022184

[6.16] K. Tan, J. Song, Q. Zhang, M. Sridharan, A Compound TCP Approach for High-speed and Long Distance Networks, Microsoft Research Report MSR-TR-2005-86, July 2005, https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2005-86.pdf

[6.17] S. Shenker, L. Zhang, D. Clark, Some observations on the dynamics of a congestion control algorithm. *SIGCOMM Comput. Commun. Rev.* 20, 5 (October 1990), 30-39. DOI=http://dx.doi.org/10.1145/381906.381931

[6.18] G. Appenzeller, I. Keslassy, N. McKeown, Sizing router buffers. *SIGCOMM Comput. Commun. Rev.* 34, 4 (August 2004), 281-292. DOI: https://doi.org/10.1145/1030194.1015499

[6.19] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," in *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397-413, Aug 1993. doi: 10.1109/90.251892

[6.20] K. Ramakrishnan, S. Floyd, D. Black, The Addition of Explicit Congestion Notification (ECN) to IP, IETF RFC 3168, September 2001., https://tools.ietf.org/html/rfc3168

[6.21] D. E. Murray, T. Koziniec, S. Zander, M. Dixon, P. Koutsakis, An Analysis of Changing Enterprise Network Traffic Characteristics, The 23rd Asia-Pacific Conference on Communications (APCC 2017). 11-13 December 2017, Perth, Australia

[6.22] T. Koren, S. Casner, J. Geevarghese, B. Thompson, P. Ruddy, Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering, IETF RFC 3545, July 2003, https://tools.ietf.org/html/rfc3545

7. Практична реализација мрежних технологија

У овом поглављу су дати примери практичне реализације мрежних технологија описаних у претходним поглављима. Реализација примера је на урађена на симулатору који може да се користи на стандардним персоналним рачунарима, тако да студенти могу да користе ово поглавље и за самостални рад и за припрему лабораторијских вежби које се изводе на истом систему. Сви примери су објашњени корак по корак са датим свим конфигурационим командама које су потребне да би се остварила жељена функционалност, те је могућа потпуна репродукција показаних резултата, али и проширење и детаљнија анализа по потреби.

Прве две вежбе су припремне и обухватају теме из очекиваног предзнања студента које је стекао на предмету Рачунарске мреже 1: интерни протокол рутирања RIP и редистрибуцију рута. Наредне вежбе представљају скуп одабраних тема из области које покрива ова књига: BGP, *Frame mode* MPLS, MPLS VPN, IPsec VPN, мултикаст и SNMP. Предвиђено је да се вежбе раде редом како су изложене у наредним поглављима, јер се поједини елементи рада са симулатором уводе поступно, тако да би прескакање појединих вежби довело до тежег разумевања рада и показаних концепата.

7.1. Мрежни симулатор GNS3

На вежама и лабораторијским вежбама у оквиру предмета Рачунарске мреже 2 се користи симулатор рачунарских мрежа GNS3 (скраћено од *Graphical Network Simulator 3*)[51]. Овај симулатор омогућава креирање сложених рачунарских мрежа и великог броја мрежних функционалности на потпуно реалистичан начин захваљујући томе што се у оквиру њега користе стварни оперативни системи и софтвер мрежних уређаја. Прве верзије GNS3 су биле

51 http://www.gns3.net/

направљене за симулацију рада Cisco рутера са IOS оперативним сиситемом, док је у најновијим верзијама могуће симулирање рада рутера и других произвођача помоћу виртуелних машина, као и других уређаја (свичеви, радне станице, итд.). У свим примерима у наставку текста ће бити показан начин рада на Cisco рутерима са IOS оперативним сиситемом јер су студенти већ навикнути на тај начин рада из претходног предмета у којем је коришћен алат *Packet Tracer*[52].

GNS3 симулатор је погодан пре свега за учење начина рада и имплементације различитих мрежних протокола и механизама. Није погодан за рад у реалном времену и провере перформанси мреже (па тиме и за проверу рада механизама квалитета сервиса) јер интерфејси виртуелних рутера и везе између њих не одсликавају верно капацитете веза и стварни начин и динамику прослеђивања пакета. Симулатор је могуће повезати путем рачунара на којем се извршава и у реалне мреже чиме рачунар на којем се извршава симулација може да се понаша као прави мрежни уређај са свим функционалностима које он има, а могу да се добију у њему подаци са реалних мрежа (на пример таβеле рутирања).

GNS3 симулатор је бесплатан, ради на већини популарних оперативних система (Windows, Linux, Mac OS, FreeBSD) и може се преузети на адреси пројекта уз обавезну регистрацију. Симулатор ради тако што се на рачунару на коме се извршава покрећу виртуелне машине у којима су оперативни системи правих рутера и других подржаних мрежних уређаја. Сваки виртуелни рутер користи онолико RAM меморије рачунара колико реално има хардверски рутер (нпр. 64MB до 256MB за мање рутере који су погодни за вежбе у оквиру предмета), тако да је потребно проценити да ли рачунар има довољно слободне меморије за креирање топологије која се жели. Симулатор се не испоручује са оперативним системима рутера, већ их је потребно набавити посебно.

7.1.1. Инсталација и почетна конфигурација мрежног симулатора GNS3

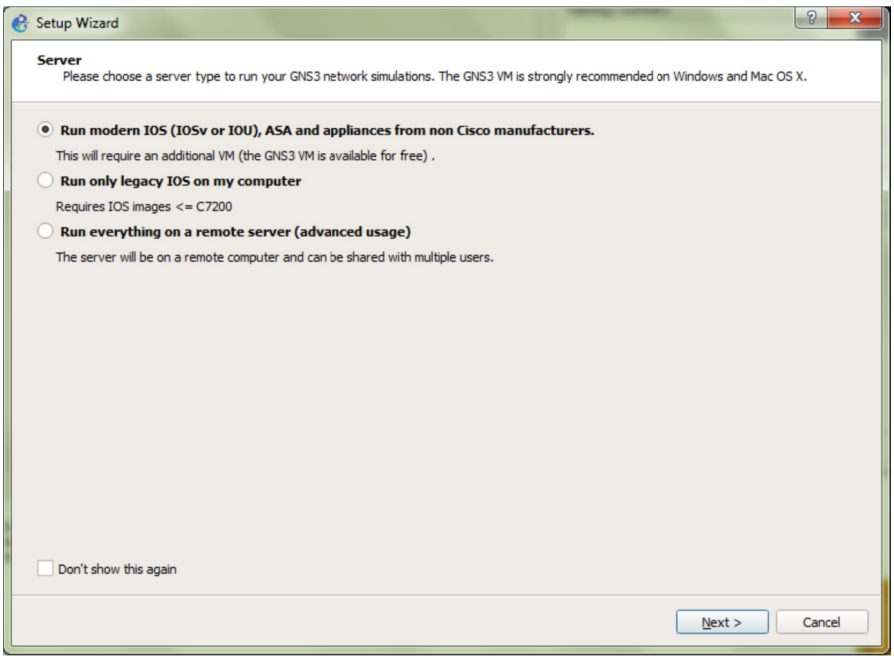
Ово упутство се односи на верзију GNS3 2.0.3 за Windows оперативни систем. Подешавања су врло слична и за остале платформе и новије верзије симулатора, али неки кораци могу да се разликују у односу на приказано у наставку текста.

Најједноставнији начин инсталирања је преузимањем тзв. „*all-in-one*” верзије симулатора у којој се налазе и други програми који могу да буду корисни у анализи рада мреже (нпр. Wireshark, Npcap и/или WinPCAP за анализу пакета који пролазе мрежом, dynamips и QEMU симулатори итд.). Инсталација је релативно једноставна уз неколико одговора на питања о инсталацији зависних компоненти (у случају да су неке већ инсталиране, да је потребна новија њихова верзија итд.). Након инсталације могуће је одмах стартовати симулатор.

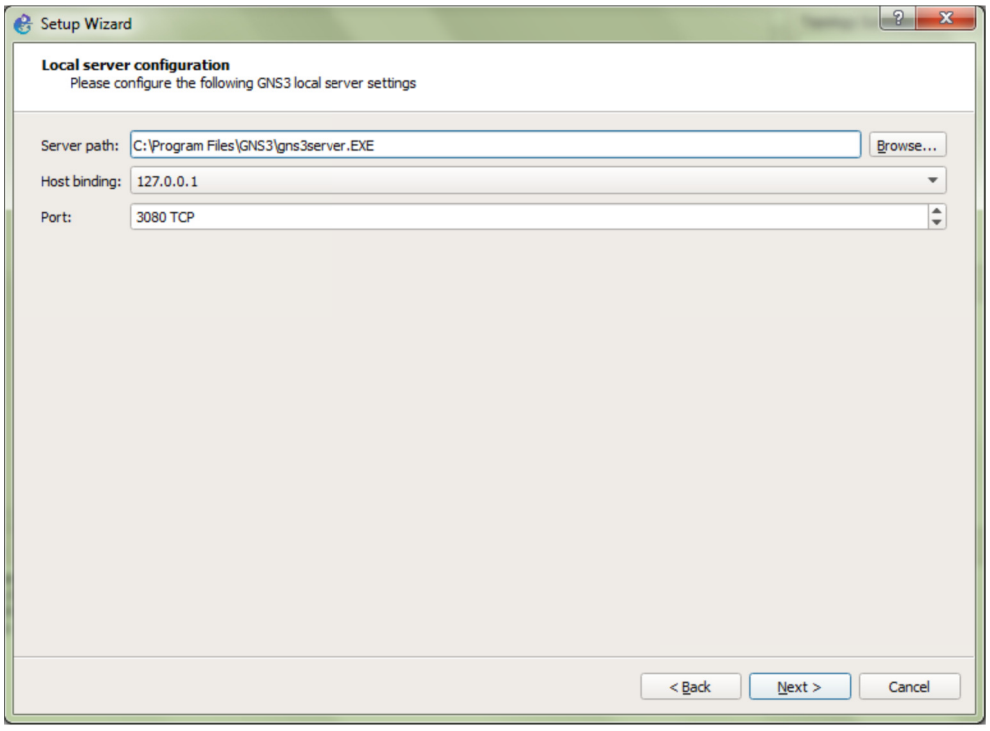
52 https://www.netacad.com/courses/packet-tracer-download/

7.1.2. Подешавање Dynamips сервера

Први прозор који се појављује по првом стартовању симулатора даје избор врсте сервера симулатора која ће се користити за симулације (слика 7.1).



Слика 7.1 Подешавање Dynamips сервера



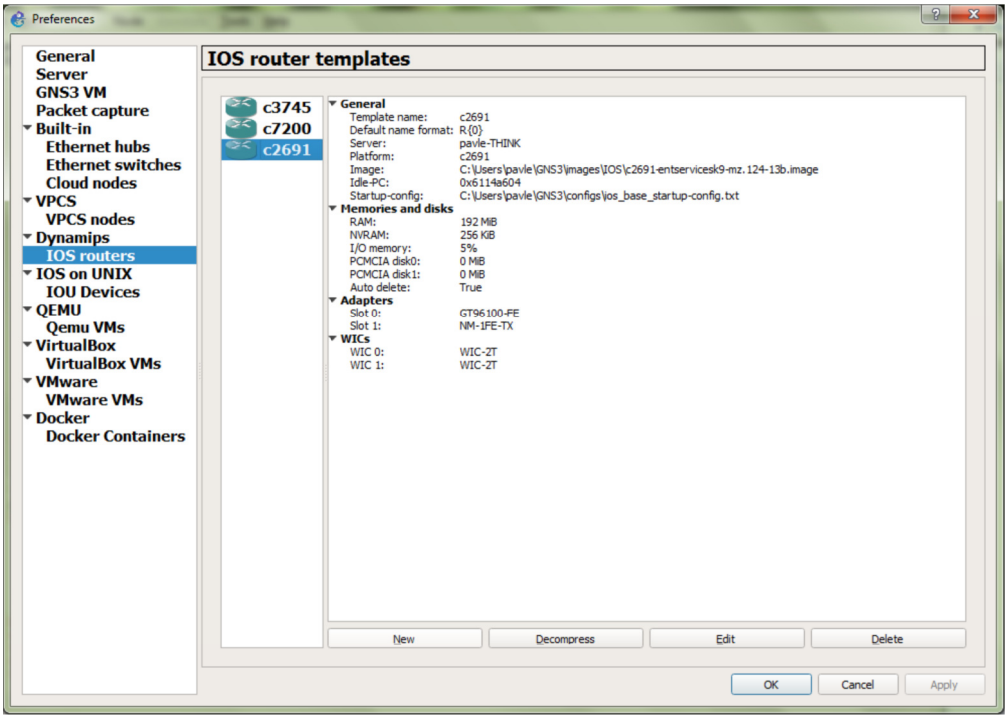
Слика 7.2 Основна конфигурација сервера симулатора GNS3

За потребе предмета Рачунарске мреже 2 је довољно изабрати опцију „*Run only legacy IOS on my computer*“, јер ће све лабораторијске вежбе у оквиру предмета бити изведене на рутерима Cisco Systems са IOS оперативним системом, док је за симулације са рутерима других произвођача и виртуелним машинама потребно изабрати прву опцију. Након избора треба кликнути *Next*, после чега, следећи екран даје основну конфигурацију сервера симулатора која изгледа као на слици 7.2 и након прихватања подразумеване конфигурације треба кликнути поново на *Next*.

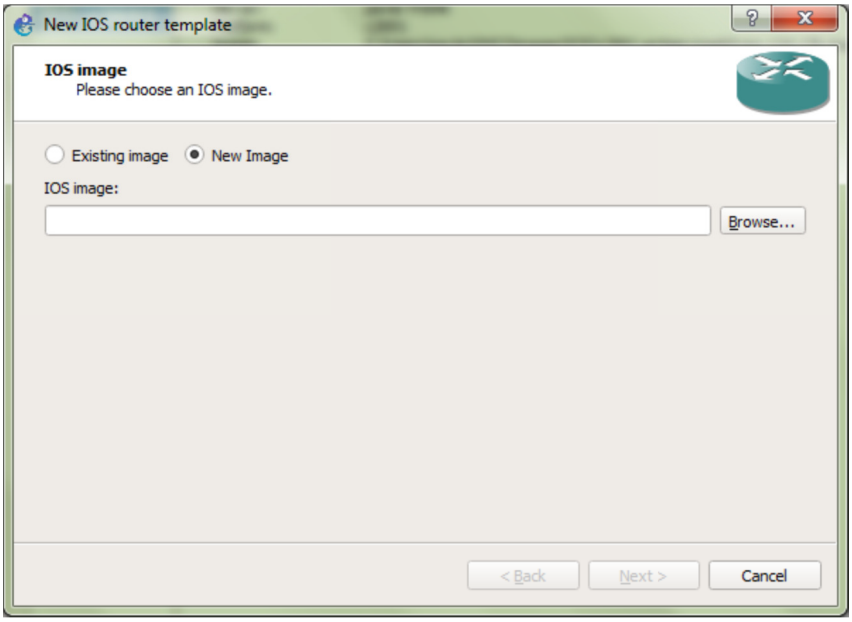
7.1.3. Учитавање оперативних система рутера

Након конфигурације сервера потребно је учитати оперативне системе рутера. То се ради у оквиру секције *Edit/Preferences/Dynamips/IOS routers* кликом на *New* као на слици 7.3, а затим уносом одговарајућег фајла са оперативним системом (слика 7.4 и слика 7.5).

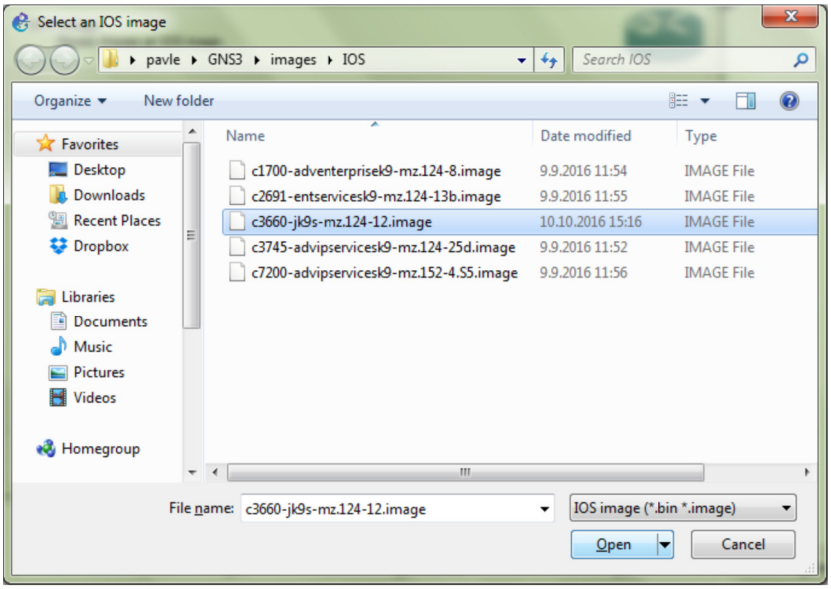
Након уноса оперативног система GNS3 ће препознати о којој рутерској платформи се ради (слика 7.6.) и захтеваће да се направи спецификација хардверске конфигурације виртуелног рутера. Ово се пре свега односи на додатне мрежне модуле којима могу да се додају додатни интерфејси на рутер (слика 7.7), а чиме се омогућава прављење сложенијих топологија са већим степенима повезивања сваког од рутера.



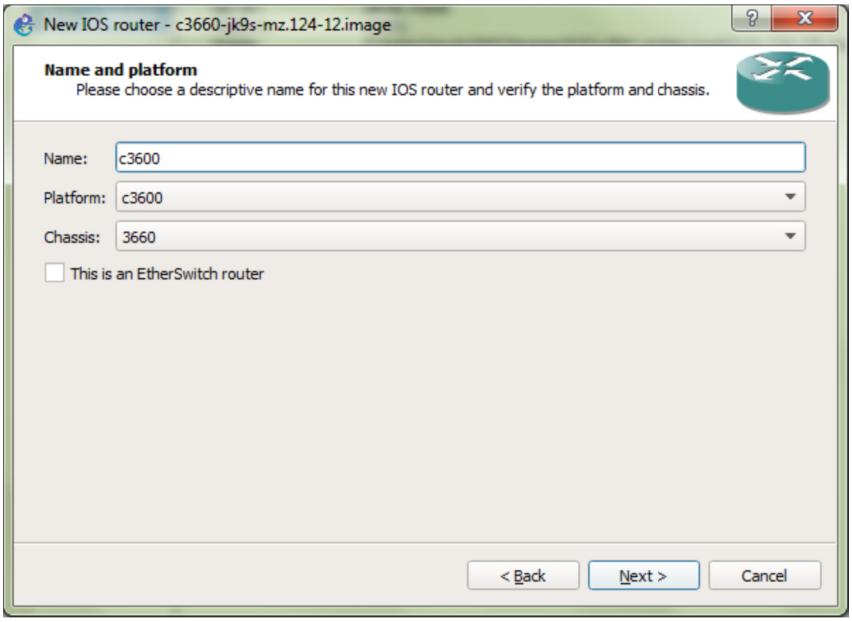
Слика 7.3 Прозор за уношење рутера у симулатор



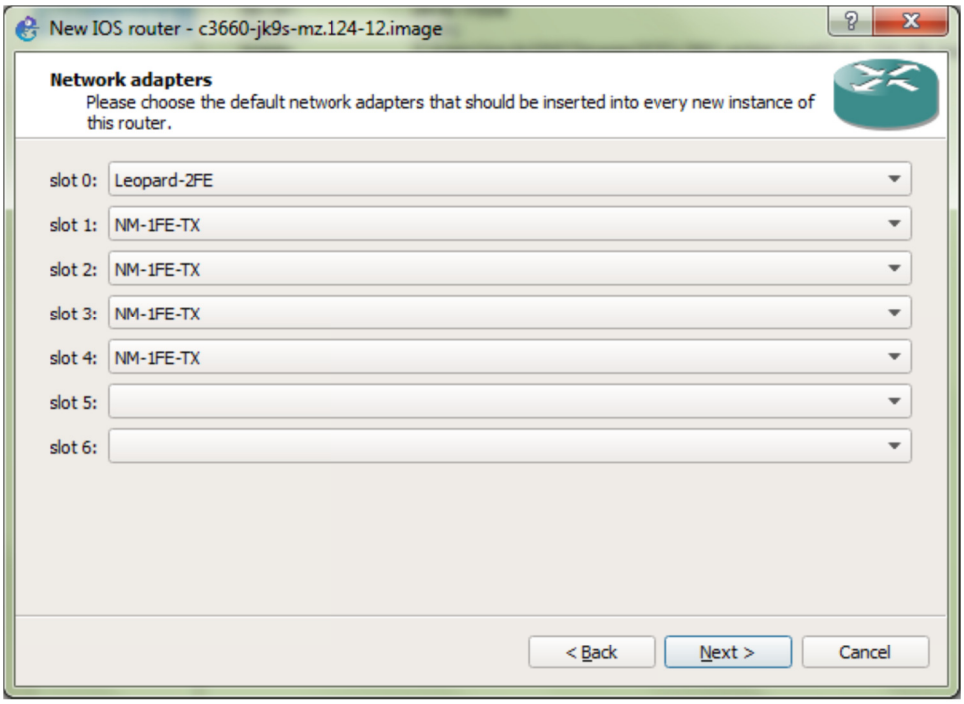
Слика 7.4 Прозор за уношење оперативних система рутера у симулатор (1)



Слика 7.5 Прозор за уношење оперативних система рутера у симулатор (2)



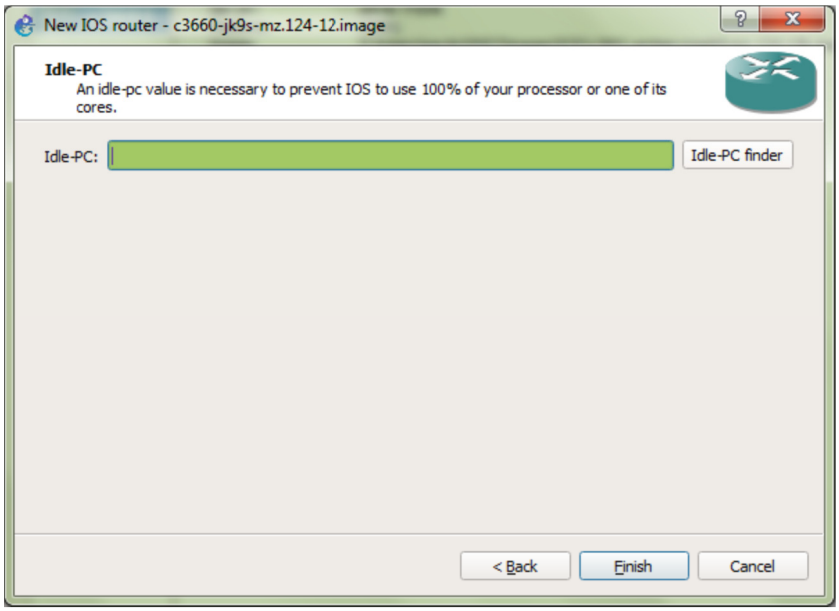
Слика 7.6 Избор платформе рутера



Слика 7.7 Спецификација хардверске конфигурације виртуелног рутера

Последњи корак у конфигурацији рутера је проналажење *Idle-PC* вредности за дати рутер (слика 8). Овај корак иако необавезан се препоручује зато што се на тај начин спречава ситуација у којој сваки емулирани рутер заузима процесор (или језгро на процесору) 100% и тиме се добија могућност да се креирају веће топологије које раде стабилно и без значајног оптерећења рачунара. Проналажењем *Idle-PC* вредности симулатор стиче могућност да буде

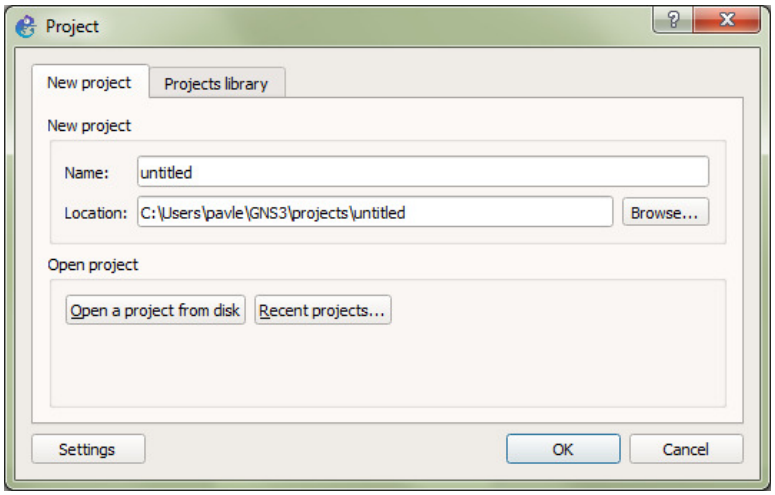
у *Sleep* режиму одређено време чиме може да препусти процесор другим процесима рачунара.



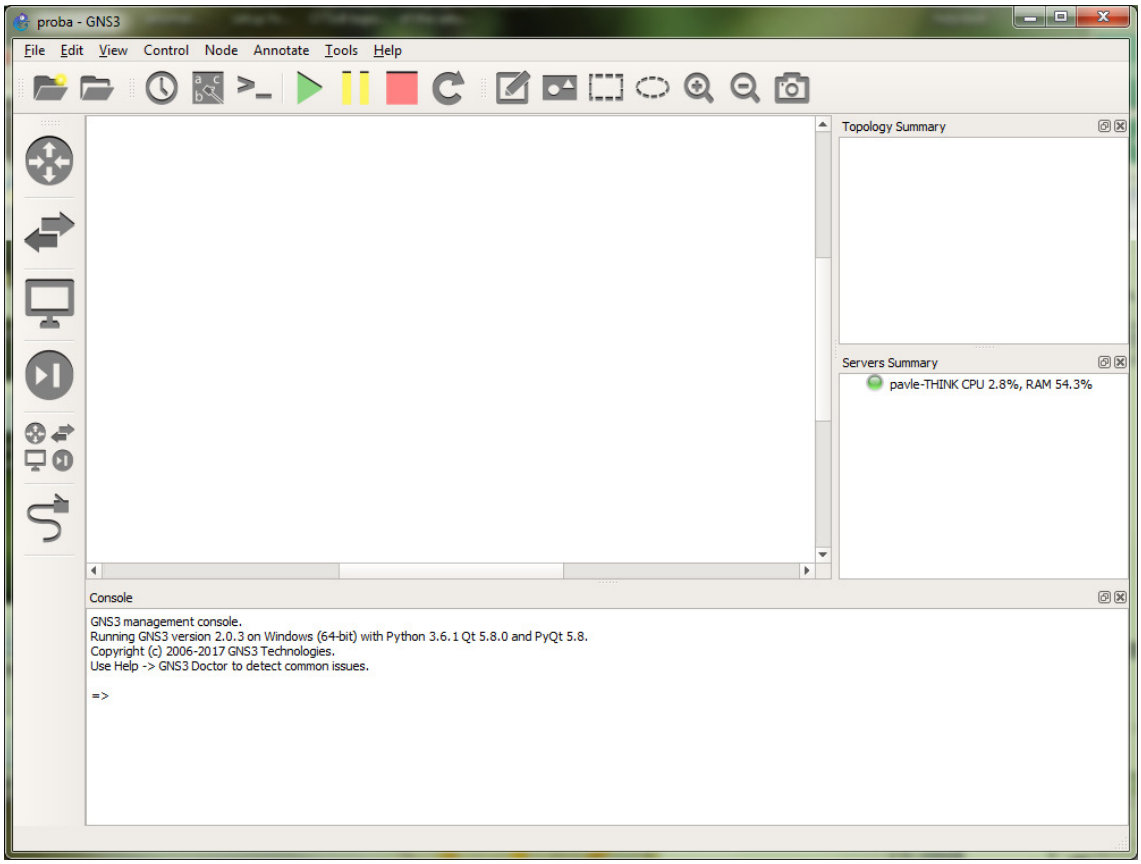
Слика 7.8 Проналажење idle-PC вредности

7.1.4. Стартовање симулација

По стартовању програма појавиће се дијалог у који је потребно унети име топологије односно пројекта на којем ће се радити као на слици 9. Пројекат се смешта у *default* директоријум који је могуће произвољно променити. Ту ће бити сви радни фајлови који се креирају током рада симулације. У оквиру овог прозора могуће је покренути и претходно снимљене симулације.



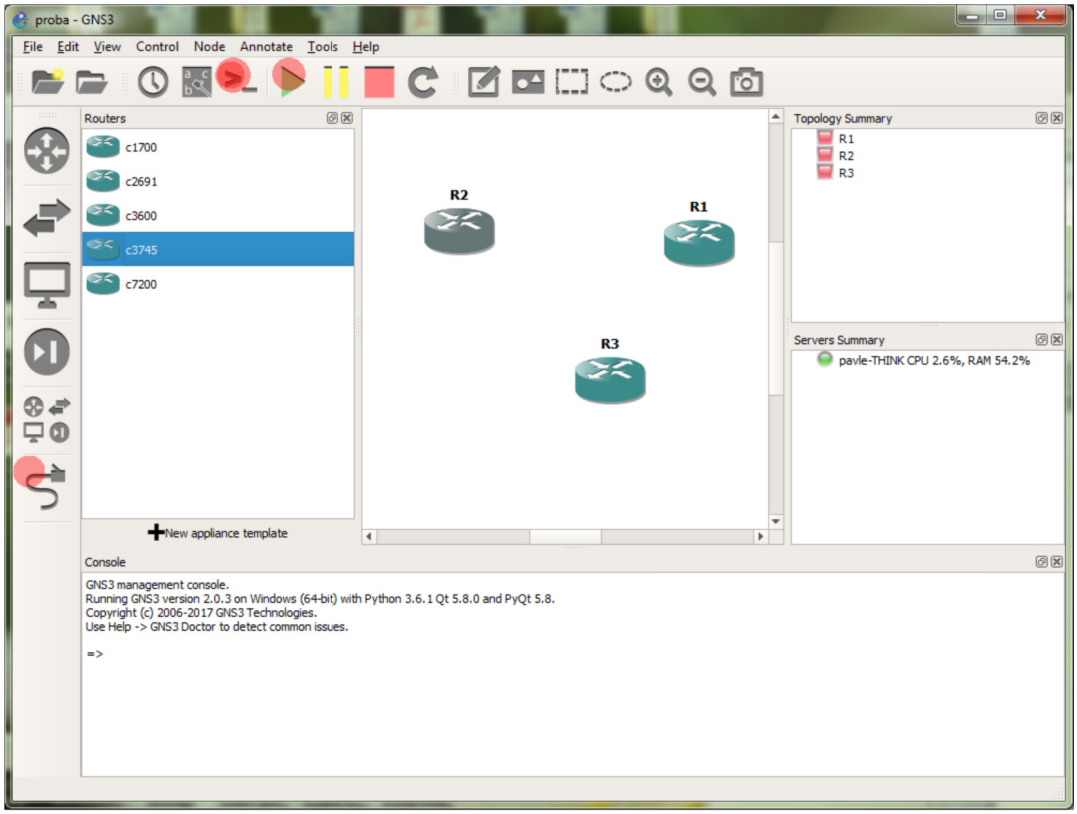
Слика 7.9 Отварање нове или покретање снимљене симулације



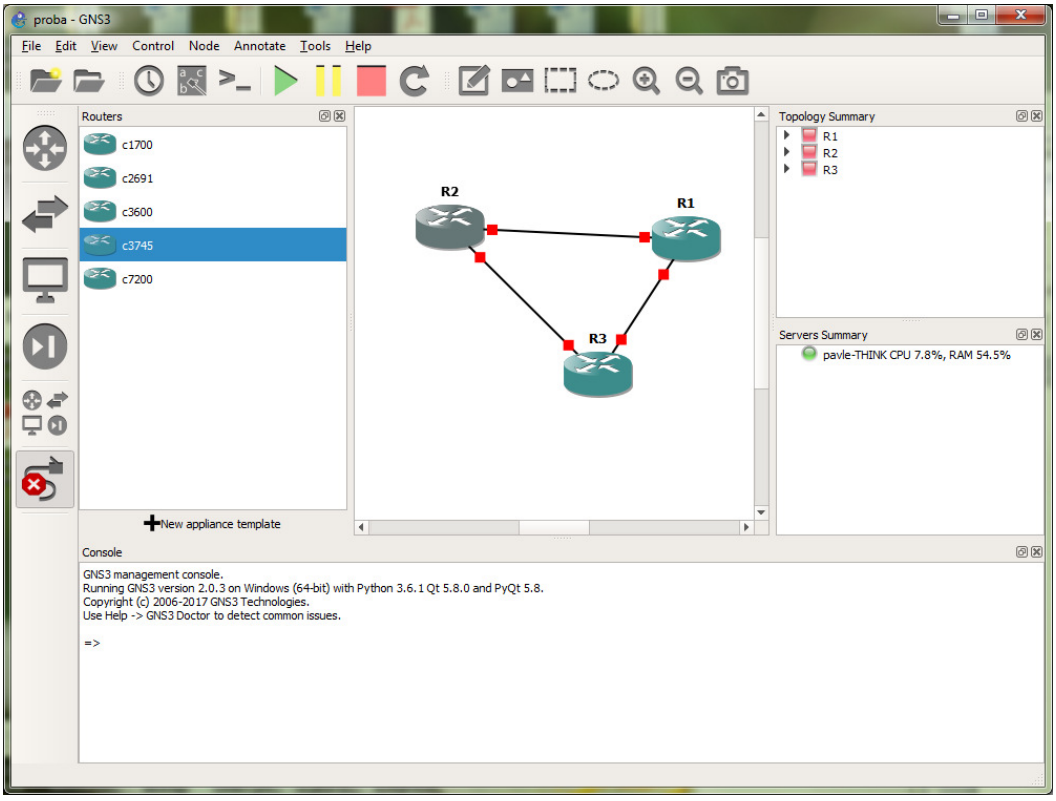
Слика 7.10 Основни прозор GNS3 симулатора верзија 2.0.3.

Сада је могуће креирати топологију мреже. То се ради тако што се превуку одговарајући рутери (тип за који је подешен оперативни систем) из левог поља у поље у којем се црта топологија, као на слици 7.10. На слици 7.11 су црвеним тачкама обележене иконе које ће се користити у наставку овог примера (са лева на десно): икона за конфигурацију веза између рутера, икона за покретање командних конзола рутера и икона за покретање рада свих рутера (старт симулације).

Кликом на икону за конфигурацију везе између рутера може да се изабере тип интерфејса између рутера. Рутери Cisco 3660 у основној конфигурацији имају два *Fast Ethernet* интерфејса али је могуће GNS3 симулатором додати још интерфејса уколико то није урађено на начин објашњен сликом 7.7 (десни клик на рутер који је у топологији, па *Configure*, па *Slots*). Затим треба одабрати *Fast Ethernet* интерфејсе и повезати превлачењем везе између рутера R1-R2, R1-R3 и R2-R3. Добиће се топологија као на слици 7. Црвене тачке на везама између рутера значе да везе нису успостављене. У овом случају то је због тога што рутери нису покренути. У десном пољу могуће је видети детаљније везе између рутера – који су интерфејси везани са којим, што се види на слици 7.12.

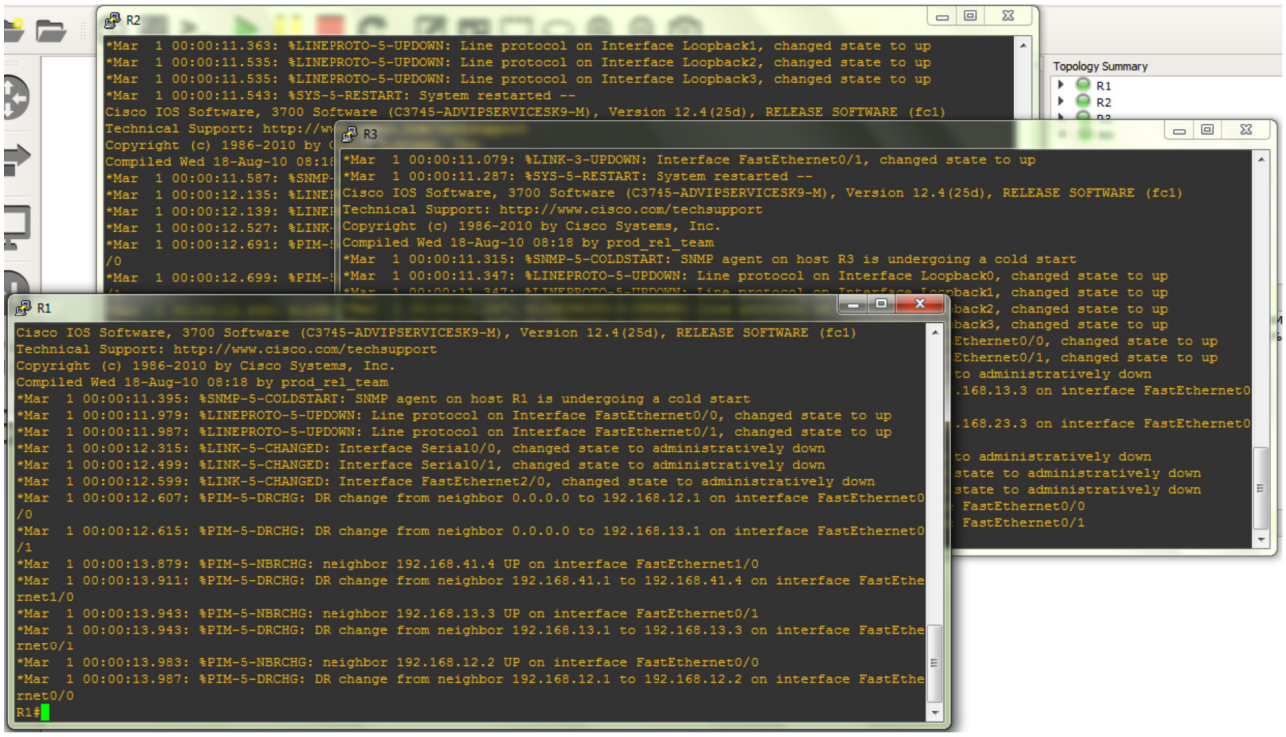


Слика 7.11 Прављење мреже превлачењем рутера у прозор за дефиницију топологије



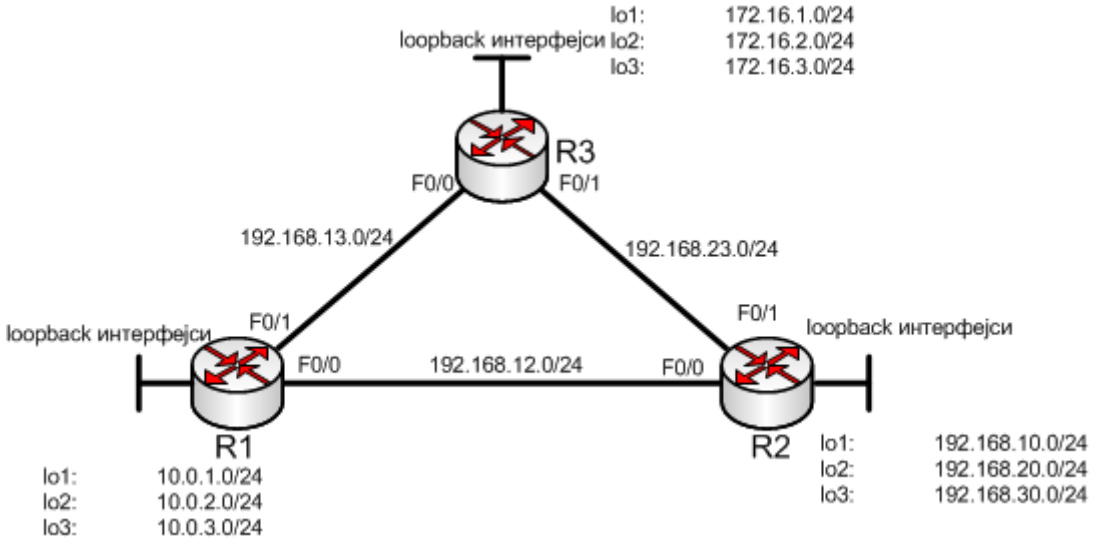
Слика 7.12 Повезивање рутера у топологији

Након овога треба кликнути на икону за почетак рада рутера и онда на икону за старт административних конзола рутера. Требало би да се отворе све три конзоле рутера као на слици 7.13.



Слика 7.13 Приступ рутерима преко конзолних портова

7.2. RIP протокол



Слика 7.14 Топологија мреже у примеру конфигурације RIP протокола

На слици 7.14 је дата мрежа на којој ће бити инсталиран RIP протокол рутирања. Адресе на портовима рутера су као на слици. Такође, сваки рутер има конфигурисана три тзв. *loopback* интерфејса. *Loopback* интерфејси су софтверски интерфејси који у овом случају симулирају

постојање мрежа повезаних на сваки од рутера. У алату GNS3 креирати приказану топологију, повезати рутере, укључити их и стартовати конзолни приступ.

7.2.1. Основна конфигурација рутера

Конфигурисати основне параметре (име рутера и IP адресе) на рутерима према следећем моделу: (не куцати коментаре који су иза знака !, као ни промпт **Router>, Router#, Router(config)#** и слично)

Рутер R1:

```
Router>enable                              !ulaz u privilegovani režim rada
Router#conf t                              !ulaz u konfiguracioni režim rada
Router(config)#no ip domain-lookup         !ne pretvara adrese u imena – brži
                                           !trace i ping
Router(config)#hostname R1                 !konfiguracija imena rutera
R1(config)#interface f0/0                  !ulazak u režim za konfiguraciju
                                           !interfejsa
R1(config-if)#ip address 192.168.12.1 255.255.255.0   !konfigurisanje IP adrese
                                                       !na datom interfejsu
R1(config-if)#no shutdown                        !uključivanje interfejsa, po
                                                 !defaultu je shutdown
R1(config-if)#interface f0/1
R1(config-if)#ip address 192.168.13.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#interface loopback1
R1(config-if)#ip address 10.0.1.1 255.255.255.0
R1(config-if)#interface loopback2
R1(config-if)#ip address 10.0.2.1 255.255.255.0
R1(config-if)#interface loopback3
R1(config-if)#ip address 10.0.3.1 255.255.255.0
R1(config-if)#Ctrl-z                       !izlaz iz konfig režima rada i
R1#                                        !povratak u privilegovani
```

Рутер R2:

```
Router>enable
Router#conf t
Router(config)#no ip domain-lookup
Router(config)#hostname R2
R2(config)#interface f0/0
R2(config-if)#ip address 192.168.12.2 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#interface f0/1
R2(config-if)#ip address 192.168.23.2 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#interface loopback1
R2(config-if)#ip address 192.168.10.1 255.255.255.0
R2(config-if)#interface loopback2
R2(config-if)#ip address 192.168.20.1 255.255.255.0
R2(config-if)#interface loopback3
R2(config-if)#ip address 192.168.30.1 255.255.255.0
R2(config-if)#Ctrl-z
R2#
```

Рутер R3:

```
Router>enable
Router#conf t
Router(config)#no ip domain-lookup
Router(config)#hostname R3
R3(config)#interface f0/0
R3(config-if)#ip address 192.168.13.3 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#interface f0/1
R3(config-if)#ip address 192.168.23.3 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#interface loopback1
R3(config-if)#ip address 172.16.1.1 255.255.255.0
R3(config-if)#interface loopback2
R3(config-if)#ip address 172.16.2.1 255.255.255.0
R3(config-if)#interface loopback3
R3(config-if)#ip address 172.16.3.1 255.255.255.0
R3(config-if)#Ctrl-z
R3#
```

Откуцати на сваком рутеру X: `RX#show ip interface brief` и тиме проверити статус свих интерфејса на свим рутерима. На слици 7.15 се види пример излаза ове команде за рутер R1 у ситуацији када су интерфејси коректно конфигурисани, када им је статус *up/up*.

```
R1#sh ip int brie
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            192.168.12.1    YES manual up                    up
Serial0/0                  unassigned      YES unset  administratively down down
FastEthernet0/1            192.168.13.1    YES manual up                    up
Serial0/1                  unassigned      YES unset  administratively down down
FastEthernet1/0            unassigned      YES unset  administratively down down
FastEthernet2/0            unassigned      YES unset  administratively down down
Loopback1                  10.0.1.1        YES manual up                    up
Loopback2                  10.0.2.1        YES manual up                    up
Loopback3                  10.0.3.1        YES manual up                    up
```

Слика 7.15 Испис команде `show ip interface brief` на рутеру 1 када су интерфејси добро конфигурисани

Проверити функционалност веза – на рутеру R1 покренути:

```
ping 192.168.13.3
```

```
ping 192.168.12.2
```

на рутеру R2 покренути

```
ping 192.168.23.3
```

Уколико је све у реду у испису ће стајати: !!!! што значи да пакети пролазе везом. Уколико нешто није у реду, проверити да ли су коректне адресе уписане на одговарајуће интерфејсе, као и да ли топологија симулације одговара топологији из вежбе (водити рачуна о томе који су интерфејси међусобно повезани).

У овом тренутку су конфигурисане само адресе на рутерима, а не и протокол рутирања, тако да ако се на нпр. R1 проба: `ping 172.16.1.1`, то неће радити, јер рутер R1 нема у својој табели рутирања ову руту. Погледати садржај табела рутирања на рутерима са:

```
R1,2,3#show ip route
```

```
R1#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, FastEthernet0/0
C    192.168.13.0/24 is directly connected, FastEthernet0/1
     10.0.0.0/24 is subnetted, 3 subnets
C       10.0.2.0 is directly connected, Loopback2
C       10.0.3.0 is directly connected, Loopback3
C       10.0.1.0 is directly connected, Loopback1
```

Слика 7.16 Изглед табеле рутирања пре конфигурисања протокола рутирања

На слици 7.16 је приказан излаз ове команде за рутер 1. Види се да у табели рутирања постоји само 5 рута и то су две руте за мреже којима је овај рутер повезан са рутерима 2 и 3 и три руте ка *loopback* интерфејсима. Заједничко за све руте је да имају ознаку C, што означава *Connected*, односно да су то мреже које су директно повезане на рутер. На рутеру нема ни једне мреже која је повезана на друге рутере јер протокол рутирања није још увек активиран.

7.2.2. Конфигурација RIP протокола

Конфигурисати следеће:

```
R1#conf t
R1(config)#router rip                !režim za konfigurisanje RIP protokola
R1(config-router)#network 10.0.1.0   !specifikacija mreža koje se oglašavaju i
                                     !preko kojih se šalju
R1(config-router)#network 10.0.2.0   !RIP update-i
R1(config-router)#network 10.0.3.0
R1(config-router)#network 192.168.12.0
R1(config-router)#network 192.168.13.0
R1(config-router)#Ctrl-Z

R2#conf t
R2(config)#router rip
R2(config-router)#network 192.168.10.0
R2(config-router)#network 192.168.20.0
R2(config-router)#network 192.168.30.0
R2(config-router)#network 192.168.12.0
R2(config-router)#network 192.168.23.0
R2(config-router)#Ctrl-Z

R3#conf t
R3(config)#router rip
R3(config-router)#network 172.16.1.0
R3(config-router)#network 172.16.2.0
R3(config-router)#network 172.16.3.0
R3(config-router)#network 192.168.13.0
R3(config-router)#network 192.168.23.0
```

```
R3(config-router)#Ctrl-Z
```

Сачекати мало да се успостави проток рута између рутера и онда поново откуцати команду:
`R1,2,3#sh ip route`

```
R1#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, FastEthernet0/0
C    192.168.13.0/24 is directly connected, FastEthernet0/1
R    192.168.30.0/24 [120/1] via 192.168.12.2, 00:00:06, FastEthernet0/0
R    192.168.10.0/24 [120/1] via 192.168.12.2, 00:00:06, FastEthernet0/0
R    172.16.0.0/16 [120/1] via 192.168.13.3, 00:00:16, FastEthernet0/1
R    192.168.20.0/24 [120/1] via 192.168.12.2, 00:00:06, FastEthernet0/0
     10.0.0.0/24 is subnetted, 3 subnets
C       10.0.2.0 is directly connected, Loopback2
C       10.0.3.0 is directly connected, Loopback3
C       10.0.1.0 is directly connected, Loopback1
R    192.168.23.0/24 [120/1] via 192.168.13.3, 00:00:19, FastEthernet0/1
                     [120/1] via 192.168.12.2, 00:00:09, FastEthernet0/0
```

Слика 7.17 Изглед табеле рутирања након исправног конфигурисања протокола рутирања

На слици 7.17 је дат садржај таβеле рутирања у овом случају. Види се да су се појавиле руте са ознаком R што су руте добијене путем RIP протокола. У таβели рутирања су следеће руте добијене из RIP:

- 192.168.X0.0/24, што су *loopback* адресе рутера 2
- 172.16.0.0/16, што је агрегирана рута *loopback* адреса рутера 2[53]
- 192.168.23.0/24, што је рута ка мрежи која чини везу између рутера 2 и 3. Као што може да се види, ова рута указује на два различита интерфејса рутера 1 зато што је једнака метрика од рутера 1 до ове мреже преко рутера 2 и рутера 3. У овој ситуацији ће се вршити балансирање пакета ка 192.168.23.0/24 преко оба интерфејса.

7.2.3. Промена топологије мреже

Сада ће се административно угасити интерфејс на рутеру R1 између R1 и R3 чиме ће се променити топологија мреже. То ће се урадити овако:

```
R1#conf t
R1(config-if)#interface f0/1
R1(config-if)#shutdown
R1(config-if)#Ctrl-Z
R1#
```

53 RIP је тзв. *classful* протокол рутирања који шаље информације о рутама A, B и C класе, без слања маски. Стога у оваквим ситуацијама RIP шаље агрегирану руту класе B.

На рутеру 1 `show ip interface brief` и тиме проверити статус свих интерфејса на свим рутерима.

```
R1#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, FastEthernet0/0
R    192.168.13.0/24 [120/2] via 192.168.12.2, 00:00:01, FastEthernet0/0
R    192.168.30.0/24 [120/1] via 192.168.12.2, 00:00:01, FastEthernet0/0
R    192.168.10.0/24 [120/1] via 192.168.12.2, 00:00:01, FastEthernet0/0
R    172.16.0.0/16 [120/2] via 192.168.12.2, 00:00:01, FastEthernet0/0
R    192.168.20.0/24 [120/1] via 192.168.12.2, 00:00:01, FastEthernet0/0
     10.0.0.0/24 is subnetted, 3 subnets
C       10.0.2.0 is directly connected, Loopback2
C       10.0.3.0 is directly connected, Loopback3
C       10.0.1.0 is directly connected, Loopback1
R    192.168.23.0/24 [120/1] via 192.168.12.2, 00:00:03, FastEthernet0/0
```

Слика 7.18 Табела рутирања рутера 1 након промене топологије

Откуцати поново команду `R1#sh ip route`.

На слици 7.18 је приказан излаз табеле рутирања у овом случају. Број рута је остао исти, али је различито то што:

- Све руте воде преко интерфејса `FastEthernet0/0` јер је то једина расположива путања
- Рута ка мрежи 172.16.0.0/16 има метрику 2, јер од рутера до ове мреже треба проћи кроз рутере 2 и 3.
- Више нема две руте ка мрежи 192.168.23.0/24.

7.2.4. Финалне конфигурације рутера

У овом делу су дате потпуне конфигурације рутера које могу да се унесу *paste*-ом команди у конзолне прозоре након старта рада рутера у симулацији (подразумева се да је активан основни промпт). У *Putty* програму за *telnet* и SSH повезивање, *paste* се ради десним кликом у прозор програма. Овим ће рутери бити конфигурисани као на крају поглавља 7.2.2 ове вежбе. Пожељно је да се први пут конфигурације куцају поступно према претходном упутству, како би се схватили механизми конфигурисања уређаја, а конфигурације ниже служе као помоћ у каснијим фазама, када су студенти већ савладали основно конфигурисање уређаја, како би се брже извела конфигурација и као референца студентима. (обратити

пажњу на то да су све команде доле написане у скраћеном облику што Cisco CLI омогућава када су команде недвосмислене).

Рутер 1

```
enable
conf t
int f0/0
ip addr 192.168.12.1 255.255.255.0
no shut
int f0/1
ip addr 192.168.13.1 255.255.255.0
no shut
int lo1
ip addr 10.0.1.1 255.255.255.0
int lo2
ip addr 10.0.2.1 255.255.255.0
int lo3
ip addr 10.0.3.1 255.255.255.0
router rip
network 192.168.12.0
network 192.168.13.0
network 10.0.0.0
```

Рутер 2

```
enable
conf t
int f0/0
ip addr 192.168.12.2 255.255.255.0
no shut
int f0/1
ip addr 192.168.23.2 255.255.255.0
no shut
int lo1
ip addr 192.168.10.1 255.255.255.0
int lo2
ip addr 192.168.20.1 255.255.255.0
int lo3
ip addr 192.168.30.1 255.255.255.0
router rip
network 192.168.12.0
network 192.168.23.0
network 192.168.10.0
network 192.168.20.0
network 192.168.30.0
```

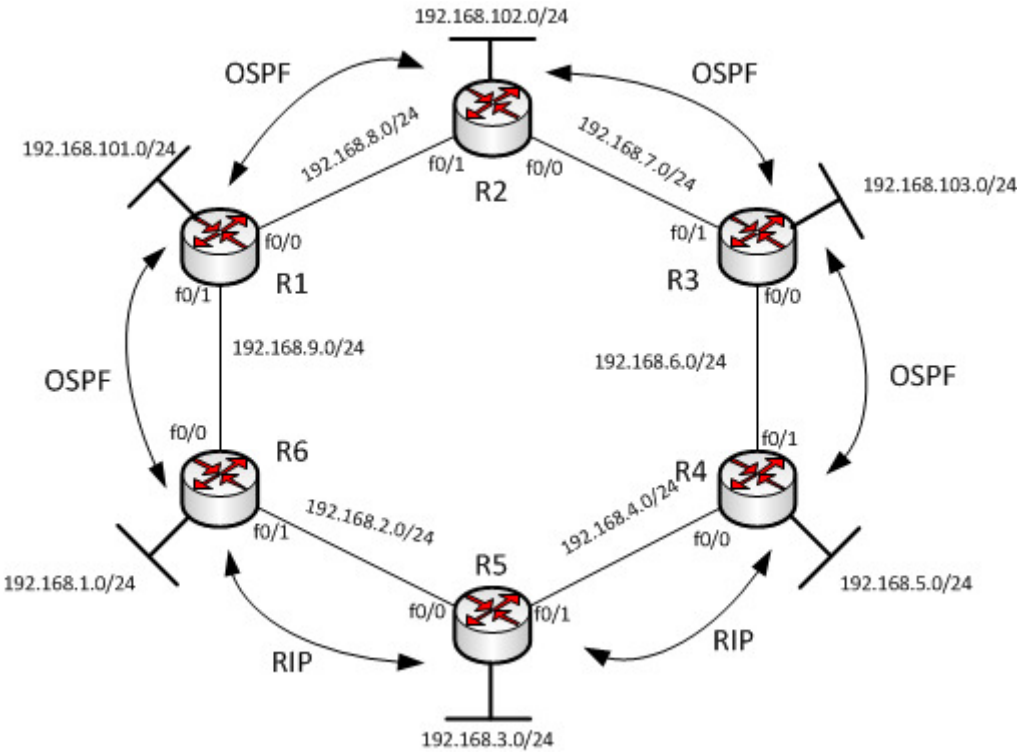
Рутер 3

```
enable
conf t
int f0/0
ip addr 192.168.13.3 255.255.255.0
no shut
int f0/1
ip addr 192.168.23.3 255.255.255.0
no shut
```

```
int lo1
ip addr 172.16.1.1 255.255.255.0
int lo2
ip addr 172.16.2.1 255.255.255.0
int lo3
ip addr 172.16.3.1 255.255.255.0
router rip
network 192.168.13.0
network 192.168.23.0
network 172.16.0.0
```

7.3. Редистрибуција рута[54]

Дата је рачунарска мрежа са слике 7.19. У мрежи се користе интерни протоколи рутирања између рутера како је назначено на слици, а на рутерима R4 и R6 се врши обострана редистрибуција рута (из RIP у OSPF и обрнуто). Сви рутери у интерне протоколе рутирања који су на њима конфигурисани оглашавају све мреже које су повезане на интерфејсе рутера (физички и *loopback* интерфејси).



Слика 7.19 Топологија мреже у примеру са редистрибуцијом рута

Одредити којим путањама ће ићи саобраћај:

- од рачунара на мрежи 192.168.5.0/24 ка рачунарима на мрежи 192.168.1.0/24
- од рачунара на мрежи 192.168.3.0/24 ка рачунарима на мрежи 192.168.101.0/24

54 Овај задатак је био на испиту из предмета 29.8.2014. и овде ће бити решен коришћењем симулатора

Решење овог проблема може да се уради анализом пропагације рута кроз мрежу уважавајући правила редистрибуције и правила убацивања рута на основу административних дистанци протокола рутирања. Међутим, овде ће решење бити реализовано у GNS3 симулатору. У GNS3 симулатору креирати и покренути одговарајућу топологију. Покренути све конзолне интерфејсе и конфигурисати рутере према задатом задатку, што је најлакше тако што ће се у конфигурационом режиму рада урадити *paste* следећих команди за одговарајуће рутере:

R1:
```
hostname R1
no ip domain lookup
!
interface Loopback0
 ip address 192.168.101.1 255.255.255.0
 ip ospf network point-to-point
!
interface FastEthernet0/0
 ip address 192.168.8.1 255.255.255.0
 no shut
!
interface FastEthernet0/1
 ip address 192.168.9.1 255.255.255.0
 no shut
!
router ospf 1
 log-adjacency-changes
 network 192.168.8.0 0.0.0.255 area 0
 network 192.168.9.0 0.0.0.255 area 0
 network 192.168.101.0 0.0.0.255 apea 0
!
end
```

R2:
```
hostname R2
no ip domain lookup
!
interface Loopback0
 ip address 192.168.102.2 255.255.255.0
 ip ospf network point-to-point
!
interface FastEthernet0/0
 ip address 192.168.8.2 255.255.255.0
 no shut
!
interface FastEthernet0/1
 ip address 192.168.7.2 255.255.255.0
 no shut
!
router ospf 1
 network 192.168.7.0 0.0.0.255 area 0
 network 192.168.8.0 0.0.0.255 area 0
 network 192.168.102.0 0.0.0.255 area 0
!
end
```

R3:
```
hostname R3
!
```

```
no ip domain lookup
!
interface Loopback0
 ip address 192.168.103.3 255.255.255.0
 ip ospf network point-to-point
!
interface FastEthernet0/0
 ip address 192.168.7.3 255.255.255.0
 no shut
!
interface FastEthernet0/1
 ip address 192.168.6.3 255.255.255.0
 no shut
!
router ospf 1
 log-adjacency-changes
 network 192.168.6.0 0.0.0.255 area 0
 network 192.168.7.0 0.0.0.255 area 0
 network 192.168.103.0 0.0.0.255 area 0
!
end
```

R4:

```
hostname R4
no ip domain lookup
!
interface Loopback0
 ip address 192.168.5.4 255.255.255.0
 ip ospf network point-to-point
!
interface FastEthernet0/0
 ip address 192.168.6.4 255.255.255.0
 no shut
!
interface FastEthernet0/1
 ip address 192.168.4.4 255.255.255.0
 no shut
!
router ospf 1
 log-adjacency-changes
 redistribute rip
 network 192.168.4.0 0.0.0.255 area 0
 network 192.168.5.0 0.0.0.255 area 0
 network 192.168.6.0 0.0.0.255 area 0
!
router rip
 redistribute ospf 1 metric 3    !3 oznacava metriku redistribuiranih ruta
 network 192.168.4.0
 network 192.168.5.0
 network 192.168.6.0
!
end
```

R5:

```
hostname R5
no ip domain lookup
!
interface Loopback0
 ip address 192.168.3.5 255.255.255.0
!
```

```
interface FastEthernet0/0
 ip address 192.168.4.5 255.255.255.0
 no shut
!
interface FastEthernet0/1
 ip address 192.168.2.5 255.255.255.0
 no shut
!
router rip
 network 192.168.2.0
 network 192.168.3.0
 network 192.168.4.0
!
!
end
```

R6:

```
hostname R6
no ip domain lookup
!
interface Loopback0
 ip address 192.168.1.6 255.255.255.0
 ip ospf network point-to-point
!
interface FastEthernet0/0
 ip address 192.168.2.6 255.255.255.0
 no shut
!
interface FastEthernet0/1
 ip address 192.168.9.6 255.255.255.0
 no shut
!
router ospf 1
 log-adjacency-changes
 redistribute rip
 network 192.168.2.0 0.0.0.255 area 0
 network 192.168.9.0 0.0.0.255 area 0
 network 192.168.1.0 0.0.0.255 area 0
!
router rip
 redistribute ospf 1 metric 3
 network 192.168.1.0
 network 192.168.2.0
 network 192.168.9.0
!
end
```

Обратити пажњу на црвеном бојом означене команде на рутерима R4 и R6 којима се реализује редистрибуција RIP рута у OSPF и обрнуто. Када се врши редистрибуција OSPF рута у RIP, потребно је тачно специфицирати метрику редистрибуираним рутама (у овом случају на вредност 3), зато што би у супротном рутер сваку метрику другу вредност која је већа од 15 протумачио као недоступну руту, према начину рада RIP протокола.

Након што су конфигурације унете, потребно је мало сачекати да се све везе подигну и успоставе протоколи рутирања. Након тога анализирати табеле рутирања. Овде ће бити показане табеле рутирања на рутерима 3, 4 и 5.

```
R3#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

O    192.168.8.0/24 [110/20] via 192.168.7.2, 00:02:08, FastEthernet0/0
O    192.168.9.0/24 [110/30] via 192.168.7.2, 00:02:08, FastEthernet0/0
O    192.168.4.0/24 [110/20] via 192.168.6.4, 00:02:08, FastEthernet0/1
O    192.168.5.0/24 [110/11] via 192.168.6.4, 00:02:08, FastEthernet0/1
C    192.168.6.0/24 is directly connected, FastEthernet0/1
C    192.168.7.0/24 is directly connected, FastEthernet0/0
O    192.168.102.0/24 [110/11] via 192.168.7.2, 00:02:08, FastEthernet0/0
O    192.168.1.0/24 [110/31] via 192.168.7.2, 00:02:10, FastEthernet0/0
C    192.168.103.0/24 is directly connected, Loopback0
O    192.168.2.0/24 [110/40] via 192.168.7.2, 00:02:10, FastEthernet0/0
O E2 192.168.3.0/24 [110/20] via 192.168.7.2, 00:02:10, FastEthernet0/0
O    192.168.101.0/24 [110/21] via 192.168.7.2, 00:02:10, FastEthernet0/0
```

Слика 7.20 Изглед табеле рутирања на рутеру 3

```
R5#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

R    192.168.8.0/24 [120/3] via 192.168.4.4, 00:00:27, FastEthernet0/0
                    [120/3] via 192.168.2.6, 00:00:00, FastEthernet0/1
R    192.168.9.0/24 [120/1] via 192.168.2.6, 00:00:00, FastEthernet0/1
C    192.168.4.0/24 is directly connected, FastEthernet0/0
R    192.168.5.0/24 [120/1] via 192.168.4.4, 00:00:27, FastEthernet0/0
R    192.168.6.0/24 [120/1] via 192.168.4.4, 00:00:27, FastEthernet0/0
R    192.168.7.0/24 [120/3] via 192.168.4.4, 00:00:27, FastEthernet0/0
                    [120/3] via 192.168.2.6, 00:00:02, FastEthernet0/1
R    192.168.102.0/24 [120/3] via 192.168.4.4, 00:00:29, FastEthernet0/0
                      [120/3] via 192.168.2.6, 00:00:02, FastEthernet0/1
R    192.168.1.0/24 [120/1] via 192.168.2.6, 00:00:02, FastEthernet0/1
R    192.168.103.0/24 [120/3] via 192.168.4.4, 00:00:29, FastEthernet0/0
                      [120/3] via 192.168.2.6, 00:00:02, FastEthernet0/1
C    192.168.2.0/24 is directly connected, FastEthernet0/1
C    192.168.3.0/24 is directly connected, Loopback0
R    192.168.101.0/24 [120/3] via 192.168.4.4, 00:00:31, FastEthernet0/0
                      [120/3] via 192.168.2.6, 00:00:04, FastEthernet0/1
```

Слика 7.21 Изглед табеле рутирања на рутеру 5

Рутер 3 се у потпуности налази у OSPF делу мреже, те су све руте на овом рутеру типа O – OSPF, осим директно повезаних (Слика 7.20). Од ових рута, неке су типа O E2, што је ознака за екстерне OSPF руте тип 2, односно оне руте које су добијене редистрибуцијом и имају фиксну метрику 20. Таква је рута ка мрежи 192.168.3.0/24 која се налази повезана на RIP рутер 5. Такође обратити на то да је вредност административне дистанце OSPF протокола 110.

Рутер 5 је у потпуности у RIP делу мреже, па су све руте на овом рутеру типа R – RIP (слика 7.21). Руте из OSPF дела мреже, као на пример рута 192.168.102.0/24 се на рутеру 5 виде као RIP руте јер су их редистрибуирали рутери R4 или R6. Пошто оба рутера, и R4 и R6 када редистрибуирају OSPF руте шаљу их са метриком 3, ова рута ка 192.168.102.0/24 има две једнаке путање. Док је за ову руту то интуитивно и очекивано јер су од рутера 5 две путање до ове мреже тополошки једнако, може да се види да је слично и за друге руте из OSPF дела мреже попут 192.168.101.0/24 и 192.168.103.0/24, иако је јасно да тополошки гледано путање у овим случајевима нису једнаке. То је последица административног додељивања метрике од стране рутера који врше редистрибуцију. Такође обратити на то да је вредност административне дистанце RIP протокола 120.

```
R4#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

O    192.168.8.0/24 [110/30] via 192.168.6.3, 00:00:01, FastEthernet0/0
O    192.168.9.0/24 [110/40] via 192.168.6.3, 00:00:01, FastEthernet0/0
C    192.168.4.0/24 is directly connected, FastEthernet0/1
C    192.168.5.0/24 is directly connected, Loopback0
C    192.168.6.0/24 is directly connected, FastEthernet0/0
O    192.168.7.0/24 [110/20] via 192.168.6.3, 00:00:01, FastEthernet0/0
O    192.168.102.0/24 [110/21] via 192.168.6.3, 00:00:01, FastEthernet0/0
O    192.168.1.0/24 [110/41] via 192.168.6.3, 00:00:03, FastEthernet0/0
O    192.168.103.0/24 [110/11] via 192.168.6.3, 00:00:03, FastEthernet0/0
O    192.168.2.0/24 [110/50] via 192.168.6.3, 00:00:03, FastEthernet0/0
O E2 192.168.3.0/24 [110/20] via 192.168.6.3, 00:00:03, FastEthernet0/0
O    192.168.101.0/24 [110/31] via 192.168.6.3, 00:00:03, FastEthernet0/0
```

Слика 7.22 Изглед табеле рутирања на рутеру 4

Рутер 4 се налази на граници између два домена – са једне стране преко интерфејса `f0/0` је доступан OSPF део мреже, а преко интерфејса `f0/1` је доступан RIP део мреже. Стога би у табели рутирања овог рутера требало да постоје руте оба типа (Слика 7.22) јер рутери који врше редистрибуцију задржавају типове рута које су добили од својих суседа (не мењају им порекло према правилима редистрибуције), већ редистрибуцију врше у излазном смеру, када оглашавају руте суседима. Међутим, интересантан је случај руте 192.168.3.0/24 која се у

таβели рутирања овог рутера налази као екстерна OSPF рута упркос томе што би интуитивно било очекивано да је ово RIP рута јер је доβијена од директног суседа - рутера 5 путем RIP протокола. Ово је последица редистриβуције исте руте која се врши и на рутеру 6. Зβог ове редистриβуције рута 192.168.3.0/24 долази до рутера 4 са две различите стране: од рутера 5 путем RIP протокола и путем OSPF као рута редистриβуирана на рутеру 6, преко рутера 1, 2 и 3. Пошто је административна дистанца OSPF протокола (110) нижа од административне дистанце RIP протокола (120), рутер 4 у таβелу рутирања уβацити ону руту која долази из протокола са нижом административном дистанцом, односно у овом случају OSPF руту и доβиће се неоптимална путања која води преко рутера 3, 2, 1, 6 и 5, уместо директно везом R4-R5.

Да би се експериментално одредило куда иду пакети у две ситуације наведене на почетку поглавља, потребно је покренути команду `traceroute` али у режиму рада који дозвољава подешавање изворишне адресе. Уколико се укуца само `traceroute aaa.bbb.ccc.ddd`↵ рутер ће као изворишну адресу да стави адресу оног интерфејса који је у смислу мрежне топологије (табеле рутирања) најближи датој дестинацији (адресу излазног интерфејса). Да би се подесила изворишна адреса укуцава се само `traceroute` и онда се улази у интерактивни режим рада у којем је могуће подесити изворишну адресу. Ово изгледа овако:

```
R4#traceroute ↵
Protocol [ip]: ↵
Target IP address: 192.168.1.6↵     (дестинациона адреса)
Source address: 192.168.5.4 ↵       (изворишна адреса)
Numeric display [n]: ↵
Timeout in seconds [3]: ↵
Probe count [3]: ↵
Minimum Time to Live [1]: ↵
Maximum Time to Live [30]: ↵
Port Number [33434]: ↵
Loose, Strict, Record, Timestamp, Verbose[none]: ↵
Type escape sequence to abort.
Tracing the route to 192.168.1.6
...
```

(ако се извршавање `traceroute` команде из неког разлога заглави, излазна секвенца је *Ctrl-Alt-6*).

Излаз ове команде за први случај је приказан на слици 7.23. Види се да је најβоља путања од 192.168.5.4 ка 192.168.1.6 преко рутера 3, 2 и 1, иако је тополошки βлижа путања преко рутера 5. Ово је и очекивано ако се погледа таβела рутирања на рутеру 4, која за мрежу 192.168.1.6/24 (адреса *loopback* интерфејса) има екстерну OSPF руту која указује на излазни интерфејс f0/0, односно интерфејс према рутеру 3. Разлог за ово је сличан као у претходном примеру – редистриβуција која се врши на рутеру R6 и то што OSPF протокол има нижу административну дистанцу од RIP протокола.

```
R4#trace
Protocol [ip]:
Target IP address: 192.168.1.6
Source address: 192.168.5.4
Numeric display [n]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Type escape sequence to abort.
Tracing the route to 192.168.1.6

  1 192.168.6.3 52 msec 44 msec 44 msec
  2 192.168.7.2 72 msec 88 msec 72 msec
  3 192.168.8.1 92 msec 92 msec 68 msec
  4 192.168.9.6 112 msec 96 msec 92 msec
```

Слика 7.23 Излаз команде `traceroute` на рутеру 4

Слично може да се уради и за други пример када се посматра путања пакета од рутера 5 ка мрежи 192.168.101.0/24. Како је и очекивано с обзиром на то да постоје две руте на рутеру 5 које указују ка мрежи 192.168.101.0/24, рутер ће вршити балансирање саобраћаја према овој дестинацији тако што ће слати пакете преко рутера 4 и 6 наизменично (мреже 192.168.2.0/24 и 192.168.4.0/24) иако ове путање нису стварно једнаке (Слика 7.24).

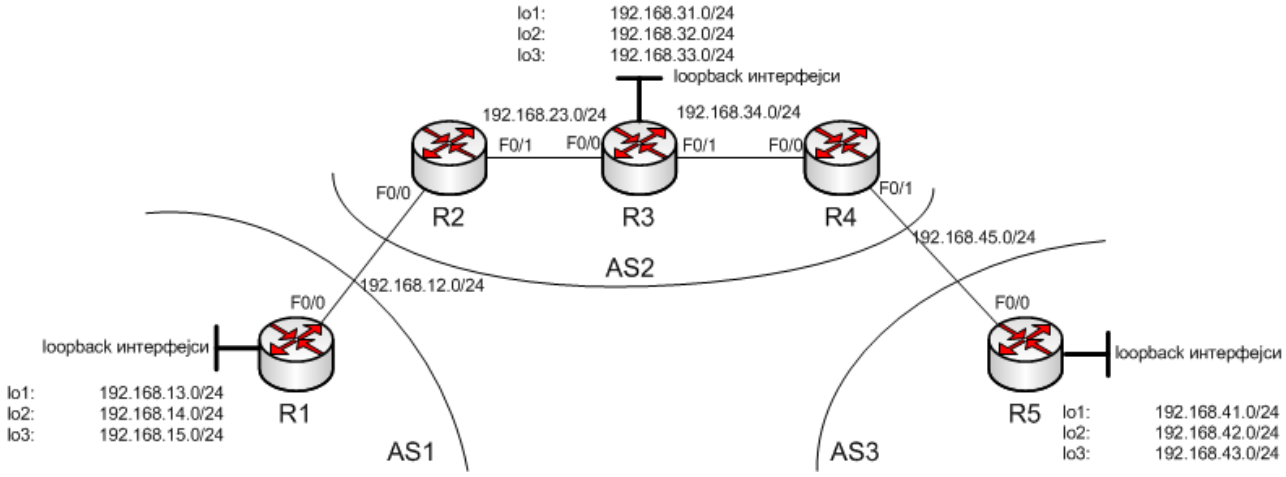
```
R5#trace
Protocol [ip]:
Target IP address: 192.168.101.1
Source address: 192.168.3.5
Numeric display [n]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Type escape sequence to abort.
Tracing the route to 192.168.101.1

  1 192.168.4.4 92 msec
    192.168.2.6 60 msec
    192.168.4.4 92 msec
  2 192.168.9.1 148 msec
    192.168.6.3 160 msec
    192.168.9.1 104 msec
```

Слика 7.24 Излаз команде `traceroute` на рутеру 5

Све ово је последица двоструке редистрибуције која се врши на рутерима 4 и 6. Ако би се редистрибуција искључила на једном од ових рутера, путање између OSPF и RIP делова мреже би биле јасно одређене јединим могућим путем где се врши размена рута између

протокола рутирања. Студентима се препоручује даураде ово на рутеру 6 и проанализирају таβеле рутирања и путање пакета у ситуацији са једном редистриβуцијом.

7.4. BGP протокол



Слика 7.25 Тоūолоūија мреже у ūримеру конфиūурисања BGP ūроūокола

У алату GNS3 креирати топологију као на слици 7.25 која треβа да симулира три повезана аутономна система, повезати рутере, укључити их и стартовати конзолни приступ. Након тога ће се конфигурисати основна конфигурација рутера (адресе интерфејса и интерни протокол рутирања унутар аутономног система 2) према следећим конфигурацијама:

Рутер R1

```
enable
conf t
int f0/0
ip addr 192.168.12.1 255.255.255.0
no shut
int lo1
ip addr 192.168.13.1 255.255.255.0
int lo2
ip addr 192.168.14.1 255.255.255.0
int lo3
ip addr 192.168.15.1 255.255.255.0
```

Рутер R2

```
enable
conf t
int f0/0
ip addr 192.168.12.2 255.255.255.0
no shut
int f0/1
ip addr 192.168.23.2 255.255.255.0
no shut
router rip
```

```
network 192.168.23.0
```

Рутер R3

```
enable
conf t
int f0/0
ip addr 192.168.23.3 255.255.255.0
no shut
int f0/1
ip addr 192.168.34.3 255.255.255.0
no shut
int lo1
ip addr 192.168.31.1 255.255.255.0
int lo2
ip addr 192.168.32.1 255.255.255.0
int lo3
ip addr 192.168.33.1 255.255.255.0
router rip
network 192.168.23.0
network 192.168.34.0
network 192.168.31.0
network 192.168.32.0
network 192.168.33.0
```

Рутер R4

```
enable
conf t
int f0/0
ip addr 192.168.34.4 255.255.255.0
no shut
int f0/1
ip addr 192.168.45.4 255.255.255.0
no shut
router rip
network 192.168.34.0
```

Рутер R5

```
enable
conf t
int f0/0
ip addr 192.168.45.5 255.255.255.0
no shut
int lo1
ip addr 192.168.41.1 255.255.255.0
int lo2
ip addr 192.168.42.1 255.255.255.0
int lo3
ip addr 192.168.43.1 255.255.255.0
```

Loopback адресе симулирају мреже које аутономни систем оглашава другим аутономним системима. Након ових конфигурација биће могућа само комуникација унутар аутономног система 2, где је успостављен интерни протокол рутирања и то само на мрежама (интерфејсима) који су потпуно унутар тог аутономномног система. Проверити ово тако што се на рутеру 4 изврше следеће команде:

```
ping 192.168.23.2
ping 192.168.31.1
```

Ако је све исправно конфигурисано, добиће се информација да су пакети прошли мрежом.

7.4.1. Конфигурација BGP протокола

Следећи корак је конфигурација BGP протокола између аутономних система (екстерни BGP). То се ради следећим командама:

Рутер R1

```
router bgp 1
neighbor 192.168.12.2 remote-as 2
network 192.168.13.0 mask 255.255.255.0
network 192.168.14.0 mask 255.255.255.0
network 192.168.15.0 mask 255.255.255.0
```

конфигурација се састоји од 3 кључна дела:

- спецификација протокола рутирања са бројем аутономног система у којем је рутер 1 (`router` команда)
- спецификација суседа (`neighbor` команда) са бројем његовог аутономног система (2). Да би се успоставила BGP сесија бројеви аутономних система и IP адресе суседа морају да се поклопе. У супротном се неће успоставити BGP сесија, већ ће се размењити *Notification* поруке којима се сигнализира проблем у конфигурацији.
- декларација мрежа које ће бити оглашаване ка другим аутономним системима (`network` команде)

Након овога статус конфигурисане BGP сесије према аутономном систему 2 ће бити *Active* (Слика 7.26) зато што још увек није конфигурисан BGP на рутеру 2, те не може да се успостави ни TCP сесија са суседом.

```
R1#sh ip bgp summary
BGP router identifier 192.168.15.1, local AS number 1
BGP table version is 4, main routing table version 4
3 network entries using 351 bytes of memory
3 path entries using 156 bytes of memory
2/1 BGP path/bestpath attribute entries using 248 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 755 total bytes of memory
BGP activity 3/0 prefixes, 3/0 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down State/PfxRcd
192.168.12.2    4     2       0       0        0    0    0 never    Active
```

Слика 7.26 Статус конфигурисаних BGP сесија

Даље ће бити конфигурисане eBGP сесије на осталим рутерима према плану адресирања:

Рутер R2

```
router bgp 2
neighbor 192.168.12.1 remote-as 1
network 192.168.31.0 mask 255.255.255.0
network 192.168.32.0 mask 255.255.255.0
network 192.168.33.0 mask 255.255.255.0
```

Рутер R4

```
router bgp 2
neighbor 192.168.45.5 remote-as 3
network 192.168.31.0 mask 255.255.255.0
network 192.168.32.0 mask 255.255.255.0
network 192.168.33.0 mask 255.255.255.0
```

Рутер R5

```
router bgp 3
neighbor 192.168.45.4 remote-as 2
network 192.168.41.0 mask 255.255.255.0
network 192.168.42.0 mask 255.255.255.0
network 192.168.43.0 mask 255.255.255.0
```

```
R1#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, FastEthernet0/0
C    192.168.13.0/24 is directly connected, Loopback1
C    192.168.14.0/24 is directly connected, Loopback2
B    192.168.31.0/24 [20/1] via 192.168.12.2, 00:00:11
C    192.168.15.0/24 is directly connected, Loopback3
B    192.168.32.0/24 [20/1] via 192.168.12.2, 00:00:11
B    192.168.33.0/24 [20/1] via 192.168.12.2, 00:00:11

R1#sh ip bgp sum
BGP router identifier 192.168.15.1, local AS number 1
BGP table version is 7, main routing table version 7
6 network entries using 702 bytes of memory
6 path entries using 312 bytes of memory
3/2 BGP path/bestpath attribute entries using 372 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1410 total bytes of memory
BGP activity 6/0 prefixes, 6/0 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/DownState/PfxRcd
192.168.12.2    4     2       6       6        7    0    0 00:01:34        3
```

Слика 7.27 Изглед табеле рутирања и статус BGP сесија на рутеру 1

Сваки пут приликом промене конфигурације BGP протокола потребно је сачекати двадесетак секунди да се успоставе BGP сесије. Слика 7.27 показује стање на рутеру 1: добијене су три руте путем BGP протокола (ознака B), све три од суседа из аутономног система 2, а види се и да статус BGP сесије више није *Active*, већ да су добијене 3 руте (префикса). Сличан резултат се добија и на рутерима 2, 4 и 5. Проверити ово.

Као што може да се види, аутономни систем 1 није добио руте из аутономног система 3, а разлог за то је тај што нема успостављених iBGP сесија унутар аутономног система 2.

Успостављање iBGP сесија ће бити сада урађено на следећи начин:
Рутер R2

```
router bgp 2
neighbor 192.168.34.4 remote-as 2
```

Рутер R4

```
router bgp 2
neighbor 192.168.23.2 remote-as 2
```

Овим је успостављена iBGP сесија између рутера 2 и 4. Ако би се погледао садржај табеле рутирања на рутеру 1, он би био исти као не слици 7.27, односно у њој и даље не би било рута из аутономног система 3. Да би се разумело зашто је ово овако, биће приказане табеле рутирања (слика 7.28) и BGP табела (слика 7.30) на рутеру 2.

```
R2#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, FastEthernet0/0
B    192.168.13.0/24 [20/0] via 192.168.12.1, 00:14:07
B    192.168.14.0/24 [20/0] via 192.168.12.1, 00:14:07
R    192.168.31.0/24 [120/1] via 192.168.23.3, 00:00:19, FastEthernet0/1
B    192.168.15.0/24 [20/0] via 192.168.12.1, 00:14:07
C    192.168.23.0/24 is directly connected, FastEthernet0/1
R    192.168.34.0/24 [120/1] via 192.168.23.3, 00:00:19, FastEthernet0/1
R    192.168.32.0/24 [120/1] via 192.168.23.3, 00:00:19, FastEthernet0/1
R    192.168.33.0/24 [120/1] via 192.168.23.3, 00:00:21, FastEthernet0/1
```

Слика 7.28 Табела рутирања рутера 2

Као што може да се види, иако је коректно успостављена iBGP сесија између рутера 2 и 4 унутар аутономног система 2: слика 7.29 показује да је од суседа 192.168.34.4, што је рутер 4 добијено укупно 6 рута, у табели рутирања рутера 2 нема рута из аутономног система 3. То исто показује и BGP табела (слика 7.30), у којој се види да су путем BGP протокола добијене руте из аутономног система 3, али да нису одабране као најбоље (уз њих не стоји ознака >,

што означава изабрану најбољу руту). Разлог за то је тај што рутер 2 нема у својој табели рутирања руту ка мрежи 192.168.45.0/24, што је мрежа на којој је *Next Hop* ових рута (слика 7.28).

```
R2#sh ip bgp summa
...

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
192.168.12.1    4     1      19      20        7    0    0 00:15:59        3
192.168.34.4    4     2      13      13        7    0    0 00:07:14        6
```

Слика 7.29 Сумарни статус BGP сесија рутера 2

Да би се овај проблем решио биће додате и ове мреже, између аутономних система у RIP протокол на следећи начин:

Рутер R2

```
router rip
network 192.168.12.0
```

Рутер R4

```
router rip
network 192.168.45.0
```

```
R2#sh ip bgp
BGP table version is 7, local router ID is 192.168.23.2
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.13.0     192.168.12.1             0             0 1 i
*> 192.168.14.0     192.168.12.1             0             0 1 i
*> 192.168.15.0     192.168.12.1             0             0 1 i
* i192.168.31.0     192.168.34.3             1    100      0 i
*>                  192.168.23.3             1         32768 i
* i192.168.32.0     192.168.34.3             1    100      0 i
*>                  192.168.23.3             1         32768 i
* i192.168.33.0     192.168.34.3             1    100      0 i
*>                  192.168.23.3             1         32768 i
* i192.168.41.0     192.168.45.5             0    100      0 3 i
* i192.168.42.0     192.168.45.5             0    100      0 3 i
* i192.168.43.0     192.168.45.5             0    100      0 3 i
```

Слика 7.30 BGP табела рутера 2

Након ове конфигурације, аутономни систем 2 ће регуларно пренети руте између аутономних система 1 и 3. Ово може да се види понављањем претходних команди, а табела рутирања на рутеру 1 ће изгледати као на слици 7.31.

```
R1#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, FastEthernet0/0
C    192.168.13.0/24 is directly connected, Loopback1
C    192.168.14.0/24 is directly connected, Loopback2
B    192.168.31.0/24 [20/1] via 192.168.12.2, 00:20:04
C    192.168.15.0/24 is directly connected, Loopback3
B    192.168.42.0/24 [20/0] via 192.168.12.2, 00:00:08
B    192.168.43.0/24 [20/0] via 192.168.12.2, 00:00:08
B    192.168.41.0/24 [20/0] via 192.168.12.2, 00:00:08
B    192.168.32.0/24 [20/1] via 192.168.12.2, 00:20:04
B    192.168.33.0/24 [20/1] via 192.168.12.2, 00:20:06
```

Слика 7.31 Табела рутирања рутера 1

Међутим, упркос овоме, ако би се покушала комуникација између мрежа повезаних на рутере 1 и 5, пакети не би пролазили (слика 7.32).

```
R1#ping
Protocol [ip]:
Target IP address: 192.168.41.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 192.168.13.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.41.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.13.1
.....
Success rate is 0 percent (0/5)
```

Слика 7.32 Неуспешан пролазак пакета између мрежа на рутерима 1 и 5

Разлог за ово је нерегуларно рутирање унутар аутономног система 2: рутер R3 нема руте ка мрежама из аутономних система 1 и 3, јер их још увек није добио путем било ког протокола. Табела рутирања на рутеру 3 у овој ситуацији је приказана на слици 7.33. На слици се види да рутер R3 има само руте из аутономног система 2. Стога ће пакети када се шаљу из аутономног система 1 бити регуларно прослеђени рутеру R2. Он ће их проследити даље ка

рутеру 3, али пошто овај рутер нема руту ка мрежама изван свог аутономног система, он ће те пакете одбацити.

```
R3#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

R    192.168.12.0/24 [120/1] via 192.168.23.2, 00:00:08, FastEthernet0/0
C    192.168.31.0/24 is directly connected, Loopback1
R    192.168.45.0/24 [120/1] via 192.168.34.4, 00:00:11, FastEthernet0/1
C    192.168.23.0/24 is directly connected, FastEthernet0/0
C    192.168.34.0/24 is directly connected, FastEthernet0/1
C    192.168.32.0/24 is directly connected, Loopback2
C    192.168.33.0/24 is directly connected, Loopback3
```

Слика 7.33 Табела рутирања на рутеру 3

Да би се ова нерегуларна ситуација исправила, потребно је да се направи потпун граф iBGP сесија унутар аутономног система 2. Ово ће бити урађено додавањем iBGP сесија према рутеру 3 на следећи начин:

Рутер R2

```
router bgp 2
neighbor 192.168.23.3 remote-as 2
```

Рутер R3

```
router bgp 2
neighbor 192.168.23.2 remote-as 2
neighbor 192.168.34.4 remote-as 2
```

Рутер R4

```
router bgp 2
neighbor 192.168.34.3 remote-as 2
```

Сада је рутирање у ова три аутономна система потпуно регуларно и сви пакети пролазе без проблема, што може да се види на слици 7.34.

Проверити табелу рутирања на рутеру 2 и уочити iBGP и eBGP руте и вредности њихових административних дистанци – која руте имају дистанцу 20, а које 200?

```
R1#ping
Protocol [ip]:
Target IP address: 192.168.41.1
...
Source address or interface: 192.168.13.1
…
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.41.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.13.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/151/180 ms
```

Слика 7.34 Успешан пролазак пакета између мрежа на рутерима 1 и 5 кроз аутономни систем 2

7.4.2.Промене атрибута BGP рута

У овом поглављу ће бити показане основне манипулације BGP рутама и њиховим атрибутима које се користе за организацију рутирања.

7.4.2.1. Постављање Local Preference атрибута

Local Preference атрибут се поставља на улазу у аутономни систем и преноси се интерним BGP протоколом. Следећим командама ће бити постављен *Local Preference* на вредност 200 за руту 192.168.13.0/24 приликом њеног уласка у аутономни систем 2.

```
R2(config)#router bgp 2
R2(config-router)#neighbor 192.168.12.1 route-map as2-in in !postavljanje
                                                            !ulaznog filtra
R2(config-router)#exit                          !filter as2-in je postavljen za
                                                !dolazne (in) rute
R2(config)#route-map as2-in permit 10           !definicija filtra
R2(config-route-map)#match ip address 10        !za adrese definisane listom 10
R2(config-route-map)#set local-preference 200    !podesi LP=200
R2(config-route-map)#exit
R2(config)#route-map as2-in permit 50             !za sve ostale rute
R2(config-route-map)#set local-preference 100     !podesi LP=100 (default)
R2(config-route-map)#exit
R2(config)#access-list 10 permit 192.168.13.0 !definicija liste 10
R2(config)#Ctrl-Z
```

Да би ове промене имале ефекта мора да се изврши ресет BGP сесије ка аутономном систему 1 следећом командом:

```
R2#clear ip bgp 192.168.12.1                !restart BGP sesije ka AS2
```

```
R2#sh ip bgp
BGP table version is 16, local router ID is 192.168.23.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.13.0     192.168.12.1             0    200      0 1 i
*> 192.168.14.0     192.168.12.1             0    100      0 1 i
*> 192.168.15.0     192.168.12.1             0    100      0 1 i
* i192.168.31.0     192.168.34.3             1    100      0 i
*>                  192.168.23.3             1         32768 i
* i192.168.32.0     192.168.34.3             1    100      0 i
*>                  192.168.23.3             1         32768 i
* i192.168.33.0     192.168.34.3             1    100      0 i
*>                  192.168.23.3             1         32768 i
*>i192.168.41.0     192.168.45.5             0    100      0 3 i
*>i192.168.42.0     192.168.45.5             0    100      0 3 i
*>i192.168.43.0     192.168.45.5             0    100      0 3 i
```

Слика 7.35 Атрибут Local Preference постављен на рутеру 2

На слици 7.35 се види да је у BGP табели рутера 2 рута ка мрежи 192.168.13.0/24 добила постављену вредност *Local Preference* атрибута. Такође, ако се погледа BGP табела на рутерима 4 и 5 (слика 7.36), видеће се да је на рутеру 4 иста вредност овог атрибута као на рутеру 2, док је на рутеру 5 остала *default* вредност зато што се *Local Preference* атрибут не преноси путем екстерног BGP, већ је локална за један аутономни систем.

```
R4#sh ip bgp
BGP table version is 16, local router ID is 192.168.45.4
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i192.168.13.0     192.168.12.1             0    200      0 1 i

R5#sh ip bgp
BGP table version is 16, local router ID is 192.168.43.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.13.0     192.168.45.4                           0 2 1 i
```

Слика 7.36 Атрибути руте 192.168.13.0 на рутерима 4 и 5

Слика 7.37 показује неке могућности које пружа команда `set` приликом дефиниције филтера рута. Може да се види да се на исти начин на који је у претходном примеру додељена вредност *Local Preference* атрибуту могуће: урадити *AS-Path preprending*, променити вредности атрибута *Community*, MED (*metric*), *Origin* и *Weight*.

```
R2(config-route-map)#set ?
  as-path            Prepend string for a BGP AS-path attribute
  community          BGP community attribute
  dampening          Set BGP route flap dampening parameters
  local-preference   BGP local preference path attribute
  metric             Metric value for destination routing protocol
  origin             BGP origin code
  weight             BGP weight for routing table
```

Слика 7.37 Моīућности за īостављање BGP атрибута

7.4.2.2. Промена AS-Path атрибута

На рутеру 2 ће се модификовати филтер тако да се за руту 192.168.14.0/24 у AS-Path додају две додатне вредности аутономног система 1, из ког потиче ова рута када улазе у аутономни систем 2. То ће бити изведено следећим командама:

```
route-map as2-in permit 20
match ip address 20
set as-path prepend 1 1              ! dodaj dve oznake AS 1

access-list 20 permit 192.168.14.0
```

И на крају рестартом BGP сесије ка суседу помоћу:

```
R2#clear ip bgp 192.168.12.1
```

Ово ће довести до промене AS-Path атрибута за руту 192.168.14.0 на свим рутерима, а што је показано за рутере 2, 4 и 5 у скраћеним исписима BGP табела на слици 7.38. Као што може да се види једном унета промена *AS-Path* ће пропагирати до свих наредних аутнономних система.

```
R2#sh ip bgp
...
   Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.13.0     192.168.12.1             0    200      0 1 i
*> 192.168.14.0     192.168.12.1             0             0 1 1 1 i
*> 192.168.15.0     192.168.12.1             0    100      0 1 I

R4#sh ip bgp
...
   Network          Next Hop            Metric LocPrf Weight Path
*>i192.168.13.0     192.168.12.1             0    200      0 1 i
*>i192.168.14.0     192.168.12.1             0    100      0 1 1 1 i
*>i192.168.15.0     192.168.12.1             0    100      0 1 I

R5#sh ip bgp
...
   Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.13.0     192.168.45.4                           0 2 1 i
*> 192.168.14.0     192.168.45.4                           0 2 1 1 1 i
*> 192.168.15.0     192.168.45.4                           0 2 1 i
```

Слика 7.38 Атрибути руте 192.168.14.0 на рутерима 2, 4 и 5

Додатно за ту исту руту ће бити додате 3 ознаке аутономног система 2 када се оглашава ка аутономном систему 3. Ово ће бити изведено на рутеру 4 следећим командама:

```
router bgp 2
neighbor 192.168.45.5 route-map as2-out out
route-map as2-out permit 10
match ip address 20
set as-path prepend 2 2 2
route-map as2-out permit 20

access-list 20 permit 192.168.14.0
```

Уз обавезан рестарт BGP сесије ка суседу у аутономном систему 3 на крају: `clear ip bgp 192.168.45.5`

```
R5#sh ip bgp
BGP table version is 42, local router ID is 192.168.43.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.13.0     192.168.45.4                           0 2 1 i
*> 192.168.14.0     192.168.45.4                           0 2 2 2 2 1 1 1 i
*> 192.168.15.0     192.168.45.4                           0 2 1 i
```

Слика 7.39 Атрибути руте 192.168.14.0 након излазног AS-Path preprending-a

Резултат додавања броја аутономног система у излазном смеру може да се види на слици 7.39 на којој се види да рута 192.168.14.0 има у *AS-Path*-у три додатне вредности аутономног система 2, а да су при томе остале и додатне ознаке аутономног система 1 које су додате раније, приликом модификације ове руте на уласку у аутономни систем 2, на рутеру 2.

7.5. Frame Mode MPLS

У алату GNS3 креирати топологију као на слици 7.40 која треба да симулира три повезана рутера, повезати рутере, укључити их и стартовати конзолни приступ. Након тога ће се унети основна конфигурација рутера (адресе интерфејса и интерни протокол рутирања) према следећим конфигурацијама:

Рутер R1:

```
enable
conf t
int f0/0
ip addr 192.168.12.1 255.255.255.0
no shut
int lo1
ip addr 192.168.111.1 255.255.255.0
int lo2
```

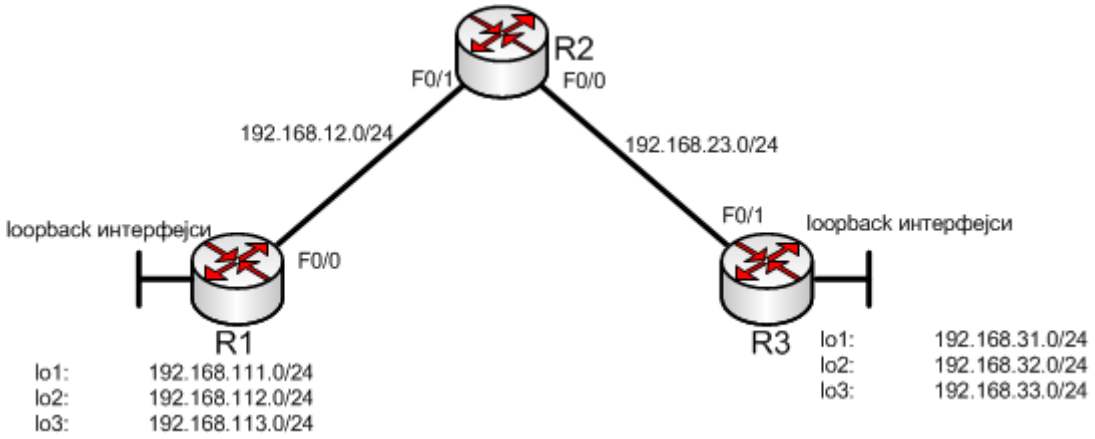
```
ip addr 192.168.112.1 255.255.255.0
int lo3
ip addr 192.168.113.1 255.255.255.0
router rip
network 192.168.12.0
network 192.168.111.0
network 192.168.112.0
network 192.168.113.0
```

Рутер R2:

```
enable
conf t
int f0/0
ip addr 192.168.23.2 255.255.255.0
no shut
int f0/1
ip addr 192.168.12.2 255.255.255.0
no shut
router rip
network 192.168.12.0
network 192.168.23.0
```

Рутер R3:

```
enable
conf t
int f0/0
ip addr 192.168.23.3 255.255.255.0
no shut
int lo1
ip addr 192.168.31.1 255.255.255.0
int lo2
ip addr 192.168.32.1 255.255.255.0
int lo3
ip addr 192.168.33.1 255.255.255.0
router rip
network 192.168.12.0
network 192.168.31.0
network 192.168.32.0
network 192.168.33.0
```



Слика 7.40 Топологија мреже у примеру са Frame Mode MPLS-ом

Овим ће бити успостављено повезивање рутера, пропагација свих рута и потпуна повезаност у мрежи. Да би се у овој мрежи успоставило прослеђивање на основу MPLS лабела потребно је конфигурисати интерфејсе између рутера на следећи начин:

Рутер R1:

```
enable
conf t
int f0/0
mpls ip
```

Рутер R2:

```
enable
conf t
int f0/0
mpls ip
int f0/1
mpls ip
```

Рутер R3:

```
enable
conf t
int f0/0
mpls ip
```

Врло брзо након уношења ових команди биће успостављене LDP сесије између рутера и *Frame-mode* MPLS. На слици 7.41 приказан је излаз команде којом се проверава статус LDP сесија рутера 2. Може да се види да постоје две LDP сесије са рутерима чији су идентификатори 192.168.33.1 и 192.168.113.1, што су рутери 1 и 3 који као идентификаторе имају највишу конфигурисану адресу.

```
R2#sh mpls ldp neighbor
    Peer LDP Ident: 192.168.33.1:0; Local LDP Ident 192.168.23.2:0
        TCP connection: 192.168.33.1.16224 - 192.168.23.2.646
        State: Oper; Msgs sent/rcvd: 27/28; Downstream
        Up time: 00:14:49
        LDP discovery sources:
          FastEthernet0/0, Src IP addr: 192.168.23.3
        Addresses bound to peer LDP Ident:
          192.168.23.3    192.168.31.1    192.168.32.1    192.168.33.1
    Peer LDP Ident: 192.168.113.1:0; Local LDP Ident 192.168.23.2:0
        TCP connection: 192.168.113.1.14931 - 192.168.23.2.646
        State: Oper; Msgs sent/rcvd: 27/27; Downstream
        Up time: 00:14:34
        LDP discovery sources:
          FastEthernet0/1, Src IP addr: 192.168.12.1
        Addresses bound to peer LDP Ident:
          192.168.12.1    192.168.111.1   192.168.112.1   192.168.113.1
```

Слика 7.41 LDP суседи рутера 2

Командом `sh mpls ldp bindings` могу да се виде парови (рута, лабела) које су рутери доделили и које су добијене од суседа путем LDP протокола. Види се да је свака рута из табеле рутирања добила другачију лабелу, како локално, тако и од суседа. Ако се посматра на пример рута ка мрежи 192.168.32.0/24 која је повезана на рутер 3, њој је рутер 2 доделио лабелу 20, рутер 1 лабелу 18, а рутер 3 лабелу *implicit-null*, што значи да ће се у мрежи користити механизам *Penultimate-Hop-Popping* (слика 7.42). Ово значи да се очекује да пакети који путују од рутера 1 ка мрежи 192.168.32.0/24 на сегменту R1-R2 имају лабелу 20 (добијена од низводног рутера за овај сегмент), а да на сегменту R2-R3 неће бити лабеле јер се користи механизам PHP. Треба обратити пажњу на то да не може да се очекује да ће приликом сваке реализације овог примера лабеле које се додељују бити исте, јер рутери додељују лабеле случајно и може да се деси да додељене лабеле буду другачије.

```
R2#sh mpls ldp bindings
  tib entry: 192.168.12.0/24, rev 2
        local binding:  tag: imp-null
        remote binding: tsr: 192.168.33.1:0, tag: 16
        remote binding: tsr: 192.168.113.1:0, tag: imp-null
  tib entry: 192.168.23.0/24, rev 10
        local binding:  tag: imp-null
        remote binding: tsr: 192.168.33.1:0, tag: imp-null
        remote binding: tsr: 192.168.113.1:0, tag: 17
  tib entry: 192.168.31.0/24, rev 4
        local binding:  tag: 16
        remote binding: tsr: 192.168.33.1:0, tag: imp-null
        remote binding: tsr: 192.168.113.1:0, tag: 16
  tib entry: 192.168.32.0/24, rev 14
        local binding:  tag: 20
        remote binding: tsr: 192.168.33.1:0, tag: imp-null
        remote binding: tsr: 192.168.113.1:0, tag: 18
  tib entry: 192.168.33.0/24, rev 16
        local binding:  tag: 21
        remote binding: tsr: 192.168.33.1:0, tag: imp-null
        remote binding: tsr: 192.168.113.1:0, tag: 19
  tib entry: 192.168.111.0/24, rev 6
        local binding:  tag: 17
        remote binding: tsr: 192.168.33.1:0, tag: 17
        remote binding: tsr: 192.168.113.1:0, tag: imp-null
  tib entry: 192.168.112.0/24, rev 12
        local binding:  tag: 19
        remote binding: tsr: 192.168.33.1:0, tag: 19
        remote binding: tsr: 192.168.113.1:0, tag: imp-null
  tib entry: 192.168.113.0/24, rev 8
        local binding:  tag: 18
        remote binding: tsr: 192.168.33.1:0, tag: 18
        remote binding: tsr: 192.168.113.1:0, tag: imp-null
```

Слика 7.42 Парови (рута, лабела) које је доделио рутер 2 и које су добијене од његових суседа путем LDP протокола

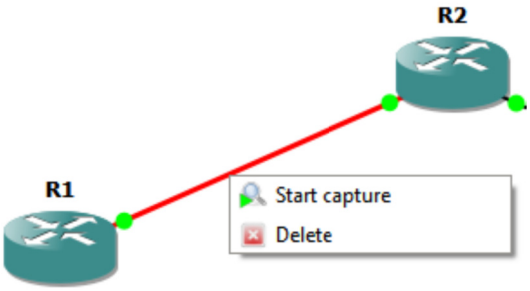
Сличан резултат може да се добије командом `sh mpls ip binding` чији је скраћени излаз, само за мрежу 192.168.32.0/24 приказан на слици 7.43. Види се да је за ову мрежу очекивана долазна лабела 20, а да у одлазном смеру нема лабеле због механизма PHP.

```
R2#sh mpls ip bind
...
  192.168.32.0/24
        in label:      20
        out label:     imp-null  lsr: 192.168.33.1:0   inuse
        out label:     18        lsr: 192.168.113.1:0
```

Слика 7.43 Лабеле за мрежу 192.168.32.0/24 на рутеру 2

Ово ће бити проверено и снимањем пакета који путују мрежом помоћу алата Wireshark који је повезан са GNS3 софтвером, а у „*all-in-one*“ GNS3 пакетима се налази и његова инсталација. На некој од веза десним кликом се изабере „*Start capture*“ што стартује

Wireshark на тој вези (слика 7.44). Том приликом је потребно је изабрати интерфејс у оквиру симулације на којем ће се вршити снимање пакета.



Слика 7.44 Покретање снимања пакета

Ако се изврши слање *ping* (ICMP ECHO) пакета са неке од адреса са рутера 1 (у овом примеру 192.168.111.1) ка адреси 192.168.32.1, на сегменту R1-R2 ови пакети ће изгледати као на слици 7.45.

```
▷ Frame 26: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
▷ Ethernet II, Src: c4:01:4a:bc:00:00 (c4:01:4a:bc:00:00), Dst: c4:02:3e:8c:00:01 (c4:02:3e:8c:00:01)
◢ MultiProtocol Label Switching Header, Label: 20 (Flow Label), Exp: 0, S: 1, TTL: 255
    0000 0000 0000 0001 0100 .... .... .... = MPLS Label: 20
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
    .... .... .... .... .... .... 1111 1111 = MPLS TTL: 255
▷ Internet Protocol Version 4, Src: 192.168.111.1, Dst: 192.168.32.1
▷ Internet Control Message Protocol
```

Слика 7.45 Пакет ка дестинацији 192.168.32.1 снимљен на вези R1-R2

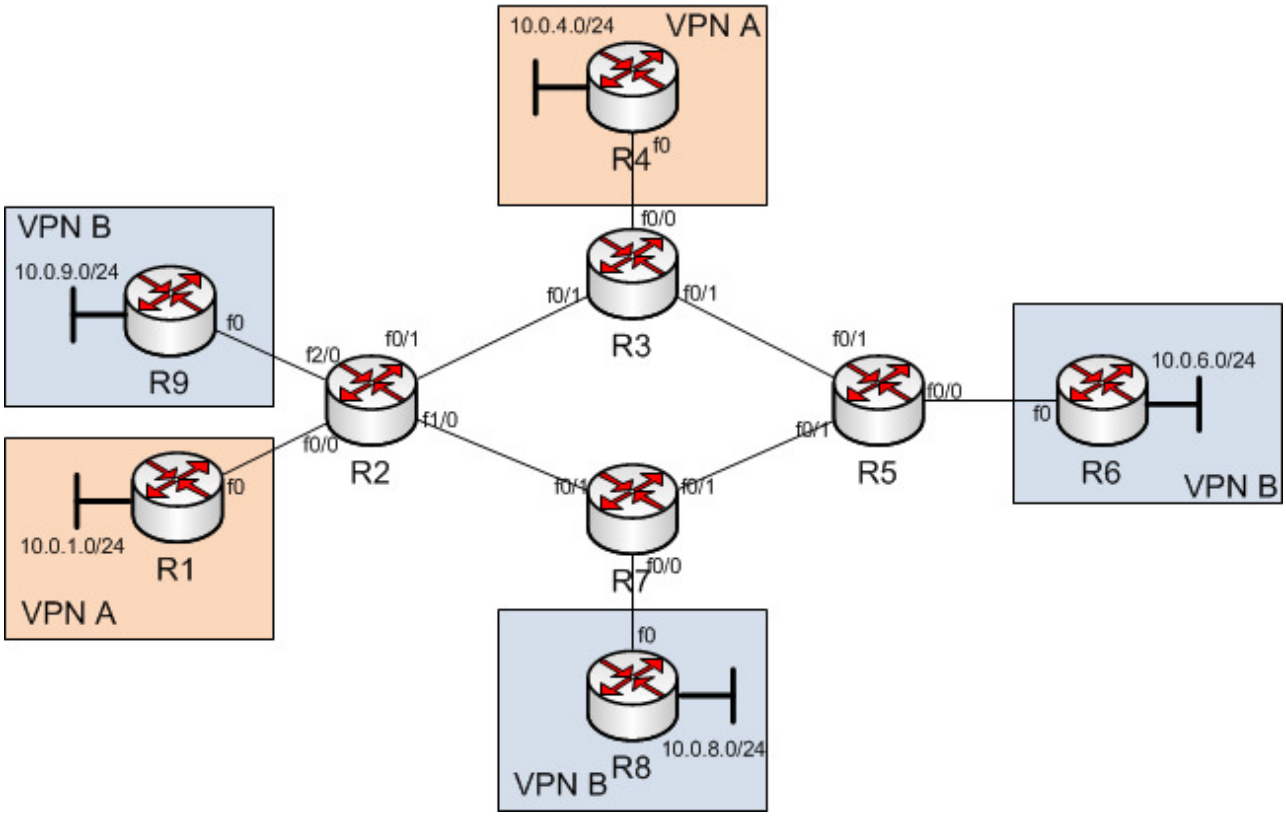
Може да се види да пакети имају између Етернет и IP заглавља MPLS лабелу са вредношћу 20, што је и очекивано из претходног излагања. Такође, у повратном смеру пакети неће имати лабелу (нема ничега између етернет и IP заглавља) и изгледаће као на слици 7.46, зато што се у мрежи користи механизам PHP. Ово може и да се додатно провери из исписа команде на слици 7.42, где се види да је рутер 1 за дестинацију 192.168.111.0/24 послао рутеру 2 лабелу *imp-null*, односно да очекује да пакети са дестинацијом 192.168.111.0/24 немају лабелу. Овако ће изгледати и претходно показани пакет са слике на сегменту између рутера 2 и 3 – неће имати лабелу.

```
Frame 27: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0
Ethernet II, Src: c4:02:3e:8c:00:01 (c4:02:3e:8c:00:01), Dst: c4:01:4a:bc:00:00 (c4:01:4a:bc:00:00)
Internet Protocol Version 4, Src: 192.168.32.1, Dst: 192.168.111.1
Internet Control Message Protocol
```

Слика 7.46 Пакет ка дестинацији 192.168.111.1 снимљен на вези R1-R2

7.6. Реализација L3VPN мрежа

У овом поглављу ће бити приказан начин на који се реализују MPLS L3VPN мреже. У GNS3 симулатору направити топологију као на слици 7.47. На овој топологији ће бити реализоване две виртуелне приватне мреже: VPN A која повезује две локације на којима су CE рутери 1 и 4 и VPN B која повезује три локације на којима су CE рутери 6, 8 и 9, док централни рутери 2, 3, 5 и 7 чине MPLS део мреже.



Слика 7.47 Топологија мреже у примеру реализације MPLS L3VPN

Прво ће бити конфигурисани CE рутери. Њихова конфигурација је једноставна – потребно је конфигурисати адресе и протокол рутирања и то само тако да се размењују руте између CE и PE рутера. Као и раније *Loopback* адресе симулирају мреже на CE локацијама. У овој мрежи је изабрано да се користи OSPF протокол рутирања за размену рута између CE и PE рутера.

R1

```
interface Loopback0
 ip address 10.0.1.1 255.255.255.0
!
interface FastEthernet0
```

```
 ip address 192.168.12.12 255.255.255.0
!
router ospf 1
 network 10.0.1.0 0.0.0.255 area 0
 network 192.168.12.12 0.0.0.0 area 0
```

R4

```
interface Loopback0
 ip address 10.0.4.1 255.255.255.0
!
interface FastEthernet0
 ip address 192.168.34.43 255.255.255.0
!
router ospf 1
 network 10.0.4.0 0.0.0.255 area 0
 network 192.168.34.43 0.0.0.0 area 0
```

R6

```
interface Loopback0
 ip address 10.0.6.1 255.255.255.0
!
interface FastEthernet0
 ip address 192.168.56.65 255.255.255.0
!
router ospf 100
 network 10.0.6.0 0.0.0.255 area 0
 network 192.168.56.65 0.0.0.0 area 0
```

R8

```
interface Loopback0
 ip address 10.0.8.1 255.255.255.0
!
interface FastEthernet0
 ip address 192.168.78.87 255.255.255.0
!
router ospf 101
 network 10.0.8.0 0.0.0.255 area 0
 network 192.168.78.87 0.0.0.0 area 0
```

R9

```
interface Loopback0
 ip address 10.0.9.1 255.255.255.0
!
interface FastEthernet0
 ip address 192.168.29.92 255.255.255.0
!
router ospf 100
 network 10.0.9.0 0.0.0.255 area 0
 network 192.168.29.0 0.0.0.255 area 0
```

Конфигурације PE рутера су знатно комплексније јер је поред основне мрежне конфигурације интерфејса потребно: декларисати VRF инстанце и дефинисати *Route Distinguisher* за њих, доделити интерфејсе VRF инстанцама, конфигурисати посебне инстанце протокола рутирања за сваку VRF инстанцу, те редистрибуцију у мултипротоколарни BGP. Ово је детаљније показано у оквиру коментара на конфигурацију рутера R2 која је и најсложенија јер овај рутер повезује две VPN мреже.

R2

```
!
! Konfiguracija VRFova i Route distinguishera
!
ip vrf VPNA
 rd 65000:110
 route-target export 65000:1100
 route-target import 65000:1100
!
ip vrf VPNB
 rd 65000:120
 route-target export 65000:1200
 route-target import 65000:1200
!
! Konfiguracija interfejsa
!
interface Loopback0
 ip address 172.16.2.1 255.255.255.255
!
interface FastEthernet0/0
 !dodeljivanje interfejsa VRF-u
 ip vrf forwarding VPNA
 ip address 192.168.12.21 255.255.255.0
!
interface FastEthernet0/1
 ip address 192.168.23.23 255.255.255.0
 mpls ip
!
interface FastEthernet1/0
 ip address 192.168.27.27 255.255.255.0
 mpls ip
!
interface FastEthernet2/0
 ip vrf forwarding VPNB
 ip address 192.168.29.29 255.255.255.0
!
! Konfiguracija protokola rutiranja prema CE ruteru
! Obratiti pažnju na redistribuciju BGP ruta u OSPF
!
router ospf 100 vrf VPNA
 redistribute bgp 65000 subnets
 network 192.168.12.21 0.0.0.0 area 0
!
! Konfiguracija protokola rutiranja prema CE ruteru
!
router ospf 101 vrf VPNB
 redistribute bgp 65000 subnets
 network 192.168.29.29 0.0.0.0 area 0
!
! Konfiguracija protokola rutiranja unutar MPLS mreže
!
router ospf 1
 network 172.16.2.1 0.0.0.0 area 0
 network 192.168.23.23 0.0.0.0 area 0
 network 192.168.27.27 0.0.0.0 area 0
!
! Konfiguracija BGP-a, MPLS mreža je u AS 65000
!
router bgp 65000
 no synchronization
```

```
bgp log-neighbor-changes
redistribute connected
neighbor iBGP peer-group
neighbor iBGP remote-as 65000
neighbor iBGP password iBGP_Password
neighbor iBGP update-source Loopback0
neighbor iBGP next-hop-self
neighbor iBGP send-community
neighbor iBGP soft-reconfiguration inbound
! Susedi su u istom AS i to su samo PE ruteri
neighbor 172.16.3.1 peer-group iBGP
neighbor 172.16.5.1 peer-group iBGP
neighbor 172.16.7.1 peer-group iBGP
no auto-summary
!
address-family vpnv4
 neighbor iBGP send-community extended
 neighbor 172.16.3.1 activate
 neighbor 172.16.5.1 activate
 neighbor 172.16.7.1 activate
exit-address-family
!
! Konfiguracija vrf-a za VPNB - Obratiti pažnju na to koji se
! OSPF redistribuira u MPBGP za ovaj VRF
!
address-family ipv4 vrf VPNB
 redistribute ospf 101 vrf VPNB
 neighbor 172.16.3.1 remote-as 65000
 neighbor 172.16.3.1 update-source Loopback0
 neighbor 172.16.3.1 activate
 neighbor 172.16.3.1 send-community extended
 neighbor 172.16.3.1 next-hop-self
 neighbor 172.16.5.1 remote-as 65000
 neighbor 172.16.5.1 update-source Loopback0
 neighbor 172.16.5.1 activate
 neighbor 172.16.5.1 send-community extended
 neighbor 172.16.5.1 next-hop-self
 neighbor 172.16.7.1 remote-as 65000
 neighbor 172.16.7.1 update-source Loopback0
 neighbor 172.16.7.1 activate
 neighbor 172.16.7.1 send-community extended
 neighbor 172.16.7.1 next-hop-self
 no synchronization
exit-address-family
!
! Konfiguracija vrf-a za VPNA - Obratiti pažnju na to koji se
! OSPF redistribuira u MPBGP za ovaj VRF
!
address-family ipv4 vrf VPNA
 redistribute ospf 100 vrf VPNA
 neighbor 172.16.3.1 remote-as 65000
 neighbor 172.16.3.1 update-source Loopback0
 neighbor 172.16.3.1 activate
 neighbor 172.16.3.1 send-community extended
 neighbor 172.16.3.1 next-hop-self
 neighbor 172.16.5.1 remote-as 65000
 neighbor 172.16.5.1 update-source Loopback0
 neighbor 172.16.5.1 activate
 neighbor 172.16.5.1 send-community extended
 neighbor 172.16.5.1 next-hop-self
 neighbor 172.16.7.1 remote-as 65000
```

```
  neighbor 172.16.7.1 update-source Loopback0
  neighbor 172.16.7.1 activate
  neighbor 172.16.7.1 send-community extended
  neighbor 172.16.7.1 next-hop-self
  no synchronization
 exit-address-family
```

R3

```
ip vrf VPNA
 rd 65000:110
 route-target export 65000:1100
 route-target import 65000:1100
!
ip vrf VPNB
 rd 65000:120
 route-target export 65000:1200
 route-target import 65000:1200
!
interface Loopback0
 ip address 172.16.3.1 255.255.255.255
!
interface FastEthernet0/0
 ip vrf forwarding VPNA
 ip address 192.168.34.34 255.255.255.0
!
interface FastEthernet0/1
 ip address 192.168.23.32 255.255.255.0
 mpls ip
!
interface FastEthernet1/0
 ip address 192.168.35.35 255.255.255.0
 mpls ip
!
router ospf 100 vrf VPNA
 redistribute bgp 65000 subnets
 network 192.168.34.34 0.0.0.0 area 0
!
router ospf 1
 network 172.16.3.1 0.0.0.0 area 0
 network 192.168.23.32 0.0.0.0 area 0
 network 192.168.35.35 0.0.0.0 area 0
!
router bgp 65000
 neighbor iBGP peer-group
 neighbor iBGP remote-as 65000
 neighbor iBGP password iBGP_Password
 neighbor iBGP update-source Loopback0
 neighbor 172.16.2.1 peer-group iBGP
 neighbor 172.16.2.1 update-source Loopback0
 neighbor 172.16.5.1 remote-as 65000
 neighbor 172.16.5.1 peer-group iBGP
 neighbor 172.16.5.1 update-source Loopback0
 neighbor 172.16.7.1 remote-as 65000
 neighbor 172.16.7.1 peer-group iBGP
 neighbor 172.16.7.1 update-source Loopback0
 !
 address-family ipv4
  redistribute connected
  neighbor iBGP send-community
  neighbor iBGP next-hop-self
```

```
  neighbor iBGP soft-reconfiguration inbound
  neighbor 172.16.2.1 activate
  neighbor 172.16.5.1 activate
  neighbor 172.16.7.1 activate
  no auto-summary
  no synchronization
 exit-address-family
 !
 address-family vpnv4
  neighbor iBGP send-community extended
  neighbor 172.16.2.1 activate
  neighbor 172.16.5.1 activate
  neighbor 172.16.7.1 activate
 exit-address-family
 !
 address-family ipv4 vrf VPNA
  redistribute ospf 100 vrf VPNA
  neighbor 172.16.2.1 remote-as 65000
  neighbor 172.16.2.1 update-source Loopback0
  neighbor 172.16.2.1 activate
  neighbor 172.16.2.1 send-community extended
  neighbor 172.16.2.1 next-hop-self
  neighbor 172.16.5.1 remote-as 65000
  neighbor 172.16.5.1 update-source Loopback0
  neighbor 172.16.5.1 activate
  neighbor 172.16.5.1 send-community extended
  neighbor 172.16.5.1 next-hop-self
  neighbor 172.16.7.1 remote-as 65000
  neighbor 172.16.7.1 update-source Loopback0
  neighbor 172.16.7.1 activate
  neighbor 172.16.7.1 send-community extended
  neighbor 172.16.7.1 next-hop-self
  no synchronization
 exit-address-family
```

R5

```
ip vrf VPNB
 rd 65000:120
 route-target export 65000:1200
 route-target import 65000:1200
!
interface Loopback0
 ip address 172.16.5.1 255.255.255.255
!
interface FastEthernet0/0
 ip vrf forwarding VPNB
 ip address 192.168.56.56 255.255.255.0
!
interface FastEthernet0/1
 ip address 192.168.35.53 255.255.255.0
 mpls ip
!
interface FastEthernet1/0
 ip address 192.168.57.57 255.255.255.0
 mpls ip
!
router ospf 101 vrf VPNB
 redistribute bgp 65000 subnets
 network 192.168.56.56 0.0.0.0 area 0
!
```

```
router ospf 1
 network 172.16.5.1 0.0.0.0 area 0
 network 192.168.35.53 0.0.0.0 area 0
 network 192.168.57.57 0.0.0.0 area 0
!
router bgp 65000
 no synchronization
 redistribute connected
 neighbor iBGP peer-group
 neighbor iBGP remote-as 65000
 neighbor iBGP password iBGP_Password
 neighbor iBGP update-source Loopback0
 neighbor iBGP next-hop-self
 neighbor iBGP send-community
 neighbor iBGP soft-reconfiguration inbound
 neighbor 172.16.2.1 peer-group iBGP
 neighbor 172.16.3.1 peer-group iBGP
 neighbor 172.16.7.1 peer-group iBGP
 no auto-summary
 !
 address-family vpnv4
  neighbor iBGP send-community extended
  neighbor 172.16.2.1 activate
  neighbor 172.16.3.1 activate
  neighbor 172.16.7.1 activate
 exit-address-family
 !
 address-family ipv4 vrf VPNB
  redistribute ospf 101 vrf VPNB
  neighbor 172.16.2.1 remote-as 65000
  neighbor 172.16.2.1 update-source Loopback0
  neighbor 172.16.2.1 activate
  neighbor 172.16.2.1 send-community extended
  neighbor 172.16.2.1 next-hop-self
  neighbor 172.16.3.1 remote-as 65000
  neighbor 172.16.3.1 update-source Loopback0
  neighbor 172.16.3.1 activate
  neighbor 172.16.3.1 send-community extended
  neighbor 172.16.3.1 next-hop-self
  neighbor 172.16.7.1 remote-as 65000
  neighbor 172.16.7.1 update-source Loopback0
  neighbor 172.16.7.1 activate
  neighbor 172.16.7.1 send-community extended
  neighbor 172.16.7.1 next-hop-self
  no synchronization
 exit-address-family
```

R7

```
ip vrf VPNB
 rd 65000:120
 route-target export 65000:1200
 route-target import 65000:1200
!
interface Loopback0
 ip address 172.16.7.1 255.255.255.255
!
interface FastEthernet0/0
 ip vrf forwarding VPNB
 ip address 192.168.78.78 255.255.255.0
```

```
!
interface FastEthernet0/1
 ip address 192.168.57.75 255.255.255.0
 mpls ip
!
interface FastEthernet1/0
 ip address 192.168.27.72 255.255.255.0
 mpls ip
!
router ospf 101 vrf VPNB
 redistribute bgp 65000 subnets
 network 192.168.78.78 0.0.0.0 area 0
!
router ospf 1
 network 172.16.7.1 0.0.0.0 area 0
 network 192.168.27.72 0.0.0.0 area 0
 network 192.168.57.75 0.0.0.0 area 0
 network 192.168.87.7 0.0.0.0 area 0
!
router bgp 65000
 no synchronization
 redistribute connected
 neighbor iBGP peer-group
 neighbor iBGP remote-as 65000
 neighbor iBGP password iBGP_Password
 neighbor iBGP update-source Loopback0
 neighbor iBGP next-hop-self
 neighbor iBGP send-community
 neighbor iBGP soft-reconfiguration inbound
 neighbor 172.16.2.1 peer-group iBGP
 neighbor 172.16.3.1 peer-group iBGP
 neighbor 172.16.5.1 peer-group iBGP
 no auto-summary
 !
 address-family vpnv4
  neighbor iBGP send-community extended
  neighbor 172.16.2.1 activate
  neighbor 172.16.3.1 activate
  neighbor 172.16.5.1 activate
 exit-address-family
 !
 address-family ipv4 vrf VPNB
  redistribute ospf 101 vrf VPNB
  neighbor 172.16.2.1 remote-as 65000
  neighbor 172.16.2.1 update-source Loopback0
  neighbor 172.16.2.1 activate
  neighbor 172.16.2.1 send-community extended
  neighbor 172.16.2.1 next-hop-self
  neighbor 172.16.3.1 remote-as 65000
  neighbor 172.16.3.1 update-source Loopback0
  neighbor 172.16.3.1 activate
  neighbor 172.16.3.1 send-community extended
  neighbor 172.16.3.1 next-hop-self
  neighbor 172.16.5.1 remote-as 65000
  neighbor 172.16.5.1 update-source Loopback0
  neighbor 172.16.5.1 activate
  neighbor 172.16.5.1 send-community extended
  neighbor 172.16.5.1 next-hop-self
  no synchronization
 exit-address-family
```

Након што се унесу конфигурације и након што мрежа потпуно конвергира, може да се провери стање рутера и начин на који преноси пакете. На CE рутерима у таβелама рутирања постоје само руте које су директно повезане на ове рутере и руте доβијене од других CE рутера (мреже 10.0.1.0/24 и 10.0.4.0/24 на рутеру 1) (слика 7.48).

```
R1>sh ip ro
...
C    192.168.12.0/24 is directly connected, FastEthernet0
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.1.0/24 is directly connected, Loopback0
O IA    10.0.4.1/32 [110/12] via 192.168.12.21, 00:34:28, FastEthernet0
O IA 192.168.34.0/24 [110/2] via 192.168.12.21, 00:39:20, FastEthernet0
```

Слика 7.48 Таβела рутирања на клијентском рутеру

Са друге стране, PE рутери имају више инстанци таβела рутирања. Скраћени прикази таβела рутирања, за рутер 2 и за VRF инстанце за VPN A и VPN B су приказане на сликама 7.49, 7.50 и 7.51.

```
R2#sh ip ro
...
C    192.168.27.0/24 is directly connected, FastEthernet1/0
O    192.168.57.0/24 [110/11] via 192.168.27.72, 00:31:25, FastEthernet1/0
     172.16.0.0/32 is subnetted, 4 subnets
O       172.16.5.1 [110/12] via 192.168.27.72, 00:31:25, FastEthernet1/0
                   [110/12] via 192.168.23.32, 00:31:25, FastEthernet0/1
O       172.16.7.1 [110/2] via 192.168.27.72, 00:31:25, FastEthernet1/0
O       172.16.3.1 [110/11] via 192.168.23.32, 00:31:25, FastEthernet0/1
C       172.16.2.1 is directly connected, Loopback0
C    192.168.23.0/24 is directly connected, FastEthernet0/1
B    192.168.186.0/24 [200/0] via 172.16.7.1, 00:30:56
O    192.168.35.0/24 [110/11] via 192.168.23.32, 00:31:28, FastEthernet0/1
```

Слика 7.49 Основна таβела рутирања ивичног рутера

```
R2#sh ip ro vrf VPNA
...
C    192.168.12.0/24 is directly connected, FastEthernet0/0
     10.0.0.0/32 is subnetted, 2 subnets
O       10.0.1.1 [110/11] via 192.168.12.12, 00:31:50, FastEthernet0/0
B       10.0.4.1 [200/11] via 172.16.3.1, 00:27:17
B    192.168.34.0/24 [200/0] via 172.16.3.1, 00:30:48
```

Слика 7.50 Таβела рутирања VRF инстанце A на рутеру 2

У таβели рутирања рутера 2 се виде руте које се налазе само на PE рутерима у оквиру MPLS мреже, док посеβне таβеле рутирања VRF инстанци имају руте доβијене од CE рутера. Дакле, нема мешања рута између различитих VRF инстанци, већ свака има своје независно рутирање.

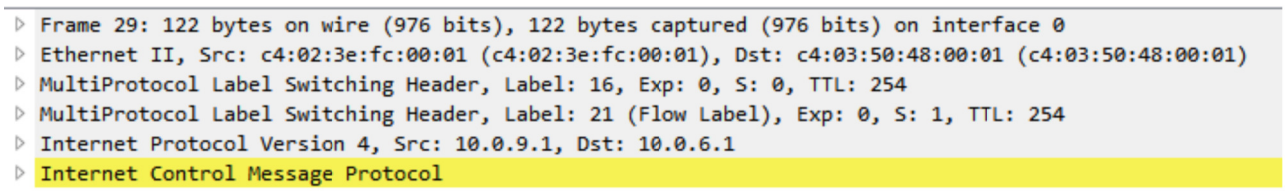
```
R2#sh ip ro vrf VPNB
...

C    192.168.29.0/24 is directly connected, FastEthernet2/0
     10.0.0.0/32 is subnetted, 3 subnets
O       10.0.9.1 [110/2] via 192.168.29.92, 00:31:55, FastEthernet2/0
B       10.0.8.1 [200/11] via 172.16.7.1, 00:30:53
B       10.0.6.1 [200/11] via 172.16.5.1, 00:30:53
```

Слика 7.51 Табела рутирања VRF инстанце B на рутеру 2

Лабеле које ће имати пакети када се рутирају кроз MPLS мрежу могу да се виде командама са слике 7.52. Посматраће се пакети који путују од рутера 9 ка рутеру 6 кроз VPN B. Као што може да се види лабела за мрежу 10.0.6.1/32 је 21. Пошто је ово рута која припада VRF инстанци B, ова лабела ће бити унутрашња лабела у пакетима који излазе из рутера 2, а који иду ка тој мрежи. Спољашња лабела може да се види уз руту ка мрежи 172.16.5.1/32 што је идентификатор рутера 5 који је излазни PE рутер за мрежу 10.0.6.1/32 и ова лабела је 16.

```
R2>sh ip bgp vpnv4 all labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 65000:110 (VPNA)
...
Route Distinguisher: 65000:120 (VPNB)
   10.0.6.1/32      172.16.5.1      nolabel/21
   10.0.8.1/32      172.16.7.1      nolabel/21
...

R2>sh mpls forwarding
Local  Outgoing    Prefix            Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
...
19     20          172.16.5.1/32     0          Fa1/0       192.168.27.72
       16          172.16.5.1/32     0          Fa0/1       192.168.23.32
20     Pop tag     172.16.7.1/32     0          Fa1/0       192.168.27.72
21     Untagged    10.0.1.1/32[V]    0          Fa0/0       192.168.12.12
...
```

Слика 7.52 Лабеле пакета који се рутирају од рутера 9 до рутера 6 кроз VPN B на првом мрежном сегменту од рутера 2



Слика 7.53 Пакет који припада VPN B на вези између рутера 2 и 3 – 2 лабеле

Ово је проверено снимањем пакета у мрежи на везама између рутера 2 и 3 и рутера 3 и 5, што може да се види на сликама 7.53 и 7.54.

На првој вези се виде обе лабеле и то баш у редоследу како је и очекивано на основу MPLS табела: унутрашња лабела, ближа IP заглављу је 21, а спољашња, ближа етернет заглављу је 16. Обратити пажњу да спољашња лабела има вредност S бита 0 што значи да иза спољашње лабеле следи још једна лабела. Са друге стране овај бит има вредност 1 на унутрашњој лабели која је последња.

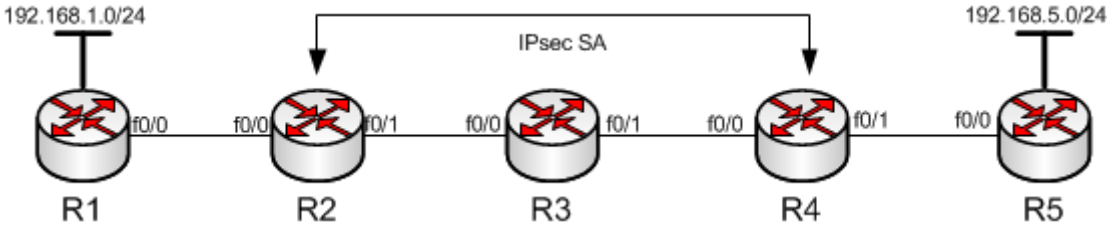
```
Frame 17: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
Ethernet II, Src: c4:03:50:48:00:10 (c4:03:50:48:00:10), Dst: c4:05:3a:dc:00:01 (c4:05:3a:dc:00:01)
MultiProtocol Label Switching Header, Label: 21 (Flow Label), Exp: 0, S: 1, TTL: 253
Internet Protocol Version 4, Src: 10.0.9.1, Dst: 10.0.6.1
Internet Control Message Protocol
```

Слика 7.54 Пакет који припада VPN B на вези између рутера 3 и 5 – 1 лабела

На другој вези постоји само једна лабела зато што се код MPLS L3VPN мрежа подразумева механизам PHP. Ово је лабела 21, унутрашња са првог рутера која мора да остане непромењена приликом проласка кроз мрежу, како би пакет могао да се правилно разврста на долазном PE рутеру.

7.7. Реализација IPsec VPN

У овом примеру биће направљена симулација *Site-to-Site* IPsec VPN којом се криптованим мрежним тунелом повезују локалне рачунарске мреже као на слици 7.55. У овом случају реализована IPsec сигурносна асоцијација криптује садржај пакета који путују између рутера 1 и 5, а уређаји који успостављају IPsec сигурносну асоцијацију су рутери 2 и 4.



Слика 7.55 Топологија мреже у примеру реализације IPsec VPN

Конфигурација рутера 1 и 5 је једноставна и садржи само конфигурацију интерфејса и *default* руту која указује на једину могућу излазну путању.

R2

```
interface Loopback0
 ip address 192.168.1.1 255.255.255.0
!
interface FastEthernet0
 ip address 192.168.12.1 255.255.255.0
 no shut
```

```
ip route 0.0.0.0 0.0.0.0 192.168.12.2
```

R5

```
interface Loopback0
 ip address 192.168.5.1 255.255.255.0
!
interface FastEthernet0
 ip address 192.168.45.5 255.255.255.0
 no shut

ip route 0.0.0.0 0.0.0.0 192.168.45.4
```

Између рутера 2, 3 и 4 ће бити успостављен OSPF протокол рутирања. Конфигурација рутера 2 и 4 ће бити сложенија јер се на њима дефинишу параметри IKE и IPsec сигурносних асоцијација што ће бити показано у описима конфигурација, док рутер 3 нема никакву информацију о томе да постоје сигурносне асоцијације већ само преноси пакете.

R2

```
! definicija prve faze razmene ključeva:
! definišu se protokol enkripcije (AES256), način autentikacije (preshare –
! unapred deljeni ključevi), veličina Difi Helman grupe, lozinka i adresa
! drugog kraja sigurnosne asocijacije
!
crypto isakmp policy 10
 encr aes 256
 authentication pre-share
 group 5
crypto isakmp key vpnuser address 192.168.34.4
!
! definicija IPsec protokola koji se koristi – ESP sa AES algoritmom
!
crypto ipsec transform-set myset esp-aes esp-sha-hmac
!
! definicija SPD polise koja određuje za koje pakete se primenjuje ova SA
!
crypto map mymap 10 ipsec-isakmp
 set peer 192.168.34.4
 set transform-set myset
 match address 100
!
interface FastEthernet0/0
 ip address 192.168.12.2 255.255.255.0
 no shut
!
! IPsec se primenjuje na ovom interfejsu kada paketi izlaze ka R3
!
interface FastEthernet0/1
 ip address 192.168.23.2 255.255.255.0
 crypto map mymap
 no shut
!
router ospf 1
 network 192.168.23.0 0.0.0.255 area 0
ip route 0.0.0.0 0.0.0.0 192.168.23.3
ip route 192.168.1.0 255.255.255.0 192.168.12.1
!
! U IPsec sigurnosnu asocijaciju se unose samo
```

```
!
access-list 100 permit ip 192.168.1.0 0.0.0.255 192.168.5.0 0.0.0.255
```

R3

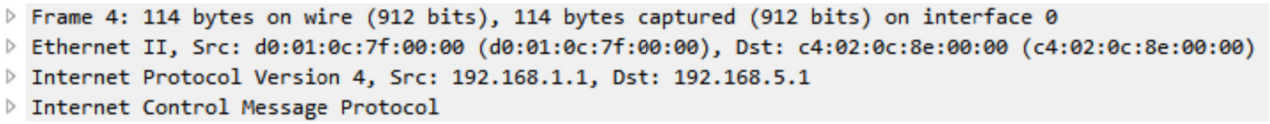
```
interface FastEthernet0/0
 ip address 192.168.23.3 255.255.255.0
 no shut
interface FastEthernet0/1
 ip address 192.168.34.3 255.255.255.0
 no shut
router ospf 1
 network 192.168.23.0 0.0.0.255 area 0
 network 192.168.34.0 0.0.0.255 area 0
```

R4

```
crypto isakmp policy 10
 encr aes 256
 authentication pre-share
 group 5
crypto isakmp key vpnuser address 192.168.23.2
!
crypto ipsec transform-set myset esp-aes esp-sha-hmac
!
crypto map mymap 10 ipsec-isakmp
 set peer 192.168.23.2
 set transform-set myset
 match address 100
!
interface FastEthernet0/0
 ip address 192.168.34.4 255.255.255.0
 crypto map mymap
 no shut
!
interface FastEthernet0/1
 ip address 192.168.45.4 255.255.255.0
 no shut
!
router ospf 1
 network 192.168.34.0 0.0.0.255 area 0
!
ip route 0.0.0.0 0.0.0.0 192.168.34.3
ip route 192.168.5.0 255.255.255.0 192.168.45.5
!
access-list 100 permit ip 192.168.5.0 0.0.0.255 192.168.1.0 0.0.0.255
```

По конфигурацији рутера, сигурносне асоцијације ће се успоставити приликом проласка првог пакета који пролази између мрежа 192.168.1.0/24 и 192.168.5.0/24 (може да се пусти *ping* са рутера 1 ка рутеру 5 уз спецификацију изворишне и одредишне адресе). Пошто је креирање сигурносних асоцијација процесорски интензивно, првих неколико пакета који би требало да прођу сигурносном асоцијацијом ће се изгубити. Међутим, након успостављања сигурносне асоцијације сви пакети између поменутих мрежа ће пролазити кроз криптовани мрежни тунел. Ово ће бити показано снимањем пакета на везама између рутера 1 и 2 и рутера 2 и 3 (слике 7.56 и 7.58). На првој вези је ово обичан пакет у којем је ICMP ECHO

порука енкапсулирана у IP и који није криптован. IP адресе у овом пакету су адресе рутера 1 и 5 које су дефинисане *ping* командом, а које упадају у SPD филтер дефинисан аксес листом.



Слика 7.56 Пакет снимљен на вези између рутера 1 и 2, пре енкрипције на рутеру 2

На другој вези се види да је пакет криптован, да се користи ESP заглавље, али и да се користи тунел режим рада, зато што су IP адресе у овом пакету, адресе рутера 2 и 4 који су на крајевима сигурносне асоцијације, док су оригиналне IP адресе криптоване и невиљиве унутар ESP заглавља.

```
R2#sh crypto ipsec sa

interface: FastEthernet0/1
    Crypto map tag: mymap, local addr 192.168.23.2

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (192.168.5.0/255.255.255.0/0/0)
   current_peer 192.168.34.4 port 500
     PERMIT, flags={origin_is_acl,ipsec_sa_request_sent}
...
     current outbound spi: 0x9F216367(2669765494)

     inbound esp sas:
      spi: 0x9FA5C252(2678440530)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
...
        Status: ACTIVE
...
     outbound esp sas:
      spi: 0x9F216367(2669765494)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
...
        Status: ACTIVE
...
```

Слика 7.57

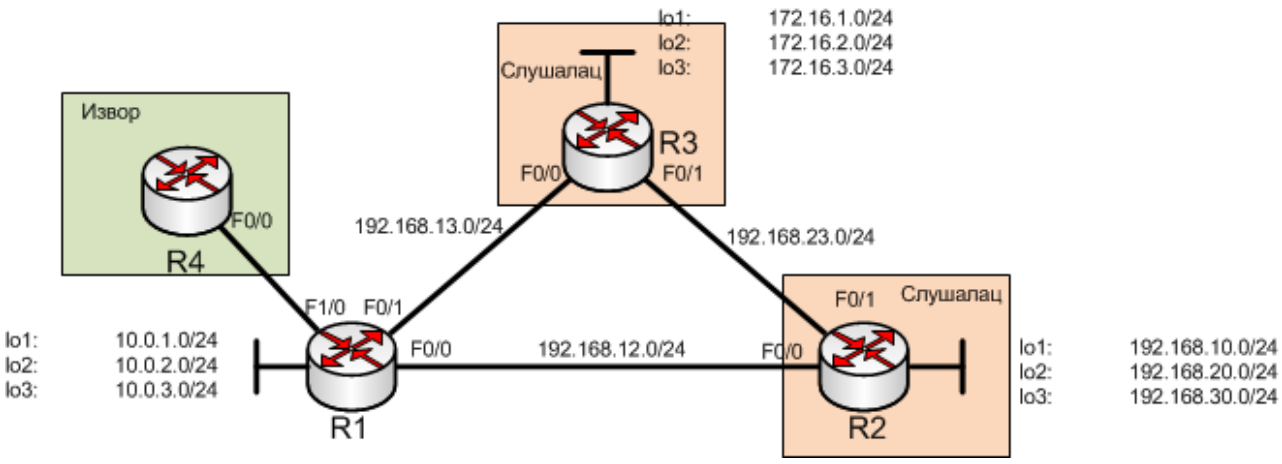
Ово може да се провери и контролним командама на рутерима 2 или 4. На слици 7.57 је приказан статус сигурносних асоцијација које су креиране овом приликом. Пошто је комуникација двосмерна, биће креиране две сигурносне асоцијације, по једна у сваком смеру, а као што може да се види, SPI индекс са рутера одговара индексу у послатим пакетима.

```
Frame 6: 182 bytes on wire (1456 bits), 182 bytes captured (1456 bits) on interface 0
Ethernet II, Src: c4:02:0c:8e:00:01 (c4:02:0c:8e:00:01), Dst: c4:03:0c:9d:00:00 (c4:03:0c:9d:00:00)
Internet Protocol Version 4, Src: 192.168.23.2, Dst: 192.168.34.4
Encapsulating Security Payload
    ESP SPI: 0x9f216376 (2669765494)
    ESP Sequence: 9
```

Слика 7.58 Исти пакет са слике 7.56, снимљен на вези између рутера 2 и 3

7.8. Конфигурисање мултикаста у рачунарским мрежама

У овом примеру ће се користити мало модификована топологија из поглавља 7.2 за мрежу са RIP протоколом рутирања, а сматраће се да су у почетном стању на рутерима 1, 2 и 3 конфигурисане адресе и протокол рутирања, као на крају поглавља 7.2.2. Рутер 4 има конфгурисану адресу и RIP протокол рутирања. У овој топологији рутер 4 ће бити извор мултикаст саобраћаја, а рутери 2 и 3 ће бити слушаоци.



Слика 7.59 Топологија мреже у примеру конфигурације PIM DM мултикаста

Након што је направљена основна конфигурација рутера 1, 2 и 3, потребно да се омогући мултикаст рутирање на свим рутерима и пусти PIM DM протокол на свим интерфејсима између рутера на следећи начин:

R1

```
ip multicast-routing
int f0/0
 ip pim dense-mode
int f0/1
 ip pim dense-mode
int f1/0
 ip address 192.168.41.1 255.255.255.0
 ip pim dense-mode
 no shut
router rip
 network 192.168.41.0
```

R2

```
ip multicast-routing
```

```
int f0/0
 ip pim dense-mode
int f0/1
 ip pim dense-mode
```

R3

```
ip multicast-routing
int f0/0
 ip pim dense-mode
int f0/1
 ip pim dense-mode
```

R4

```
ip multicast-routing
int f0/0
 ip address 192.168.41.4 255.255.255.0
 ip pim dense-mode
 no shut
router rip
 network 192.168.41.0
```

Након овога мрежа је спремна за пренос мултикаст пакета. Да би се рутери 2 и 3 регистровали као слушаоци мултикаст групе 233.233.233.233, потребно је на неком од њихових интерфејса урадити следеће:

```
int lo1
 ip igmp join 233.233.233.233
```

```
R4#ping 233.233.233.233

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 233.233.233.233, timeout is 2 seconds:

Reply to request 0 from 192.168.13.3, 72 ms
Reply to request 0 from 192.168.12.2, 72 ms
```

Слика 7.60 мултикаст пинг – на један послати пакет долазе два одговора

Уколико се сада на рутеру R4 покрене *ping* команда са дестинационом адресом 233.233.233.233, за сваки послати пакет добиће се два одговора, један од рутера 2 (са адресе 192.168.12.2) и један од рутера 3 (са адресе 192.168.13.3) који су регистровани као слушаоци ове мултикаст групе (слика 7.60).

Интересантно је погледати како изгледају мултикаст табеле рутирања у овој ситуацији (слика 7.61). Рутер 2 ће имати у мултикаст табели рутирања следећи улаз: (192.168.41.4, 233.233.233.233), где је 192.168.41.4 адреса извора мултикаст пакета. У листи излазних интерфејса ће бити интерфејс f0/1 што је интерфејс према рутеру 3, али ће овај интерфејс бити у статусу *Prune*, односно на њега неће бити прослеђивани мултикаст пакети, јер овај интерфејс не пролази RPF проверу за дати извор мултикаст пакета. RPF сусед рутера 2 је

рутер 1, односно рутер са адресом 192.168.12.1 који је доступан преко интерфејса f0/0 – то је рутер са кога се очекују мултикаст пакети од извора 192.168.41.1.

```
R2#sh ip mroute
...
(*, 233.233.233.233), 00:00:17/stopped, RP 0.0.0.0, flags: DL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/1, Forward/Dense, 00:00:17/00:00:00
    FastEthernet0/0, Forward/Dense, 00:00:17/00:00:00

(192.168.41.4, 233.233.233.233), 00:00:17/00:02:47, flags: PLT
  Incoming interface: FastEthernet0/0, RPF nbr 192.168.12.1
  Outgoing interface list:
    FastEthernet0/1, Prune/Dense, 00:00:19/00:02:40
```

Слика 7.61 Изглед дела мултикаст табеле рутирања за мултикаст адресу 233.233.233.233 на рутеру 2

Слично ће бити на рутеру 3 – излазни интерфејс према рутеру 2 ће бити у статусу *Prune* (слика 7.62).

```
R3#sh ip mro
...
(*, 233.233.233.233), 00:00:10/stopped, RP 0.0.0.0, flags: DL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/1, Forward/Dense, 00:00:10/00:00:00
    FastEthernet0/0, Forward/Dense, 00:00:10/00:00:00

(192.168.41.4, 233.233.233.233), 00:00:10/00:02:50, flags: PLT
  Incoming interface: FastEthernet0/0, RPF nbr 192.168.13.1
  Outgoing interface list:
    FastEthernet0/1, Prune/Dense, 00:00:12/00:02:48, A
```

Слика 7.62 Изглед дела мултикаст табеле рутирања за мултикаст адресу 233.233.233.233 на рутеру 3

Рутер 1 ће имати 2 излазна интерфејса: f0/2 и f0/1 и оба ће бити у статусу *Forward* што значи да се преко њих преносе пакети суседима (слика 7.61). Све ово значи да је у мултикаст мрежи у овом примеру формирано мултикаст стабло са кореном у рутеру 4, са гранама 4-1, 1-2 и 1-3, док је грана 2-3 избачена из мултикаст стабла.

Уколико би се сачекало 3 минута (период плављења PIM DM) улази у мултикаст табели који се односе не мултикаст групу 233.233.233.233 би се изгубили, јер у недостатку нових мултикаст пакета за ову групу неће се радити периодично плављење мреже мултикаст пакетима.

Студентима се препоручује да испробају и PIM SM тако што ће на свим интерфејсима свих рутера који су на путањи пакета искључити PIM DM и укључити PIM SM на следећи начин:

```
interface f0/0
 no ip pim dense/mode
 ip pim sparse mode
```

И тако што ће се на свим рутерима дефинисати *Rendez-vous Point* у глобалном конфигурационом режиму:

```
ip pim rp-address 192.168.12.1
```

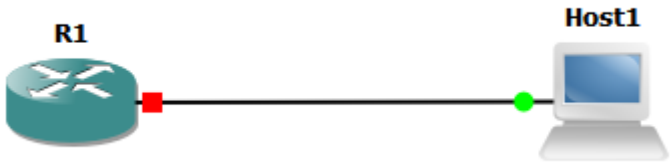
чиме ће рутер 1, са адресом 192.168.12.1 добити улогу RP.

```
R1#sh ip mro
...
(*, 233.233.233.233), 00:00:38/stopped, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet1/0, Forward/Dense, 00:00:38/00:00:00
    FastEthernet0/1, Forward/Dense, 00:00:38/00:00:00
    FastEthernet0/0, Forward/Dense, 00:00:38/00:00:00

(192.168.41.4, 233.233.233.233), 00:00:38/00:02:25, flags: T
  Incoming interface: FastEthernet1/0, RPF nbr 192.168.41.4
  Outgoing interface list:
    FastEthernet0/0, Forward/Dense, 00:00:41/00:00:00
    FastEthernet0/1, Forward/Dense, 00:00:41/00:00:00
```

Слика 7.63 Изглед дела мултикаст табеле рутирања за мултикаст адресу 233.233.233.233 на рутеру 1

7.9. Надгледање уређаја SNMP протоколом

У сваку топологију креирану у GNS3 симулатору може да се дода и рачунар на којем је покренут симулатор (слика 7.64). Повезивање се обавља између неког од интерфејса рутера и неког од виртуелних интерфејса на рачунару (то могу да буду *Loopback* интерфејси или интерфејси које креирају алати за виртуелизацију). Адресе на рачунару и на рутеру на тој вези морају да припадају истој мрежи.



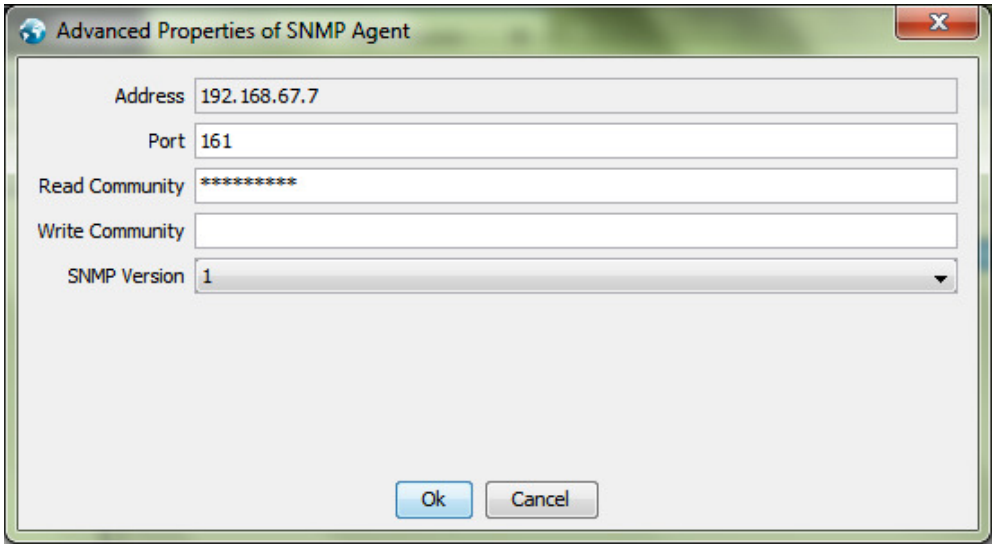
Слика 7.64 Топологија мреже у примеру надгледања помоћу SNMP протокола

Након тога може да се конфигурише рутер на следећи начин:

```
interface FastEthernet0/0
 ip address 192.168.67.7 255.255.255.0
 no shutdown
snmp-server community snmpproba RO
```

чиме се конфигурише адреса за комуникацију ка рачунару (рачунар треба да има адресу из истог адресног опсега) и SNMP агент на рутеру тако да ради као верзија 1 са вредношћу `community` параметра за читање SNMP променљивих: `snmpproba`.

Након овога може на рачунару да се покрене неки алат за претрагу MIB база каквих има више бесплатно доступних. У овом примеру је коришћен iReasoning MIB Browser[55].

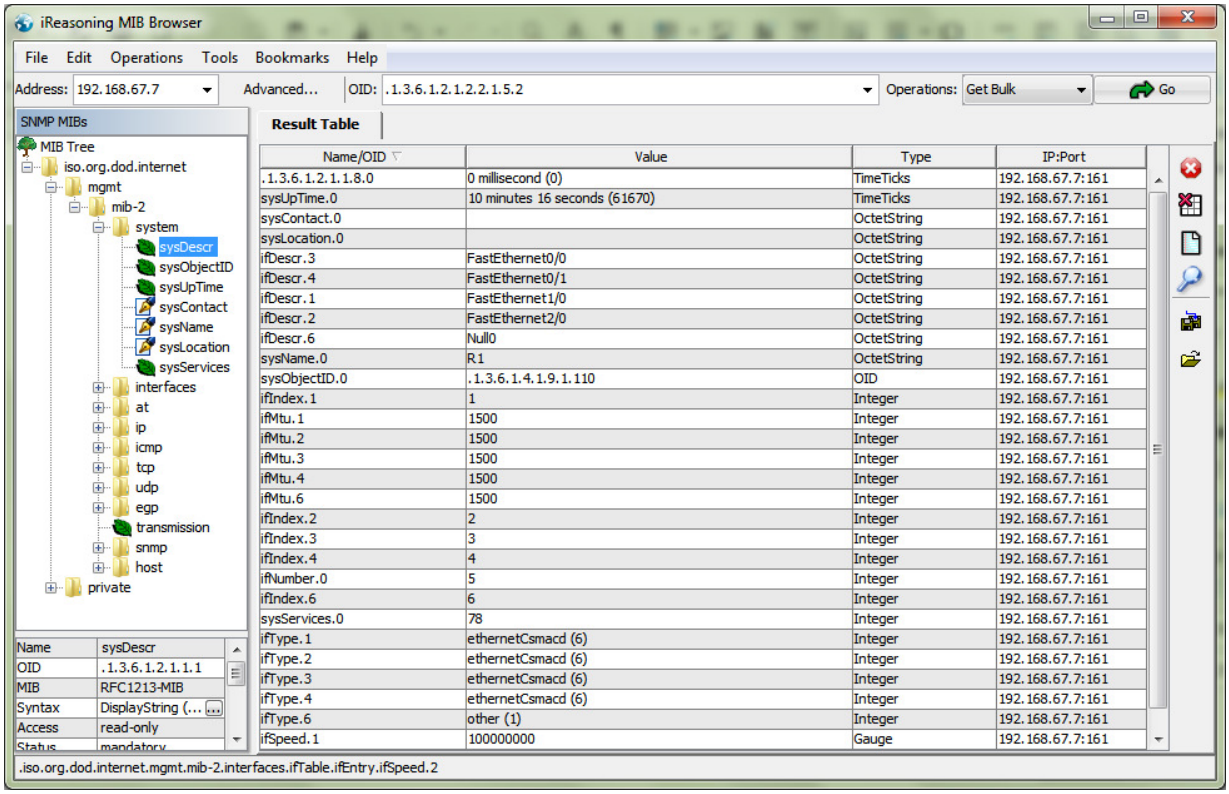


Слика 7.65 Конфигурација права приступа за SNMPv1

У оквиру алата је потребно конфигурисати адресу агента и исту вредност која је конфигурисана на рутеру и након тога је могуће прочитати све SNMP променљиве рутера (слика 7.65). Пример очитавања променљивих је дат на слици 7.66.

Да би се очитала нека променљива потребно је дефинисати њен OID и операцију преузумања, а да би се лакше одредило шта је улога одређене променљиве и који је податак у њој, на левој страни екрана је стабло и опис сваке од променљивих. На сличан начин је могуће очитавати SNMP променљиве и у сложенијим топологијама, када је потребно конфигурисати SNMP на свим рутерима, као и омогућити да рачунар буде повезан на IP нивоу са сваким од њих.

55 http://ireasoning.com/mibbrowser.shtml



Слика 7.66 Очитавање SNMP променљивих у MIB прегледачу

CIP - Каталогизација у публикацији - Народна библиотека Србије, Београд

004.7(075.8)(0.034.2)

ВУЛЕТИЋ, Павле, 1972-

Рачунарске мреже 2 [Електронски извор] :

[електронски уџбеник] / Павле Вулетић. - Београд : Електротехнички факултет, 2018 (Београд : Електротехнички факултет). - 1 електронски оптички диск (CD-ROM) ; 12 cm

Системски захтеви: Нису наведени. - Насл. са насловне стране документа. - Тираж 50. - Библиографија уз свако поглавље.

ISBN 978-86-7225-067-1

а) Рачунарске мреже

COBISS.SR-ID 260840204