ვიტალი აივაზოვი

დინამიკური მარშრუტიზაციის ალგორითმების შემუშავება IP ქსელებში პროგრამული ნაკადების გათვალისწინებით

წარდგენილია დოქტორის აკადემიური ხარისხის მოსაპოვებლად

საქართველოს ტექნიკური უნივერსიტეტი
თბილისი, 0175, საქართველო
2013

საქართველოს ტექნიკური უნივერსიტეტი

ინფორმატიკისა და მართვის სისტემების ფაკულტეტი

ჩვენ ქვემორე ხელისმომწერნი ვადასტურებთ, რომ გავეცანით "ვიტალი აივაზოვის" მიერ შესრულებულ სადისერტაციო ნაშრომს დასახელებით: "დინამიკური მარშრუტიზაციის ალგორითმების შემუშავება IP ქსელებში პროგრამული ნაკადების გათვალისწინებით" და ვაძლევთ რეკომენდაციას საქართველოს ტექნიკური უნივერსიტეტის "ინფორმატიკისა და მართვის სისტემების ფაკულტეტის" სადისერტაციო საბჭოში მის განხილვას დოქტორის აკადემიური ხარისხის მოსაპოვებლად.

ხელმძღვანელი: სრული პროფ. ტ.მ.დ. რომან სამხარაძე

რეცენზენტი: ასოც. პროფ. ტ.მ.კ. მზია კიკნაძე

რეცენზენტი: ასოც. პროფ. ტ.მ.კ. მარინა ქურდაძე

საქართველოს ტექნიკური უნივერსიტეტი

2013

ავტორი: ვიტალი აივაზოვი

დასახელება: "დინამიკური მარშრუტიზაციის ალგორითმების შემუშავება IP ქსელებში პროგრამული ნაკადების გათვალისწინებით"

ფაკულტეტი: "ინფორმატიკისა და მართვის სისტემების ფაკულტეტი"

აკადემიური ხარისხი: დოქტორი

სხდომა ჩატარდა: 2013 წლის 25 ივნისს, 14:00 საათზე

დისერტაციას ვუძღვნი

ამ ნაშრომს ვუძღვნი ჩემს ოჯახს უდიდესი მხარდაჭერისა და გვერდში დგომისათვის.

რეზიუმე

სადისერტაციო ნაშრომის "დინამიკური მარშრუტიზაციის ალგორითმების შემუშავება IP ქსელებში პროგრამული ნაკადების გათვალისწინებით", ძირითად მიზანს წარმოადგენს კომპიუტერულ ქსელებში მომსახურების ხარისხის ამაღლების და კონტროლის ალგორითმების შემუშავება.

მოცემული ნაშრომის მეცნიერულ სიახლეს წარმოადგენს ქსელის მომსახურების ხარისხის ფუნდამენტალური მახასიათებლების გაუარესების შემთხვევაში აპლიკაციების* მუშაობის მაღალი წარმადობის შენარჩუნება. შემუშავებულია მეთოდი, რომლის თანახმად არსებული სიტუაციიდან გამომდინარე ხორციელდება აპლიკაციების ნაკადების კრიტიკული მახასიათებლების მიხედვით ჯგუფებად დაყოფა. ამ ჯგუფებს ენიჭებათ პრიორიტეტები და სრულდება ამ ჯგუფებში მოქცეული მომსახურების ხარისხის მახასიათებლების გაზომვა ალტერნატიული მარშრუტების გამოყენებით. თითოეული ჯგუფისათვის გაზომილი მახასიათებლის ქვედა ზღვარის მიღწევისას, ხორციელდება ჯგუფში მოქცეული აპლიკაციების ნაკადების გადართვა ალტერნატიულ მარშრუტზე. სისტემა ღებულობს ამ გადაწყვეტილებას ჩატარებული გაზომვების შედეგების საფუძველზე. ასეთი მიდგომა უზრუნველყოფს კრიტიკული პარამეტრების მუდმივ კონტროლს, რაც თავის მხრივ იძლევა ეფექტური გადაწყვეტილების სწრაფად მიღების შესაძლებლობას. ამის შედეგად, მკვეთრად უმჯობესდება აპლიკაციების მუშაობის ხარისხი.

პირველ თავში ჩატარებულია კლასიკური მარშრუტიზაციის პრინციპების მიმოხილვა და კრიტიკული ანალიზი. ნაჩვენებია, რომ კომპიუტერულ ქსელებში გარკვეულ შემთხვევებში მომსახურების ხარისხთან დაკავშირებული პრობლემების გადასაწყვეტად აღარ არის ეფექტური არსებული კლასიკური მეთოდების გამოყენება. ამიტომ, აუცილებელია ახალი მიდგომის შემუშავება, რომელიც ინდივიდუალურად განიხილავს ქსელში მომუშავე ყველა აპლიკაციების ნაკადებს.

დასაბუთებულია კომპიუტერულ ქსელებში ინტელექტუალური მარშრუტიზაციის მეთოდების გამოყენების აუცილებლობა. ჩამოყალიბებულია ინტელექტუალური მარშრუტიზაციის მიერ შესრულებული ფუნქციები. ესენია: აპლიკაციების ნაკადების იდენტიფიცირება, ხელმისაწვდომი მარშრუტების ფუნდამენტალური მახასიათებლების გაზომვა, მონაცემების შეგროვება, მათი ანალიზი, აპლიკაციების მოთხოვნის დადგენა. ნაჩვენებია, რომ ინტელექტუალური მარშრუტიზაციის გამოყენების სფერო ფართოა და მეტად აქტუალური.

განხილულია კლასიკური მარშრუტიზაციის სტატიკური მეთოდი სხვადასხვა ოპერაციული სისტემების გამოყენებით, კერძოდ Linux და Cisco IOS მაგალითზე. ჩატარებულია სტატიკური მარშრუტიზაციის პრინციპების და სპეციფიკის ანალიზი. ნაჩვენებია, ამ მეთოდის ნაკლოვანობები და მათი ზემოქმედება მარშრუტიზაციის დინამიკაზე. განხილულია სტატიკური მეთოდის დადებითი მხარეებიც.

ჩატარებულია წესებზე დაფუძნებული მარშრუტიზაციის (წდმ) პრინციპების მიმოხილვა და კრიტიკული ანალიზი. ნაჩვენებია, რომ წდმ მეთოდის გამოყენების დროს შესაძლებელია აპლიკაციების ნაკადების მრაფალფეროვანი პარამეტრით იდენტიფიცირება. განხილულია წდმ–ს გამოყენებით პაკეტების მარკირების მეთოდებიც. იმის გამო, რომ წდმ და სტატიკური მარშრუტიზაცია არ წარმოადგენენ დინამიკურ მეთოდებს, მათი გამოყენების შემთხვევაში შეუძლებელია ქსელში მომუშავე აპლიკაციების ნაკადების გადაცემის ხარისხის კონტროლი. მაშასადამე, მეტად მნიშვნელოვანია ამ პრობლემის გადაწყვეტის გზების ძებნა და შესაბამისი მოდელებისა და ალგორითმების შემუშავება.

მეორე თავში აგებულია ინტელექტუალური მარშრუტიზაციის მოდელი. განხილულია მომსახურების ხარისხის ფუნდამენტალური მახასიათებლების კონტროლის მეთოდები. დადგენილია, რომ არსებული დინამიკური მარშრუტიზაციის პროტოკოლები დანიშნულების მისამართისათვის ოპტიმალური მარშრუტის არჩევაში არიან შეზღუდული.

ჩამოყალიბებულია დღევანდელი გლობალური ქსელების პრობლემები და შემუშავებულია ამ პრობლემების გადაწყვეტის მეთოდები. გაანალიზებულია თითოეული მათგანი.

შემუშავებულია ალგორითმი, რომელსაც საფუძვლად უდევს ძირითადი მოთხოვნა, რომ მარშრუტიზატორებს უნდა შეეძლოთ ნაკადების მარშრუტიზაცია არა მარტო დანიშნულების პრეფიქსების მიხედვით, არამედ აპლიკაციების ტიპებზე დაყრდნობით. მარშრუტიზაცია ირჩევს აპლიკაციისათვის საუკეთესო მარშრუტს და არა მხოლოდ დანიშნულების პრეფიქსისათვის. კრიტიკული აპლიკაციების მარშრუტიზაცია ხორციელდება ძირითადი (მაღალხარისხიანი) არხის გამოყენებით, რაც ხელს უწყობს აპლიკაციების მაღალი ხარისხის უზრუნველყოფას. ამავე დროს, დანარჩენი სტანდარტული აპლიკაციების ნაკადები იყენებენ სარეზერვო არხებს. საჭიროა აღინიშნოს, რომ ძირითადი არხის დაზიანების შემთხვევაში, სისტემას შეუძლია კრიტიკული აპლიკაციების ნაკადების გადამისამართება სარეზერვო არხებისაკენ.

ამის შედეგად, ქსელში უეცარი ცვლილებების შემთხვევაში, მარშრუტიზატორები ახდენენ სწრაფ რეაგირებას და ხორციელდება ნაკადების გადანაწილება სასურველი მიმართულებებით.

შემუშავებული არლგორითმი იყენებს ტრაფიკის მართვის ინტელექტუალურ მეთოდს, რომელსაც შეუძლია დინამიკური მარშრუტიზაციის გადაწყვეტილებების მიღება ქსელის არხების შეცდომებზე დაყრდნობით და ასევე მარშრუტიზაციის კორექტირება კომპლექსურ კრიტერიუმების გათვალისწინებით. ასეთ კრიტერიუმებს შეიძლება წარმოადგენდნენ პაკეტების მიწოდების დაყოვნება, პაკეტების დანაკარგი, დაყოვნების ვარიაციის ფაქტორი, არხების დატვირთულობა და სხვა წესები. ინტელექტუალური მარშრუტიზაციის პროტოკოლი აუმჯობესებს ტრადიციულ მარშრუტიზაციას (BGP, OSPF, და წდმ) წარმადობაზე გავლენის მქონე პარამეტრების რეალურ რეჟიმში კონტროლით.

შემუშავებული მოდელების საფუძველზე აგებულია ალგორითმი, რომელიც ითვალისწინებს არსებული მარშრუტების მომსახურების ხარისხის მახასიათებლების მნიშვნელობებს და ახორციელებს ნაკადების ჯგუფებად დაყოფას. ჯგუფების შექმნა ხორციელდება აპლიკაციების პრიორიტეტული მახასიათებლის მიხედვით.

მესამე თავში განხილულია ინტელექტუალური მარშრუტიზაციის ალგორითმის შემადგენელი ნაწილების პროტოკოლში ინტეგრირება და მათი მუშაობის პრინციპის ანალიზი. დასაბუთებულია ქსელის მომსახურების ხარისხის მახასიათებლების საზომი ხელსაწყოების შესაძლებლობები და მათი გამოყენების ასპექტები გაზომვის ფაზაში. შემუშავებულია ამ ხელსაწყოების გამოყენების მეთოდები და ალგორითმი. ამ ალგორითმის ნაწილს წარმოადგენენ NetFlow და მომსახურების ხარისხის მახასიათებლების გაზომვის სხვადასხვა ხელსაწყოები.

ჩატარებულია ინტელექტუალური მარშრუტიზაციის ალგორითმის აპრობაცია ქსელის სიმულატორის ბაზაზე.

შემუშავებულია აპლიკაციების ნაკადების იდენტიფიცირების და კლასიფიცირების მეთოდები. შემდეგ, განხილულია მომსახურების ხარისხის მახასიათებლების კონტროლის ალგორითმების მუშაობის შედეგები. წარმოდგენილია მათი გრაფიკული გამოხატულება.

დასკვნების ნაწილში შეჯამებულია სადისერტაციო ნაშრომის ძირითადი მიზნები და მიღებული შედეგები, თემის აქტუალურობა და შემუშავებული მეთოდების მეცნიერული სიახლე.

ABSTRACT

The main objective of the presented dissertation „Developing of an application based IP dynamic routing algorithms “ is a development of an algorithm, which is controlling network quality of service.

The scientific innovation of the presented work lies in maintaining high application performance in an event of degradation of the network fundamental QoS characteristics. A method has been developed, that performs grouping of application flows per their critical parameters taking into account the current network QoS conditions. Different priorities are assigned to the QoS parameters that are included into groups and measurements of each characteristic are carried out via all available links. For each group, in the event of measured characteristic reaching the lowest threshold, the algorithm performs re-routing of the flows included into the group to an alternative route using other available interface. A system makes this decision based on the measurements performed. This kind of approach ensures a constant control of the critical characteristic, which in turn provides a capability of fast and effective decision making. As a result, the performance of the applications is dramatically increased.

In the first chapter, an overview and critical analisys of the classical routing is performed. Limited capabilities of existing classical methods have been discussed in resolving some QoS related problems. Therefore, it is necessary to develop a new method that will individually consider each application flow passing in the network.

The necessity of deployment of intelligent routing methods is highlighted. The functions executed by intelligent routing algorithm are defined. They include: identification of application flows, measurement of fundamental characteristics of available paths, collection of results, their analysis and identification of application requirements. Further, the actuality and area of usage of the intelligent routing are highlighted.

The static method of classical routing is analized on different operating systems, in particular using Linux and Cisco IOS. This is followed by reviewing the static routing principles and specifics. The influence of this method on the dynamics of routing is spotted. Further, the advantages and shortcomings of this method are underlined.

Later in the chapter, the policy based routing (PBR) methods are overviewed along with their critical analysis. The capabilities of PBR to identify different application flows with the help of various parameters are highlighted. Furthermore, packet marking thecniques of PBR are discussed and evaluated. Due to the fact, that PBR and static routing are not representing dynamic methods, it is not possible to control the quality of application flow transmission while using them. Therefore, it is highly important to seek for a solution to this problem and to develop corresponding methods and algorithms.

In the second chapter, the intelligent routing model is constructed. Various QoS characteristic measurement methods are described. It is evaluated, that existing dynamic routing protocols are limited in number of QoS characteristics that are taken into account in the best route selection process.

Modern global network problems are defined and each of them is analized. This is followed by the proposition of corresponding solution methods.

An algorithm has been developed, which is based on the main idea of routers being able to make routing decisions based not only on destination prefix, but also on the application flows. Therefore, critical applications are routed via a high performance path, whereas standard applications are directed towards an alternative path. It is important to mention, that in case of high performance interface failure, the algorithm is able to reroute the critical application traffic towards the alternative path.

As a result, in the moment of unexpected performance degradation, routers perform fast reaction and redirect traffic towards a desired path.

The developed algorithm uses intelligent traffic control mechanism that is able to make dynamic routing decisions based on the interface errors and complex QoS criteria. Such criteria can be RTT, packet delay, packet loss, PDV, interface load and other. The intelligent routing protocol improves classical routing by measuring performance characteristics in real time.

An algorithm has been created based on the developed model, which takes into consideration QoS characteristics of the available paths and according to that information performs grouping of flows.

The third chapter starts with description of the components of the intelligent routing algorithm, aspects of their integration into the model and analisys of their functionality. Capabilities of the network QoS characteristic measurement tools are constituted and their application in a scope of measurement phase are defined. This tool set includes various measurement utilities and Netflow statistics.

Probation of the intelligent routing algorithm is executed using a network simulator. The results of the quality of service characteristic measurements are presented along with their graphical representation.

The dissertation work is summarized with a conclusion part constituting the achieved results, the relevance of the project and the scientific innovation of the developed methods.

შინაარსი

ცხრილების ნუსხა

ნახაზების ნუსხა

გამოყენებული აბრევიატურების ნუსხა

(ტექსტში აღნიშნულია * ნიშნით)

აპლიკაციები – პროგრამული უზრუნველყოფა, რომელიც ურთიერთქმედებს მომხმარებელთან.

წდმ – წესებზე დაფუძნებული მარშრუტიზაცია (Policy Based Routing).

ჰოსტი – ქსელური კვანძი (host), მაგალითად პერსონალური კომპიუტერი.

ჰოპი - მარშრუტის კვანძი (hop), რომლის გავლა უწევს პაკეტს ადრესატამდე მიღწევისას.

ჰოპიდან–ჰოპამდე – პაკეტის მიმდევრობითი გადაგზავნა ერთი მარშრუტიზატორიდან შემდგომ მარშრუტიზატორამდე (hop-by-hop).

IP გადაგზავნა – მარშრუტიზატორის ფუნქცია, რომელიც ახორციელებს მიღებული პაკეტების გადაგზავნას შემდგომ მარშრუტიზატორზე ან ჰოსტზე (IP forwarding).

გამოტანილი ინფორმაცია – გაცემული ბრძანების შედეგად მიღებული ინფორმაცია (output).

კარიბჭე – მარშრუტიზატორი, რომლის მეშვეობით შესაძლებელია ლოკალური ქსელიდან გარეთ გახწევა და უკუმიმართულებით (gateway).

Default კარიბჭე - უკანასკნელი მიმართვის კარიბჭე.

ACL – მისაწვდომობის მართვის ნუსხა (Access Control List).

ავტონომიური სისტემა – ერთი ორგანიზაციის ფარგლებში ქსელების ერთობლიობა (AS).

პდვ – პაკეტების დაყოვნების ვარიაცია, PDV ასევე ცნობილი როგორც jitter.

RTT - ორი გზის დაყოვნება, Round-Trip Time.

Syslog რეფორთი – syslog სერვერის მოხსენების შეტყობინება.

მადლიერება

მადლობას ვუხდი ჩემს ხელმძღვანელს, სრულ პროფესორს ტ.მ.დ. რომან სამხარაძეს ნაშრომზე მუშაობის პერიოდში უამრავი გაწეული კონსულტაციებისა და ფასდაუდებელი დახმარებისათვის.

შესავალი

როგორც ცნობილია თანამედროვე კომპიუტერულ სისტემებში პროგრამების უდიდესი ნაწილი კავშირისათვის იყენებს ინტერნეტს. ასეთი პროგრამები შეგვიძლია დავყოთ რამდენიმე ჯგუფად კავშირის სპეციფიკის მიხედვით, მაგალითად ხმის გადამცემი პროგრამები გამოირჩევიან დაყოვნების მიმართ დიდი მგრძნობიარობით, მაგრამ ფაილების გადამცემ პროგრამებზე დაყოვნება ისეთ გავლენას არ ახდენს, როგორც მაღალი გამტარუნარიანობა.

თანამედროვე გლობალური ქსელების უდიდესი პრობლემა მდგომარეობს იმაში, რომ საუკეთესო მარშრუტი, რომელსაც ირჩევენ დღეს არსებული მარშრუტიზაციის პროტოკოლები არ არის ერთნაირად საუკეთესო OSI მოდელის ზედა შრეებზე მომუშავე ყველა აპლიკაციისათვის. მარშრუტიზაციის არსებული პროტოკოლები ირჩევენ ერთ საუკეთესო მარშრუტს ნაკადის ადრესატამდე და ამ მიმართულებით აგზავნიან ყველა ტიპის პროგრამის პაკეტებს. ეს პროცესი ხორციელდება წინასწარ დაკონფიგურირებული კრიტერიუმებით და გამოიყენება უნივერსალურად, პროგრამული ნაკადების სპეციფიკის გათვალისწინების გარეშე. ეს არის განპირობებული იმით, რომ არსებული პროტოკოლები ვერ ანსხვავებენ პროგრამების პაკეტების ტიპებს. შედეგად ვიღებთ იმას, რომ მარშრუტიზაციის პროტოკოლის მიერ არჩეული საუკეთესო მარშრუტი არ არის ერთნაირად საუკეთესო ყველა პროგრამისთვის და ამით ზოგიერთი პროგრამის კავშირის ხარისხი უარესდება.

ამ პრობლემის ყველაზე ეფექტურ გადაწყვეტას წარმოადგენს საუკეთესო გზის არჩევა თითოეული პროგრამული ნაკადისათვის ინდივიდუალურად მისი სპეციფიკის გათვალისწინებით, რის შედეგად ყველა პროგრამა მიიღებს მისთვის კრიტიკული პარამეტრის მიხედვით საუკეთესო გზას. საუკეთესო გზის არჩევის შემდეგ პროტოკოლი აგრძელებს

მარშრუტის მონიტორინგს რაც ხელს უწყობს პროგრამული ნაკადის გადაცემის მაღალ ხარისხს და ამ ხარისხის მუდმივ კონტროლს.

ზემოთ თქმულიდან გამომდინარე, მეტად აქტუალურია აღნიშნული პრობლემის გადაწყვეტის გზებისა და საშუალებების შემუშავება.

პირველ თავში ჩატარებულია კლასიკური მარშრუტიზაციის პრინციპების მიმოხილვა და კრიტიკული ანალიზი. ნაჩვენებია, რომ კომპიუტერულ ქსელებში გარკვეულ შემთხვევებში მომსახურების ხარისხთან დაკავშირებული პრობლემების გადასაწყვეტად აღარ არის ეფექტური არსებული კლასიკური მეთოდების გამოყენება. ამიტომ, აუცილებელია ახალი მიდგომის გამოყენება, რომელიც ინდივიდუალურად განიხილავს ქსელში მომუშავე ყველა აპლიკაციების ნაკადებს.

დასაბუთებულია კომპიუტერულ ქსელებში ინტელექტუალური მარშრუტიზაციის მეთოდების გამოყენების აუცილებლობა. ჩამოყალიბებულია ინტელექტუალური მარშრუტიზაციის მიერ შესრულებული ფუნქციები: აპლიკაციების ნაკადების იდენტიფიცირება, ხელმისაწვდომი მარშრუტების ფუნდამენტალური მახასიათებლების გაზომვა, მონაცემების შეგროვება, მათი ანალიზი და აპლიკაციების მოთხოვნის დადგენა. ნაჩვენებია, რომ ინტელექტუალური მარშრუტიზაციის გამოყენების სფერო ფართოა და საკმაოთ აქტუალური.

განხილულია კლასიკური მარშრუტიზაციის სტატიკური მეთოდი სხვადასხვა ოპერაციული სისტემების გამოყენებით, კერძოდ Linux და Cisco IOS მაგალითზე. ჩატარებულია სტატიკური მარშრუტიზაციის პრინციპების და სპეციფიკის ანალიზი. ნაჩვენებია ამ მეთოდის ნაკლოვანობები და მათი ზემოქმედება მარშრუტიზაციის დინამიკაზე. განხილულია სტატიკური მეთოდის დადებითი მხარეებიც.

ჩატარებულია წესებზე დაფუძნებული მარშრუტიზაციის (წდმ) პრინციპების მიმოხილვა და კრიტიკული ანალიზი. ნაჩვენებია, რომ წდმ–ს მეთოდის გამოყენების დროს შესაძლებელია აპლიკაციების ნაკადების მრავალფეროვანი პარამეტრებით იდენტიფიცირება. განხილულია წდმ–ს

გამოყენებით პაკეტების მარკირების მეთოდებიც. იმის გამო, რომ წდმ და სტატიკური მარშრუტიზაცია არ წარმოადგენენ დინამიკურ მეთოდებს, მათი გამოყენების შემთხვევაში შეუძლებელია ქსელში მომუშავე აპლიკაციების ნაკადების გადაცემის ხარისხის კონტროლი. მაშასადამე, მეტად მნიშვნელოვანია ამ პრობლემის გადაწყვეტის გზების ძებნა და შესაბამისი მოდელებისა და ალგორითმების შემუშავება.

აქედან გამომდინარე, კომპიუტერულ ქსელებში აპლიკაციების მუშაობის მაღალი ხარისხის უზრუნველყოფის პრობლემატიკაში, აუცილებელია შემდეგი ამოცანების გადაწყვეტა:

1. ქსელში მომუშავე აპლიკაციების ნაკადების იდენტიფიცირება და მათი ქსელის მომსახურების ხარისხის მიმართ მოთხოვნების გამოვლენა.
2. მომსახურების ხარისხის ფუნდამენტალური მახასიათებლების გაზომვის მოდელებისა და ალგორითმების შემუშავება.
3. შემუშავებული მოდელებისა და ალგორითმების ბაზაზე ინტელექტუალური მარშრუტიზაციის პროტოტიპის შემუშავება.

მეორე თავში ჩამოყალიბებულია დღევანდელი გლობალური ქსელების პრობლემები და შემუშავებულია ამ პრობლემების გადაწყვეტის მეთოდები. გაანალიზებულია თითოეული მათგანი. განხილულია მაღალი წარმადობის მქონე მარშრუტიზაციის 5 ფაზიანი მოდელი. აგებულია ინტელექტუალური მარშრუტიზაციის მოდელი და მოქმედების ალგორითმი. განხილულია მომსახურების ხარისხის ფუნდამენტალური მახასიათებლების კონტროლის მეთოდები. დადგენილია, რომ არსებული დინამიკური მარშრუტიზაციის პროტოკოლები დანიშნულების მისამართისათვის ოპტიმალური მარშრუტის არჩევაში არიან შეზღუდული.

შემუშავებულია ალგორითმი, რომელსაც საფუძვლად უდევს ძირითადი მოთხოვნა, რომ მარშრუტიზატორებს უნდა შეეძლოთ ნაკადების მარშრუტიზაცია არა მარტო დანიშნულების პრეფიქსების მიხედვით, არამედ აპლიკაციების ტიპებზე დაყრდნობით. მარშრუტიზაცია ირჩევს

აპლიკაციისათვის საუკეთესო მარშრუტს და არა მხოლოდ დანიშნულების პრეფიქსისათვის. კრიტიკული აპლიკაციების მარშრუტიზაცია ხორციელდება ძირითადი (მაღალხარისხიანი) არხის გამოყენებით, რაც ხელს უწყობს აპლიკაციების მაღალი ხარისხის უზრუნველყოფას. ამავე დროს, დანარჩენი სტანდარტული აპლიკაციების ნაკადები იყენებენ სარეზერვო არხებს. საჭიროა აღინიშნოს, რომ ძირითადი არხის დაზიანების შემთხვევაში, სისტემას შეუძლია კრიტიკული აპლიკაციების ნაკადების გადამისამართება სარეზერვო არხებისაკენ.

ამის შედეგად, ქსელში უეცარი ცვლილებების შემთხვევაში, მარშრუტიზატორები ახდენენ სწრაფ რეაგირებას და ხორციელდება ნაკადების გადანაწილება სასურველი მიმართულებებით.

შემუშავებული ალგორითმი იყენებს ტრაფიკის მართვის ინტელექტუალურ მეთოდს, რომელსაც შეუძლია დინამიკური მარშრუტიზაციის გადაწყვეტილებების მიღება ქსელის არხების შეცდომებზე დაყრდნობით და ასევე მარშრუტიზაციის კორექტირება კომპლექსურ კრიტერიუმების გათვალისწინებით. ასეთ კრიტერიუმებს შეიძლება წარმოადგენდნენ პაკეტების მიწოდების დაყოვნება, პაკეტების დანაკარგი, დაყოვნების ვარიაციის ფაქტორი, არხების დატვირთულობა და სხვა მითითებული წესები. ინტელექტუალური მარშრუტიზაციის პროტოკოლი აუმჯობესებს ტრადიციულ მარშრუტიზაციას (BGP, OSPF, და წდმ) წარმადობაზე გავლენის მქონე პარამეტრების რეალურ რეჟიმში კონტროლით.

შემუშავებული მოდელების საფუძველზე აგებულია ალგორითმი, რომელიც ითვალისწინებს არსებული მარშრუტების მომსახურების ხარისხის მახასიათებლების მნიშვნელობებს და ახორციელებს ნაკადების ჯგუფებად დაყოფას. ჯგუფების შექმნა ხორციელდება აპლიკაციების პრიორიტეტული მახასიათებლის გათვალისწინებით.

მესამე თავში განხილულია ინტელექტუალური მარშრუტიზაციის ალგორითმის შემადგენელი ნაწილების პროტოკოლში ინტეგრირება და

მათი მუშაობის პრინციპის ანალიზი. დასაბუთებულია ქსელის მომსახურების ხარისხის მახასიათებლების საზომი ხელსაწყოების შესაძლებლობები და მათი გამოყენების ასპექტები გაზომვის ფაზაში. შემუშავებულია ამ ხელსაწყოების გამოყენების მეთოდები და ალგორითმი. ამ ალგორითმის ნაწილს წარმოადგენენ NetFlow და მომსახურების ხარისხის მახასიათებლების გაზომვის სხვადასხვა ხელსაწყოები.

ჩატარებულია ინტელექტუალური მარშრუტიზაციის ალგორითმის აპრობაცია ქსელის სიმულატორის ბაზაზე.

შემუშავებულია აპლიკაციების ნაკადების იდენტიფიცირების და კლასიფიცირების მეთოდები. შემდეგ, განხილულია მომსახურების ხარისხის მახასიათებლების კონტროლის ალგორითმების შედეგები. წარმოდგენილია მათი გრაფიკული გამოხატულება.

დასკვნების ნაწილში შეჯამებულია სადისერტაციო ნაშრომის ძირითადი მიზნები და მიღებული შედეგები, თემის აქტუალურობა და შემუშავებული მეთოდების მეცნიერული სიახლე.

ლიტერატურული მიმოხილვა

თავი I

კლასიკური მარშრუტიზაციის პრინციპები და პროტოკოლები

ამ თავში არის მიმოხილული მარშრუტიზაციის ის პრინციპები და პროტოკოლები, რომლებიც არიან პირდაპირ კავშირში ინტელექტუალურ მარშრუტიზაციის ალგორითმთან. განხილულია მათი უპირატესობები და ნაკლოვანობები, ასევე მათი მუშაობის პრინციპები, რომლებიც პირდაპირ ზეგავლენას ახდენენ მოცემულ ნაშრომში დამუშავებულ ალგორითმებზე. პირველ ქვეთავში არის აღწერილი მარშრუტიზაციის ზოგადი პროცესი, რომლის მეშვეობით ხორციელდება IP პაკეტების მიწოდება წყაროდან დანიშნულების მისამართამდე. კერძოდ, განხილულია ჰოსტსა* და მარშრუტიზატორს შორის პაკეტის დამუშავების პრინციპების საბაზისო განსხვავება. ასევე დეტალურად არის აღწერილი მარშრუტიზაციის ცხრილის შედგენის პროცესი და სხვადასხვა პარამეტრების ფუნქციონალური გამოყენება.

პაკეტების მარშრუტიზაციის ალგორითმი, რომლის გამოყენებით ხორციელდება პაკეტების გადაგზავნა, წარმოადგენს ფუნდამენტალურ პროცესს და განხილულია მისი რეალიზაცია, როგორც Linux ასევე Cisco IOS ოპერაციულ სისტემებზე.

1.1 მარშრუტიზაციის პრინციპები

IP წარმოადგენს კვანძთაშორის პაკეტების მიწოდების მომსახურებას და უზრუნველყოფს მათ მიწოდებას საწყისი კვანძიდან დანიშნულების ადგილამდე. IP პაკეტების მიწოდების მომსახურება ხორციელდება მარშრუტიზატორების დახმარებით. მოცემულ ქვეთავში არის აღწერილი ამ მომსახურების საბაზისო პრინციპები.

1.1.1 IP გადაცემა და მარშრუტიზაცია

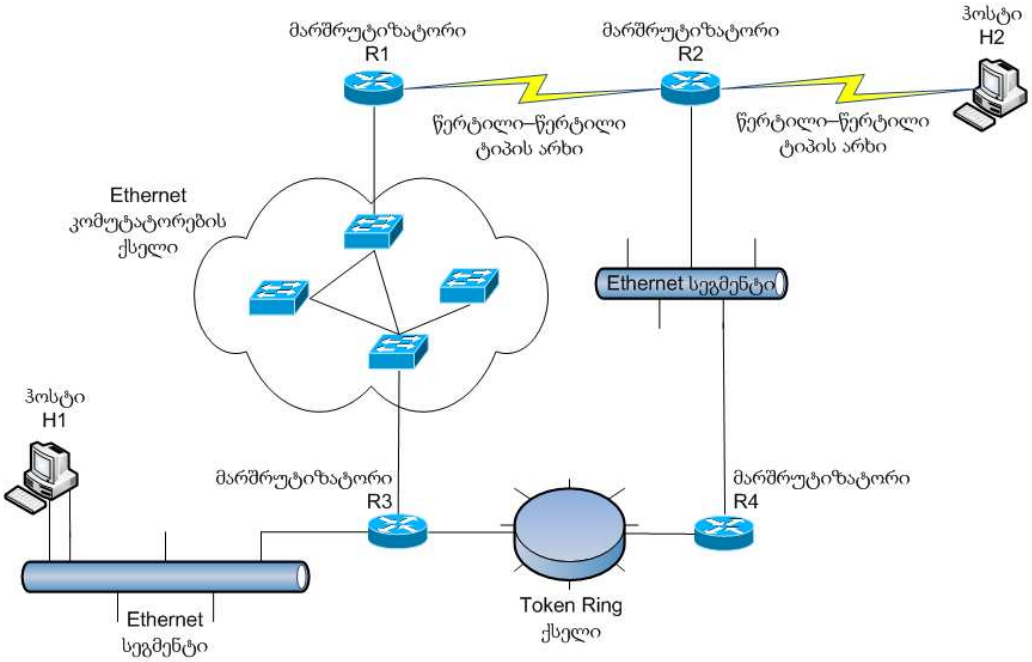
ინტერნეტი წარმოადგენს სხვადასხვა ქსელების ერთობლიობას. ყოველივე ქსელი შედგება ლოკალური ქსელისგან (LAN), კომუტირებული ქსელისაგან ან წერტილი–წერტილი ტიპის სეგმენტებისგან(point-to-point). ინტერნეტის IP პროტოკოლი უზრუნველყოფს ამ ერთეული ქსელების შერწყმას ერთიან გლობალურ ქსელში, სადაც მასთან შეერთებულ ნებისმიერ კვანძთა წყვილს შეუძლია პაკეტების მიმოცვლა. იმ სისტემებს, რომლებიც აკავშირებენ ქსელებს ერთმანეთთან, ეწოდებათ მარშრუტიზატორები. მარშრუტიზატორს გააჩნია ორი ან მეტი ქსელის ინტერფეისი, რომლებიც არიან შეერთებული სხვადასხვა ქსელებთან და ახორციელებენ პაკეტების გადაცემას ამ ქსელებს შორის.

ქსელების ერთობლიობის მაგალითი არის მოყვანილი ნახ.1.1. ამ ნახაზზე არის აღწერილი ორი ჰოსტი, ოთხი მარშრუტიზატორი და ექვსი არხის შრის ქსელი, რომლებიც მოიცავენ ორ Ethernet LAN–ს, ორ Token Ring LAN–ს, კომუტირებად Ethernet ქსელს და ორ წერტილი–წერტილი ტიპის არხებს. IP შრეზე ქსელის აბსტრაქცია განსხვავდება არხის შრისაგან. IP ქსელი განისაზღვრება ქსელის პრეფიქსით, მაგალითად 10.0.1.0/24. ერთ IP ქსელში გაერთიანებულ ჰოსტებს და მარშრუტიზატორებს გააჩნიათ ქსელის ინტერფეისები, რომლებიც იყენებენ IP მისამართებს ერთი და იგივე IP პრეფიქსიდან. ნახაზზე 1.2 წარმოდგენილია იგივე ქსელი რაც ნახ.1.1–ზე, მხოლოდ IP შრეზე. მასზე ილუსტრირებულია IP ქსელი, რომელიც აღნიშნულია ღრუბლის სახით, რაც მეტყველებს იმაზე, რომ IP-ის შრეზე აპარატურული სპეციფიკა არის უგულვებელყოფილი.

IP ქსელებს აქვთ მინიჭებული შემდეგი ქსელის პრეფიქსები: 10.1.0.0/24, 10.1.2.0/24, 10.2.1.0/24, 10.3.0.0/16, 20.1.0.0/16 და 20.2.1.0/28. ყოველივე პრეფიქსი განსაზღვრავს IP მისამართების დიაპაზონს. მაგალითად, 20.1.0.0/16 ქსელის IP მისამართების დიაპაზონი შეზღუდულია 20.1.0.0 – 20.1.255.255 –ით. მისამართი 20.1.0.0 წარმოადგენს IP ქსელის მისამართს, მაშინ როდესაც მისამართი 20.1.255.255 წარმოადგენს

მაუწყებლობის მისამართს(broadcast), ე.ი. ყველა ჰოსტი ქსელში. ამ დიაპაზონის დანარჩენი მისამართები შეიძლება იყვნენ დაკონფიგურირებული ქსელის ინტერფეისებზე.

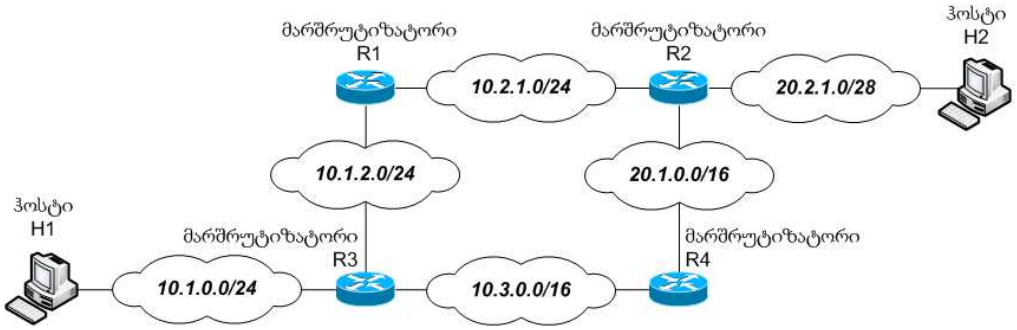
ხშირ შემთხვევებში, ყველა არხის შრის ქსელი შეესაბამება ერთ IP ქსელს, ამიტომ ტერმინი „ქსელი“ არის დამოკიდებული კონტექსტზე.



ნახ. 1.1 ქსელის ხედვა OSI მოდელის არხის შრეზე

ტერმინი ქსელი შეიძლება იყოს გამოყენებული როგორც არხის შრის ქსელის შემთხვევაში (ნახ. 1.1), ასევე IP ქსელი მიმართებაში (ნახ. 1.2). აუცილებელია აღინიშნოს, რომ ეს ორი შემთხვევა არის საკმაოდ განსხვავებული. არხის შრის ქსელი განიზასღვრება ქსელის აპარატურით, კერძოდ Ethernet კომუტატორებით, რომლებიც აკავშირებენ ცალკეულ Ethernet ინტერფეისების. IP ქსელის შემთხვევაში, ტერმინი „ქსელი“ განისაზღვრება ლოგიკური სეგმენტით, რომელიც შემოსაზღვრულია ცალკეული IP მისამართების დიაპაზონით. ამ განსხვავების გამოსაკვეთად, არხის შრის ქსელის მიმართ ხშირად გამოიყენება ტერმინი „ფიზიკური ქსელი“, მხოლოდ IP ქსელის მიმართ გამოიყენება ტერმინი „ლოგიკური ქსელი“.

როდესაც საუბარია იმაზე, თუ როგორ გადაეცემა პაკეტები ინტერნეტში, აუცილებელია აღინიშნოს, რომ IP შრეზე ჰოსტებს და მარშრუტიზატორებს გააჩნიათ ქსელის ისეთი ხედვა, როგორიც მოცემულია ნახაზზე 1.2 და არა ისეთი, როგორიც არის ნახაზზე 1.1. პაკეტების გადასაგზავნად, ჰოსტები და მარშრუტიზატორები იყენებენ მხოლოდ ქსელის პრეფიქსებს.



ნახ. 1.2 ქსელის ხედვა OSI მოდელის ქსელის შრეზე

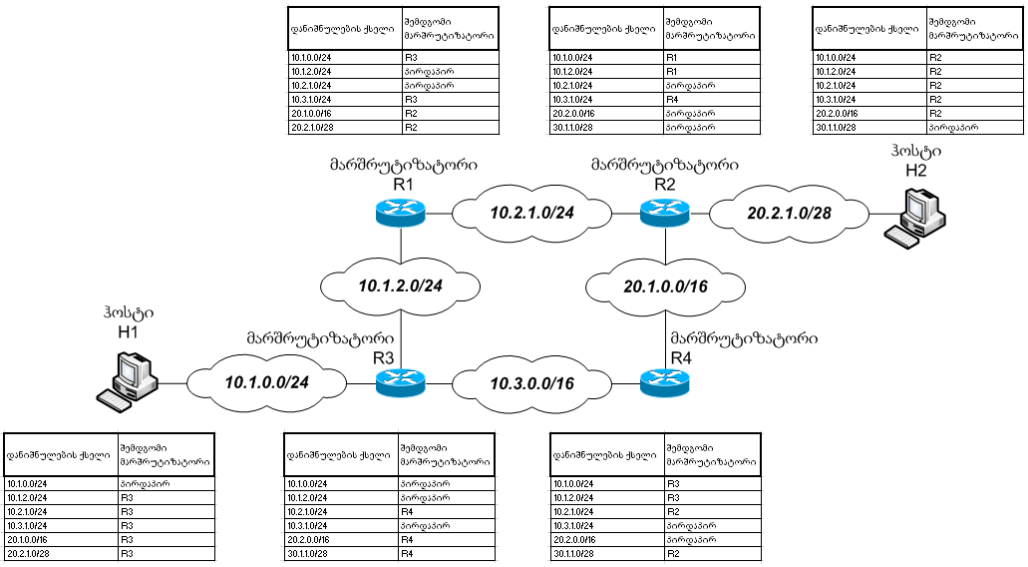
ორ კვანძს შორის IP პაკეტების მისაწოდებლად, არხის შრის ქსელებსა და IP ქსელებს შორის ურთიერთკავშრი უნდა აკმაყოფილებდეს განსაზღვრულ პირობებს. პირველი პირობა მდგომარეობს იმაში, რომ დანიშნულების IP მისამართის ქსელის პრეფიქსი უნდა შეესაბამებოდეს არხის შრის უნიკალურ ქსელს ინტერნეტში. მაშინ, როდესაც უკუპირობა არ არის აუცილებელი, კერძოდ ერთ ფიზიკურ ქსელს შეიძლება შეესაბამებოდეს რამოდენიმე IP ქსელი. მაგალითად, ნახ. 1.3–ზე არის მოცემული ქსელი, რომელშიც ჰოსტები არიან დაკავშირებული ერთი Ethernet სეგმენტით, მაგრამ მიეკუთვნებიან სხვადასხვა IP ქსელებს. ეს პირობა არის მიღწეული ქვექსელების მინიჭების მეშვეობით. მეორე პირობა მდგომარეობს იმაში, რომ მარშრუტიზატორებს და ჰოსტებს, რომლებიც არიან დაკავშირებული ერთი IP პრეფიქსით, უნდა შეეძლოთ IP პაკეტების მიმოცვლა არხის შრის პროტოკოლის მეშვეობით, ე.ი. Ethernet კადრების მეშვეობით. მესამე პირობა განსაზღვრავს იმას, რომ არხის შრის ყველა ქსელს უნდა გააჩნდეს სულ მცირე ერთი მარშრუტიზატორი და ეს მარშრუტიზატორი უნდა იყოს დაკავშირებული სულ მცირე არხის შრის

ერთ სხვა ქსელთან. თუ კი ზემოთ მოყვანილი პირობებიდან ერთი მაინც არ არის შესრულებული, მაშინ ქსელთაშორის პაკეტების მიმოცვლა ვერ მოხერხდება.

მარშრუტიზაციის ცხრილები წარმოადგენენ IP პაკეტების გადაცემის პროცესის უმნიშვნელოვანეს ნაწილს და გამოიყენებიან როგორც მარშრუტიზატორებში ასევე ჰოსტებში. მარშრუტიზაციის ცხრილი მიუთითებს ჰოსტს ან მარშრუტიზატორს იმას, თუ სად უნდა იყოს გადაგზავნილი პაკეტი. ყოველი პაკეტის გაგზავნისათვის ხორციელდება მარშრუტიზაციის ცხრილის შემოწმება. აგრეთვე არსებობს სხვადასხვა ქეშირების მეთოდი, რომლებიც ამცირებენ მარშრუტიზაციის ცხრილის შემოწმების სიხშირეს, რაც საგრძნობლად აუმჯობესებს პაკეტების გადაგზავნის პროცესს. ყველა დანიშნულების IP მისამართისათვის ხორციელდება მარშრუტიზაციის შემოწმება და იმ შემთხვევაში თუ კი დანიშნულების მისამართი მდებარეობს იმავე ქსელში რაც წყარო, ითვლება რომ პაკეტი არის ლოკალურად მიწოდებადი. ამავე დროს, თუ კი მარშრუტიზაციის ცხრილის შემოწმების შედეგან მიღებულია შემდგომი მარშრუტიზატორის მისამართი, ეს იმას ნიშნავს, რომ პაკეტები უნდა იყვნენ გადაგზავნილი ამ მარშრუტიზატორისაკენ. ნახაზზე 3.2 მოცემულია ჰოსტი A და მარშრუტიზატორი R1, რომლებსაც შეუძლიათ პაკეტების პირდაპირი გადაგზავნა. ჰოსტი H1-ისათვის არსებობს მხოლოდ ერთი შემდგომი მარშრუტიზატორი, რომელიც არის R3. მარშრუტიზატორ R1-ს გააჩნია ორი შესაძლო შემდგომი მარშრუტიზატორი, რომლებიც არიან R2 და R3.

როდესაც მარშრუტიზაციის ცხრილი მიუთითებს იმას, რომ პაკეტის მიწოდება შეუძლებელია პირდაპირი სახით, IP პაკეტი ენკაპსულირდება არხის შრის კადრში და გადაიგზავნება შემდგომი მარშრუტიზატორისაკენ. როდესაც ეს IP პაკეტი მიეწოდება მიმღებ მარშრუტიზატორს, კვლავ ხორციელდება მარშრუტიზაციის ცხრილის შემოწმება, რის შემდეგ ეს პაკეტი გადაიგზავნება მიმღები მარშრუტიზატორიდან შემდგომი მარშრუტიზატორის მიმართულებით. ასეთი სახით ხორციელდება პაკეტის

გადაგზავნა ქსელიდან ქსელში და ეს პროცესი გრძელდება იქამდე, სანამ პაკეტი არ იქნება მიღებული იმ მარშრუტიზატორზე, რომელსაც შეუძლია პაკეტის პირდაპირი მიწოდება. პაკეტის ამგვარი მიწოდების სქემით, ყოველ მარშრუტიზატორზე ხორციელდება მარშრუტიზაციის ადგილობრივი გადაწყვეტილების მიღება და მათ არ გააჩნიათ ინფორმაცია სრული მარშრუტის შესახებ.



დანიშნულების ქსელი	შემდგომი მარშრუტიზატორი
10.1.0.0/24	R3
10.1.2.0/24	პირდაპირ
10.2.1.0/24	პირდაპირ
10.3.1.0/24	R3
20.1.0.0/16	R2
20.2.1.0/28	R2

დანიშნულების ქსელი	შემდგომი მარშრუტიზატორი
10.1.0.0/24	R1
10.1.2.0/24	R1
10.2.1.0/24	პირდაპირ
10.3.1.0/24	R4
20.2.0.0/16	პირდაპირ
30.1.1.0/28	პირდაპირ

დანიშნულების ქსელი	შემდგომი მარშრუტიზატორი
10.1.0.0/24	R2
10.1.2.0/24	R2
10.2.1.0/24	R2
10.3.1.0/24	R2
20.2.0.0/16	R2
30.1.1.0/28	პირდაპირ

დანიშნულების ქსელი	შემდგომი მარშრუტიზატორი
10.1.0.0/24	პირდაპირ
10.1.2.0/24	R3
10.2.1.0/24	R3
10.3.1.0/24	R3
20.1.0.0/16	R3
20.2.1.0/28	R3

დანიშნულების ქსელი	შემდგომი მარშრუტიზატორი
10.1.0.0/24	პირდაპირ
10.1.2.0/24	პირდაპირ
10.2.1.0/24	R4
10.3.1.0/24	პირდაპირ
20.2.0.0/16	R4
30.1.1.0/28	R4

დანიშნულების ქსელი	შემდგომი მარშრუტიზატორი
10.1.0.0/24	R3
10.1.2.0/24	R3
10.2.1.0/24	R2
10.3.1.0/24	პირდაპირ
20.2.0.0/16	პირდაპირ
30.1.1.0/28	R2

ნახ. 1.3 პაკეტის გადაგზავნა ჰოპიდან–ჰოპამდე

ნახაზზე 1.3 წარმოდგენილია პაკეტის ჰოპიდან–ჰოპამდე* გადაგზავნის სქემა და აგრეთვე მარშრუტიზაციის ცხრილების შიგთავსი. ქსელის ტოპოლოგია არის იგივე, რაც ნახ. 1.2–ზე, მაგრამ შეიცავს ასევე მარშრუტიზაციის ცხრილებს. ნახაზზე 1.3 დანიშნულების მისამართები არიან წარმოდგენილი ქსელის პრეფიქსის სახით. შემდგომი ჰოპის* ველი შეიცავს ორ შესაძლო მნიშვნელობას და ესენი არიან „შემდგომი მარშრუტიზატორის ჰოსტის სახელი“ ან „პირდაპირი მიწოდება“. უკანასკნელი მიუთითებს იმაზე, რომ დანიშნულების IP მისამართი მიეკუთვნება ქსელის პრეფიქს, რომელიც შეიძლება იყოს მიწოდებული პირდაპირი სახით. რეალური მარშრუტიზაციის ცხრილში, შემდგომი მარშრუტიზატორის ჰოსტის სახელის მაგივრად მითითებულია მისი IP მისამართი და ასევე ინტერფეისის ინდექსი, რომლის მეშვეობით

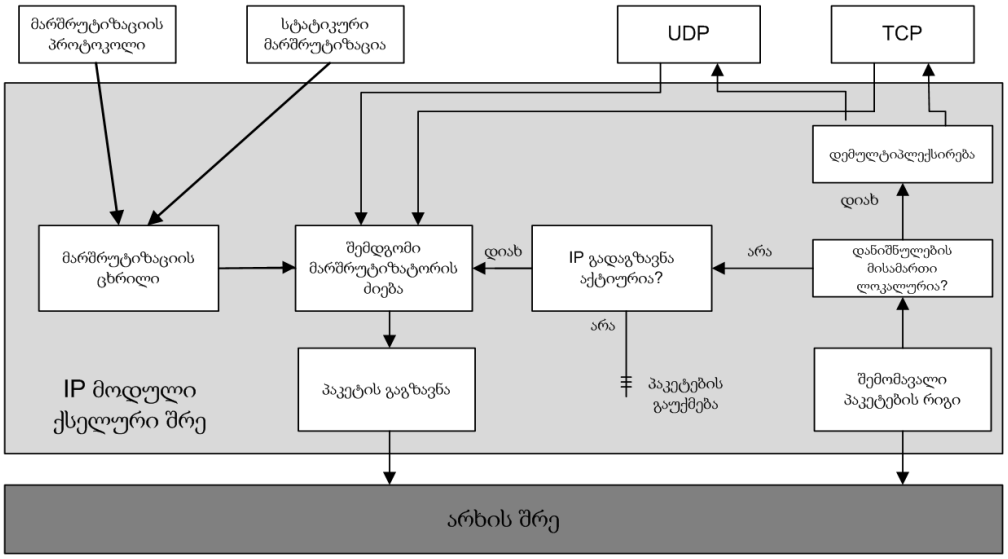
შესაძლებელია პაკეტის პირდაპირი მიწოდება. დავუშვათ, რომ ჰოსტი H1 უგზავნის IP პაკეტს ჰოსტს H2, სადაც დანიშნულების IP მისამართი არის 20.2.1.2. ჰოსტი H1–ზე მარშრუტიზაციის ცხრილის შემოწმების შედეგად მიღებული ინფორმაცია მიუთითებს იმაზე, რომ ამ კერძო დანიშნულების ქსელისათვის არსებობს შემდგომი მარშრუტიზატორი. ამ მარშრუტიზატორს წარმოადგენს R3. შესაბამისად ჰოსტი H1 გადასცემს პაკეტს მარშრუტიზატორი R3-ის IP მისამართზე. H2–ისაკენ მარშრუტის შესამოწმებლად, მარშრუტიზატორ R3–ზე ხორციელდება მარშრუტიზაციის ცხრილის შემოწმება, რომელიც მიუთითებს მარშრუტიზატორ R4-ისაკენ. როდესაც მარშრუტიზატორი R4 ღებულობს ამ პაკეტს, ის ამოწმებს საკუთარ მარშრუტიზაციის ცხრილს და მიმართავს პაკეტს R2-ის მისამართისაკენ. R2-ის მარშრუტიზაციის ცხრილში 20.2.1.0/28 პრეფიქსის ჩანაწერი მიუთითებს იმაზე, რომ პაკეტების მიწოდება შესაძლებელია პირდაპირი სახით. ამიტომ მარშრუტიზატორი R2 პირდაპირ უგზავნის პაკეტს ჰოსტს H2.

მარშრუტიზატორების მიერ IP პაკეტების მიწოდების პროცესი მოიცავს ორ ნაწილს. პირველი ნაწილი მოიცავს პაკეტის მიღებას, ამ პაკეტის დანიშნულების მისამართის მოძიებას მარშრუტიზაციის ცხრილში და მის გადაგზავნას შემდგომი მარშრუტიზატორისაკენ. ამ პროცესს ეწოდება IP გადაგზავნა* (IP forwarding). მეორე ნაწილი მოიცავს მარშრუტების წონის გამოთვლას და ამის საფუძველზე მარშრუტიზაციის ცხრილების შედგენას. ამ პროცესს ეწოდება მარშრუტიზაცია. მარშრუტიზაცია შეიძლება იყოს განხორციელებული როგორც სტატიკურად ასევე დინამიკურად. სტატიკური მარშრუტიზაციის შემთხვევაში, მარშრუტიზაციის ცხრილის ჩანაწერების შედგენა ხორციელდება ქსელის ადმინისტრატორის მიერ. დინამიკური მარშრუტიზაციის შემთხვევაში, მარშრუტის გამოთვლა და ცხრილის შედგენა ხორციელდება მარშრუტიზაციის პროტოკოლის მეშვეობით. მარშრუტიზაციის პროტოკოლები იყენებენ განაწილებულ ალგორითმს, რომელიც იმის საშუალებას იძლევა, რომ მარშრუტიზატორებს

შორის მოხერხდეს მარშრუტების ინფორმაციის მიმოცვლა და ჰოსტებს შორის საუკეთესო მარშრუტის გამოთვლა, ხშირ შემთხვევაში „უმოკლესი გზის“ ალგორითმის გამოყენებით. [1]

ჰოსტებზე და მარშრუტიზატორებზე IP პაკეტების დამუშავება წარმოადგენს იდენტურ პროცესს ერთი განსხვავებით, რომ მარშრუტიზატორები ახორციელებენ პაკეტების გადაგზავნას, მაშინ, როდესაც ჰოსტებზე ეს ფუნქცია მიუწვდომელია. ამ ფუნქციის არარსებობა ნიშნავს იმას, რომ ჰოსტს შეუძლია მხოლოდ ლოკალურად შექმნილი პაკეტების გადაგზავნა. ჰოსტს არ შეუძლია იმ პაკეტების გადაგზავნა, რომლებიც მან ქსელიდან მიიღო. IP მოდულში პაკეტების დამუშავების პროცესი არის აღწერილი ნახაზზე 1.4.

როდესაც იგზავნება IP პაკეტი, IP მოდული ახორციელებს მარშრუტიზაციის ცხრილის შემოწმებას იმისათვის, რომ დადგინდეს შემდგომი მარშრუტიზატორის მისამართი და პაკეტი გადაიგზავნოს ცხრილში მითითებულ მისამართზე/ინტერფეისზე.



ნახ. 1.4 IP პაკეტების დამუშავების პროცესი

როდესაც მარშრუტიზატორი ან ჰოსტი ღებულობს IP პაკეტს, IP მოდული ამოწმებს პაკეტის დანიშნულების მისამართს. თუ კი ის ემთხვევა ლოკალურ სისტემას, IP პაკეტი დემულტიპლექსირდება და გადაეცემა OSI

მოდელის ზედა შრის პროტოკოლს, როგორიც არის TCP ან UDP. იმ შემთხვევაში, თუ პაკეტის დანიშნულების მისამართს არ წარმოადგენს ლოკალური სისტემა, პროცესი ამოწმებს IP გადაგზავნის ფუნქციის მდგომარეობას. თუ კი ის ჩართულია, ეს იმაზე მიუთითებს, რომ ლოკალური სისტემა წარმოადგენს მარშრუტიზატორს და აუცილებელია მარშრუტიზაციის ცხრილის შემოწმება, რის შემდეგ პაკეტი უნდა გადაიგზავნოს შემდგომ მარშრუტიზატორზე ან პირდაპირ დანიშნულების ჰოსტზე. იმ შემთხვევაში, თუ IP გადაგზავნის ფუნქცია არ არის ჩართული, სისტემა წარმოადგენს ჰოსტს და მიღებული პაკეტი უქმდება. მაშასადამე, მარშრუტიზატორსა და ჰოსტს შორის ეს მცირე, მაგრამ კრიტიკული განსხვავება განაპირობებს იმას, რომ თუ პაკეტის დანიშნულების მისამართი არ წარმოადგენს ლოკალურ სისტემას, მარშრუტიზატორი გადააგზავნის პაკეტს შემდგომ ჰოპზე, მაშინ როდესაც ჰოსტი მას გააუქმებს. [2]

ქსელის მაგისტრალზე მდებარე მარშრუტიზატორებს უწევთ წამში მილიონობით პაკეტების გადაგზავნა, ამიტომ დროის ფაქტორი წარმოადგენს IP გადაგზავნის კრიტიკულ პარამეტრს. ამის გამო მარშრუტიზატორებზე IP გადაგზავნის პროცესი არის ძალზე ოპტიმიზირებული. ქვევით არის წარმოდგენილი ამ პროცესის დეტალური განხილვა და მასთან დაკავშირებული წარმადობის პრობლემები.

<u>პაკეტის სათაური</u>

როდესაც მარშრუტიზატორი ღებულობს IP პაკეტს, ის პირველ რიგში ამოწმებს პაკეტის სათაურის სიგრძეს, ვერსიის ნომერს და სათაურის საკონტროლო ჯამს. ვერსიის ნომერი უნდა უდრიდეს 4 (IP ვერსია 4 შემთხვევაში). სათაურის სიგრძის ველი წარმოადგენს 4 ბიტს, რომლებიც საკმარისია იმისათვის, რომ აღიწეროს სათაურის სიგრძე, რომელიც მოითხოვს მინიმუმ 20 ბაიტს. სათაურის საკონტროლო ჯამის შესამოწმებლად, მარშრუტიზატორი ითვლის სათაურის 16 ბიტიანი განყოფილებების ჯამს და შედეგს ადარებს საკონტროლო ჯამის ველს. IP პაკეტის სრული სათაურის შემოწმება წარმოადგენს სწრაფ პროცესს.

პარამეტრების დამუშავება

თუ კი IP პაკეტის სათაური შეიცავს სხვადასხვა პარამეტრებს, მარშრუტიზატორი ეცდება დაამუშავოს ეს პარამეტრები იქამდე, სანამ განხორციელდება მარშრუტიზაციის ცხრილის შემოწმება. თუმცა, ზოგი პარამეტრი ითხოვს დამატებით დამუშავებას იმის შემდეგ, როდესაც მარშრუტიზაციის გადაწყვეტილება უკვე იყო მიღებული. სათაურის პარამეტრები შეიძლება შეიცავდნენ მრავალრიცხოვან მეხსიერების შემოწმებებს, ასევე შეიძლება მოითხოვდნენ ამ პარამეტრების მოდიფიკაციას ან გაუქმებას. ზოგადად, IP პაკეტის სათაურის დამუშავება შეიძლება წარმოადგენდეს დროში გაწელილ პროცესს. ამის თავიდან ასარიდებლად, ხშირ შემთხვევებში მარშრუტიზატორები განცალკევებულად ამუშავებენ უპარამეტრო და პარამეტრიან პაკეტებს.

დანიშნულების IP მისამართის დამუშავება და მარშრუტიზაციის ცხრილის შემოწმება

პაკეტის სათაურის შემოწმების შემდეგ, მარშრუტიზატორი ამუშავებს პაკეტის დანიშნულების IP მისამართს და ამოწმებს იმას თუ ეს მისამართი წარმოადგენს ლოკალურს მისამართს. იმ შემთხვევაში, თუ კი ეს მისამართი არის ლოკალური, მარშრუტიზატორი ამოწმებს ფრაგმენტირების ველს და საჭიროების შემთხვევაში ახორციელებს ფრაგმენტირებული პაკეტების ხელახლა რეკონსტრუქციას. იმის გამო, რომ მარშრუტიზატორები იშვიათად წარმოადგენენ პაკეტების დანიშნულების კვანძს, მათ იშვიათად უწევთ ფრაგმენტირებული პაკეტების რეკონსტრუქცია და შესაბამისად ამ პროცესის ზეგავლენა მარშრუტიზატორის წარმადობაზე შეიძლება იყოს უგულვებელყოფილი. როდესაც ლოკალური მარშრუტიზატორი არ წარმოადგენს პაკეტის დანიშნულების კვანძს, იგი ახორციელებს მარშრუტიზაციის ცხრილის შემოწმებას იმისათვის, რომ მოიძიოს შემდგომი მარშრუტიზატორის მისამართი. თუ კი ამ შემოწმების შედეგად არ იქნა აღმოჩენილი შემდგომი ჰოპი, ე.ი. დანიშნულების მისამართის შესახებ მარშრუტიზაციის ცხრილში არ აღმოჩნდა შესაბამისი ჩანაწერი, ეს

პაკეტი იქნება გაუქმებული და პაკეტის გამგზავნს გადაეწოდება ICMP პაკეტი შეცდომის შესახებ.

TTL ველი

მარშრუტიზაციის ცხრილის შემოწმების შემდეგ, მარშრუტიზატორი ამოწმებს TTL ველს. თუ ეს ველი უდრის 1, პაკეტი იქნება გაუქმებული, და გამგზავნს გადაეცემა ICMP შეცდომის შეტყობინება. სხვა ყველა შემთხვევაში TTL ველის მნიშვნელობას აკლდება 1. იმის გამო, რომ TTL ველი მოწმდება მარშრუტიზაციის ცხრილის შემოწმების შემდეგ, ის პაკეტები, რომლების TTL ველი უდრის 1 და დანიშნულების მისამართი წარმოადგენს ლოკალურ მარშრუტიზატორს არ უქმდებიან.

პაკეტის ფრაგმენტირება და რეკონსტრუქცია

იმ შემთხვევაში თუ პაკეტის ზომა აღემატება გამავალი ქსელის ინტერფეისის MTU–ს მნიშვნელობას, მარშრუტიზატორი ვალდებულია მოახდინოს პაკეტის ფრაგმენტირება. იმ შემთხვევაში თუ კი პაკეტი მოითხოვს ფრაგმენტირებას და იმავე დროს მის სათაურში DF ბიტი უდრის 1, პაკეტი იქნება გაუქმებული და გამგზავნს გადაეცემა ICMP შეცდომის შეტყობინება. ფრაგმენტირება მოითხოვს შედარებით მეტ დამუშავებას. ეს პროცესი მოიცავს პაკეტის ნაწილებად დაყოფას, სულ მცირე ერთი ახალი პაკეტის შექმნას და სათაურის სხვადასხვა ველების ხელახლა დათვლას. IP ფრაგმენტირება საგრძნობლად ამცირებს მარშრუტიზატორის გამტარუნარიანობას. ფრაგმენტირების თავიდან ასარიდებლად, ჰოსტებს შეუძლიათ პაკეტების გაგზავნა მინიმალური MTU ზომით, რაც შეადგენს 576 ბაიტს.

საკონტროლო ჯამის გამოთვლა

იმის გამო, რომ ყველა მარშრუტიზატორი ახორციელებს პაკეტის სათაურის ზოგიერთი პარამეტრის მოდიფიკაციას, მარშრუტიზატორმა უნდა მოახდინოს პაკეტების სათაურების საკონტროლო ჯამის ხელახლა გამოთვლა. სულ მცირე, TTL ველის მნიშვნელობა უნდა იყოს შეცვლილი ყოველივე ჰოპზე, ამიტომ სათაურის ხელახლა გამოთვლა სავალდებულოა.

მნიშვნელოვანია აღინიშნოს, რომ მარშრუტიზატორი არ ახდენს მთლიანი პაკეტის ზომის გამოთვლას, არამედ გამოითვლება მხოლოდ სათაურის სიგრძე. ამის გამო ეს პროცესი არ წარმოადგენს დაყოვნების ან წარმადობის გაუარესების მიზეზს. [3]

შემდეგ ჰოპზე გადაცემა

პაკეტი გადაეცემა შემდეგ ჰოპს, რომელიც შეიძლება წარმოადგენდეს დანიშნულების ჰოსტს ან შემდგომ მარშრუტიზატორს. შეიძლება საჭირო გახდეს შემდგომი ჰოპის IP მისამართის დაკავშირება მის არხის შრის მისამართთან, რაც შესაძლებელია ARP პროტოკოლის გამოყენებით.

1.1.2 მარშრუტიზაციის ცხრილის შემოწმება

როდესაც მარშრუტიზატორი ან ჰოსტი აპირებს პაკეტის გაგზავნას, იგი აწარმოებს მარშრუტიზაციის ცხრილის შემოწმებას. პაკეტის IP მისამართი წარმოადგენს ძიების საკვანძო ელემენტს, რის მიხედვით ხორციელდება მარშრუტიზაციის ცხრილის შემოწმება და შესაბამისი ჩანაწერის იდენტიფიკაცია. ამ ძიების შედეგს წარმოადგენს შემდგომი მარშრუტიზატორის IP მისამართი ან ქსელის ინტერფეისის სახელი, რომლის მეშვეობით შესაძლებელია მოცემული პაკეტის პირდაპირი მიწოდება დანიშნულების ჰოსტამდე. მარშრუტიზაციის ცხრილის შემოწმება არ ცვლის ცხრილის შიგთავსს, ამის განხორციელება შეუძლია მხოლოდ სტატიკური მარშრუტის კონფიგურირებას ან დინამიკური მარშრუტიზაციის პროტოკოლებს.

მარშრუტიზაციის ცხრილები შეიძლება საგრძნობლად განსხვავდებოდნენ ერთმანეთისაგან, მაგრამ ყველას გააჩნია სულ მცირე ორი საერთო ველი. პირველი ველი წარმოადგენს დანიშნულების მისამართს მაშინ, როდესაც მეორე აღნიშნავს იმას, თუ როგორ უნდა იყოს გადაგზავნილი IP პაკეტი. ცხრილი 1.1–ზე წარმოდგენილია მარშრუტიზაციის ცხრილის ზოგადი სტრუქტურა.

დანიშნულების მისამართი	შემდგომი მარშრუტიზატორი
ქსელის პრეფიქსი	შემდგომი მარშრუტიზატორის IP მისამართი
ან	
ჰოსტის IP მისამართი	
ან	ან
loopback მისამართი	ქსელის ინტერფეისის სახელი
ან	
default მარშრუტი	

ცხრილი 1.1 მარშრუტიზაციის ცხრილის ზოგადი სტრუქტურა

მარშრუტიზაციის ცხრილში დანიშნულების მისამართი შეიძლება წარმოადგენდეს ქსელის პრეფიქს, ჰოსტის IP მისამართს, „loopback“ მისამართს ან default მარშრუტს. მარშრუტიზაციის ცხრილის იმ ჩანაწერებს, რომლებიც შეიცავენ ქსელის პრეფიქს ეწოდებათ ქსელის მარშრუტები. ასეთი ჩანაწერები წარმოადგენენ მარშრუტიზაციის ცხრილის ჩანაწერების უდიდეს ნაწილს. მარშრუტიზაციის ცხრილის იმ ჩანაწერებს, რომლებიც შეიცავენ დანიშნულების IP მისამართს ეწოდებათ ჰოსტის მარშრუტები. ჰოსტის მარშრუტი გამოიყენება იმ შემთხვევაში, როდესაც საჭიროა რაიმე ქსელში კერძო მისამართამდე ნაკადის გაგზავნა განსხვავებული მარშრუტით ვიდრე ქსელის დანარჩენ მისამართებამდე. ბევრ ოპერაციულ სისტემაში, loopback მისამართი, რომელიც არის 127.0.0.1/32, წარმოადგენს მარშრუტიზაციის ცხირილის ცალკეულ ჩანაწერს. აგრეთვე მარშრუტიზაციის ცხრილს შეიძლება გააჩნდეს default მარშრუტი. ეს ჩანაწერი გამოიყენება იმ შემთხვევაში, თუ პაკეტის დანიშნულების მისამართი არ დაემთხვა მარშრუტიზაციის ცხრილის არც ერთ სხვა ჩანაწერს. [4]

მარშრუტიზაციის ცხრილს შეიძლება გააჩნდეს მხოლოდ ერთი ასეთი ჩანაწერი(დატვირთვის განაწილების შემთხვევის გარდა). მარშრუტიზაციის ცხრილში შემდგომი მარშრუტიზატორის სვეტი, რომელიც არის წარმოდგენილი ნახ. 1.5 შეიცავს შემდგომი მარშრუტიზატორის IP მისამართს ან ქსელის ინტერფეისის სახელს. ინტერფეისის სახელი

გამოიყენება იმ შემთხვევაში, როდესაც შესაძლებელია პაკეტის პირდაპირი მიწოდება ჰოსტამდე. მარშრუტიზატორს, რომელიც წარმოადგენს default მარშრუტის შემდგომ მარშრუტიზატორს, ეწოდება default კარიბჭე* ან უკანასკნელი მიმართვის კარიბჭე (gateway of last resort). რეალური ოპერაციული სისტემის მარშრუტიზაციის ცხრილი ასევე შეიცავს დამატებით სვეტებს. მაგალითად Linux ოპერაციულ სისტემაში, მარშრუტიზაციის ცხრილი ყოველთვის მოიცავს, როგორც შემდგომი მარშრუტიზატორის IP მისამართს ასევე იმ ქსელის ინტერფეისის სახელს, რომლის მეშვეობით შესაძლებელია ამ მარშრუტიზატორამდე მიღწევა. მარშრუტიზატორების მარშრუტიზაციის ცხრილები შეიცავენ მრავალ ათას ჩანაწერს და ძირითადად არიან შედგენილი დინამიკური მარშრუტიზაციის პროტოკოლების მიერ. ამის გარდა, ჰოსტების მარშრუტიზაციის ცხრილი მოითხოვს მხოლოდ რამოდენიმე ჩანაწერს, რომლებიც შეიძლება იყვნენ სტატიკურად დაკონფიგურირებული. ტიპიური ჰოსტი, რომელსაც გააჩნია ერთი ქსელის ინტერფეისი მოითხოვს მარშრუტიზაციის ცხრილის ერთი ქსელის პრეფიქსის ჩანაწერს, ერთი loopback-ის მისამართის ჩანაწერს და ერთი default მარშრუტის ჩანაწერს.

როდესაც მარშრუტიზატორი ან ჰოსტი აწარმოებს მარშრუტიზაციის ცხრილის შემოწმებას, ის ეძებს ჩანაწერს, რომელიც მაქსიმალურად ემთხვევა პაკეტის დანიშნულების IP მისამართს. ამ შერჩევაში უპირატესობა ენიჭება იმ ჩანაწერს, რომლის შესატყვისი ბიტების რაოდენობა არის უმაღლესი. ამ ლოგიკას ეწოდება უგრძელესი პრეფიქსის შერჩევა. პირველ რიგში მოწმდება მარშრუტიზაციის ცხრილის /32 ჩანაწერები. თუ პაკეტის დანიშნულების მისამართი არ ემთხვევა არც ერთი ჰოსტის მარშრუტს, შემდეგ ეტაპზე მოწმდება მარშრუტიზაციის ცხრილის ყველა /31 მარშრუტი. თუ დამთხვევა არ იყო ნაპოვნი, პროცესი გრძელდება /30 მარშრუტების შემოწმებით და ასე შემდეგ. თუ ამ პროცესის ბოლომდე ვერ იყო მოხერხებული დანიშნულების მისამართის შესატყვისი ჩანაწერის მოძიება მარშრუტიზაციის ცხრილში, მაშინ მარშრუტიზატორი ირჩევს default

მარშრუტს. იმის გამო, რომ default მარშრუტი მოწმდება ბოლო რიგში, ხშირ შემთხვევაში მარშრუტიზაციის ცხრილი აღწერს default მარშრუტს 0.0.0.0/0 სახით, რაც წარმოადგენს დანიშნულების მისამართს 0 ბიტიანი პრეფიქსით. იმ შემთხვევაში, თუ მარშრუტიზატორი ვერ მოახერხებს დანიშნულების მისამართის შესაფერისი ჩანაწერის პოვნას მარშრუტიზაციის ცხრილში და ამავე დროს მარშრუტიზაციის ცხრილი არ შეიცავს default მარშრუტს, პაკეტი იქნება გაუქმებული და გამგზავნს მიეწოდება ICMP „მიუწვდომელი ქსელი“–ის შეცდომის შეტყობინება.

183.43.11.88 →

დანიშნულების მისამართი	შემდგომი მარშრუტიზატორი
10.0.0.0/8	150.199.20.1
183.43.0.0/16	150.199.20.1
183.43.64.0/20	88.51.10.1
183.43.192.0/20	150.199.20.1
183.43.11.0/24	88.51.10.1
183.43.11.90/32	88.51.10.1
default	88.51.10.1

ცხრილი 1.2 უგრძელესი პრეფიქსის შერჩევა

ცხრილი 1.2 წარმოადგენს იმის მაგალითს თუ როგორ ახორციელებს მარშრუტიზატორი უგრძელესი შესატყვისი პრეფიქსის არჩევას. პაკეტის დანიშნულების IP მისამართი არის 183.43.11.88 და წარმოადგენს 183.43.0.0/16 და 183.43.11.0/24 ქსელების ნაწილს. აქედან 183.43.11.0/24 ჩანაწერს გააჩნია დანიშნულების მისამართთან 24 ბიტიანი დამთხვევა, რაც მოცემულ შემთხვევაში წარმოადგენს ყველაზე გრძელ შესატყვის პრეფიქს და შესაბამისად პროცესი ირჩევს ამ ჩანაწერს. [5,6]

1.2 სტატიკური მარშრუტიზაცია

Default მარშრუტის კონფიგურირებისათვის, ჰოსტების უმეტესობა იყენებს სტატიკურ მარშრუტიზაციას. სტატიკური მარშრუტიზაციის ცხრილების აკურატულობის შენარჩუნება წარმოადგენს რუტინულ შრომატევად პროცესს. ამის გამო, მარშრუტიზატორების უმეტესობა იყენებს დინამიკური მარშრუტიზაციის პროტოკოლებს იმისათვის, რომ ავტომატურად განახლდნენ მარშრუტიზაციის ცხრილები და ქსელის

ტოპოლოგიაში ნებისმიერი ცვლილება აისახოს მარშრუტიზაციის ცხრილის განახლებით. სტატიკური მარშრუტების ჩანაწერების დამატება შესაძლებელია იმ შემთხვევაშიც კი, როდესაც მარშრუტიზატორი იყენებს მარშრუტიზაციის პროტოკოლს, თუმცა სტატიკურმა მარშრუტებმა შეიძლება ზევალენა მოახდინონ მარშრუტის დინამიკური შერჩევის პროცესზე. [7,8]

როდესაც სტატიკური მარშრუტიზაციის ჩანაცერების დამატება ხორციელდება ხელით, ბრძანებათა სტრიქონის (command line) გამოყენებით, ისინი ძალაში რჩებიან მხოლოდ ოპერაციული სისტემის გადატვირთვამდე. იმისათვის, რომ მოხერხდეს სტატიკური მარშრუტების ჩანაწერების მუდმივი შენარჩუნება მარშრუტიზაციის ცხრილში, საჭიროა სტატიკური მარშრუტის დამატების კომანდის შენახვა კონფიგურაციის ფაილში. IOS ოპერაციულ სისტემაში, მას წარმოადგენს საწყისი კონფიგურაციის ფაილი(start-up configuration file).

მარშრუტიზაციის ცხრილის გარკვეული ჩანაწერები ავტომატურად კონფიგურირდებიან. მაგალითად, როდესაც ხდება ქსელის ახალ ინტერფეისზე IP მისამართის მინიჭება, ოპერაციული სისტემების უმრავლესობა ავტომატურად ამატებენ ამ ქსელის პრეფიქსის ჩანაწერს მარშრუტიზაციის ცხრილში და აღნიშნავენ მას, როგორც პირდაპირ შეერთებულ ქსელს. [9,10]

1.2.1 IP გადაგზავნა(forwarding) Linux და IOS ოპერაციულ სისტემებში

Linux სისტემაში, IP გადაგზავნის გასააქტიურებლად, საჭიროა ფაილში /proc/sys/net/ipv4/ip_forward ჩაიწეროს 1, მხოლოდ მის გასაუქმებლად საჭიროა 1 შეიცვალოს 0–ით. ამის განხორციელება ასევე შესაძლებელია ბრძანებით:

PC1% echo “1” > /proc/sys/net/ipv4/ip_forward

ბრძანება echo წერს მოცემულ არგუმენტს, ამ შემთხვევაში სტრიქონს 1, სტანდარტული ინფორმაციის გამომტანი. ინფორმაციის ფაილში

ჩასაწერად, საჭიროა გადამისამართების ოპერატორის „>“ შემდეგ ფაილის სახელის მითითება. IP გადაგზავნის გამოსართავად შაჭიროა შემდეგი ბრძანების გამოყენება:

PC1% echo “0” > /proc/sys/net/ipv4/ip_forward

ამ ბრძანებას აქვს დაუყოვნებელი ეფექტი, სამაგიეროთ მისი მოქმედება არ არის მუდმივი და სისტემის გადატვირთვის შემთხვევაში ეს ცვლილებები იკარგება. იმისათვის რომ IP გადაგზავნის ფუნქცია ჩაირთოს მუდმივ რეჟიმში, საჭიროა შემდეგი კონფიგურაციის ფაილის რედაქტირება /etc/sysctl.conf. IP გადაგზავნა ჩართულია, თუ კი ეს ფაილი შეიცავს სტრიქონს net.ipv4.ip_forward = 1 და გამოთულია თუ კი ეს სტრიქონი არ არსებობს ან შეიცავს net.ipv4.ip_forward = 0. ამ კონფიგურაციის ფაილის ცვლილება ძალაში შევა Linux სისტემის გადატვირთვის შემდგომ ან ქსელის მომსახურების მოდულის გადატვირთვის შემდეგ. [11]

IOS ოპერაციულ სისტემაში IP გადაგზავნის ჩასართავად საჭიროა გლობალურ კონფიგურაციის რეჟიმში ჩაიწეროს ბრძანება “ip routing”

Router1(config)# ip routing

და ამ ფუნქციის გასაუქმებლად საჭიროა შემდეგი ბრძანების შეყვანა:

Router1(config)# no ip routing

IP გადაგზავნის გამორთვის შემთხვევაში, მარშრუტიზატორი ან ჰოსტი წაშლის IP მარშრუტიზაციის ცხრილებს და მასთან დაკავშირებულ სხვა ინფორმაციას. [12]

1.2.2 სტატიკური მარშრუტიზაცია Linux სისტემებში

Linux ოპერაციულ სისტემას გააჩნია მარშრუტიზაციის ცხრილები და მარშრუტიზაციის ქეში. მარშრუტიზაციის ცხრილი შეიცავს მარშრუტების მუდმივ ჩანაწერებს, რომლების ჩამატება ხორციელდება სტატიკური ან დინამიკური მარშრუტიზაციის მეთოდებით. მარშრუტიზაციის ქეში მოიცავს ბოლოს გამოყენებული მარშრუტების ჩანაწერებს. მარშრუტიზაციის ქეშის დანიშნულება მდგომარეობს იმაში, რომ

გაუმჯობესდეს მარშრუტიზაციის ცხრილის შემოწმების ეფექტურობა და გაიზარდოს მისი სწრაფქმედება. Linux მარშრუტების ქეში მდებარეობს მთავარ მეხსიერებაში. მარშრუტის ძიებისას, სისტემა პირველ რიგში ამოწმებს მარშრუტის არსებობას ქეშში და მხოლოდ ამის შემდეგ მოწმდება მარშრუტიზაციის ცხრილი. თუ კი ქეში შეიცავს ძიებად მარშრუტს, მარშრუტიზაციის ცხრილის შემოწმება აღარ არის საჭირო. იმ შემთხვევაში, თუ მარშრუტიზაციის ქეში არ შეიცავს ძიებად მარშრუტს, სისტემა ამოწმებს მარშრუტიზაციის ცხრილს და სასურველი მარშრუტის აღმოჩენის შემდგომ ის იწერება ქეშში.

1.2.3 Linux მარშრუტიზაციის ცხრილი

მარშრუტიზაციის ცხრილის გამოსაჩენად საჭიროა შემდეგი ბრძანების გაცემა: netstat –rn. ამ ბრძანების გაცემის შედეგად გამოტანილი ინფორმაცია* წარმოდგენილია ნახაზზე 1.5.

```
% netstat -rn
Kernel IP routing table
Destination     Gateway      Genmask         Flags   MSS Window  irtt Iface
10.0.1.4        0.0.0.0      255.255.255.255 UH      40  0       0     eth0
10.0.3.0        0.0.0.0      255.255.255.0   U       40  0       0     eth1
10.0.2.0        0.0.0.0      255.255.255.0   U       40  0       0     eth0
10.0.5.0        10.0.2.1     255.255.255.0   UG      40  0       0     eth0
127.0.0.0       0.0.0.0      255.0.0.0       UH      40  0       0     lo
0.0.0.0         10.0.3.1     0.0.0.0         UG      40  0       0     eth1
```

ნახ. 1.5 მარშრუტიზაციის ცხრილის მაგალითი Linux სისტემაში

ყოველი სტრიქონი წარმოადგენს მარშრუტიზაციის ცხრილის ჩანაწერს. პირველი სვეტი მიუთითებს დანიშნულების მისამართზე, რომელიც შეიძლება იყოს ქსელის ან ჰოსტის მისამართი. მესამე სვეტი ასახავს დანიშნულების მისამართის ქსელის ნიღაბს. შესაბამისად, მოცემულ მარშრუტიზაციის ცხრილში ჩანაწერები 10.0.1.4/32, 10.0.3.0/24, 10.0.2.0/24, 10.0.5.0/24, 127.0.0.0/8 და 0.0.0.0/0 წარმოადგენენ დანიშნულების მისამართებს. დანიშნულების მისამართი 127.0.0.0/24 წარმოადგენს loopback ჩანაწერს და 0.0.0.0/0 კი მიუთითებს default მარშრუტს. მეორე სვეტი Gateway(კარიბჭე*) წარმოადგენს შემდგომი მარშრუტიზატორის მისამართს.

იმ შემთხვევაში, თუ კი ეს ველი შეიცავს 0.0.0.0 მნიშვნელობას, ეს მიუთითებს იმაზე, რომ მოცემული დანიშნულების ქსელი არის უშუალოდ პირდაპირ შეერთებული. ბოლო სვეტი მიუთითებს იმ ქსელის ინტერფეისს, რომელიც გამოყენებული იქნება ამ ჩანაწერის მიხედვით პაკეტების გადასაცემად. ნახაზზე 1.5 მოცემულია ორი პირდაპირ შეერთებული ქსელი, ესენი არიან 10.0.2.0/24, რომელიც ხელმისაწვდომია eth0 ინტერფეისის მეშვეობით და 10.0.3.0/24 eth1 ინტერფეისის მეშვეობით. ასევე ცხრილში აღნიშნულია სხვადასხვა ალამი, რომლებიც მიუთითებენ მარშრუტის ტიპს. ალამი G აღნიშნავს იმას, რომ მარშრუტის შემდგომი მარშრუტიზატორი წარმოადგენს კარიბჭეს (gateway). ალამი H მიუთითებს იმას, რომ ჩანაწერი წარმოადგენს ჰოსტის მარშრუტს, ხოლო ალამი U კი მიანიშნებს იმას, რომ ცხრილში მოყვანილი ინტერფეისი არის გამოყენებადი. ამის გარდა, სვეტები MSS, Window და irtt წარმოადგენენ ამ მარშრუტების TCP კავშირის პარამეტრებს და შესაბამისად მიუთითებენ TCP სეგმენტის მაქსიმალურ ზომას, განცხადებული ფანჯრის(window) ზომას და საწყისი პაკეტის ორმხრივი გადაცემის დროს. იმ შემთხვევაში, თუ ამ ველების მნიშვნელობები წარმოადგენენ 0, ეს იმაზე მეტყველებს, რომ გამოიყენება სტანდარტული მნიშვნელობები.

როდესაც Linux ოპერაციული სისტემა ახორციელებს მარშრუტიზაციის ცხრილის შემოწმებას, ის პირველ რიგში ამოწმებს მარშრუტიზაციის ქეშს(cache). თუ კი შესაბამისი ჩანაწერის მოძებნა ვერ მოხერხდა, ამის შემდეგ სისტემა ამოწმებს მარშრუტიზაციის ცხრილს. მარშრუტიზაციის ცხრილის ყოველი შემოწმების შემდეგ, სისტემა ამატებს ახალ ჩანაწერს მარშრუტიზაციის ქეშში. მარშრუტიზაციის ქეში არ ახორციელებს ქსელების ჩანაწერების შეერთებას(aggregation) და ამის გამო ყველა დანიშნულების მისამართს გააჩნია ცალკე ჩანაწერი. ამის შედეგად, მარშრუტიზაციის ქეშის შემოწმებას არ ესაჭიროება ყველაზე გრძელი პრეფიქსის დამთხვევა. თუ კი მარშრუტიზაციის ქეშის ჩანაწერი არ

გამოიყებენა გარკვეული დროის განმავლობაში, სტანდარტულად 10 წუთი, მაშინ ეს ჩანაწერი უქმდება. [13]

როდესაც სისტემა ღებულობს ICMP გადამისამართების შეტყობინებას, შესაბამისი ჩანაწერი ემატება მარშრუტიზაციის ქეშს, მაგრამ მარშრუტიზაციის ცხრილი რჩება უცვლელი. მარშრუტიზაციის ქეშის ჩანაწერების გამოსაჩენად საჭიროა route -Cn ბრძანების გაცემა. ამ ბრძანების გაცემის შედეგად გამოტანილი ინფორმაცია წარმოდგენილია ნახაზზე 1.6.

```
PC1%route -Cn

Kernel IP routing cache
Source          Destination     Gateway         Flags Metric Ref    Use Iface
10.0.1.11       10.0.1.21       10.0.1.21       il    0      0        0 lo
10.0.2.1        10.0.1.11       10.0.1.11       i     0      0        6 eth0
10.0.1.11       10.0.2.1        10.0.2.1        i     0      0        6 eth1
10.0.1.11       10.0.3.1        10.0.2.1        i     0      0        1 eth1
10.0.3.1        10.0.1.11       10.0.1.11             0      0        1 eth0
10.0.1.11       10.0.3.41       10.0.2.1        i     0      0        1 eth1
10.0.3.41       10.0.1.11       10.0.1.11             0      0        1 eth0
```

ნახ.1.6 მარშრუტიზაციის ქეშის ცხრილი

საჭიროა აღინიშნოს, რომ მარშრუტიზაციის ქეშის ჩანაწერები არ იყენებენ პრეფიქსებს ან ქსელის მასკებს. ყოველი დანიშნულების მისამართისათვის არსებობს შესაბამისი ჩანაწერი. ამის გარდა, მარშრუტიზაციის ქეშის ცხრილს გააჩნია წყაროს მისამართის ჩანაწერები. ალმების სვეტში (Flags) ალამი “i” აღნიშნავს იმას, რომ წყაროს IP მისამართი არის უშუალოდ პირდაპირ შეერთებული მოცემულ სისტემასთან და ალამი „l“ მიუთითებს იმაზე, რომ დანიშნულების მისამართი წარმოადგენს ლოკალურ სისტემას. სვეტი “Metric” წარმოადგენს არხების შეფასების ერთეულს და გამოიყენება დინამიკური მარშრუტიზაციის პროტოკოლების მიერ საუკეთესო მარშრუტის შერჩევისათვის. პირდაპირ შეერთებული ქსელების მეტრიკის მნიშვნელობას წარმოადგენს 0. სვეტი Ref მიუთითებს კერძო ჩანაწერის გამოყენების რაოდენობაზე, ხოლო სვეტი Use ასახავს იმ პაკეტების რაოდენობას, რომლებიც იყვნენ გადაგზავნილი მარშრუტიზაციის ქეშის კერძო ჩანაწერის გამოყენებით. ნახაზებზე 1.5 და 1.6, ბოლო სვეტი Iface ასახავს ქსელის ინტერფეისს, რომლის გამოყენებით ხორციელდება კერძო ჩანაწერის მიხედვით პაკეტების გაგზავნა. [14]

Linux უახლესი სისტემების შესაძლებლობები საკმაოთ გაუმჯობესებულია მარშრუტიზაციის ფუნქციონალობის თვალსაზრისით. შესაძლებელია რამოდენიმე მარშრუტიზაციის ცხრილის მითითება და იმ წესების მითითება, თუ როდის და სად უნდა იყოს გამოყენებული კერძო მარშრუტიზაციის ცხრილი. ახალი მარშრუტიზაციის ცხრილების მეშვეობით შესაძლებელია შემდგომი მარშრუტიზატორის მისამართის მითითება არა მარტო დანიშნულების მისამართის მიხედვით, არამედ წყაროს IP მისამართის, DiffServ ველის ან სხვა კრიტერიუმის მიხედვით. ამ ფუნქციონალური გაუმჯობესების მისაღებად საჭიროა დამატებითი პროგრამული უზრუნველყოფის დაყენება, კერძოთ iproute2. ეს უტილიტა უზრუნველყოფს მრავალრიცხოვან ბრძანებათა არჩევანს, რომელიც საგრძნობლად განსხვავდება სხვა Unix-ის ტიპის ოპერაციული სისტემებისაგან.

მარშრუტიზაციის ცხრილები და სტატიკური მარშრუტიზაცია Cisco IOS ოპერაციულ სისტემებში Linux სისტემების მსგავსად, იყენებს ქეშს იმისათვის, რომ მოახდინოს მარშრუტიზაციის ცხრილის შემოწმების აჩქარება. ამასთან ერთად, მარშრუტების ქეშირებისათვის IOS სისტემა იყენებს რამოდენიმე სხვადასხვა მეთოდს.

მარშრუტიზაციის ცხრილის გამოსაჩენად საჭიროა მარშრუტიზატორის კონსოლში შემდეგი ბრძანების გაცემა show ip route. ამ ბრძანების გამოტანილი ინოფრმაცია გამოიყურება შემდეგნაირად:

```
router#show ip route
    Codes: C - connected, S - static, I - IGRP, 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, E – EGP
    i - IS-IS, 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 10.0.2.2 to network 0.0.0.0
```

10.0.0.0/24 is subnetted, 2 subnets

C 10.0.1.0 is directly connected, FastEthernet0/1

C 10.0.2.0 is directly connected, FastEthernet0/0

S* 0 .0.0.0/0 [1/0] via 10.0.2.2

პირველი სტრიქონი განმარტავს ყველა არსებული და შესაძლო ალმის მნიშვნელობას და ასევე იმას, თუ როგორ იყო შედგენილი მოცემული მარშრუტიზაციის ცხრილი. ალმები მდებარეობენ მარშრუტიზაციის ცხრილის პირველ სვეტში. მოცემული მარშრუტიზაციის ცხრილი შეიცავს ორი ქსელის ჩანაწერს, 10.0.1.0/24 და 10.0.2.0/24. ალამი C, რომელიც წარმოდგენილია პირველ სვეტში, მიუთითებს იმას, რომ ორივე ჩანაწერი მიეკუთვნება პირდაპირ შეერთებულ ქსელებს. ასეთი ტიპის ქსელებისათვის, IOS ოპერაციული სისტემა იყენებს ინტერფეისის სახელს როგორც შემდგომი მარშრუტიზატორის ინფორმაციას. ბოლო ჩანაწერი შეიცავს S ალამს და მონიშნულია * ნიშნით. ეს მიუთითებს იმაზე, რომ კერძო ჩანაწერი შექმნილია სტატიკური მარშრუტის საფუძველზე და წარმოადგენს default მარშრუტს.

საჭიროა აღინიშნოს, რომ როგორც Linux ოპერაციულ სისტემაში ასევე IOS სისტემაში default მარშრუტი აღინიშნება პრეფიქსით 0.0.0.0/0. ამ ჩანაწერის შემდგომი მარშრუტიზატორის მისამართს წარმოადგენს 10.0.2.2. IOS ოპერაციულ სისტემაში default მარშრუტს ეწოდება „უკანასკნელი მიმართვის კარიბჭე“ (gateway of last resort). მარშრუტიზაციის ცხრილის ბოლო ჩანაწერს გააჩნია ორი პარამეტრი [1/0], რომლებიც შესაბამისად აღნიშნავენ „ადმინისტრაციულ მანძილს“ და „მეტრიკას“. ადმინისტრაციული მანძილი თამაშობს გადამწყვეტ როლს მაშინ, როდესაც მარშრუტიზატოს გააჩნია კერძო მარშრუტის ორი ან მეტი წყარო. ეს შეიძლება იყოს რამოდენიმე დინამიკური მარშრუტიზაციის პროტოკოლი ან სტატიკური მარშრუტი და დინამიკური მარშრუტიზაციის პროტოკოლი, ე.ი. ერთი ჩანაწერი და ორზე მეტი ნებისმიერი წყარო. ამგვარად, იმისათვის, რომ მარშრუტიზატორმა მოახერხოს ყველაზე სანდო წყაროს დადგენა, ის ამოწმებს ადმინისტრაციული მანძილის მნიშვნელობას და ირჩევს იმ

წყაროს მიერ მოწოდებულ ჩანაწერს, რომლის ადმინისტრაციული მანძილის მნიშვნელობა არის ყველაზე მცირე.

ყოველი მარშრუტიზაციის პროტოკოლისათვის არსებობს შესაბამისი ადმინისტრაციული მანძილის სტანდარტული მნიშვნელობა, მათშორის ყველაზე ფართოდ გამოყენებული პროტოკოლები არის ჩამოთვლილი ცხრილში 1.3

მარშრუტის წყარო	ადმინისტრაციული მანძილის სტანდარტული მნიშვნელობა
პირდაპირ შეერთებული ინტერფეისი	0
სტატიკური მარშრუტი	1
Enhanced Interior Gateway Routing Protocol EIGRP შეჯამებული მარშრუტი (Summary route)	5
External Border Gateway Protocol (BGP)	20
EIGRP შიდა მარშრუტი (Internal)	90
IGRP	100
OSPF	110
Intermediate System-to-Intermediate System (IS-IS)	115
Routing Information Protocol (RIP)	120
Exterior Gateway Protocol (EGP)	140
On Demand Routing (ODR)	160
EIGRP გარე მარშრუტი (External)	170
BGP შიდა მარშრუტი (Internal)	200
უცნობი წყაროს მარშრუტი	255

ცხრილი 1.3 წარმოადგენს ადმინისტრაციული მანძილის მნიშვნელობებს

ადმინისტრაციული მანძილი წარმოადგენს კონფიგურირებად პარამეტრს. კვადრატულ ფრჩხილებში მეორე პარამეტრს წარმოადგენს მეტრიკა, რომელიც გამოიყენება დინამიკური მარშრუტიზაციის

პროტოკოლების მიერ უმოკლესი გზის ასარჩევად. სტატიკური მარშრუტებისათვის მეტრიკის პარამეტრს არ აქვს მნიშვნელობა და ამიტომ IOS ანიჭებს მასს 0 მნიშვნელობას. [15]

მარშრუტიზაციის ქეშის გამოსაჩენად საჭიროა მარშრუტიზატორის კონსოლში შემდეგი ბრძანების გაცემა:

router#1 show ip cache
IP routing cache 165 entries, 19800 bytes
Minimum invalidation interval 2 seconds, maximum interval 5 seconds, quiet interval 3 seconds, threshold 0 requests
Invalidation rate 0 in last second, 0 in last 3 seconds
Last full cache invalidation occurred 0:00:00 ago

Prefix/Length Age Interface Next Hop
10.0.1.10/24 0:01:48 Ethernet0 10.0.1.10
10.0.2.10/24 0:04:29 Ethernet0 10.0.1.1
10.0.1.137/24 0:12:18 Ethernet1 10.0.4.2
10.0.3.10/24 0:13:19 Ethernet1 10.0.3.1

საჭიროა აღინიშნოს, რომ Linux მარშრუტიზაციის ქეშის ცხრილისაგან განსხვავებით, IOS სისტემის ქეშის ჩანაწერებში ინახავს ქსელის პრეფიქსებს.

1.3 წესებზე დაფუძნებული მარშრუტიზაცია

თანამედროვე მაღალი წარმადობის მქონე ქსელებში, ორგანიზაციებს ესაჭიროებათ პაკეტების გადამისამართების და მარშრუტიზაციის დანერგვის მოქნილი მეთოდები, რომლებიც ორგანიზაციაში არსებული წესების რეალიზაციის საშუალებას იძლევიან. ხშირ შემთხვევებში, ასეთი წესები ცდებიან ტრადიციული დინამიკური მარშრუტიზაციის პროტოკოლების შესაძლებლობებს. იმ შემთხვევებში, როდესაც ადმინისტრაციული წესების შეზღუდვა მოითხოვს გარკვეული ნაკადების განსაკუთრებული მარშრუტით გადამისამართებას, წესებზე დაფუძნებული მარშრუტიზაციის (წდმ*) ალგორითმი წარმოადგენს ყველაზე მოსახერხებელ და მოქნილ გადაწყვეტილებას. წდმ–ს გამოყენებით,

ორგანიზაციებს საშუალება ეძლევათ დანერგონ სხვადასხვა წესებზე აგებული მარშრუტიზაცია და ამის შედეგად განხორციელდეს პაკეტების შერჩევა და სხვადასხვა მარშრუტით გადაგზავნა.

წესებზე დაფუძნებული მარშრუტიზაცია ასევე იძლევა პაკეტების მონიშვნის შესაძლებლობას, რაც შემდგომ ეტაპებზე შეიძლება იყოს გამოყენებული განსხვავებული მომსახურების ხარისხის უზრუნველსაყოფად.

ამ ქვეთავში განხილული მაგალითების რეალიზაცია განხორციელებულია Cisco-ს აპარატურაზე და ბრძანებათა სიმრავლე ასევე წარმოადგენს IOS ოპერაციული სისტემის სპეციფიკას. ამავე დროს წდმ–ის საბაზისო პრინციპები რჩებიან უცვლელი მიუხედავად იმისა, თუ რომელი მწარმოებლის მოწყობილობა იყო გამოყენებული. [16]

წესებზე დაფუძნებული მარშრუტიზაციის ძირითად უპირატესობებს წარმოადგენენ:

- მარშრუტის შერჩევა წყაროზე დაყრდნობით – ინტერნეტის პროვაიდერს ან ნებისმიერ ორგანიზაციას შეუძლია გამოიყენოს წესებზე დაფუძნებული მარშრუტიზაცია იმისათვის, რომ მოახდინოს სხვადასხვა მომხმარებლებისაგან მიღებული ნაკადების მარშრუტიზაცია წინასწარ შერჩეული სხვადასხვა მარშრუტებით.
- მომსახურების ხარისხის უზრუნველსაყოფად – ორგანიზაციებს შეუძლიათ განახორციელონ მომსახურების ხარისხის უზრუნველყოფის ალგორითმები. ამის მისაღწევად საჭიროა მიღებული პაკეტების სათაურში Precedence ან TOS ველების მოდიფიკაცია. მნიშვნელოვანია აღინიშნოს, რომ ეს პროცესი საჭიროა განხორციელდეს ქსელის პერიფერიაზე, რის შედეგად ქსელის შიგნით შეაღწევენ მხოლოდ მონიშნული პაკეტების.
- ხარჯების დაზოგვა – ორგანიზაციებს შეუძლიათ შეამცირონ არხებთან დაკავშირებული ხარჯები. ეს შესაძლებელია ძვირსა და იაფფასიან არხებს შორის ნაკადების გადანაწილების მეშვეობით.

მაგალითად, ჩვეულებრივი არაკორპორაციული ნაკადების გადამისამართება დაბალი გამტარუნარიანობის მქონე არხებზე და ორგანიზაციის საქმიანობისათვის კრიტიკული ნაკადების მარშრუტიზაცია ძვირფასიანი მაღალი გამტარუნარიანობის მქონე არხების გამოყენებით.

- დატვირთვის განაწილება – დინამიკური მარშრუტიზაციის პროტოკოლების ფუნქციონალურ თვისებებთან ერთად შესაძლებელია წდმ–ის გამოყენება რათა მოხერხდეს არხების დატვირთვის თანაბრად განაწილება მომხმარებლების ნაკადების განაწილების მეშვეობით. დინაკიმური მარშრუტიზაციის პროტოკოლებისაგან განსხვავებით, წდმ–ს შეუძლია ამ ამოცანის გადაწყვეტა ნაკადების გაცილებით მეტ პარამეტრებზე დაყრდნობით.

წესებზე დაფუძნებული მარშრუტიზაცია წარმოადგენს პაკეტების გადამისამართების და მარშრუტიზაციის მექანიზმს, რომელიც გადაწყვეტილების მისაღებად იყენებს ქსელის ადმინისტრატორის მიერ განსაზღვრულ წესებს. ეს მეთოდი წარმოადგენს მარშრუტიზატორებს შორის პაკეტების გადამისამართების მეტად მოქნილ მექანიზმს და ავსებს დინამიკური მარშრუტიზაციის პროტოკოლების ფუნქციონალურობას. მარშრუტიზატორები ახორციელებენ პაკეტების გადაგზავნას დანიშნულების მისამართის მიხედვით და ამის განსახორციელებლად იყენებენ სტატიკურ მარშრუტებს ან დინამიკური მარშრუტიზაციის სხვადასხვა პროტოკოლებს, როგორებიც არიან Routing Information Protocol (RIP), Open Shortest Path First (OSPF) ან Enhanced Interior Gateway Routing Protocol (EIGRP). დანიშნულების მისამართის მაგივრად, წდმ სთავაზობს ქსელის ადმინისტრატორს დაადგინოს და განახორციელოს მარშრუტიზაციის წესი, რომელიც ნებას დართავს ან აკრძალავს მარშრუტებს, ნაკადების შემდეგ პარამეტრებზე დაყრდნობით:

- დანიშნულების მისამართი/ჰოსტი
- აპლიკაცია

- პროტოკოლი
- პაკეტების ზომა

წესი შეიძლება გამოიყურებოდეს საკმაოდ მარტივად. მაგალითად, ქსელი A-დან ქსელი B-ში ყველა ნაკადის აკრძალვით, ან შედარებით რთული წესით, მაგ. ქსელი A-დან ქსელი B-ში მიმართული ნაკადებისაგან იმ ნაკადების აკრძალვით, რომლებსაც გააჩნიათ C, D, E მახასიათებლები. C, D ან E შეიძლება წარმოადგენდეს TOS ველის რაიმე მნიშვნელობას, წყაროს გარკვეულ მისამართს ან ტრანსპორტის შრის პროტოკოლის გარკვეულ პორტს. [17]

1.3.1 ქსელური ნაკადების ტეგირება

წდმ საშუალებას აძლევს ქსელის ადმინისტრატორს მოახდინოს ნაკადების კლასიფიცირება მისაწვდომობის მართვის ნუსხის* (ACL) გამოყენებით, რის შედეგად შესაძლებელი ხდება პაკეტების სათაურის IP Precedence და TOS ველების მნიშვნელობების მოდიფიკაცია. ამ პროცესს ეწოდება პაკეტების მითითებული კლასიფიკაციით ტეგირება.[18]

წდმ–ს გამოყენებით ნაკადების კლასიფიკაცია საშუალებას აძლევს ქსელის ადმინისტრატორს მოახდინოს ნაკადების სხვადასხვა მომსახურების კლასების იდენტიფიცირება. მომსახურების ხარისხის სხვადასხვა კლასს დაერთვება წესით განსაზღვრული მომსახურების ხარისხი priority, custom ან weighted fair queuing მეთოდების გამოყენებით.

1.3.2 წესებზე დაფუძნებული მარშრუტიზაციის გამართვა

წდმ დაერთვება ინტერფეისის შემომავალ პაკეტებს. იმ ინტერფეისზე, რომელზეც წდმ არის აქტიური, ყველა მიღებული პაკეტი ექვემდებარება წესებისადმი შემოწმებას. მარშრუტიზატორი ახორციელებს პაკეტების ფილტრში გატარებას, რომელსაც ეწოდება მარშრუტის რუკა (route map). მარშრუტის რუკაში მითითებული კრიტერიუმების მიხედვით

ხორციელდება პაკეტების მარშრუტიზაცია ან გადამისამართება შემდგომი მარშრუტიზატორისაკენ.

1.3.3 მარშრუტის წესების რუკა

მარშრუტის რუკის ყველა ჩანაწერი შედგება match და set ნაწილისაგან. პუნქტი match განსაზღვრავს იმის კრიტერიუმს, თუ როგორი პაკეტი შეესაბამება ამ ჩანაწერის პირობას. ამის გარდა, set პუნქტი განსაზღვრავს ამ პაკეტის მიმართ შესასრულებელ მოქმედებას. იმ შემთხვევაში თუ კი მარშრუტის რუკის ერთ ჩანაწერში რამოდენიმე match პუნქტია, საჭიროა რომ შესრულდეს ყველა ეს პირობა იმისათვის, რომ განხორციელდეს set პუნქტში აღწერილი მოქმედება. ამის გარდა, მარშრუტის რუკის ჩანაწერებს გააჩნიათ permit ან deny პარამეტრი. თუ ჩანაწერი შეიცავს deny პარამეტრს, ის პაკეტები რომლებიც აკმაყოფილებენ ამ ჩანაწერის კრიტერიუმს იქნებიან გადამისამართებულნი სტანდარტული მარშრუტიზაციის ცხრილის მიხედვით, ე.ი. ამ ჩანაწერის პუნქტი set იქნება უგულვებელყოფილი. მაშასადამე, set პუნქტის მოქმედება შესრულდება მხოლოდ მაშინ, როდესაც მარშრუტიზაციის რუკის ჩანაწერს გააჩნია permit პარამეტრი და match პუნქტის პირობა შესრულებულია.

<u>პუნქტი match - კრიტერიუმის განსაზღვრა</u>

პუნქტი match კრიტერიუმის ასაღწერად შეიძლება იყოს გამოყენებული სტანდარტული ან გაფართოებული ACL. სტანდარტული ACL გამოიყენება იმ შემთხვევაში, თუ კრიტერიუმად საჭიროა წყაროს IP მისამართის მითითება. მაშინ, როდესაც გაფართოვებული ACL-ის გამოყენებით შესაძლებელი ხდება აპლიკაციის, პროტოკოლის ტიპის, TOS ველის ან precedence მნიშვნელობის მითითება. პუნქტი match–ის კრიტერიუმად ასევე შესაძლებელია გამოიყენოს პაკეტის ზომის მნიშვნელობა, კერძოთ პაკეტის მინიმალური და მაქსიმალური ზომა. ამის შედეგად ქსელის ადმინისტრატორს შეუძლია ინტერაქტიული აპლიკაციების პაკეტების განსხვავება (ასეთი პაკეტები ჩვეულებრივ მცირე

ზომის არიან) დიდი მოცულობის მქონე პაკეტებისაგან. წესებზე დაფუძნებული მარშრუტიზაცია აგრძელებს პაკეტის პარამეტრების შემოწმებას იქამდე, სანამ რომელიმე match პუნქტის პირობა არ დაკმაყოფილდება. იმ შემთხვევაში, თუ მარშრუტის რუკაში არც ერთი დამთხვევა არ იქნა დადგენილი ან შესატყვისი ჩანაწერი შეიცავს deny პარამეტრს, პაკეტის მარშრუტიზაცია განხორციელდება სტანდარტული მარშრუტიზაციის ცხრილის მიხედვით, ე.ი. დანიშნულების მისამართის მიხედვით.

<u>პუნქტი set - მარშრუტის მითითება</u>

თუ პუნქტი match დაკმაკოფილებულია, ამის შემდეგ მარშრუტიზატორი ამოწმებს set პუნქტით მითითებული შესასრულებელი მოქმედებების ნუსხას. საჭიროა აღინიშნოს, რომ ქვემოთ ჩამოთვლილი კრიტერიუმების თანამიმდევრობას აქვს მნიშვნელობა და ისინი გამოიყურებიან შემდეგნაირად.

1. ინტერფეისების ნუსხა – იმ ინტერფეისების ნუსხა, რომლების მეშვეობით შესაძლებელია პაკეტის მარშრუტიზირება. თუ კი არის მითითებული ერთზე მეტი ინტერფეისი, პაკეტების გასაგზავნად მარშრუტიზატორი აირჩევს სიაში პირველ გამართულ ინტერფეისს.
2. IP მისამართების ნუსხა – შემდგომი მარშრუტიზატორის მისამართები, რომლებიც შეიძლება იყვნენ გამოყენებული პაკეტების შემდეგ გასაგზავნად. თუ კი ერთზე მეტი IP მისამართი არის მითითებული, პაკეტების გასაგზავნად მარშრუტიზატორი აირჩევს იმ სიაში პირველ მისამართს, რომლის შესაბამისი ინტერფეისი არის გამართულ მდგომარეობაში.
3. Default ინტერფეისების ნუსხა – იმ შემთხვევაში, თუ მოცემული პაკეტის დანიშნულების მისამართისათვის მარშრუტიზაციის ცხრილში არ მოიძებნება შესაბამისი მარშრუტი, მაშინ წდმ გადაამისამართებს ამ პაკეტს default ინტერფეისების ნუსხაში მითითებული პირველი გამართული ინტერფეისის მიმართულებით.

4. Default შემდგომი მარშრუტიზატორის IP მისამართების ნუსხა – თუ კი მარშრუტიზაციის ცხრილში არ მოიძებნება მოცემული პაკეტის დანიშნულების მისამართისათვის შესაბამისი ჩანაწერი, პაკეტი იქნება გადამისამართებული ამ set პუნქტში მითითებული ინტერფეისის ან შემდგომი მარშრუტიზატორის მისამართისაკენ.
5. IP TOS ველი – ეს პუნქტი შეიძლება იყოს გამოყენებული იმისათვის, რომ მიღებული პაკეტის TOS ველში ჩაიწეროს სასურველი მნიშვნელობა, რის შედეგად შესაძლებელი გახდება შესაბამისი მომსახურების ხარისხის უზრუნველყოფა.
6. IP Precedence – ეს პუნქტი შეიძლება იყოს გამოყენებული იმისათვის, რომ მიღებული პაკეტის IP Precedence ველში ჩაიწეროს სასურველი მნიშვნელობა, რის შედეგად შესაძლებელი გახდება შესაბამისი მომსახურების ხარისხის უზრუნველყოფა.

საჭიროა აღინიშნოს, რომ შესაძლებელია ზემოთხსენებული set პუნქტების ერთად გამოყენება. ასევე, პუნქტში set მითითებული შემდგომი მარშრუტიზატორი უნდა იყოს ხელმისაწვდომი, ე.ი. გამართული და საერთო ქვექსელის მქონე IP მისამართით.

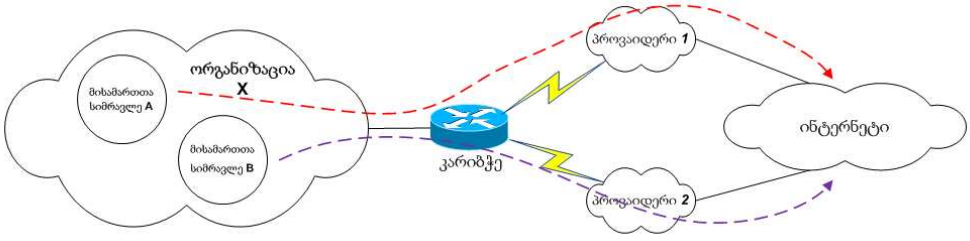
იმ შემთხვევაში, თუ პაკეტები არ აკმაყოფილებენ არც ერთ match პუნქტის პირობას, მაშინ ასეთი პაკეტების მარშრუტიზაცია განხორციელდება სტანდარტული მარშრუტიზაციის ცხრილის გამოყენებით, ე.ი. დანიშნულების მისამართისამებრ. თუ კი ეს მოქმედება არ არის სასურველი და აუცილებელია მსგავსი პაკეტების გაუქმება, მაშინ set პუნქტის ინტერფეისების ნუსხაში საჭიროა ბოლო ჩანაწერის ნაცვლად მიეთითოს Null0 ინტერფეისი. ამ შემთხვევაში, ნებისმიერი პაკეტი, რომელიც არ დააკმაყოფილებს მარშრუტიზაციის რუკაში match პუნტებში აღწერილ არც ერთ პირობას, იქნებიან გაუქმებულნი. [19]

წესში მითითებული მარშრუტი შეიძლება იყოს განსხვავებული იმისაგან, თუ რომელი იყო არჩეული დინამიკური მარშრუტიზაციის პროტოკოლის მიერ. ამიტომ, წდმ წარმოადგენს მარშრუტიზაციის

ალტერნატიულ მექანიზმს იმ შემთხვევებში, როდესაც აუცილებელია გარკვეული პაკეტების (პაკეტის წყაროს მისამართის, ზომის ან შიგთავსის მიხედვით) გადამისამართება ალტერნატიული მარშრუტით. ასევე მნიშვნელოვანია აღინიშნოს, რომ ერთი და იგივე დანიშნულების მისამართის მქონე პაკეტები შეიძლება იყვნენ მიმართულნი სხვადასხვა მარშრუტებისაკენ, რაც გარკვეულ სიტუაციებში წარმოადგენს გადამწყვეტ ფაქტორს. ეს ფუნქციონალური თვისება სხვადასხვა აპლიკაციების ნაკადების განცალკევების შესაძლებლობას ქმნის.

მაგალითად, მიზანშეწონილად შეიძლება ჩაითვალოს ორგანიზაციის ბიზნეს აპლიკაციების ნაკადების განცალკევება სხვა დანარჩენი ნაკადებისაგან. იმ შემთხვევაში, თუ ორივე ტიპის ნაკადის დანიშნულების მისამართი წარმოადგენს ერთი და იგივე ქსელს, კლასიკურ მარშრუტიზაციაში ასეთი ნაკადების განცალკევება შეუძლებელი იქნებოდა. წდმ წარმოადგენს ამ პრობლემის ნაწილობრივ გადაწყვეტას, მაგრამ მნიშვნელოვანია აღინიშნოს, რომ წდმ არის მთლიანად დაყრდნობილი ქსელის ადმინისტრატორის მიერ შექმნილ წესებზე და კონფიგურაციებზე და არ წარმოადგენს დინამიკურ პროტოკოლს. [20,21]

ნახ. 1.7–ზე წარმოდგენილია ორგანიზაცია X–ის მაგალითი, სადაც განხორციელებულია აპლიკაციების ნაკადების განცალკევება. მაშასადამე, როდესაც ნაკადის წყაროს მისამართი წარმოადგენს მისამართთა სიმრავლეს A, კარიბჭე ახორციელებს ამ ნაკადის გადამისამართებას პროვაიდერი 1–საკენ. ამავე დროს, როდესაც ნაკადის წყაროს მისამართი წარმოადგენს



ნახ. 1.7 აპლიკაციების ნაკადების განაწილება ორგანიზაცია X მაგალითზე

მისამართთა სიმრავლეს B, კარიბჭე ახორციელებს ამ ნაკადის გადამისამართებას პროვაიდერი 2–საკენ.

1.3.4 მომსახურების ხარისხის კონტროლი

წდმ მიერ პაკეტების ტეგირების მეშვეობით, ქსელის ადმინისტრატორს შეუძლია ქსელში არსებული ნაკადების სხვადასხვა ჯგუფებად(კლასებად) დაყოფა. ამის შედეგად, შესაძლებელი ხდება ამ კლასებისათვის სხვადასხვა მომსახურების ხარისხის უზრუნველყოფა. ამის განსახორციელებლად შეიძლება იყოს გამოყენებული priority, custom ან weighted fair queuing ტექნოლოგიები. მაშასადამე, ქსელის შესასვლელთან განხორციელებული პაკეტების კლასიფიცირება თავიდან აშორებს ქსელის შიგნით შემდგომი პაკეტების კლასიფიცირების აუცილებლობას. ამის შედეგან, ტეგირებული პაკეტები იმის საშუალებას იძლევიან, რომ მოხდეს ქსელში მომუშავე აპლიკაციების ხარისხის გარანტია და კონტროლი. ეს პროცესი არის აღწერილი ნახ. 1.8–ზე.

ნახ.1.8 მომსახურების ხარისხის კონტროლი ტეგირებული პაკეტებისათვის

შემდეგ თავში განხილულია კლასიკური მარშრუტიზაციის პრობლემები და შემუშავებულია მათი მოგვარების მეთოდები.

I თავის დასკვნა

1. თანამედროვე გლობალური ქსელების ამოცანებისა და პრობლემების გადაწყვეტის მიზნით გამოყენებული ალგორითმებისადმი მიძღვნილი ლიტერატურის მიმოხილვა და ანალიზი გვიჩვენებს, რომ მარშრუტიზაციის არსებული პროტოკოლები ირჩევენ ერთ საუკეთესო მარშრუტს ნაკადის ადრესატამდე და ამ მიმართულებით აგზავნიან ყველა ტიპის აპლიკაციების პაკეტებს.
2. შედეგად ვიღებთ იმას, რომ მარშრუტიზაციის პროტოკოლის მიერ არჩეული საუკეთესო მარშრუტი არ არის ერთნაირად საუკეთესო ყველა აპლიკაციისათვის და ამით ზოგიერთი აპლიკაციის კავშირის ხარისხი უარესდება.
3. ჩატარებულია კლასიკური მარშრუტიზაციის მეთოდების ანალიზი.
4. ნაჩვენებია, რომ ეს პრობლემა წარმოადგენს ერთ-ერთ მთავარ პრობლემას ქსელში აპლიკაციების მუშაობის მაღალი ხარისხის უზრუნველყოფისას.

შედეგები და მათი განსჯა

თავი II

ინტელექტუალური მარშრუტიზაციის მოდელი

ამ თავში აგებულია ინტელექტუალური მარშრუტიზაციის მოდელი. განხილულია მომსახურების ხარისხის ფუნდამენტალური მახასიათებლების კონტროლის მეთოდები. დადგენილია, რომ არსებული დინამიკური მარშრუტიზაციის პროტოკოლები დანიშნულების მისამართისათვის ოპტიმალური მარშრუტის არჩევაში არიან შეზღუდული.

ჩამოყალიბებულია დღევანდელი გლობალური ქსელების პრობლემები და შემუშავებულია ამ პრობლემების გადაწყვეტის მეთოდები. გაანალიზებულია თითოეული მათგანი. შემუშავებულია ალგორითმი, რომელსაც საფუძვლად უდევს ძირითადი მოთხოვნა, რომ მარშრუტიზატორებს უნდა შეეძლოთ ნაკადების მარშრუტიზაცია არა მარტო დანიშნულების პრეფიქსების მიხედვით, არამედ აპლიკაციების ტიპებზე დაყრდნობით. მარშრუტიზაცია ირჩევს აპლიკაციისათვის საუკეთესო მარშრუტს და არა მხოლოდ დანიშნულების პრეფიქსისათვის. კრიტიკული აპლიკაციების მარშრუტიზაცია ხორციელდება ძირითადი (მაღალხარისხიანი) არხის გამოყენებით, რაც ხელს უწყობს აპლიკაციების მაღალი ხარისხის უზრუნველყოფას. ამავე დროს, დანარჩენი სტანდარტული აპლიკაციების ნაკადები იყენებენ სარეზერვო არხებს. საჭიროა აღინიშნოს, რომ ძირითადი არხის დაზიანების შემთხვევაში, სისტემას შეუძლია კრიტიკული აპლიკაციების ნაკადების გადამისამართება სარეზერვო არხებისაკენ.

ამის შედეგად, ქსელში უეცარი ცვლილებების შემთხვევაში, მარშრუტიზატორები ახდენენ სწრაფ რეაგირებას და ხორციელდება ნაკადების გადანაწილება სასურველი მიმართულებებით.

შემუშავებული არლგორითმი იყენებს ტრაფიკის მართვის ინტელექტუალურ მეთოდს, რომელსაც შეუძლია დინამიკური მარშრუტიზაციის გადაწყვეტილებების მიღება ქსელის არხების

შეცდომებზე დაყრდნობით და ასევე მარშრუტიზაციის კორექტირება კომპლექსურ კრიტერიუმების გათვალისწინებით. ასეთ კრიტერიუმებს შეიძლება წარმოადგენდნენ პაკეტების მიწოდების დაყოვნება, პაკეტების დანაკარგი, დაყოვნების ვარიაციის ფაქტორი, არხების დატვირთულობა და სხვა წესები. ინტელექტუალური მარშრუტიზაციის პროტოკოლი აუმჯობესებს ტრადიციულ მარშრუტიზაციას (BGP, OSPF, და წდმ) წარმადობაზე გავლენის მქონე პარამეტრების რეალურ რეჟიმში კონტროლით.

შემუშავებული მოდელების საფუძველზე აგებულია ალგორითმი, რომელიც ითვალისწინებს არსებული მარშრუტების მომსახურების ხარისხის მახასიათებლების მნიშვნელობებს და ახორციელებს ნაკადების ჯგუფებად დაყოფას. ჯგუფების შექმნა ხორციელდება აპლიკაციების პრიორიტეტული მახასიათებლის მიხედვით.

2.1 კლასიკური მარშრუტიზაციის ნაკლოვანობები

არსებული დინამიკური მარშრუტიზაციის პროტოკოლები დანიშნულების მისამართისათვის ოპტიმალური მარშრუტის არჩევაში არიან შეზღუდული. ამ პრობლემის აქტუალურობის უკეთესად გასაგებად, საჭიროა განვიხილოთ მომსახურების ხარისხის მოთხოვნები სხვადასხვა ტიპის აპლიკაციებისათვის. ამ აპლიკაციების ნაკადების მოთხოვნები შეიძლება დაიყოს ოთხ ძირითად ნაწილად:

- საიმედოობა
- დაყოვნება
- დაყოვნების ვარიაცია
- გამტარუნარიანობა

ამ პარამეტრების ერთობლიობა განსაზღვრავს ნაკადისათვის აუცილებელ ქსელის მომსახურების ხარისხს (QoS). უმეტეს შემთხვევაში, აპლიკაციების ფართო სპექტრი მოითხოვს მაღალი წარმადობის მქონე ქსელს იმისათვის, რომ დაამყაროს კავშირი ჰოსტი–სერვერის ან peer-to-peer

ტიპის კომუნიკაციაში. კორპორაციული აპლიკაციების ნაწილი არის ქსელში პაკეტების მიწოდების დაყოვნებისადმი მგრძნობიარე, სხვა აპლიკაციები კი მაღალი გამტარუნარიანობის მომთხოვნი.

ამის მაგალითს წარმოადგენს იმ ტიპის აპლიკაციები, რომლებიც გადასცემენ ხმოვან ინფორმაციას(VoIP). ასეთი აპლიკაციები არიან განსაკუთრებულად მგრძნობიარე დაყოვნებისადმი. ამის საპირისპიროთ, ფაილების გადაცემის აპლიკაციები არ არიან იმდენად მგრძნობიარე დაყოვნებისადმი, მაგრამ მოითხოვენ შედარებით დიდ გამტარუნარიანობას.

ამ ორი ტიპის აპლიკაციებს გააჩნიათ მომსახურების ხარისხისადმი საპირისპირო მოთხოვნები და ამის გამო მათი ნაკადების დამუშავება უნდა ხორციელდებოდეს ინდივიდუალური მიდგომით. როგორც ვხედავს, ყოველივე აპლიკაციას გააჩნია თავისი QoS მოთხოვნები. ეს გარემოება შეიძლება ნიშნავდეს იმას, რომ დინამიკური მარშრუტიზაციის მიერ არჩეული საუკეთესო მარშრუტი არ აღმოჩნდეს ერთნაირად საუკეთესო ქსელში მომუშავე სხვადასხვა ტიპის აპლიკაციებისათვის.

იმის მიუხედავად, რომ ყველა დინამიკური მარშრუტიზაციის პროტოკოლს გააჩნია თავისი გადაწყვეტილების მიღების ალგორითმი, მაგრამ ყველა მათგანის ძირითად მიზანს წარმოადგენს ერთი და იგივე ამოცანა. ეს ამოცანა მდგომარეობს საუკეთესო მარშრუტის არჩევაში. ამის შემდეგ, მარშრუტიზატორი იყენებს ამ მარშრუტს იმისათვის, რომ განახორციელოს კერძო დანიშნულების მისამართისაკენ ყველა ტიპის ნაკადების პაკეტების გადაგზავნა. ამგვარად, ეს ნიშნავს იმას, რომ მარშრუტიზატორს არ შეუძლია სხვადასხვა ტიპის ნაკადების გარჩევა, ე.ი. შეუძლებელია დადგინდეს ფაილების გადაცემის ნაკადსა და VoIP ტიპის ნაკადებს შორის განსხვავება.

მაშასადამე, ხშირ შემთხვევაში ადგილი აქვს პრობლემას, რომელიც მდგომარეობას იმაში, რომ დინამიკური მარშრუტიზაციის პროტოკოლის მიერ არჩეული საუკეთესო მარშრუტი წარმოადგენს ოპტიმალურ არჩევანს

ერთი აპლიკაციისათვის, მაგრამ სხვა აპლიკაციისათვის ეს კერძო მარშრუტი არ წარმოადგენს საუკეთესო არჩევანს სხვა არსებულ მარშრუტებთან შედარებით.

ამ პრობლემის გადასაწყვეტად იყო შემუშავებული ალგორითმი, რომელიც ირჩევს საუკეთესო მარშრუტს ყველა აპლიკაციების ნაკადების სპეციფიკის გათვალისწინებით. ეს ალგორითმი იყენებს მომსახურების ხარისხის (QoS) პარამეტრების გაზომვის ხელსაწყოებს და ახორციელებს ფუნდამენტალური მახასიათებლების მნიშვნელობების დადგენას ნაკადის წყაროს მისამართიდან დანიშნულების კვანძამდე. ამის შედეგად, ქსელში მომუშავე ყველა აპლიკაციისათვის ხორციელდება საუკეთესო მარშრუტის ინდივიდუალური შერჩევა. მაგალითად, ალგორითმის მუშაობის შედეგად, VoIP ტიპის აპლიკაციების ნაკადები გადამისამართდებიან იმ მარშრუტისაკენ, რომელსაც გააჩნია ყველაზე დაბალი დაყოვნება დანიშნულების კვანძამდე. ამავე დროს, ფაილის გადაგზავნის აპლიკაციების ნაკადები გადამისამართდებიან იმ არხისაკენ, რომელსაც გააჩნია ყველაზე მაღალი გამტარუნარიანობა.

იმის შემდეგ, რაც პროტოკოლმა მოახდინა საუკეთესო მარშრუტის შერჩევა, ის აგრძელებს თითოეული მარშრუტის მდგომარეობის შემოწმებას. იმ შემთხვევაში, თუ კი რომელიმე კრიტიკული პარამეტრი შეიცვლება და გადააჭარბებს დასაშვებ ლიმიტს, პროტოკოლი კვლავ დაბრუნდება საუკეთესო მარშრუტის შერჩევის ფაზაში და მოახდენს ახალი მარშრუტის არჩევას მიმდინარე გარემოებების გათვალისწინებით.

ამ ალგორითმის მუშაობის შედეგად, ქსელს ეძლევა იმის საშუალება, რომ მუდმივად უზრუნველყოს ხელმისაწვდომი მიმართულებებისაგან საუკეთესო მომსახურების ხარისხის მქონე მარშრუტების გამოყენება და ამის განხორციელება ყველა აპლიკაციისათვის ინდივიდუალურად. ასევე, ფუნდამენტალური მახასიათებლების მუდმივი კონტროლის შედეგად მოახდინოს ქსელის ფლუქტუაციების შემთხვევაში სწრაფი რეაგირება.

2.2 მაღალი წარმადობის მქონე მარშრუტიზაციის მოდელი

თანამედროვე ქსელებში, ბევრ ორგანიზაციებს გააჩნიათ ერთზე მეტი არხი, რომლების მეშვეობით ისინი ახორციელებენ გარე სამყაროსთან კავშირს. ეს არხები წარმოადგენენ ორგანიზაციების კერძო საკუთრებას ან წარმოადგენენ ინტერნეტ პროვაიდერის ინფრასტრუქტურის ნაწილს. ბევრ შემთხვევაში, ერთერთი არხი წარმოადგენს ძირითად არხს, მაგრამ დანარჩენი არხები არიან დამხმარე. ასეთ შემთხვევებში, ძირითადი დატვირთვა მოდის პირველად არხზე, როდესაც დამხმარე არხების რესურსები იმყოფებიან გამოყენების გარეშე.

დღევანდელი გლობალური ქსელების პრობლემები გამოიხატება შემდეგში:

- ტრაფიკის ზრდა და ცენტრალური ქსელის ინტერნეტთან დამაკავშირებელი არხების დატვირთვის გაწონასწორების მოუქნელი ალგორითმები
- ძვირადღირებული დამხმარე არხების რესურსების გამოუყენებლობა
- დატვირთვის განაწილება BGP-ის რთული მექანიზმებით და წდმ–ით [22, 23]
- NetFlow სტატისტიკის, IP SLA და პრეფიქსების ხელით შემოწმების აუცილებლობა [24]

ამ პრობლემების გადაწყვეტის საშუალებებად შეიძლება ჩაითვალოს:

- ძვირადღირებული არხების და ინფრასტრუქტურის სრული გამოყენება
- ტრაფიკის ეფექტური განაწილება არხების დატვირთვის გათვალისწინებით
- ტრაფიკის ოპტიმიზირება არხების საფასურის პროფილებზე დაყრდნობით
- ძვირადღირებული გამოუყენებელი არხების მინიმიზირება

2.2.1 აპლიკაციების წარმადობის გაზრდა

ბევრ ორგანიზაციას ესაჭიროება ქსელში არსებული ფლუქტუაციების და პაკეტების დაკარგვის სტატისტიკა. მარშრუტიზაციის პროტოკოლებს არ გააჩნიათ ასეთი მოვლენების შესახებ ინფორმაცია. ამ ფაქტმა შეიძლება გამოიწვიოს მომხმარებლის ქსელში მუშაობის ეფექტურობის გაუარესება და აპლიკაციების წარმადობის დეგრადაცია. საჭიროა აღინიშნოს, რომ ასეთ შემთხვევებში, მარშრუტიზაციის პროტოკოლები აგრძელებენ ჩვეულებრივ მუშაობას და როგორც წესი მსგავსი პრობლემები არანაირ მინიშნებას არ ახდენენ დინამიკური მარშრუტიზაციის პროტოკოლებზე. შედეგად ვღებულობთ:

- სრულ გაუარესებას: 100% გადაცემული ინფორმაციის დაკარგვას იმ დროს, როდესაც სამართავი ბლოკი სრულად გამართულია
- ნაწილობრივ გაუარესებას: შედარებით მაღალი პაკეტების დანაკარგის ფაქტორს და პაკეტების გადაცემის მაღალ დაყოვნებას

ეს პრობლემები თავისმხრივ ზეგავლენას ახდენენ აპლიკაციების წარმადობაზე და ხელს უწყობენ მის დეგრადაციას, რის შედეგად მომხმარებლის მუშაობის ეფექტურობა უარესდება და ამავე დროს მარშრუტიზაციის პრობლემების არსებობა არსად არ ფიქსირდება. ამ პრობლემის მიზეზი შეიძლება იყოს, როგორც გლობალურ ქსელთან დამაკავშირებელი არხების გადატვირთვა, ასევე ინტერნეტ პროვაიდერის ავტონომიურ სისტემაში* მიმდინარე პრობლემები.

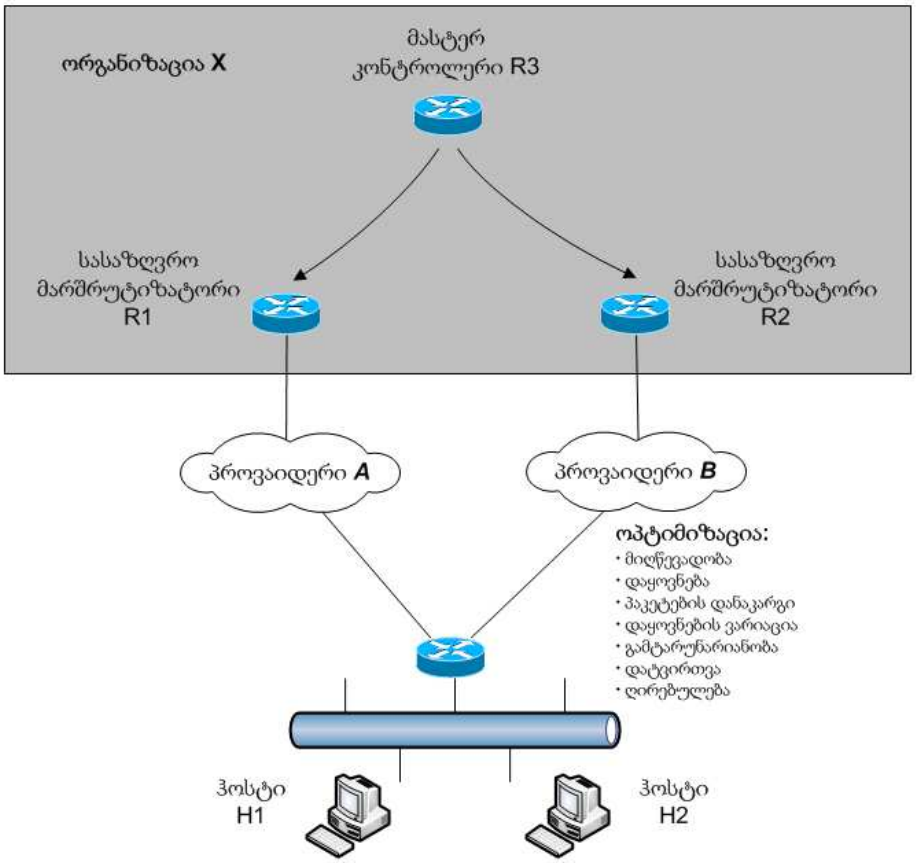
მსგავსი პრობლემის გადაწყვეტას წარმოადგენს ინტელექტუალური მარშრუტიზაციის ალგორითმების დანერგვა, რომელიც დაყრდნობილია აპლიკაციების წარმადობის მონიტორინგზე და მომხმარებლის აპლიკაციების წარმადობის გაუარესების შემთხვევაში სწრაფ რეაგირებაზე. ამის მისაღწევად, საჭიროა აპლიკაციების მომსახურების ხარისხის კრიტიკული პარამეტრების მონიტორინგი. ასეთ პარამეტრებს

წარმოადგენენ პაკეტების გადაცემის დაყოვნება, პაკეტების დანაკარგი, დაყოვნების ვარიაცია და გამტარუნარიანობა.

ამ ალგორითმს საფუძვლად უდევს ძირითადი მოთხოვნა, რომელიც მდგომარეობს იმაში, რომ მარშრუტიზატორებს უნდა შეეძლოთ ნაკადების მარშრუტიზაცია არა მარტო დანიშნულების პრეფიქსების მიხედვით, არამედ აპლიკაციების ტიპებზე დაყრდნობით. მარშრუტიზაციამ უნდა აირჩიოს აპლიკაციისათვის საუკეთესო მარშრუტი და არა მხოლოდ დანიშნულების პრეფიქსისათვის. კრიტიკული აპლიკაციების მარშრუტიზაცია უნდა ხორციელდებოდეს ძირითადი (მაღალხარისხიანი) არხის გამოყენებით, რაც ხელს შეუწყობს აპლიკაციების მაღალი ხარისხის შენარჩუნებას. ამავე დროს, დანარჩენი სტანდარტული აპლიკაციების ნაკადები უნდა იყენებდნენ სარეზერვო არხებს. საჭიროა აღინიშნოს, რომ ძირითადი არხის დაზიანების შემთხვევაში, სისტემამ უნდა შეძლოს კრიტიკული აპლიკაციების ნაკადების გადამისამართება სარეზერვო არხებისაკენ. ამის მისაღწევად შეიძლება იყოს გამოყენებული, როგორც EEM სკრიპტები, ასევე წდმ და სტატიკური მარშრუტიზაცია. [25, 26]

მაგრამ, ეს მეთოდები არ წარმოადგენენ ავტომატურ დამოუკიდებელ მექანიზმებს და ითვალისწინებენ ადმინისტრატორის ჩარევას და პარამეტრების ხელით კონფიგურირებას. ამის გამო, ქსელში უეცარი ცვლილებების შემთხვევაში, მარშრუტიზატორები ვერ მოახდენენ სწრაფ რეაგირებას და მხოლოდ ადმინისტრატორის ჩარევის შემდეგ შესაძლებელი გახდება ნაკადების გადანაწილება სასურველი მიმართულებებით. ამ პრობლემის უფრო მოქნილ გადაწყვეტას წარმოადგენს პროტოკოლი Performance Routing.

ტრადიციული მარშრუტიზაცია იყენებს სტატიკურ მეტრიკას იმისათვის, რომ დაადგინოს თუ პრეფიქსი წარმოადგენს ხელმისაწვდომ მიმართულებას. საჭიროა აღინიშნოს, რომ ამ შემთხვევაში ტრადიციულ მარშრუტიზაციას არ გააჩნია არანაირი ინფორმაცია არხების დატვირთულობის, მონაცემთა გადაცემის ან არხების შეცდომების შესახებ.



ნახ. 2.1 აპლიკაციების წარმადობის გაზრდა ორგანიზაცია X მაგალითზე

Cisco Performance Routing ან მოკლეთ PfR აუმჯობესებს მარშრუტიზაციის ალგორითმს იმ თვალსაზრისით, რომ საუკეთესო მარშრუტის არჩევა შესაძლებელი ხდება მომხმარებლის წესებზე დაყრდნობით. PfR-ის წესი შეიძლება იყოს დაყრდნობილი, როგორც არხებს შორის დატვირთვის განაწილებაზე ასევე აპლიკაციებისათვის საუკეთესო და მაქსიმალური წარმადობის მქონე მარშრუტის არჩევაზე. [27, 28]

Cisco PfR იყენებს ტრაფიკის მართვის ინტელექტუალურ მეთოდს, რომელსაც შეუძლია დინამიკური მარშრუტიზაციის გადაწყვეტილებების მიღება კორპორაციული ქსელის ან ინტერნეტის არხების შეცდომებზე დაყრდნობით და მარშრუტიზაციის კორექტირება კომპლექსურ კრიტერიუმებზე დაყრდნობით. ასეთ კრიტერიუმებს შეიძლება წარმოადგენდნენ პაკეტების მიწოდების დაყოვნება, პაკეტების დანაკარგი, დაყოვნების ვარიაციის ფაქტორი, არხების დატვირთულობა და სხვა წესები.

Cisco PfR აუმჯობესებს ტრადიციულ მარშრუტიზაციას (BGP, OSPF, EIGRP და წდმ) წარმადობაზე გავლენის მქონე პარამეტრების რეალურ რეჟიმში კონტროლით. ამ ალგორითმის ნაწილს წარმოადგენს NetFlow და IP SLA პროტოკოლები. [29, 30]

2.2.2 პაკეტების დაყოვნების ძირითადი მიზეზები

ქსელში პაკეტების დაყოვნების მიზეზები შეიძლება დაიყოს ოთხ ძირითად ჯგუფად, და ესენი არიან დამუშავების დაყოვნება, ბუფერიზაციის დაყოვნება, გადაგზავნის დაყოვნება და გავრცელების დაყოვნება.

<u>დამუშავების დაყოვნება</u>

დამუშავების დაყოვნებას წარმოადგენს დრო, რომელიც საჭიროა პაკეტი დასამუშავებლად ყოველივე ჰოპზე. პაკეტების კომუტაციის ქსელებში, დამუშავების დაყოვნება შეადგენს დროს, რომელიც ესაჭიროება მარშრუტიზატორს პაკეტის სათაურის დასამუშავებლად. დამუშავების დაყოვნება წარმოადგენს ჯამური დაყოვნების უმნიშვნელოვანეს ნაწილს. დამუშავების დროს, მარშრუტიზატორებს უწევთ პაკეტის სათაურის წაკითხვა, შეცდომების ველის შემოწმება და ასევე პაკეტის დანიშნულების მისამართის დადგენა. მაღალ სიჩქარიან მარშრუტიზატორებზე, დამუშავების დაყოვნება როგორც წესი შეადგენს რამოდენიმე ათეულ მიკრო წამს. მნიშვნელოვანია აღინიშნოს, რომ ზოგიერთ მარშრუტიზატორებზე ეს დაყოვნება შეიძლება იყოს გაცილებით მაღალი. ამის მიზეზს წარმოადგენს მარშრუტიზატორის მიერ განხორციელებული დაშიფრის ან პაკეტების დეტალური შემოწმების ალგორითმები. დამუშავების დაყოვნება შეიძლება იყოს მაღალი ასევე იმ მარშრუტიზატორებზე, რომლებიც ახორციელებენ ქსელის მისამართის თარგმნას, რადგან ამ ამოცანის შესასრულებლად მარშრუტიზატორს უწევს როგორც შემომავალი ასევე გამავალი პაკეტების დამუშავება. ამ ეტაპის შესრულების შემდეგ, მარშრუტიზატორი აგზავნის პაკეტს გამავალი

ინტერფეისის ბუფერში, რომელიც წარმოადგენს დაყოვნების შემდეგ არეალს.

ბუფერიზაციის დაყოვნება

ბუფერიზაციის დაყოვნებას წარმოადგენს დრო, რომლის განმავლობაში პაკეტი იმყოფება მარშრუტიზატორის გამავალი ინტერფეისის ბუფერში. როდესაც პაკეტები აღწევენ მარშრუტიზატორის ინტერფეისს, აუცილებელი ხდება მათი დამუშავება და შემდეგ გადაგზავნა. ბუფერიზაციის დაყოვნება აღინიშნება იმ შემთხვევაში, როდესაც რამოდენიმე შემომავალი ინტერფეისიდან ხორციელდება პაკეტების გადაგზავნა ერთი და იგივე გამავალი ინტერფეისისაკენ. ამ ინტეფეისის გამავალ ბუფერში იქმნება პაკეტების რიგი, ანუ პაკეტები ბუფერდებიან. ყოველი გამავალი ბუფერისათვის, მარშრუტიზატორს ერთდროულად შეუძლია მხოლოდ ერთი პაკეტის დამუშავება ნებისმიერ მომენტში. დანარჩენი პაკეტები უნდა ჩადგნენ გამავალი ინტერფეისის ბუფერის რიგში. როდესაც მარშრუტიზატორი ღებულობს პაკეტებს უფრო ჩქარა ვიდრე მას შეუძლია ამ პაკეტების შემდგომ გადაგზავნა, ასეთ შემთხვევაშიც აუცილებელია პაკეტების ბუფერიზაცია. ბუფერიზაციის დაყოვნების მაქსიმალური დაყოვნება არის ბუფერის ზომის პროპორციული. რაც უფრო დიდია ბუფერში მყოფი პაკეტების რაოდენობა, მით უფრო დიდია საშუალო ლოდინის დრო და ბუფერის გადავსების შემთხვევაში მარშრუტიზატორი ახორციელებს პაკეტების გაუქმებას.

როდესაც პაკეტები ტოვებენ ინტერფეისის ბუფერს, ინტერფეისს ისინი გადაყავს ბიტებში და ხორციელდება მათი სერიალიზაცია ფიზიკურ კაბელში.

გადაგზავნის დაყოვნება

გადაგზავნის დაყოვნება წარმოადგენს დროს, რომელიც საჭიროა პაკეტების ბიტებში გადასაყვანად და შემდგომ ამ ბიტების გადაცემა ელექტრული ან ოპტიკური სიგნალის მეშვეობით. მაშასადამე, ეს დაყოვნება გამოწვეულია ინტერფეისის გადაცემის სიხშირის შეზღუდვით.

გადაგზავნის დაყოვნება წარმოადგენს პაკეტის ზომის ფუნქციას, და არ არის დამოკიდებული ორ კვანძს შორის მანძილზე. ეს დაყოვნება არის პაკეტის ზომის (ბიტებში) პროპორციული.

$D_T = \frac{N}{R}$

სადაც D_T წარმოადგენს გადაგზავნის დაყოვნებას, N არის ბიტების რაოდენობა და R გადაგზავნის სიხშირე.

<u>გავრცელების დაყოვნება</u>

გავრცელების დაყოვნება წარმოადგენს დროს, რომელიც საჭიროა სიგნალის სადენში გადასაგზავნად. ეს შეიძლება იყოს ოპტიკურ–ბოჭკოვანი ან სპილენძის კაბელი, ასევე უკაბელი გადაცემა. სადენის მახასიათებლებს განსაზღვრავს სპილენძში ელექტრონების გავრცელების სიჩქარე, შუშის არეკვლის ფაქტორი და ასე შემდეგ. ეს მახასიათებლები ახდენენ ზეგავლენას გავრცელების დაყოვნების სიდიდეზე. გავრცელების დაყოვნება უდრის d/s სადაც d არის კვანძებს შორის მანძილი და s ტალღების გავრცელების სიჩქარე. უკაბელო კომუნიკაციაში, s=c სადაც c არის შუქის სიჩქარე. სპილენძის კაბელში, როგორც წესი დაყოვნება შეადგენს 0.59-0.77c.

<u>პაკეტების დაყოვნების ვარიაცია (პდვ*)</u>

კვანძთაშორისი დაყოვნების გარდა, ქსელის წარმადობა ასევე განისაზღვრება პაკეტების დაყოვნების ვარიაციით, ან პდვ. კომპიუტერულ ქსელებში, პდვ წარმოადგენს კვანძებს შორის ერთი ნაკადის პაკეტების ერთი გზის დაყოვნებათა სხვაობას. ამასთან ერთად, ამ მახასიათებლის გაზომვისას ხორციელდება დაკარგული პაკეტები უგულვებელყოფა.

პდვ–ს ასევე ეწოდება jitter. ეს მოვლენა წარმოადგენს მეტად მნიშვნელოვან და კრიტიკულ პარამეტრს გარკვეული ტიპის აპლიკაციებისათვის. განსაკუთრებით მნიშვნელოვანია აღინიშნოს ხმის გადაცემის პროტოკოლები და ინტერაქტიული აპლიკაციები. წამიერი პაკეტების დაყოვნების ვარიაციას წარმოადგენს თანამიმდევრულად მიღებული პაკეტების დაყოვნებათა სხვაობას. მაგალითად, განვიხილოთ სიტუაცია, როდესაც იგზავნება პაკეტი ყოველ 20 მილიწამში ერთხელ. თუ კი მეორე პაკეტი იყო მიღებული პირველი პაკეტის 30 მლ.წმ. შემდეგ, მაშინ

პდვ=–10მლ.წმ. ამ შემთხვევას ეწოდება დისპერსია. თუ კი მეორე პაკეტი იყო მიღებული პირველი პაკეტის 10მლ.წმ. შემდეგ, მაშინ პდვ=+10მლ.წმ. და ამას ეწოდება “clumping”-ი.

ორი გზის დაყოვნება

ქსელის წარმადობის დახასიათებისას გამოიყენება კიდევ ერთი ტერმინი ორი გზის დაყოვნება, ან RTT*. ეს მახასიათებელი წარმოადგენს იმ დროს, რომელის საჭიროა დანიშნულების მისამართამდე პაკეტის გასაგზავნად და მისგან საპასუხო პაკეტის მისაღებად. დიდი მასშტაბების ქსელებში და ასევე ინტერნეტში, ორი გზის დაყოვნება წარმოადგენს დაყოვნების ერთერთ ძირითად ფაქტორს, რომელიც ახდენს ზემოქმედებას მომხმარებლის აპლიკაციების ხარისხზე. კერძოდ, ეს პარამეტრი დიდ ზეგავლენას ახდენს მომხმარებლის მიერ გაგზავნილი მოთხოვნის და ამ მოთხოვნაზე პასუხის მიღების სისწრაფეზე. RTT–ს მნიშვნელობა შეიძლება წარმოადგენდეს რამოდენიმე მილიწამს, მაშინ როდესაც ორი კვანძი ძალიან ახლოს იმყოფებიან ერთი მეორისაგან, ან რამოდენიმე ასეულ მილიწამს, იმ შემთხვევაში, როდესაც პაკეტებს უწევთ კონტინენტებს შორის მანძილის გადალახვა. RTT-ს მნიშვნელობის თეორიული მინიმუმი წარმოადგენს იმ დროს, რომელიც ესაჭიროება სიგნალს რომ გადალახოს მოცემული მანძილი.

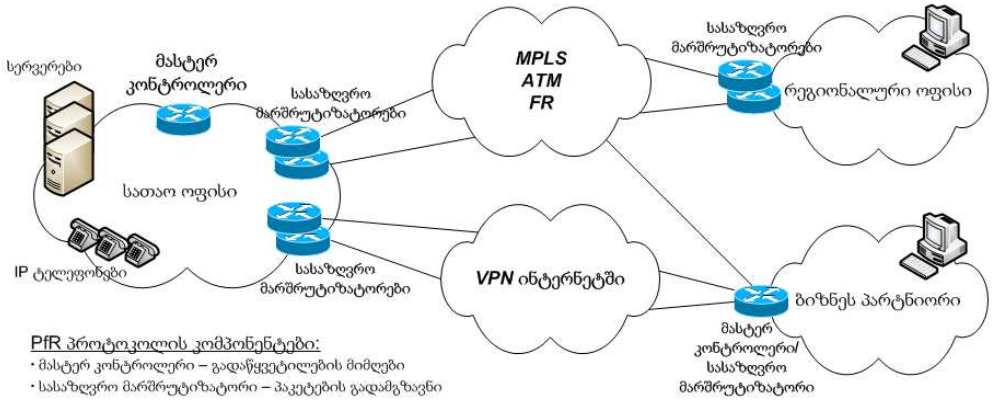
2.3 PfR ალგორითმის მოდელი

PfR პროტოკოლის ნებისმიერ რეალიზაციაში აუცილებელია არსებობდეს მასტერ კონტროლერი და სულ მცირე ერთი სასაზღვრო მარშრუტიზატორი. მასტერ კონტროლერი თავის თავზე იღებს სასაზღვრო მარშრუტიზატორის კონტროლს და ბრძანებათა მიწოდებას. ამის გარდა, მასტერ კონტროლერი ღებულობს და ინახავს სასაზღვრო მარშრუტიზატორებისაგან მიღებულ ინფორმაციას. მასტერ კონტროლერის და სასაზღვრო მარშრუტიზატორის ფუნქციები შეიძლება იყვნენ

შეთავსებული, როგორც ერთ ფიზიკურ მოწყობილობაში ასევე ორ ცალკეულ მარშრუტიზატორში.

გამავალი ინტერფეისების არჩევის კრიტერიუმებს წარმოადგენენ მიღწევადობა, დაყოვნება, პაკეტების დანაკარგი, დაყოვნების ვარიაცია, დატვირთვა და ღირებულება.

სასაზღვრო მარშრუტიზატორები მდებარეობენ მომხმარებლის ნაკადების მარშრუტის გასწვრივ. ისინი აწარმოებენ ამ ნაკადების სტატისტიკის დაგროვებას NetFlow პროტოკოლის გამოყენებით და ასევე IP SLA პროტოკოლის მიერ აქტიური სინჯების შედეგების შეჯამებით. [31]



ნახ.2.2 PfR პროტოკოლის ზოგადი სქემა

ამ ინფორმაციის შეგროვების და ანალიზის შედეგად, ეს პროტოკოლი აწარმოებს მომხმარებლის ნაკადებისათვის საუკეთესო მარშრუტის არჩევას და ამ ნაკადების შესაბამისი მიმართულებებით გადამისამართებას.

მასტერ კონტროლერი წარმოადგენს წესებზე დაფუძნებულ, გადაწყვეტილებების მიმღებ კვანძს. როგორც წესი, დიდ ქსელში, მასტერ კონტროლერის ფუნქციას ასრულებს ცალკე მდებარე მარშრუტიზატორი მაშინ, როდესაც შედარებით მცირე ზომის ქსელში მასტერ კონტროლერი და სასაზღვრო მარშრუტიზატორი წარმოადგენენ ერთი და იგივე მოწყობილობას. ამის მიზეზს წარმოადგენს ის პირობა, რომ მსხვილ ქსელში არსებობს გაცილებით მეტი ქსელის პრეფიქსი და შესაბამისად მარშრუტიზატორებს უწევთ მეტი აპლიკაციების ნაკადების დამუშავება, ვიდრე მცირე ქსელებში. ეს გარემო თავის მხრივ ზეგავლენას ახდენს

მასტერ კონტროლერის ფუნქციის შემსრულებელი მარშრუტიზატორის პროცესორის დატვირთვაზე და მისი მეხსიერების მეტი მოცულობის საჭიროებაზე. ამის გამო, მსგავს სიტუაციებში მეტად ეფექტურია მასტერ კონტროლერის განცალკევება სასაზღვრო მარშრუტიზატორისაგან.

როგორც წესი, მცირე ქსელებში, მაგალითად რეგიონალურ ოფისებში, ქსელში არსებობს ნაკლები პრეფიქსი და შესაბამისად ნაკლები აპლიკაციების ნაკადები. ამის გარდა, ცალკეული მარშრუტიზატორის მაღალი ღირებულების გამო, მიზანშეწონილად შეიძლება ჩაითვალოს მასტერ კონტროლერის და სასაზღვრო მარშრუტიზატორის ფუნქციების შეთავსება ერთი და იგივე ფიზიკურ მოწყობილობაში. აუცილებელია სრულდებოდეს მასტერ კონტროლერის ფუნქციის მქონე მარშრუტიზატორის პროცესორის და მეხსიერების გამოყენების მონიტორინგი და სტატისტიკის დაგროვება. ამ შედეგების ანალიზის საფუძველზე შესაძლებელი იქნება მარშრუტიზატორის დატვირთვის კონტროლი და იმ შემთხვევაში თუ ეს დატვირთვა აჭარბებს დაშვებულს, აუცილებელია მასტერ კონტროლერის და სასაზღვრო მარშრუტიზატორის ფუნქციების განცალკევება სხვადასხვა მარშრუტიზატორებში.

მასტერ კონტროლერი აწარმოებს სასაზღვრო მარშრუტიზატორთან მუდმივ კავშირს იდენტიფიცირებადი TCP socket-ის მეშვეობით. ამავე დროს, მას არ გააჩნია იმის აუცილებლობა, რომ გაავრცელოს მარშრუტიზაციის ცხრილი გარდა მარშრუტისა სასაზღვრო მარშრუტიზატორამდე. [32]

იმის გამო, რომ PfR წარმოადგენს მარშრუტის არჩევის ტექნოლოგიას, ამ პროტოკოლის კონტროლის ქვეშ უნდა არსებობდეს სულ მცირე ორი ან მეტი გარე ინტერფეისი და ერთი შიდა ინტერფეისი მაინც. ასევე, სულ მცოირე ერთი სასაზღვრო მარშრუტიზატორი უნდა იყოს გამართული. იმ შემთხვევაში, თუ მხოლოდ ერთი სასაზღვრო მარშრუტიზატორი არის გამართული, მაშინ ორივე გარე ინტერფეისი უნდა იყოს ამ მარშრუტიზატორთან შეერთებული. თუ ერთზე მეტი სასაზღვრო

მარშრუტიზატორი არის ხელმისაწვდომი, მაშინ ორი ან მეტი გარე ინტერფეისი შეიძლება იყვნენ განაწილებული ამ მარშრუტიზატორებს შორის. მაშასადამე, გარე ინტერფეისები, ე.ი. გამავალი არხები, წარმოადგენენ სასაზღვრო მარშრუტიზატორების ნაწილს. ასეთ არხებს შეიძლება წარმოადგენდეს ვირტუალური გვირაბი ან ფიზიკური ინტერფეისი. იმისათვის, რომ PfR პროტოკოლმა მოახერხოს პრეფიქსების ან აპლიკაციების ნაკადების კონტროლი, ამ ტრაფიკმა უნდა გაიაროს სასაზღვრო მარშრუტიზატორის შიდა ინტერფეისი და გადამისამართდეს ერთერთ გარე ინტერფეისისაკენ.

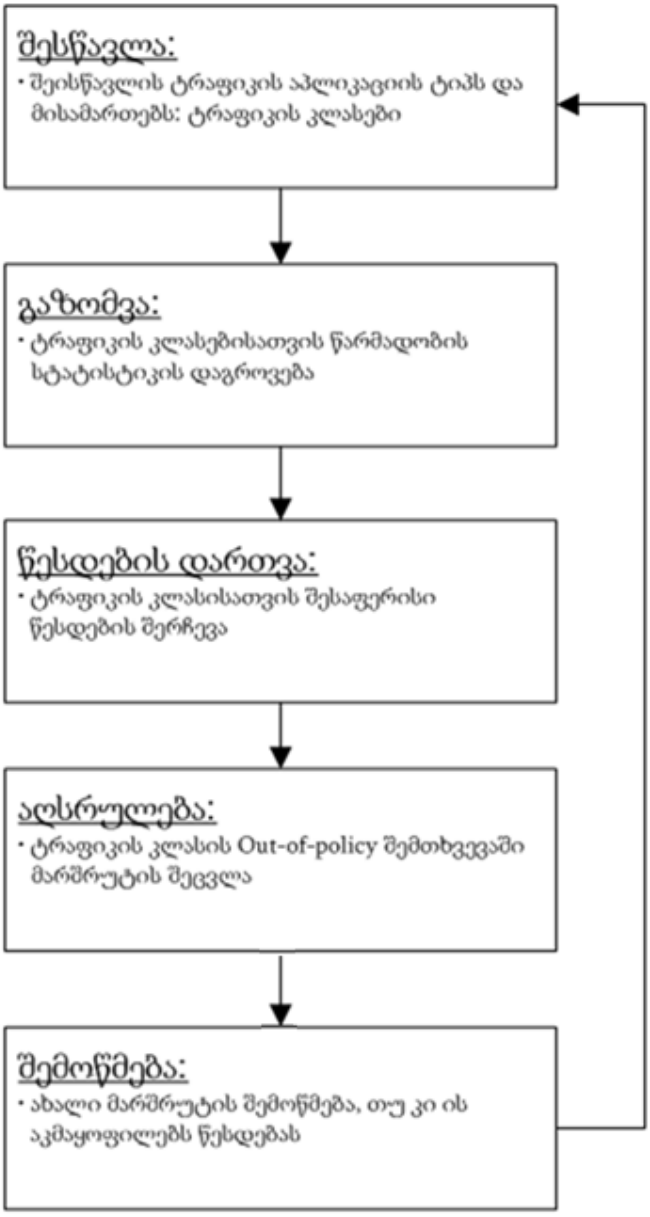
2.3.1 წესების მექანიზმი

ნახაზზე 2.3 მოცემულია PfR პროტოკოლის ალგორითმის ზოგადი მოდელი და ის მოიცავს 5 სტადიას.

- აპლიკაციების შესწავლა: მასტერ კონტროლერი უბრძანებს სასაზღვრო მარშრუტიზატორს აწარმოოს მნიშვნელოვანი აპლიკაციების ნაკადების შესწავლა, რომელსაც ეწოდება ტრაფიკის კლასები.
- აპლიკაციების წარმადობის გაზომვა: შესწავლილი აპლიკაციების ნაკადების სტატისტიკის დაგროვება. არსებობს მონიტორინგის სამი მეთოდი: აქტიური, პასიური და კომბინირებული. პასიური მონიტორინგის შემთხვევაში გამოიყენება NetFlow პროტოკოლი. იმ შემთხვევაში, თუ აპლიკაციის პაკეტები წარმოადგენენ UDP ნაკადს შესაძლებელია გამტარუნარიანობის გაზომვა. ამის გარდა, თუ კი აპლიკაციების პაკეტები წარმოადგენენ TCP ნაკადს, მაშინ პროტოკოლს შეუძლია დაყოვნების, გამტარუნარიანობის, ხელმისაწვდომობის ან პაკეტების დანაკარგის გაზომვა. აქტიური მონიტორინგის ალგორითმში შედის IP SLA უტილიტი, რომელსაც შეუძლია როგორც UDP ასევე TCP ნაკადების დაყოვნების,

დაყოვნების ვარიაციის, ხელმისაწვდომობის და გამტარუნარიანობის გაზომვა და მონიტორინგი.

- წესის დართვა: იყენებს დაგროვილი სტატისტიკის ინფორმაციას იმისათვის, რომ დაადგინოს ტრაფიკის კლასის მდგომარეობა. კერძოთ, იმ შემთხვევაში თუ ტრაფიკის კლასი წესს არ ექვემდებარება, PfR პროტოკოლი განახორციელებს ისეთი ალტერნატიული მარშრუტის შერჩევას, რომელიც დააკმაყოფილებს მოცემული ტრაფიკის კლასის წესს.
- მარშრუტის ცვლილების განხორციელება: პრეფიქსების კონტროლი შესაძლებელია სტატიკური მარშრუტების დამატებით ან BGP–ს ატრიბუტების მეშვეობით. აპლიკაციების მართვა ხორციელდება დინამიკური მარშრუტის რუკების ან წდმ–ს დახმარებით, რომლებიც შეიცავენ ACL–ს მომხმარებლის ტრაფიკის კლასების ჩანაწერებით. ასევე გამოიყენება NBAR ან დინამიკური მარშრუტიზაციის პროტოკოლები.
- შემოწმების ფაზა: ამ ეტაპზე ხორციელდება ახალი მარშრუტების შემოწმება, თუ კი ისინი აკმაყოფილებენ წესს.



ნახ. 2.3 Performance Routing-ის ალგორითმის ხუთი ფაზა

2.3.2 მიღწევადობის შემოწმება

იმისათვის, რომ PfR პროკოტოლმა განიხილოს გამავალი ინტერფეისი ალტერნატიული მარშრუტის კანდიდატად საჭიროა, რომ დანიშნულების მისამართამდე შემოწმდეს მიღწევადობა. როდესაც PfR არის გამართული პასიურ რეჟიმში, ე.ი. NetFlow-ს გამოყენებით, აუცილებელია, რომ გამავალ ინტერფესზე არსებობდეს TCP ნაკადები. ამ პირობის

შეუსრულებლობის შემთხვევაში, შეუძლებელი იქნება ამ ინტერფეისის მეშვეობით მიღწევადობის შემოწმება.

იმისათვის, რომ NetFlow პროტოკოლმა მოახერხოს TCP ნაკადების მეშვეობით მიღწევადობის შემოწმება კონკრეტული გამავალი ინტერფეისის გამოყენებით, აუცილებელია, რომ არსებობდეს მარშრუტი ამ ინტერფეისის მიმართულებით. მაშასადამე, თუ მოცემულ გამავალ ინტერფეისზე არ მოიძებნება TCP ტრაფიკი, ასეთ შემთხვევაში შეუძლებელია NetFlow-ს მეშვეობით პასიური შემოწმების განხორციელება. არსებობს შემთხვევები, როდესაც TCP ნაკადები შედარებით ხანგრძლივი დროის განმავლობაში აქტიურები არიან, ეს დროის მონაკვეთი უფრო ხანგრძლივია ვიდრე PfR მონიტორინგის პერიოდი. ამის შედეგად, დიდი ხნის განმავლობაში არ გადაეცემა TCP-SYN და TCP-SYN/ACK ალმების მქონე პაკეტები. ამის შედეგად, დაყოვნების და მიღწევადობის გაზომვა შეუძლებელი ხდება.

რაც მეტია TCP ნაკადები, მით უფრო ვრცელია მასტერ კონტროლერთან ანალიზისათვის წარდგენილი ინფორმაცია. ეს თავისთავად განაპირობებს PfR პროტოკოლის პასიური მეთოდის მუშაობის ოპტიმიზირებას. რაც უფრო დიდია TCP ნაკადების რიცხვი, მით უფრო ვრცელია სტატისტიკის მონაცემთა ბაზა. ამის შედეგად, დანიშნულების პრეფიქსის დაყოვნების, პაკეტების დანაკარგის და მიღწევადობის ინფორმაცია უფრო დეტალურია და მონაცემთა ანალიზი ნაკლები ცდომილებებით ხორციელდება. [33]

2.4 შესწავლის ფაზა

იქამდე, სანამ PfR პროტოკოლი მოახდენს ტრაფიკის ოპტიმიზირებას, აუცილებელია სასაზღვრო მარშრუტიზატორებში გამავალი ტრაფიკის იდენტიფიცირება და კლასებად დაყოფა. ტრაფიკის მარშრუტიზაციის ოპტიმიზირებისათვის, ჯამური ტრაფიკი უნდა იყოს იდენტიფიცირებული და დაყოფილი კლასებად, რომლებსაც ეწოდებათ ტრაფიკის კლასები. ტრაფიკის კლასების ნუსხას ეწოდება მონიტორებადი ტრაფიკის კლასების

ნუსხა. ტრაფიკის კლასები შეიძლება იყვნენ დაყრდნობილი მესამე შრის ინფორმაციაზე, ე.ი. პრეფიქსებზე ან მეოთხე შრის პორტებზე.

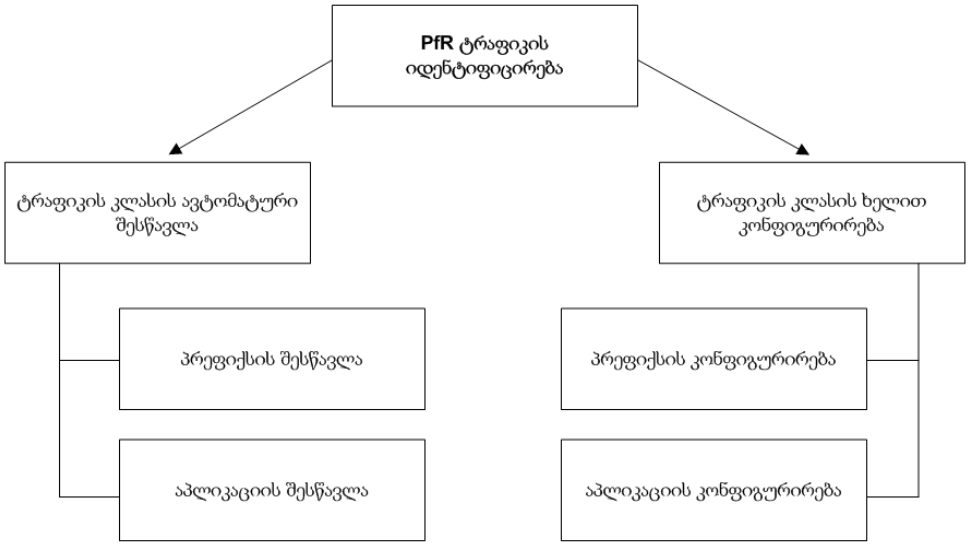
ტიპი		მაგალითი
დაწიშწულების პრეფიქსი (სავალდებულო)		10.0.0.0/8
		20.1.1.0/24
აპლიკაციები (არასავალდებულო)	მისაწვდომობის მართვის წუსხა	10.1.1.0/24 dscp ef
		10.1.1.0/24 dst-port 50
	ცნობილი	10.1.1.0/24 telnet
		20.1.0.0/16 ssh
	NBAR	10.1.1.0/24 nbar RTP
		20.1.1.0/24 nbar citrix

ცხრილი 2.1 ტრაფიკის კლასების შესაძლო პირობები

ასევე კლასების იდენტიფიცირება შეიძლება ხორციელდებოდეს პაკეტების DSCP ველის მნიშვნელობაზე ან აპლიკაციებზე დაყრდნობით, რომლის იდენტიფიცირება შესაძლებელია Cisco NBAR-ის მეშვეობით.

მონიტორებადი ტრაფიკის კლასების ნუსხის ჩანაწერების შექმნა შესაძლებელია როგორც ავტომატურ რეჟიმში, ე.ი. გადაცემული და მიღებული ტრაფიკის შესწავლით NetFlow სტატისტიკის მეშვეობით, ასევე ტრაფიკის კლასების ხელით კონფიგურირებით. შესწავლილი და კონფიგურირებული ტრაფიკის კლასები შეიძლება ერთდროულად იყვნენ მოქცეული მონიტორებადი ტრაფიკის კლასების(მტკ) ნუსხაში. [34]

ტრაფიკის იდენტიფიცირების პროცესი არის სქემატურად წარმოდგენილი ნახაზზე 2.4.



ნახ. 2.4 შესწავლის ფაზა მოიცავს, როგორც შესწავლის ასევე კონფიგურირების მექანიზმებს

2.4.1 ხელით კონფიგურირება

ამ რეჟიმში ხორციელდება პრეფიქსების და აპლიკაციების უშუალოდ ხელით შეტანა PfR პროტოკოლის მონაცემთა ბაზაში. ამ შემთხვევაში შესწავლა არაა აუცილებელი. როგორც კი ქსელის ადმინისტრატორი დაამატებს პრეფიქსის ჩანაწერს, ხორციელდება ამ პრეფიქსის ჩამატება PfR მონაცემთა ბაზაში.

ეს მეთოდი გამოიყენება იმ შემთხვევაში, როდესაც წინასწარ ცნობილია ის პრეფიქსები და აპლიკაციები, რომლების ოპტიმიზირება არის საჭირო.

2.4.2 ავტომატური კონფიგურირება

ამ რეჟიმში, Performance Routing-მა უნდა აღმოაჩინოს ტრაფიკის კლასები სასაზღვრო მარშრუტიზატორებში გამავალი ტრაფიკის საფუძველზე. ეს მეთოდი წარმოადგენს PfR კონფიგურაციის „შესწავლის“ ნაწილს. ავტომატური შესწავლის მარტივ კონფიგურაციას წარმოადგენს შემდეგი შემთხვევა, სადაც PfR პროტოკოლი ავტომატურად თვალყურს

ადევნებს NetFlow სტატისტიკის Top Talkers ნაწილს, რომელსაც ის ღებულობს სასაზღვრო მარშრუტიზატორებისაგან.

2.4.3 შესწავლის ნუსხა learn-list

ტრაფიკის კლასების შესწავლის გასამარტივებლად და ასევე მექანიზმის მოქნილობის გაუმჯობესების მიზნით, PfR პროტოკოლს გააჩნია შესწავლის ნუსხის მეთოდი. ყველა შესწავლის ნუსხაში შეიძლება იყოს მითითებული ტრაფიკის კლასების შესწავლის სხვადასხვა პარამეტრები.
ასეთ პარამეტრს შეიძლება წარმოადგენდეს პრეფიქსი, აპლიკაცია, ფილტრი, შეკრებილი ქსელი ან სხვა. PfR ავტომატურად ახორციელებს ტრაფიკის კლასის შესწავლას და ეს პროცესი დაყრდნობილია შესწავლის ნუსხის კრიტერიუმებზე.

ყველა შესწავლის ნუსხის ჩანაწერს გააჩნია თავისი მიმდევრობითი ნომერი. ეს მიმდევრობითი ნომერი აღნიშნავს შესწავლის ნუსხის შესრულების თანამიმდევრობას. შესწავლის ნუსხა იმის საშუალებას იძლევა, რომ ყველა ტრაფიკის კლასს მიენიჭოს სხვადასხვა PfR წესი.

2.5 გაზომვის ფაზა

გაზომვის ფაზა წარმოადგენს PfR პროტოკოლის ალგორითმის მეორე ეტაპს. წინა ეტაპის შესრულების შედეგად, ტრაფიკის კლასების ნუსხა შეიცავს სხვადასხვა ტრაფიკის კლასების ჩანაწერებს. PfR პროტოკოლმა უნდა განახორციელოს ამ ტრაფიკის კლასების ჩანაწერების წარმადობის მახასიათებლის გაზომვა.

გაზომვის პროცესი წარმოადგენს გარკვეული დროის ინტერვალით განხორციელებული პერიოდული გაზომვების მიმდევრობას. ამ შემთხვევაში პროტოკოლი ამოწმებს მომხმარებლის მიერ მითითებული გაზომვების ბარიერს. PfR პროტოკოლი ახორციელებს გაზომვებს აქტიური და პასიური მონიტორინგის მეთოდებით. ამის გარდა ხორციელდება ინტერფეისის დატვირთვის შემოწმება.

სასაზღვრო მარშრუტიზატორები აგროვებენ პასიური და აქტიური მონიტორინგის სტატისტიკას და უგზავნიან ამ ინფორმაციას მასტერ კონტროლერს. გაზომვის ფაზა ითვლება დასრულებულად, როდესაც გაზომვის ტრაფიკის ნუსხის ყველა ტრაფიკის კლასის ჩანაწერს გააჩნია შესაბამისი წარმადობის გაზომვის შედეგები.

საჭიროა აღინიშნოს, რომ TCP, UDP და დაყოვნების ვარიაციის პარამეტრების გასაზომად საჭიროა დანიშნულების კვანძზე IP SLA responder-ის გამართვა.

<table>
<tr><td colspan="3">პასიური</td><td colspan="3">აქტიური</td></tr>
<tr><td>მიღწევადობა</td><td colspan="2">დაყოვნება | დანაკარგი</td><td>მიღწევადობა</td><td colspan="2">დაყოვნება | დანაკარგი</td></tr>
<tr><td colspan="2">გამავალი გამტარუნარიანობა</td><td>შემომავალი გამტარუნარიანობა</td><td colspan="2">დაყოვნების ვარიაცია</td><td>MOS</td></tr>
<tr><td colspan="3">• PfR NetFlow მონიტორინგი
• ნაკადების სიმეტრიულობა არ არის აუცილებელი</td><td colspan="3">• PfR იყენებს IP SLA უტილიტს
• სასაზღვრო მარშრუტიზატორი აგზავნის სინჯებს
• ICMP სინჯების შესწავლა ან კონფიგურირება
• TCP, UDP და დაყოვნების ვარიაციას ესაჭიროება IP SLA responder-ი</td></tr>
<tr><td colspan="6">ჰიბრიდული მეთოდი</td></tr>
<tr><td colspan="2">შეთავსებული</td><td>სწრაფი</td><td colspan="3">აქტიური გამტარუნარიანობა</td></tr>
<tr><td colspan="2">• წარმადობის პასიური გაზომვა
• საჭიროების შემთხვევაში აქტიური სინჯები
• წარმოადგენს საწყის მნიშვნელობას</td><td>• გამტარუნარიანობის პასიური შემოწმება
• აქტიური სინჯები ყველა მარშრუტით შეუწყვეტლივ</td><td colspan="3">• გამტარუნარიანობის პასიური შემოწმება
• მიმდინარე გამავალი ინტერფეისის აქტიური სინჯები</td></tr>
</table>

ცხრილი 2.2 გაზომვის ფაზის წარმადობის პარამეტრები

2.5.1 პასიური მონიტორინგის მეთოდი

პასიური მონიტორინგი წარმოადგენს NetFlow–ს მიერ შეგროვებული მომხმარებლის ტრაფიკის ნაკადების ინფორმაციას. როგორც წესი, პასიური მონიტორინგის მეთოდი გამოიყენება ინტერნეტთან მოსაზღვრე მარშრუტიზატორებზე. ამას განაპირობებს ის ფაქტი, რომ სხვადასხვა უსაფრთხოების წესების შეზღუდვის გამო, ინტერნეტთან მოსაზღვრე მარშრუტიზატორებზე აქტიური სინჯების მეთოდი ხშირად არის არაეფექტური. PfR პროტოკოლის გამართვის შემთხვევაში, სასაზღვრო მარშრუტიზატორების მართვად ინტერფეისებზე ავტომატურად

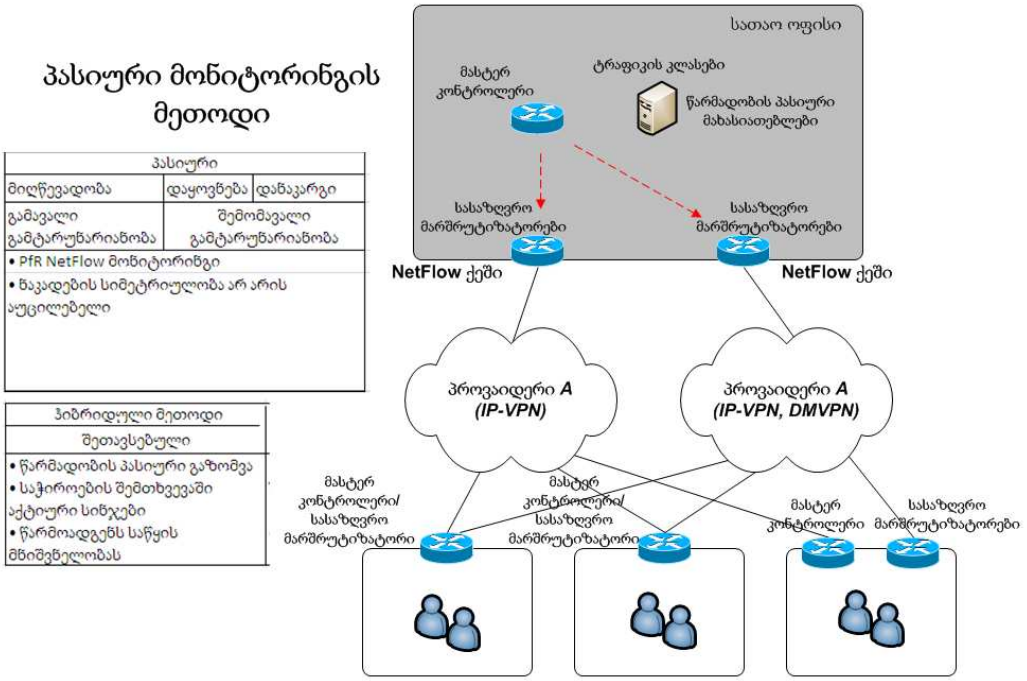
აქტიურდება NetFlow. ამის შედეგად, სასაზღვრო მარშრუტიზატორები აგროვებენ ნაკადების ინფორმაციას და პერიოდულად მიაწოდებენ შეჯამებულ ინფორმაციას მასტერ კონტროლერს. მაშასადამე, ხორციელდება ქსელში მომუშავე აპლიკაციების და აქტიური ქსელის პრეფიქსების ავტომატურად შესწავლა.

სასაზღვრო მარშრუტიზატორები უგზავნიან მასტერ კონტროლერს მოხსენებას იმის თაობაზე, თუ რა ნაკადები იყო დადგენილი NetFlow-ს მიერ. NetFlow ინფორმაციაზე დაყრდნობით, მარშრუტიზატორი ადგენს ყველა დანიშნულების მისამართისათვის შემდეგ პარამეტრებს:

- აღმოჩენილი ნაკადების გადაცემის საშუალო დაყოვნებას
- პაკეტების დანაკარგის სტატისტიკას
- დანიშნულების მისამართის მიღწევადობას
- გამტარუნარიანობას, განსაზღვრულს ბიტები/წამში

TCP ნაკადების გაზომვა ხასიათდება შემდეგი პარამეტრებით:

- დაყოვნება – TCP სამ მხრივი ხელშეკრულების SYN და SYN/ACK პაკეტებს შორის დროის მონაკვეთი
- დანაკარგი – ხორციელდება TCP მიმდევრობითი ნომრების მონიტორინგი, პაკეტების დანაკარგზე მიუთითებს დაბალი მიმდევრობითი ნომრის მქონე პაკეტები მიღება, მაღალი მიმდევრობითი ნომრის მქონე პაკეტების მაგივრად.
- მიღწევადობა – გამეორებითი TCP SYN პაკეტების აღმოჩენისას შესაბამისი SYN/ACK პაკეტების გარეშე.
- გამტარუნარიანობა – ამ პარამეტრის გამოთვლა ხორციელდება NetFlow-ს მეშვეობით და განისაზღვრება ბიტები/წამში. TCP-ისაგან განსხვავებული ნაკადების დახასიათება შესაძლებელია მხოლოდ გამტარუნარიანობის პარამეტრის მეშვეობით.

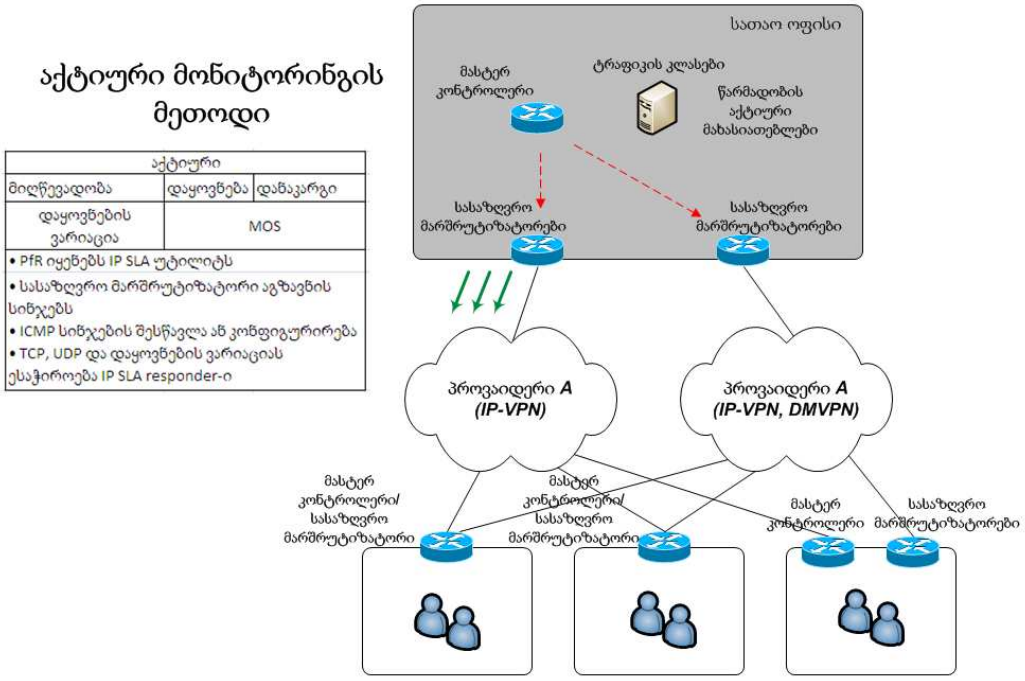


ნახ. 2.5 პასიური მონიტორინგის მეთოდი NetFlow-ს გამოყენებით

2.5.2 აქტიური მონიტორინგის მეთოდი

აქტირული მონიტორინგი წარმოადგენს Cisco IP SLA უტილიტის მიერ სინჯების წარმოშობას. ამ სინჯების გამოყენებით წარმოიშობა საცდელი ტრაფიკი, რომლის მიზანს წარმოადგენს კომუნიკაციის არხების მახასიათებლების დადგენა. აქტიური სინჯების წარმოშობა შეიძლება ხორციელდებოდეს ავტომატურად, როდესაც პასიური მონიტორინგის მექანიზმმა დაადგინა დანიშნულების მისამართები. ასევე, აქტიური სინჯების წარმოშობა შესაძლებელია ქსელის ადმინისტრატორის მიერ PfR პროკოტოკოლის კონფიგურირების შედეგად.

მოცემული მეთოდით IP SLA სინჯების წარმოშობა ხორციელდება სასაზღვრო მარშრუტიზატორების მიერ გამავალი ინტერფეისების მიმართულებით. მათი გაგზავნის სიხშირე დამოკიდებულია შესაბამისი კონფიგურაციის პარამეტრის მნიშვნელობაზე. ალტერნატიული მიმართულებებით სინჯების გაგზავნა ხორციელდება მხოლოდ იმ შემთხვევაში, როდესაც არსებული მარშრუტი იმყოფება out-of-policy მდგომარეობაში, რაც ნიშნავს, რომ PfR წესი არ არის დაკმაყოფილებული.



ნახ. 2.6 აქტიური მონიტორინგის მეთოდი სინჯების გამოყენებით

ამის გარდა, პროტოკოლი ავტომატურად აწარმოებს ICMP Echo მოთხოვნების წარმოშობას, რაც თავის მხრივ ქმნის დამატებით ტრაფიკს. ინტერნეტში გამოყენების შემთხვევაში, აქტიური სინჯების არსებობა შეიძლება წარმოადგენდეს არასასურველ ფაქტს იმის გამო, რომ შესაძლებელია ICMP პაკეტების ბლოკირება მარშრუტის გასწვრივ, ან შეიძლება იყოს ადმინისტრაციულად აკრძალული დანიშნულების კვანძის ქსელში. სტანდარტულ შემთხვევაში, აქტიურ სინჯს წარმოადგენს ICMP echo ტიპის პაკეტი. იმ შემთხვევაში, თუ საჭიროა VoIP ტიპის ტრაფიკის მახასიათებლების დადგენა, მიზანშეწონილად შეიძლება ჩაითვალოს აქტიური სინჯების ხელით კონფიგურირება. ქვემოთ არის მოყვანილი იმის მაგალითი, თუ როგორ ხორციელდება ტრაფიკის კლასის განსაზღვრა PfR პროტოკოლის კონფიგურაციაში. ეს ტრაფიკის კლასი აღწერს VoIP ტიპის ნაკადებს და წარმოშობს სინჯებს 2 წამის ინტერვალით.

set active-probe jitter 30.1.0.11 target-port 33033 codec g729a

set probe frequency 2

მიზანშეწონილად შეიძლება ჩაითვალოს გარკვეული ტრაფიკის კლასებისათვის აქტიური სინჯების გამოყენება მაშინ, როდესაც დანარჩენი

TCP ნაკადებისათვის გამოიყენოს გლობალური კონფიგურაციის საფუძველზე პასიური მონიტორინგის მეთოდი.

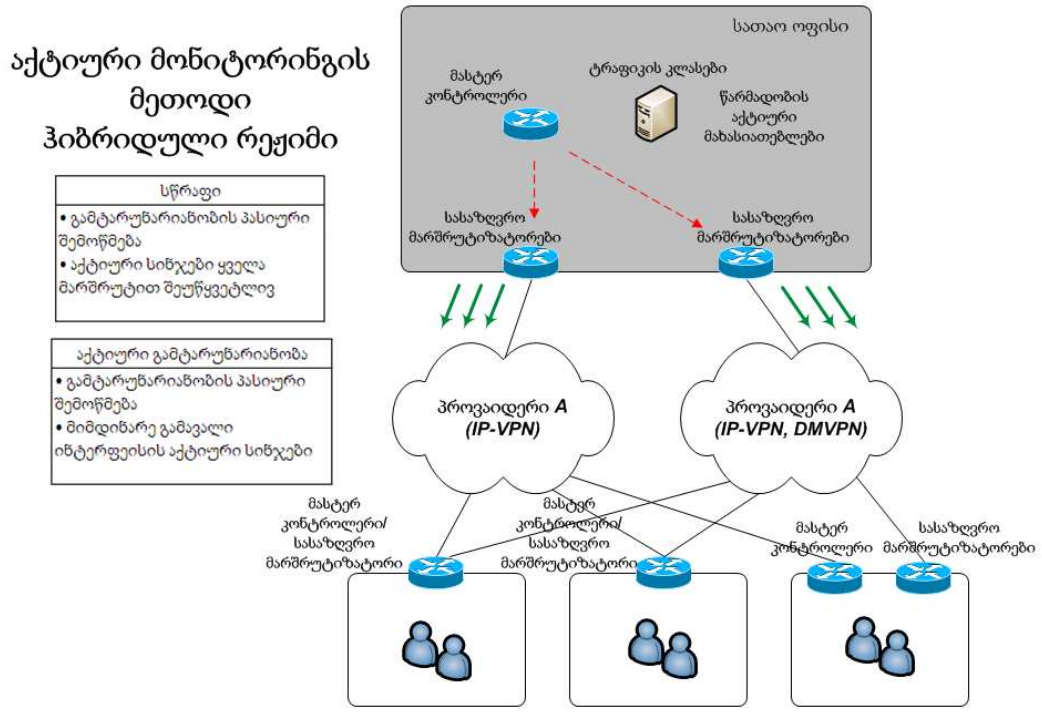
2.5.3 ჰიბრიდული მეთოდი

შეთავსებული(ჰიბრიდული) რეჟიმი წარმოადგენს PfR პროტოკოლის საწყის მეთოდს და შეიცავს როგორც აქტიური ასევე პასიური მონიტორინგის საშუალებებს. პასიური მეთოდით შესწავლილი ყველა დანიშნულების ქსელის პრეფიქსისათვის შესაძლებელია ხუთი IP მისამართისათვის აქტიური სინჯებით გაზომვა. საწყისი კონფიგურაციის თანახმად, ყოველი შესწავლილი IP მისამართისათვის ავტომატურად წარმოიშობა IP SLA ICMP echo სინჯი.

ამ რეჟიმში, სასაზღვრო მარშრუტიზატორები ახორციელებენ IP SLA სინჯების წარმოშობას არსებული გამავალი ინტერფეისების გამოყენებით და ამ სინჯების გაგზავნის სიხშირეს განაპირობებს კონფიგურირებული ან საწყისი მნიშვნელობა. სწრაფი მონიტორინგისაგან განსხვავებით, აქტიური სინჯების მეთოდი არ აწარმოებს ყველა მიმართულების შეუწყვეტლივ შემოწმებას. ხორციელდება მხოლოდ მიმდინარე მიმართულებების შემოწმება, რომლებიც იმყოფებიან in-policy მდგომარეობაში. საჭიროა აღინიშნოს, რომ სინჯების წარმოშობა ხორციელდება მხოლოდ პრეფიქსის დროის მთვლელის ამოწურვის შემდეგ. მიმდინარე მარშრუტის out-of-policy მდგომარეობაში ყოფნისას, სინჯების წარმოშობა ხორციელდება მხოლოდ ალტერნატიული გამავალი ინტერფეისების მეშვეობით.

აქტიური და პასიური მონიტორინგის ერთობლივი გამოყენებით, რაც მოიცავს ორ სხვადასხვა ხელსაწყოს, შესაძლებელი ხდება ქსელის პრეფიქსის დამატებითი მონაცემთა მიღება. პასიური მონიტორინგის ხელსაწყოს წარმოადგენს NetFlow და აქტიური მონიტორინგის განსახორციელებლად გამოიყენება IP SLA. საჭიროა აღინიშნოს, რომ აქტიური სინჯების გამოყენებას გააჩნია უარყოფითი მხარეებიც. ამიტომ,

ჰიბრიდული მეთოდის გამოყენება წარმოადგენს მეტად ეფექტურ მექანიზმს კერძო შიდა ქსელებისათვის.



ნახ. 2.7 აქტიური მონიტორინგის მეთოდის ჰიბრიდული რეჟიმი

2.5.4 სწრაფი მონიტორინგის მეთოდი

სწრაფი მონიტორინგის მეთოდი წარმოადგენს PfR პროტოკოლის შედარებით ახალ ფუნქციონალურ დამატებას. ამ რეჟიმში, კონფიგურირებული სიხშირის მიხედვით ხორციელდება ყველა გამავალი ინტერფეისების მიმართულებით აქტიური სინჯების წარმოშობა. მოცემული მეთოდი განსხვავდება აქტიური და ჰიბრიდული მეთოდებისაგან იმითი, რომ ალტერნატიული მიმართულებებით სინჯების წარმოშობა ხოცრიელდება მხოლოდ იმ შემთხვევაში, რომდესაც მიმდინარე მარშრუტი იმყოფება out-of-policy მდგომარეობაში, ე.ი. ტრაფიკის კლასის წესი აღარ კმაყოფილდება.

სწრაფი მონიტორიგის რეჟიმში, ალტერნატიული მიმართულებების მახასიათებლები ყოველთვის წნობილია და ეს პირობა იძლევა იმის

საშუალებას, რომ საჭიროების შემთხვევაში განხორციელდეს მარშრუტის სწრაფი შეცვლა. იმ შემთხვევაში, როდესაც მიმართულება მიუწვდომელია და ეს განსაზღვრავს დანიშნულების მისამართის out-of-policy მდგომარეობაში ყოფნას, PfR პროტოკოლი ახორციელებს ამ მიმართულებისათვის ახალი მარშრუტის შერჩევას. ამ მარშრუტის არჩევა ხორციელდება in-policy მდგომარეობაში მყოფი მიმდინარე ალტერნატიული მარშრუტებისაგან. სასურველი მარშრუტის არჩევისას ხორციელდება მარშრუტის სწრაფი გადართვა.

უშედეგო სინჯების რიცხვის შეფარდება მილიონ სინჯთან განაპირობებს მიღწევადობის ბარიერს. თუ კი მიღწევადობის მნიშვნელობა წარმოადგენს 1, ამ შემთხვევაში მიმდინარე მიმართულებით ერთეული სინჯის დაკარგვა გამოიწვევს ალტერნატიული მარშრუტის შერჩევის პროცესს. ამავე დროს, თუ კი ალტერნატიული მარშრუტიც შეიცავს ერთ დაკარგულ სინჯს, ამ შემთხვევაში მარშრუტის შეცვლა არ განხორციელდება. ამის მიზეზს წარმოადგენს ის პირობა, რომ როგორც მიმდინარე ასევე ალტერნატიული მარშრუტი იმყოფება out-of-policy მდგომარეობაში.

მაშასადამე, სწრაფი გადართვის ფუნქცია ახერხებს მარშრუტის შეცვლას სინჯების კონფიგურირებული ინტერვალის სიდიდის პერიოდში. სწრაფი რეჟიმის გამოყენების შემთხვევაში, სინჯების გაგზავნის სიხშირის მნიშვნელობა შეიძლება წარმოადგენდეს 2 წამამდე მცირე ინტერვალს. ეს იძლევა იმის საშუალებას, რომ მარშრუტების გადართვის სიხშირე წარმოადგენდეს სინჯების სიხშირის ინტერვალზე ოდნავ მეტ მნიშვნელობას. ეს მეთოდი საშუალებას იძლევა განხორციელდეს out-of-policy ტრაფიკის მხოლოდ 3 წამში ახალ მარშრუტზე გადამისამართება, რაც წარმოადგენს საკმაოთ სწრაფი რეაგირების ფაქტორს.

ამ მეთოდის ძირითად უარყოფით მხარეს წარმოადგენს ქსელში წარმოშობილი დამატებითი ტრაფიკი, რომელიც გამოწვეულია სინჯების

გადაგზავნით. ასევე ეს პროცესი შეიძლება მოითხოვდეს სასაზღვრო მარშრუტიზატორის პროცესორის დამატებით რესურსებს.
სინჯების წარმოშობა წარმოადგენს უწყვეტ პროცესს იქამდე, სანამ არ მოხდება კონფიგურაციიდან პრეფიქსის ამოღება ან პროტოკოლის საწყის მდგომარეობაში გადართვა.

აქტიური სინჯების შედეგები გამოიყენება მარშრუტიზაციის და out-of-policy–ის სამართავად. პასიური მონიტორინგის ინფორმაცია გამოიყენება მხოლოდ სტატისტიკისათვის. გაგზავნილი და მიღებული ინფორმაციის მნიშვნელობები (გამოხატული კბწმ–ში) გამოიყენება დატვირთვის განაწილების მექანიზმისათვის.

2.6 წესის დართვის ფაზა

წესის დართვის ფაზა ახორციელებს გაზომილი წარმადობის მახასიათებლების შედარებას არსებული ან კონფიგურირებული ბარიერების მიმართ. ამის შედეგად, პროტოკოლი ცდილობს დაადგინოს, თუ ტრაფიკი აკმაყოფილებს საჭირო მომსახურების ხარისხის დონეს ან თუ აუცილებელია რაიმე მოქმედების ჩატარება. იმ შემთხვევაში, როდესაც წარმადობის მახასიათებლები არ აკმაყოფილებენ მიღებულ ბარიერს, PfR პროტოკოლი იღებს გადაწყვეტილებას მოახდინოს ტრაფიკის კლასის გადამისამართება ან მდგომარეობის შეცვლა.

ტრაფიკის კლასი		არხი	
წარმადობა	ხელმისაწვდომობა	წარმადობა	ადმინისტრაციული
• დაყოვნება • პაკეტების დანაკარგი • მიღწევადობა • MOS • დაყოვნების ვარიაცია	• sinkhole • blackhole	• დატვირთვის განაწილება • მაქსიმალური გამოყენება	• არხების დაჯგუფება • ღირებულება

ცხრილი 2.3 წესის დართვა შესაძლებელია გლობალურად, ტრაფიკის კლასის დონეზე ან არხის მიხედვით

2.6.1 PfR პროტოკოლის დატვირთვის განაწილების ფუნქცია

არხის დატვირთვა

ამ წესის გამოყენება იწვევს გარკვეული არხის შესაძლო ტრაფიკის რაოდენობის ზედა ზღვარის განსაზღვრას. PfR პროტოკოლის წესში შესაძლებელია განისაზღვროს როგროც გამავალი, ასევე შემომავალი ტრაფიკის დატვირთვის ზღვარი. მაგალითად, თუ კი გამტარუნარიანობის ზღვარი შეადგენს ინტერფეისის სრული გამტარუნარიანობის 90%, და მიმდინარე დატვირთვა წარმოადგენს 95%, ასეთი არხი ჩაითვლება out-of-policy მდგომარეობაში. Cisco PfR პროტოკოლი ეცდება ინტერფეისის დაბრუნებას in-policy მდგომარეობაში, რაც მოიცავს არხის დატვირთვის შემცირებას. ამის მისაღწევად, პროტოკოლი ახორციელებს პრეფიქსების გადატვირთული ინტერფეისიდან სხვა ინტერფეისებზე გადანაწილებას.

დიაპაზონი

ამ წესის გამოყენება იწვევს ყველა არსებული არხის დატვირთვის შენარჩუნებას გარკვეული დიაპაზონის ფარგლებში, რის შედეგად ხორციელდება ყველა არხის დატვირთვის თანაბარი განაწილება. დიაპაზონის ფუნქცია საშუალებას აძლევს ქსელის ადმინისტრატორს მიუთითოს არხების დატვირთვის პროცენტული დიაპაზონი, რომელიც არის არხებს შორის შეფარდება. იმ შემთხვევაში, თუ არხებს შორის დატვირთვის სხვაობა ძალიან გაიზრდება, PfR პროტოკოლი ეცდება გაათანასწოროს ეს განსხვავება და ამისათვის გადაამისამართებს ნაკადების ნაწილს ნაკლებად დატვირთული არხის მიმართულებით.

ტრაფიკის კლასის წარმადობა

ამ წესის გამოყენების შედეგად მომხმარებელს ეძლევა საშუალება გარკვეული ტრაფიკის კლასისათვის მიუთითოს რამოდენიმე მარშრუტი. ამგვარად შესაძლებელია, რომ VoIP ტრაფიკისათვის გამოიყენებოდეს რამოდენიმე ინტერფეისი იმ პირობით, რომ მათი წარმადობის მახასიათებლები აკმაყოფილებდნენ ტრაფიკის კლასის წესს. მაგალითად, VoIP ტიპის ტრაფიკისათვის კლასს შეიძლება გააჩნდეს წესი, რომლის

დაყოვნების ზედა ზღვარს წარმოადგენს 250 მილიწამი. იმ შემთხვევაში, თუ ყველა არსებული მარშრუტი (სხვადასხვა ინტერფეისებით) აკმაყოფილებენ ამ პირობას, PfR პროტოკოლს შეუძლია გამოიყენოს ყველა მათგანი.

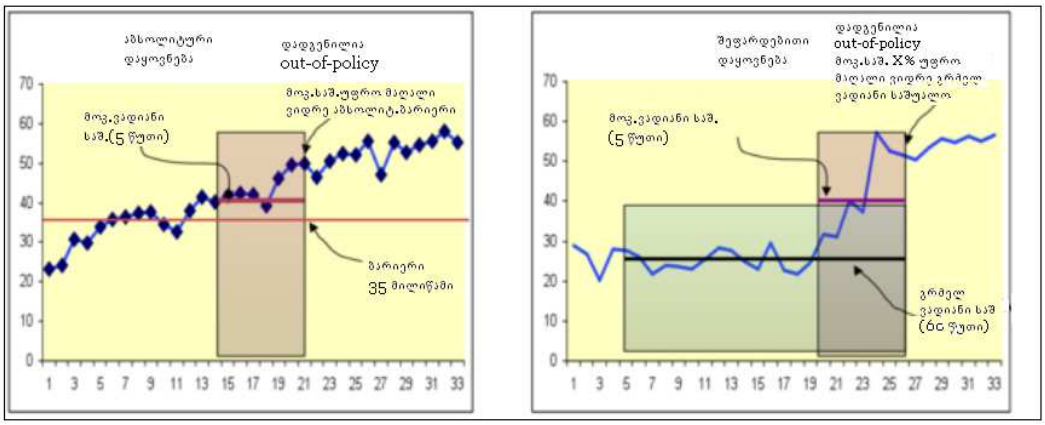
2.6.2 PfR წესის ზღვარი

PfR წესი განისაზღვრება ტრაფიკის კლასების გაზომვის შედეგების და მომხმარებლის მიერ მითითებული ზღვარის მნიშვნელობების მეშვეობით. ეს პროტოკოლი იმის საშუალებას იძლევა, რომ ყველა ტრაფიკის კლასს მიენიჭოს სხვადასხვა წესი. ერთ ტრაფიკის კლასს შეიძლება გააჩნდეს წესი დაყრდნობილი დაყოვნებაზე და პაკეტების დანაკარგის ფაქტორზე, როგორც წესი ასეთი ტრაფიკის კლასები გამოიყენება კრიტიკული აპლიკაციებისათვის. სხვა ტრაფიკის კლასს შეიძლება გააჩნდეს წესი დაყრდნობილი არხებს შორის დატვირთვის განაწილების მექანიზმზე .

- პასიური რეჟიმი: როგროც წესი, Cisco PfR უზრუნველყოფს მოკლე ვადიან მთვლელს, რომელიც აგროვებს ბოლო 5 წუთის სტატისტიკას და გრძელ ვადიან მთვლელს, რომელიც აგროვებს ბოლო 60 წუთის სტატისტიკას.
- აქტიური რეჟიმი: Cisco PfR უზრუნველყოფს მოკლე ვადიან მთვლელს, რომელიც აგროვებს ბოლო 5 სინჯის შედეგების სტატისტიკას და გრძელ ვადიან მთვლელს, რომელიც აგროვებს ბოლო 60 სინჯის შედეგების სტატისტიკას.

<u>აბსოლუტური</u>

ხორციელდება აბსოლუტური ბარიერის შედარება მოკლვე ვადიანი დაყოვნების მნიშვნელობასთან, რომელიც წარმოადგენს პასიური მეთოდისათვის 5 წუთს და აქტიური მეთოდისათვის ბოლო 5 სინჯს. მოკლე ვადიანი მთვლელი აჭარბებს ბარიერს N მილიწამით. ასეთი ტიპის ბარიერები გამოიყენება დაყოვნების, პაკეტების დანაკარგის, MOS, მიღწევადობის და დაყოვნებების ვარიაციისათვის.



ნახ. 2.8 Cisco PfR პროტოკოლის ბარიერები შეიძლება იყვნენ წარმოდგენილი როგორც შეფარდებითი ასევე აბსოლუტური სახით.

<u>შეფარდებითი</u>

ხორციელდება შეფარდებითი ბარიერის შედარება, როგორც მოკლე ასევე გრძელ ვადიან მნიშვნელობასთან. მოკლე ვადიანი მთვლელი აჭარბებს გრძელ ვადიან მთვლელს X პროცენტით. ასეთი ტიპის ბარიერები გამოიყენება დაყოვნების, პაკეტების დანაკარგის და მიღწევადობისათვის. დაყოვნების მაგალითი: თუ მოკლე ვადიანი დაყოვნება უდრის 120 მლწ. და გრძელ ვადიანი დაყოვნება უდრის 100 მლწ. მაშინ პროცენტთა სხვაობა უდრის 20. ხორციელდება ამ მნიშვნელობის შემოწმება ბარიერის მნიშვნელობასთან, რომელიც არის კონფიგურაციაში განსაზღვრული.

ცხრილი 2.4 წარმოადგენს საბაზისო ერთეულებს, რომლებიც გამოიყენება ტრაფიკის კლასებში ყველა პარამეტრის კონფიგურირებისათვის.

განვიხილოთ მაგალითი, სადაც წესი შეიცავს პაკეტების დანაკარგის და მიღწევადობის პარამეტრებს. პაკეტების დანაკარგის წესი განისაზღვრება პაკეტები/მილიონში (პ/მ) ერთეულით.

- პასიურ რეჟიმში: 10000 TCP პაკეტებიდან 100 დაკარგული პაკეტი უდრის 10000 პ/მ.
- აქტიურ რეჟიმში: 100 სინჯიდან 1 დაკარგული სინჯი უდრის 10000 პ/მ.

	ერთეული	პასიური	აქტიური
მიღწევადობა	ნაკადები/მილიონი	100000 ნაკადიდან 100 მიუღწეველი=100005/მ	5 გაგზავნილი სინჯიდან 1 დაკარგული=2000005/მ
დაყოვნება	მილიწამი	TCP SYN და ACK შორის დაყოვნება	სინჯის ორი გზის დაყოვნება
პაკეტების დანაკარგი	პაკეტები/მილიონი	10000 TCP პაკეტიდან 100 დაკარგული=100003/მ	100 სინჯიდან 1 დაკარგული=100003/მ
დაყოვნების ვარიაცია	მილიწამი	–	პაკეტებს შორის დაყოვნების ვარიაცია
MOS	ბარიერის ქვემოთ მყოფი MOS-ის პროცენტი	–	5 სინჯიდან 3მიღებული – MOS 3.85–ზე ნაკლები ~ 60% – MOS 3.85–ზე ნაკლები

აუცილებელია დაყოვნების ვარიაციის სინჯები

ცხრილი 2.4 ტრაფიკის კლასების პარამეტრების საბაზისო ერთეულები

მიღწევადობის წესი განისაზღვრება ნაკადები/მილიონში (ნ/მ) ერთეულით.

- პასიურ რეჟიმში: 10000 TCP ნაკადისაგან 100 მიუღწევადი ნაკადი წარმოადგენს 10000 ნ/მ.
- აქტიურ რეჟიმში: 5 გაგზავნილი სინჯიდან 1 დაკარგული სინჯი წარმოადგენს 200000 ნ/მ.

<u>ფუნქცია Resolvers</u>

PfR პროტოკოლის ყველა ტრაფიკის კლასს შეიძლება გააჩნდეს ერთზე მეტი წესი. იმისათვის, რომ პროტოკოლმა მოახდინოს სასურველი წესის არჩევა, ის იყენებს ფუნქციას “Resolvers”, რომელიც იმის საშუალებას იძლევა, რომ ყველა წესს მიენიჭოს პრიორიტეტის მნიშვნელობა.

საწყის მდგომარეობაში, PfR პროტოკოლი ანიჭებს პრიორიტეტებს შემდეგი მიმდევრობით: უმაღლესი პრიორიტეტი ენიჭება დაყოვნების წესს, შემდეგ არის დატვირთულობის წესი. საჭიროა აღინიშნოს, რომ ყველა შემთხვევაში დაერთვება მიღწევადობის წესი, რომელსაც აქვს პრიორიტეტი 0. ამ პრიორიტეტის შეცვლა შეუძლებელია და ის მუდმივად არის ამ მნიშვნელობის.

ქვემოთ არის წარმოდგენილი არსებული წესების ნუსხა:

```
MC(config-pfr-mc)#resolve ?
  cost        Specify OER cost policy resolver settings
```

```
  delay        Specify OER delay policy resolver settings
  jitter       Specify OER jitter policy resolver settings
  loss         Specify OER loss policy resolver settings
  mos          Specify OER MOS policy resolver settings
  range        Specify OER range policy resolver settings
  utilization  Specify OER utilization policy resolver settings
MC(config-pfr-mc)#
```

ყველა resolve ფუნქციას უნდა მიეთითოს პრიორიტეტი ან პრიორიტეტი იქნება მინიჭებული გლობალური კონფიგურაციიდან. ამ პრიორიტეტის შეცვლა ასევე შესაძლებელია გლობალური pfr-map კონფიგურაციიდან, მაგრამ მიღწევადობის პრიორიტეტი ყოველთვის რჩება უცვლელი, და მისი მნიშვნელობა არის 0.

PfR ახორციელებს იმ მარშრუტის არჩევას, რომელიც აკმაყოფილებს წესს, და ეს პროცესი შეიცავს შემდეგ ეტაპებს:

- თავი ეყრება ყველა ტრაფიკის კლასისათვის არსებული გამავალი ინტერფეისების გამოყენებით გაზომილ ინფორმაციას
- თავი ეყრება ყველა გამავალი ინტერფეისების დატვირთვის ინფორმაციას
- იგნორირდება ის გამავალი ინტერფეისები, რომლების გამოყენებით არ არსებობს გაზომილი ინფორმაცია
- გაზომილი ინფორმაცია დაერთვება პრიორიტეტების ვარიაციით
- ვარიაციის მქონე ინტერფეისები წარმოადგენენ კანდიდატებს

ყველა პრიორიტეტის შემოწმების შემდეგ:

- ერთეული კანდიდატის შემთხვევაში გამოიყენება ცალკეული კანდიდატი
- რამოდენიმე კანდიდატის არსებობის შემთხვევაში (მიმდინარე მარშრუტის ჩათვლით) გამოიყენება მიმდინარე მარშრუტი
- სხვა დანარჩენ შემთხვევებში გამოიყენება შემთხვევით არჩეული კანდიდატი

PfR პროტოკოლის resolvers გლობალური კონფიგურაციის ნიმუში:

```
pfr master
 resolve delay priority 4 variance 20
 resolve loss priority 6 variance 20
 resolve util priority 8 variance 20
```

2.6.3 ტრაფიკის კლასების მდგომარეობები

ქვემოთ არის წარმოდგენილი ტრაფიკის კლასების შესაძლო მდგომარეობები.

საწყისი მნიშვნელობა

ქსელის პრეფიქსი შეიძლება იყოს წარმოდგენილი საწყისი მნიშვნელობით მაშინ, როდესაც ის იყო ხელით კონფიგურირებული ან შესწავლილი, მაგრამ ის არ იყო დადგენილი out-of-policy მდგომარეობაში. ამის გარდა, ამ მდგომარეობაში შეიძლება გადავიდეს პრეფიქსი იმ შემთხვევაში, როდესაც რაიმე მიზეზის გამო პროტოკოლი ვეღარ ახერხებს მოცემული პრეფიქსის კონტროლს. ამის მიზეზი შეიძლება გახდეს ის პირობა, რომ ყველა არსებული გამავალი ინტერფეისი იმყოფება out-of-policy მდგომარეობაში. საწყისი მნიშვნელობა აღნიშნავს იმას, რომ მშობელი IP მარშრუტები აკონტროლებენ დანიშნულების პრეფიქსის გამავალ ინტერფეისს. ეს მდგომარეობა აღნიშნავს, რომ PfR პროტოკოლი არ ფუნქციონირებს.

მდგომარეობა in-policy

როდესაც პრეფიქსი იმყოფება in-policy მდგომარეობაში, ეს მიუთითებს იმაზე, რომ ის აკმაყოფილებს მასთან შესაბამის წესში მყოფ პრეფიქსის ან აპლიკაციის პირობას. პრეფიქსი შეიძლება იმყოფებოდეს in-policy და იმართებოდეს PfR-ის მიერ, ან იმყოფებოდეს in-policy* და არ იმართებოდეს PfR-ის მიერ. როდესაც in-policy მდგომარეობას თან ერთვის ფიფქის (*) სიმბოლო, ეს მიუთითებს იმაზე, რომ ქსელი არის PfR-ისათვის ცნობილი, მაგრამ იმართება მშობელი მარშრუტის მიერ. როდესაც ფიფქის

ნიშანი არ ერთვის მდგომარეობის ატრიბუტს, მაშინ მოცემული პრეფიქსი იმართება PfR პროტოკოლის მიერ. მდგომარეობა in-policy ითვლება სასურველ მდგომარეობად.

როგორც წესი, რამოდენიმე ციკლის შემდეგ, ტრაფიკის კლასი აღმოჩნდება in–policy მდგომარეობაში.

მდგომარეობა out-of-policy

როგორც წესი, ეს მდგომარეობა მიუთითებს იმაზე, რომ პრეფიქსი ან აპლიკაცია არ აკმაყოფილებს შესაბამის წესს. იმ შემთხვევაში, როდესაც ტრაფიკი იმყოფება out-of-policy მდგომარეობაში, PfR პროტოკოლი ცდილობს გადაამისამართოს მოცემული ტრაფიკი ალტერნატიული მარშრუტისაკენ, რათა დაუბრუნოს მას in-policy მდგომარეობა. თუ ამის მიღწევა შეუძლებელია, მაშინ პროტოკოლი ახორციელებს პრეფიქსის კონტროლის გაუქმებას, რის შედეგად პრეფიქსის კონტროლს თავის თავზე იღებს სტანდარტული მარშრუტიზაციის ცხრილი და ტრაფიკი მიემართება საწყისი გამავალი ინტერფეისისაკენ. ამის შემდეგ, PfR პროტოკოლი შემდეგ ციკლებში კვლავ ეცდება ამ პრეფიქსზე კონტროლის დაბრუნებას ზუსტად ისე, როგორც დანარჩენი სხვა პრეფიქსებისათვის. მდგომარეობა out-of-policy ითვლება არასასურველად.

ამ მდგომარეობაში იყო დადგენილი, რომ პრეფიქსი ან აპლიკაცია არ აკმაყოფილებს შესაბამისი წესის პირობებს. წესის პარამეტრები, როგორებიც არიან დაყოვნება, პაკეტების დანაკარგი და სხვა, არიან განსაზღვრული oer-map-ის ტრაფიკის კლასებში.

მდგომარეობა hold-down

მდგომარეობა hold-down აქტიურია მაშინ, როდესაც PfR ახორციელებს ტრაფიკის კლასის საწყის კონტროლს. ეს მდგომარეობა აუცილებელი არის იმისათვის, რომ გარკვეულ სიტუაციებში არ მოხდეს PfR-ის მიერ კონტროლირებადი მარშრუტების ფლუქტუაცია. კერძოთ, მარშრუტების მარშრუტიზაციის ცხრილიდან რეგულარული ამოშლა და კვლავ ჩამატება, რაც გამოიწვევს ქსელში სტაბილურობის დარღვევას და

მომხმარებლის აპლიკაციების მუშაობის შეფერხებას. როდესაც პრეფიქსი შეიცვლება, ის გადადის hold-down მდგომარეობაში და იწყება გარკვეული დროის მონაკვეთის ათვლა. სანამ ეს დროის მონაკვეთი არ ამოიწურება, პრეფიქსის ძალაში შესვლა შეუძლებელია. ქსელის პრეფიქსს შეუძლია დატოვოს hold-down მდგომარეობა იქამდე, სანამ ამოიწურება მთვლელი მხოლოდ იმ შემთხვევაში, როდესაც მიმდინარე გამავალი ინტერფეისი იმყოფება მიუღწევადობის გამო out-of-policy მდგომარეობაში. ამ მდგომარეობაში ყოფნისას, ყველა დანარჩენი out-of-policy–ის მიზეზი უგულვებელყოფილია.

როდესაც PfR ახორციელებს გარკვეული ტრაფიკის კლასის მარშრუტის ცვლილებას, ეს ტრაფიკის კლასი გადადის hold-down მდგომარეობაში იმისათვის, რომ თავიდან იყოს აცილებული მარშრუტის ფლუქტუაცია. მარშრუტი იმყოფება ამ მდგომარეობაში hold-down პერიოდის განმავლობაში, რომელსაც მართავს სპეციალური მთვლელი. ამ მთვლელის მნიშვნელობა შეიძლება იყოს მითითებული გლობალურ კონფიგურაციაში.

2.7 აღსრულების ფაზა

საწყისი კონფიგურაციების თანახმად, PfR პროტოკოლი მოქმედებს შესწავლის, გაზომვის და წესის დართვის ფაზების დაკვირვების რეჟიმში. დაკვირვების რეჟიმში მასტერ კონტროლერი ახორციელებს ტრაფიკის კლასების და გამავალი ინტერფეისების მონიტორინგს, რომელიც ეფუძვნება საწყის კონფიგურაციებს ან მომხმარებლის მიერ მითითებულ წესებს. ამის შემდეგ ის ახორციელებს ქსელში არსებული ისეთი მოვლენების მოხსენებას, როგორის არის out-of-policy მდგომარეობა და ადგენს იმ გადაწყვეტილებებს, რომლებიც უნდა იყნენ მიღებული კერძო სიტუაციებში. მნიშნელოვანია აღინიშნოს, რომ ამ რეჟიმში მასტერ კონტროლერი არ ახდენს ქსელში არანაირ მოქმედებას ან ცვლილებას. PfR პროტოკოლის აღსრულების ფაზა მოქმედებს მხოლოდ კონტროლის

რეჟიმში, და არა დაკვირვების რეჟიმში. კონტროლის რეჟიმის გასააქტიურებლად საჭიროა მისი კონფიგურირება ბრძანებით mode route control.

კონტროლის რეჟიმში, მასტერ კონტროლერი ახორციელებს სასაზღვრო მარშრუტიზატორებისაგან მიღებული ინფორმაციის კოორდინაციას ზუსტად ისევე, როგორც დაკვირვების რეჟიმში, მაგრამ იმ განსხვავებით, რომ კონტროლის რეჟიმში მასტერ კონტროლერი სასაზღვრო მარშრუტიზატორებს აწვდის საკონტროლო ბრძანებებს. ამის შედეგად ხორციელდება PfR-ის მიერ კონროლირებადი ქსელების მარშრუტების შეცვლა წესების თანახმად.

PfR ახორციელებს მარშრუტების შეცვლას მაშინ, როდესაც ერთერთი შემდეგი პირობა სრულდება:

- ტრაფიკის კლასი გადადის out-of-policy მდგომარეობაში
- გამავალი ინტერფეისი გადადის out-of-policy მდგომარეობაში
- პერიოდული მთვლელის დრო ამოიწურა

PfR პროტოკოლის აღსრულების ფაზის შესრულებისას, მასტერ კონტროლერი აგრძელებს იმ ტრაფიკის კლასების მონიტორინგს, რომლებიც აკმაყოფილებენ წესით განსაზღვრულ წარმადობის მახასიათებლებს. ეს მექანიზმი იძლევა იმის გარანტიას, რომ ზემოთხსენებული ტრაფიკის კლასები შეინარჩუნებენ in-policy მდგომარეობას. ცვლილებები ხორციელდება მხოლოდ იმ ტრაფიკის კლასებისათვის ან ინტერფეისებისათვის, რომლებიც მოცემულ მომენტში იმყოფებიან out-of-policy მდგომარეობაში და ამ ცვლილებების შედეგად გადაინაცვლებენ in-policy მდგომარეობაში. ქსელში წარმადობის სასურველი დონის შესანარჩუნებლად, აუცილებელია ყურადღება მიექცეს იმ კონფიგურაციის ნაწილს, რომელიც ზეგავლენას ახდენს მასტერ კონტროლერის წესის საფუძველზე გადაწყვეტილების მიღებაზე.

მნიშვნელოვანია აღინიშნოს, რომ არსებობს გარემოებები, როდესაც PfR-ისათვის შეუძლებელი ან მეტად რთული ხდება ტრაფიკის კლასის in-

policy მდგომარეობაში მოქცევა. ასეთი გარემოება შეიძლება იყოს მაშინ, როდესაც წესში არის გამოყენებული დაყოვნების და პაკეტების დანაკარგის მკაცრი კონტროლი და ამავე დროს ინტერფეისი არის დაშვებულზე მეტად დატვირთული. ასეთ შემთხვევაში, იმის მიუხედავად, რომ ტრაფიკის კლასი იმყოფება out-of-policy მდგომარეობაში, მასტერ კონტროლერმა შეიძლება აირჩიოს ის არხი, რომლის მახასიათებლები ყველაზე მეტად უახლოვდებიან წარმადობის წესის მოთხოვნებს. თუ ეს არ მოხდება, მაშინ მასტერ კონტროლერი ამოიღებს მოცემულ პრეფიქს PfR პროტოკოლის კონტროლიდან. ეს პროტოკოლი იმის საშუალებას იძლევა, რომ მოხდეს არსებული და ხელმისაწვდომი გამტარუნარიანობის ოპტიმალური გამოყენება, მაგრამ დაშვებულზე მეტად დატვირთული არხების პრობლემის მოგვარება არ შედის PfR პროტოკოლის მოვალეობებში.

2.8 შემოწმების ფაზა

შემოწმების ფაზა წარმოადგენს PfR პროტოკოლის ალგორითმის ბოლო სტადიას და ითვალისწინებს პროტოკოლის კონტროლის ფაზის შესრულების შედეგების შემოწმებას. კერძოდ, ხორციელდება იმის შემოწმება, თუ შესრულდა ტრაფიკის ნაკადების მიმართულების ცვლილება და თუ კი ტრაფიკის კლასის ან ინტერფეისის წარმადობის მახასიათებლები იმყოფებიან in-policy მდგომარეობაში. PfR იყენებს NetFlow პროტოკოლს იმისათვის, რომ მოხერხდეს მარშრუტის კონტროლის ავტომატური შემოწმება. მასტერ კონტროლერი მოელის NetFlow ტრაფიკის კლასის ინფორმაციას ახალი ინტერფეისისაგან და უგულვებელყოფს NetFlow ინფორმაციას წინა მარშრუტების შესახებ. იმ შემთხვევაში, თუ NetFlow ინფორმაცია არ მიეწოდება მასტერ კონტროლერს ორი წუთის განმავლობაში, ის ახორციელებს ტრაფიკის კლასის გადაყვანას საწყის მდგომარეობაში. ტრაფიკის კლასი იმყოფება საწყის მდგომარეობაში მაშინ, როდესაც ის არ არის PfR პროტოკოლის კონტროლის ქვეშ.

NetFlow შემოწმების გარდა, PfR პროტოკოლი იყენებს სხვა ორ მეთოდს, რომლებიც შეიძლება იყვნენ გამოყენებული PfR–ის მიერ ქსელში განხორციელებული ცვლილებების შესამოწმებლად.

2.8.1 Syslog რეფორთი* (report)

იმისათვის, რომ ნათელი გახდეს თუ რა ცვლილებები იყო განხორციელებული PfR პროტოკოლის მიერ ქსელში, შესაძლებელია logging ბრძანების გამოყენება და ასევე ეს ფუნქცია იძლება იმის საშუალებას, რომ პროტოკოლმა ცვლილების შესახებ შეატყობინოს ქსელის ადმინისტრატორს იმ კონკრეტულ მომენტში, როდესაც განხორციელდა რაიმე ცვლილება. მასტერ კონტროლერი ელოდება გარკვეული პრეფიქსისათვის ორმხრივ ტრაფიკს და syslog ბრზანება იძლევა ამის დამოწმების საშუალებას. [35]

2.8.2 შემოწმების ბრძანებები

ქსელში განხორციელებული ცვლილებების შესახებ ინფორმაციის მოსაძიებლად არსებობს ალტერნატიული მეთოდი, რომელიც მდგომარეობს PfR show ბრძანებების გამოყენებაში. ამის გარდა, show ბრძანების დახმარებით შესაძლებელია პრეფიქსის მდგომარეობის შემოწმება, კერძოთ თუ კი ის იმყოფება in-policy მდგომარეობაში. მონიტორებადი პრეფიქსების მდგომარეობის შესამოწმებლად შესაძლებელია გამოიყენოს show pfr master prefix ბრძანება. ამ ბრხანების მიერ გამოტანილი ინფორმაცია მოიცავს მიმდინარე გამავალ ინტერფეისს, პრეფიქსის დაყოვნებას, შემომავალი და გამავალი ინტერფეისების გამტარუნარიანობას და ასევე კონკრეტული სასაზღვრო მარშრუტიზატორისაგან მიღებული მარშრუტის ინფორმაციას. სასაზღვრო მარშრუტიზატორის მიერ კონტროლირებადი პრეფიქსების შესახებ ინფორმაციის მოსაძიებლად შეიძლება გამოიყენოს ბრძანება show pfr border routes.

მნიშვნელოვანია აღინიშნოს, რომ PfR მოდელის უდავო უპირატესობების მიუხედავად, მას გააჩნია ნაკლოვანობები, რომელთა

შორის უმნიშვნელოვანესი არის მწარმოებლის ფიზიკურ მოწყობილობებისადმი დამოკიდებულება. კერძოდ, ამ მეთოდის ქსელში დასანერგად აუცილებელია Cisco-ს აპარატურის გამოყენება. ამ პრობლემის მოგვარების გზას წარმოადგენს მწარმოებლისაგან დამოუკიდებელი პროტოკოლის შექმნა და დანერგვა, რაც წარმოდგენილია შემდეგ ქვეთავში. ამ მოდელს ეწოდება ინტელექტუალური მარშრუტიზაცია და მის შემადგენლობაში შედიან მხოლოდ სტანდარტული ღია კოდის მქონე კომპონენტები.

2.9 ინტელექტუალური მარშრუტიზაციის მოდელი

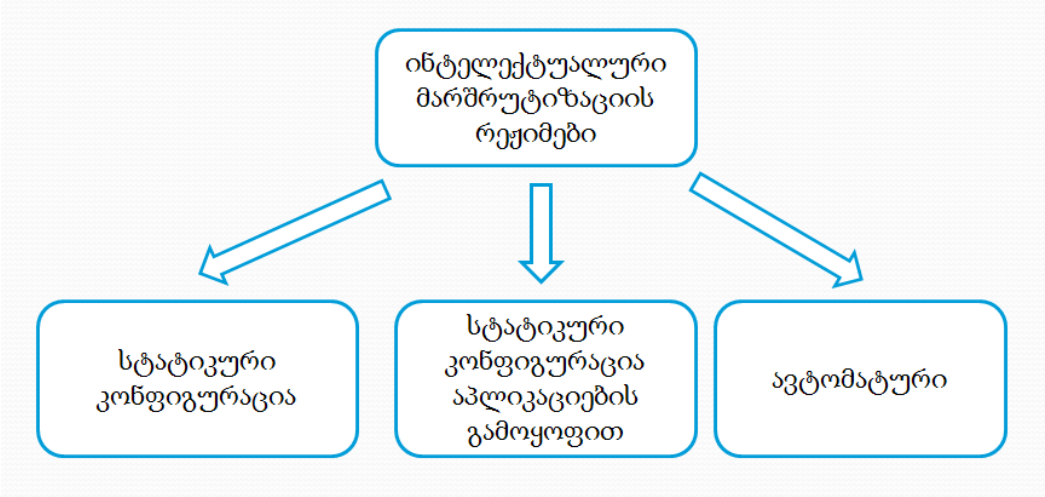
თანამედროვე ქსელებში IP კავშირის უზრუნველყოფისათვის გამოიყენება დინამიკური მარშრუტიზაციის პროტოკოლი. ინტერნეტთან კავშირის დამყარებისათვის მსხვილი ორგანიზაციები იყენებენ პროტოკოლს BGP, რომელიც წარმოადგენს დინამიკური მარშრუტიზაციის პროტოკოლს. ის გამოიყენება ინტერნეტში ძირითადი მარშრუტიზაციის გადაწყვეტილებების მისაღებად. ის მოიცავს და მუშაობს IP ქსელების ცხრილებთან ან პრეფიქსებთან, რის შედეგად შესაძლებელი ხდება სხვადასხვა ავტონომიურ სისტემებს შორის კავშირის დამყარება. BGP წარმოადგენს ვექტორული მარშრუტის ტიპის პროტოკოლს, რაც წარმოადგენს მანძილთა ვექტორის მიხედვით მარშრუტიზაციის ვარიაციას.

ინტელექტუალური მარშრუტიზაციის ალგორითმის ნაწილს წარმოადგენს BGP პროტოკოლთან ურთიერთქმედება. ამის შედეგად შესაძლებელი ხდება BGP პროტოკოლის გადართვა ინტელექტუალური მარშრუტიზაციის მიერ არჩეულ მარშრუტზე. ამის მისაღწევად გამოიყენება როგორც local preference ასევე community ატრიბუტების მორგება.

<u>მუშაობის 3 რეჟიმი</u>

ინტელექტუალური მარშრუტიზაციის მოდელში შეიძლება გამოიყოს მუშაობის სამი ძირითადი რეჟიმი. ესენია სტატიკური კონფიგურაცია, სტატიკური კონფიგურაცია აპლიკაციების გამოყოფით და ავტომატური

კონფიგურაციის რეჟიმი. შესაძლებელია სამივე რეჟიმის ერთდროულად გამოყენება სხვადასხვა ნაკადებისათვის. კონკრეტული ნაკადისათვის ხორციელდება ხელსაყრელი რეჟიმის არჩევა სხვადასხვა ფაქტორებზე დაყრდნობით.



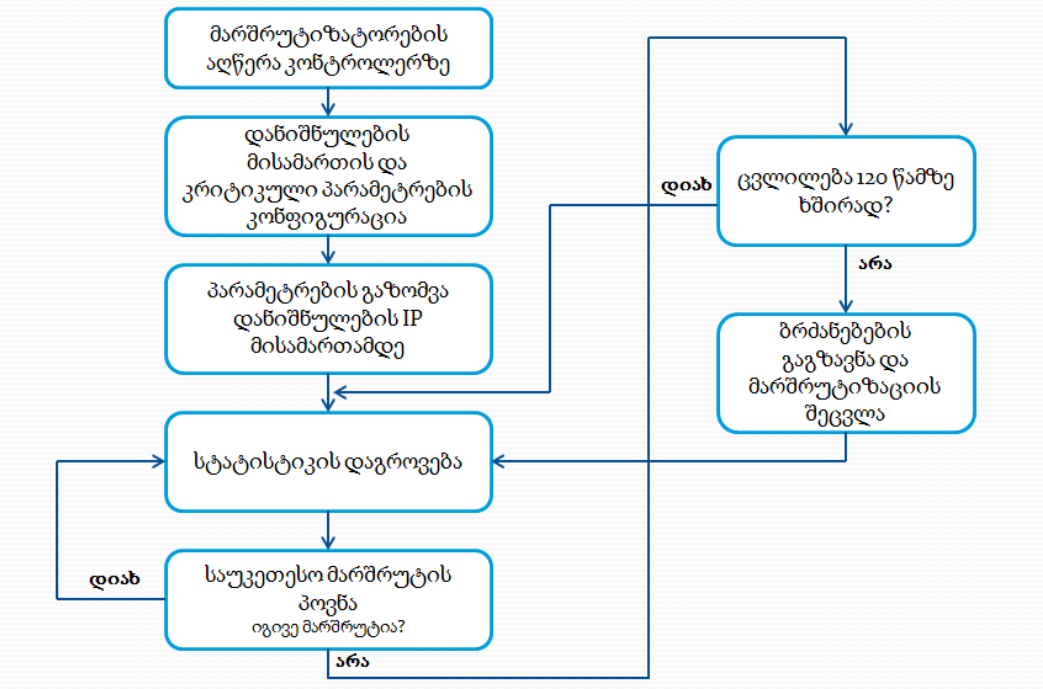
ნახ.2.9 ინტელექტუალური მარშრუტიზაციის მუშაობის სამი რეჟიმი

ეს სამი რეჟიმი იძლევა იმის საშუალებას, რომ ქსელის მიმდინარე გარემოებისამებრ განხორციელდეს ინტელექტუალური მარშრუტიზაციის მოქმედების მორგება. თითოეულ რეჟიმს გააჩნია თავისი უპირატესობები და შეზღუდვები და ისინი განხილულია შემდეგ ქვეთავებში.

2.9.1 სტატიკური კონფიგურაცია

სტატიკური კონფიგურაციის რეჟიმის გამოყენების ძირითადი მიზანი მდგომარეობს ქსელში არსებული ნაკადების სტატიკურ მითითებაში. ამ რეჟიმის გამართვა ხელსაყრელია იმ შემთხვევაში, როდესაც ცნობილია კონკრეტული ნაკადების პარამეტრები და ისინი უცვლელი რჩებიან. ასეთ შემთხვევაში ხორციელდება აპლიკაციების ნაკადების მახასიათებელი პარამეტრების კონფიგურირება მარშრუტიზაციის კონტროლერზე და ამ ინფორმაციაზე დაყრდნობით სხვადასხვა ჯგუფების შექმნა.

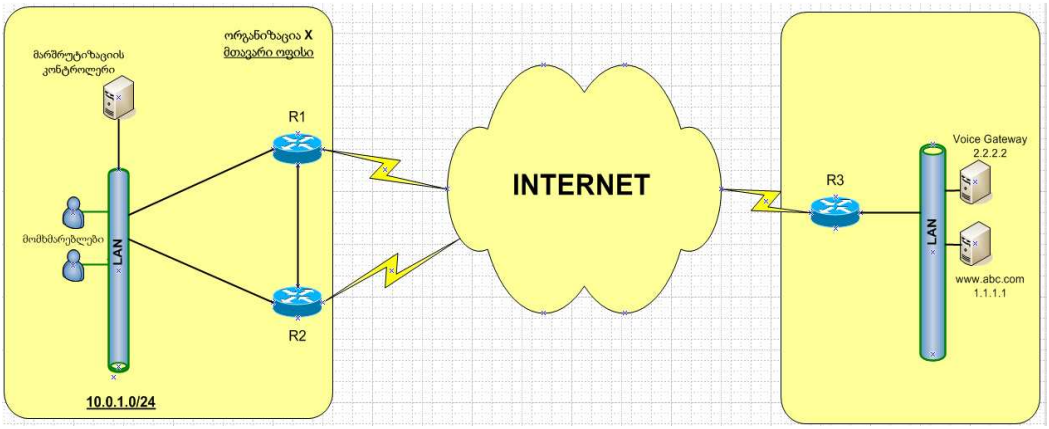
სტატიკური კონფიგურაციის რეჟიმის ალგორითმი წარმოდგენილია ნახაზზე 2.10.



ნახ.2.10 ინტელექტუალური მარშრუტიზაციის სტატიკური კონფიგურაციის რეჟიმის ალგორითმი

ინტელექტუალური მარშრუტიზაციის მოდელის უმთავრეს ნაწილს წარმოადგენს მარშრუტიზაციის კონტროლერი. ამ მოწყობილობის ძირითადი მიზანია ქსელში არსებული ყველა კარიბჭის კონტროლი. კერძოდ, მის მოვალეობებში შედის აპლიკაციების ნაკადების დადგენა (ავტომატური რეჟიმი) ან განსაზღვრა (სტატიკური რეჟიმი). ამ ინფორმაციაზე დაყრდნობით ნაკადების ჯგუფების შექმნა და ჯგუფში პარამეტრების პრიორიტეტების მითითება. აგრეთვე, ყველა მარშრუტის მომსახურების ხარისხის მახასიათებლების გაზომვა, მარშრუტიზატორებისათვის ბრძანებების მიწოდება და სხვა.

ინტელექტუალური მარშრუტიზაციის ტიპიური ტოპოლოგია წარმოდგენილია ნახაზზე 2.11.



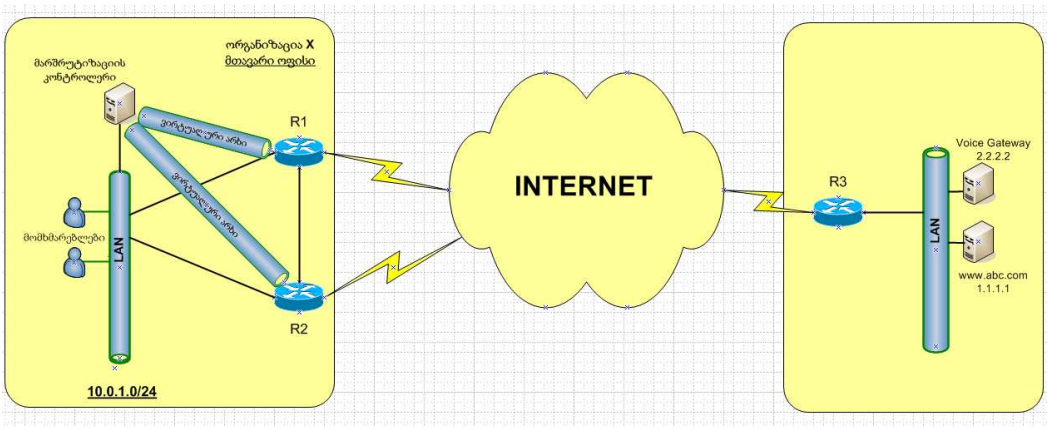
ნახ. 2.11 ინტელექტუალური მარშრუტიზაციის მოდელი იყენებს მარშრუტიზაციის კონტროლერს

მარშრუტიზაციის კონტროლერი წარმოადგენს სერვერს Linux ოპერაციული სისტემის ბაზაზე, რომელიც აღჭურვილია სხვადასხვა საზომი ხელსაწყოებით და დამატებითი უტილიტებით, ასევე NetFlow ვერსია 9 და IPFIX სტატისტიკის ანალიზის პროგრამებით. ფიზიკურ დონეზე, კონტროლერი შეერთებულია ლოკალური ქსელის კომუტატორთან ისევე, როგორც დანარჩენი არსებული ქსელური მოწყობილობები.

პირველ რიგში, ინტელექტუალური მარშრუტიზაციის გასამართად აუცილებელია ქსელში არსებული მარშრუტიზატორების (კარიბჭეების) აღწერა მარშრუტიზაციის კონტროლერზე. ამის განსახორციელებლად მარშრუტიზაციის კონტროლერი თითოეულ კარიბჭესთან ამყარებს ვირტუალურ არხებს. ასეთი არხების გასამართად გამოიყენება დაშიფრული არხების SSL ტექნოლოგია. [36]

ვირტუალური არხების გამართვის შემდეგ შესაძლებელი ხდება მარშრუტიზატორებთან პირდაპირი კავშირის დამყარება. ვირტუალური არხის მეშვეობით ხორციელდება მარშრუტიზატორიდან სტატისტიკის გადაგზავნა კონტროლერზე, ასევე კონტროლერიდან წარმოშობილი სინჯების შემდგომ გადაგზავნა და ბრძანებების მიწოდება. მარშრუტიზატორებთან კავშირის დამყარების შემდეგ კონტროლერი ახორციელებს აპლიკაციების ნაკადების დაჯგუფებას კრიტიკული

პარამეტრების მიხედვით. ყოველ ჯგუფში მომსახურების ხარისხის პარამეტრებს ენიჭებათ პრიორიტეტები.



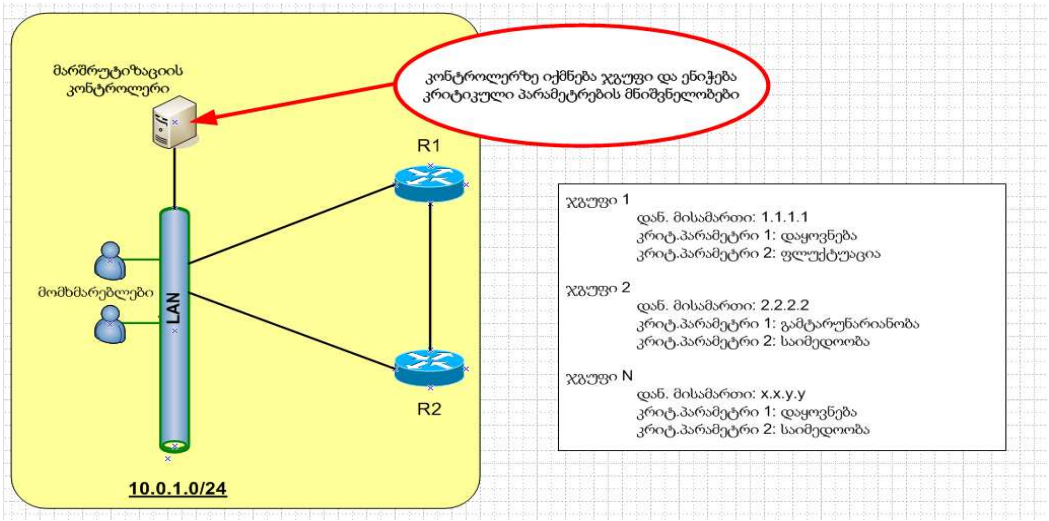
ნახ.2.12 მარშრუტიზატორებთან კავშირის დასამყარებლად კონტროლერი იყენებს ვირტუალურ არხებს.

პრიორიტეტის მნიშვნელობა განსაზღვრავს თითოეული პარამეტრის უპირატესობას ჯგუფში, და შესაბამისად განსაზღვრავს ამ პარამეტრის საზომი სინჯების მნიშვნელობას. ჯგუფების შექმნა შესაძლებელია დანიშნულების მისამართის მიხედვით და დასაშვებია ერთი ან მეტი პარამეტრის მითითება. აპლიკაციების ნაკადების დაჯგუფების მაგალითი მოყვანილია ნახაზზე 2.13, სადაც იქმნება ჯგუფები დანიშნულების მისამართის მიხედვით და თითოელ მათგანს ენიჭება ორი კრიტიკული პარამეტრი.

ჯგუფი 1 მოიცავს ყველა აპლიკაციას, რომლების დანიშნულების მისამართი არის 1.1.1.1 და ხორციელდება ამ მისამართამდე დაყოვნების და დაყოვნების ვარიაციის (ფლუქტუაციის) გაზომვა.

ჯგუფი 2 მოიცავს ყველა იმ აპლიკაციების ნაკადებს, რომლების დანიშნულების მისამართი არის 2.2.2.2 და ამ აპლიკაციების ნაკადებისათვის ხორციელდება გამტარუნარიანობის და საიმედოობის გაზომვა. ამ პარამეტრების განსაზღვრა შესაძლებელია როგორც ადმინისტრატორის ჩარევით, ასევე ავტომატურად საყოველთაოდ ცნობილი

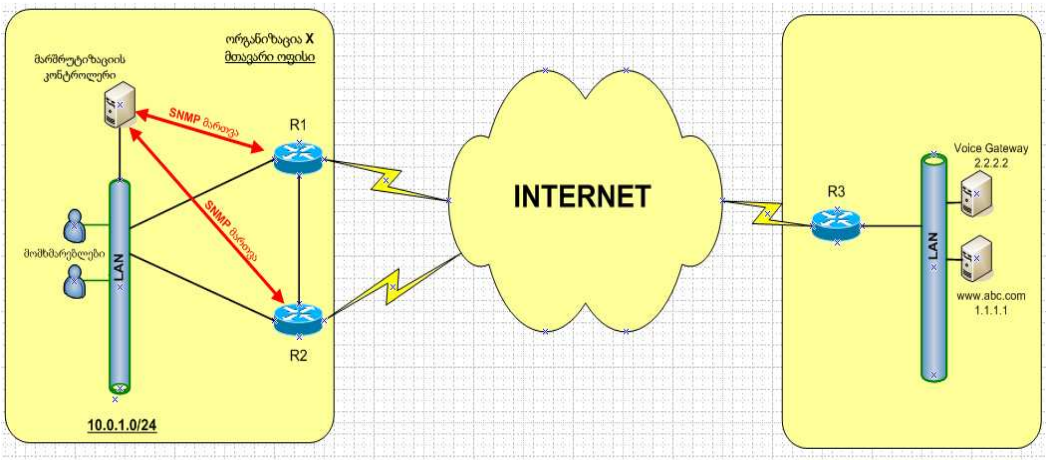
აპლიკაციების ნუსხის შემოწმებით და შესაბამისი კრიტიკული პარამეტრების დადგენით.



ნახ. 2.13 მარშრუტიზაციის კონტროლერი ქმნის აპლიკაციების ჯგუფებს

ჯგუფების დანიშნულების მისამართებამდე გაზომვების შედეგების საფუძველზე, კონტროლერი ღებულობს გადაწყვეტილებას დატოვოს არსებული საუკეთესო მარშრუტი ან აირჩიოს ალტერნატიული მიმართულება. ალტერნატიული მარშრუტის არჩევის შემთხვევაში, კონტროლერი ახორციელებს შესაბამისი ბრძანებების გაცემას, რომლებიც მიეწოდებათ მარშრუტიზატორებს SNMP პროტოკოლის მეშვეობით. [37] ეს პროტოკოლი წარმოადგენს ღია კოდზე აგებულ სტანდარტს და ხელმისაწვდომია ნებისმიერი მწარმოებლის მარშრუტიზატორებზე.

ასეთი კავშირის დასამყარებლად აუცილებელია კონკრეტული მწარმოებლის მოწყობილობის პარამეტრების მითითება, კერძოდ SNMP MIB და COMMUNITY სტრიქონის განსაზღვრა.

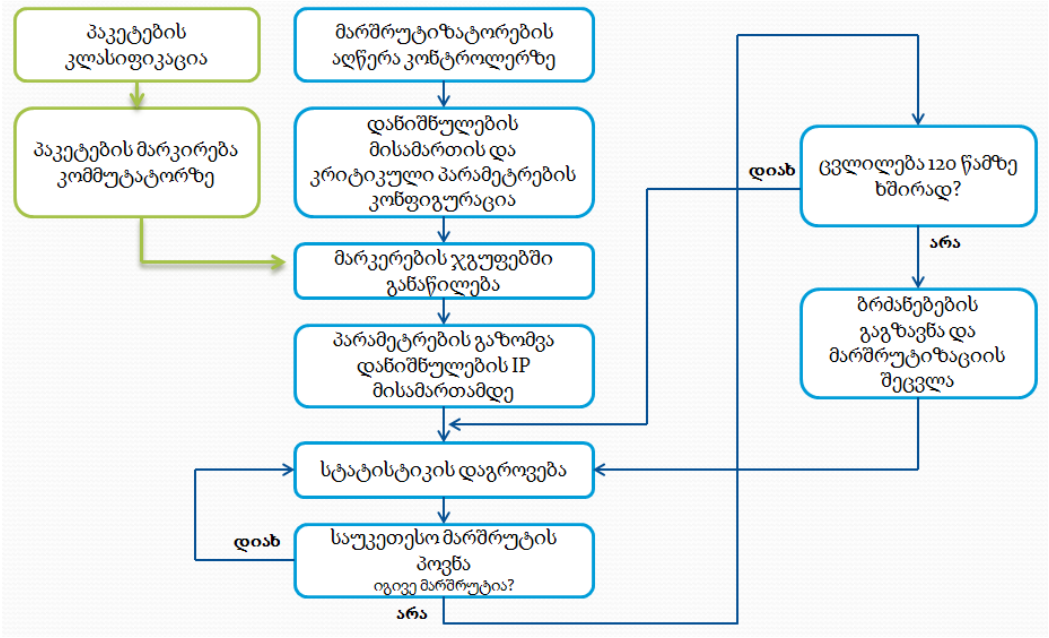


ნახ.2.14 მარშრუტიზატორების მართვისათვის კონტროლერი იყენებს SNMP პროტოკოლს

2.9.2 სტატიკური კონფიგურაცია აპლიკაციების გამოყოფით

ინტელექტუალური მარშრუტიზაციის ალგორითმის შემდეგ რეჟიმს წარმოადგენს სტატიკური კონფიგურაცია აპლიკაციების გამოყოფით. ამ რეჟიმის გამოყენებისას ხორციელდება აპლიკაციების ნაკადების წინასწარი მარკირება. მარკირების საშუალებებს წარმოადგენენ DSCP, ToS ან IP Precedence მეთოდები. ამ მეთოდების საშუალებით ხორციელდება პაკეტის ველების წინასწარი განსაზღვრა ქსელის საზღვრებთან და ამ ველების მიხედვით მათი შემდგომი კლასიფიკაცია და დაჯგუფება კონტროლერზე.

მოცემული რეჟიმის ალგორითმის სქემა წარმოდგენილია ნახაზზე 2.15 და მისი მუშაობის პრინციპი სტატიკური კონფიგურაციის რეჟიმის ალგორითმის მსგავსია. ძირითად განსხვავებას წარმოადგენს პაკეტების კლასიფიკაციის და აპლიკაციების ნაკადების განსაზღვრისათვის გამოყენებული მეთოდები.

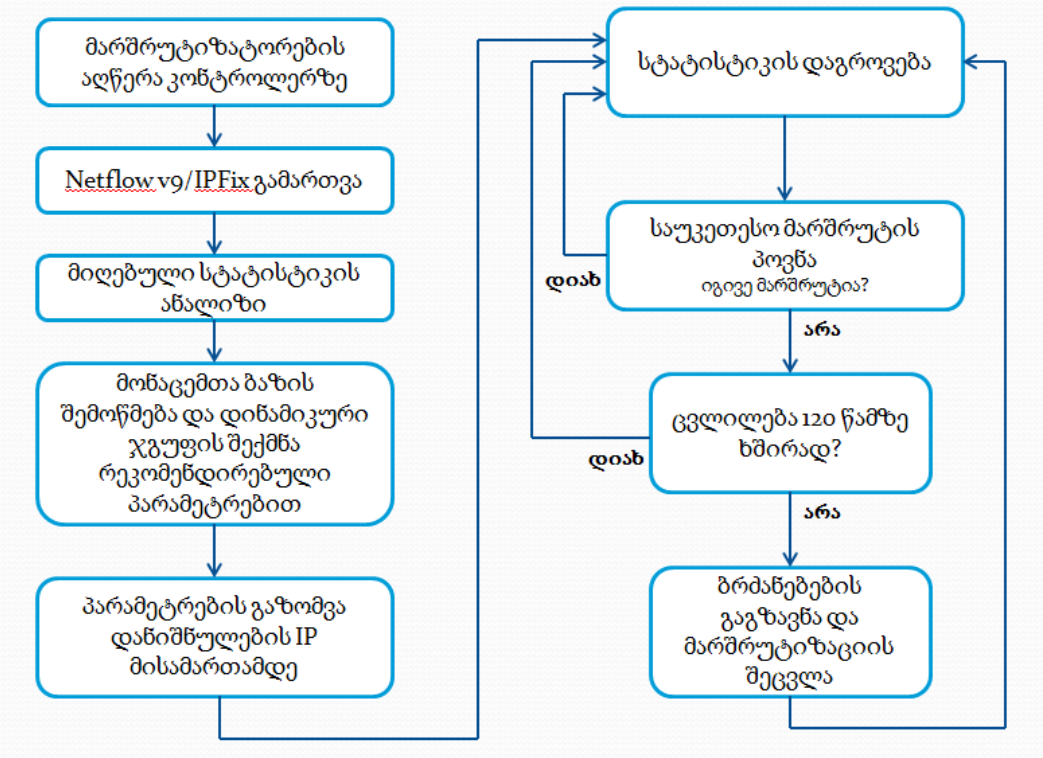


ნახ.2.15 ინტელექტუალური მარშრუტიზაციის სტატიკური კონფიგურაცია აპლიკაციების გამოყოფით რეჟიმის ალგორითმი

2.9.3 ავტომატური კონფიგურაცია

ავტომატური კონფიგურაციის რეჟიმი დაყრდნობილია მარშრუტიზატორებისაგან პასიური სტატისტიკის დაგროვებაზე და ამ სტატისტიკის შემდგომ კონტროლერის მიერ ანალიზზე. სტატისტიკის დაგროვება ხორციელდება NetFlow ვერსია 9 და IPFIX პროტოკოლების მეშვეობით. ეს პროტოკოლები ახორციელებენ მარშრუტიზატორში გამავალი ყველა პაკეტის აღრიცხვას და ამ ინფორმაციის პერიოდულ გადაცემას კონტროლერზე. სტატისტიკის მიღების და ანალიზის შემდეგ, კონტროლერი ადგენს ქსელში ყველაზე აქტიურ ნაკადებს, ე.ი. დანიშნულების მისამართებს და ახორციელებს მათ ჯგუფებში ჩამატებას. ჯგუფის პარამეტრების დასადგენად, კონტროლერი მიმართავს მონაცემთა ბაზას, რომელშიც მითითებულია საყოვლეთაოდ ცნობილი აპლიკაციების მოთხოვნები და ამ ინფორმაციის საფუძველზე ხორციელდება ჯგუფში პარამეტრების ჩამატება და მათი პრიორიტეტების მითითება. იმ შემთხვევაში, თუ კი ვერ მოხერხდა აპლიკაციის ზუსტი სახეობის დადგენა ან მოცემული აპლიკაცია არ მოიძებნება მონაცემთა ბაზაში, დინამიკური

ჯგუფის პარამეტრების მითითება ხორციელდება საწყისი მიმდევრობით, ე.ი. დაყოვნება, საიმედოობა, დაყოვნების ვარიაცია და გამტარუნარიანობა. ავტომატური კონფიგურაციის რეჟიმის ალგორითმი წარმოდგენილია ნახაზზე 2.16.



ნახ.2.16 ინტელექტუალური მარშრუტიზაციის ავტომატური რეჟიმის ალგორითმი

იმ შემთხვევაში, როდესაც აუცილებელია ჯგუფისათვის საუკეთესო მარშრუტის შეცვლა, ინტელექტუალური მარშრუტიზაცია ახორციელებს მარშრუტის ცვლილების სიხშირის დადგენას. იმ შემთხვევაში, თუ ეს მნიშვნელობა აღემატება 120 წამს, ხორციელდება ახალი მარშრუტის შეცვლის ბრძანებების შედგენა და მათი მარშრუტიზატორებამდე მიწოდება. ეს პირობა უზრუნველყოფს მარშრუტის სტაბილურობას.

II თავის დასკვნა

1. გაანალიზებულია არსებული მაღალი წარმადობის მარშრუტიზაციის მოდელი და ალგორითმი. ჩამოყალიბებულია ამ მოდელის მუშაობის პრინციპი. გაანალიზებულია მისი ნაკლოვანობები, მათ შორის ყველაზე მნიშვნელოვანია მისი ურთიერთდამოკიდებულება მწარმოებლის მოწყობილობებისადმი. დასაბუთებულია ამ პრობლემის მოგვარების გზა, რომელიც მდგომარეობს მწარმოებლისაგან დამოუკიდებელი მეთოდის შემუშავებაში.
2. შემუშავებულია ინტელექტუალური მარშრუტიზაციის მოდელი, რომელიც ითვალისწინებს ქსელის მომსახურების ხარისხის მდგომარეობას და სხვადასხვა კრიტიკული მახასიათებლების გაზომვას.
3. შემუშავებულია ალგორითმი, რომელსაც საფუძვლად უდევს ძირითადი მოთხოვნა, რომ მარშრუტიზატორებს უნდა შეეძლოთ ნაკადების მარშრუტიზაცია არა მარტო დანიშნულების პრეფიქსების მიხედვით, არამედ აპლიკაციების ტიპებზე დაყრდნობით. მარშრუტიზაცია ირჩევს აპლიკაციისათვის საუკეთესო მარშრუტს და არა მხოლოდ დანიშნულების პრეფიქსისათვის. შემუშავებული მეთოდის შედეგად, ქსელში უეცარი ცვლილებების შემთხვევაში, მარშრუტიზატორები ახდენენ სწრაფ რეაგირებას და ხორციელდება ნაკადების გადანაწილება სასურველი მიმართულებებით.
4. შემუშავებული არლგორითმი იყენებს ტრაფიკის მართვის ინტელექტუალურ მეთოდს, რომელსაც შეუძლია დინამიკური მარშრუტიზაციის გადაწყვეტილებების მიღება ქსელის არხების შეცდომებზე დაყრდნობით და ასევე მარშრუტიზაციის კორექტირება შემდეგი კრიტერიუმების გათვალისწინებით: პაკეტების მიწოდების დაყოვნება, პაკეტების დანაკარგი, დაყოვნების ვარიაციის ფაქტორი, არხების დატვირთვა და სხვა წესები.

5. ინტელექტუალური მარშრუტიზაციის პროტოკოლი აუმჯობესებს ტრადიციულ მარშრუტიზაციას წარმადობაზე გავლენის მქონე პარამეტრების რეალურ რეჟიმში კონტროლით.
6. შემუშავებული მოდელების საფუძველზე აგებულია ალგორითმი, რომელიც ითვალისწინებს არსებული მარშრუტების მომსახურების ხარისხის მახასიათებლების მნიშვნელობებს და ახორციელებს ნაკადების ჯგუფებად დაყოფას. ჯგუფების შექმნა ხორციელდება აპლიკაციების პრიორიტეტული მახასიათებლის მიხედვით.

ექსპერიმენტული ნაწილი
თავი 3

ინტელექტუალური მარშრუტიზაციის რეალიზაცია

მოცემულ თავში განხილულია ინტელექტუალური მარშრუტიზაციის მოდელის რეალიზაციის საკითხები. ჩატარებულია ამ მოდელის შემადგენელი უტილიტების მუშაობის პრინციპების განხილვა. შემუშავებულია მათი პრაქტიკული გამოყენების მეთოდები ინტელექტუალური მარშრუტიზაციის ალგორითმში.

ჩატარებულია შემუშავებული ალგორითმის აპრობაცია ლაბორატორიულ პირობებში. გაანალიზებულია მიღებული შედეგები და წარმოდგენილია მათი გრაფიკული გამოხატულება.

3.1 აპლიკაციების კლასიფიცირება

ინტელექტუალური მარშრუტიზაციის ალგორითმის პირველი ფაზა მოიცავს ქსელში მომუშავე სხვადასხვა ტიპის აპლიკაციების აღმოჩენას და მათი კრიტიკული პარამეტრების დადგენას. ამ პარამეტრების მიხედვით, ალგორითმი ახორციელებს აპლიკაციების დაჯგუფებას და ყოველი ჯგუფის წესის განსაზღვრას. [38]

აპლიკაციების დაყოფა შესაძლებელია სხვადასხვა კლასებად და ყველა ტიპის აპლიკაციას გააჩნია მისთვის კრიტიკული მომსახურების ხარისხის პარამეტრი. სხვადასხვა ტიპის აპლიკაციების კავშირი მომსახურების ხარისხის ფუნდამენტალურ პარამეტრებთან მიმართებაში წარმოდგენილია ცხრილში 3.1.

აპლიკაცია	სანდოობა	დაყოვნება	დაყოვნების ვარიაცია	გამტარუნარია–ნობა
ელ. ფოსტა	მაღალი	დაბალი	დაბალი	დაბალი
ინფ. გადაცემა	მაღალი	დაბალი	დაბალი	საშუალო
ვებ ბრაუზინგი	მაღალი	საშუალო	დაბალი	საშუალო
მოშორებული წვდომა	მაღალი	საშუალო	საშუალო	დაბალი

აუდიო მოთხოვნით	დაბალი	დაბალი	მაღალი	საშუალო
ვიდეო მოთხოვნით	დაბალი	დაბალი	მაღალი	მაღალი
ტელეფონია	დაბალი	მაღალი	მაღალი	დაბალი
ვიდეო კონფერენცია	დაბალი	მაღალი	მაღალი	მაღალი

ცხრილი 3.1 სხვადასხვა ტიპის აპლიკაციების მოთხვოვნები მომსახურების ხარისხის მიმართ [39]

აპლიკაციის ნაკადი შეიძლება იყოს დახასიათებული OSI მოდელის მესამე და მეოთხე შრეების სათაურების მიხედვით. იმისათვის, რომ პაკეტები $P_1, P_2 \ldots P_n$ მიეკუთვნონ ერთსა და იმავე ნაკადს G_m აუცილებელია შესრულდეს პირობა, რომელიც მდგომარეობს იმაში, რომ ყველა პაკეტს უნდა გააჩნდეს ერთი და იგივე წყაროს მისამართი და მსგავსი პირობა უნდა ვრცელდებოდეს მათი დანიშნულების მისამართების მიმართებაში. ამ შემთხვევაში პაკეტები $P_1, P_2 \ldots P_n$ მიეკუთვნებიან ერთსა და იგივე ჰოსტების ნაკადებს, მაგრამ ეს არ ნიშნავს იმას, რომ ეს ნაკადები მიეკუთვნებიან ერთსა და იგივე აპლიკაციას. ამისათვის საჭიროა, რომ ამ პაკეტების მეოთხე შრის პორტები იყვნენ ერთი და იგივე.

ამის გარდა, ნაკადის იდენტიფიცირებისათვის ასევე შეიძლება გამოიყენებოდეს IP პაკეტის სათაურის ToS ველი, რომელიც შეიძლება მიუთითებდეს აპლიკაციისათვის სასურველ მომსახურების ხარისხის დონეს.

აპლიკაციების ნაკადების იდენტიფიცირებისათვის უცილებელია პაკეტების სათაურების შემოწმება და შემდგომ მათი ანალიზი. ამის განსახორციელებლად შესაძლებელია როგორც სტატისტიკის შეგროვების მეთოდით (NetFlow ვერსია 9 ან IPFIX) და ასევე ხელით კონფიგურირების ხერხით.

NetFlow ვერსია 9 [40] წარმოადგენს ტრაფიკის მონიტორინგის პროტოკოლს, რომელიც მოქმედებს NetFlow ქეშის შექმნის პრინციპით. ეს ქეში შეიცავს ჩანაწერებს ქსელში ყველა აქტიური ნაკადების შესახებ.

NetFlow ქეშის შესადგენად პროტოკოლი ახორციელებს ყველა ნაკადის პირველი პაკეტის პროცესორით დამუშავებას და ამის შედეგად შესაძლებელი ხდება პაკეტის სათაურის წაკითხვა და ქეშში შენახვა. ნაკადების სტატისტიკისათვის, მარშრუტიზატორი იყენებს პირველი პაკეტის სათაურის ინფორმაციას სხვა მიღებული პაკეტების დასადგენად და ამის საფუძველზე ხორციელდება პაკეტების სტატისტიკის შეგროვება. შესაძლებელია ამ სტატისტიკის პერიოდული გადაგზავნა სერვერზე შემდგომი ანალიზისათვის. ამ სტატისტიკის ანალიზის შემდეგ ხორციელდება ნაკადების კლასებად დაყოფა.

NetFlow ვერსია 9 აღწერილია RFC-3954 მაგრამ არ წარმოადგენს სტანდარტს. IETF-ის მიერ ჩამოყალიბებულ სტანდარტს წარმოადგენს IPFIX [41] პროტოკოლი, რომელიც NetFlow მსგავსად ახორციელებს ნაკადების სტატისტიკის შეგროვებას და ამ ინფორმაციის სერვერზე გადაგზავნას. ამ კავშირის დასამყარებლად გამოიყენება პროტოკოლი SCTP. [42] [43]

ნაკადების იდენტიფიცირების ხელით კონფიგურირების აუცილებელია წინასწარ განსაზღვრული პარამეტრების მითითება და შემდგომ ამ პარამეტრებზე დაყრდნობით პაკეტების ანალიზი. მაგალითად, შესაძლებელია მარკირებული პაკეტების დადგენა და ასეთი პაკეტებისათვის ცალკე ჯგუფის შექმნა. ასეთი მეთოდის უპირატესობას წარმოადგენს ადმინისტრატორის მიერ მითითებული პარამეტრების მიხედვით არასტანდარტული აპლიკაციების ნაკადების აღმოჩენა.

3.2 ხელმისაწვდომი გამტარუნარიანობის საზომი მეთოდები

IP ქსელებში კვანძთაშორის ინფორმაციის წარმატებული გადაცემის საშუალო სიდიდეს ეწოდება ხელმისაწვდომი გამტარუნარიანობა. ამ ინფორმაციის მიწოდება შესაძლებელია როგორც ფიზიკური ასევე ლოგიკური არხებით და ასევე სხვადასხვა ქსელის კვანძების გამოყენებით. როგორც წესი, გამტარუნარიანობა გამოიხატება ბიტები/წამში ერთეულით და ცალკეულ შემთხვევებში პაკეტები/წამში ერთეულის მეშვეობით. ამ

მახასიათებლის ზუსტი და აკურატული გაზომვა წარმოადგენს უმნიშვნელოვანეს ამოცანას იმისათვის, რომ იყოს მიღწეული მომხმარებლის აპლიკაციების მუშაობის მაღალი წარმადობა და ხარისხი. ხელმისაწვდომი გამტარუნარიანობა ასევე წარმოადგენს ქსელის გადატვირთულობის არიდების ალგორითმების და ინტელექტუალური მარშრუტიზაციის სისტემების უმნიშვნელოვანეს ნაწილს.

მიღწევადი გამტარუნარიანობის საზომი ხელსაწყოები შეიძლება დაიყოს ორ ძირითად ნაწილად – პასიური და აქტიური საზომი მექანიზმები. პასიური გაზომვა ხორციელდება ქსელში არსებული ტრაფიკის მონიტორინგის საშუალებით და არ ახდებს არანაირ ზეგავლენას ქსელის ფუნქციონირებაზე. იმისათვის, რომ მოხერხდეს ნაკადების ინფორმაციის დაგროვება, საჭიროა არხების სრული დატვირთულობის დამუშავება და ყველა შუამავალი კვანძებისადმი წვდომის განხორციელება. პასიური მონიტორინგის მკაფიო მაგალითს წარმოადგენს NetFlow და IPFIX პროტოკოლები.

აქტიური გაზომვა ხორციელდება სინჯების უშუალო წარმოშობით და მათი ქსელში გაგზავნით, რის შედეგად შესაძლებელი ხდება ქსელის წარმადობის მახასიათებლების დადგენა. იმის მიუხედავად, რომ აქტიური სინჯები წარმოადგენენ ქსელის დამატებით დატვირთვას, ეს მეთოდი წარმოადგენს კვანძთაშორის მიღწევადი გამტარუნარიანობის გაზომვის ყველაზე ოპტიმალურ მეთოდს. არსებობს ბევრი ტიპის და კლასის გამტარუნარიანობის აქტიური საზომი ხელსაწყო და მეთოდი და მათი ზოგადი აღწერა არის წარმოდგენილი ქვემოთ.

გამტარუნარიანობა შეიძლება იყოს გამოხატული ორი მახასიათებლით, რომლებიც არიან ფიზიკური და მიღწევადი გამტარუნარიანობა და არც ერთი მათგანი არ არის დამოკიდებული არც ბოლო კვანძზე და არც გამოყენებულ პროტოკოლზე.

ფიზიკური გამტარუნარიანობა, ე.ი. ტევადობა (C), წარმოადგენს მაქსიმალურ ბიტთა რაოდენობას წამში, რომლის გადაცემა შეუძლია

ქსელის ელემენტს. კვანძთაშორის მარშრუტის ფიზიკური გამტარუნარიანობა განისაზღვრება მარშრუტის გასწვრივ ყველაზე მცირე ზომის არხით.

დატვირთულობა (U) წარმოადგენს ტევადობის პროცენტულ ნაწილს, რომელიც მოცემულ მომენტში გამოიყენება არხის ან მარშრუტის ჯამური ტრაფიკის მიერ.

$U = \frac{Traffic}{C}$

ხელმისაწვდომი გამტარუნარიანობა (A) გამოითვლება შემდეგნაირად: ტევადობას გამოკლებული დატვირთულობის შეფარდება დროის მოცემულ ინტერვალთან. შემდეგი ფორმულა გამოიყენება მარშრუტის ხელმისაწვდომი გამტარუნარიანობის გამოსათვლელად:

$A(t_s, t_e)$ = ტევადობა – ტრაფიკი= $C \times (1 - U) \neq A(T_{ინტერვალი})$

$T_{ინტერვალი} = t_s - t_e$

t_s – წარმოადგენს გაზომვის საწყის დროს.

t_e –წარმოადგენს გაზომვის დასასრულის დროს.

მიღწევადი გამტარუნარიანობა წარმოადგენს ქსელში ორ კვანძს შორის წარმატებულად გადაცემული ინფორმაციის რაოდენობას წამში. ეს პარამეტრი შეიძლება იყოს შეზღუდული მარშრუტის გასწვრივ მყოფი კვანძების როგორც ფიზიკური ასევე პროგრამული უზრუნველყოფით. მაქსიმალური მიღწევადი გამტარუნარიანობის კონტექსტში შეიძლება გამოიყენოს ოთხი სხვადასხვა პარამეტრი, რომლებიც აღნიშნავენ კონცეპტუალური წარმადობის ზედა ზღვარს. ამ ოთხ პარამეტრს წარმოადგენენ: მაქსიმალური თეორიული გამტარუნარიანობა, მაქსიმალური მიღწევადი გამტარუნარიანობა, გამტარუნარიანობის უმაღლესი გაზომილი მნიშვნელობა და მაქსიმალური მდგრადი გამტარუნარიანობა. ეს ოთხი პარამეტრი წარმოადგენენ განსხვავებულ მნიშვნელობებს და მაქსიმალური გამტარუნარიანობის მნიშვნელობების შედარების დროს საჭიროა ამ განსხვავებების გათვალისწინება.

თუ მარშრუტი შედგება მიმდევრობით შეერთებული რამოდენიმე არხისაგან, რომლებს გააჩნიათ სხვადასხვა ბიტების გადაცემის მახასიათებლები, მოცემული მარშრუტის მაქსიმალურ გამტარუნარიანობას წარმოადგენს ამ მარშრუტის გასწვრივ მყოფი ყველაზე მცირე არხი, ე.ი. არხი რომელსაც გააჩნია ყველაზე დაბალი ბიტების გადაცემის მახასიათებელი. ასეთ არხს ეწოდება „საცობი“. [44]

3.2.1 მაქსიმალური თეორიული გამტარუნარიანობა

ამ პარამეტრის მნიშვნელობა ძალიან უახლოვდება სისტემის არხის ტევადობას, და წარმოადგენს ინფორმაციის მაქსიმალურ რაოდენობას, რომლის გადაგზავნა შესაძლებელია იდეალურ პირობებში. ზოგ შემთხვევებში ეს მნიშვნელობა უტოლდება არხის ფიზიკურ ტევადობას, თუმცა აუცილებელია აღინიშნოს, რომ მხოლოდ ასინქრონულ ტექნოლოგიებში შესაძლებელია ინფორმაციის შეკუმშვის გარეშე ასეთი შედეგის მიღწევა. მაქსიმალური თეორიული გამტარუნარიანობის გასაზომად საჭიროა იყოს გათვალისწინებული ზედნადები ინფორმაციის ფორმატი და სპეციპიკა. ეს პარამეტრი და ასევე ქვემოთ განხილული მაქსიმალური მიღწევადი გამტარუნარიანობა ძირითადად წარმოადგენენ ქსელის დაგეგმარების ეტაპზე გასათვალისწინებელ მახასიათებლებს.

3.2.2 გამტარუნარიანობის უმაღლესი გაზომილი მნიშვნელობა

ზემოთ განხილული მნიშვნელობები წარმოადგენენ თეორიულ მნიშვნელობებს. თუმცა გამტარუნარიანობის უმაღლესი გაზომილი მნიშვნელობა წარმოადგენს იმ მნიშვნელობას, რომელიც არის გაზომილი რეალური ან სიმულირებული სისტემების მიერ. გამტარუნარიანობის მნიშვნელობა გამოითვლება დროის მოკლე მონაკვეთში. მათემატიკურად, ამას წარმოადგენს გამტარუნარიანობასთან მიმართებაში მყოფ ზღვარს, როდესაც დრო მიისწრაფის ნულისაკენ. ეს მნიშვნელობა

განსაკუთრებულად მნიშვნელოვანია იმ სისტემებისათვის, რომლებიც დამოკიდებულები არიან burst ტიპის ინფორმაციის გადაცემაზე.

3.2.3 მაქსიმალური მდგრადი გამტარუნარიანობა

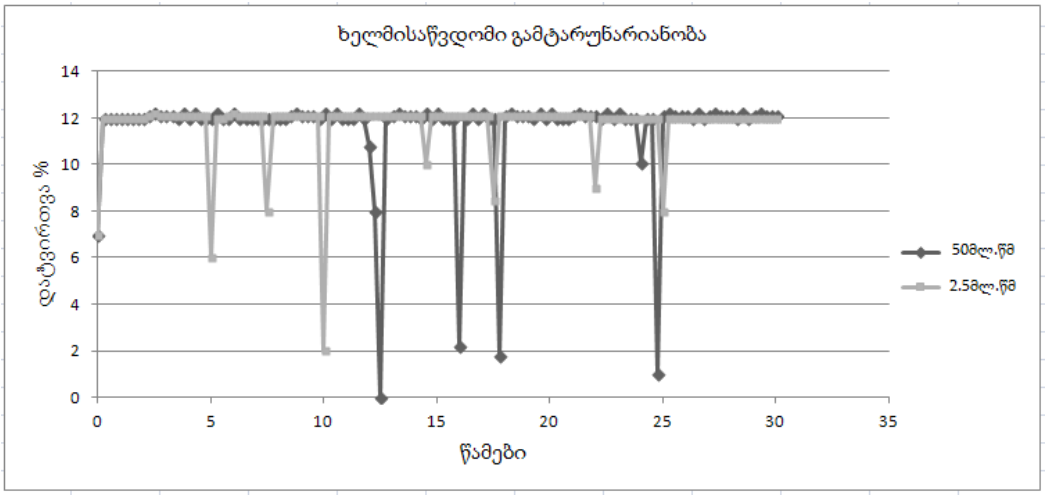
მაქსიმალური მდგრადი გამტარუნარიანობა წარმოადგენს მნიშვნელობას, რომელიც გამოითვლება დროის გრძელი მონაკვეთის პერიოდში და ის წარმოადგენს საშუალო სიდიდეს. ის აღწერილია როგორც ასიმპტოტური გამტარუნარიანობა, როდესაც დატვირთვა ძალიან დიდია. პაკეტების კომუტაციის სისტემებში, სადაც დატვირთვა და გამტარუნარიანობა ყოველთვის ტოლია, მაქსიმალური გამტარუნარიანობა შეიძლება იყოს გამოხატული, როგორც მინიმალური დატვირთულობა ბიტები/წმ. და რომელიც აიძულებს პაკეტების მიწოდების დაყოვნების არამდგრადობის არსებობას. ეს თავის მხრივ იწვევს დაყოვნების გაზრდას უსასრულობისაკენ.

3.2.4 მაქსიმალური მიღწევადი გამტარუნარიანობა

მაქსიმალური მიღწევადი გამტარუნარიანობა წარმოადგენს ქსელში განსაზღვრულ პირობებში ორ კვანძს შორის გადაცემული ინფორმაციის რაოდენობას. ამ პირობებში შედის გადაცემის კონტროლის პროტოკოლი, მიმღები ან გადამცემი კვანძის აპარატურული უზრუნველყოფა, ოპერაციული სისტემა, კონფიგურაციის სპეციფიკა და სხვა. ეს პარამეტრები განაპირობებენ იმ წარმადობას, რომლის მიღწევაც აპლიკაციას შეუძლია მოცემულ პირობებში. იმის გამო, რომ საცობი შესაძლებელია შეიქმნას ასევე ბოლო კვანძთან, მიღწევადი გამტარუნარიანობა შეიძლება შეესაბამებოდეს ან არ შეესაბამებოდეს ხელმისაწვდომ გამტარუნარიანობას. ხელმისაწვდომი გამტარუნარიანობა მკაცრად დამოკიდებულია გაზომვის დროის ინტერვალზე. თუმცა, ხელმისაწვდომი გამტარუნარიანობა არ წარმოადგენს იმის მაჩვენებელს თუ რას მიიღებს აპლიკაცია რეალურად. ამის

დასადგენად, უფრო ზუსტ გაზომვის პარამეტრს წარმოადგენს მიღწევადი გამტარუნარიანობა. [45]

მიღწევადი გამტარუნარიანობის განსაზღვრა მკაცრად დამოკიდებულია გაზომვის დროის ინტერვალზე. ნახაზი 3.1 ასახავს იმის მაგალითს, როდესაც ქსელის დატვირთვის მნიშვნელობა გაზომილია 50მლ.წმ. და 2,5მლ.წმ. დროის ინტერვალებში. მნიშვნელოვანია აღინიშნოს, რომ მიღწევადი გამტარუნარიანობის მნიშვნელობა არ წარმოადგენს იმის გარანტიას, თუ რას უშუალოდ მიიღებს აპლიკაცია. ამ პარამეტრის დასადგენად, მიღწევადი გამტარუნარიანობა წარმოადგენს უფრო ზუსტ გასაზომ პარამეტრს. [46]



ნახ. 3.1 ხელმისაწვდომი გამტარუნარიანობა ორ სხვადასხვა დროის ინტერვალში

3.3 გამტარუნარიანობის საზომი ხელსაწყოები

TTCP, Nuttcp, NetPerf და Iperf

TTCP, Nuttcp, NetPerf და Iperf წარმოადგენენ მიღწევადი გამტარუნარიანობის საზომ ხელსაწყოებს. კვანძთაშორის მიღწევადი გამტარუნარიანობის დასადგენად, ისინი გადასცემენ დიდი რაოდენობის ინფორმაციას TCP პროტოკოლის გამოყენებით. ამ მეთოდის შედეგად, შესაძლებელი ხდება მიმდინარე მარშრუტის მიღწევადი გამტარუნარიანობის გაზომვა. ეს ხელსაწყოები გამოიყენებიან ინტელექტუალური მარშრუტიზაციის პროტოკოლის გაზომვის ფაზაში.

მომხმარებელს შეუძლია მართოს სოკეტის ბუფერის ზომა და შესაბამისად ინფორმაციის გადაცემის მაქსიმალური TCP window–ს ზომა. პროტოკოლი TTCP [47] წარმოადგენს საკმაოთ ფართოდ გამოყენებად უტილიტს და მის ბაზაზე შექმნილია ისეთი საზომი ხელსაწყოები, როგორებიც არიან NetPerf და Iperf. ამ უკანასკნელებს შეუძლიათ მრავალი ერთდროული სესიის მართვა. გაზომვების ჩასატარებლად, სამივე უტილიტს ესაჭიროება საბოლოო ჰოსტების კონტროლი. [48, 49]

BWPing

კვანძთაშორის დაყოვნების და გამტარუნარიანობის გასაზომად ასევე გამოიყენება უტილიტი BWPing. ეს უტილიტი ამოცანის შესასრულებლად იყენებს ICMP პროტოკოლის ექო მოთხოვნების და პასუხების მეთოდს. ამ უტილიტის მუშაობის პრინციპი საკმაოდ მარტივია და ის ეფუძვნება ICMP პაკეტების დამუშავებას და არ მოიცავს განსაკუთრებულ მოთხოვნებს ოპერაციული სისტემისადმი. ამის მიუხედავად, მნიშვნელოვანია აღინიშნოს, რომ ამ უტილიტს გააჩნია ქსელისადმი მნიშვნელოვანი მოთხოვნა. ეს მოთხოვნა მდგომარეობს იმაში, რომ ორ კვანძს შორის მარშრუტის გასწვრივ არ უნდა იყოს ICMP პაკეტების ბლოკირება. ამის გარდა, ქსელის მომსახურების ხარისხის მექანიზმები არ უნდა ახდენდნენ ICMP პაკეტებზე რაიმე ზეგავლენას და ამ ტიპის პაკეტების გადაცემის სიჩკარე არ უნდა იზღუდებოდეს. ამ უტილიტის გაზომვის შედეგების სიზუსტე მეტად დამოკიდებულია ზემოთ ხსენებულ პირობებზე. [53]

იმისათვის, რომ გაზომვის შედეგები მაქსიმალურად აკურატულად ასახავდნენ რეალურ მდგომარეობას, საჭიროა ამ უტილიტის მიერ წარმოშობილი სატესტო პაკეტების მოხვედრა იგივე მომსახურების ხარისხის კლასში რაც რეალური აპლიკაციის პაკეტები. იმ შემთხვევებში, როდესაც ინტელექტუალური მარშრუტიზაციის პროტოკოლს ესაჭიროება მარშრუტის გამტარუნარიანობის გაზომვა და დანიშნულების კვანძის მართვა შეუძლებელია, გამოიყენება BWPing უტილიტი.

Pipechar და Pathchar

მარშრუტის გამტარუნარიანობის ზღვარის გასარკვევად ასევე ხელმისაწვდომია უტილიტები Pipechar და Pathchar [50]. მათი გაზომვის შედეგები მეტად მსგავსია, მაგრამ ამ შედეგების მისაღებად ისინი იყენებენ სხვადასხვა ალგორითმებს. საჭიროა აღინიშნოს, რომ ამ უტილიტების გამოყენების დანიშნულება განსხვავდება. Pathchar ახორციელებს მარშრუტის გამტარუნარიანობის და პაკეტების დანაკარგის ზუსტ შეფასებას. მეორეს მხრივ, Pipechar ადგენს მარშრუტის ყველაზე დაბალი გამტარუნარიანობის მქონე კვანძს და ამას ახორციელებს ინტერფეისის სიჩქარის ანალიზის მეშვეობით. მუშაობის ასეთი პრინციპის გამო, ყველაზე დაბალი გამტარუნარიანობის მქონე კვანძის შემდეგ მდებარე სეგმენტების შეფასება შეუძლებელი ხდება.

მნიშვნელოვანია აღინიშნოს, რომ თუ კი მარშრუტის გასწვრივ საცობის ადგილმდებარეობა დინამიკურია, მაშინ pipechar ჩაატარებს ყველაზე დაბალი გამტარუნარიანობის მქონე კვანძის მახასიათებლების შეფასებას და ამ ინფორმაციაზე დაყრდნობით მოხერხდება დანარჩენი არხების შეფასება. მაგალითად, თუ კი მარშრუტი შედგება 5 სეგმენტისაგან და მესამე სეგმენტი არის ყველაზე ნელი, მაშინ pipechar–ის მიერ ყველა სეგმენტის გაზომვის შედეგები იქნებიან დაყრდნობილი ამ ყველაზე ნელი სეგმენტის სიჩქარეზე. ასეთი მეთოდის გამოყენება მეტად აქტუალურია იმ აპლიკაციების ნაკადებისათვის, რომლებიც იყენებენ TCP პროტოკოლს იმისათვის რომ მოარგონ window-ს ზომა.

Pathload და PathChirp

ინტელექტუალური მარშრუტიზაციის გაზომვის ფაზის კიდევ ერთ ხელსაწყოს წარმოადგენს უტილიტი Pathload [51]. ეს უტილიტი, ასევე გამოიყენება მარშრუტის ხელმისაწვდომი გამტარუნარიანობის გაზომვისათვის და ამის მისაღწევად ის იყებეს SLOP ალგორითმს. მნიშვნელოვანია აღინიშნოს, რომ ამ უტილიტს ესაჭიროება დანიშნულების კვანძზე გამართვა და კონფიგურირება, მაგრამ ადმინისტრატორის

უფლებები არ არის საჭირო. ამ მოთხოვნის მიზეზს წარმოადგენს Pathload-ის მუშაობის პრინციპი, რომელიც დაყრდნობილია UDP პაკეტების გაგზავნაზე. ის ახორციელებს გაზომვებს და შედეგების გამოტანას დიაპაზონის სახით. ამ დიაპაზონის შუალედს წარმოადგენს გაზომილი მიღწევადი გამტარუნარიანობის შედეგების საშუალო მნიშვნელობას. მნიშვნელოვანია აღინიშნოს, რომ მოცემული დიაპაზონი წარმოადგენს გაზომვების შედეგების მიღებული გამტარუნარიანობის ვარიაციას.

მსგავსი ამოცანის შესასრულებლად ინტელექტუალური მარშრუტიზაცია იყენებს კიდევ ერთ ხელსაწყოს pathChirp [52]. მისი მუშაობის პრინციპი ოდნავ განსხვავდება სხვა დანარჩენი ხელსაწყოებისაგან და იყენებს მოდიფიცირებულ SLOP მექანიზმს. კერძოდ, ის იყენებს აქტიური სინჯების სხვადასხვა სახის ნიმუშებს, რომლებსაც ეწოდებათ chirp. ეს უტილიტი ზრდის სინჯის ზომას ყოველ chirp-ში, რომლის შედეგად შესაძლებელი ხდება მიღწევადი გამტარუნარიანობის დინამიკური გაზომვა.

Pathchirp-ის ძირითად უპირატესობას წარმოადგენს მისი გაზომვის შედეგების სიზუსტე და ამავე დროს გაზომვების განსახორციელებლად შედარებით ნაკლები დროის საჭიროება.

3.4 დაყოვნების საზომი ხელსაწყოები

<u>უტილიტი Ping</u>

უტილიტი Ping [54] წარმოადგენს ერთ ერთ ყველაზე გავრცელებულ საზომ ხელსაწყოს რომელიც უზრუნველყოფს როგორც დაყოვნების, ასევე პდვ და პაკეტების დანაკარგის გაზომვას და სტატისტიკას. Ping-ის სახელწოდება აღნიშნავს „Packet Internet Groper“, მისი მოქმედების ალგორითმი დაყრდნობილია ICMP პროტოკოლის Echo ფუნკციაზე და ჩამოყალიბებულია RFC 792 დოკუმენტში. ICMP [55] წარმოადგენს IP პროტოკოლის საკონტროლო პაკეტების სპეციალურ ტიპს და მისი

პროტოკოლის ნომერია 1.[57] ეს საკონტროლო პაკეტები გამოიყენებიან ორ კვანძს შორის ქსელური ინფორმაციის გასაცვლელად.

Ping-ის მოქმედების პრინციპი მდგომარეობას დანიშნულების მისამართამდე Echo მოთხოვნების გაგზავნაში და მისგან საპასუხო პაკეტების მოლოდინში. როდესაც დანიშნულების კვანძი ღებულობს ICMP Echo მოთხოვნის პაკეტს, ის პასუხობს ამ მოთხოვნას Echo პასუხის პაკეტით. მნიშვნელოვანია აღინიშნოს, რომ საპასუხო პაკეტის ინფორმაციის მატარებელ ველში იდება საწყისი მოთხოვნის პაკეტი და ისე ეგზავნება წყაროს კვანძს. ICMP echo მოთხოვნის პაკეტის სტრუქტურა წარმოდგენილია ნახ. 3.2.

00 01 02 03 04 05 06 07	08 09 10 11 12 13 14 15	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Type = 8	Code = 0	Header Checksum
Identifier		Sequence Number
Data :::		

ნახ.3.2 ICMP Echo მოთხოვნის და პასუხის პაკეტის სტრუქტურა

განვიხილოთ ICMP მოთხოვნის და პასუხის პაკეტების სხვადასხვა ველების მნიშვნელობები. პაკეტის ტიპის ველი განსაზღრავს იმას თუ რა სახის ICMP შეტყობინებასთან გვაქვს საქმე. კერძოდ, როდესაც ამ ველში ჩაწერილია 8 ეს იმას ნიშნავს, რომ პაკეტი არის echo მოთხოვნის ტიპის, და როდესაც ამ ველის მნიშვნელობა არის 0, პაკეტი წარმოადგენს echo პასუხის შეტყონიბენას. ეს ველი არის 8 ბიტის ზომის და მის შემდეგ მდებარეობს კოდის ველი. ეს ველი გამოიყენება მხოლოდ გარკვეულ შემთხვევებში და მისი მნიშვნელობა განსაზღვრავს პაკეტის შეტყობინების მოკლე აღწერას. შემდეგი ველი წარმოადგენს იდენტიფიკატორს და გამოიყენება echo მოთხოვნის უნიკალურობის უზრუნველსაყოფად. ეს აუცილებელია იმისათვის, რომ მოხერხდეს რამოდენიმე Ping სესიის ერთდროული წამოწყება და შესაბამისად მოხერხდეს მიღებული echo პასუხების დაკავშირება გაგზავნილ echo მოთხოვნებთად. ამასთან ერთად, გამოიყენება თანამიმდევრობის ველი, რომელიც ახორციელებს ყველა პაკეტის რიგითი ნომრით დანიშვნას. როგორც წესი, ამ ველის საწყისი მნიშვნელობა არის 1 და ყოველ მომდევნო პაკეტში იზრდება 1–ით.

ნახაზზე 3.3 წარმოდგენილია Ping უტილიტის მუშაობის შედეგები. ცდა ჩატარებულია ლაბორატორიის კომპიუტერზე, რომლის ოპერაციულ სისტემას წარმოადგენს Linux. ამ შედეგებში თვალნათლივ ჩანს, რომ ჰოსტი "lab" ახორციელებს ICMP echo მოთხოვნების წარმოშობას და მათ მიმართვას საქართველოს ტექნიკური უნივერსიტეტის ვებ სერვერისაკენ (www.gtu.ge). ამ პაკეტების წარმოშობის სიხშირე არის 1 პაკეტი/წამში.

```
user@lab:~$ ping www.gtu.ge
PING spider.gtu.ge (217.147.232.2) 56(84) bytes of data.
64 bytes from 217.147.232.2: icmp_req=1 ttl=44 time=74.2 ms
64 bytes from 217.147.232.2: icmp_req=2 ttl=44 time=76.6 ms
64 bytes from 217.147.232.2: icmp_req=3 ttl=44 time=74.8 ms
64 bytes from 217.147.232.2: icmp_req=4 ttl=44 time=73.7 ms
64 bytes from 217.147.232.2: icmp_req=5 ttl=44 time=76.1 ms
64 bytes from 217.147.232.2: icmp_req=6 ttl=44 time=73.3 ms
--- spider.gtu.ge ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 33422ms
rtt min/avg/max/mdev = 73.343/75.718/81.029/2.466 ms
```

ნახ. 3.3 Ping უტილიტის შედეგები, www.gtu.ge–საგან მიღებული echo პასუხები.

ყოველი მიღებული echo პასუხის პაკეტის შემდეგ, ping უტილიტს გამოაქვს ეკრანზე სტრიქონი დანიშნულების ჰოსტის IP მისამართით. მას თან შეიძლება ერთვას ამ ჰოსტის დომენის სახელი და ასევე წარმოდგენილია echo მოთხოვნის რიგითი ნომერი და TTL მნიშვნელობა. უკანასკნელი მიუთითებს ჰოპების მაქსიმალურ რაოდენობაზე, რომლის გავლა შეუძლია მოცემულ პაკეტს. სტრიქონის ბოლოში მოყვანილია ორი გზის დაყოვნების მნიშვნელობა, რომელიც წარმოადგენს ჩვენს შემთხვევაში ყველაზე მნიშვნელოვან მახასიათებელს. მიმდევრობითი ნომერი წარმოადგენს გაგზავნილი მოთხოვნების უნიკალურ იდენტიფიკატორს და შესაბამისად ახორციელებს ამ მოთხოვნებზე მიღებული პასუხების განსაზღვრას. საჭიროა აღინიშნოს, რომ როდესაც მიღებული პასუხების

მიმდევრობითი ნომრები არ არიან მიყოლებით, ეს იმას ნიშნავს, რომ გამოტოვებული ნომრის მქონე პაკეტი დაიკარგა. ასეთი დაკარგული პაკეტების რაოდენობის საფუძველზე იქმნება პაკეტების დანაკარგის ფაქტორი, რომელიც წარმოადგენს მომსახურების ხარისხის ფუნდამენტალურ მახასიათებელს. შეიძლება გამოიყოს ორი შემთხვევა: პირველი, როდესაც მოთხოვნის პაკეტმა ვერ მიაღწია დანიშნულების მისამართამდე, ან მეორე, როდესაც გამოგზავნილი პასუხი გზაში დაიკარგა და ვერ მიაღწია წყარომდე. ამის მიზეზი შეიძლება იყოს ფიზიკური ინტერფეისების გაუმართავი მდგომარეობა, კერძოდ ინტერფეისის შეცდომების ფაქტორი, ასევე გადატვირთული არხების არსებობა მარშრუტის გასწვრივ. როგორც წესი, TCP/IP პროტოკოლის გამოყენების შემთხვევაში, პაკეტების დანაკარგის 0.1% და ნაკლები ფაქტორი წარმოადგენს დასაშვებ მნიშვნელობას.

მნიშვნელოვანია აღინიშნოს, რომ თუ პაკეტების დანაკარგის ფაქტორი წარმოადგენს 0.1% დიდ მნიშვნელობას, ამან შეიძლება გამოიწვიოს ქსელში მომუშავე აპლიკაციების წარმადობის გაუარესება ან სულაც მუშაობის შეფერხება. გაზომვის შედეგები წარმოდგენილია სტატისტიკის სახით და მოიცავენ მინიმალურ, მაქსიმალურ და ორი გზის დაყოვნებას, ასევე პდვ–ს. [57]

უტილიტი Ping-ის მუშაობის პრინციპი მეტად ინტუიტიურია და შედეგები წარმოადგენენ ქსელის მდგომარეობის ნათელ სურათს. ამის გარდა, ამ უტილიტს აქვს სხვადასხვა პარამეტრების მითითების შესაძლებლობა, კერძოდ გაგზავნილი პაკეტების ზომა, გამოტანილი სტატისტიკის ფორმა, გასაგზავნი პაკეტების რაოდენობა და სხვა.

Ping უტილიტი წარმოადგენს ინტელექტუალური მარშრუტიზაციის გაზომვის ფაზის ხელსაწყოს და გამოიყენება იმ შემთხვევებში, როდესაც ICMP პროტოკოლი არის დაშვებული როგორც მარშრუტის გაყოლებაზე ასევე დანიშნულების კვანძზე. [58, 59]

უტილიტი traceroute

უტილიტი traceroute წარმოადგენს მარშრუტის ყველა კვანძის დადგენის ხელსაწყოს და ამოცანის შესასრულებლად იყენებს ICMP echo მოთხოვნის პაკეტებს. მისი მუშაობის პრინციპი მდგომარეობს იმაში, რომ traceroute იყენებს პაკეტების TTL ველებს და ყოველი პაკეტის სათაურში უთითებს მის გარკვეულ მნიშვნელობას. Linux ოპერაციულ სისტემებში ამ ველის საწყისი მნიშვნელობა 64–ის ტოლია. როგორც იყო გარჩეული პირველ თავში, როდესაც მარშრუტიზატორი ღებულობს IP პაკეტს, ის აკლებს ამ მნიშვნელობას 1 და აგზავნის შემდგომი მარშრუტიზატორის მისამართზე. იმ შემთხვევაში, თუ მარშრუტიზატორი მიიღებს პაკეტს, რომლის TTL მნიშვნელობა არის 1–ის ტოლი, ის მოახდენს პაკეტის გაუქმებას და პაკეტის წყაროს გაუგზავნის შეტყობინებას შეცდომის შესახებ „ICMP Time Exceeded“. ICMP პროტოკოლის ზუსტად ამ მექანიზმს იყენებს traceroute თავისი ამოცანის შესასრულებლად. კერძოდ, პირველი სინჯის TTL ველში ის უთითებს 1 და აგზავნის დანიშნულების მისამართზე. როდესაც ამ პაკეტს მიიღებს მარშრუტის გასწვრივ მდებარე პირველი მარშრუტიზატორი, ის შეამოწმებს TTL მნიშვნელობას და რადგან ის არის 1–ის ტოლი, გააუქმებს პაკეტს და დაუბრუნებს წყაროს კვანძს შეცდომის შეტყობინებას. შემდეგი სინჯის გაგზავნისას, traceroute მიუთითებს TTL ველში 2. ამჯერად, მარშრუტის პირველი მარშრუტიზატორი ამ მნიშვნელობას გამოაკლებს 1 და გააგზავნის შემდეგ. მეორე მარშრუტიზატორი პაკეტის მიღებისას შეამოწმებს TTL ველს და გააუქმებს პაკეტს რის შემდეგ დაუბრუნებს გამგზავნს შეცდომის შეტყობინებას. ეს პროცესი გრძელდება იქამდე, სანამ პაკეტები არ მიაღწევენ დანიშნულების კვანძს, რის შემთხვევაში დანიშნულების კვანძი მიიღებს ამ მოთხოვნის პაკეტს და წყაროს დაუბრუნებს echo პასუხის ტიპის პაკეტს. ასეთი პაკეტის მიღებისას, traceroute ადგენს, რომ მისი მოთხოვნის პაკეტი იყო მიწოდებული დანიშნულების კვანძამდე, ე.ი. მარშრუტის ბოლო კვანძი მიღწეულია. ამის შემდეგ, traceroute წყვეტს სინჯების წარმოშობას და

გამოაქვს შესაბამისი შეტყობინება ეკრანზე. მარშრუტის ყველა კვანძის აღმოჩენისას, traceroute აგზავნის ერთი და იგივე TTL მნიშვნელობის მქონე 3 სინჯს და ამ სინჯების საფუძველზე ის ადგენს სტატისტიკას. ამ სტატისტიკის ნაწილს წარმოადგენს მინიმალური, საშუალო და მაქსიმალური ორი გზის დაყოვნება. გაზომვის ამ შედეგებს ის ათავსებს ერთ სტრიქონზე და გამოაქვს ეკრანზე.

საჭიროა აღინიშნოს, რომ Unix-ის ტიპის ოპერაციულ სისტემებში, traceroute ICMP პროტოკოლის მაგივრად იყენებს UDP პროტოკოლის მაღალ პორტებს. მაღალი პორტების გამოყენებისას, ნაკლებად სავარაუდოა დანიშნულების კვანძზე იგივე პორტების გამოყენება რაიმე აპლიკაციის მიერ. ამიტომ, გამოიყენება პორტები შემდეგ დიაპაზონში: 33434 – 33534. როდესაც დანიშნიშნულების კვანძი ღებულობს UDP პაკეტს, რომლის TTL მნიშვნელობა 1–ის ტოლია, ის რასაკვირველია აუქმებს მას, მაგრამ ასევე ამოწმებს UDP სოკეტს, იმისათვის რომ შემოწმდეს თუ კი რომელიმე აპლიკაცია იყენებს ამ კონკრეტულ პორტს. ამ შემოწმების შემდეგ, დანიშნულების კვანძი წყაროს უბრუნებს ICMP შეცდომის შეტყობინებას “Port unreachable”. ასეთი შეტყობინების მიღება მიუთითებს traceroute-ს იმაზე, რომ პაკეტმა მიაღწია დანიშნულების მისამართს. როგორც წესი, traceroute უტილიტს აქვს სინჯების ტიპის არჩევის შესაძლებლობა, ე.ი. გამოიყენებოდეს ICMP ან UDP სინჯები. [60, 61]

Traceroute უტილიტი მიღებული ინფორმაციის საფუძველზე ადგენს სტატისტიკას და შედეგები გამოაქვს ეკრანზე. ეს სტატისტიკა შეიცავს საპასუხო პაკეტების მიღების დროს, ყველა სატრანზიტო ჰოპის IP მისამართს და დაყოვნების მნიშვნელობას გამოხატულს მილიწამებში. ნახ.3.4 წარმოდგენილია უტილიტი traceroute-ის სტატისტიკა, სადაც ხორციელდება საქართველოს ტექნიკური უნივერსიტეტის ვებ სერვერამდე მარშრუტის გამოკვლევა.

```
user@lab:~$ traceroute www.gtu.ge
traceroute to www.gtu.ge (217.147.232.2), 30 hops max, 60 byte packets
1  192.168.1.1 (192.168.1.1)  6.125 ms  21.723 ms  22.831 ms
2  * * *
```

3 ip-84-41-32-119.net.upcbroadband.cz (84.41.32.119) 23.980 ms 24.979 ms 25.684 ms
4 84.116.220.237 (84.116.220.237) 39.868 ms 43.673 ms 44.004 ms
5 84.116.221.37 (84.116.221.37) 45.360 ms 49.171 ms 53.141 ms
6 cz-prg02a-ra1-ge-1-0-0.0.aorta.net (213.46.172.26) 53.442 ms 213.46.172.222 (213.46.172.222)
31.931 ms *
7 cz-prg01a-ra1-ge-0-0-0-v50.aorta.net (213.46.172.18) 34.718 ms 40.163 ms 40.627 ms
8 prag-bb1-link.telia.net (213.155.131.62) 50.814 ms prag-bb1-link.telia.net (213.155.132.164)
51.288 ms prag-bb1-link.telia.net (213.155.131.64) 59.871 ms
9 sfia-b2-link.telia.net (80.91.247.83) 90.358 ms 90.664 ms sfia-b2-link.telia.net (80.91.253.249)
92.478 ms
10 ccsbulgaria-ic-153568-sfia-b2.c.telia.net (80.239.192.242) 124.232 ms 141.254 ms 141.622 ms
11 host-80-241-177-246.customer.co.ge (80.241.177.246) 137.021 ms 140.554 ms 76.819 ms
12 217.147.237.18 (217.147.237.18) 76.097 ms 75.658 ms 78.117 ms
13 217.147.237.173 (217.147.237.173) 88.551 ms 88.893 ms 89.904 ms
14 217.147.237.65 (217.147.237.65) 80.538 ms 80.842 ms 90.236 ms
15 217.147.237.102 (217.147.237.102) 90.531 ms 90.828 ms 91.500 ms
16 217.147.237.82 (217.147.237.82) 89.137 ms 89.482 ms 91.146 ms
17 217.147.232.130 (217.147.232.130) 92.778 ms 75.081 ms 73.941 ms
18 217.147.232.2 (217.147.232.2) 74.817 ms 72.805 ms 73.706 ms

ნახ. 3.4 traceroute უტილიტის მარშრუტის კვლევის შედეგი, სადაც დანიშნულების მისამართს წარმოადგენს სტუ ვებ სერვერი.

მნიშვნელოვანია აღინიშნოს, რომ სტატისტიკის მესამე ჰოპის სტრიქონში ფიფქების არსებობა მიუთითებს იმაზე, რომ წყაროს ჰოსტმა ვერ მიიღო საპასუხო პაკეტები გარკვეული დროის მანძილზე, რის შემდეგად გამოსატანი ინფორმაციის მაგივრად ჩნდება ფიფქების სიმბოლოები. ამ მაღალი დაყოვნების მიზეზი შეიძლება იყოს ის, რომ ICMP პაკეტების დამუშავება ხორციელდება მარშრუტიზატორის პროცესორის მიერ, რომელიც შეიძლება იყოს დატვირთული სხვა ამოცანების შესრულებით. ამის გამო შესაძლებელია პაკეტების დაგვიანება. სხვა შესაძლო მიზეზს შეიძლება წარმოადგენდეს ICMP პაკეტების ბლოკირება უსაფრთხოების წესის შესაბამისად. იმის მიუხედავად, რომ რომელიმე ჰოპიდან ვერ მოხერხდა სტატისტიკის მიღება, traceroute უტილიტი აგრძელებს მუშაობას

და ზრდის TTL ველის მნიშვნელობას, რის შედეგად სინჯები მიემართებიან შემდეგ მარშრუტიზატორისაკენ. [62]

<u>უტილიტი Pathchar</u>

უტილიტი Pathchar წარმოადგენს ინტელექტუალური მარშრუტიზაციის კიდევ ერთ კომპონენტს. როგორც წინამორბედი უტილიტები ასევე Pathchar–ის ძირითად ამოცანას წარმოადგენს დაყოვნების გაზომვა. ამის გარდა მას შეუძლია არხების სხვა მახასიათებლების დადგენაც, მაგალითად გამტარუნარიანობის. ამის განსახორციელებლად ის იყენებს ორი გზის დაყოვნების გაზომვის ალგორითმს და მიღებული ინფორმაციის საფუძველზე ადგენს სტატისტიკას. Pathchar ალგორითმის თავისებურებას წარმოადგენს გაგზავნილი პაკეტების მიმდევრობის ანალიზი, და ამ მიმდევრობების შედარებით ხორციელდება საბოლოო სტატისტიკის შედგენა.

ისევე როგორც უტილიტები Ping და traceroute, Pathchar იყენებს TTL ველს იმისათვის, რომ დაადგინოს მარშრუტის გასწვრივ მდებარე მარშრუტიზატორების მისამართები და დაყოვნების მაჩვენებლები. გაგზავნილი სინჯების ანალიზის საფუძველზე, ეს უტილიტი ზომავს ორი გზის დაყოვნების მნიშვნელობას ყოველივე ჰოპამდე, რომელიც უდრის დროის სხვაობას გაგზავნილი მოთხოვნასა და მიღებულ პასუხს შორის. მიღებული შედეგების ანალიზის საფუძველზე, Pathchar ახორციელებს სტატისტიკის შედგენას, რომელიც მოიცავს გამტარუნარიანობას, დაყოვნებას და ბუფერიზაციის დაყოვნების მნიშვნელობებს. პაკეტის ზომის გაზრდის შემთხვევაში იზრდება ორი გზის დაყოვნების მნიშვნელობაც და ზუსტად ამ მახასიათებელს იყენებს ეს უტილიტი ბუფერიზაციის დაყოვნების დასადგენად.

Pathchar მეტად ზუსტად ახორციელებს მარშრუტის გამტარუნარიანობის, დაყოვნების და პაკეტების დანაკარგის გაზომვას და გამოიყენება ინტელექტუალური მარშრუტიზაციის გაზომვის ფაზაში.

უტილიტი HTTPing

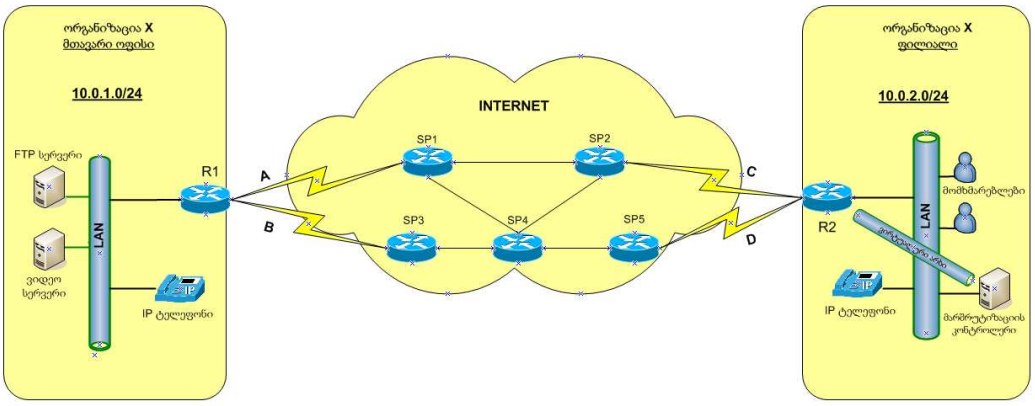
ზემოთ განხილული უტილიტების გამოყენების დანიშნულებაა ქსელში ორ კვანძს შორის პაკეტების დაყოვნების გაზომვა. საჭიროა აღინიშნოს, რომ ICMP პროტოკოლის ბლოკირება გამოიწვევს მათი მუშაობის შეფერხებას, ასევე კვანძთაშორის დაყოვნების მახასიათებელი შეიძლება განსხვავდებოდეს იმისაგან, თუ რა დაყოვნებას განიცდის მომხმარებლის მოთხოვნის პაკეტი. ამის მიზეზი შეიძლება იყოს სერვერის დატვირთულობა, ან აპლიკაციის სპეციფიკა. ამ მახასიათებლის უფრო ზუსტად გასაზომად ინტელექტუალური მარშრუტიზაცია იყენებს უტილიტს HTTPing, რომელიც ამყარებს სესიას სასურველ სერვერთან და ამოწმებს სერვერის პასუხის დაყოვნებას. ამის შედეგად მიღებული სტატისტიკა მეტად დაახლოვებულია იმასთან თუ რა დაყოვნებას განიცდის მომხმარებლის მოთხოვნის პაკეტი.

3.5 ინტელექტუალური მარშრუტიზაციის შედეგების ანალიზი

მოცემულ ქვეთავში წარმოდგენილია ინტელექტუალური მარშრუტიზაციის ალგორითმის მუშაობის შედეგები და მათი ანალიზი. ალგორითმის აპრობაცია განხორციელებულია ლაბორატორიულ პირობებში Linux და Cisco IOS ოპერაციულ სისტემებზე.

აგებულია ქსელის მოდელი, რომელიც გამოხატულია ნახაზზე 3.5. ნახაზის მარცხენა ნაწილში წარმოდგენილია ორგანიზაცია X–ის სათაო ოფისი, რომელში მდებარეობენ ორგანიზაციის სერვერები. ამ ოფისს გააჩნია ერთი კარიბჭე R1, რომელსაც აქვს ინტერნეტ პროვაიდერამდე ორი კომუნიკაციის არხი, A და B. ნახაზის მარჯვენა ნაწილში წარმოდგენილია ამავე ორგანიზაციის ფილიალი, რომელშიც მდებარეობენ მხოლოდ მომხმარებლები. ისინი უკავშირდებიან მთავარ ოფისში განლაგებულ სერვერებს. ფილიალს გააჩნია ერთი კარიბჭე R2 და ასევე ინტერნეტ პროვაიდერთან ორი შეერთების არხი C და D.

მნიშვნელოვანია აღინიშნოს, რომ წარმოდგენილი ორგანიზაციის ქსელში მუშაობს რამდენიმე სხვადასხვა ტიპის აპლიკაცია და მათი ქსელის მიმართ მოთხოვნები განსხვავებულია. მაგალითად, ნახაზზე ნაჩვენებია,



ნახ.3.5 ექსპერიმენტალური ქსელის ტოპოლოგია

რომ მთავარ ოფისში მდებარეობს FTP სერვერი, რომელიც უზრუნველყოფს ფაილების შენახვის და გადაცემის მომსახურებას. ასეთი ტიპის აპლიკაციის ნაკადს ესაჭიროება მაღალი გამტარუნარიანობა, მაგრამ დაყოვნებისადმი მოთხოვნა იმდენად მკაცრი არ არის. ამის გარდა, მთავარ ოფისში არსებობს IP ტელეფონიის და ვიდეო სთრიმინგის სერვერები, რომლებსაც ესაჭიროებათ დაბალი დაყოვნება და დაბალი დაყოვნების ვარიაცია. ინტელექტუალური მარშრუტიზაციის მიერ კონტროლირებად მომსახურების ხარისხის კრიტერიუმებს შეიძლება წარმოადგენდეს:

- დაყოვნება
- საიმედოობა (პაკეტების დანაკარგის ფაქტორი)
- დაყოვნების ვარიაცია
- გამტარუნარიანობა
- სხვა კრიტერიუმების დამატების შესაძლებლობა

ორ ოფის შორის არსებული 5 მარშრუტიზატორი წარმოადგენენ ინტერნეტის იმიტაციას, სადაც კავშირს უზრუნველყოფს 5 სხვადასხვა პროვაიდერი, ე.ი. მარშრუტიზატორები SP1 – SP5.

ალგორითმის მუშაობის შესამოწმებლად მოცემულ ტოპოლოგიაში ჩატარებულია სამი ცდა. პირველი ცდა ხორციელდება პაკეტთა დაყოვნების

კონტროლის მიმართულებით, მეორე ცდა ამოწმებს დაყოვნების ვარიაციის პარამეტრის კონტროლს და მესამე ცდა ხორციელდება პაკეტების დანაკარგის მიმართულებით. მნიშვნელოვანია აღინიშნოს, რომ ალგორითმის შემოწმების მექანიზმი არ არის შეზღუდული ამ სამი ტიპის მახასიათებლით და შესაძლებელია სხვა პარამეტრების კონტროლი. ალგორითმის მუშაობის დემონსტრირებისათვის, ჩვენ გავარჩევთ მხოლოდ სამ შემთხვევას. დანარჩენი პარამეტრების კონტროლი ხორციელდება იგივე ლოგიკით.

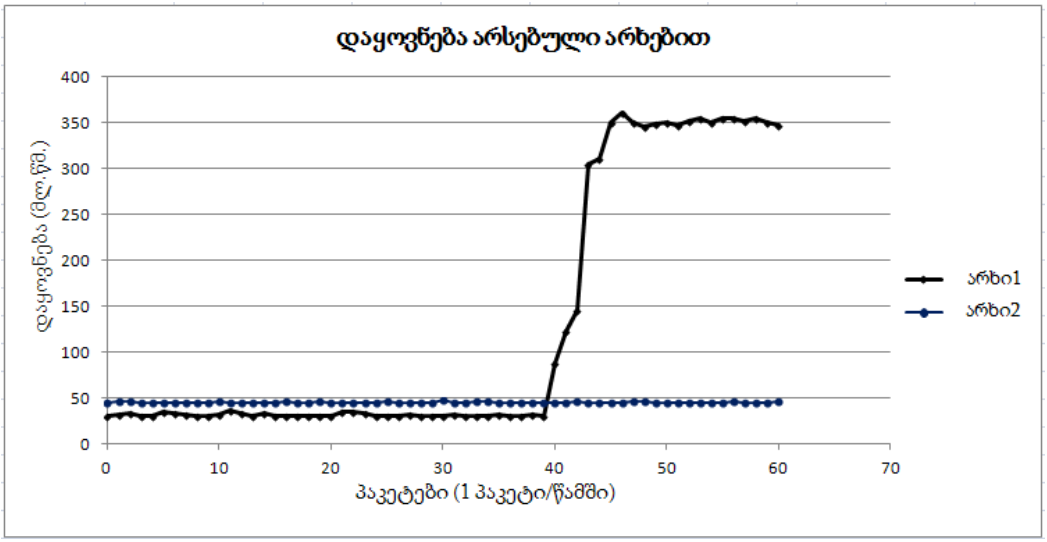
3.5.1 ცდა 1 – დაყოვნების კონტროლი

პირველი ცდის მიზანია შემოწმდეს ინტელექტუალური მარშრუტიზაციის დაყოვნების კონტროლის მექანიზმი და გაანალიზდეს მიღებული შედეგები. ნახ.3.5-ზე მოცემული ტოპოლოგიის ბაზაზე მარშრუტიზატორი R2 ახორციელებს მთავარ ოფისში მდებარე ვიდეო სერვერამდე დაყოვნების გაზომვას ყველა არსებული არხის გამოყენებით, ე.ი. არხებით C და D.

დაყოვნების გაზომვისათვის, მარშრუტიზაციის კონტროლერი იყენებს Ping უტილიტს, რომლის მუშაობის პრინციპი განხილულია ქვეთავში 3.4. მარშრუტიზაციის კონტროლერი ახორციელებს Ping უტილიტის ორი სესიის წამოწყებას და პაკეტების გადაგზავნას მარშრუტიზატორი R2-მდე. ამისათვის გამოიყენება ვირტუალური არხი, რომელიც შექმნილია მარშრუტიზაციის კონტროლერსა და მარშრუტიზატორი R2–ს შორის. ICMP პაკეტების მიღებისას, R2 ახორციელებს მათ შემდგომ გადაგზავნას ორი სხვადასხვა არხის გამოყენებით. ამ პაკეტების მიღებისას, ვიდეო სერვერი პასუხობს ICMP echo პასუხის პაკეტებით. როდესაც ეს პაკეტები აღწევენ მარშრუტიზაციის კონტროლერს, მიღებული პაკეტების საფუძველზე ის ახორციელებს დაყოვნების და დაყოვნების ვარიაციის გაზომვას.

ცდა 1–ის ფარგლებში, დაყოვნების კონტროლის წესში მითითებულია დასაშვები დაყოვნების ზღვარი, რომელიც 150მლ.წმ. ტოლია, ე.ი. თუ კი მარშრუტიზაციის მიერ არჩეული მარშრუტის დაყოვნება გადააჭარბებს ამ ზღვარს, პროტოკოლმა უნდა მოახდინოს ალტერნატიული მარშრუტის არჩევა, რომლის დაყოვნება უნდა შეადგენდეს 150მწ.–ზე ნაკლებ მნიშვნელობას.

მოცემული ცდა ჩატარდა 1 წუთის განმავლობაში, ე.ი. 60 წამი და პაკეტების გაგზავნის სიხშირე შეადგენდა 1 პაკეტი/წამში. ცდის შედეგად მიღებულია 60 ICMP echo პასუხი და მათი დაყოვნებების საფუძველზე აგებულია გრაფიკი, რომელიც წარმოდგენილია ნახ. 3.6.



ნახ.3.6 მარშრუტიზაციის კონტროლერიდან სერვერამდე არსებული არხებით გაზომილი დაყოვნება

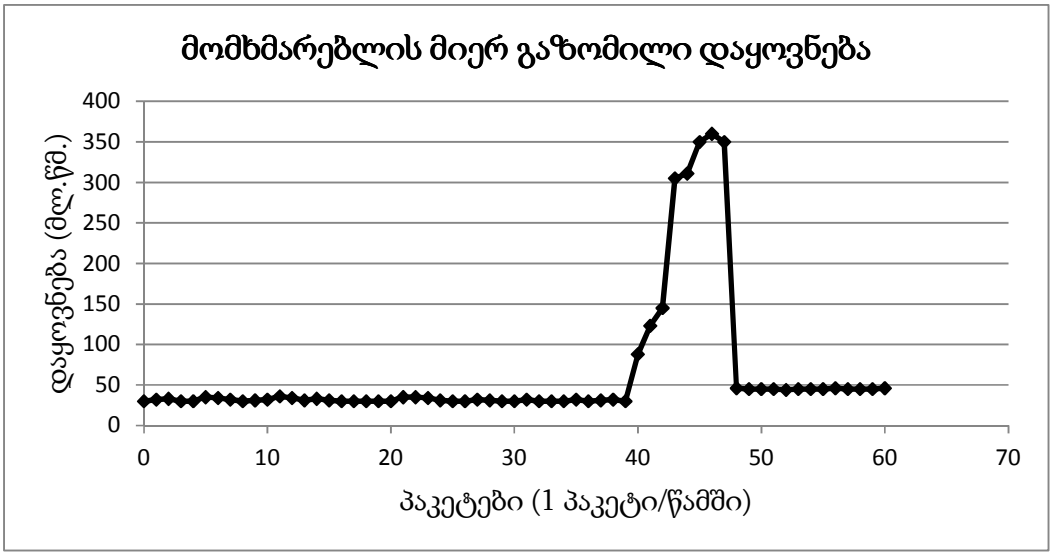
ამ გრაფიკზე წარმოდგენილია მარშრუტიზაციის კონტროლერის მიერ გაზომილი დაყოვნებები ორი არსებული არხით. საწყისს მდგომარეობაში, მარშრუტიზაციის მიერ იყო არჩეული არხი 1 (C), რადგან მისი გამოყენებით სერვერამდე დაყოვნება მერყეობს 30–35მლ.წმ. ფარგლებში. ეს მნიშვნელობა შეიძლება ჩაითვალოს საუკეთესოდ ორ არხს შორის, რადგან არხი 2 (D) გამოყენებით იგივე სერვერამდე დაყოვნება შეადგენს ~45მლ.წმ.

პრობლემის იმიტირების მიზნით, ინტერნეტ პროვადერის ISP-2 მარშრუტიზატორის ინტერფეისზე შეცვლილია პარამეტრები, რის შედეგად C არხის მარშრუტის დაყოვნება მკვეთრად გაიზარდა და მერყეობს ~350მლ.წმ. ფარგლებში. გრაფიკზე ეს მოვლენა ჩანს 39–ე პაკეტის შემდეგ, დაყოვნება იზრდება 145მწ.–მდე, შემდეგ 300–მდე და სტაბილურდება ~345–355მლ.წმ. დიაპაზონში.

ვინაიდან, ცდა 1–ის დაყოვნების კონტროლის წესში იყო მითითებული დაყოვნების ზედა ზღვარის მნიშვნელობა ტოლი 150მლ.წმ., ინტელექტუალური მარშრუტიზაციის პროტოკოლმა უნდა განახორციელოს უფრო ოპტიმალურ მარშრუტზე გადართვა. მოცემულ შემთხვევაში, არხი 2–ის დაყოვნება სტაბილურია და შეადგენს 45მლ.წმ. რაც აკმაყოფილებს მითითებული წესის პირობას.

იმისათვის, რომ თვალნათლად დავინახოთ ინტელექტუალური მარშრუტიზაციის ალგორითმის მუშაობის ზეგავლენა მომხმარებლის ნაკადებზე, საჭიროა მთელი ცდის განმავლობაში მომხმარებლის კომპიუტერიდან დაყოვნების მონიტორინგის წარმოება. ამ მონიტორინგის შედეგები წარმოდგენილია ნახ.3.7 რასაც მოყვება შედეგების განმარტება და ანალიზი.

როგორც ვხედავთ, მომხმარებლის კომპიუტერიდან იგივე დანიშნულების მისამართამდე გაზომილი დაყოვნების შედეგები გვიჩვენებენ, რომ ვიდეო სერვერამდე დაყოვნება მერყეობდა 30–35მლ.წმ. დიაპაზონში. ეს განპირობებულია იმით, რომ სერვერის მისამართამდე საუკეთესო მარშრუტად იყო არჩეული არხი 1, რომელიც მიმდინარე პირობებში ყველაზე დაბალი დაყოვნების მქონე არხია. ინტერნეტში პრობლემის იმიტაციის შემდეგ (39–ე წამიდან) მომხმარებლის პაკეტების დაყოვნება მატულობს. ნახ.3.7 წარმოდგენილი გრაფიკი შეიძლება დაიყოს 4 ძირითად ნაწილად.



ნახ.3.7 მომხმარებლის კომპიუტერიდან სერვერამდე გაზომილი დაყოვნება

პირველი ნაწილი მოიცავს 1–დან 39–მდე პაკეტებს, ამ ეტაპზე ქსელი სტაბილურია და საუკეთესო მარშრუტად არჩეულია არხი 1.

მეორე ეტაპი მოიცავს სამ პაკეტს, 40–დან – 42–მდე. ამ ეტაპზე ინტერნეტში პრობლემის გამო დაიწყო პაკეტების დაყოვნების ზრდა, მაგრამ ამ დაყოვნებების მნიშვნელობა ჯერ კიდევ დასაშვების დიაპაზონშია, ე.ი. 150მლ.წმ.–ზე ნაკლები.

მესამე ეტაპი მოიცავს 5 პაკეტს, 43–დან 47–მდე და ამ პაკეტებით გაზომილი დაყოვნება შეადგენს ~300–350მლ.წმ. რაც ცდება წესით განსაზღვრულ ზღვარს. მნიშვნელოვანია აღინიშნოს, რომ ზუსტად ამ 5 პაკეტის საფუძველზე, ინტელექტუალური მარშრუტიზაციის ალგორითმი ადგენს კონკრეტული მარშრუტის პრობლემას და ზღვარს აცილებული 5 პაკეტის მიღების შემდეგ ახორციელებს მარშრუტის გადართვას არხი 2–საკენ. ალგორითმი იღებს ბოლო 5 მიღებული პაკეტის საშუალო დაყოვნებას და ადარებს მას ზღვარის მნიშვნელობას. ამ მოქმედების განსახორციელებლად აუცილებელი სინჯების რაოდენობა შეიძლება შეირჩეს ადმინისტრატორის სურვილისამებრ, მაგრამ ის არ უნდა იყოს ძალინ მცირე, რომ არ მოხდეს მარშრუტის ფლუქტუაცია. ასევე, მისი მნიშვნელობა არ უნდა იყოს ძალიან დიდი, რომ თავიდან იყოს არიდებული მომხმარებლის ნაკადების გადაცემის ხარისხის გაუარესება.

მეოთხე ეტაპი მოიცავს დანარჩენ 12 პაკეტს, ე.ი. დაწყებული 48–ე პაკეტიდან ცდის ბოლო მე–60 პაკეტამდე. ამ ეტაპზე განხორციელდა მარშრუტის გადართვა, რის შემდეგ დაყოვნება შემცირდა 45მლ.წმ–მდე და ეს მნიშვნელობა წარმოადგენს არხი 2 მარშრუტის დაყოვნებას. თუ კი ჩვენ შევადარებთ ნახ.3.6 და ნახ.3.7, თვალნათლად ჩანს, რომ გადართვის შემდეგ მომხმარებლის პაკეტების დაყოვნება გაუტოლდა არხი 2 გაზომილ დაყოვნებას.

მნიშვნელოვანია აღინიშნოს, რომ მიმდინარე არხის დაყოვნების ზრდისას და წესით მითითებული წღვარის გადაჭარბებისას, პროტოკოლი ახორციელებს ალტერნატიულ მარშრუტზე გადართვას მხოლოდ იმ შემთხვევაში, როდესაც ალტერნატიული მარშრუტის დაყოვნება წესით განსაზღვრულ დიაპაზონშია. მოცემულ ცდაში, წესით განსაზღვრული დაყოვნების ზღვარს შეადგენს 150მლ.წმ.–ზე მცირე მნიშვნელობა.

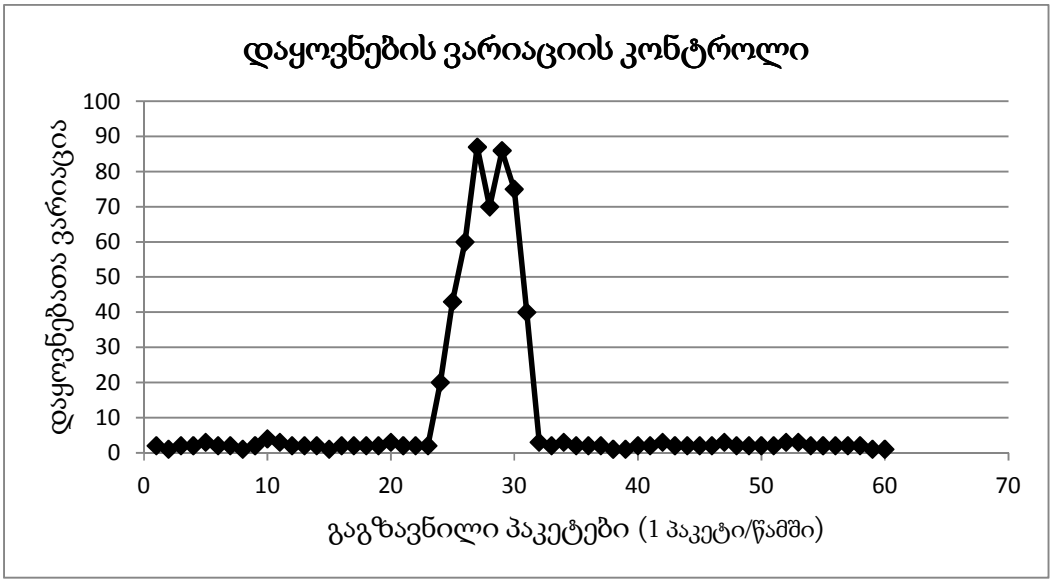
3.5.2 ცდა 2 – დაყოვნების ვარიაციის კონტროლი

წინა ქვეთავში განხილული ალგორითმის მსგავსად, ინტელექტუალურ მარშრუტიზაციას შეუძლია აკონტროლოს დაყოვნების ვარიაციის პარამეტრი, რომელიც წარმოადგენს ორ თანამიმდევრულად მიღებული პაკეტებს შორის დაყოვნების სხვაობას. ამ პარამეტრის უფრო დაწვრილებითი განმარტება განხილულია ქვეთავში 3.4.

საჭიროა აღინიშნოს, რომ თუ კი წინა ქვეთავში განხილული ცდა 1–ის ფარგლებში წესში ასევე იქნებოდა მითითებული დაყოვნების ვარიაციის პარამეტრი და მას უფრო მაღალი პრიორიტეტი ექნებოდა ვიდრე პაკეტების დაყოვნების პარამეტრს, მაშინ საწყის მდგომარეობაში საუკეთესო მარშრუტად არჩეული იქნებოდა არხი 2. ამის მიზეზი არის არხი 1–ის დაყოვნების ვარიაციის შედარებით მაღალი მაჩვენებელი ~5მლ.წმ.

ცდა 2 მიზანია განხორციელდეს დაყოვნების ვარიაციის პარამეტრის მონიტორინგი და კონტროლი. ცდა ჩატარებულია იგივე მეთოდით რაც წინა ქვეთავში და მიღებული შედეგები წარმოდგენილია ნახ. 3.8.

მაღალი დაყოვნების ვარიაციის მახასიათებლის ძირითად მიზეზს წარმოადგენს გადატვირთული არხები ან ინტერფეისების გადავსებული ბუფერები. ეს მახასიათებელი დიდ ზეგავლენას ახდენს ხმის გადამცემი და ინტერაქტიული აპლიკაციების ხარისხზე და მისი კონტროლი დაყოვნების კონტროლთან შეთავსებით წარმოადგენს ინტერაქტიული აპლიკაციების მაღალი ხარისხის უზრუნველყოფის ეფექტურ გადაწყვეტილებას.



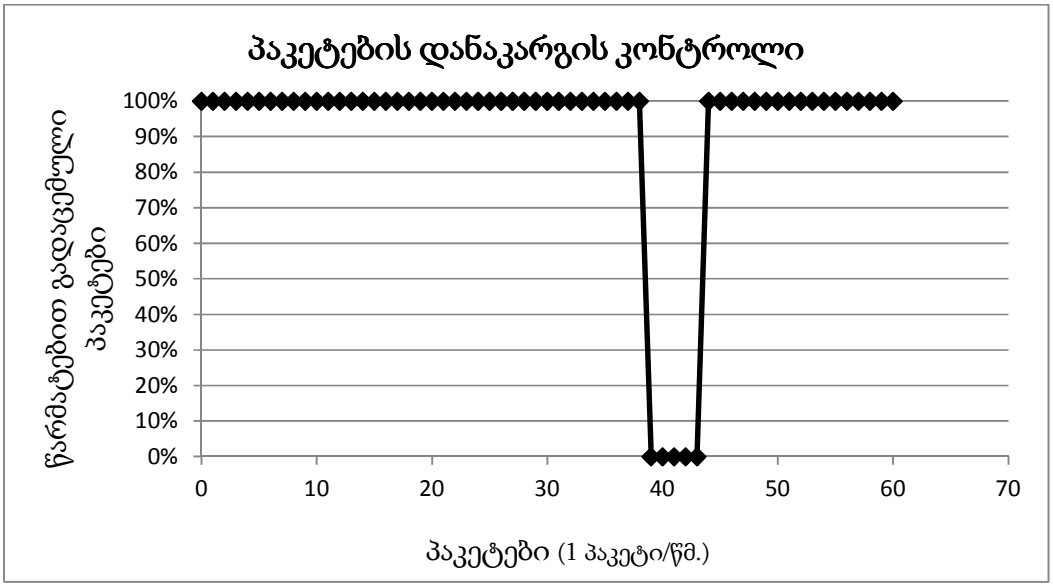
ნახ.3.8 მომხმარებლის კომპიუტერიდან სერვერამდე გაზომილი დაყოვნების ვარიაცია

3.5.3 ცდა 3 – პაკეტების დანაკარგის კონტროლი

ცდა 3 ფარგლებში იყო შემოწმებული ინტელექტუალური მარშრუტიზაციის მიერ პაკეტების დანაკარგის მახასიათებლის კონტროლის მექანიზმი. მარშრუტიზაციის კონტროლერზე გამოიყენება უტილიტი Ping, რის მეშვეობით ხორციელდება დანიშნულების მისამართამდე სინჯების გაგზავნა 1 პაკეტი/წამში სიხშირით. პაკეტების დანაკარგის წესით მითითებული ზღვარი წარმოადგენს 5 დაკარგულ პაკეტს.

ჩატარებული ცდა 3–ის შედეგებზე აგებულია გრაფიკი, რომელიც დაფუძნებულია მომხმარებლის კომპიუტერიდან გაზომვის შედეგად მიღებულ მნიშვნელობებზე. ეს ფრაფიკი წარმოდგენილია ნახ. 3.9.

ნახ. 3.5 ტოპოლოგიის საფუძველზე საუკეთესო მარშრუტად იყო არჩეული არხი C. ამ არხის გამოყენებით პაკეტების დანაკარგი უდრის 0, ე.ი. ყველა გაგზავნილ სინჯზე იყო მიღებული საპასუხო პაკეტი. 39 გაგზავნილი პაკეტის შემდეგ ხორციელდება პრობლემის იმიტაცია, რის შედეგად იკარგება 5 პაკეტი (პაკეტები 39 – 43). ხუთი დაკარგული პაკეტის საფუძველზე, ინტელექტუალური მარშრუტიზაციის ალგორითმი იღებს გადაწყვეტილებას გადართოს არსებული დანიშნულების მისამართის მარშრუტი ალტერნატიულ არხზე, ე.ი. არხზე D. ამის შედეგად 44–ე სინჯიდან დაწყებული, პაკეტების დანაკარგის მახასიათებელი ისევ 0 ტოლი ხდება, ე.ი. პასუხების მიღების მაჩვენებელი 100% ტოლია. ამას მოყვება დანიშნულების მისამართამდე კავშირის აღდგენა.



ნახ.3.9 მომხმარებლის კომპიუტერიდან სერვერამდე გაზომილი პაკეტების დანაკარგი

ზემოთხსენებული სამი ცდის შედეგად ჩატარებულია ინტელექტუალური მარშრუტიზაციის აპრობაცია და მიღებული შედეგების

საფუძველზე დასაბუთებულია შემუშავებული ალგორითმის მაღალი ეფექტურობა და თანამედროვე ქსელებში დანერგვის აუცილებლობა.

III თავის დასკვნა

1. განხილულია ინტელექტუალური მარშრუტიზაციის ალგორითმის შემადგენელი ნაწილების პროტოკოლში ინტეგრირების საკითხები. ასევე გაანალიზებულია მათი მუშაობის პრინციპები. დასაბუთებულია ქსელის მომსახურების ხარისხის მახასიათებლების საზომი ხელსაწყოების შესაძლებლობები და მათი გამოყენების ასპექტები გაზომვის ფაზაში.
2. ჩატარებულია ინტელექტუალური მარშრუტიზაციის ალგორითმის აპრობაცია ქსელის სიმულატორის ბაზაზე. განხილულია მომსახურების ხარისხის მახასიათებლების კონტროლის ალგორითმების მუშაობის შედეგები. წარმოდგენილია მათი გრაფიკული გამოხატულება.
3. შემუშავებული ალგორითმი არის დაფუძნებული სტანდარტულ კომპონენტებზე და ღია კოდის უტილიტებზე. ეს იძლევა მოცემული სისტემის ნებისმიერი მწარმოებლის მოწყობილობებზე დანერგვის უპირატესობას.

დასკვნები

სადისერტაციო ნაშრომში მიღებული შედეგების მიმართ შეიძლება გაკეთდეს შემდეგი დასკვნები:

1. თანამედროვე გლობალური ქსელების ამოცანებისა და პრობლემების გადაწყვეტის მიზნით გამოყენებული ალგორითმებისადმი მიძღვნილი ლიტერატურის მიმოხილვა და ანალიზი გვიჩვენებს, რომ მარშრუტიზაციის არსებული პროტოკოლები ირჩევენ ერთ საუკეთესო მარშრუტს ნაკადის ადრესატამდე და ამ მიმართულებით აგზავნიან ყველა ტიპის აპლიკაციების პაკეტებს. შედეგად ვიღებთ იმას, რომ მარშრუტიზაციის პროტოკოლის მიერ არჩეული საუკეთესო მარშრუტი არ არის ერთნაირად საუკეთესო ყველა აპლიკაციისათვის და ამით ზოგიერთი აპლიკაციის კავშირის ხარისხი უარესდება.
2. ჩატარებულია კლასიკური მარშრუტიზაციის მეთოდების ანალიზი. ნაჩვენებია, რომ ეს პრობლემა წარმოადგენს ერთერთ მთავარ პრობლემას ქსელში აპლიკაციების მუშაობის მაღალი ხარისხის უზრუნველყოფისას.
3. გაანალიზებულია არსებული მაღალი წარმადობის მარშრუტიზაციის მოდელი და ალგორითმი. ჩამოყალიბებულია ამ მოდელის მუშაობის პრინციპი. გაანალიზებულია მისი ნაკლოვანობები, მათ შორის ყველაზე მნიშვნელოვანია მისი ურთიერთდამოკიდებულება მწარმოებლის მოწყობილობებისადმი. დასაბუთებულია ამ პრობლემის მოგვარების გზა, რომელიც მდგომარეობს მწარმოებლისაგან დამოუკიდებელი მეთოდის შემუშავებაში.
4. შემუშავებულია ინტელექტუალური მარშრუტიზაციის მოდელი, რომელიც ითვალისწინებს ქსელის მომსახურების ხარისხის მდგომარეობას და სხვადასხვა კრიტიკული მახასიათებლების გაზომვას.

5. შემუშავებულია ალგორითმი, რომელსაც საფუძვლად უდევს ძირითადი მოთხოვნა, რომ მარშრუტიზატორებს უნდა შეეძლოთ ნაკადების მარშრუტიზაცია არა მარტო დანიშნულების პრეფიქსების მიხედვით, არამედ აპლიკაციების ტიპებზე დაყრდნობით. მარშრუტიზაცია ირჩევს აპლიკაციისათვის საუკეთესო მარშრუტს და არა მხოლოდ დანიშნულების პრეფიქსისათვის. შემუშავებული მეთოდის შედეგად, ქსელში უეცარი ცვლილებების შემთხვევაში, მარშრუტიზატორები ახდენენ სწრაფ რეაგირებას და ხორციელდება ნაკადების გადანაწილება სასურველი მიმართულებებით.
6. შემუშავებული არლგორითმი იყენებს ტრაფიკის მართვის ინტელექტუალურ მეთოდს, რომელსაც შეუძლია დინამიკური მარშრუტიზაციის გადაწყვეტილებების მიღება ქსელის არხების შეცდომებზე დაყრდნობით და ასევე მარშრუტიზაციის კორექტირება შემდეგი კრიტერიუმების გათვალისწინებით: პაკეტების მიწოდების დაყოვნება, პაკეტების დანაკარგი, დაყოვნების ვარიაციის ფაქტორი, არხების დატვირთვა და სხვა წესები.
7. ინტელექტუალური მარშრუტიზაციის პროტოკოლი აუმჯობესებს ტრადიციულ მარშრუტიზაციას წარმადობაზე გავლენის მქონე პარამეტრების რეალურ რეჟიმში კონტროლით.
8. შემუშავებული მოდელების საფუძველზე აგებულია ალგორითმი, რომელიც ითვალისწინებს არსებული მარშრუტების მომსახურების ხარისხის მახასიათებლების მნიშვნელობებს და ახორციელებს ნაკადების ჯგუფებად დაყოფას. ჯგუფების შექმნა ხორციელდება აპლიკაციების პრიორიტეტული მახასიათებლის მიხედვით.
9. განხილულია ინტელექტუალური მარშრუტიზაციის ალგორითმის შემადგენელი ნაწილების პროტოკოლში ინტეგრირების საკითხები. ასევე გაანალიზებულია მათი მუშაობის პრინციპები. დასაბუთებულია ქსელის მომსახურების ხარისხის მახასიათებლების

საზომი ხელსაწყოების შესაძლებლობები და მათი გამოყენების ასპექტები გაზომვის ფაზაში.

10. ჩატარებულია ინტელექტუალური მარშრუტიზაციის ალგორითმის აპრობაცია ქსელის სიმულატორის ბაზაზე. განხილულია მომსახურების ხარისხის მახასიათებლების კონტროლის ალგორითმების მუშაობის შედეგები. წარმოდგენილია მათი გრაფიკული გამოხატულება.
11. შემუშავებული ალგორითმი არის დაფუძნებული სტანდარტულ კომპონენტებზე და ღია კოდის უტილიტებზე. ეს იძლევა მოცემული სისტემის ნებისმიერი მწარმოებლის მოწყობილობებზე დანერგვის უპირატესობას.

ლიტერატურა

1. J. Doyle, J. Carroll. Routing TCP/IP Volume 1, Second Edition, Cisco Press, 2006, 936p.
2. IETF. Transmission control protocol, RFC: 793, 15p.
3. IETF. Standard Internet Protocol, RFC: 791, 1981, 11p.
4. Adrian Farrel. The Internet ans its Protocols, Morgan Kaufmann Publishers, 2004, 809p.
5. S.Halabi, D.McPherson. Internet Routing Architectures, second edition, Cisco Press, 2000, 528p.
6. В.Олифер, Н.Олифер. Компьютерные сети: принципы, технологии, протоколы, 4-е издание, Питер, 2013, с.944
7. Andrew S. Tanenbaum. Computer networks, 4th edition, Prentice Hall, 2007, 993p.
8. Mark A. Sportack. TCP/IP First Step, Cisco Press, 2005, 401p.
9. Mark A. Sportack. IP Addressing Fundamentals, Cisco Press, 2002, 368p.
10. Khalid Raza, Mark Turner. Large-Scale IP Network Solutions, Cisco Press, 1999, 576p.
11. Эллен Сивер, Стивен Спейнауэр, Стивен Фиггинс, Джессика П. Хекман. Linux Справочник, O'Reilly, 3-е издание, 2001, с.916
12. Cisco System. Internetworking Technologies Handbook, 4th Edition, Cisco Press, 2003, 1128p.
13. Vicki Stanfield, Roderick W. Smith. Linux System Administration, Second Edition, SYBEX, 2002, 479p.
14. Matthew Marsh. Policy Routing Using Linux, 2001, 224p.
15. Vijay Bollapragada, Curtis Murphy, Russ White. Inside Cisco IOS Software Architecture, Cisco Press, 2000, 160p.
16. Cisco Systems, Policy-Based Routing, Cisco Press, White Papers, 1996, pp.1-7
17. http://www.cisco.com/en/US/products/ps6599/products_white_paper09186a00800a4409.shtml, უკანასკნელად იქნა გადამოწმებული - 11.06.2012
18. H. Jonathan Chao, Xiaolei Guo. Quality of Service Control in High-Speed Networks, John Wiley & Sons, 2002, 432p.
19. Neill Wilkinson. Next Generation Network Services, John Wiley & Sons, 2002, 196p.
20. Tim Szigeti, Christina Hattingh. End-to-End QoS Network Design, Cisco Press, 2004, 768p.
21. Barreiros M., Lundqvist P. QOS-Enabled Networks: Tools and Foundations, Wiley, 2011, 244 p.
22. http://docwiki.cisco.com/wiki/Border_Gateway_Protocol, უკანასკნელად იქნა გადამოწმებული - 10.12.2012
23. Rekhter Y., Li T., Hares S. A Border Gateway Protocol 4 (BGP-4), RFC4271, IETF, 2006, 104p.

24. J. Doyle, J. DeHaven Carroll. Routing TCP/IP Volume 2, Cisco Press, 2001, 976p.
25. D. Meyer, K. Patel. BGP-4 Protocol Analysis, RFC4274, IETF, 2006, 14p.
26. Chandra R., Scudder J. Capabilities Advertisement with BGP-4, RFC2842, IETF, 2000, 5p
27. http://www.cisco.com/en/US/tech/tk920/tsd_technology_support_sub-protocol_home.html, უკანასკნელად იქნა გადამოწმებული - 21.01.2013
28. http://www.cisco.com/en/US/docs/ios/12_4/ip_sla/configuration/guide/hstthresh.html, უკანასკნელად იქნა გადამოწმებული - 22.01.2013
29. http://routerjockey.com/2011/05/06/ip-sla-basics/, უკანასკნელად იქნა გადამოწმებული - 04.02.2013
30. IETF. Cisco Systems NetFlow Services Export Version 9, RFC: 3954, 2004. Pp.2-25
31. http://routerjockey.com/2011/05/06/ip-sla-basics/, უკანასკნელად იქნა გადამოწმებული - 15.10.2012
32. K. R. Fall, W. R. Stevens. TCP/IP Illustrated, Volume 1, 2nd Edition, Addison-Wesley, 2011, 1056p.
33. http://www.cisco.com/en/US/products/ps8787/products_ios_protocol_option_home.html, უკანასკნელად იქნა გადამოწმებული - 10.12.2012
34. http://docwiki.cisco.com/wiki/PfR:Technology_Overview, უკანასკნელად იქნა გადამოწმებული - 15.01.2013
35. http://www.gnu.org/software/libc/manual/html_node/Overview-of-Syslog.html, უკანასკნელად იქნა გადამოწმებული - 25.03.2013
36. Matt Messier Pravir Chandra John Viega. Network Security with OpenSSL, O'Reilly Media, 2002, 386p.
37. Douglas Mauro, Kevin Schmidt. Essential SNMP, 2nd Edition, O'Reilly, 2005, 462p.
38. Laurent Bernaille, Renata Teixeira and Kave Salamatian. Early Application Identification, Conference on Future Networking Technologies, CONEXT 2006, pp.1-12
39. Vitali Aivazov, Roman Samkharadze. Identification of flows in the application based routing, Transactions N3(485), Georgian Technical University, pp.62-66
40. http://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186a00800a3db9_ps6601_Products_White_Paper.html, უკანასკნელად იქნა გადამოწმებული - 10.12.2012
41. IETF. Requirements for IP Flow Information Export “IPFIX”, RFC: 3917, 2004. pp.9- 10
42. IETF. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information, RFC 5101, 2008. Pp.11-14
43. IETF. IP Flow Information Export (IPFIX) Applicability, RFC 5472, 2009,24p.

44. Vitali Aivazov, Roman Samkharadze. Available end-to-end throughput measurement tools, Transaction N2(13), Georgian Technical University, pp.123-127
45. R. S. Prasad, M. Murray, C. Dovrolis, K. Claffy. Bandwidth estimation: metrics, measurement techniques, and tools, 11p.
46. Alok Shriram, Margaret Murray, Young Hyun, Nevil Brownlee, Andre Broido. Comparison of Public End-to-End Bandwidth Estimation Tools on High-Speed Links, University of California, San Diego. pp. 6-12
47. http://linux.die.net/man/1/ttcp, უკანასკნელად იქნა გადამოწმებული - 11.11.2012
48. http://www.cisco.com/en/US/tech/tk801/tk36/technologies_tech_note09186a0080094694.shtml, უკანასკნელად იქნა გადამოწმებული - 18.01.2013
49. http://sd.wareonearth.com/~phil/net/ttcp/, უკანასკნელად იქნა გადამოწმებული - 25.03.2013
50. Computational Research Division. Basic usage of Pathchar and pipechar, Lawrence Berkeley, National Laboratory, 8p.
51. Manish Jain and Constantine Dovrolis. Pathload: A measurement tool for end-to-end available bandwidth, PAM 2002, pp.1-4
52. Vinay J. Ribeiro, Rudolf H. Riedi, Jiri Navratil. PathChirp: Efficient Available Bandwidth Estimation for Network Path, Rice University, pp.2-4
53. Guojun Jin, Brian Tierney. Netest: A Tool to Measure the Maximum Burst Size, Available Bandwidth and Achievable Throughput, Lawrence Berkeley National Laboratory, 2003, pp.2-5
54. Addison-Wesley. TCP/IP Illustrated, Volume 1: The Protocols, 1994, ISBN 0-201-63346-9. pp.85-109
55. J. Postel. Internet control message protocol, RFC792, 1981, pp.2-20
56. http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml, უკანასკნელად იქნა გადამოწმებული - 22.02.2013
57. A. Morton, B. Claise. Packet Delay Variation Applicability Statement, RFC5481, 2009, pp.10-12
58. http://www.cisco.com/en/US/tech/tk652/tk698/technologies_tech_note09186a00800945df.shtml, უკანასკნელად იქნა გადამოწმებული - 23.02.2013
59. IETF. IP Packet Delay Variation Metric for IP Performance Metrics, (IPPM), RFC3393, 2002, pp.2-6
60. http://www.latencystats.com/blog/latency-and-network-jitter, უკანასკნელად იქნა გადამოწმებული - 28.11.2012
61. A.Hernandez, E.Magana. One-way Delay Measurement and Characterization, Universidad Publica de Navarra, Pamplona, Spain, IEEE, 2007, pp.1-2
62. Allen B. Downey. Using pathchar to estimate Internet link characteristics, Colby College, pp.2-3