გ.ოთხოზორია ზ.ცირაშუა შ.სვანიშვილი

# მარშუტიზაცია და კომუტაცია ქსელებში
## (ქსელის ადმინისტრირება 1)

თბილისი 2015

# შესავალი

წინამდებარე სახელმძღვანელო შედგენილია პროფესიული სტუდენტებისათვის, საგანმანათლებლო პროგრამის „კომპიუტერული ქსელის ადმინისტრირება“ სწავლებისათვის და მოიცავს ამ პროგრამით გათვალისწინებულ I ეტაპზე სწავლებად ძირითად მოდულებს.

სახელმძღვანელოს მიზანია დაეხმაროს სტუდენტს დაეუფლოს კომპიუტერული ქსელის დაგეგმვა-გამართვის პრინციპებს და შეძლოს მისი ეფექტური გამოყენება პროფესიულ საქმიანობაში. სახელმძღვანელოში მოცემული საკითხები საინტერესო და აქტუალური იქნება ქსელის ადმინისტრატორებისათვის და დაეხმარება მათ მცირე და საშუალო ქსელების ადმინისტრირებაში. სახელმძღვანელოში მოცემული ინსტრუმენტების გამოყენება შესაძლებელია, როგორც სწავლების, ასევე დასაქმებისა და ყოველდღიური საქმიანობის პირობებში.

სახელმძღვანელოში აღწერილია მიმდინარე პერიოდში აქტუალური და ფართოდ გამოყენებადი პროგრამულ-აპარატურული უზრუნველყოფის ელემენტები და სერვისები.

სახელმძღვანელო დაყოფილია 6 ნაწილად (თავი). ყოველი თავი შეესაბამება კონკრეტული მოდულის დასახელებას, შესაბამისი ქვეთავები კი ეხმაურება მოდულის ჩარჩოთი განსაზღვრულ სწავლის შედეგებს. თითოეულ ნაწილში თემატურ ტექსტურ-ილუსტრირებულ მასალასთან ერთად წარმოდგენილია სავარჯიშოები, სახელმძღვანელოს ყოველი თავის ბოლოს შეფასების მიდგომებიდან გამომდინარე დართული აქვს შემაჯამებელი სამუშაო, კონკრეტული მოდულების ჩარჩოთი გათვალისწინებული შეფასების რუბრიკით გათვალისწინებული აქტივობები, პრაქტიკული დავალება-სავარჯიშო ან/და ტესტის ნიმუში

რეცენზენტები: მიხეილ სამხარაძე

განათლების მართვის საინფორმაციო სისტემა(EMIS)

საგანმანათლებლო პროგრამის შემმუშავებელი ჯგუფი

ვლადიმერ ადამია

სტუ-ს პროფესორი, სტუ-ს კომპიუტერული ქსელის ადმინისტრატორი

სარჩევი

# 1. ქსელის საფუძვლები, კომპონენტები და აპარატურა, უსადენო ქსელის საფუძვლები

## 1.1. ქსელის ტიპები

ადამიანებს შორის კომუნიკაცია მნიშვნელოვან როლს თამაშობს მათ ცხოვრებაში. მათ სჭირდებათ მიიღონ ინფორმაცია ერთმანეთზე, ახალ ამბებზე, ამინდზე, ფინანსურ მაჩვენებლებზე და ა.შ. ინფორმაციის მიღების და გადაცემის მეთოდები იცვლებოდა და ვითარდებოდა წლების განმავლობაში. ინფორმაციულ საუკუნეში რომელშიც ჩვენ ვცხოვრობთ ინფორმაციის დროული მიღება და ფლობა უაღრესად მნიშვნელოვანია. ამიტომ ინფორმაციის მიღებასა და გადაცემაში კომპიუტერული ქსელი უმნიშვნელოვანეს როლს თამაშობს. კომპიუტერული ქსელი ეხმარება ადამიანებს უსწარაფესად გადასცენ ინფორმაცია მსოფლიოს ნებისმიერ ადგილას. მსოფლიოში მონაცემების გადაცემა გახდა კომპიუტერული სისტემების ფუნდამენტური ნაწილი. კომპიუტერული ტექნოლოგიების სწრაფმა განვითარებამ მოითხოვა კომპიუტერული სისტემების საიმედო, სწრაფი და დაცული კავშირების უზრუნველყოფა. ამიტომ კომპიუტერული ქსელების დაპროექტების, აგების და მართვის სისტემები მნიშვნელოვან როლს თამაშობს თანამედროვე ინფორმაციულ ტექნოლოგიებში.

## 1.2. ქსელების ფუნდამენტური პრინციპები და ტიპები

**რა არის ქსელი?** - ქსელი (Network) - ინფორმაციის გაცვლისა და რესურსების ერთობლივად გამოყენებისათვის, ერთმანეთთან ფიქსირებულად ან/და მობილურად დაკავშირებული კომპიუტერების ჯგუფი.

საინფორმაციო ქსელები ერთმანეთისაგან განსხვავდებიან სხვადასხვა შესაძლებლობებით, მაგრამ ყველა ქსელს გააჩნია ოთხი ძირითადი საერთო ელემენტი:

- წესები (პროტოკოლი), თუ როგორ უნდა მოხდეს ინფორმაციის გაგზავნა და მიღება;

- ინფორმაცია ან ინფორმაციის ერთეული, რომელიც იგზავნება ერთი მოწყობილობიდან მეორეში;
- მედია საშუალება, რომლითაც ხდება ამ მოწყობილობების დაკავშირება;
- ქსელური მოწყობილობები, რომლებიც ცვლიან ერთმანეთთან ინფორმაციას.

*სურ.1.2. 1*

**ქსელში გამოყენებული რესურსები** - პროგრამები, მონაცემთა ფაილები, აგრეთვე პრინტერები და ქსელში სხვა ერთობლივად მოხმარებადი პერიფერიული მოწყობილობები.

**ქსელში შეიძლება იყოს გაზიარებული მრავალი ტიპის რესურსი** -

- სერვისები, როგორიც არის ამობეჭდვა და სკანირება.
- მონაცემების შესანახი სივრცე და მოძრავი(removable) მოწყობილობები, როგორებიც არის მყარი და ოპტიკური დისკები
- პროგრამები, მონაცემთა ბაზები.

კომპიუტერული ქსელი წარმოადგენს ურთიერთდაკავშირებულ და შეთანხმებულად ფუნქციონირებად პროგრამული და აპარატურული კომპონენტების რთულ კომპლექსს. ის არის კომპიუტერების და პერიფერიული მოწყობილობების ერთიანობა, რომლებსაც სპეციალური საკომუნიკაციო საშუალებების და პროგრამული უზრუნველყოფის საშუალებით შეუძლიათ ინფორმაციის გაცვლა. კომპიუტერულ ქსელში კომპიუტერების რაოდენობა ორიდან რამდენიმე ათასამდე შეიძლება იცვლებოდეს.

კომპიუტერული მონაცემთა ქსელი არის ჰოსტების(Host კვანძი) ერთობლიობა, დაკავშირებული ერთმანეთთან ქსელური მოწყობილობების საშუალებით. ჰოსტი არის ნებისმიერი მოწყობილობა რომელიც აგზავნის და ღებულობს ინფორმაციას ქსელში.

ჰოსტებთან დაკავშირებულ მოწყობილობებს ეწოდებათ პერიფერიული მოწყობილობები. მაგ. პრინტერი დაკავშირებული ქსელში ჩართულ კომპიუტერთან. თუმცა თუ პრინტერი არის დაკავშირებული პირდაპირ ისეთ ქსელურ მოწყობილობასთან როგორიც არის კონცენტრატორი, კომუტატორი ან მარშრუტიზატორი, ამ შემთხვევაში პრინტერიც არის ჰოსტი.

შესაძლებელია კომპიუტერული ქსელების კლასიფიკაციის მრავალი სხვადასხვა ხერხი, მათ შორის რაოდენობისა და ქსელის ზომის მიხედვით, მონაცემთა გადაცემის ტიპის მიხედვით, ინფორმაციის გადაცემის სიჩქარის მიხედვით.

ქსელები შეიძლება დავყოთ 3 ძირითად კლასად:

**ლოკალური ქსელი (LAN - Local Area Network)** - ერთმანეთთან დაკავშირებული, ერთი ადმინისტრირების ქვეშ მოქცეული კომპიუტერების შედარებით მცირე ჯგუფი.

მნიშვნელოვანია დავიმახსოვროთ, რომ ლოკალური ქსელის ელემენტები იმყოფება ადმინისტრირების ერთი ჯგუფის მართვის ქვეშ, რომელიც განსაზღვრავს ქსელში მომქმედ წვდომის მართვასთან დაკავშირებულ პოლიტიკასა და უსაფრთხოებას ამ კონტექსტში სიტყვა ”ლოკალური” მიანიშნებს ერთობლივ ”ლოკალურ” მართვას და არა კომპონენტებს შორის ფიზიკურ სიახლოვეს

**რეგიონალური ქსელი (MAN – Metropolitanl Area Network)**- ქსელი, რომელიც აერთიანებს ბევრ ლოკალურ ქსელს ერთი რაიონის, ქალაქის ან რეგიონის ფარგლებში.

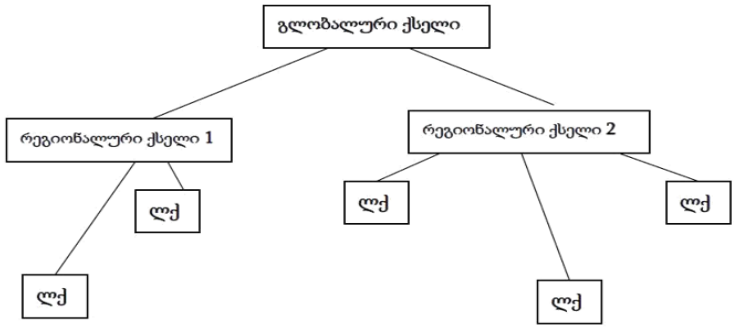
**გლობალური ქსელი (WAN – Wide Area Network)**- ქსელი, რომელიც აერთიანებს სხვადასხვა ქალაქების, რეგიონების და სახელმწიფოების კომპიუტერებს.

გლობალური ქსელის თვალსაჩინო მაგალითს წარმოადგენს ინტერნეტი (Internet). Internet-ი ეს გახლავთ ფართო გლობალური ქსელი, რომელიც თავის თავში მოიცავს მილიონობით ურთიერთდაკავშირებულ ლოკალურ ქსელს. ლოკალურ ქსელებს შორის კავშირის რეალიზაციას ახდენენ ტელეკომუნიკაციური მომსახურების მომწოდებლები.

ლოკალური ქსელები შეიძლება შედიოდეს რეგიონულ ქსელებში კომპონენტების სახით; რეგიონალური ქსელები - გაერთიანდნენ გლობალური ქსელის შემადგენლობაში; გლობალურმა ქსელებმა შეიძლება შექმნან უფრო მსხვილი სტრუქტურები. პლანეტა

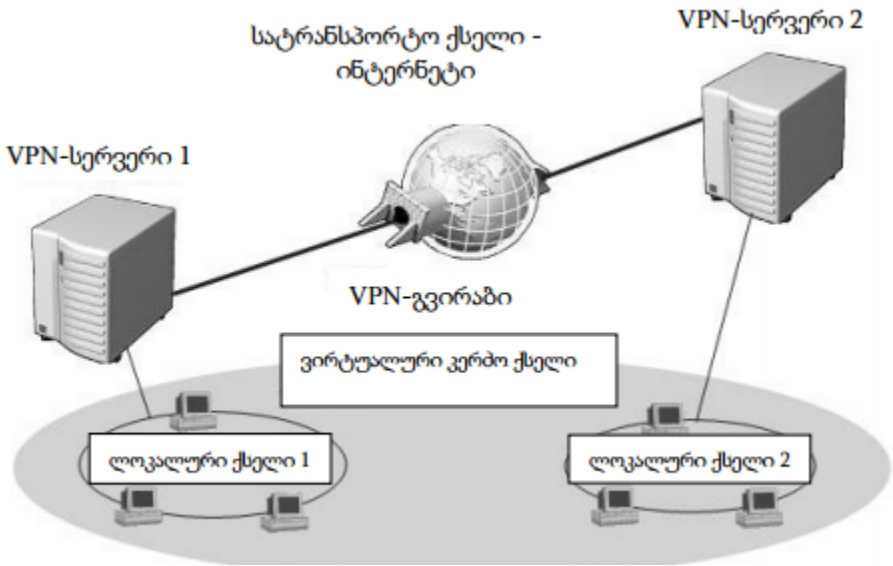
დედამიწის მასშტაბით დღეისათვის კომპიუტერული ქსელების ყველაზე დიდი გაერთიანებაა ”ქსელთა ქსელი” - ინტერნეტი.

გლობალური, რეგიონალური და ლოკალური ქსელების გაერთიანება იძლევა მრავალდონიანი იერარქიების შექმნის საშუალებას, რომლებიც თავის მხრივ იძლევა მონაცემთა უზარმაზარი მასივების დამუშავებისა და ინფორმაციული რესურსებისადმი პრაქტიკულად შეუზღუდავი ხელმისაწვდომობის მძლავრ საშუალებებს



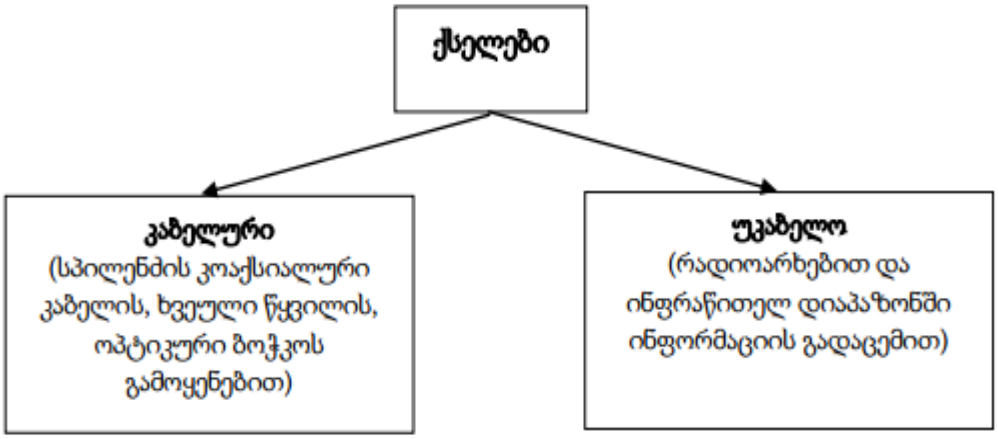

*სურ.1.2. 2*

ლოკალური და გლობალური ქსელების გაერთიანების საინტერესო მაგალითია **ვირტუალური კერძო ქსელი (Virtual Private Network, VPN)**. ასე ეწოდება ორგანიზაციის ქსელს, რომელიც მიიღება ორი ან რამოდენიმე ტერიტორიულად განცალკევებული ლოკალური ქსელის გაერთიანებით საყოველთაოდ ხელმისაწვდომი გლობალური ქსელების არხების დახმარებით, მაგალითად, ინტერნეტით.




*სურ.1.2. 3*

მონაცემების გადაცემის ტიპის მიხედვით ქსელები იყოფა **კაბელურ და უკაბელო ქსელებად**

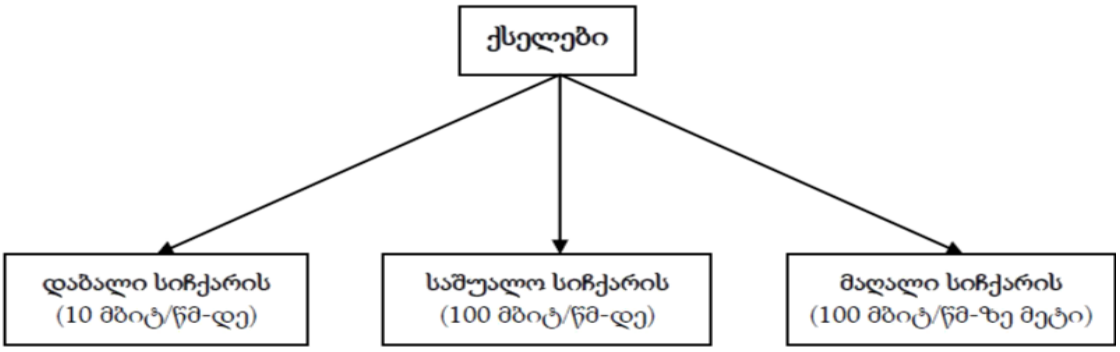



*სურ.1.2. 4*

ქსელური მოწყობილობები ურთიერთდაკავშირებულნი არიან სხვადასხვა ტიპის კავშირის საშუალებებით:

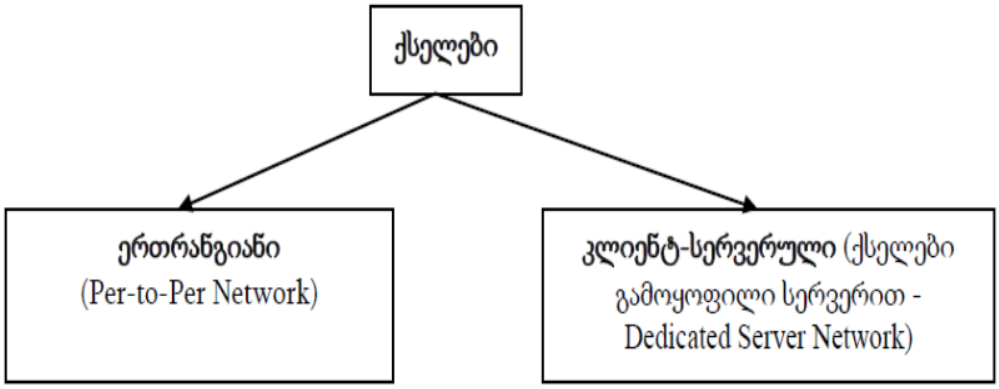
- **სპილენძის კაბელებით** - მოწყობილობებს შორის მონაცემთა გადასაცემათ იყენებს დენის სიგნალს.
- **ოპტიკურ-ბოჭკოვანი კაბელებით** - იყენებს შუშას და პლასტმასის სადენს, ე.წ. ბოჭკოვანს, რათა გადასცეს სინათლის სხივის იმპულსების მეშვეობით ინფორმაცია.
- **უკაბელო კავშირი** - იყენებს რადიო სიგნალებს, ინფრაწითელ ტექნოლოგიას (ლაზერებს), ან სატელიტურ კავშირებს.

ინფორმაციის გადაცემის სიჩქარის მიხედვით ქსელები შეიძლება დავყოთ **დაბალი, საშუალო,** და **მაღალი სიჩქარის ქსელებად:**



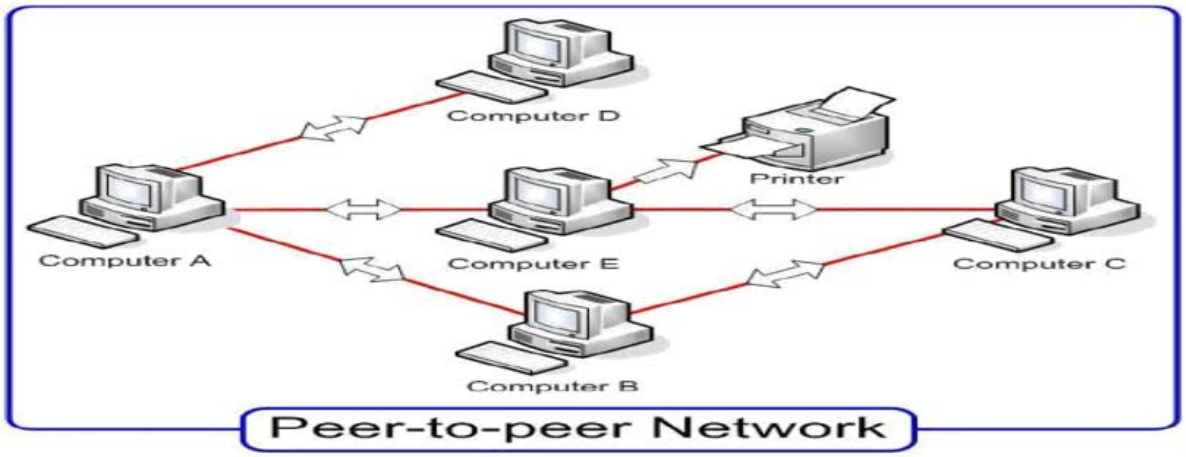

*სურ.1.2. 5*

კომპიუტერებს შორის როლების განაწილების თვალსაზრისით ქსელები არსებობენ **ერთრანგიანი** და **კლიენტ-სერვერული**



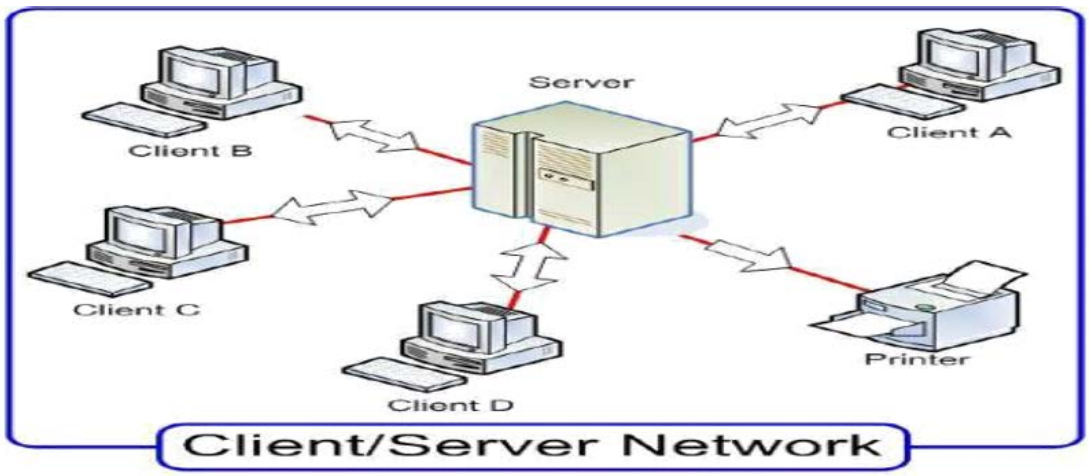

*სურ.1.2. 6*

**ერთრანგიანი ქსელები** - ერთრანგიან ქსელში ყველა კომპიუტერი თანასწორუფლებიანია. ყოველ მათგანს შეუძლია შეასრულოს როგორც სერვერის როლი, ე. ი. მიაწოდოს ფაილები და აპარატურული რესურსები (დამგროვებლები, პრინტერები და სხვა) დანარჩენ კომპიუტერებს, ასევე კლიენტის როლი, რომელიც სარგებლობს სხვა კომპიუტერების რესურსებით




*სურ.1.2. 7*

**ქსელები გამოყოფილი სერვერით (”კლიენტ-სერვერ” ტიპის ქსელები) -** ასეთ ქსელებში ხდება ერთი ან რამოდენიმე კომპიუტერის გამოყოფა - სერვერების სახით, რომელთა ამოცანაც მდგომარეობს სხვა კომპიუტერების - კლიენტების მრავალრიცხოვანი მოთხოვნების სწრაფ და ეფექტურ დამუშავებაში. ამავე დროს კლიენტური მოთხოვნები შეიძლება იყოს სრულიად განსხვავებული, დაწყებული უმარტივესით - სისტემაში შესვლისას მომხმარებლის სახელის და პაროლის შემოწმებით, დამთავრებული მონაცემთა ბაზებისადმი რთული საძიებო მოთხოვნებით.

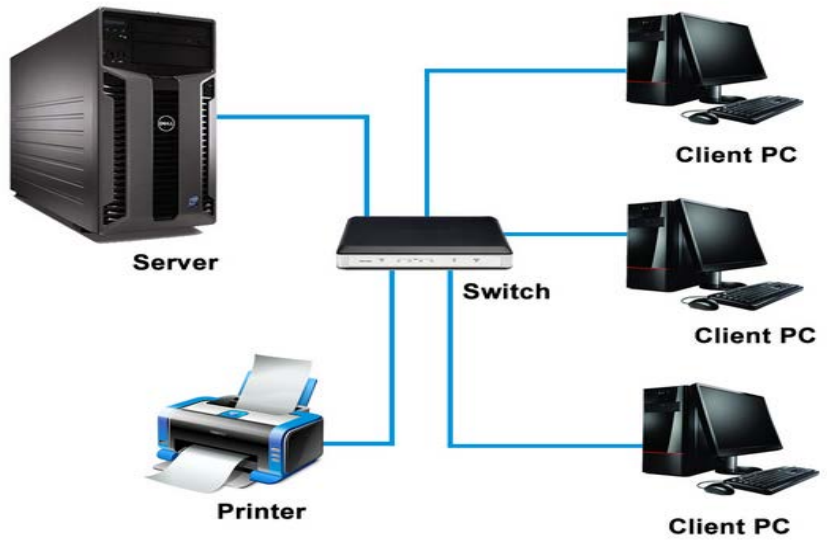



*სურ.1.2. 8*

**სერვერი** - სპეციალურად გამოყოფილი მაღალმწარმოებლური კომპიუტერი, აღჭურვილი შესაბამისი სერვერული პროგრამული უზრუნველყოფით, რომელიც

ცენტრალიზებულად მართავს ქსელის მუშაობას და/ან აწვდის ქსელის სხვა კომპიუტერებს თავის რესურსებს

**კლიენტური კომპიუტერი (კლიენტი, მუშა სადგური)** - ქსელის რიგითი მომხმარებლის კომპიუტერი, რომელიც ღებულობს დაშვებას სერვერის (სერვერების) რესურსებისადმი.




*სურ.1.2. 9*

**ქსელის უპირატესობები:**

- ქსელში საჭიროა ნაკლები პერიფერიული მოწყობილობა.
  - იმის გამო რომ ქსელში გვაქვს შესაძლებლობა გავანაწილოთ რესურსები და მივცეთ დაშორებულ კომპიუტერებს წვდომა ჩვენს პერიფერიულ მოწყობილობებზე, გამოირიცხა მიზეზი, რომ თითოეულ კომპიუტერს შეიძლებოდა დასჭირვებოდა ცალკე პრინტერი თუ სკანერი ან სხვა მოწყობილობა
- ქსელის მეშვეობით იზრდება კავშირგაბმულობის შესაძლებლობები
  - ქსელი გვაძლევს სხვადასხვა ტიპის ხელსაწყოების გამოყენების შესაძლებლობას კავშირგაბმულობისათვის - იქნება ეს ფორუმები, ჩეთები, იმეილები, აუდიო თუ ვიდეო კავშირის საშუალებები, ამ ხელსაწყოების გამოყენებით ადამიანებს შეუძლიათ გაცვალონ

ინფორმაცია, დაუკავშირდნენ თავიანთ მეგობრებს, ოჯახის წევრებსა და კოლეგებს.

- ფაილების დუბლირებისა და დაზიანებისაგან დაცვა
    - სერვერი განაგებს ქსელურ რესურსებს, ის ინახავს მონაცემებს და ანაწილებს მათ მომხმარებლებს შორის, კონფიდენციალური მონაცემების დაცვა შეიძლება განხორციელდეს და მასზე წვდომა იყოს დაშვებული მხოლოდ განსაკუთრებული მომხმარებლებისათვის.
- ლიცენზირების უფრო დაბალი ფასი
    - პროგრამების ლიცენზიები ხშირად უფრო ძვირია ინდივიდუალურ მანქანებზე დასაყენებლად. ბევრი მწარმოებელი კომპანია იძლევა ე.წ. ”Site license”-ის შემოთავაზებას - ლიცენზია ქსელებისათვის, რაც ნიშნავს რომ ერთი კონკრეტული ფასით, ადამიანთა რაიმე ჯგუფს ან კომპანიის ყველა თანამშრომელს შუძლია ჰქონდეს წვდომა შესაბამის პროგრამულ უზრუნველყოფაზე
- ცენტრალიზირებული ადმინისტრირება
    - ცენტრალიზირებული ადმინისტრირება ამცირებს ხალხის რაოდენობას, რომელიც საჭიროა ქსელური მოწყობილობებისა და ქსელში მონაცემების სამართავად, რაც თავის მხრივ ამცირებს კომპანიის დანახარჯებს როგორც ფინანსურს ასევე დროითს, ინდივიდუალურ მომხმარებლებს არ სჭირდებათ თავიანთი მონაცემებისა და მოწყობილობების მართვა, ერთ ადმინსიტრატორს შეუიძლია მართოს მონაცემები, მოწყობილობები და მომხმარებლების დაშვების უფლებები ქსელში, მონაცემების რეზერვირებაც მარტივდება, რადგან ისინი სრულად ინახება ერთ ცენტრალურ ადგილზე.
- რესურსების ეკონომია
    - სამუშაო შეიძლება იქნას გადანაწილებულ იქნას რამოდენიმე კომპიუტერს შორის და შედეგად არ მოხდეს ინფორმაციის გადამუშავებით არცერთი ცალკე აღებული კომპიუტერის გადატვირთვა

**კითხვები თვითშემოწმებისთვის:**

1. რა არის ქსელი?
2. როგორი ტიპის ქსელები იცით?
3. როგორ უპირატესობებს იძლევა ქსელი?
4. რა არის ერთრანგიანი ქსელი? როგორია მისი უპირატესობები და ნაკლოვანებები?
5. რა არის ”კლიენტ-სერვერული” ქსელი? როგორია მისი უპირატესობები და ნაკლოვანებები?
6. რას გულისხმობს ცნება ”ქსელის ადმინისტრირება”?
7. როგორი აპარატურული და პროგრამული საშუალებებია საჭირო ქსელში კომპიუტერების ურთიერთქმედების უზრუნველსაყოფად?

## 1.3. ქსელური მოწყობილობების დაკავშირება

ნებისმიერი კომპიუტერული მოწყობილობის ქსელში ჩასართავად მას უნდა ჰქონდეს ქსელის ადაპტერი(NIC) განკუთვნილი კაბელური ან უკაბელო შეერთებისთვის


*სურ.1.3. 1*

კომპიუტერი ქსელის ადაპტერით - კაბელით (სპილენძის გრეხილი წყვილი TP ან ოპტიკურ-ბოჭკოვანი fiber optic) ან უსადენოდ უკავშირდება რომელიმე ქსელურ მოწყობილობას


*სურ.1.3. 2*

## ქსელური მოწყობილობების სახელები, დანიშნულება და მახასიათებლები

- ✓ მოდემი (Modem)
- ✓ კონცენტრატორი (Hub)
- ✓ კომუტატორი (Switch)
- ✓ მარშრუტიზატორი (Router)
- უკაბელო წვდომის წერტილები (Wireless Access Point)
- მრავალფუნქციური მოწყობილობები

*სურ.1.3. 3*

### მოდემი

*სურ.1.3. 4*

მოდემი - ელექტრონული მოწყობილობაა, რომელიც სატელეფონო ხაზებში, ანალოგური სიგნალის მეშვეობით გადასცემს ინფორმაციას კომპიუტერებს შორის.

მოდემი ინფორმაციის გადაცემისას გარდაქმნის ციფრულ მონაცემებს ანალოგურ სიგნალად, ხოლო მიღებისას ანალოგურ სიგნალს გარდაქმნის ციფრულ მონაცემებად, რომლის ინტერპრეტირებაც ხდება კომპიუტერის მიერ. ეს პროცესი იწოდება მოდულაცია/დემოდულაციად.

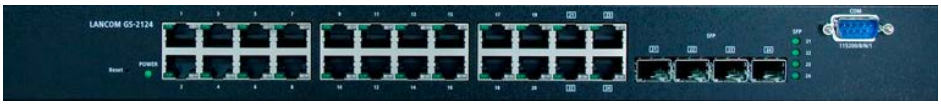
### კონცენტრატორი (Hub)

*სურ.1.3. 5*

უზრუნველყოფს ერთ ქსელში რამოდენიმე კომპიუტერის ჩართვას(პორტების რაოდენობის მიხედვით) შემოსულ პაკეტებს უგზავნის ყველა კომპიუტერს, მიუხედავად იმისა, არის თუ არა მისთვის განკუთვნილი. ამის გამო ხდება ქსელში მოძრაობის, იგივე ტრაფიკის (Trafic), გადატვირთვა.

ჰაბს შეიძლება ჰქონდეს 6, 12 და მეტი RJ 45 პორტი.

## კომუტატორი (Switch)




*სურ.1.3. 6*

უზრუნველყოფს ერთ ქსელში რამოდენიმე კომპიუტერის ჩართვას(პორტების რაოდენობის მიხედვით)

ჰაბისაგან განსხვავებით კომუტატორს შეუძლია პაკეტის თავსართში ამოიკითხოს MAC მისამართი, გაარკვიოს რომელი ქსელის ადაპტერს(NIC) ეკუთვნის პაკეტი და გაუგზავნის ადრესატ კომპიუტერს. ანუ სვიჩი მონაცემებს უგზავნის იმ კომპიუტერს, რომლისთვისაცაა განკუთვნილი. არსებობს ორი სახის სვიჩი: გამჭოლი და შემნახველი. გამჭოლი სვიჩები ჩვეულებრივ მიიღებენ პაკეტებს და გადაუგზავნიან შესაბამის კომპიუტერებს, ხოლო შემნახველ სვიჩებს აქვთ საკუთარი პროცესორი და მეხსიერების ბუფერი. ისინი აგროვებენ შემოსულ პაკეტებს, ამოწმებენ შეცდომებს, შემდეგ ისევ ანაწილებენ და გადასცემენ შესაბამის კომპიუტერებს. მუშაობის პრინციპიდან გამომდინარე, სვიჩებს უფრო მეტი შესაერთებლები აქვთ და ჰაბების მსგავსად მათი ერთმანეთთან მიერთებაც შეიძლება.

### მარშრუტიზატორი (Router)

*სურ.1.3. 7*

გამოიყენება სხვადასხვა ქსელების ერთმანეთთან დასაკავშირებლად, განსაზღვრავს მარშრუტს დაშორებულ ქსელებში ინფორმაციის გადაცემისას

### უკაბელო წვდომის წერტილები (Wireless Access Point)

*სურ.1.3. 8*

უკაბელო წვდომის წერტილებთან შესაძლებელია მოხდეს დაკავშირება კომპიუტერებით, რომლებსაც აქვთ უკაბელო ქსელური ადაპტერი. ისინი კომუნიკაციისათვის რადიოტალღებს იყენებენ. მათი დაფარვის ზონა შეზღუდულია. დიდ ქსელებს ესაჭიროებათ რამდენიმე ასეთი წერტილი ადეკვატური დაფარვისათვის.

### მრავალფუნქციური მოწყობილობები

არსებობს მოწყობილობები, რომლებსაც ერთად რამდენიმე ფუნქცია აქვთ ჩაშენებული. გაცილებით მოხერხებულია ამგვარი მოწყობილობებით სარგებლობა, განსაკუთრებით საცხოვრებელ ბინებში. ერთ ამგვარ მოწყობილებას შეუძლია შეითავსოს მარშრუტიზატორის, კომუტატორის და უკაბელო წვდომის წერტილის ფუნქციები.

*სურ.1.3. 9*

## ქსელური კაბელების სახელები, დანიშნულება და მახასიათებლები

### *კოაქსიალური კაბელი და მისი შეერთება ქსელის ადაპტერთან*

კოაქსიალური კაბელი ყველაზე მეტად იყო გავრცელებული თავისი ფასის, წონისა და პრაქტიკულობის და ასევე დაყენების სიმარტივის გამო. მარტივი კოაქსიალური კაბელი შედგება სპილენძის გამტარისაგან, ირგვლივ შემოხვეული საიზოლაციო შრისაგან, მეტალის წნულისაგან (ეკრანისაგან) და გარე გარსისაგან. ზოგჯერ მეტალის წნულის გარდა აქვს ფოლგის ფენაც და ასეთს ეწოდება კაბელი ორმაგი ეკრანიზაციით. კოაქსიალური კაბელი შეფერხებების მიმართ უფრო გამძლეა, ვიდრე ხვეულა წყვილი და სიგნალების მილევაც ნაკლებია მასში. სიგნალის მილევა არის კაბელში გავლისას სიგნალების შესუსტება.

კოაქსიალური კაბელის ორი ტიპი არსებობს: წვრილი კოაქსიალური კაბელი (thinnet) და მსხვილი კოაქსიალური კაბელი (thicknet).

წვრილი კაბელი მოქნილია, ასეთ კაბელებს ინფორმაციის დაუმახინჯებლად გადაცემა შეუძლია 185 მ-მდე. სქელი კოაქსიალური კაბელი შედარებით ხისტია, დიამეტრი 1 სმ-მდე აქვს. სქელ კოაქსიალური კაბელს მონაცემთა დაუმახინჯებლად გადაცემა შეუძლია 500 მ-დე მანძილზე.

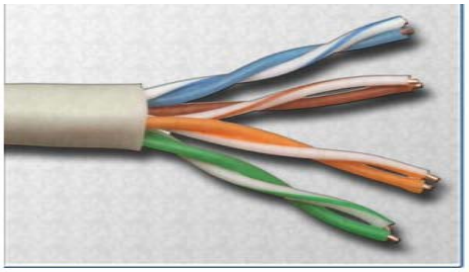
*სურ.1.3. 10*

### *გრეხილი წყვილი –TP (Twisted Par)*

- UTP **(Unshielded twisted pair)**
- STP **(Shielded twisted pair)**
- FTP **(Foiled Twisted Pair)**

თავის მხრივ UTP კაბელები
4 კატეგორიისა:

- UTP 3,
- UTP 5,
- UTP 5e
- UTP 6.



*სურ.1.3. 11*

UTP კაბელით მონაცემთა გადაცემა შესაძლებელია 100 მეტრამდე მანძილზე, უფრო შორს სიგნალების გადასაცემად საჭიროა ყოველ 100 მეტრში ჩავაყენოთ ქსელური მოწყობილობა, თუმცა 500 მეტრზე შორს ამ კაბელის გამოყენება აღარ შეიძლება, ანუ ერთ გზაზე შეგვიძლია ჩავაყენოთ მხოლოდ 4 ქსელური მოწყობილობა.

ამ კაბელებს ხვიურ წყვილებს იმიტომ უწოდებენ, რომ შედგება სადენთა 4 წყვილისაგან, რომელთაგან თითოეული ერთმანეთზეა დახვეული. ეს შემთხვევით არ არის ასე, ცნობილია, რომ სადენთა ერთმანეთზე გადახვევა ხელს უშლის ელექტრო-მაგნიტური ველის შექმნას, ე.ი. კაბელში მონაცემთა დამახინჯებას. თითოეული წყვილი განსხვავდება თავისი ფერით. ერთმანეთზე დახვეულია ლურჯი და თეთრი-ლურჯი ზოლით, მწვანე და

თეთრი-მწვანე ზოლით, ნარინჯისფერი და თეთრი-ნარინჯისფერი ზოლით, ყავისფერი და თეთრი-ყავისფერი ზოლით. ფერთა ეს განლაგება ყველა კაბელში ერთნაირია და ამას თავისი მიზეზი აქვს, რასაც მოგვიანებით გავიგებთ. UTP 5e-ს განსხვავებით UTP 5-ისგან მეტი გრეხილი აქვს. ხოლო UTP 6 კაბელი შეიცავს „პლასტიკურ გამყოფს“ წყვილებს შორის. რაც ხელს უშლის ხარვეზებს (დაბრკოლებებს).

### ოპტიკურ-ბოჭკოვანი კაბელი

*სურ.1.3. 12*

ოპტიკურ-ბოჭკოვან კაბელში მონაცემთა გადაცემა ხდება მოდულირებული სინათლის იმპულსების სახით. იგი მონაცემთა გადაცემის შედარებით დაცული ხერხია. ასეთი ტიპის ხაზები გამოიყენება დიდი მოცულობის მონაცემების გადასაცემად დიდი სისწრაფით (10 გიგაბიტი/წამამდე). მათში სიგნალების მილევა და დამახინჯება თითქმის არ ხდება. ოპტიკური ბოჭკო წვრილი შუშის ცილინდრია (5-60 მიკრონი), რომელსაც ქვია შუშის ფენით დაფარული სასიგნალო გამტარი. ყოველი ოპტიკური ბოჭკო სიგნალს გადაცემს ერთი მიმართულებით, ამიტომ ყოველი კაბელი შედგება ორი ოპტიკური ბოჭკოსგან, რომლებსაც აქვთ დამოუკიდებელი კონექტორები; ერთი მათგანი გამოიყენება გადასაცემად, მეორე – მიმღებად. დღესდღეობით კომპიუტერულ ქსელებში გამოიყენება სამივე ტიპის კაბელი, მაგრამ ყველაზე პერსპექტიულია ოპტიკურ-ბოჭკოვანი, ის გამოიყენება მაგისტრალების ასაგებად.

ოპტიკურ-ბოჭკოვანი კაბელით ინფორმაციის გადაცემის დროს მასზე არ მოქმედებს ელექტრული შეფერხებები, არ ხდება სიგნალის დამახინჯება და მილევა, ამიტომ გადაცემა

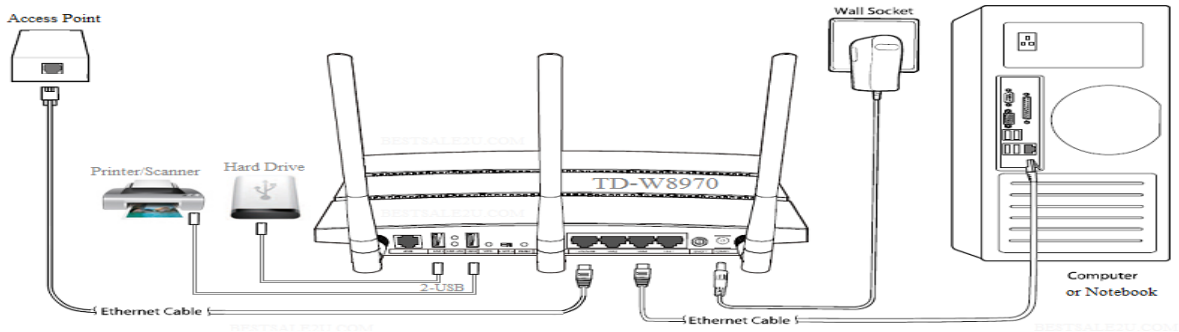
ხდება ძალიან დიდი, წამში ასობით მეგაბიტი, სიჩქარით, რომლის თეორიული ზღვარი 200000 მგბტ/წმ-ის ტოლია. არსებობს ორი ტიპის ოპტიკურ-ბოჭკოვანი კაბელი:

• Multimode - ამ ტიპის კაბელს სქელი „გული“ აქვს, შესაბამისად მისი დამზადება უფრო ადვილია. სინათლის წყაროდ შესაძლებელია გამოვიყენოთ უფრო მარტივი წყარო (შუქდიოდი). ის კარგად მუშაობს რამდენიმე კილომეტრზე.

• Singlemode - მას გააჩნია ძალიან თხელი „გული“ აქვს და შესაბამისად მისი დამზადებაც უფრო ძვირია. ის სინათლის წყაროდ იყენებს ლაზერს და თავისუფლად შეუძლია გადასცეს ინფორმაცია ათეულობით კილომეტრზე.

## 1.4. მარტივი სადენიანი და უსადენო ქსელის გამართვა

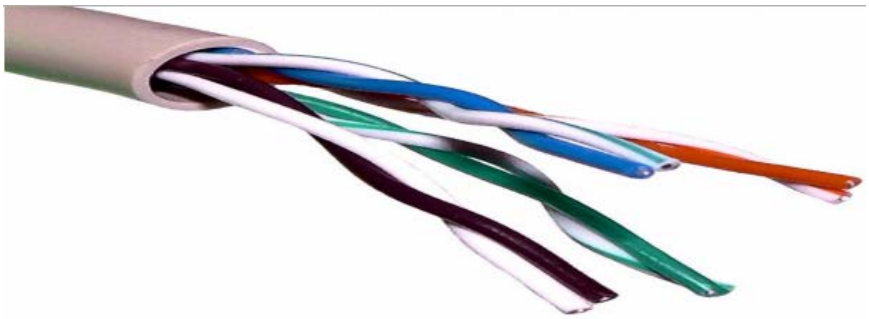
### *1.4.1. სადენიანი ქსელი*




*სურ.1.4.1. 1*

მცირე ზომის ქსელებში სადენის სახით უმრავლეს შემთხვევაში გამოიყენება სპილენძის გრეხილი წყვილი –TP (Twisted Par). ამ კაბელებს ხვიურ წყვილებს იმიტომ უწოდებენ, რომ შედგება სადენთა 4 წყვილისაგან, რომელთაგან თითოეული ერთმანეთზეა დახვეული. ეს შემთხვევით არ არის ასე, ცნობილია, რომ სადენთა ერთმანეთზე გადახვევა ხელს უშლის ელექტრო-მაგნიტური ველის შექმნას, ე.ი. კაბელში მონაცემთა დამახინჯებას. თითოეული წყვილი განსხვავდება თავისი ფერით. ერთმანეთზე დახვეულია ლურჯი და თეთრი-ლურჯი ზოლით, მწვანე და თეთრი-მწვანე ზოლით, ნარინჯისფერი და თეთრი-

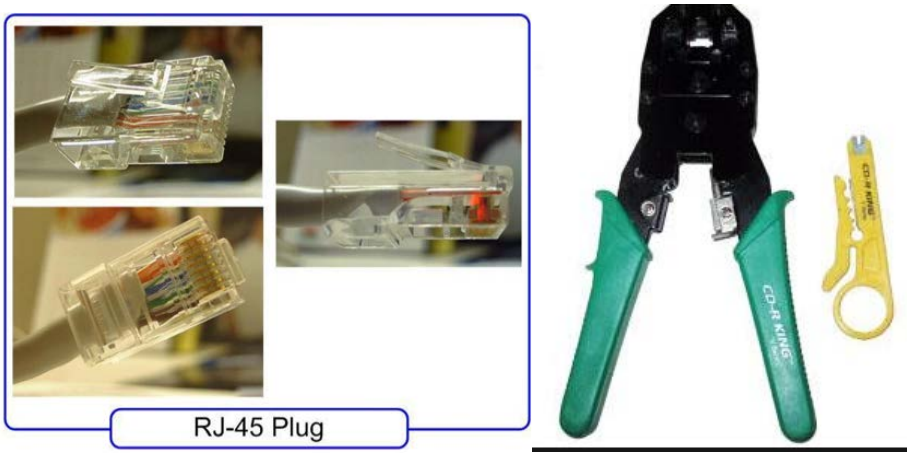
ნარინჯისფერი ზოლით, ყავისფერი და თეთრი-ყავისფერი ზოლით. ფერთა ეს განლაგება ყველა კაბელში ერთნაირია



*სურ.1.4.1. 2*

სიიაფის, დაყენების სიმარტივისა და უნივერსალურობის გამო (შეიძლება გამოვიყენოთ ქსელური ტექნოლოგიების უმრავლესობაში), ამჟამად ლოკალური ქსელების აგებისას ყველაზე გავრცელებული ტიპის კაბელია არაეკრანირებული ხვეული წყვილი. მიუხედავად ხელშეშლების წინააღმდეგ მდგრადობისა, მონტაჟის სირთულის გამო (საჭიროა ზრუნვა დამიწებაზე), არაეკრანირებულ ხვეულ წყვილთან შედარებით, ეკრანირებული ხვეული წყვილი მეტი სიხისტის გამო არ არის ფართოდ გავრცელებული.

ხვეული წყვილი უერთდება კომპიუტერსა და სხვა მოწყობილობებს რვაკონტაქტიანი გასართით (კონექტორით) RJ-45 (Registered Jack 45). ეს კონექტორი ჰგავს სატელეფონო ქსელებში გამოყენებად RJ-11 კონექტორს, ოღონდ მასზე ცოტათი მოზრდილია.



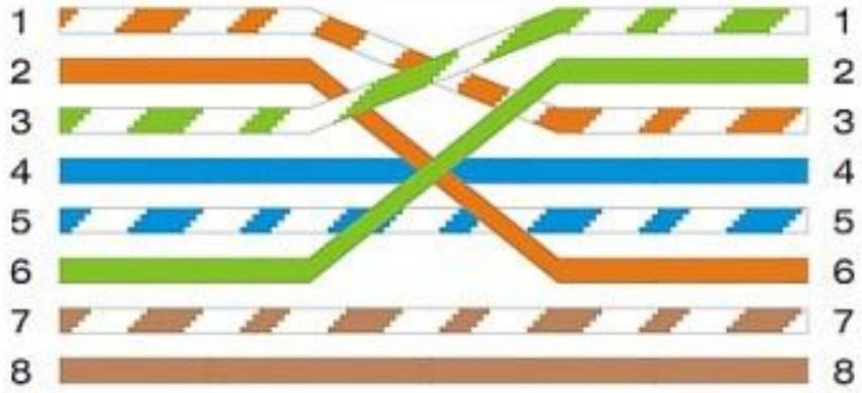

*სურ.1.4.1. 3*

სურათზე მოყვანილია RJ-45 კონექტორში ”ხვეული წყვილი” კაბელის ჩამაგრების ხერხები EIA/TIA 568A და EIA/TIA 568 B სტანდარტების შესაბამისად; ეს ოპერაცია სრულდება სპეციალური დასაწნეხი ინსტრუმენტით. (თუ გასართს განვალაგებთ კონტაქტებით ზემოთ და მივმართავთ ჩვენგან, მაშინ კონტაქტები უნდა დაინომროს მარცხნიდან მარჯვნივ 1-ნ 8-დე).

| კონტაქტი | მავთულის წნულის ფერი | |
|---|---|---|
| | **568A** | **568B** |
| 1 | თეთრი და მწვანე | თეთრი და ვარდისფერი |
| 2 | მწვანე | ვარდისფერი |
| 3 | თეთრი და ვარდისფერი | თეთრი და მწვანე |
| 4 | ცისფერი | ცისფერი |
| 5 | თეთრი და ცისფერი | თეთრი და ცისფერი |
| 6 | ვარდისფერი | მწვანე |
| 7 | თეთრი და ყავისფერი | თეთრი და ყავისფერი |
| 8 | ყავისფერი | ყავისფერი |

*სურ.1.4.1. 4*

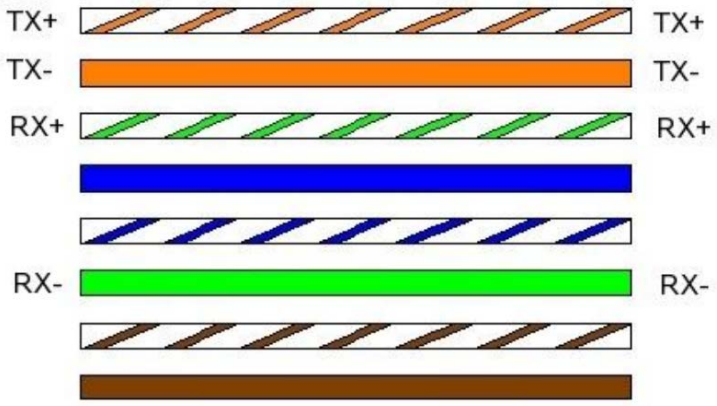
ერთი დონის მოწყობილობების (კომპიუტერი-კომპიუტერთან; კომუტატორი(Switch)-კომუტატორთან(Switch); მარშრუტიზატორი(Router) - მარშრუტიზატორთან(Router) პირდაპირი შეერთებისას გამოიყენება „Crossover” ჯვარედინი შეერთება. იხ. სურათი




*სურ.1.4.1. 5*

როგორც სურათზე ჩანს, 1,2,3 და 6 კონტაქტები კონტაქტები გადაჯვარედინებულია კაბელის თავსა და ბოლოში

სხვადასხვა დონის მოწყობილობების (კომპიუტერი-კომუტატორი(Switch); კომუტატორთან(Switch) - მარშრუტიზატორი(Router)) შეერთებისას გამოიყენება „Straight” პირდაპირი შეერთება. იხ. სურათი

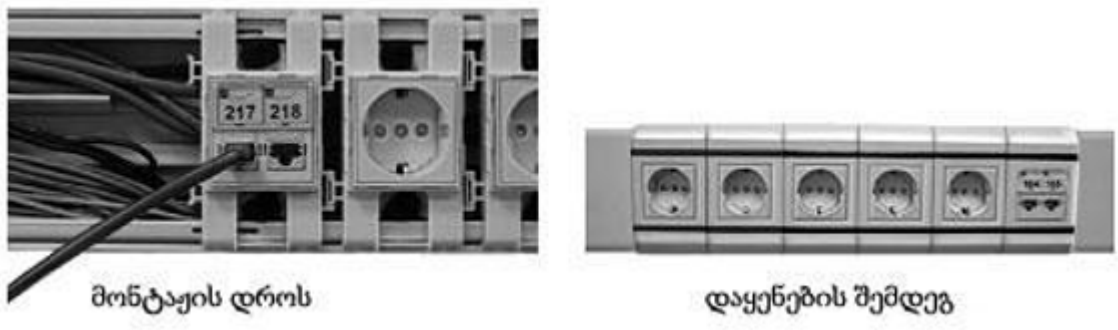



*სურ.1.4.1. 6*

ანუ კაბელის თავსა და ბოლოში სადენთა ფერების იდენტური განლაგებით.

შენობაში კაბელის ჩაგებისას, გაყვანილობას ჩვეულებრივ ჩაამაგრებენ კედელში, ათავსებენ სპეციალურ სივრცეებში შიგნით და შემდეგ გამოყავთ გარეთ კედლის ქსელური როზეტები.

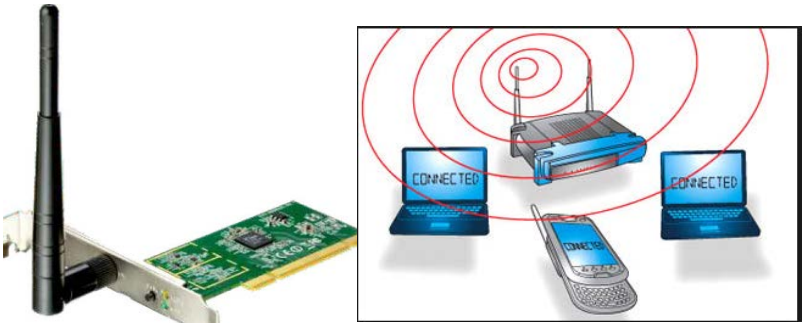
თუ კაბელის გაყვანა ვერ ხერხდება მითითებულ ადგილას, მაშინ იყენებენ კედლის (უფრო იშვიათად - იატაკის) კაბელ-არხებს (კოლოფებს). კოლოფი (box) - პლასტიკური, ჩვეულებრივად მართკუთხა, ასაწყობ-დასაშლელი ცარიელი მილი, რომელშიც გაჰყავთ ქსელური კაბელები, უფრო ხშირად ელექტრულთან ერთად.




*სურ.1.4.1. 7*

ქსელის კომპონენტების ფიზიკური შეერთების შემდგომ აუცილებელია ლოგიკური მისამართების შერჩევა. მისამართები როგორც უკვე ვიცით შესაძლებელია დავნიშნოთ სტატიკურად ან მოგვეწოდოს დინამიურად DHCP პროტოკოლით.

### *1.4.2. უკაბელო ქსელი*




*სურ.1.4.2. 1*

უკაბელო ქსელები, რომლებიც **802.11** (**Wi-Fi**) ოჯახის ერთ-ერთი სტანდარტით მუშაობენ, უფრო და უფრო ფართო გავრცელებას პოულობენ მოწყობილობის ხელმისაწვდომობის, მომართვის სიმარტივით და შემაერთებელი კაბელების არ არსებობის წყალობით. მაგრამ ამ ქსელებს აქვთ გარკვეული ნაკლოვანებებიც. მაგალითად, მონაცემთა გადაცემის დაბალი სიჩქარე, საკაბელო ქსელებთან შედარებით და მგრძნობელობა სხვადასხვა სახის შეფერხებებისა და წინაღობის დროს. ამ მიზეზით მხოლოდ პერსონალური კომპიუტერების არსებობისას უმჯობესია საკაბელო ქსელის შექმნა, რომელიც იმუშავებს სწრაფად და საიმედოდ.

პრაქტიკულად თითქმის ყველა თანამედროვე მობილურ კომპიუტერულ სისტემაში არის ჩაშენებული **Wi-Fi** ადაპტერი და მისი საკაბელო ქსელთან მიერთება ძალიან მოუხერხებელია. ამიტომ მათთვის უკაბელო ქსელი ხშირად წარმოადგენს ოპტიმალურ ვარიანტს. პერსონალური კომპიუტერის უკაბელო ქსელში ჩასართავად საჭიროა სისტემურ ბლოკში **Wi-Fi** ადაპტერის დაყენება გაფართოებული პლატის სახით ან/და გამოყენებულ იქნეს **USB**-პორტში ჩასართავი ადაპტერი ცალკე მოწყობილობის სახით.

უკაბელო ქსელის შექმნისას ასევე საჭიროა ერთ-ერთი შემდეგი მოწყობილობა:

- **წვდომის უკაბელო წერტილი (Wireless Access Point)** - გამოიყენება რამდენიმე კომპიუტერის უკაბელო ქსელში გასაერთიანებლად და უკაბელო ქსელის საკაბელოსთან

მისაერთებლად. იმისათვის რომ უკაბელო ქსელის ყველა მომხმარებელი შევიდეს ინტერნეტში, ქსელში ასევე უნდა არსებობდეს მოწყობილობა, რომელიც როუტერის ფუნქციას შეასრულებს (ეს შეიძლება იყოს **ADSL**- მოდემი).

- **უკაბელო როუტერი (მარშრუტიზატორი)** - წვდომის წერტილისაგან განსხვავებით, ქსელში აერთიანებს არა მარტო რამდენიმე უკაბელო მომხმარებელს, არამედ ასევე საშუალებას აძლევს, რომ მათ მიიღონ ინტერნეტი ერთი ჩქაროსნული შეერთებიდან.
- **უკაბელო ADSL-როუტერი** - ეს მოწყობილობა ითავსებს **ADSL**- მოდემისა და უკაბელო როუტერის ფუნქციებს. ასეთი მოწყობილობის შეძენა ხელსაყრელია ინტერნეტში საერთო წვდომის მქონე უკაბელო ქსელის შესაქმნელად ბინაში ან მცირე ოფისში.

თანამედროვე უკაბელო მოწყობილობები მომართულია ვებ-ინტერფეისის საშუალებით. ამისათვის საჭიროა მოწყობილობის კომპიუტერთან მიერთება საკაბელო ქსელის დახმარებით, **Internet Explorer**-ის სამისამართო სტრიქონში უნდა შევიტანოთ მოწყობილობის მისამართი, შემდეგ მივუთითოთ მომხმარებლის სახელი და პაროლი. ყველა ამ მონაცემის გაგება შესაძლებელია მოწყობილობაზე თანდართული დოკუმენტაციიდან, სადაც ასევე მოცემულია უკაბელო ქსელის დაყენების წესები თანმიმდევრობით. უკაბელო მოწყობილობის კონფიგურირებისათვის ასევე შესაძლებელია სპეციალური უტილიტების გამოყენება მოწყობილობაზე თანდართული კომპაქტ-დისკიდან.

უკაბელო მოწყობილობის მომართვის შემდეგ, შესაძლებელია კომპიუტერის ან ნოუთბუქის მიერთება შექმნილ ქსელთან. ამისათვის შევასრულოთ შემდეგი მოქმედებები:

1. **Control panel**-ის დათვალიერების არეში დავაწკაპუნოთ ქსელის ნიშანზე.

2. გამოსულ ფანჯარაში გამოჩნდება არსებული ადაპტერის რადიუსის ყველა ქსელის ჩამონათვალი.

3. დააწკაპუნეთ საჭირო ქსელის დასახელებაზე და დააჭირეთ ღილაკს შეერთება. იმისათვის რომ შემდეგში შეერთება მოხდეს ავტომატურად, თქვენ ასევე შეგიძლიათ დააყენოთ შესაბამისი ალამი მიერთების ღილაკის გვერდით.

4. შემდეგ ფანჯარაში აუცილებლობის შემთხვევაში შეიყვანეთ უსაფრთხოების გასაღები და დააჭირეთ ღილაკს **OK.** ეს გასაღები ჩვეულებრივ გამოდის ქსელის დაყენებისას წვდომის წერტილზე ან უკაბელო როუტერთან.

5. მას შემდეგ რაც, მოხდება მიერთება ახალ უკაბელო ქსელთან, გამოჩნდება ფანჯარა ქსელის განთავსების ასარჩევად.

*პრაქტიკული სამუშაო*

- გაამზადეთ TP კაბელები შესაბამისად Crossover და Straight შეერთებებისთვის
- დაუკავშირეთ კომპიუტერები კომუტატორს, მიანიჭეთ ლოგიკური მისამართები, შეამოწმეთ კავშირი
- დაუკავშირეთ კომპიუტერი მრავალფუნქციურ მოწყობილობას, შეცვალეთ მოწყობილობის ქსელური სახელი და პაროლი, დაუკავშირეთ უკაბელო ქსელის ადაპტერით აღჭურვილი მოწყობილობები მრავალფუნქციურ მოწყობილობას

*პრაქტიკული სამუშაო - პირდაპირი შეერთების (Straight-Through) და ჯვარედინი შეერთების (Crossover) UTP კაბელების აწყობა*

**შესავალი**

დაბეჭდეთ და შეავსეთ მოცემული ლაბორატორიული სამუშაო

ამ დავალებაში თქვენ უნდა ააწყოთ და შეამოწმოთ პირდაპირი (**Straight-Through**) და ჯვარედინი (**Crossover**) შეერთების არაეკრანირებული, გრეხილი წყვილი (**UTP**) **Ethernet** ქსელის კაბელი.

**შენიშვნა:** პირდაპირი შეერთების (**Straight-through**) კაბელის დროს, სადენის ფერი, რომელიც გამოყენებულია ერთი მხარის დაბოლოების პირველ კონტაქტზე არის იგივე ფერის, რომელიც გამოყენებულია მეორე მხარის დაბოლოების პირველ კონტაქტზე. ასევეა დანარჩენი შვიდი კონტაქტიც. კაბელი შეიძლება შექმნილი იყოს **TIA/ EIA T568A** ან **T568B Ethernet** სტანდარტის გამოყენებით, რომელიც განსაზღვრავს სადენის ფერს, რაც გამოყენებულია თითოეულ შესასვლელში. პირდაპირი შეერთების (**Straight-Through**) კაბელები როგორც წესი გამოიყენება ჰოსტის უშუალოდ დასაკავშირებლად კონცენტრატორთან (**Hub**), კომუტატორთან (**Switch**) ან კედელზე დასამაგრებელ აუთლეტთან ოფისში.

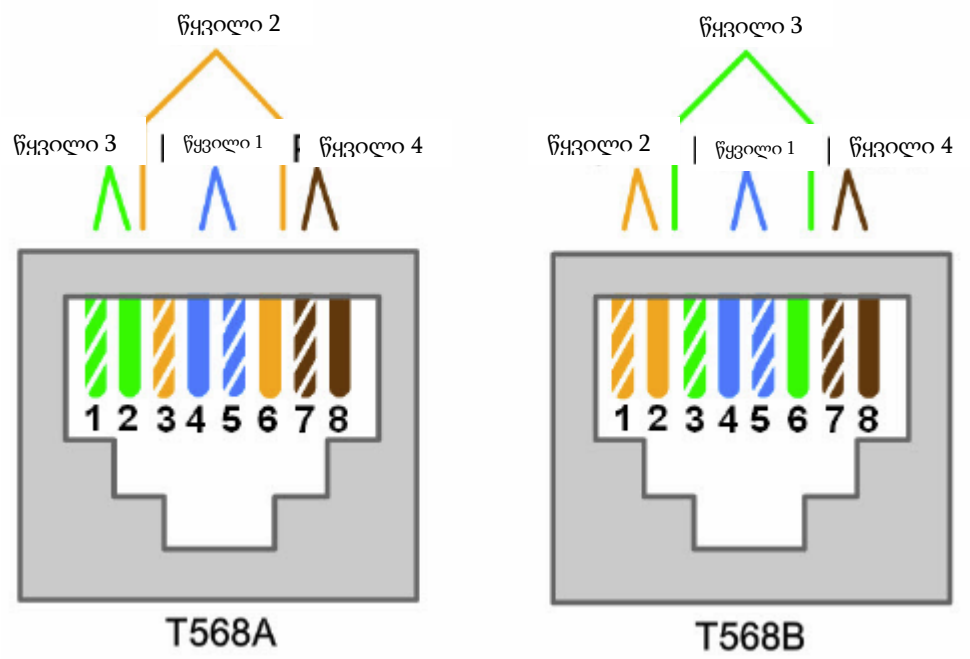
ჯვარედინი შეერთების (**Crossower**) კაბელის დროს, მეორე და მესამე წყვილები კაბელის ერთი ბოლოს **RJ-45** კონექტორზე არის საპირისპირო კაბელის მეორე ბოლოზე. კაბელის ერთი მხარის კონტაქტები არის **T568A** სტანდარტის, ხოლო მეორე მხარეს - **T568B** სტანდარტი. ჯვარედინი შეერთების კაბელი (**Crossower**) როგორც წესი გამოიყენება კონცენტრატორების (**Hub**) და კომუტატორების (**Switch**) დასაკავშირებლად ან ორი კომპიუტერის პირდაპირ შესაერთებლად, მარტივი ქსელის შექმნისთვის.

**რეკომენდებული მოწყობილობები:**

- ორი 0.6-დან 0.9 მეტრამდე სიგრძის 5 ან 5e კატეგორიის (**Cat5**, **Cat5e**) კაბელი
- მინიმუმ ოთხი ცალი **RJ-45** კონექტორი (მეტი შეიძლება საჭირო გახდეს არასწორად დამზადების შემთხვევაში)

- **RJ-45 Crimping tool**
- ორი კომპიუტერი, რომელზეც ინსტალირებულია **Windows 8.1** ან **7** სისტემა
- სადენის საჭრელი
- კაბელის გარსის შემოსაცლელი (**Wire stripper**)

## კაბელის სადენების დიაგრამა




*სურ.5.2. 1*

| T568A სტანდარტი | | | |
|---|---|---|---|
| კონტაქტის ნომერი | წყვილის ნომერი | სადენის ფერი | ფუნქცია |
| 1 | 3 | თეთრი/მწვანე | გადაცემა |
| 2 | 3 | მწვანე | გადაცემა |
| 3 | 2 | თეთრი/ნარინჯისფერი | მიღება |
| 4 | 1 | ლურჯი | არ გამოიყენება |
| 5 | 1 | თეთრი/ლურჯი | არ გამოიყენება |
| 6 | 2 | ნარინჯისფერი | მიღება |
| 7 | 4 | თეთრი/ყავისფერი | არ გამოიყენება |
| 8 | 4 | ყავისფერი | არ გამოიყენება |

| T568B სტანდარტი | | | |
|---|---|---|---|
| კონტაქტის ნომერი | წყვილის ნომერი | სადენის ფერი | ფუნქცია |
| 1 | 2 | თეთრი/ნარინჯისფერი | გადაცემა |
| 2 | 2 | ნარინჯისფერი | გადაცემა |
| 3 | 3 | თეთრი/მწვანე | მიღება |
| 4 | 1 | ლურჯი | არ გამოიყენება |
| 5 | 1 | თეთრი/ლურჯი | არ გამოიყენება |
| 6 | 3 | მწვანე | მიღება |
| 7 | 4 | თეთრი/ყავისფერი | არ გამოიყენება |
| 8 | 4 | ყავისფერი | არ გამოიყენება |

**პირდაპირი შეერთების (Straight-through) კაბელის აწყობა და შემოწმება**

**პირველი ეტაპი: კაბელის მიღება და მომზადება**

ა. განსაზღვრეთ საჭირო კაბელის სიგრძე. ეს შეიძლება იყოს მანძილი კომპიუტერიდან კომუტატორამდე ან მოწყობილობასა და **RJ-45** აუთლეტის ბუდეს შორის. დაამატეთ მინიმუმ 30.48 სმ (12 ინჩი) მანძილი. **TIA/EIA** სტანდარტით მაქსიმუმი სიგრძე არის 5 მეტრი (16.4 ფუტი). სტანდარტული **Ethernet** კაბელის სიგრძეები, როგორც წესი არის: 6 მ (2 ფუტი), 1.83 მ (6 ფუტი) ან 3.05 მეტრი (10 ფუტი).

ბ. რა სიგრძის კაბელი აირჩიეთ და რატომ აირჩიეთ ამ სიგრძის კაბელი?

_____

გ. მოჭერით სასურველი სიგრძის კაბელი. სტანდარტული **UTP** კაბელი გამოიყენება პატჩ კაბელებისთვის (კაბელები საბოლოო ქსელურ მოწყობილობასა, კომპიუტერის ჩათვლით, და **RJ-45** კონექტორს შორის) იმიტომ რომ ის არის მეტად გამძლე მრავალჯერადი მოკეცვის დროს. სტანდარტული ჰქვია იმიტომ რომ კაბელის თითოეული სადენი შედგება წვრილი სპილენძის გამტარის ბევრი ბოჭკოსაგან, ერთი მყარი გამტარის ნაცვლად. მყარი სადენი გამოიყენება კაბელური ტრასისთვის, **RJ-45** ბუდესა და **Punch-down** ბლოკს შორის.

დ. სადენის გარსის შემოსაცლელის გამოყენებით, მოაცილეთ 5.08 სმ (2 ინჩი) კაბელის გარსი კაბელის ორივე მხარის ბოლოში.

**მეორე ეტაპი. სადენების მომზადება და ჩასმა**

ა. განსაზღვრეთ კაბელის რომელი სტანდარტი იქნება გამოყენებული. შემოხაზეთ სტანდარტი: [**T568A | T568B**]

ბ. მოძებნეთ სწორი ცხრილი ან სურათი „კაბელების დიაგრამიდან“, რომელიც დაფუძნებულია გამოყენებული კაბელების სტანდარტზე.

გ. გააცალკევეთ კაბელის წყვილები და დაალაგეთ ისინი შერჩეული სტანდარტის შესაბამისად.

დ. დაშალეთ მოკლე სიგრძის ხვეული წყვილები და დაალაგეთ ისინი იმ მიმდევრობით, რომელსაც სტანდარტი მოითხოვს. დალაგება დაიწყეთ მარცხნიდან მარჯვნივ პირველი კონტაქტიდან. მნიშვნელოვანია ხვეული წყვილების ძალიან მცირედით დაცალკევება. **წყვილები არის აუცილებელი იმიტომ, რომ ისინი უზრუნველყოფენ ხმაურის გაუქმებას.**

ე. გაასწორეთ და გაათანაბრეთ წვერები ცერა და საჩვენებელ თითებს შორის.

ვ. დარწმუნდით რომ კაბელის წვერები არის სწორი მიმდევრობით დალაგებული, როგორც სტანდარტი მოითხოვს.

ზ. მოჭერით კაბელი სწორხაზოვნად კაბელის გარსიდან დაახლოებით 1.25-დან 1.9 სანტიმეტრამდე (1/2-დან ¾ ინჩი). თუ კაბელის გარსი უფრო მოკლეა ვიდრე წვერები, კაბელი იქნება მგრძნობიარე ხელის შემშლელ ხმაურთან მიმართებაში (ერთი წვერის ბიტების მიერ ხელის შეშლა მეზობელი წვერისთვის).

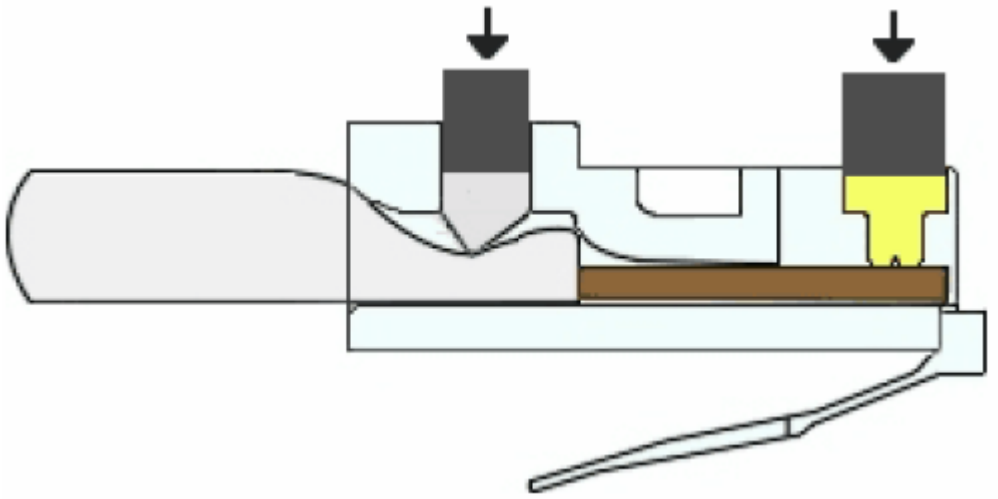
თ. გასაღები (კბილი, რომელიც მიწებებულია **RJ-45** კონექტორზე) უნდა იყოს ქვედა მხრისკენ მიმართული, როდესაც ხდება სადენების ჩასმა კონექტორში. დარწმუნდით რომ წვერები არის დალაგებული მარცხნიდან მარჯვნივ, დაწყებული პირველი კონექტორიდან. ჩასვით წვერები **RJ-45** კონექტორში, სანამ ყველა წვერი შეძლებისდაგვარად ბოლომდე არ მივა კონექტორის ბოლოში.

**მესამე ეტაპი: დათვალიერება, დაჭეკვა და შემოწმება**

ა. ვიზუალურად დაათვალიერეთ კაბელი და დარწმუნდით რომ სწორი ფერთა კოდებია შეერთებული სწორ კონტაქტების ნომრებთან.

ბ. ვიზუალურად შეამოწმეთ კონექტორის ბოლო. რვავე წვერი უნდა იყოს მყარად დაჭერილი **RJ-45** კონექტორის ბოლოში. კაბელის გარსის ნაწილი შეიძლება მოექცეს კონექტორის პირველ სექციაში, რაც უზრუნველყოფს კაბელის დაწოლისაგან გათავისუფლებას. თუ კაბელის გარსი არ არის საკმარისად შესული კონექტორში, ამან შეიძლება გამოიწვიოს კაბელის დაზიანება.

გ. თუ ყველაფერი სწორადაა დალაგებული და ჩასმული, მოათავსეთ **RJ-45** კონექტორი და კაბელი დასაჭეკ მოწყობილობაში (**Crimper**). დასაჭეკი მოწყობილობა დაუშვებს ქვემოთ ორ დგუშს **RJ-45** კონექტორზე.



დ. ვიზუალურად დაათვალიერეთ კონექტორი. თუ არასწორადაა დამზადებული, მოაჭერით ბოლო და გაიმეორეთ პროცესი.

**მეოთხე ეტაპი: კაბელის მეორე მხარის დასრულება**

ა. გამოიყენეთ ზემოთ აღწერილი ეტაპები **RJ-45** კონექტორის დასაკავშირებლად კაბელის მეორე მხარეზე.

ბ. ვიზუალურად გადაამოწმეთ კონექტორი. თუ არასწორადაა გაკეთებული მოაჭერით ბოლო და გაიმეორეთ პროცესი.

გ. **Patch** კაბელის რომელი სტანდარტია [**T568A|T568B**] გამოყენებული თქვენს სასწავლებელში?

_____

**მეხუთე ეტაპი: კაბელის ტესტირება**

ა. გამოიყენეთ კაბელი პერსონალური კომპიუტერის ქსელთან დასაკავშირებლად.

ბ. ვიზუალურად შეამოწმეთ ქსელის ადაპტერის **LED** ნათურების მდგომარეობა. თუ ისინი ჩართულია (როგორც წესი მწვანე ან ქარვისფერი) ე.ი კაბელი არის ფუნქციური.

გ. პერსონალურ კომპიუტერზე გახსენით ბრძანებათა სტრიქონი

დ. აკრიფეთ **ipconfig** ბრძანება

ე. ქვემოთ ჩაწერეთ ნაგულისხმევი გასასვლელი **IP** მისამართი.

_____

ვ. ბრძანებათა ველში აკრიფეთ **Ping <ნაგულისხმევი გასასვლელი IP მისამართი>**. თუ კაბელი ფუნქციურია, **Ping**-ი უნდა გავიდეს წარმატებულად (იმ პირობით, რომ სხვა ქსელური პრობლემა არ არსებობს და ნაგულისხმევი გასასვლელი მარშრუტიზატორი დაკავშირებული და ფუნქციური).

ზ. დასრულდა **Ping**-ი წარმატებით?

_____

თ. თუ **Ping**-ი ჩავარდა, გაიმეორეთ ლაბორატორიული სამუშაო.

## Ethernet ჯვარედინი შეერთების კაბელის აწყობა და ტესტირება

### პირველი ეტაპი: კაბელის მიღება და მომზადება

ა. განსაზღვრეთ მოთხოვნილი კაბელის სიგრძე. ეს შეიძლება იყოს კონცენტრატორიდან (**Hub**) კონცენტრატორამდე, კონცენტრატორიდან კომუტატორამდე (**Switch**), კომუტატორიდან კომუტატორამდე, კომპიუტერიდან მარშრუტიზატორამდე (**Router**), ან ერთი კომპიუტერიდან სხვა კომპიუტერამდე. დაამატეთ დაახლოებით 30.48 სანტიმეტრი (12 ინჩი) მანძილი. რა სიგრძის კაბელი აირჩიეთ და რატომ აირჩიეთ ეს სიგრძე?

_____

_____

ბ. მოჭერით სასურველი სიგრძის კაბელის ნაჭერი და წვერების გასასუფთავებელი მოწყობილობით (**Wire strippers**) მოაცალეთ 5.08 სანტიმეტრი (2 ინჩი) კაბელის გარსი, კაბელის ორივე დაბოლოვებაზე.

### მეორე ეტაპი : T568A წყვილების მომზადება და ჩასმა

ა. იპოვეთ **T568A** ცხრილი ლაბორატორიული სამუშაოს დასაწყისში

ბ. გააცალკევეთ კაბელის წყვილები და დაალაგეთ ისინი შერჩეული **T568A** სტანდარტის შესაბამისად.

გ. გააცალკევეთ მოკლე სიგრძის ხვეული წყვილები და დაალაგეთ ისინი იმ მიმდევრობით, რომელსაც სტანდარტი მოითხოვს. დალაგება დაიწყეთ მარცხნიდან მარჯვნივ პირველი კონტაქტიდან. მნიშვნელოვანია ხვეული წყვილების ძალიან მცირედით დაშორება ერთმანეთისგან. წყვილები არის აუცილებელი იმიტომ, რომ ისინი უზრუნველყოფენ ხმაურის გაუქმებას.

დ. გაასწორეთ და გაათანაბრეთ წვერები ცერა და საჩვენებელ თითებს შორის.

ე. დარწმუნდით რომ კაბელის წვერები არის სწორი მიმდევრობით დალაგებული, როგორც სტანდარტი მოითხოვს.

ვ. მოჭერით კაბელი სწორხაზოვნად კაბელის გარსიდან დაახლოებით 1.25-დან 1.9 სანტიმეტრამდე (1/2-დან ¾ ინჩი). თუ კაბელის გარსი უფრო მოკლეა ვიდრე წვერები, კაბელი იქნება მგრძნობიარე ხელის შემშლელ ხმაურთან მიმართებაში (ერთი წვერის ბიტების მიერ ხელის შეშლა მეზობელი წვერისთვის).

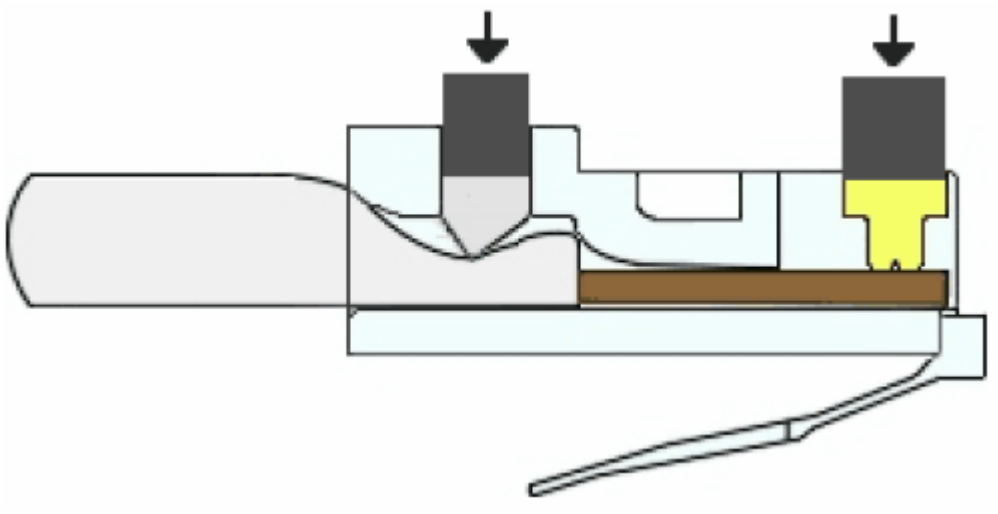
ზ. გასაღები (კბილი, რომელიც მიწებებულია **RJ-45** კონექტორზე) უნდა იყოს ქვედა მხრისკენ მიმართული, როდესაც ხდება სადენების ჩასმა კონექტორში. დარწმუნდით რომ წვერები არის დალაგებული მარცხნიდან მარჯვნივ, დაწყებული პირველი კონექტორიდან. ჩასვით წვერები **RJ-45** კონექტორში, სანამ ყველა წვერი შეძლებისდაგვარად ბოლომდე არ მივა კონექტორის ბოლოში.

**მესამე ეტაპი: დათვალიერება, დაჭეკვა და შემოწმება**

ა. ვიზუალურად დაათვალიერეთ კაბელი და დარწმუნდით რომ სწორი ფერთა კოდებია შეერთებული სწორ კონტაქტების ნომრებთან.

ბ. ვიზუალურად შეამოწმეთ კონექტორის ბოლო. რვავე წვერი უნდა იყოს მყარად დაჭერილი **RJ-45** კონექტორის ბოლოში. კაბელის გარსის ნაწილი შეიძლება მოექცეს კონექტორის პირველ სექციაში, რაც უზრუნველყოფს კაბელის დაწოლისაგან გათავისუფლებას. თუ კაბელის გარსი არ არის საკმარისად შესული კონექტორში, ამან შეიძლება გამოიწვიოს კაბელის დაზიანება.

გ. თუ ყველაფერი სწორადაა დალაგებული და ჩასმული, მოათავსეთ **RJ-45** კონექტორი და კაბელი დასაჭეკ მოწყობილობაში (**Crimper**). დასაჭეკი მოწყობილობა დაუშვებს ქვემოთ ორ დგუშს **RJ-45** კონექტორზე.



დ. ვიზუალურად დაათვალიერეთ კონექტორი. თუ არასწორადაა დამზადებული, მოაჭერით ბოლო და გაიმეორეთ პროცესი.

**მეოთხე ეტაპი: T568B კაბელის მეორე მხარის დასრულება**

ა. გამოიყენეთ ზემოთ აღწერილი ეტაპები (მაგრამ გამოიყენეთ **T568B** ცხრილი და სტანდარტი) **RJ-45** კონექტორის დასაკავშირებლად კაბელის მეორე მხარეზე.

ბ. ვიზუალურად გადაამოწმეთ კონექტორი. თუ არასწორადაა გაკეთებული მოაჭერით ბოლო და გაიმეორეთ პროცესი.

გ. **Patch** კაბელის რომელი სტანდარტს [**T568A|T568B**] იყენებთ სახლის პირობებში, თუ თქვენ გაქვთ ან გინდათ გქონდეთ საშინაო ქსელი?

_____

**მეხუთე ეტაპი: კაბელის ტესტირება**

ა. გამოიყენეთ კაბელი პერსონალური კომპიუტერის ქსელთან დასაკავშირებლად.

ბ. ვიზუალურად შეამოწმეთ ქსელის ადაპტერის **LED** ნათურების მდგომარეობა. თუ ისინი ჩართულია (როგორც წესი მწვანე ან ქარვისფერი) ე.ი კაბელი არის ფუნქციური.

გ. ორივე პერსონალურ კომპიუტერზე გახსენით ბრძანებათა სტრიქონი

დ. ორივე კომპიუტერზე აკრიფეთ **ipconfig** ბრძანება

ე. ქვემოთ ჩაწერეთ ორივე კომპიუტერის **IP** მისამართი.

კომპიუტერი №1_____

კომპიუტერი №2_____

ვ. ერთ-ერთი კომპიუტერის ბრძანებათა ველში აკრიფეთ **Ping < მეორე კომპიუტერის IP მისამართი>**. თუ კაბელი ფუნქციურია, **Ping**-ი უნდა გავიდეს წარმატებულად. შეასრულეთ **Ping** ბრძანება ახლა უკვე მეორე კომპიუტერიდან.

ზ. დასრულდა **Ping**-ი წარმატებით?

_____

თ. თუ **Ping**-ი ჩავარდა, გაიმეორეთ ლაბორატორიული სამუშაო.

პრაქტიკული სამუშაო - ფიზიკური ტოპოლოგიები

შესწავლის მიზანი

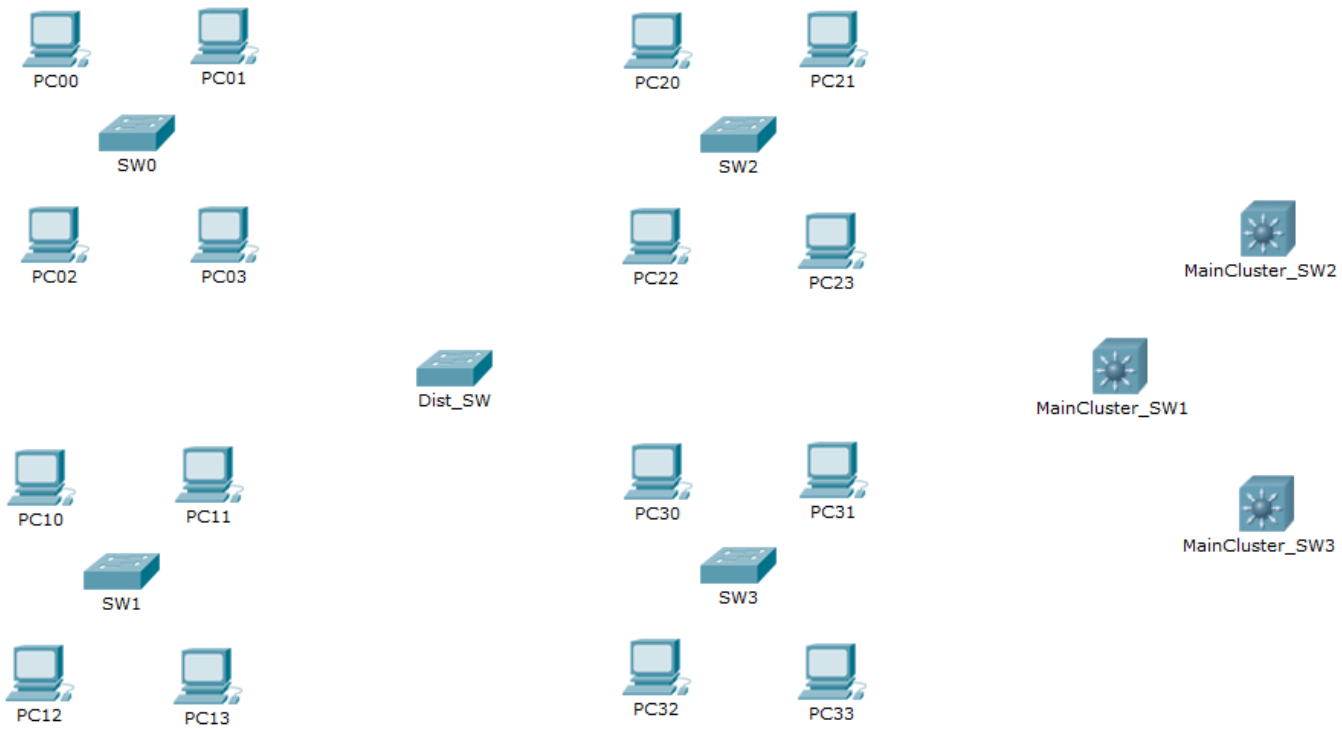
გაეცნოს სტუდენტი ვარსკვლავურ (**Star**), გაფართოებულ ვარსკვლავურ (**Extended Star**) და სრულკავშირიან/ბადისებრ (**Mesh**) ტოპოლოგიებში

**შესავალი**

მოცემულ დავალებაში თქვენ უნდა ააწყოთ რამდენიმე განსხვავებული ფიზიკური ტოპოლოგია, ქვემოთ ნაჩვენები მოწყობილობების გამოყენებით. ამ დავალებაში შესწავლილია შემდეგი ტოპოლოგიები:

- ვარსკვლავი (**Star**)
- გაფართოებული ვარსკვლავი (**Extended Star**)
- სრულკავშირიანი/ბადისებრი (**Mesh**)

მას შემდეგ რაც, მოწყობილობები იქნებიან დაკავშირებულნი მითითებული ფიზიკური ტოპოლოგიის მიხედვით, თქვენ მოგიწევთ ტოპოლოგიების ურთიერთდაკავშირება.



**დავალება №1: მოწყობილობის შეერთება ფიზიკურ Star (ვარსკვლავი) ტოპოლოგიაში**

**პირველი ეტაპი: პირველი ვარსკვლავი ტოპოლოგიის მოწყობილობების დაკავშირება**

ა) მოძებნეთ შემდეგი მოწყობილობები: **PC00, PC01, PC02, PC03** და **SW0**. ისინი მოთავსებულნი არიან **Packet Tracer**-ის სამუშაო სივრცის მარცხენა ზედა კუთხეში. ეს მოწყობილობები ჩაერთვებიან ვარსკვლავურ ტოპოლოგიაში.

ბ) **Connections** მენიუში აირჩიეთ კაბელის ტიპი **Copper Straight-Through**.

შენიშვნა: რამდენიმე შეერთების დასამატებლად, კაბელის ტიპის არჩევისას დააჭირეთ **Ctrl** ღილაკს.

გ) დააკავშირეთ მოცემული კომპიუტერები **SW0** სვიჩთან. შეაერთეთ მოწყობილობები **SW0** სვიჩთან შემდეგნაირად: **PC00** კომპიუტერი **Fast-Ethernet0/1** პორტთან, **PC01** კომპიუტერი **Fast-Ethernet0/2** პორტთან, **PC02** კომპიუტერი **Fast-Ethernet0/3** პორტთან, **PC03** კომპიუტერი **Fast-Ethernet0/4** პორტთან.

**დავალება №2: სხვა ვარსკვლავი ტოპოლოგიის შექმნა**

**პირველი ეტაპი: მეორე ვარსკვლავი ტოპოლოგიის მოწყობილობების დაკავშირება**

ა) მოძებნეთ შემდეგი მოწყობილობები: **PC10**, **PC11**, **PC12**, **PC13** და **SW1**. ისინი მოთავსებულნი არიან **Packet Tracer**-ის სამუშაო სივრცის მარცხენა ქვედა კუთხეში.

ბ) პირველი დავალების მსგავსად დააკავშირეთ კომპიუტერები მეორე ვარსკვლავ ტოპოლოგიაში. ამისათვის **Connections** მენიუში აირჩიეთ **Copper Straight-Through** კაბელის ტიპი.

გ) დააკავშირეთ მოცემული კომპიუტერები **SW1** სვიჩთან. შეაერთეთ მოწყობილობები **SW1** სვიჩთან შემდეგნაირად: **PC10** კომპიუტერი **Fast-Ethernet0/1** პორტთან, **PC11** კომპიუტერი **Fast-Ethernet0/2** პორტთან, **PC12** კომპიუტერი **Fast-Ethernet0/3** პორტთან, **PC13** კომპიუტერი **Fast-Ethernet0/4** პორტთან.

**მეორე ეტაპი: მესამე ვარსკვლავი ტოპოლოგიის მოწყობილობების დაკავშირება**

ა) მოძებნეთ შემდეგი მოწყობილობები: **PC20**, **PC21**, **PC22**, **PC23** და **SW2**. ისინი მოთავსებულნი არიან **Packet Tracer**-ის სამუშაო სივრცის ზედა ცენტრალურ ნაწილში.

ბ) ჩართეთ მოწყობილობები მესამე ვარსკვლავში. ამისათვის **Connections** მენიუში აირჩიეთ **Copper Straight-Through** კაბელის ტიპი.

გ) დააკავშირეთ კომპიუტერები **SW2** სვიჩთან: **PC20** კომპიუტერი **Fast-Ethernet0/1** პორტთან, **PC21** კომპიუტერი **Fast-Ethernet0/2** პორტთან, **PC22** კომპიუტერი **Fast-Ethernet0/3** პორტთან და**PC23** კომპიუტერი **Fast-Ethernet0/4** პორტთან.

**მესამე ეტაპი: მეოთხე ვარსკვლავი ტოპოლოგიის მოწყობილობების დაკავშირება**

ა) მოძებნეთ შემდეგი მოწყობილობები: **PC30**, **PC31**, **PC32**, **PC33** და **SW3**. ისინი მოთავსებულნი არიან **Packet Tracer**-ის სამუშაო სივრცის ქვედა ცენტრალურ ნაწილში.

ბ) ჩართეთ მოწყობილობები მეოთხე ვარსკვლავ ტოპოლოგიაში. ამისათვის **Connections** მენიუში აირჩიეთ **Copper Straight-Through** კაბელის ტიპი.

გ) დააკავშირეთ კომპიუტერები **SW3** სვიჩთან: **PC30** კომპიუტერი **Fast-Ethernet0/1** პორტთან, **PC31** კომპიუტერი **Fast-Ethernet0/2** პორტთან, **PC32** კომპიუტერი **Fast-Ethernet0/3** პორტთან და **PC33** კომპიუტერი **Fast-Ethernet0/4** პორტთან.

**დავალება №3: გაფართოებული ვარსკვლავი ტოპოლოგიის შექმნა**

ა) მოძებნეთ შემდეგი მოწყობილობები: **SW0**, **SW1**, **SW2**, **SW3** და **Dist_SW**

ბ) **Connections** მენიუში აირჩიეთ **Copper Cross-Over** კაბელის ტიპი.

გ) დააკავშირეთ **SW0**, **SW1**, **SW2**, **SW3** და **Dist_SW** მოწყობილობები ქვემოთ მოცემული ცხრილის შესაბამისად:

| მოწყობილობა | სვიჩის პორტი | Dist_SW მოწყობილობის პორტი |
|---|---|---|
| SW0 | Fast-Ethernet0/24 | Fast-Ethernet0/10 |
| SW1 | Fast-Ethernet0/24 | Fast-Ethernet0/11 |

| | | |
|---|---|---|
| SW2 | Fast-Ethernet0/24 | Fast-Ethernet0/12 |
| SW3 | Fast-Ethernet0/24 | Fast-Ethernet0/13 |

დ. საბოლოოდ მივიღებთ გაფართოებულ ვარსკვლავ ტოპოლოგიას, სადაც ოთხი პატარა ვარსკვლავი მონაწილეობს.

**დავალება №4: სრულკავშირიანი (ბადისებრი) ტოპოლოგიის შექმნა**

**პირველი ეტაპი: მაგისტრალური კომუტატორების (Switch) დაკავშირება**

ა) მოძებნეთ მოწყობილობები: **MainCluster_SW1, MainCluster_SW2** და **MainCluster_SW3**. ეს მოწყობილობები მოთავსებულნი არიან **Packet Tracer** პროგრამის სამუშაო არის მარჯვენა ნაწილში.

ბ) **Connections** მენიუში აირჩიეთ **Copper Cross-Over** კაბელის ტიპი.

გ) დააკავშირეთ მაგისტრალური მოწყობილობები ქვემოთ მოცემული სქემის მიხედვით:

| საწყისი მოწყობილობა | საწყისი პორტი | დანიშნულების მოწყობილობა | დანიშნულების პორტი |
|---|---|---|---|
| MainCluster_SW1 | GigabitEthernet0/1 | MainCluster_SW2 | GigabitEthernet0/1 |
| MainCluster_SW1 | GigabitEthernet0/2 | MainCluster_SW3 | GigabitEthernet0/1 |
| MainCluster_SW2 | GigabitEthernet0/2 | MainCluster_SW3 | GigabitEthernet0/2 |

დ) ახლა რადგან ყველა **MainCluster** კომუტატორები დაკავშირებულნი არიან ერთმანეთთან, მათ შორის შეიქმნა სრულკავშირიანი ტოპოლოგია.

**მეორე ეტაპი: ჰიბრიდული ტოპოლოგიის შექმნა**

ა) **Connections** მენიუში აირჩიეთ **Copper Cross-Over** კაბელის ტიპი.

ბ) **MainCluster_SW1** მოწყობილობის **Fast-Ethernet0/24** პორტი დააკავშირეთ **Dist_SW** მოწყობილობის **Fast-Ethernet0/24** პორტთან. სრულკავშირიანი ტოპოლოგიისა და

„გაფართოებული ვარსკვლავ“-ის დაკავშირების შედეგად მივიღეთ ჰიბრიდული ტოპოლოგია.

**დავალება №5: საკითხის დასმა**

**პირველი ეტაპი. დაზიანებული წერტილების ანალიზი (და წარმადობის გაზრდა)**

ა. რამდენი დაზიანებული წერტილი შენიშნეთ? _____

ბ. როგორ შეგიძლიათ დაზიანებული წერტილების რაოდენობის შემცირება?

_____

_____

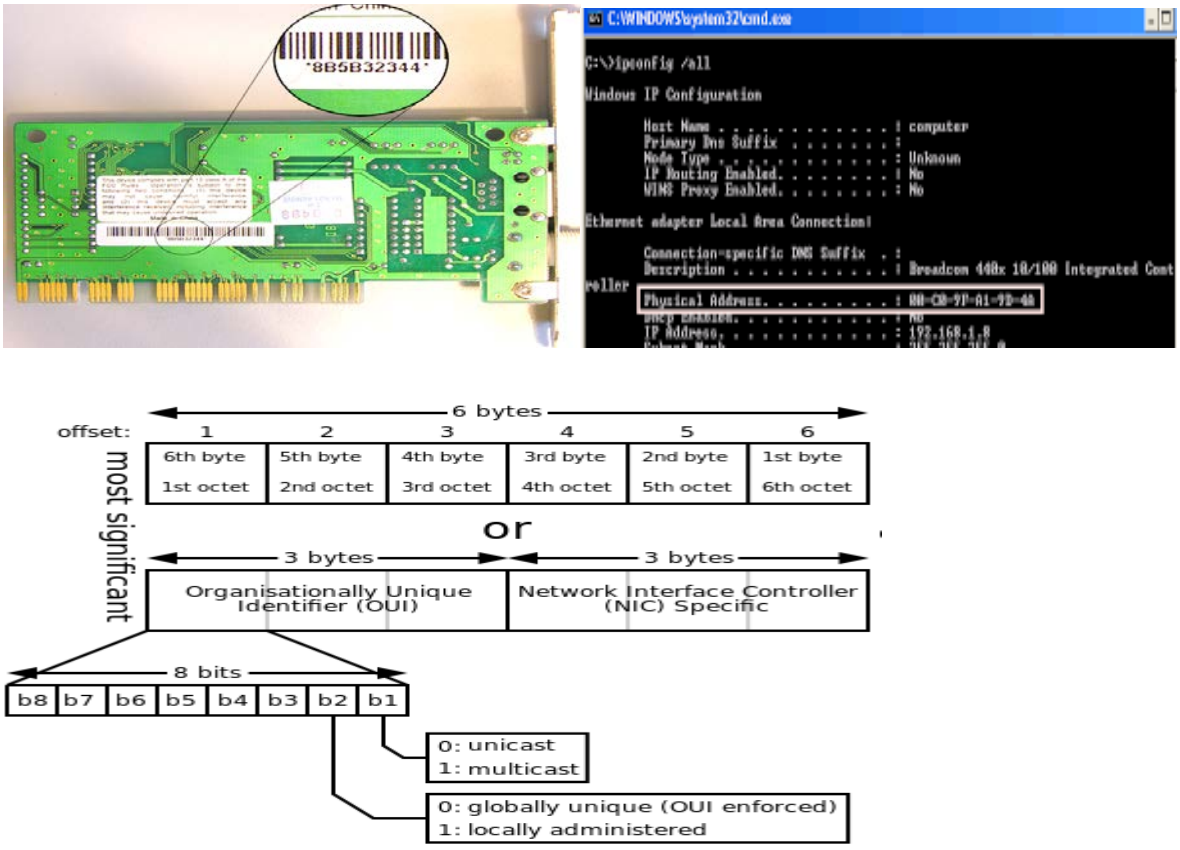
_____

## 1.5. ფიზიკური და ლოგიკური მისამართების განსხავება და მათი გამოყენება ქსელის გამართვისთვის

ნებისმიერი ორი სისტემის საკომუნიკაციოდ, ეს სისტემები უნდა იყოს იდენტიფიცირებული ქსელში. იდენტიფიცირებისთვის გამოიყენება ლოგიკური(IP) და ფიზიკური(MAC) მისამართები.

### *1.5.1. ფიზიკური(MAC) მისამართი*

ყველა კომპიუტერს გააჩნია უნიკალური ფიზიკური მისამართი, რომელიც ცნობილია როგორც MAC მისამართი. ეს მისამართი ენიჭება მწარმოებელი ფირმის მიერ ქსელურ ადაპტერს.

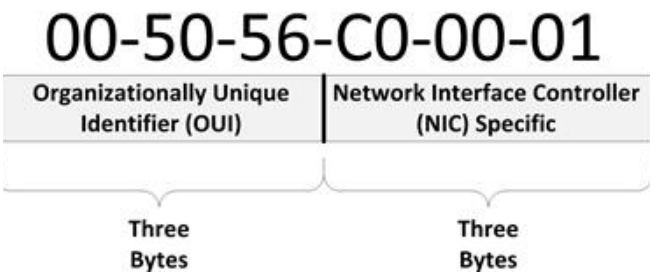



*სურ.1.5.1. 1*

Ethernet MAC მისამართი ჩვეულებრივ გამოისახება 16-ით ფორმატში და შედგება 2 ნაწილისგან:

– Organizationally Unique Identifier (OUI) – OUI არის 24-ბიტიანი ( 6 თექვსმეტობითი სიმბოლო) მწარმოებლის კოდი, მინიჭებული IEEE სტანდარტიზაციის ორგანოს მიერ.

– Device Identifier – მოწყობილობის იდენტიფიკატორი არის უნიკალური 24-ბიტიანი (6 თექვსმეტობითი სიმბოლო) მნიშვნელობა, რომელიც მიენიჭება უშუალოდ მწარმოებლის მიერ და OUI-სთან ერთად ქმნის საერთო უნიკალურ მისამართს.




*სურ.1.5.1. 2*

### *1.5.2. ლოგიკური მისამართები*

კვანძს(Host) ქსელში(მათ შორის ინტერნეტის ქსელში) იდენტიფიცირებისათვის უნდა ჰქონდეს შემდეგი ლოგიკური მისამართები:

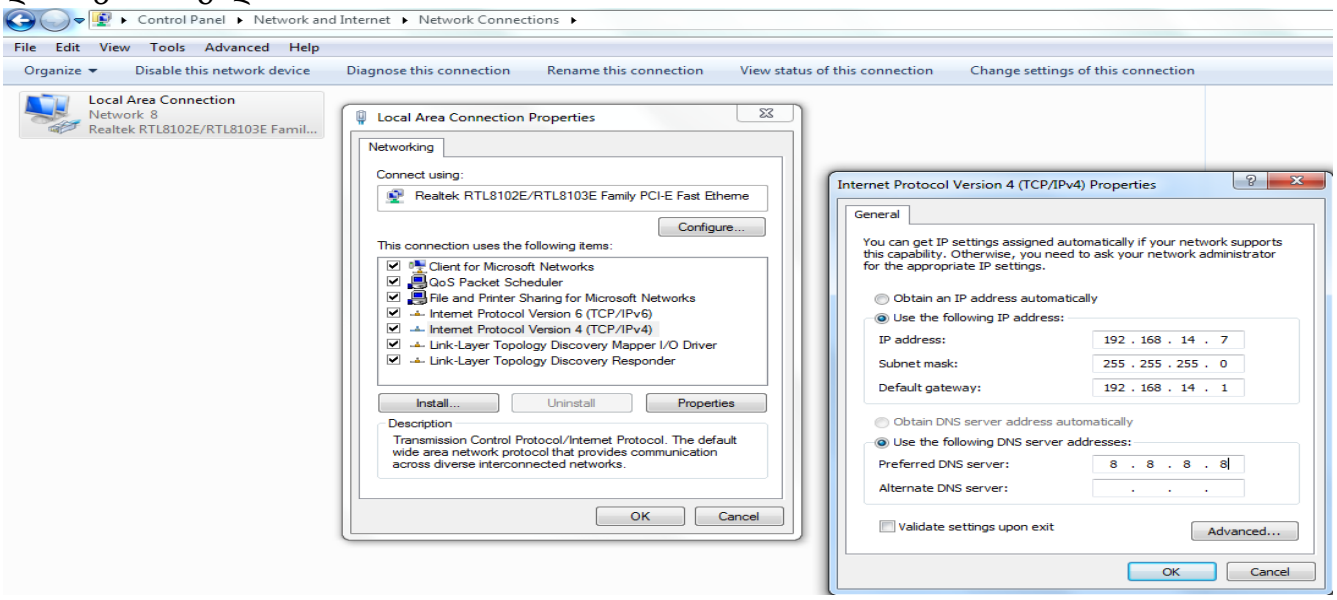
- **IP address** - აი-პი მისამართი , უშუალოდ კვანძის უნიკალური მისამართი
- **Subnet mask** - ქვექსელის ნიღაბი , განსაზღვრავს ქსელის ზომას
- **Default Gateway** – ”შლუზი”(კარიბჭე სხვა ქსელებში), იმ მოწყობილობის მისამართი, რომელიც მოცემულ ქსელს(რომელშიც კონკრეტული კვანძია ჩართული) აკავშირებს სხვა ქსელებთან(ჩვეულებრივ ასეთი მოწყობილობაა მარშრუტიზატორი(Router)
- **Domain Name System Server Address**- დომენური სახელების სისტემის სერვერის მისამართი.

არსებობს **IP**-მისამართების მინიჭების ორი ხერხი:

- **სტატიკური IP-დამისამართება**. ამ შემთხვევაში ქსელის ყოველი კომპიუტერისათვის IP-მისამართის, ქვექსელის ნიღბისა და TCP/IP პროტოკოლის სხვა პარამეტრების შეყვანა ხდება ხელით.

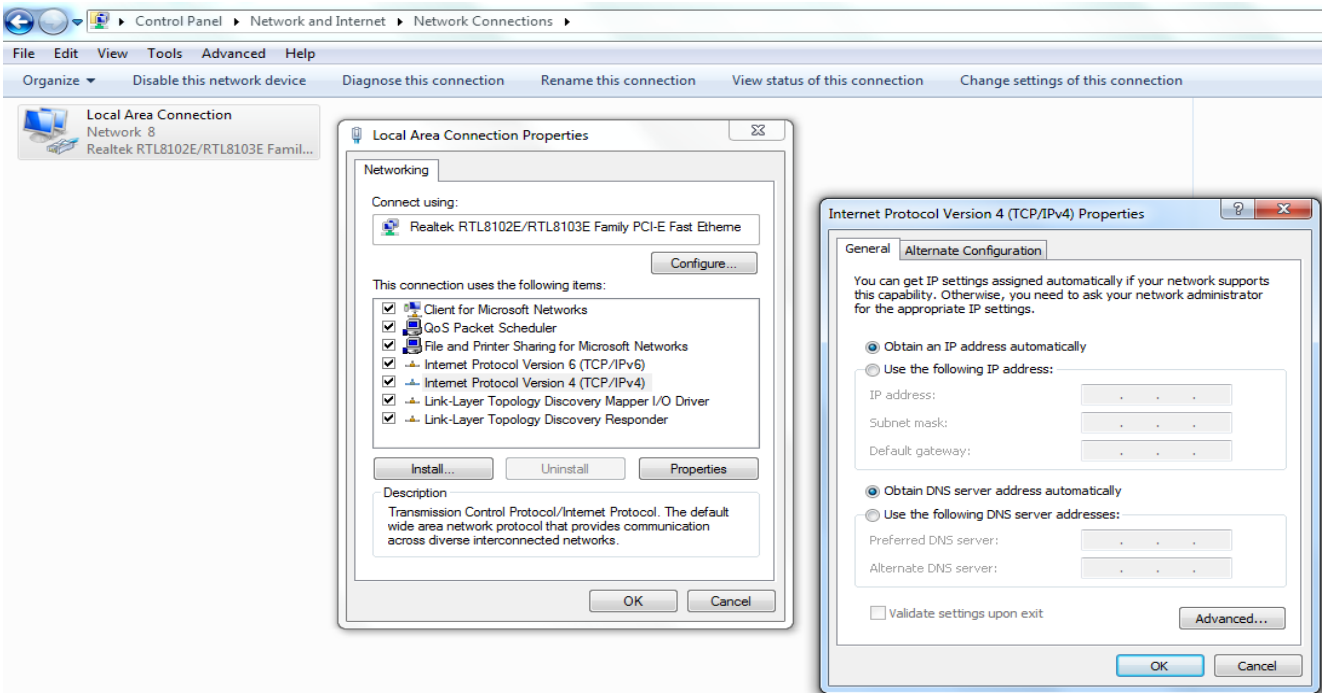
- **დინამიკური IP-დამისამართება.** ქსელში ჩართვისას კომპიუტერი ავტომატურად იღებს TCP/IP პარამეტრებს. ამისათვის ქსელის ერთ-ერთმა კომპიუტერმა უნდა შეასრულოს DHCP-სერვერის ფუნქცია, ე.ი. „დაურიგოს“ IP-მისამართები ყველა ხელახლა მიერთებულ კომპიუტერს.

ქვემოთ სურათზე ასახულია ლოგიკური მისამართების ხელით(სტატიკურად) დანიშვნის მაგალითი




*სურ.1.5.2. 1*

ქვემოთ სურათზე ასახულია ლოგიკური მისამართების ავტომატურად მიღების (DHCP პროტოკოლით) დანიშვნის მაგალითი




*სურ.1.5.2. 2*

## IP-მისამართების დანიშნვა და TCP/IP-ს ქმედითუნარიანობის შემოწმება

მნიშვნელოვანია განვიხილოთ, როგორი ხერხებით შეიძლება IP პარამეტრების(ლოგიკური მისამართების) გამართვა კომპიუტერებზე და სწრაფად როგორ შევამოწმოთ IP-ადრესაციისა და მარშრუტიზაციის ქმედითუნარიანობა.

IP პროტოკოლის პარამეტრების გამართვის ყველაზე მარტივი ხერხია - მათი ხელით დანიშვნა(სტატიკური მარშრუტი). მეთოდის უპირატესობა განპირობებულია იმით, რომ ქსელური ადმინისტრატორები მთლიანად აკონტროლებენ ქსელის ყველა კომპიუტერის IP-მისამართს, რაც მნიშვნელოვანი შეიძლება იყოს მონაცემების დაცვის ან ინტერნეტთან ურთიერთქმედების თვალსაზრისით. ამ მეთოდს აქვს ბევრი ნაკლი.

უპირველეს ყოვლისა, ადვილად შეიძლება შეცდომის დაშვება და არასწორი პარამეტრების შეყვანა ან, უფრო უარესი, ქსელში განმეორებადი IP-მისამართის დანიშვნა.

მეორედ, ქსელში IP-ადრესაციის პარამეტრების ცვლილებისას (მაგალითად, მარშრუტიზატორის IP-მისამართის შეცვლისას) მოგვიწევს ყველა კომპიუტერის ხელახალი გამართვა.

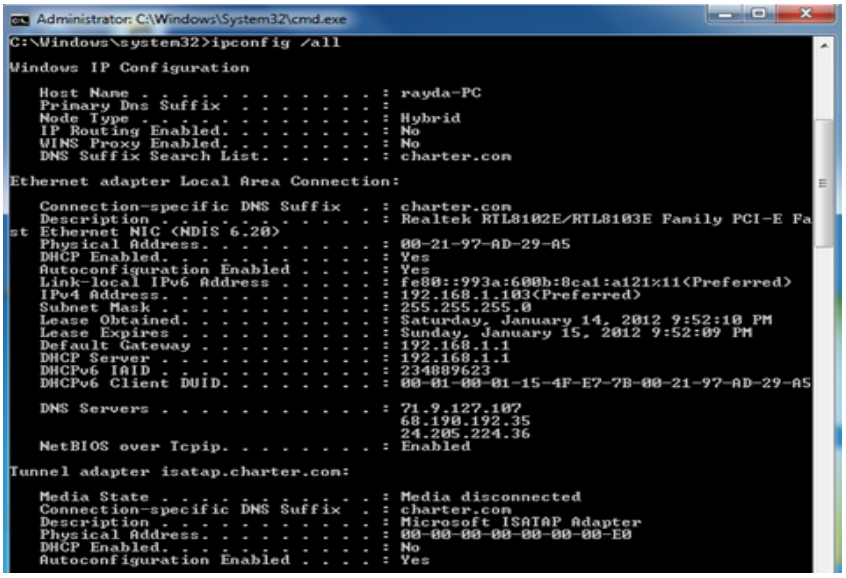
გამართვის ასეთი ხერხისას პრაქტიკულად შეუძლებელია მუშაობა მსხვილ კორპორატიულ ქსელებში ნოუთბუქებით ან ჯპკ-ს ტიპის მობილური მოწყობილობებით, რომლებიც ხშირად გადაადგილდებიან ქსელის ერთი სეგმენტიდან მეორეში.

ორგანიზაციებში უფრო ხშირად იყენებენ სპეციალურ სერვერებს, რომლებიც მხარს უჭერენ კვანძების დინამიურად კონფიგურირების პროტოკოლს(Dynamic Host Configuration Protocol, DHCP). მათი ამოცანა მდგომარეობს IP-მისამართის ან სხვა ინფორმაციის მიღებისათვის კლიენტის მოთხოვნების მომსახურებაში, რაც აუცილებელია ქსელის სწორი მუშაობისათვის. სწორედ ამიტომ კომპიუტერები Windows ოპერაციული სისტემით, Default არიs გამართული IP-მისამართის ავტომატურ მიღებაზე.

თუ DHCP სერვერი არ არის ხელმისაწვდომი (არ არის ან არ მუშაობს), მაშინ Windows ოპერაციული სისტემის კომპიუტერები თავად ინიშნავენ IP-მისამართს. ამ დროს გამოიყენება პირადი IP- ადრესაციის ავტომატური მექანიზმი (Automatic Private IP Addressing), რისთვისაც კორპორაცია Microsoft-ს მიერ IANA-ში დარეგისტრირებულ იქნა მისამართების დიაპაზონი 169.254.0.0 - 169.254.255.255.

### IP პროტოკოლის პარამეტრებისა და ქმედითუნარიანობის შემოწმება

შემოწმებისთვის ვიყენებთ IPCONFIG და PING უტილიტებს, მათთან  სამუშაოდ საჭიროა DOS-რეჟიმის ფანჯრის გახსნა, ავირჩიოთ მენიუ გაშვებაში (Start) პუნქტი შესრულება (Run), შევიყვანოთ ბრძანება cmd და დავაწკაპუნოთ თაგვით ღილაკზე OK.




*სურ.1.5.2. 3*

*<u>შეასრულეთ ბრძანება IPCONFIG/ALL,</u> თუ:*

- ეკრანზე გამოტანილი ინფორმაცია არ შეიცავს არავითარ პარამეტრს, გამოდის, რომ თქვენ არ გაქვთ აქტიური ინტერფეისები.
- თუ გამოტანილ ინფორმაციაში არის დიაგნოსტიკური შეტყობინება ”ქსელი გათიშულია”, მაშინ, თქვენ გაქვთ პრობლემები ფიზიკურ დონესთან - შეამოწმეთ კონექტორის შეერთება ქსელური ადაპტერის გასართთან და/ან კომუტატორების შრომისუნარიანობა.
- თუ თქვენი IP-მისამართის და ქვექსელის ნიღბის პარამეტრები 0.0.0.0-ს ტოლია, ეს ნიშნავს, რომ თქვენ მოიხმართ სტატიკურ IP-მისამართს, რომელიც კონფლიქტშია ქსელის სხვა კვანძთან.
- თუ თქვენი IP-მისამართი იმყოფება 169.254.x.x დიაპაზონში, მაშინ, DHCP-სერვერი არ არის ხელმისაწვდომი და ქსელში მუშაობას თქვენ შეძლებთ მხოლოდ იმ კომპიუტერებთან, რომლებმაც დამოუკიდებლად დაინიშნეს თავისთვის მისამართი.

ნორმალურ სიტუაციაში, IP-მისამართის DHCP-სერვერისგან მიღებისას ან ხელით სწორად გამართვით, ეკრანზე გამოტანილ ინფორმაციაში თქვენ უნდა დაინახოთ ისეთი

პარამეტრები, როგორიცაა კომპიუტერის IP-მისამართი, ქვექსელის ნიღაბი, Gateway, DNS-სერვერი, DHCP-სერვერი (და აგრეთვე, შესაძლოა, სხვა პარამეტრები).

*შეასრულეთ ბრძანება PING 127.0.0.1 თუ პასუხი არ არის მიღებული, ეს ადასტურებს*

- TCP/IP პროტოკოლების სტეკის არასწორ გამართვას; მოგვიწევს შესაბამისი პროგრამული მხარდაჭერის გადაყენება.

თუ პასუხი მიღებულია, ეს მიშნავს, რომ

- TCP/IP პროტოკოლთა სტეკი მუშაობს გამართულად.

*შეასრულეთ ბრძანება PING w.x.y.z, სადაც w.x.y.z არის მეზობელი კომპიუტერის IP-მისამართი.*

ასე მოწმდება ლოკალური ქსელის ქმედითუნარიანობა.

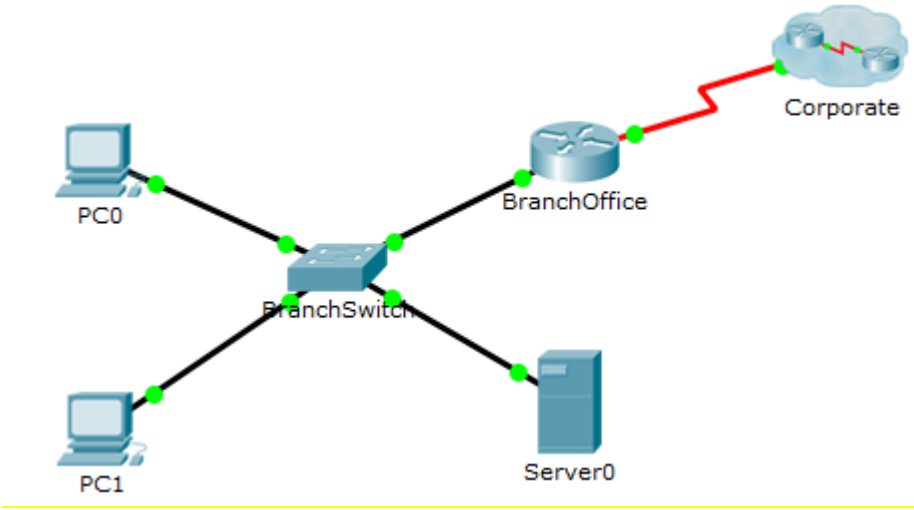
*შეასრულეთ ბრძანება PING w.x.y.z, სადაც w.x.y.z არის ძირითადი კარიბჭის(Gateway) IP-მისამართი.*

ასე მოწმდება Gateway ხელმისაწვდომობა და ქმედითუნარიანობა.

*შეასრულეთ ბრძანება PING w.x.y.z, სადაც w.x.y.z არის ნებისმიერი დაშორებული კომპიუტერის IP-მისამართი.*

ასე მოწმდება თქვენი კორპორატიული ქსელის ან ინტერნეტთან მიერთების მარშრუტიზაციის მთელი სისტემის ქმედითუნარიანობა.

*პრაქტიკული სამუშაო: კომპიუტერების დამატება არსებულ ქსელში*




**დავალება**

- მომართეთ კომპიუტერები DHCP-ის გამოყენებისთვის.
- მომართეთ სტატიკური დამისამართება.
- გამოიყენეთ ipconfig ბრძანება ჰოსტზე IP ინფორმაციის მისაღებად.
- გამოიყენეთ Ping კავშირის შესამოწმებლად.

მოცემულ დავალებაში თქვენ უნდა დაამატოთ ორი კომპიუტერი ფილიალის ქსელში. კომპანიაში დინამიური დამისამართებისათვის ყველა კომპიუტერზე გამოიყენება **DHCP**.

**პირველი ეტაპი. ტოპოლოგიის შესწავლა**

ტოპოლოგიაში შედის ორი კომპიუტერი, სვიჩი, სერვერი, როუტერი და ღრუბელი.

ა) მიაქციეთ ყურადღება რომ პერსონალური კომპიუტერები დაკავშირებულნი არიან **BranchSwitch** კომუტატორთან **Copper Straight-Through** (პირდაპირი შეერთება) კაბელით. **Packet Tracer**-ში პირდაპირი შეერთების კაბელი აღინიშნება სწორი ხაზით.

ბ) ასევე მიაქციეთ ყურადღება მწვანე წერტილებს შეერთების ბოლოებში (თითოეულ კომპიუტერთან და **BranchSwitch** კომუტატორთან). მწვანე წერტილები კაბელის ორივე ბოლოში აღნიშნავენ, რომ ამ მოწყობილობების დასაკავშირებლად არჩეულია სწორი კაბელის ტიპი.

**შენიშვნა:** მწვანე წერტილები უნდა ჩანდეს კაბელის ორივე დაბოლოებაზე. თუ მწვანე წერილები საერთოდ არ ჩანს, გადადით პროგრამის **Options>Preferences** მენიუში და დააყენეთ ალამი **Show Link Lights** პუნქტზე.

**მეორე ეტაპი. DHCP პარამეტრების მომართვა პერსონალურ კომპიუტერზე:**

- დააჭირეთ **PC0** კვანძს. გამოვა **PC0** დიალოგური ფანჯარა;
- გამოსულ ფანჯარაში გადადით **Desktop** ჩანართში;
- დააჭირეთ **IP Configuration** ღილაკს და მონიშნეთ **DHCP** პუნქტი, რის შემდეგაც გამოვა შემდეგი შეტყობინება: **DHCP request successful;**
- დახურეთ **PC0** ფანჯარა;
- დააჭირეთ **PC1** კვანძს, გამოვა **PC1** ფანჯარა;
- დიალოგურ ფანჯარაში აირჩიეთ **Desktop** ჩანართი;
- დააჭირეთ **IP Configuration** ღილაკს და მონიშნეთ **DHCP** პუნქტი, რათა კომპიუტერმა მიიღოს **DHCP** კლიენტის მახასიათებლები;
- დახურეთ **PC1** დიალოგური ფანჯარა;

**მესამე ეტაპი. თითოეული კომპიუტერის IP პარამეტრების ინფორმაციასთან გაცნობა**

- დააჭირეთ **PC0** კომპიუტერს;
- გადადით **Desktop** ჩანართში;
- აირჩიეთ **Command Prompt** მენიუ;
- **PC>** ბრძანებათა სტრიქონში შეიყვანეთ ბრძანება **ipconfig /all**
- ჩაიწერეთ **IP** მისამართი, ქვექსელის ნიღაბი, ნაგულისხმევი კარიბჭე **(Default Gateway)** და **DNS** სერვერის მისამართი, რომლებიც დინამიურადაა დანიშნული **DHCP**-ით მოცემული **PC0** კომპიუტერისთვის.
- ჩაიწერეთ **IP** მისამართი, ქვექსელის ნიღაბი, ნაგულისხმევი კარიბჭე **(Default Gateway)** და **DNS** სერვერის მისამართი, რომლებიც დინამიურად დანიშნულია **DHCP**-ით მოცემული **PC1**კომპიუტერისთვის.
- **Ping** ბრძანების გამოყენებით შეამოწმეთ კავშირი კომპიუტერებს შორის და გამოყენებულ როუტერს შორის.

- **PC0>** ბრძანებათა ველში შეიყვანეთ ბრძანება **ping <IP – address PC1>**
- **PC0>** ბრძანებათა ველში შეიყვანეთ ბრძანება **ping <IP – address router>**
- **PC1>** ბრძანებათა ველში შეიყვანეთ ბრძანება **ping <IP – address PC0>**
- **PC1>** ბრძანებათა ველში შეიყვანეთ ბრძანება **ping <IP – address router>**

**მეოთხე ეტაპი: სტატიკურ დამისამართებაზე გადასვლა**

**DHCP** დინამიური დამისამართების მთელი რიგი უპირატესობების მიუხედავად, ზოგჯერ აუცილებელია სტატიკური სქემის გამოყენება. შეცვალეთ **PC1** კომპიუტერის პარამეტრები **DHCP**-დან სტატიკურ დამისამართებაზე.

- დააჭირეთ **PC1** კვანძს, მისი პარამეტრების ფანჯრის გასახსნელად;
- გადადით **Desktop** ჩანართში;
- აირჩიეთ **IP configuration** მენიუ;
- მონიშნეთ პუნქტი **Static;**
- შეიყვანეთ ქვემოთ მოცემული **IP** მონაცემები:
    - **IP-Address: 172.16.1.20**
    - **Subnet Mask: 255.255.255.0**
    - **Default Gateway: 172.16.1.254**
    - **DNS: 200.75.100.10**
- ახლა **PC1** მომართულია სტატიკურ მისამართზე;
- დახურეთ **IP Configuration** ფანჯარა.

**მეხუთე ეტაპი: კავშირის შემოწმება**

- დააჭირეთ **PC1** კვანძს, მისი პარამეტრების ფანჯრის გასახსნელად;
- გადადით **Desktop** ჩანართში;
- აირჩიეთ **Command Prompt** მენიუ;
- გაგზავნეთ **echo** მოთხოვნა მთავარ **Gateway**-ზე **Ping 172.16.1.254** ბრძანების დახმარებით. **Echo** ტესტირება უნდა განხორციელდეს წარმატებით;

- გაგზავნეთ **echo** მოთხოვნა **Server0**-ზე **Ping 172.16.1.100** ბრძანების დახმარებით. **Echo** ტესტირება უნდა განხორციელდეს წარმატებით;
- გაგზავნეთ **echo** მოთხოვნა როუტერზე, რომელიც გამოყენებულია როგორც **Corporate** ღრუბელში (**Cloud**) შესვლის წერტილი **Ping 172.16.200.1** ბრძანების დახმარებით. **Echo** ტესტირება უნდა განხორციელდეს წარმატებით;
- გაგზავნეთ **echo მოთხოვნა** სერვერზე, რომელიც განთავსებულია **Corporate** ღრუბლის შიგნით, **Ping 200.75.100.10** ბრძანების დახმარებით. **Echo** ტესტირება უნდა განხორციელდეს წარმატებით;
- სრული კავშირი ქსელში მიღწეულია.

1.6. OSI და TCP/IP მოდელების გამოყენება

მოწყობილობებს, რომლებიც ერთმანეთთან არიან დაკავშირებულნი და ცვლიან ერთმანეთს შორის ინფორმაციას, უნდა ქონდეთ საერთო გაცვლის წესები ანუ პროტოკოლები.

**პროტოკოლი** - ეს არის წესები, რომელსაც იყენებენ ქსელური მოწყობილობები ერთმანეთთან დასაკავშირებლად. დღესდღეობით სტანდარტად მიღებულია პროტოკოლები რომლებსაც ეწოდება TCP/IP (Transmission Control Protocol/Internet Protocol). TCP/IP პროტოკოლები განსაზღვრავენ ფორმატიზაციას, დამისამართებას და მარშუტიზაციას, რომლებიც იძლევა გარანტიას, რომ ინფორმაციის მიწოდება მოხდება დანიშნულ ადგილას და უშეცდომოდ.

ინფორმაციის გადაცემისას ერთი მოწყობილობიდან მეორეზე, ინფორმაცია გადის მომზადების რთულ პროცესს, სადაც აღწერილია, თუ რა ეტაპები უნდა გაიაროს ინფორმაციამ, რომ მოხდეს მისი ისეთი მომზადება, რომ შესაძლებელი იყოს მისი ფიზიკურ მედიაში გადაიცემა, და შესამაბისად, რა ეტაპები უნდა გაიაროს, რომ მოხდეს ფიზიკური მედიიდან ორიგინალური ინფორმაციის მისაღებად.

ეს ეტაპები კომპიუტერულ ქსელებში წარმოდგენილია დონეების სახით. თითოეულ დოენზე მუშაობს გარკვეული ტიპის პროტოკოლი, რომელიც პასუხისმგებელია მის ზემოთ არსებული დონიდან მიღებული ინფორმაცია დაამუშაოს და გადასცეს მის ქვემდგომ დონეს. ამ პროცესს ენკაფსულაციას უწოდებენ.

აღნიშნული დონეები ქმნიან პროტოკოლების ნაკრებს (სტეკს). ანუ სტეკი არის პროტოკოლების ნაკრები, რომელიც უზრუნველყოფს ნებისმიერი ტიპის ინფორმაციის მომზადებას და გადაგზავნას ფიზიკურ მედიაში და პირიქით - ფიზიკური მედიიდან მის აღდგენას იმ სახით რა სახითაც იქნა გადაცემული.

კომპიუტერულ ქსელებში არსებობს პროტოკოლების სხვადასხვა ნაკრები. როგორც წესი ისინი არათავსებადი არიან ერთმანეთთან.

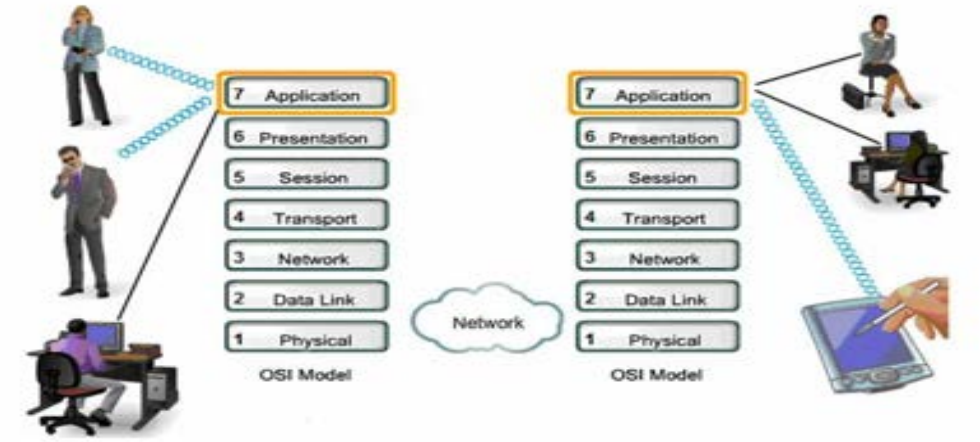
რათა წარმოვსახოთ ურთიერთქმედება სხვადასხვა პროტოკოლებს შორის, გამოიყენება დონეებად დაყოფის მოდელი. იგი აღწერს პროტოკოლების ოპერაციებს, რომლებსაც

ადგილი აქვთ თითოეულ დონეზე, და ასევე ურთიერთქმედებას მის ზედა და ქვედა დონეებთან.

### *1.6.1. OSI მოდელი*

80-იან წლებში საერთაშორისო ორგანიზაციებმა დაამუშავეს OSI (Open System Interconnection) – ღია სისტემების ურთიერთკავშირის მოდელი, რომელმაც დიდი როლი ითამაშა ქსელების განვითარებაში. ამ მოდელის შემუშავებაში გარკვეული როლი ითამაშა შემდეგმა ფაქტორმა. სანამ ამ მოდელზე დაიწყებდნენ ფიქრს, მანამდე კომპანიები რომლებიც იმ დროისთვის აწარმოებდნენ ქსელურ აპარატურას, გასაიდუმლოებულ ვითარებაში ქმნიდნენ პროტოკოლებს რათა გაეერთიანებინათ ქსელური მოწყობილობები. ამიტომ სხვადასხვა მწარმოებელმა შექმნა ინფორმაციის გაცვლის სხვადასხვა დონიანი ინფორმაციის გაცვლის პროტოკოლების სტეკი. აღსანიშნავია ის, რომ სხვადასხვა მწარმოებლის მიერ შექმნილი ქსელური მოწყობილობები ერთმანეთთან ვერ ცვლიდნენ ინფორმაციას. ეს კი იმ პერიოდისთვის მნიშვნელოვანი შემაფერხებელი გარემოება იყო. ამიტომ გახდა საჭირო შემუშავებულიყო ისეთი პროტოკოლების სტეკი, რომელიც საერთო იქნებოდა ყველა სისტემისთვის.

OSI არის ეტალონური მოდელი, რომელმაც მნიშვნელოვანი როლი შეასრულა თანამედროვე კომპიუტერული ქსელების კონცეფციების განვითარებაში. OSI მოდელში ურთიერთქმედების საშუალებები იყოფა <u>შვიდ დონედ</u>: **გამოყენებითი, წარმოდგენითი, სეანსის, ტრანსპორტის, ქსელის, არხის და ფიზიკური.** ყოველ დონეს სხვადასხვა ქსელური ოპერაციები შეესაბამება. ყოველი დონე გადამცემ კომპიუტერზე მუშაობს ისე, თითქოს ის შეესაბამებოდეს მიმღები კომპიუტერის შესაბამის დონეს. ეს ლოგიკური ანუ ვირტუალური კავშირი ნაჩვენებია სურათზე. რეალური კავშირი კი მხოლოდ მეზობელ დონეებს შორის ხორციელდება.



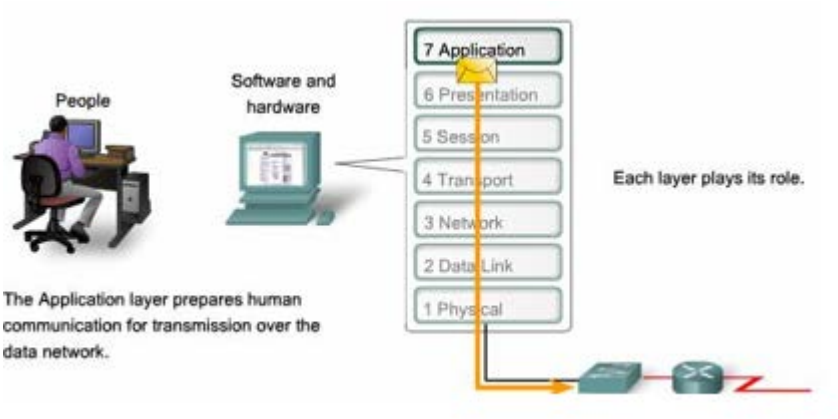

*სურ.1.6.1. 1*

OSI მოდელს ხშირად იყენებენ კომპიუტერული ქსელების აგებისას. მისი მთავარი თვისებაა სხვადასხვა დონეების ერთმანეთთან დაკავშირება, რაც ასევე უზრუნველყოფს ერთ დონეზე მომუშავე მწარმოებლის მიერ შემუშავებული აპარატურის სხვა დონეზე მომუშავე აპარატურასთან მუშაობას, თუ ამ აპარტურის ყოველი პროტოკოლი დოკუმენტირებულია და მისი აღწერილობა არსებობს. ეს აღწერილობა TCP/IP-ზე მომუშავე საზოგადოებისთვის ჩვეულებრივ ცნობილია როგორც RFC-ს დოკუმენტაცია (Request for Comments).

| OSI მოდელი | | | |
|---|---|---|---|
| | მონაცემების ერთეული | დონე | ფუნქცია |
| პროგრამული | მონაცემები | გამოყენებითი | ინფორმაციის მომზადება ქსელში გადასაცემად |
| | | წარმოდგენითი | მონაცემების შიფრაცია და წარდგენა |
| | | სესიის | კავშირი |
| | სეგმენტები | ტრანსპორტის | კავშირი ორ ბოლო წერტილს შორის და საიმედოობა |
| აპარატურული | პაკეტები | ქსელის | გზის განსაზღვრა და ლოგიკური დამისამართება (IP) |
| | კადრები | არხის | ფიზიკური მისამართები (MAC და LLC) |
| | ბიტები | ფიზიკური | მატარებელი ხაზი(მედია), სიგნალი და ორობითი გადაცემა |

*სურ.1.6.1. 2*

### *1.6.1.1. OSI დონეების აღწერა*




*სურ.1.6.1.1. 1*

**დონე 7: გამოყენებითი დონე (Application Layer)**

გამოყენებითი დონე უზრუნველყოფს ქსელურ პროგრამებს ქსელური სერვისებით. გამოყენებითი დონე ესაა სხვადასხვა პროტოკოლების ნაკრები, რომლის საშუალებით ქსელის მომხმარებელი უკავშირდებიან საერთო რესურსებს, როგორიცაა ფაილები, პრინტერი ან web გვერდები. პროტოკოლების მაგალითებია: Telnet, File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP) და Hypertext Transfer Protocol (HTTP) პროტოკოლები.

**დონე 6: წარმოდგენითი დონე (Presentation Layer)**

წარმოდგენითი დონე გარდაქმნის მონაცემებს პროგრამული დონის სტანდარტული ინტერფეისისათვის გასაგებ ენაზე. MIME კოდირება, მონაცემების შეკუმშვა, მონაცემების კოდირება და ზემდგომი დონის მოთხოვნის ფარგლებში მისი წარმოდგენა. მაგალითად: EBCDIC-ით კოდირებული ტექსტური ფაილის ASCII-კოდირებულ ფაილად გარდაქმნა, ობიექტების და სხვა მონაცემთა სტრუქტურის XML-ში გარდაქმნა და ა.შ.

**დონე 5: სესიის დონე (Session Layer)**

სესიის დონე აკონტროლებს დიალოგს (სესიებს) კომპიუტერებს შორის. ის იწყებს, მართავს და წყვეტს კავშირებს ადგილობრივ და შორეულ პროგრამებთან. ის იძლევა დუპლექსური ან ნახევრადდუპლექსური კავშირის დამყარების საშუალებას და ახდენს საბოლოო კავშირის შესრულების შემოწმებას, რეგულირებას, შეწყვეტას და განახლებას. OSI

მოდელში ეს დონე პასუხისმგებელია სესიების "მშვიდობიან დახურვაზე", რაც TCP პროტოკოლის და ინტერნეტ პროტოკოლის უმნიშვნელოვანესი ნაწილია.

### დონე 4: ტრანსპორტის დონე (Transport Layer)

ტრანსპორტის დონე უზუნველყოფს მომხმარებლებს შორის მონაცემების გამჭვირვალე, ეფექტურ გადაცემას და ამ დავალებისგან ზედა დონეების განთავისუფლებას. ტრანსპორტის დონე ამოწმებს საიმედოობას ნაკადების მართვით, სეგმენტირებით/დესეგმენტირებით და შეცდომების შემოწმებით.

მეოთხე დონის ზოგიერთი პროტოკოლი მოითხოვს ორმაგი კავშირის დამყარებას. ეს ნიშნავს, რომ ტრანსპორტის დონეს შეუძლია პაკეტების დროებით შენახვა და დანაკარგების შემთხვევაში მათი თავიდან გაგზავნა. მსგავსი პროტოკოლია (TCP) Transmission Control Protocol. ეს არის დონე, რომელიც გარდაქმნის შეტყობინებებს TCP, (UDP) User Datagram Protocol, (SCTP) Stream Control Transmission Protocol და სხვა პაკეტებში.

### დონე 3: ქსელის დონე (Network Layer)

ქსელური დონე უზრუნველყოფს მონაცემების მიმდევრობების წყაროდან დანიშნულების ადგილამდე ერთი ან რამოდენიმე ქსელის გავლით გადაცემას ტრანსპორტის დონის მიერ მოთხოვნილი მომსახურების ხარისხის (QoS) დაცვით. ქსელური დონე აწარმოებს ქსელური მარშრუტიზაციის ფუნქციებს, და ასევე შეუძლია სეგმენტირება/დესეგმენტირება და შეცდომების შეტყობინება. მარშრუტიზატორები მუშაობენ სწორედ ამ დონეზე და აგზავნიან პაკეტებს ერთი ქსელიდან მეორეში, რაც საბოლოოდ შეიძლება ქსელის მომხმარებლის ინტერნეტამდე წვდომას უზრუნველყოფდეს (ასევე არსებობს მესამე დონის კომუტატორები (ხშირად მათ IP-კომუტატორებს უწოდებენ). ეს არის მისამართების ლოგიკური სქემა – მნიშვნელობები შეირჩევა ქსელური ინჟინერის მიერ, მისამართების სქემა იერარქიულია. მესამე დონის პროტოკოლის საუკეთესო მაგალითია ინტერნეტ პროტოკოლი (IP).

### დონე 2: მონაცემთა გადაცემის არხის დონე (Data Link Layer)

მონაცემთა გადაცემის არხის დონე უზრუნველყოფს ქსელურ ობიექტებს შორის მონაცემების ელემენტარულ გადაცემას და ფიზიკურ დონეზე მომხდარი შეცდომების

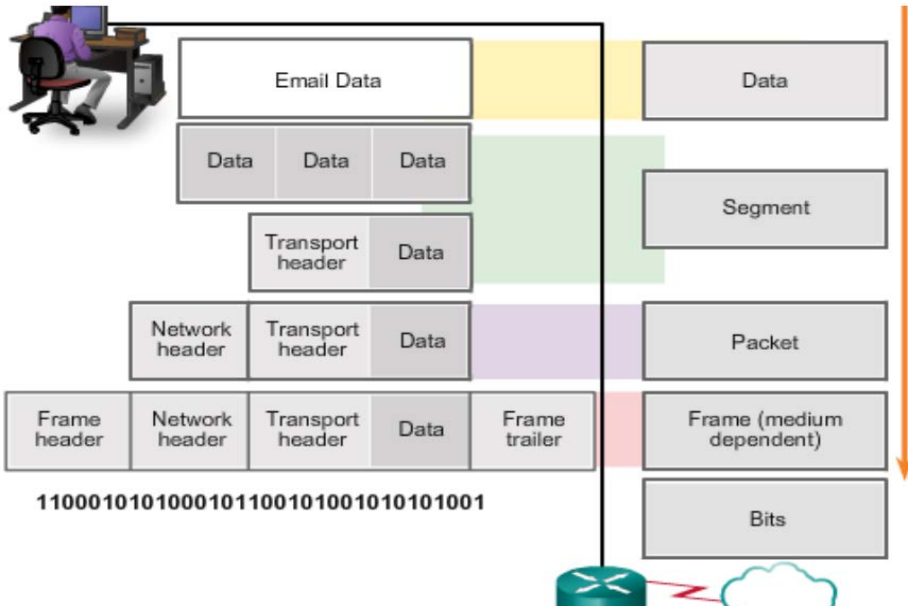
აღმოჩენას და შესაძლო აღმოფხვრას. მისამართების სქემა ფიზიკურია (MAC მისამართები). რაც ნიშნავს, რომ ისინი აპარატურულ ნაწილში ფიქსირდება წარმოების დროს. მეორე დონის პროტოკოლის მაგალითიებია: Ethernet, HDLC, ADCCP. (შენიშნვა: IEEE 802 სტანდარტის ლოკალურ ქსელებში და ზოგიერთ არა-IEEE 802 ქსელებში, მაგალითად FDDI-ში, ეს დონე იყოფა ორად: MAC დონედ და IEEE 802.2 LLC დონედ, ამ დონეზე მუშაობენ ქსელური ხიდები და კომუტატორები. არსებობს არგუმენტი, რის მიხედვითაც ამ დონეს უწოდებენ "2.5 დონეს", რადგან თვისობრივად ის მეორე დონეს მკაცრად არ უტოლდება).

**დონე 1: ფიზიკური დონე (Physical Layer)**

ფიზიკური დონე განსაზღვრავს მოწყობილობების ყველა ფიზიკურ და ელექტრულ თვისებებს. ის მოიცავს კაბელების ტიპს, მის განლაგებას, კაბელის პარამეტრებს, ტალღის სიხშირეს და ა.შ. კონცენტრატორები პირველი დონის მოწყობილობებია.

ფიზიკური დონის ძირითადი ფუნქცია და დანიშნულებაა:

- ელექტრული კავშირის დამყარება და გაწყვეტა ინფორმაციის მატარებელთან;
- მრავალ მომხმარებელს შორის საკომუნიკაციო რესურსების ეფექტურად განაწილება. მაგალითად, კავშირის მოთხოვნა და დინების მართვა;
- მოდულაცია, ან ციფრული მონაცემების გადამცემა არხებში გასატარებლად. მაგალითად ეს არის სიგნალები ფიზიკურ კაბელში (როგორც მავთული, ასევე ოპტიკურ-ბოჭკოვანი) და ეთერში.




*სურ.1.6.1.1. 2*

*1.6.2.*

*TCP/IP მოდელი*

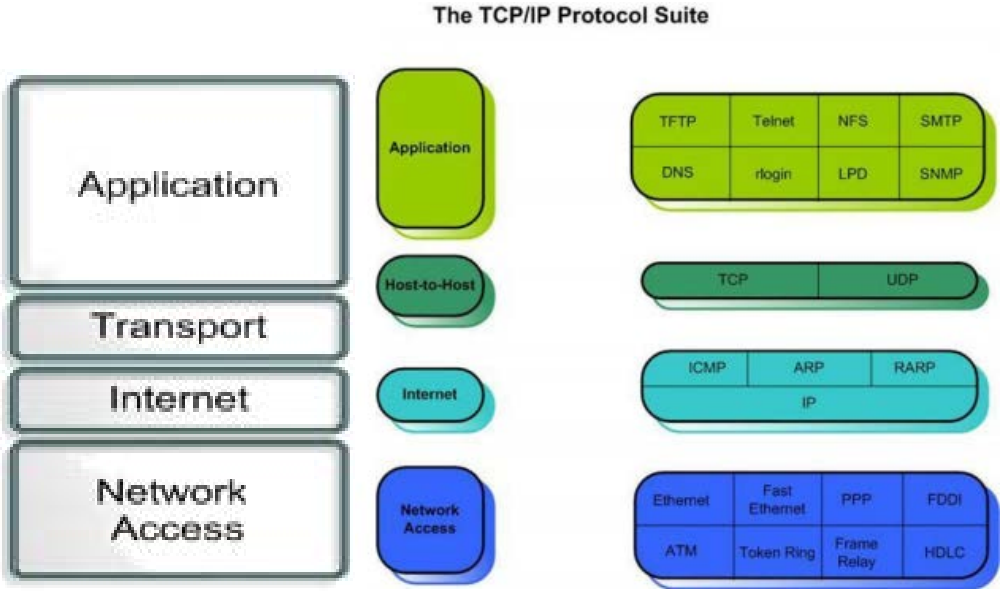
ინტერნეტი შეიქმნა და განვითარდა იმისათვის, რომ მომხდარიყო სხვადასხვა ტიპის ქსელები გააერთიანება. ინტერნეტი მუშაობს TCP/IP პროტოკოლის გამოყენებით. TCP/IP პროტოკოლის დიზაინი იდელაურია ინტერნეტის დეცენტრალიზაციისა და განვითარებისთვის.

საჭიროა ვიცოდეთ ორივე TCP/IP და OSI ქსელური მოდელი. ყოველ მოდელს გააჩნია საკუთარი სტრუქტურა, თუ როგორ უნდა იმუშაოს ქსელმა. მაგრამ ორივე მოდელს ასევე აქვს ბევრი საერთო.

TCP/IP მოდელი შედგება 4 დონისგან:

**გამოყენებითი; ტრანსპორტის; ინტერნეტის და ქსელში შეღწევის დონე.**

ზოგიერთ TCP/IP მოდელის დონეს აქვს საერთო დასახელება, როგორიც აქვს OSI მოდელს.




*სურ.1.6.2. 1*

**გამოყენებითი დონე**

გამოყენებითი დონეში შედის მაღალი დონის პროტოკოლები, რომლებიც უზრუნველყოფენ მონაცემების წარმოდგენას, კოდირებას და სეანსის კონტროლს. ამ დონის პროტოკოლებია:

File Transfer Protocol (FTP) - FTP არის კავშირზე ორიენტირებული, გარანტირებული გადაცემის სერვისი, რომელიც ტრანსპორტის დონეზე იყენებს TCP პროტოკოლს;

Trivial File Transfer Protocol (TFTP) – TFTP არის კავშირზე არაორიენტირებული, არაგარანტირებული გადაცემის სერვისი, რომელიც ტრანსპორტის დონეზე იყენებს UDP პროტოკოლს;

Simple Mail Transfer Protocol (SMTP) – SMTP უზრუნველყოფს ელექტრონული ფოსტის გადაგზავნას ქსელის საშუალებით;

Telnet – Telnet უზრუნველყოფს ერთი კომპიუტერიდან მეორე კომპიუტერში შესვლას და ბრძანებების გაშვებას რომელის სრულდება დაშორებულ კომპიუტეერში.

**ტრანსპორტის დონე**

ტრანსპორტის დონე უზრუნველყოფს ლოგიკურ კავშირს ინიციატორ ჰოსტსა და ადრესატ ჰოსტს შორის. სატრანსპორტო პროტოკოლი უკეთებს სეგმენტაციას ზედა დონის პროტოკოლიდან მოსულ ბაიტურ ნაკადს და უზრუნველყოფს მის აწყობას მეორე მხარეს, რათა გადასცეს ის ზედა დონეს მთლიან ნაკადად.

ტრანსპორტის დონის ძირითადი ფუნქციაა - უზრუნველყოს გადაცემაში მონაწილე ჰოსტებს შორის კონტროლი და მონაცემების გარანტირებული გადაცემა ქსელში. ტრანსპორტის დონის პროტოკოლებია TCP და UDP.

TCP და UDP ფუნქციები:

- გამოყენებითი დონის მონაცემების სეგმენტაცია;
- სეგმენტების გადაცემა ერთი ჰოსტიდან მეორეში.

TCP ფუნქციაა:

- ჰოსტებს შორის კავშირის დამყარება;
- მონაცემთა ნაკადის მართვა მცოცავი ფანჯრის გამოყენებით;
- საიმედოობის უზრუნველყოფა სპეციალური სისტემის გამოყენებით.

**ინტერნეტის დონე**

ინტერნეტის დონის ფუნქციაა უზრუნველყოს საუკეთესო გზის არჩევა ინტერნეტში პაკეტების მარშუტიზაციისას. მთავარი პროტოკოლი რომელიც ამ დონეზე მუშაობს არის IP.

TCP/IP-ში ინტერნეტის დონეზე მუშაობს შემდეგი პროტოკოლები:

• IP უზრუნველყოფს კავშირზე არაორიენტირებულ, მაგრამ საუკეთესო გზით პაკეტების გადაცემას;

• Internet Control Message Protocol (ICMP) უზრუნველყოფს კონტროლისა და შეტყობინებების გაგზვნას;

• Address Resolution Protocol (ARP) უზრუველყოფს IP მისამართის საშუალებით ფიზიკური MAC მისამართის დადგენას;

• Reverse Address Resolution Protocol (RARP) უზრუველყოფს IP მისამართის დადგენას ცნობილი ფიზიკური MAC მისამართის საშუალებით.

### ქსელში შეღწევის დონე

ქსელში შეღწევის დონე უზრუნველყოფს პაკეტების გადაცემას ფიზიკურ გარემოში. ამ დონეზე მუშაობს, როგორც ლოკალური ასევე გლობალური ქსელის ტექნოლოგიები.

ქსელში შეღწევის დონე აგრეთვე აკეთებს IP პაკეტების ენკაპსულაციას ფრეიმებში. ეს დონე განსაზღვრავს ფიზიკური მედიის კავშირის ტიპს დამოკიდებულს ფიზიკურ მოწყობილობაზე და ქსელურ ინტერფეისზე.

### ფიზიკური დონე

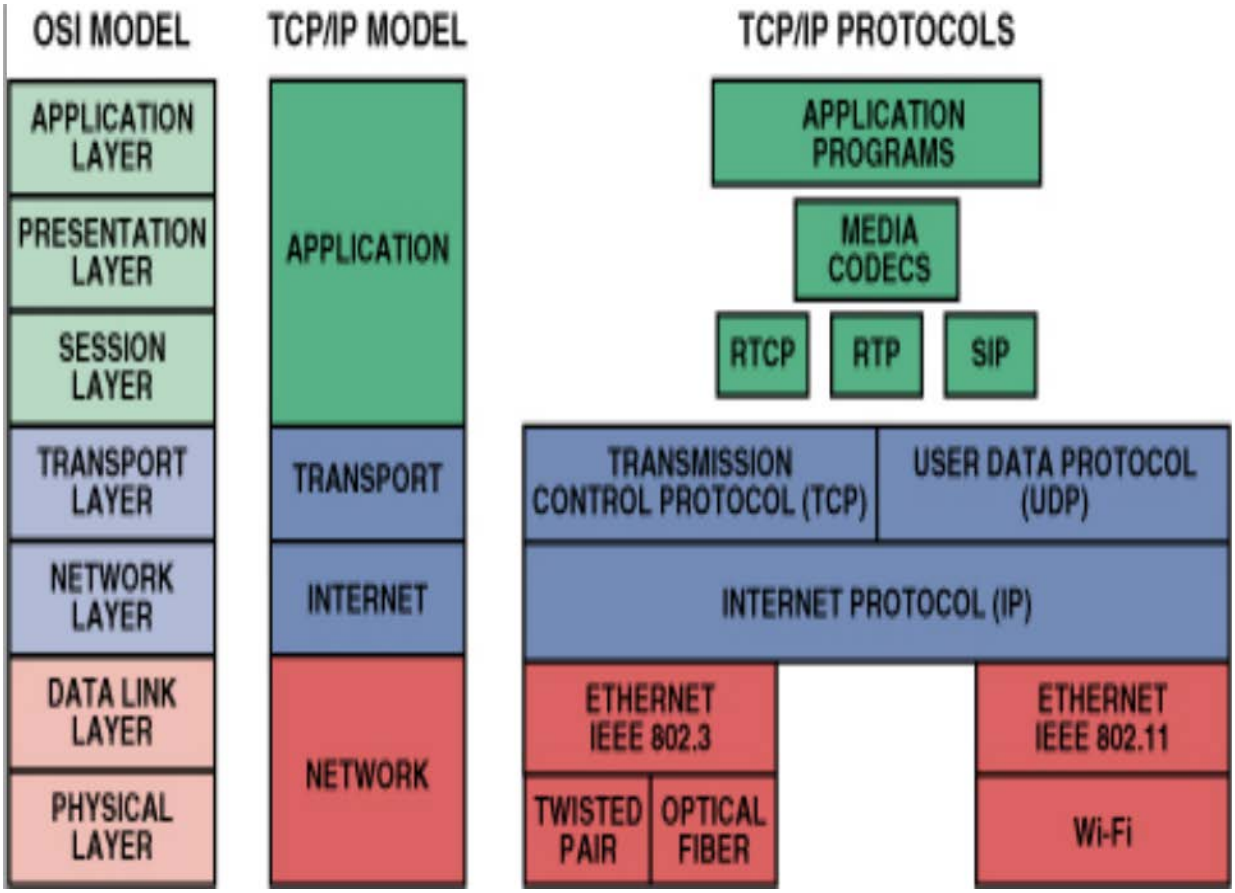
ინფორმაციის გადაცემის გარემო ესაა კომპიუტერების ერთმანეთთან დაკავშირების საშუალება, რომლითაც ხდება ინფორმაციის გაცლა. კომპიუტერულ ქსელებში გადაცემის გარემოდ გამოყენებულია კაბელები და უგამტარო კავშირები.

### შედარება OSI და TCP/IP მოდელს შორის

პროტოკოლები რომლებიც შედიან TCP/IP მოდელის შენადგენლობაში შესაძლებელია იქნან აღწერილი OSI მოდელის განმარტებით. OSI მოდელში ქსელში შეღწევის დონე და

TCP/IP მოდელის გამოყენებითი დონე არის დაყოფილი რათა აღვწეროთ ფუნქციები რომლებსაც ადგილი ექნებათ ამ დონეებზე.

ქსელური შეღწევის დონეზე TCP/IP პროტოკოლების ნაკრები არ განსაზღვრავს თუ რომელი პროტოკოლი გამოიყენება ფიზიკურ გარემოში ინფორმაციის გადასაცემად. ის მხოლოდ აღწერს დამოკიდებულებას ინტერნეტ დონიდან ქსელის ფიზიკურ პროტოკოლებამდე. OSI მოდელის 1 და 2 დონეები განიხილავენ აუცილებელ პროცედურებს, რათა მიიღონ შეღწევის უფლება მედიაზე და ფიზიკურ საშუალებებზე, რათა გააგზავნოს მონაცემი ქსელში.




*სურ.1.6.2. 2*

ძირითადი განსხვავება ორ ქსელურ მოდელს შორის ხდება OSI მოდელის მე-3 და 4 დონეზე. OSI მოდელის მე-3 დონე ეს არის ქსელური დონე, რომელიც უნივერსალურად

გამოიყენება რათა განიხილოს და დოკუმენტაცია გაუკეთოს პროცესების დიაპაზონს, რომლებიც ხდება ყველა ინფორმაციის გადამცემ ქსელში, რათა დაამისამართოს და დაამარშუტიროს შეტყობინება ქსელში გადასაცემად. ინტერნეტ პროტოკოლი (IP) წარმოადგენს TCP/IP პროტოკოლების ნაკრებს, რომელიც შეიცავს მე-3 დონის ფუნქციონალურ შესაძლებლობებს.

OSI მოდელის მე-4 დონე არის ტრანსპორტის დონე. იგი ხშირად გამოიყენება რათა აღიწეროს საერთო ფუნქციები ან მომსახურებები, რომელსაც განსაზღვრავენ (მართავენ) გამგზავნი და მიმღები ჰოსტები ერთმანეთში ინდივიდუალური ურთიერთობისას. ეს ფუნქციები შეიცავენ დასტურს (acknowledgement), შეცდომების აღმოფხვრას (error recovery) და თანმიმდევრობას(sequencing). ამ დონეზე TCP/IP პროტოკოლების TCP (Transmission Control Protocol) და UDP (User Datagram Protocol) პროტოკოლები უზრუნველყოფენ აუცილებელ ფუნქციებს.

*ტესტის ნიმუში*

1. **დაალაგეთ OSI მოდელის დონეთა თანმიმდევრობა:**

| | |
|---|---|
| 1 | ფიზიკური დონე |
| 2 | გამოყენებითი |
| 3 | სესიის |
| 4 | არხის |
| 5 | ქსელის |
| 6 | ტრანსპორტის |
| 7 | წარმოდგენითი |

2. **დაალაგეთ TCP/IP მოდელის დონეთა თანმიმდევრობა:**

| | |
|---|---|
| 1 | ინტერნეტის |
| 2 | გამოყენებითი |
| 3 | ტრანსპორტის |
| 4 | ქსელში შეღწევის |

3. **შეუსაბამეთ ერთმანეთს ქსელური დონე და ენკაპსულაციის შედეგად ფორმირებული ინფორმაციის წარმოდგენის ფორმა**

| | |
|---|---|
| ტრანსპორტი (Transport) | მონაცემები (Data) |
| არხი (Data Link) | სეგმენტი (Segment) |
| ფიზიკური (Physical) | ბიტი (Bit) |
| გამოყენებითი (Application) | კადრი (Frame) |
| | თავსართი (Header) |

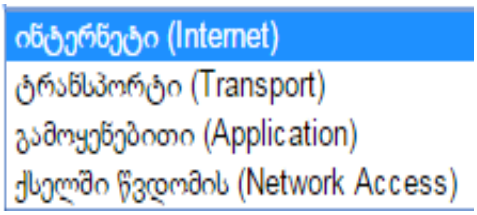
4. **შეუსაბამეთ ერთმანეთს:**

FTP პროტოკოლი

UDP პროტოკოლი

IP პროტოკოლი

IEEE 802.3 პროტოკოლი




5. **OSI მოდელში, რომელ დონეზე მუშაობს IPv4 და IPv6 პროტოკოლები?**

- გამოყენებითი (Application)
- ტრანსპორტის (Transport)
- წარდგენის (Presentation)
- ქსელის(Network)
- არხის (Data link)

6. **OSI მოდელში, რომელი დონე აწარმოებს მარშრუტიზაციის ფუნქციებს?**

- გამოყენებითი (Application)
- ტრანსპორტის (Transport)
- წარდგენის (Presentation)
- ქსელის(Network)
- არხის (Data link)
- ფიზიკური (Physical)

7. **TCP/IP მოდელში, რომელ დონეზე მუშაობენ პროტოკოლები, რომლებიც უზრუნველყოფენ მონაცემების წარმოდგენას, კოდირებას და სეანსის კონტროლს?**

- გამოყენებითი (Application)
- ტრანსპორტის (Transport)
- ინტერნეტის (Internet)
- ქსელში შეღწევის (Network Access)
- არხის (Data link)

### 8. რომელი მსჯელობაა მცდარი?

- OSI მოდელს აქვს ვერტიკალური სტრუქტურა, რომელშიც ყველა ქსელური ფუნქცია განაწილებულია შვიდ დონეს შორის
- OSI მოდელის თითოეული ქვედა დონის ამოცანაა - მიიღოს მონაცემები ზედა დონიდან, დაამატოს თავისი ე. წ. სამსახურეობრივი ინფორმაცია და გადასცეს მონაცემები შემდეგს
- ქსელური მოდელის ყველაზე დაბალი, ფიზიკური დონის მიღწევისას, ინფორმაცია მოხვდება გადაცემის გარემოში
- OSI მოდელის ოთხივე დონეს შეესაბამება, მკაცრად განსაზღვრული ოპერაციები, მოწყობილობები და პროტოკოლები

### 9. რომელი დონეზე ხდება მონაცემთა ფორმატის განსაზღვრა?

- გამოყენებითი (Application)
- ტრანსპორტის (Transport)
- წარდგენის (Presentation)
- ქსელის(Network)
- არხის (Data link)
- ფიზიკური (Physical)

პროცესზე დაკვირვება

შემფასებელი აკვირდება შესაფასებელ პირის მუშაობას პროფესიული სტანდარტით(პროგრამით / მოდულით ) განსაზღვრული ამოცანების შესრულების პროცესში . დაკვირვებახორციელდება სამუშაო ადგილზე ან სამუშაო პირობებში, თუმცა დასაშვებია ჩატარდეს კომპიუტერებით აღჭურვილ ლაბორატორიაში, სადაც შესაფასებელი პირი პრაქტიკულ საქმიანობას ეწევა. შემფასებელმა წინასწარ უნდა დაგეგმოს დაკვირვების პროცესი, იმის დასადგენად, თუ რამდენად სწორად იყენებს შესაფასებელი პირი ცოდნას, უნარებსა და ყველა რესურსს შედეგის მისაღწევად. შეფასება დადებითია იმ შემთხვევაში თუ სტუდენტი ყველა კითხვას სწორად გასცემს პასუხს.

*პრაქტიკული სამუშაო* -

**ქსელური უსადენო და სადენიანი მოწყობილობებით ქსელის გაყვანა და ოპტიმალური დამისამართება.**

| სწავლის შედეგი | N | დასახელება | შეფასება | |
|---|---|---|---|---|
| | | | კი | არა |

| | | | | |
|---|---|---|---|---|
| სწავლის შედეგი 2 ქსელური მოწყობილობების ფიზიკური და ლოგიკური მისამართების დანიშნულება, მათი გამოყენება და განსხვავებები | 1. | სწორად შეარჩია ლოგიკური მისამართების კლასი ქსელურ მოწყობილობებზე გასაწერად | | |
| | 2. | სწორად მოახდინა შერჩეული ლოგიკური მისამართების გაწერა ქსელურ მოწყობილობებზე | | |
| სწავლის შედეგი 4 მარტივი ქსელის გამართვა | 1. | სწორად შეარჩია კაბელის სტანდარტი(Straight-through, cross-over) | | |
| | 2. | სწორად მოარგო კაბელებზე კონექტორები | | |
| | 3. | დააკავშირა მოწყობილობები ერთმანეთთან კაბელის საშუალებით | | |
| | 4. | დააკონფიგურირა არსებული ქსეული მოწყობილობები | | |
| | 5. | შეამოწმა კავშირი ფიზიკურად დაკავშირებულ მოწყობილობებს შორის | | |
| სწავლის შედეგი 5 უკაბელო ქსელის კონფიგურაციაუკაბელო ქსელის კონფიგურაცია | 1. | დააკონფიგურირა არსებული უსადენო ქსელური მოწყობილობები მითითებული სტანდარტის და უსაფრთხოების ნორმების მიხედვით | | |
| | 2. | დააკავშირა მოწყობილობები ერთმანეთთან უსადენო ქსელის საშუალებით | | |
| | 3. | შეამოწმა კავშირი უსადენოდ დაკავშირებულ მოწყობილობებს შორის | | |

*მტკიცებულების (პროდუქტის / შედეგის) შეფასება*

კვალიფიკაციის მისაღებად შესაფასებელი პირი ქმნის მტკიცებულებას. მტკიცებულება შეიძლება წარმოდგენილი იყოს პროდუქტის სახით. პროდუქტი შეიძლება იყოს სტუდენტის

ან სტუდენტთა ჯგუფის მიერ შექმნილი მაკეტი, დეტალი ან სტუდენტის შრომით დამზადებული სხვა სახის არტეფაქტი. მტკიცებულება არის არამხოლოდ ხელით შექმნილი პროდუქტი (არტეფაქტი), არამედ სხვა ტიპის შედეგიც, როგორიცაა ხელსაწყოების გამოყენება, პროექტის შექმნა, სისტემის გამართვა და ა. შ.

მტკიცებულების (პროდუქტის/შედეგის) შეფასების შედეგად შემფასებელს შეუძლია განსაზღვროს და გადაწყვიტოს, აითვისა თუ არა შესაფასებელმა პირმა მოდულის სწავლის შედეგებით მოთხოვნილი ცოდნა და უნარები. მტკიცებულების შეფასებასთან დაკავშირებული რისკ ფაქტორი არის იმის გარკვევა, კონკრეტული მტკიცებულება ნამდვილად შესაფასებელი პირის მიერაა შექმნილი თუ არა. ( რისკების გამოსარიცხად მნიშვნელოვანია დაკვირვების განხორციელება). შეფასება დადებითია იმ შემთხვევაში თუ სტუდენტი ყველა კითხვას სწორად გასცემს პასუხს.

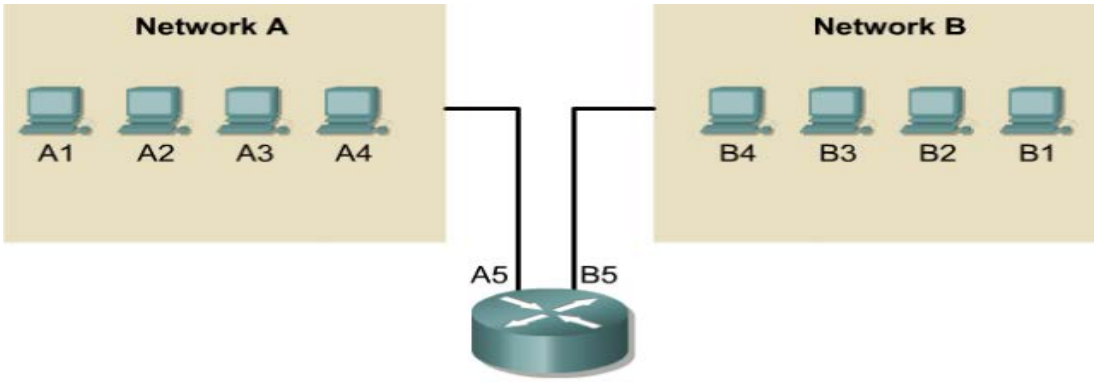
| სწავლის შედეგი | | დასახელება | შეფასება | |
|---|---|---|---|---|
| | | | კი | არა |
| ქსელური მოწყობილობების ფიზიკური და ლოგიკური მისამართების დანიშნულება, მათი გამოყენება და განსხვავებები | 1. | ლოგიკური მისამართები განაწილებულია მოთხოვნის შესაბამისად | | |
| | 2. | ლოგიკური მისამართები სწორად არის გაწერილი მოწყობილობებზე | | |
| მარტივი ქსელის გამართვა | 1. | კაბელის სტანდარტი შერჩეულია სწორად | | |
| | 2. | კაბელზე კონექტორები მორგებულია სწორად | | |
| | 3. | მოწყობილობბი დაკავშირებულია ერთმანეთთან კაბელებით | | |
| | 4. | ქსელური მოწყობილობები დაკონფიგურირებულია მოთხოვნის შესაბამისად | | |
| უკაბელო ქსელის კონფიგურაცია | 1. | უსადენო ქსელური მოწყობილობები დაკონფიგურირებულია მოთხოვნის შესაბამისად | | |
| | 2. | მოწყობილობები დაკავშირებულია ერთმანეთთან უსადენო ქსელის საშუალებით | | |

# 2. IPv4 /IPv6 ადრესაცია და ქსელის ქვექსელებად დაყოფა

გვაქვს IP მისამართის 32 ბიტიანი IPv4 და 128 ბიტიანი IPv6 სტანდარტები.

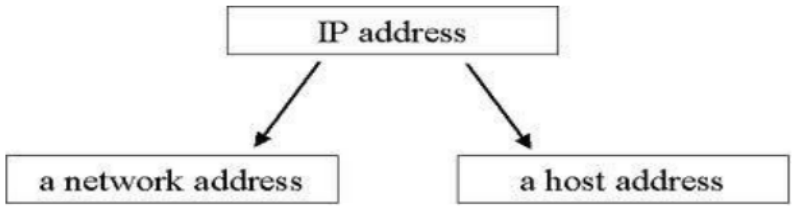
## 2.1. IPv4 დამისამართება

ნებისმიერი ორი სისტემის საკომუნიკაციოდ, ეს სისტემები უნდა იყოს იდენტიფიცირებული ქსელში.




*სურ.2.1. 1*

- კომპიუტერი უნდა იყოს დაკავშირებული არა უმცირეს ერთ ქსელთან, შესაბამისად მას უნდა ქონდეს არა უმცირეს ერთი მისამართი.
- ყოველი მისამართი იდენტიფიცირებულია სხვადასხვა ქსელში.
- IP მისამართი ლოგიკურად გაყოფილია ორ ნაწილად, ქსელის მისამართი და ჰოსტის მისამართი.

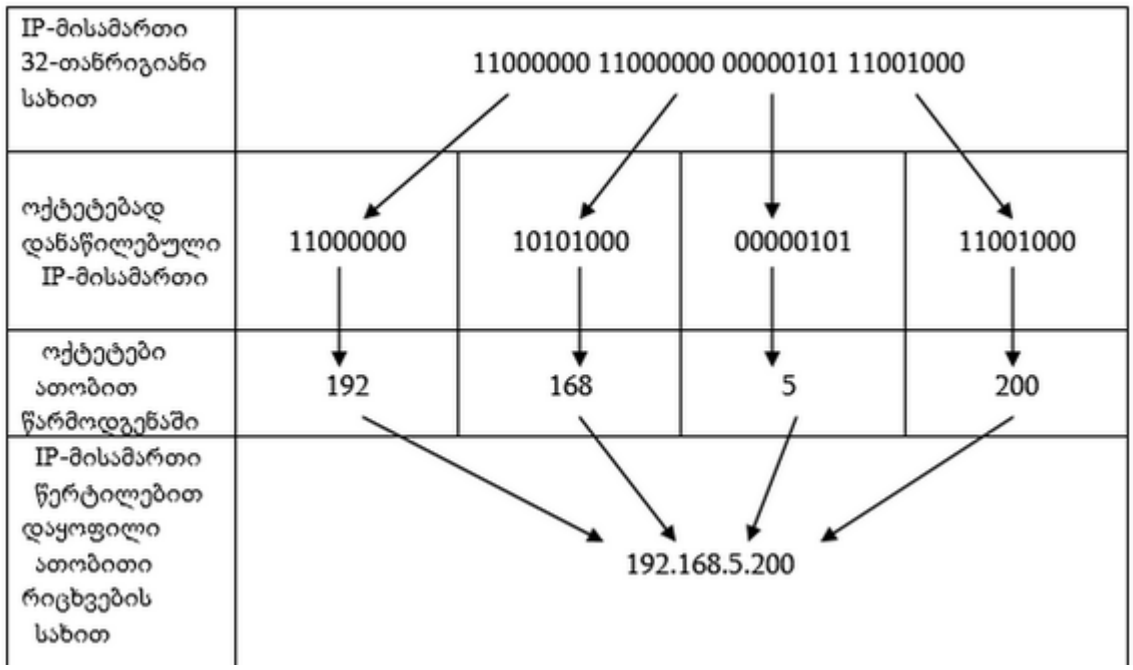



*სურ.2.1. 2*

- ქსელის მისამართისა და ჰოსტის მისამართის კომბინაცია იძლევა უნიკალურ ლოგიკურ IP მისამართს ყოველი მოწყობილობისათვის ქსელში.

- ეს მისამართი მუშაობს ქსელურ დონეზე და იძლევა შესაძლებლობას ერთმა კომპიუტერმა აღმოაჩინოს მეორე კომპიუტერი ქსელში.
- ყველა კომპიუტერს აგრეთვე გააჩნია უნიკალური ფიზიკური მისამართი, რომელიც ცნობილია როგორც MAC მისამართი. ეს მისამართი ენიჭება მწარმოებელი ფირმის მიერ ქსელურ ადაპტერს.

**IPv4 მისამართი 32 ბიტიანი რიცხვი**




*სურ.2.1. 3*

- IP მისამართი არის 32 ბიტიანი რიცხვი.
- IP მისამართებთან ადვილად სამუშაოდ, მთლიანი მისამართი დაყოფილია 4 ნაწილად, ანუ 4 ბაიტად წერტილების საშუალებით და წარმოდგენილია ათობითი ფორმატით.
- მაგალითად ერთი კომპიუტერის IP მისამართი შეიძლება იყოს
  - 192.168.7.1
  - მეორესი 172.16.2.2.
- ყოველი ნაწილს უწოდებენ ოქტეტებს, რადგანაც თითოეულში შედის 8 ბიტი.

- მაგალითად IP მისამართი 192.168.1.8 ორობით ფორმატში იქნება 11000000.10101000.00000001.00001000
- IP მისამართი შედგება ორი ნაწილისგან, ერთი აღნიშნავს ქსელის მისამართს მეორე ჰოსტის მისამართს.
- ყოველი ოქტეტი იცვლება 0 დან 255- მდე.
- ასეთი სისტემის მისამართებს იერარქიულ მისამართებსაც უწოდებენ, რადგან ისინი შედგება ორი ნაწილისაგან, ჯამში ციფრი უნდა იყოს უნიკალური, წინააღმდეგ შემთხვევაში შეუძლებელი გახდება მარშუტიზაცია.

**IP მისამართები იყოფა კლასებად**

| Class A | Network | Host | | |
|---|---|---|---|---|
| Octet | 1 | 2 | 3 | 4 |

| Class B | Network | | Host | |
|---|---|---|---|---|
| Octet | 1 | 2 | 3 | 4 |

| Class C | Network | | | Host |
|---|---|---|---|---|
| Octet | 1 | 2 | 3 | 4 |

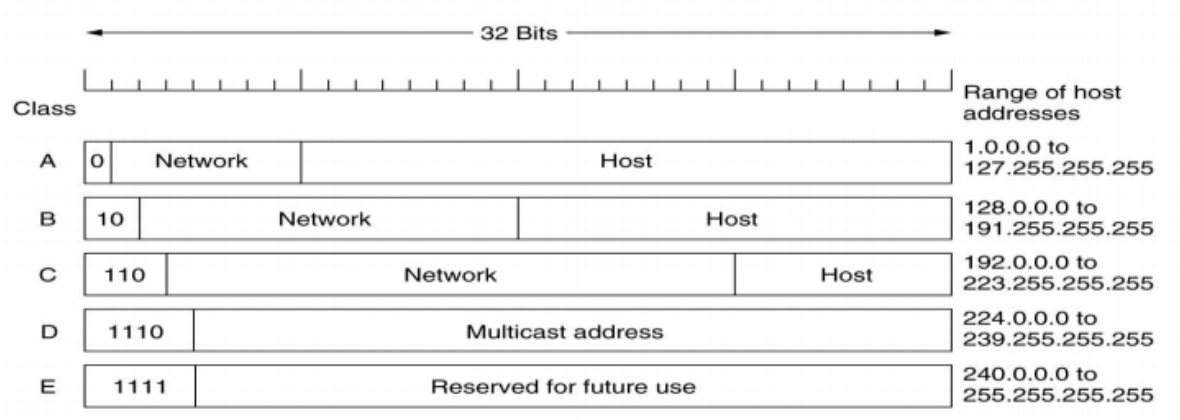
| Class D | Host | | | |
|---|---|---|---|---|
| Octet | 1 | 2 | 3 | 4 |

მეხუთე კლასი E გამოიყენება ექსპერიმენტული მიზნებისთვის.

*სურ.2.1. 4*

ყოველი 32 ბიტიანი IP მისამართი იყოფა ქსელის და ჰოსტის ნაწილად, პირველი ბიტი ან ბიტების ჯგუფი განსაზღვრავს მისამართების კლასს

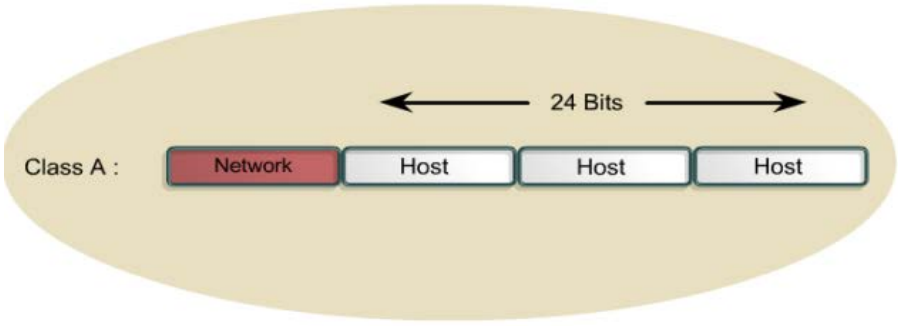
A კლასის მისამართები ენიჭება დიდ ქსელებს, B კლასის მისამართები საშუალო ზომის ქსელებს, ხოლო C კლასის მისამართები მცირე ზომის ქსელებს




*სურ.2.1. 5*

**A კლასი**

გამოიყენება დიდი ქსელების დასამისამართებლად

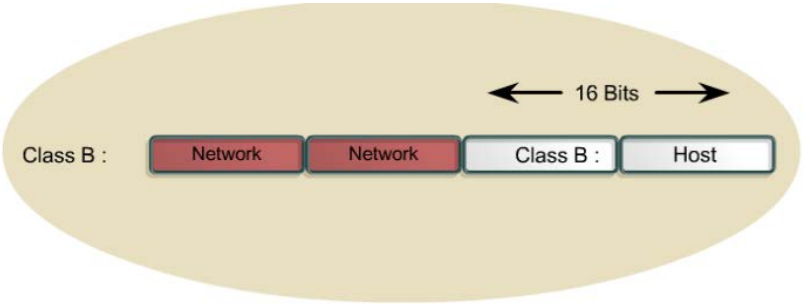



*სურ.2.1. 6*

- ერთი A კლასის ქსელი დაახლოებით შეიცავს 16 მილიონი ჰოსტის მისამართს.
- ამ კლასში ქსელის მისამართი არის პირველი ოქტეტი, დანარჩენი 3 ოქტეტი არის ამ ქსელში ჰოსტის მისამართი.
- აქედან გამომდინარე ყველაზე მცირე რაოდენობით არის A კლასის ქსელები, მაგრამ თითოეულში ჰოსტების მისამართების დიდი რაოდენობით.
- პირველი ბიტი A კლასის მისამართში ყოველთვის არის 0
- ყველაზე დაბალი რიცხვი რომელიც პირველი ბიტის 0 ის არსებობის შემთხვევაში არის ორობითში- 00000000. ათობითში - 0.
- უდიდესი რიცხვი კი 01111111, ათობითში 127.

- რიცხვები 0 და 127 არის რეზერვირებული და არ გამოიყენება ქსელის მისამართად.
- დანარჩენი მისამართები კი 1 დან 126 წარმოადგენს A კლასის ქსელის მისამართებს.
- 127.0.0.0 ქსელი არის რეზერვირებული loopback ტესტირებისთვის.

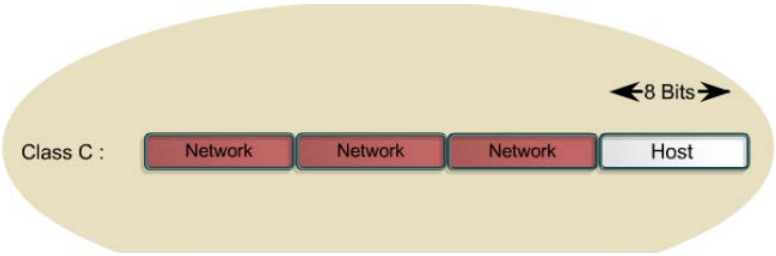
### B კლასი




*სურ.2.1. 7*

- B კლასის მისამართები შეიქმნა საშუალო ზომის ქსელების დასამისამართებლად.
- B კლასის მისამართი იყენებს პირველ ორ ოქტეტს ქსელის დასამისამართებლად, ხოლო დანარჩენ ორ ოქტეტს ჰოსტების დასამისამართებლად
- პირველი ორი ბიტი B კლასის მისამართში არის 10.
- დანარჩენი 6 ბიტი ივსება 0 და 1 –ით.
- ყველაზე დაბალი რიცხვი წარმოდგენილია 10000000. ათობითში არის 128.
- უდიდესი რიცხვი კი 10111111, ათობითში არის 191.
- B კლასის ქსელის მისამართების პირველი ოქტეტი მოქცეულია დიაპაზონში 128-191.

### C კლასი




*სურ.2.1. 8*

- C კლასის მისამართები არის ყველაზე გამოყენებადი.
- ის უზრუნველყოფს მცირე ქსელების დამისამართებას, მაქსიმუმ 254 ჰოსტი.
- C კლასის მისამართები იწყება 110. • ყველაზე დაბალი რიცხვი წარმოდგენილია 11000000. ათობითში არის 192.
- უდიდესი რიცხვი კი 11011111, ათობითში არის 223. • B კლასის ქსელის მისამართების პირველი ოქტეტი მოქცეულია დიაპაზონში 192- 223.

### D კლასი

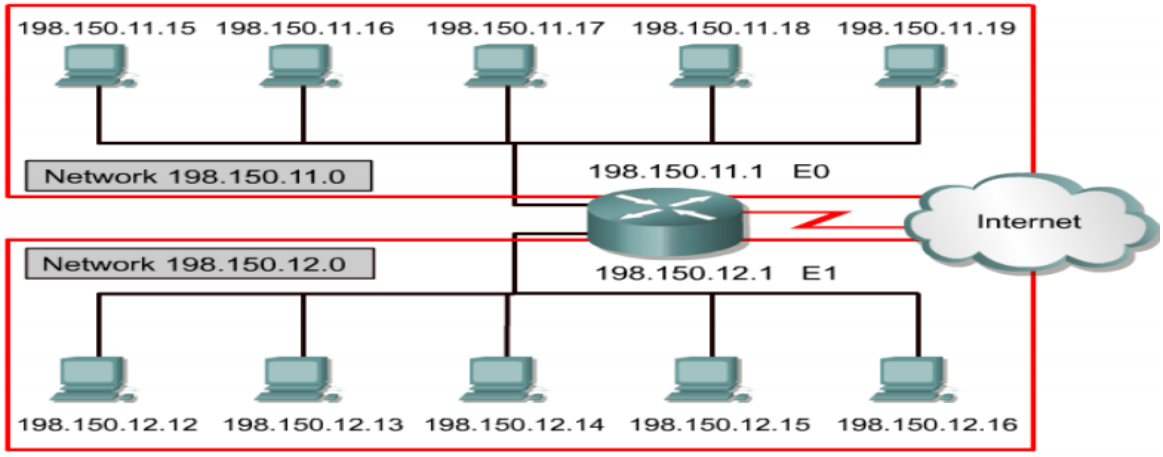
- D კლასის მისამართები გამოიყენება მულტიმაუწყებლობისთვის ანუ ერთდროული შეტყობინებების დასაგზავნად.
- პირველი ოთხი ბიტი იწყება 1110, უმცირესი რიცხვი არის 11100000 ხოლო უდიდესი 11101111.
- ანუ 224 და 239.

| IP address class | IP address range (First Octet Decimal Value) |
|---|---|
| Class A | 1-126 (00000001-01111110) * |
| Class B | 128-191 (10000000-10111111) |
| Class C | 192-223 (11000000-11011111) |
| Class D | 224-239 (11100000-11101111) |
| Class E | 240-255 (11110000-11111111) |

*სურ.2.1. 9*

არსებობს რამოდენიმე მისამართი რომელიც არ შეიძლება მინიჭებული იქნას ჰოსტზე. ასეთი მისამართია **ქსელის მისამართი**. ქსელის მისამართი განსაზღვრავს მთლიანად ქსელს

სურათზე 2.1.10 ზედა ოთკუთხედში წარმოდგენილია 198.150.11.0 ქსელი, მონაცემები რომლებიც იგზავნება ნებისმიერი ჰოსტიდან (198.150.11.1- 198.150.11.254) ქსელში გარედან ჩანს, როგორც **198.150.11.0 ქსელი.** ქვედა ოთკუთხედშიც მოცემულია იგივე ქსელის სტრუქტურა რაც ზედა ოთკუთხედში , ოღონდ განსხვავებულია მხოლოდ ქსელის მისამართი **198.150.12.0**

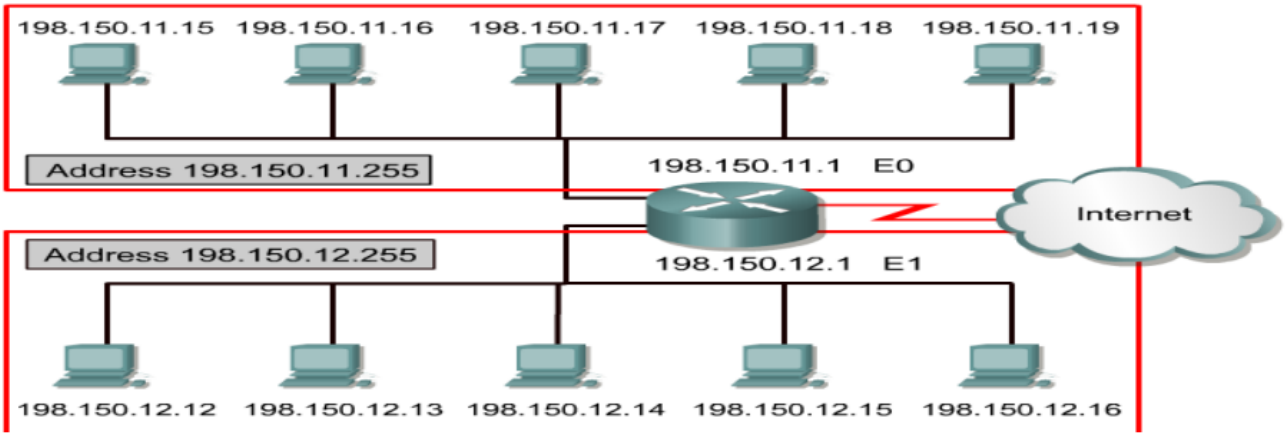



*სურ.2.1. 10*

**IP მისამართი, რომლის ჰოსტისთვის განკუთვნილ სამისამართო ბიტებში წერია 0 , ასეთი მისამართი რეზერვირებულია ქსელის მისამართად.**

- მაგალითად A კლასის ქსელში, მისამართი 113.0.0.0 არის ქსელის IP მისამართი, მარშუტიზატორი სწორედ ამ მისამართებს იყენებს როდესაც ის იღებს გადაწყვეტილებას პაკეტების მარშუტიზაციის დროს.
- B კლასის ქსელის მისამართში პირველი ორი ოქტეტი არის ქსელის მისამართი. ბოლო ორი ოქტეტი კი შეიცავს 0-ებს, ეს 16 ბიტი არის ჰოსტისთვის განკუთვნილი ანუ იმ მოწყობილობების დასამისამართებლად რომლებიც მიერეთებული იქნებიან ამ ქსელში. B კლასის ქსელში მაგალითად ქსელის IP მისამართი შეიძლება იყოს 176.10.0.0
- C კლასის ქსელის მისამართში პირველი სამი ოქტეტი არის ქსელის მისამართი. ბოლო ოქტეტი კი შეიცავს 0-ებს, ეს 8 ბიტი არის ჰოსტისთვის განკუთვნილი ანუ იმ მოწყობილობების დასამისამართებლად რომლებიც მიერეთებული იქნებიან ამ ქსელში. C კლასის ქსელში მაგალითად ქსელის IP მისამართი შეიძლება იყოს 192.168.5.0

მეორე მისამართი რომელიც არ შეიძლება იქნას მინიჭებული ჰოსტზე არის ფართომაუწყებლობითი მისამართი(**Broadcast address)**




*სურ.2.1. 11*

სურათზე 2.1.11, ზედა ოთკუთხედში წარმოდგენილია მისამართი **198.150.11.255** . რომელიც წარმოადგენს ფართომაუწყებლობით მისამართს. მონაცემები რომლებიც იგზავნება ამ მისამართზე გადაეცემა ყველა ამ ქსელში ჩართულ ჰოსტს(198.150.11.1-198.150.11.254) და დამუშავდება მათ მიერ.

- ✓ ინფორმაციის ყველა ქსელში ჩართული ჰოსტისთვის ერთდროულად გასაგზავნად გამოიყენება - ფორმირდება ფართომაუწყებლობითი მისამართი. რომ მივიღოთ აღნიშნული მისამართი ჰოსტისთვის განკუთვნილ სამისამართო ბიტებში იწერება 1-ები. მაგალითად ფართომაუწყებლობითი მისამართი არის 176.10.255.255.

ტესტის ნიმუში

***14.0.255.255** მოცემულ მისამართთან მიმართებაში რომელი მსჯელობაა მცდარი?*

- მოცემული მისამართი Broadcast(ფართომაუწყებლობითი) მისამართია
- მოცემული მისამართი არ არის ქსელის (Network) მისამართია
- მოცემულ IP მისამართის შესაბამის ქსელში კვანძს(Host) ეკუთვნის 3 ბაიტი
- მოცემულ IP მისამართის შესაბამის ქსელში ჰოსტს ეკუთვნის 24 ბიტი

***14.0.255.255** მოცემულ მისამართთან მიმართებაში რომელი მსჯელობაა სწორი?*

- მოცემული მისამართი Broadcast(ფართომაუწყებლობითი) მისამართია
- არც ერთი პასუხი არ არის სწორი
- მოცემულ IP მისამართის შესაბამის ქსელში შესაძლებელია 254 ჰოსტის ჩართვა
- მოცემულ IP მისამართის შესაბამის ქსელში ქსელს ეკუთვნის 24 ბიტი

***Gateway** მისამართთან მიმართებაში რომელი მსჯელობაა მცდარი?*

- Gateway მისამართი საერთო მისამართია ერთ ქსელში ჩართული კომპიუტერებისთვის
- Gateway მისამართი არ მიეკუთვნება დარეზერვებულ მისამართთა ჯგუფს
- Gateway მისამართში ქსელისა და ჰოსტის ბიტები არ არის აუცილებელი ერთმანეთს ემთხვეოდეს
- არც ერთი პასუხი არ არის სწორი

*ჩამოთვლილთაგან რომელი შეიძლება მიენიჭოს კვანძს **(Host)**?*

- 192.168.14.0

- 10.0.0.255
- 189.25.0.0
- 137.0.255.255

*მოცემულთაგან რომელი არ არის შიდა ლოკალური ქსელისთვის დარეზერვებული სივრცის მისამართი?*

- 172.30.12.254
- 10.0.0.255
- 192.168.1.3
- 117.15.17.9

*რომელი მსჯელობაა სწორი?*

- სხვადასხვა ქსელში ჩართულ კომპიუტერებს ერთი და იმავე Gateway მისამართი აქვს
- IP მისამართის მინიჭება შესაძლებელია მხოლოდ სტატიკურად
- IP მისამართი ქსელში ჩართული კომპიუტერის ლოგიკური მისამართია
- ყველა პასუხი სწორია

## 2.2. IPv6 დამისამართება

რატომ IPv6?

- IPV4-მისამართის 32 ბიტი თეორიულად 232, ანუ 4294967296 მისამართს იტევს ( 4.3 მილიარდამდე) - ბოლო წლებში მომრავლებული ქსელური მოწყობილობების წყალობით IP-მისამართების დეფიციტი მივიღეთ
- IPv6 128 ბიტ-ინფორმაციას შეიცავს და კოლოსალური რაოდენობის ქსელური მოწყობილობების დამისამართება შეუძლია( $2^{128}$ რაც ოცდაცხრამეტნიშნა რიცხვია!)

| Number Name | Scientific Notation | Number of Zeros |
|---|---|---|
| 1 Thousand | 10^3 | 1,000 |
| 1 Million | 10^6 | 1,000,000 |
| 1 Billion | 10^9 | 1,000,000,000 |
| 1 Trillion | 10^12 | 1,000,000,000,000 |
| 1 Quadrillion | 10^15 | 1,000,000,000,000,000 |
| 1 Quintillion | 10^18 | 1,000,000,000,000,000,000 |
| 1 Sextillion | 10^21 | 1,000,000,000,000,000,000,000 |
| 1 Septillion | 10^24 | 1,000,000,000,000,000,000,000,000 |
| 1 Octillion | 10^27 | 1,000,000,000,000,000,000,000,000,000 |
| 1 Nonillion | 10^30 | 1,000,000,000,000,000,000,000,000,000,000 |
| 1 Decillion | 10^33 | 1,000,000,000,000,000,000,000,000,000,000,000 |
| 1 Undecillion | 10^36 | 1,000,000,000,000,000,000,000,000,000,000,000,000 |

*სურ.2.2. 1*

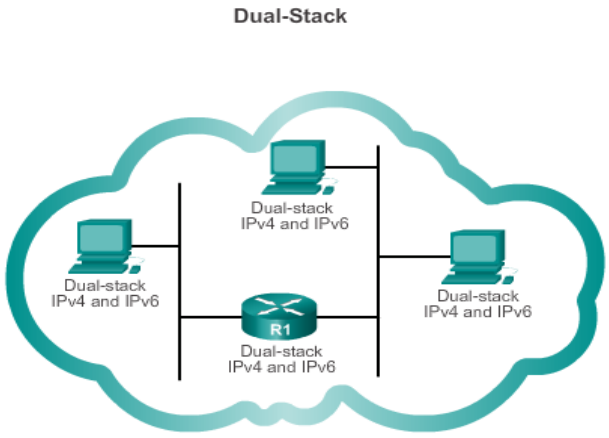
ახლო მომავალში, ორივე - IPv4 და IPv6 სტანდარტი იქნება თანაარსებობის პირობებში.

პრობლემა ისაა, რომ IPv6 ძველ მოწყობილობებთან არათავსებადია. თუ ერთ მშვენიერ დღეს, მთელი ინტერნეტი პროტოკოლის ახალ ვერსიაზე გადავა, ყველას ძველი ქსელური მოწყობილობის ახლით შეცვლა მოუწევს

IETF ქმნის სხვადასხვა პროტოკოლებს, რათა დაეხმაროს ქსელის ადმინისტრატორებს მოახდინონ მათი ქსელების მიგრაცია IPv6-ში.

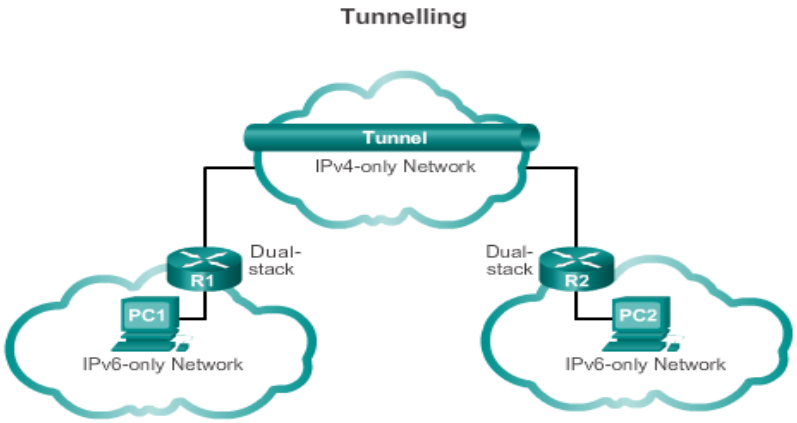
**Ipv4-დან Ipv6-ზე გადასვლის კატეგორიები**

1. **Dual-Stack** - მოწყობილობებს Dual-Stack მხარდაჭერით, ძალუძთ ერთს და იმავე ქსელში ორივე ოქმის გამოყენებაDual




*სურ.2.2. 2*

2. **Tunnelling** - Tunnelling მეთოდის დროს ხდება - IPV4 პაკეტის შიგნით IPV6 პაკეტის ინკაპსულირება




*სურ.2.2. 3*

3. **Translation** - მოცემული მეთოდის დროს ხდება - IPV6 პაკეტის გარდაქმნა IPV4-ში და პირიქით


*სურ.2.2. 4*

## IPv6 მისამართის წარმოდგენის ფორმატი

- IPV6 მისამართები წარმოდგენილია 16-ით ფორმატში

**Representing Hexadecimal Values**

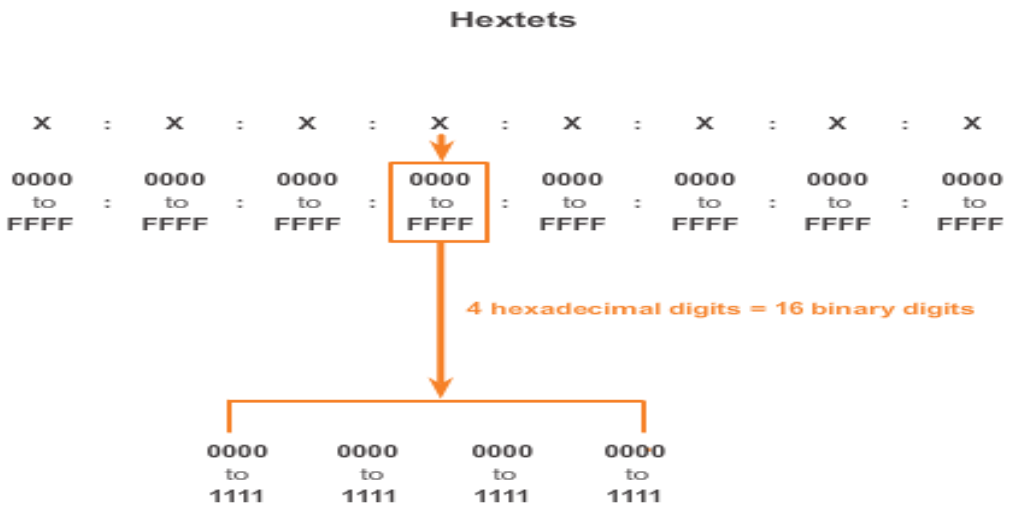
| Hexadecimal | Decimal | Binary |
|---|---|---|
| 0 | 0 | 0000 |
| 1 | 1 | 0001 |
| 2 | 2 | 0010 |
| 3 | 3 | 0011 |
| 4 | 4 | 0100 |
| 5 | 5 | 0101 |
| 6 | 6 | 0110 |
| 7 | 7 | 0111 |
| 8 | 8 | 1000 |
| 9 | 9 | 1001 |
| A | 10 | 1010 |
| B | 11 | 1011 |
| C | 12 | 1100 |
| D | 13 | 1101 |
| E | 14 | 1110 |
| F | 15 | 1111 |

**Hexadecimal Conversions of Binary Octets**

| Hexadecimal | Decimal | Binary |
|---|---|---|
| 00 | 0 | 0000 0000 |
| 01 | 1 | 0000 0001 |
| 02 | 2 | 0000 0010 |
| 03 | 3 | 0000 0011 |
| 04 | 4 | 0000 0100 |
| 05 | 5 | 0000 0101 |
| 06 | 6 | 0000 0110 |
| 07 | 7 | 0000 0111 |
| 08 | 8 | 0000 1000 |
| 0A | 10 | 0000 1010 |
| 0F | 15 | 0000 1111 |
| 10 | 16 | 0001 0000 |
| 20 | 32 | 0010 0000 |
| 40 | 64 | 0100 0000 |
| 80 | 128 | 1000 0000 |
| C0 | 192 | 1100 0000 |
| CA | 202 | 1100 1010 |
| F0 | 240 | 1111 0000 |
| FF | 255 | 1111 1111 |

*სურ.2.2. 5*

- IPV6 როგორც 128 ბიტიანი მისამართი, წარმოდგენილია 8 ჰექსტეტის სახით;
  - თითოეულ ჰექსტეტში გვაქვს 4 16-ითის სიმბოლო რაც წარმოადგენს 16 ბიტიან რიცხვს ორობითში;
  - ჰექსტეტები : (2 წერტილით) გამოიყოფა




*სურ.2.2. 6*

- ჰექსტეტებში შესაძლებელია ჩაწერის დროს 0-ის გამოტოვება

| | |
|---|---|
| Preferred | 2001:0DB8:0000:1111:0000:0000:0000:0200 |
| No leading 0s | 2001: DB8: 0:1111: 0: 0: 0: 200 |

*სურ.2.2. 7*

- მიჯრით განლაგებული მთლიანად **0-ის** შემცველი ჰექსტეტების ნაცვლად შეგვიძლია ჩავწეროთ **::**

| | |
|---|---|
| Preferred | 2001:0DB8:0000:0000:ABCD:0000:0000:0100 |
| No leading 0s | 2001: DB8: 0: 0:ABCD: 0: 0: 100 |
| Compressed | 2001:DB8::ABCD:0:0:100 |
| or | |
| Compressed | 2001:DB8:0:0:ABCD::100 |

Only one :: may be used.

*სურ.2.2. 8*

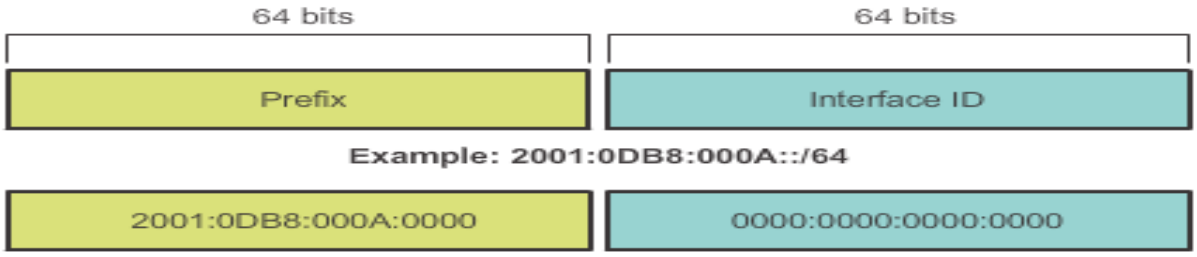
## IPV6 მისამართების ტიპები

**Unicast** - აღწერს IPV6-თავსებადი მოწყობილობის ინტერფეისს (პაკეტი გაგზავნილი ამგვარ მისამართზე მიუვა მხოლოდ(ერთადერთ) შესაბამის ინტერფეისს)

**Multicast** – IPV6 მისამართი, რომელიც გამოიყენება ერთი და იმავე პაკეტის რამოდენიმე მიმართულებით(Destination) დაგზავნისათვის (პაკეტი გაგზავნილი ამგვარ მისამართზე მიუვა ყველა იმ ინტერფეისს, რომელიც მიბმულია მრავალმისამართიანი დაგზავნის ჯგუფს)

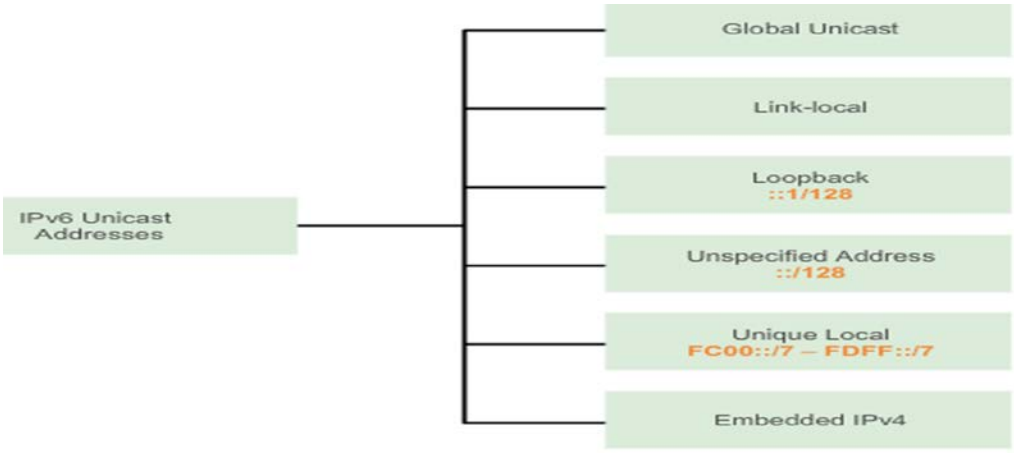
**Anycast** – Unicast IPV6 მისამართი, რომელიც შესაძლებელია მიენიჭოს რამოდენიმე კვანძს(Host). დაგზავნისას პაკეტი მიუვა ამ მისამართის მქონე უახლოეს კვანძს(Host) (პაკეტი გაგზავნილი ამგვარ მისამართზე მიუვა მარშრუტიზატორის მეტრიკით განსაზღვრულ უახლოეს კვანძს, მოცემული მისამართი შესაძლებელია გამოყენებულ იქნას მხოლოდ მარშრუტიზატორებში)

**IPv6 მისამართში პრეფიქსით გამოიყოფა ქსელური და ჰოსტის ნაწილები,** უმრავლეს ქსელებში გავრცელებულია /64 პრეფიქსი, რაც ნიშნავს -, რომ 128 ბიტიანი მისამართიდან 64 ბიტი ეკუთვნის ქსელს და დანარჩენი 64 ჰოსტს



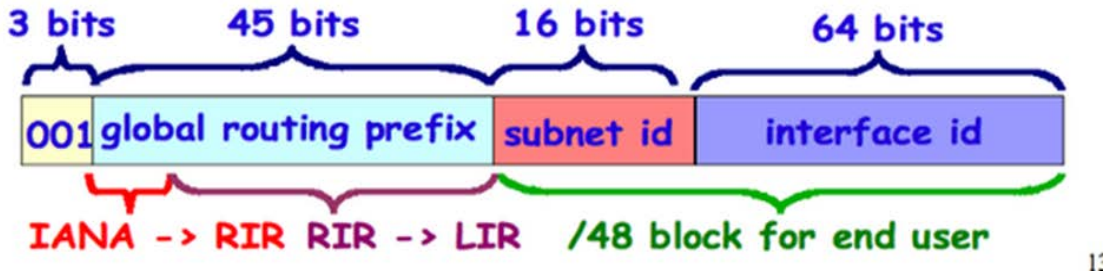

*სურ.2.2. 9*

**IPv6-ში Unicast მისამართების 6 ტიპი არსებობს**




*სურ.2.2. 10*

**Global Unicast მისამართი-** Global unicast მისამართი public IPv4 (გლობალური IP) მისამართის ანალოგურია ამგვარი მისამართები უნიკალურია გლობალური მასშტაბით, მოცემული მისამართები შეიძლება კონფიგურირებულ იყოს სტატიკურად ამ მიენიჭოს დინამიურად

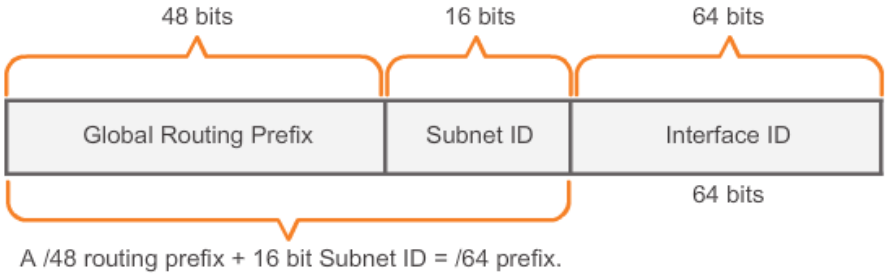



*სურ.2.2. 11*

ამჟამად პირველ 3 ბიტში მხოლოდ 001 ან 2000::/3 Global unicast მისამართებია გამოყენებაში

- Global unicast მისამართი შედგება 3 ნაწილისგან
  - Global routing prefix - ქსელის მისამართი, რომელიც პროვაიდერის მიერ მიენიჭება მომხმარებელს
  - Subnet ID - გამოიყენება ორგანიზაციების მიერ მის ქსელში არსებული ქვექსელების იდენტიფიცირებისათვის
  - Interface ID - ანალოგიურია რაც ჰოსტის ნაწილი IPv4-ში

➢ 2001:0DB8::/32 მისამართი დარეზერვებულია




*სურ.2.2. 12*

**Link-local მისამართი**

– ამგვარი მისამართები მიკუთვნებულია ერთი ცალკეული ქვექსელისთვის (Link)

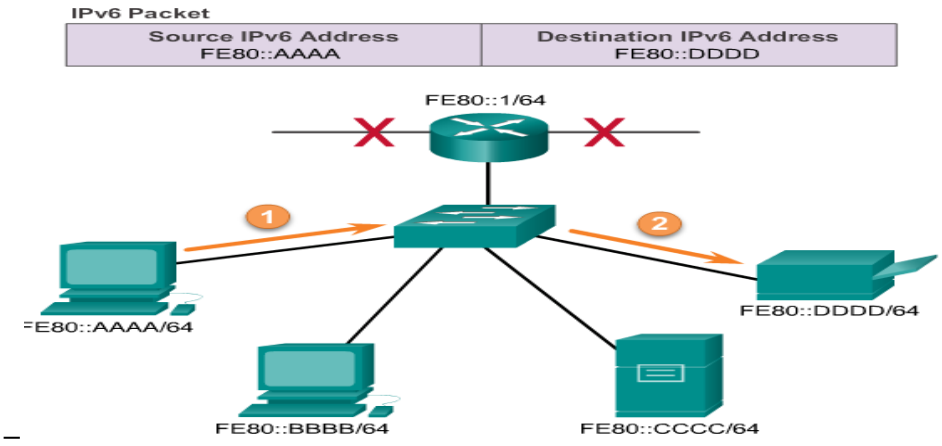
– მოცემული მისამართები შესაძლებელია შევადაროთ APIPA პროტოკოლით კონფიგურირებულ მისამართებს IPv4 სტანდარტში

–მოცემული მისამართების უნიკალურობა ვრცელდება შესაბამის ქვექსელზე, რამეთუ მარშრუტიზატორები(Routers) არ გადაამისამართებენ Link-Local მისამართის მქონე წყაროდან ან Link-Local მისამართზე მიმავალ პაკეტებს

– გამოიყენება:

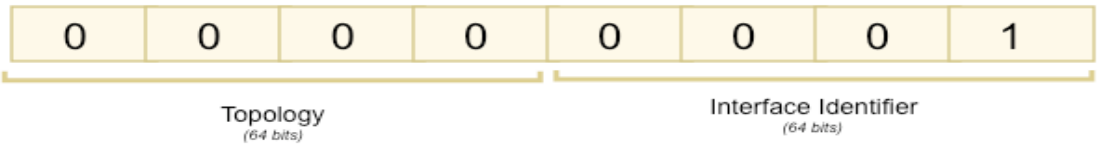
- როგორც წყაროს მისამართი მარშრუტიზატორების აღმოსაჩენად(RS და RA შეტყობინებები)
- იმავე ქსელში ჩართული კვანძების აღმოსაჩენად
- როგორც next-hop მისამართი

–მოცემული მისამართების დიაპაზონია FE80::/10 ანუ პირველი 10 ბიტია - 1111 1110 10**00 0000** (FE80) -დან 1111 1110 10**11 1111**(FEBF)-მდე.




*სურ.2.2. 13*

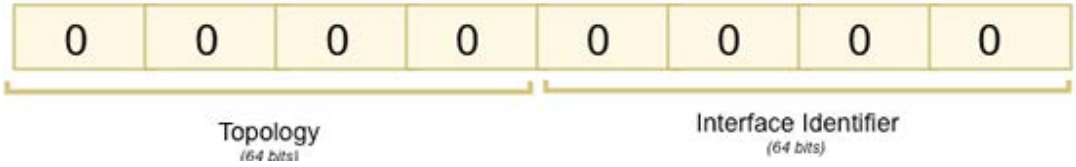
### Loopback მისამართი




*სურ.2.2. 14*

- Loopback მისამართი კვანძის(Host) მიერ, პაკეტების თვითდაგზავნისთვის გამოიყენება და ამიტომაც შეუძლებელია მიენიჭოს ფიზიკურ ინტერფეისს
- IPv6 Loopback მისამართი წარმოადგენს ყველა ნულს გარდა ერთი ბოლო ბიტისა და გამოისახება ამგვარად:
  - ✓ 0:0:0:0:0:0:0:1/128
  - ✓ ::1/128 ან ::1 შეკუმშულ ფორმატში
    - IPv6 Loopback მისამართი IPv4-ში 127.0.0.1/8 მისამართის ანალოგიურია

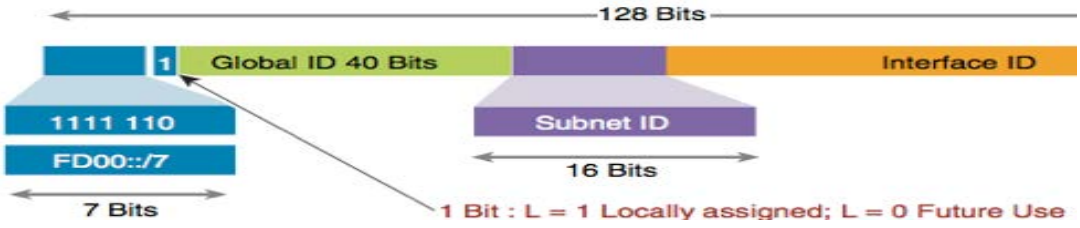
**Unspecified address (დაუზუსტებელი) მისამართი**




*სურ.2.2. 15*

- მისამართი მთლიანად 0-სგან არის წარმოდგენილი
  - ::/128 ან ::
- Unspecified address მისამართი შეუძლებელია მიენიჭოს ინტერფეისს და შესაძლებელია გამოყენებულ იყოს მხოლოდ როგორც ინფორმაციის წყაროს მისამართი IPv6 პაკეტში
  - ის წარმოჩინდება როგორც წყაროს მისამართი, როდესაც მოწყობილობას ჯერ კიდევ არ აქვს მუდმივი IPv6 მისამართი ან როცა პაკეტის წყარო - დანიშნულების მისამართის არარელევანტურია

**Unique local უნიკალური ლოკალური მისამართი**




*სურ.2.2. 16*

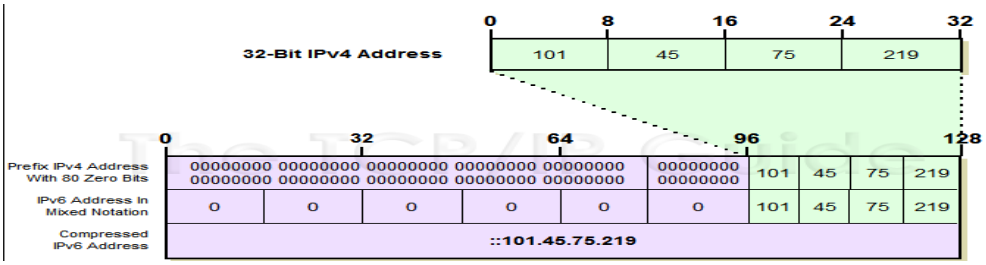
- Unique local addresses(უნიკალური ლოკალური მისამართი) გამოიყენება ლოკალური დამისამართებისათვის
  - ეს მისამართები არ ექვემდებარება მარშუტიზაციას გლობალურ IPv6-ში.

– Unique local addresses (უნიკალური ლოკალური მისამართი) დიაპაზონია **FC00::/7 - FDFF::/7**

- ამ მისამართების გამოყენების აქტუალობაა შიდა მოწყობილობის დამალვა(დაცვა) ინტერნეტის ქსელიდან

**IPv4 embedded ჩაშენებული IPv4 მისამართი**

- ჩაშენებული IPv4 მისამართი გამოიყენება IPv4 მისამართის - IPv6 მისამართში გადასაყვანად




*სურ.2.2. 17*

## SLAAC & DHCPv6

მოწყობილობებს შეუძლიათ ავტომატურად მიიღონ IPv6 global unicast address მისამართი, შემდეგი 2 გზით:

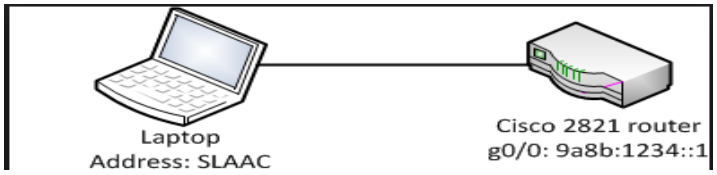
– Stateless Address Autoconfiguration (SLAAC)

– DHCPv6

Stateless Address Autoconfiguration (SLAAC) - არის მეთოდი, როდესაც მოწყობილობები იღებენ prefix, prefix length და default gateway address ინფორმაციას IPv6 მარშრუტიზატორიდან(router) DHCPv6 სერვერის გამოყენების გარეშე.

იყენებენ რა SLAAC, მოწყობილობები ემყარებიან ლოკალური როუტერების ICMPv6 Router Advertisement (RA) შეტყობინებებს სათანადო ინფორმაციის მისაღებად

მარშრუტიზატორებზე IPv6 Routing-ი Default არ არის ნებადართული და საჭიროა გააქტიურება




*სურ.2.2. 18*

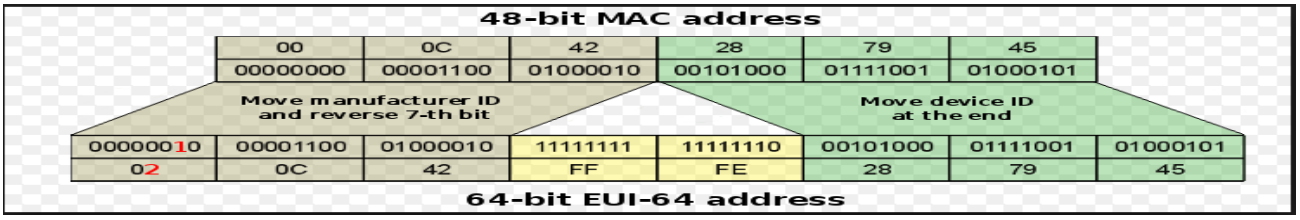
## RA შეტყობინება

- **Option 1 - SLAAC Only** – ჰოსტი იღებს prefix, prefix-length, და default gateway address ინფორმაციას, რომელსაც შეიცავს RA message. სხვა ინფორმაციის მიღება შესაძლებელია DHCPv6 სერვერიდან.
- **Option 2 – SLAAC and DHCPv6** – RA შეტყობინებიდან მიღებული მისამართების გარდა, ჰოსტი DHCPv6 სერვერისგან იღებს დამატებით ინფორმაციას, მაგ.: DNS სერვერის მისამართს.
- **Option 3 – DHCPv6 only** – ამ შემთხვევაში ჰოსტი არ იყენებს RA შეტყობინებას და სრულ ინფორმაციას იღებს DHCPv6 სერვერისგან - IPv6 global unicast address, prefix length, a default gateway address, and the addresses of DNS servers

## The Interface ID

თუ კლიენტი კომპიუტერი არ იყენებს RA შეტყობინების შემცველ ინფორმაციას და ეყრდნობა უშუალოდ DHCPv6-ს, სერვერი მიაწვდის მთლიან Unicast გლობალურ მისამართს, პრეფიქსის და ინტერფეისის(ჰოსტის) იდენტიფიკატორის ჩათვლით

- თუ გამოიყენება ვარიანტი 1(მხოლოდ SLAAC) ან ვარიანტი 2 (SLAAC და DHCPv6), კლიენტი ვერ იღებს მთლიანი მისამართის ჰოსტის ნაწილს (Interface ID).
- კლიენტმა მოწყობილობამ უნდა განსაზღვროს 64 ბიტიანი ინტერფეისის იდენტიფიკატორი ან EUI-64 პროცესის ან 64 ბიტიანი რიცხვის შემთხვევივითი გენერირებით.




*სურ.2.2. 19*

## EUI-64 პროცესი

ეს პროცესი იყენებს ჰოსტის 48 ბიტიან Ethernet MAC მისამართს და ჩასვამს დანარჩენ 16 ბიტს 48 ბიტიანი MAC მისამართის შუაში, რათა მიიღოს 64 ბიტიანი Interface ID(ჰოსტის იდენტიფიკატორი).

### EUI-64 ინტერფეისის ID

- EUI-64 ინტერფეისის მისამართი წარმოჩენილია ორობით ფორმატში და შედგება 3 ნაწილისგან:
  - 24-ბიტიანი OUI, ჰოსტის MAC-მისამართიდან, იმ პირობით, რომ მე-7 ბიტი (უნივერსალური / ლოკალური (U / L) ბიტი) იცვლება საპირისპიროთი. ეს ნიშნავს, რომ თუ მე-7 ბიტი 0-ია - მაშინ ის გახდება 1 და პირიქით
  - 16-ბიტიანი მნიშვნელობა FFFE (თექვსმეტობით ფორმატში)
  - 24-ბიტიანი ჰოსტის იდენტიფიკატორი კლიენტი კომპიუტერის MAC მისამართიდან

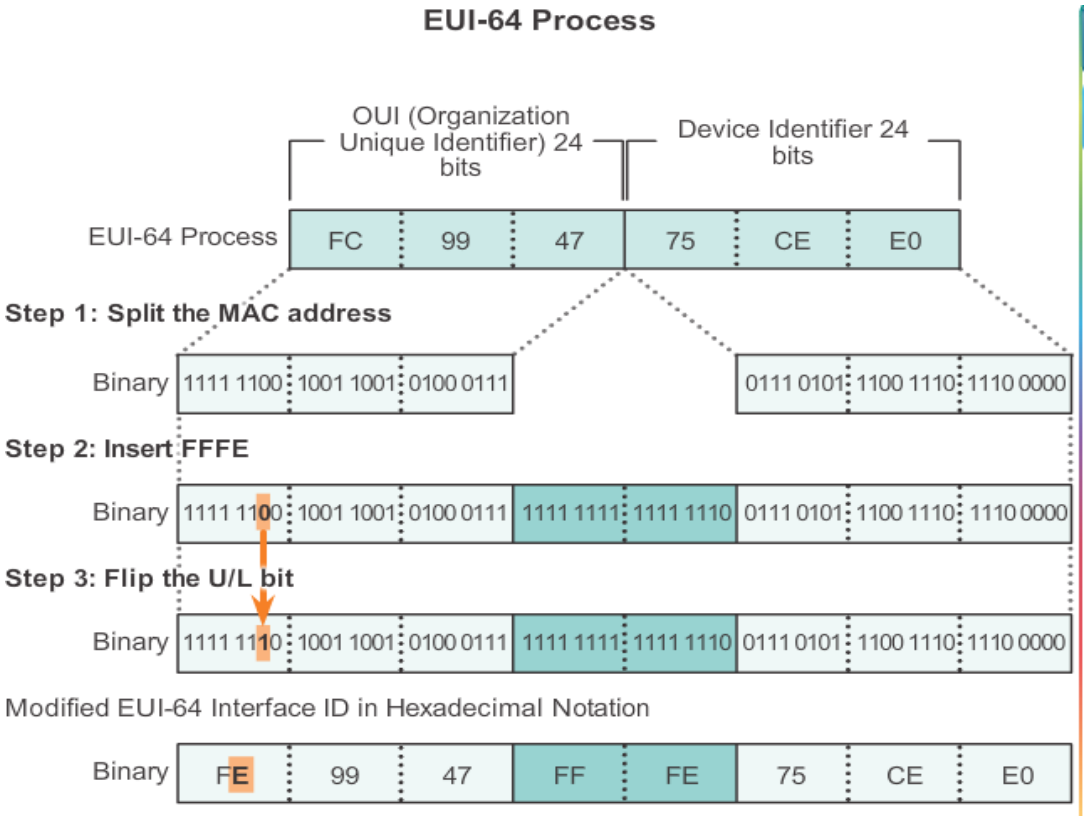
*Ethernet MAC მისამართი*

- Ethernet MAC მისამართი ჩვეულებრივ გამოისახება 16-ით ფორმატში და შედგება 2 ნაწილისგან:
  - Organizationally Unique Identifier (OUI) – OUI არის 24-ბიტიანი (6 თექვსმეტობითი სიმბოლო) მწარმოებლის კოდი, მინიჭებული IEEE სტანდარტიზაციის ორგანოს მიერ.
  - Device Identifier – მოწყობილობის იდენტიფიკატორი არის უნიკალური 24-ბიტიანი (6 თექვსმეტობითი სიმბოლო) მნიშვნელობა, რომელიც მიენიჭება უშუალოდ მწარმოებლის მიერ და OUI-სთან ერთად ქმნის საერთო უნიკალურ მისამართს.

*EUI-64 პროცესის ილუსტრირება **FC99:4775:CEE0 MAC მისამართის** მაგალითზე:*

- ✓ ***ნაბიჯი 1:*** *ხდება MAC მისამართის დაყოფა OUI და მოწყობილობის იდენტიფიკატორ მისამართებად*
- ✓ ***ნაბიჯი 2:*** *თექვსმეტობითი ფორმატის მნიშვნელობის FFFE ჩასმა, ორობითში: 1111 1111 1111 1110*
- ✓ ***ნაბიჯი 3:*** *პირველ ჰექსტეტში(ორობითში გარდაქმილი ფორმით) (უნივერსალური / ლოკალური (U / L) ბიტი) იცვლება საპირისპიროთი ანუ მე-7 ბიტი 0 იცვლება 1-ით.*

შედეგად მივიღებთ EUI-64 პროცესით გენერირებულ ინტერფეისის **ID მისამართს: FE99:47FF:FE75:CEE0.**




*სურ.2.2. 20*

**შემთხვევითად გენერირებული ინტერფეისის მისამართი Interface ID**

- EUI-64 პროცესით მიღებულმა ინტერფეისის იდენტიფიკატორმა მომხმარებლებში გამოიწვია შიში, რომ Mac მისამართის გამოყენებამ შესაძლოა მათი ფიზიკური კომპიუტერების დაუცველობა გამოიწვიოს
- შესაბამისად ალტერნატივად გამოყენებულია მისამართის შემთხვევითი გენერირების პროცესი
- ამა თუ იმ მეთოდით Interface ID-ის ფორმირების შემდეგ ხდება მისი კომბინირება IPv6 prefix-თან და მიიღება - გლობალური ან ლოკალური(Link-Local) IPv6 მისამართი

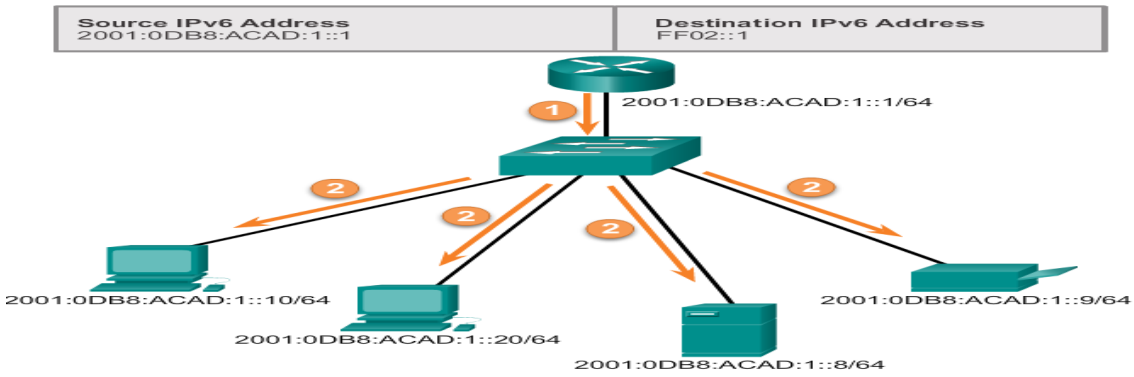
## IPv6 Multicast Addresses

IPv6 multicast მისამართები მსგავსია IPv4 multicast მისამართებისა.:

- IPv6 multicast მისამართებს აქვს პრეფიქსი FF00::/8.
- Multicast მისამართები შეიძლება იყოს მხოლოდ დანიშნულების და არა წყაროს მისამართი
- არსებობს 2 ტიპის IPv6 multicast მისამართი:
    - Assigned(დანიშნული) multicast - ამგვარი მისამართები დარეზერვებულია მოწყობილობათა გარკვეული ჯგუფისთვის
    - Solicited (მოთხოვნილი) multicast - დანიშნული Multicast მისამართი ჩვეულებრივ წარმოადგენს ერთ მისამართს, რომელიც იძლევა იმ მოწყობილობათა ჯგუფთან წვდომის საშუალებას, რომლებიც თავის მხრივ მუშაობს საერთო პროტოკოლით(მაგ.: DHCPv6) ან სერვისით

### Assigned(დანიშნული) multicast

- FF02::1 All-nodes multicast group - კვანძების მულტიკასტ მრავალმისამართიან ჯგუფს შესაძლებელია მიუერთდეს ყველა IPv6 მოწყობილობა
    - IPv6 როუტერი აგზავნის ICMPv6 (Internet Control Message Protocol version 6) RA შეტყობინებას ყველა კვანძის მულტიკასტ ჯგუფთან, რათა მიაწოდოს სამისამართო ინფორმაცია, როგორიცაა: prefix, prefix length და default gateway.



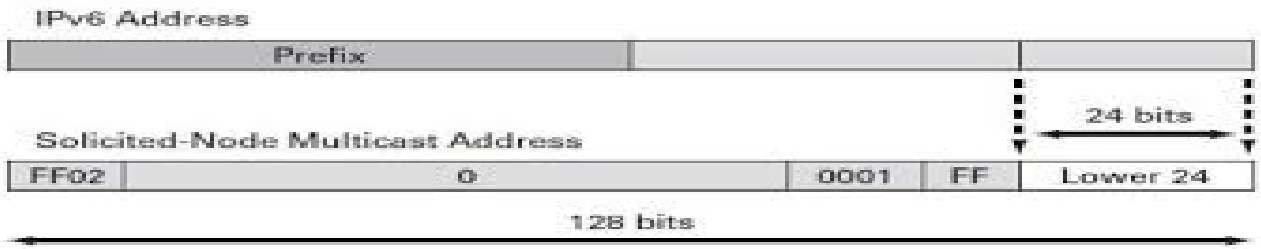

*სურ.2.2. 21*

- ✓ FF02::2 All-routers multicast group - როუტერების მულტიკასტ მრავალმისამართიან ჯგუფს შესაძლებელია მიუერთდეს ყველა IPv6 როუტერი
  - როუტერი ხდება ამ ჯგუფის წევრი, როდესაც ის შესაბამისი ბრძანებით (ipv6 unicast-routing global configuration command) კონფიგურირებულია როგორც IPv6 router

ყველა პაკეტი გაგზავნილი ამ ჯგუფის მისამართზე მიუვა და დამუშავდება სეგმენტის ან ქსელის ყველა IPv6 როუტერის მიერ

**A solicited-node multicast Solicited (მოთხოვნილი) multicast მისამართი**

- მოცემული მისამართი მსგავსია All-nodes multicast მისამართის, რომელიც თავის მხრივ IPv4 Broadcast მისამართის ანალოგურია.
- იმ მოწყობილობათა რიცხვის შესამცირებლად, რომლებმაც უნდა დაამუშაონ დაგზავნილი პაკეტები - გამოიყენება solicited-node multicast მისამართი

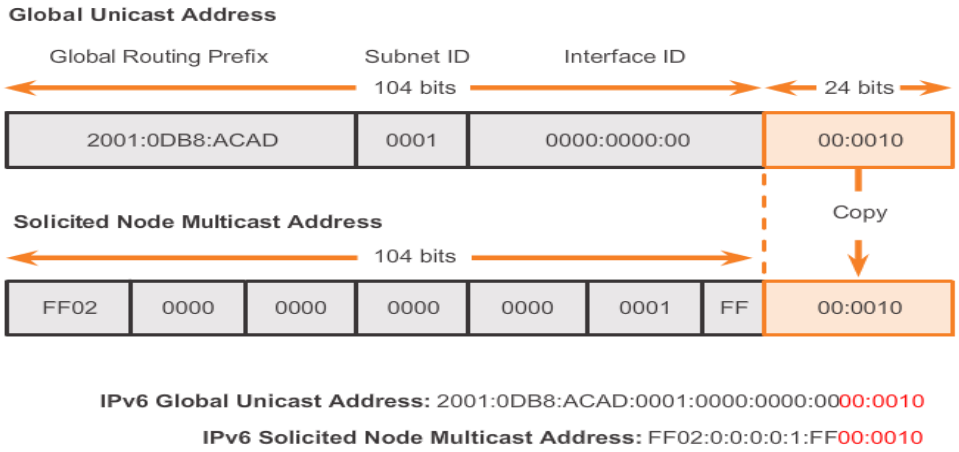



*სურ.2.2. 22*

- solicited-node multicast ის მისამართია, რომელსაც შეესატყვისება მოწყობილობის IPv6 გლობალური unicast მისამართის ბოლო 24 ბიტი. პაკეტებს დაამუშავებენ მხოლოდ

ის მოწყობილობები რომელთა ჰოსტის ნაწილის მისამართში (Interface ID) აქვთ იგივე 24 ბიტის შესაბამისი ჩანაწერი

- The IPv6 solicited-node multicast მისამართი მიიღება FF02:0:0:0:0:1:FF00::/104 პრეფიქსის კომბინირებით - unicast მისამართის ბოლო 24 ბიტთან




*სურ.2.2. 23*

*პრაქტიკული სამუშაო*

შესაბამისი ფაილი ჩამოტვირთეთ ბმულიდან [http://sdrv.ms/1fc0TMN](http://sdrv.ms/1fc0TMN)

*ტესტის ნიმუში:*

***EUI-64 პროცესთან მიმართებაში, რომელი მსჯელობაა სწორი?***

- ეს პროცესი იყენებს ჰოსტის 48 ბიტიან Ethernet MAC მისამართს და ჩასვამს დანარჩენ 16 ბიტს 48 ბიტიანი MAC მისამართის შუაში, რათა მიიღოს 64 ბიტიანი Interface ID(ჰოსტის იდენტიფიკატორი)

- ეს პროცესი იყენებს ჰოსტის 24 ბიტიან Ethernet MAC მისამართს და ჩასვამს დანარჩენ 40 ბიტს 24 ბიტიანი MAC მისამართის შუაში, რათა მიიღოს 64 ბიტიანი Interface ID(ჰოსტის იდენტიფიკატორი)

- ეს პროცესი იყენებს ჰოსტის 48 ბიტიან Ethernet MAC მისამართს და ჩასვამს დანარჩენ 8 ბიტს 48 ბიტიანი MAC მისამართის შუაში, რათა მიიღოს 56 ბიტიანი Interface ID(ჰოსტის იდენტიფიკატორი)

- ეს პროცესი იყენებს ჰოსტის 32 ბიტიან Ethernet MAC მისამართს და ჩასვამს დანარჩენ 32 ბიტს 32 ბიტიანი MAC მისამართის შუაში, რათა მიიღოს 64 ბიტიანი Interface ID(ჰოსტის იდენტიფიკატორი)

***IPv6 თავსებადი მოწყობილობა აგზავნის მონაცემების პაკეტს FF02::1 მისამართზე. ჩამოთვლილთაგან რომელი მოწყობილობისთვის ან მოწყობილობებისთვის არის ის განკუთვნილი?***

- ყველა IPv6 სერვერი
- ყველა IPv6 კვანძი(Host) ლოკალურ ქსელში (Local Link )
- ყველა IPv6 კონფიგურირებული როუტერი ლოკალურ ქსელში (Local Link )
- ყველა IPv6 კონფიგურირებული როუტერი მთელს ქსელში (Across the Network)

***IPv6 მისამართთან მიმართებაში რომელი მსჯელობაა მცდარი?***

- IPV6 როგორც 128 ბიტიანი მისამართი, წარმოდგენილია 4 ჰექსტეტის სახით
- IPV6 მისამართის თითოეულ ჰექსტეტში გვაქვს 4 16-ითის სიმბოლო
- IPV6 მისამართის ჰექსტეტები : (2 წერტილით) გამოიყოფა
- IPV6 მისამართი წარმოდგენილია 8 ჰექსტეტის სახით

***OSI მოდელის რომელ დონეზე მუშაობს IPv6 პროტოკოლი?***

- Network
- Internet
- Transport
- Data Link

- Application

*მოცემულთაგან **IPv6** მისამართის რომელი ჩანაწერი არ არის სწორი?*

- ::1
- 2001:0DBF::ACAD::0:1
- FF02:0DBF::ACAD:0:1
- FF02:0DBF:ACAD::1

*2001:0DB8::4775:9 /64 მოცემული Global Unicast IPv6 მისამართიდან გამოყავით და ჩაწერეთ მხოლოდ Interface ID ნაწილი სრულ შეკუმშულ ფორმატში:*

*FC00:ACAD:0:1::1 ჩაწერეთ მოცემული შეკუმშული IPv6 მისამართი სრულად:*

*რამდენი ბიტით აღიწერება IPv6 მისამართის ერთი ჰექსტეტი? (ჩაწერეთ მხოლოდ რიცხვი)*

*ჩაწერეთ 16-ის რიცხვი FF ოროობითში (გამოტოვების გარეშე)*

*ჩაწერეთ FE80:0000:0000:0000:02AA:0000:FE9A:4CA3 მისამართი სრულ შეკუმშულ ფორმატში*

## 2.3. ქსელის ქვექსელებად დაყოფა

რატომ გვჭირდება ქსელის ქვექსელებად დაყოფა?

- ✓ გლობალური IP მისამართები შეზღუდულია და მათი რაციონალური გამოყენებისთვის აუცილებელია ქსელის ქვექსელებად დაყოფა
- ✓ ლოკალური ქსელის ზომების ჰოსტების რაოდენობასთან მოსარგებად, მაგ.: გვინდა რომ ქსელში შესაძლებელი იყოს 30 კვანძის ჩართვა

*ქვექსელების შესაქმნელად საჭიროა ქვექსელის ნიღაბის შეცვლა*

ქვექსელის ნიღაბში (ორობით ფორმატში) 1 არის ქსელის ბიტი, ხოლო 0 კვანძის(Host)

მაგ.:

• 255.255.255.0 - ათობითში

• 11111111.11111111.11111111.00000000 - ორობითში

ე.ი. აქ გვაქვს 24 ქსელის ბიტი და 8 კვანძის (Host)

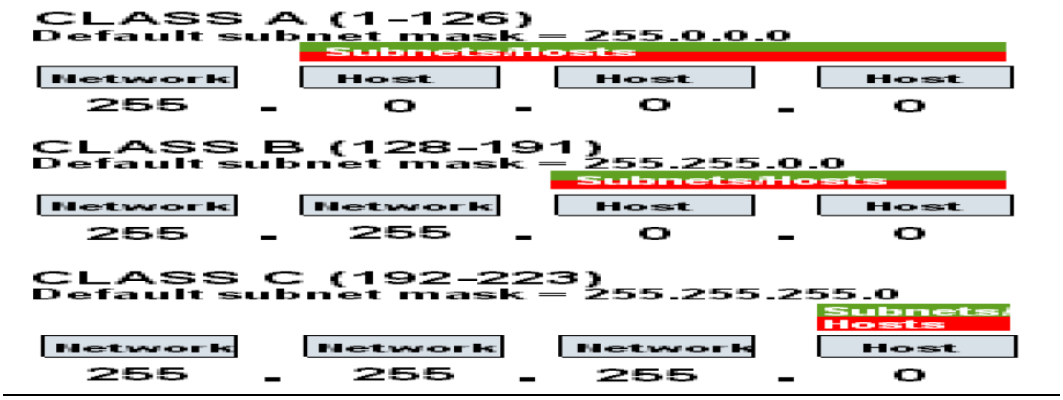
მაგ.:

• 255.255.0.0 - ათობითში

• 11111111.11111111.00000000.00000000 - ორობითში

ე.ი. აქ გვაქვს 16 ქსელის ბიტი და 16 კვანძის (Host)

**ქვექსელის ნიღაბი ქსელის კლასების მიხედვით**




*სურ.2.3. 1*

როგორც სურათიდან ჩანს - A კლასს შეესაბამება 255.0.0.0 ქვექსელის ნიღაბი; B კლასს 255.255.0.0; ხოლო C კლასს - 255.255.255.0

მაგ: IP 10.0.2.15 IP 147.15.20.8 IP 192.168.14.16

S\M 255.0.0.0 S\M 255.255.0.0 S\M 255.255.255.0

**ქვექსელის ნიღაბი ქსელის ქვექსელებად დაყოფისას**

გვაქვს C კლასის **192.168.14.0** ქსელი, მისი ნიღაბი(Mask) არის **255.255.255.0** ანუ

- ✓ **ქსელს** ეკუთვნის **24** ბიტი და კვანძს **8** ბიტი
- ✓ ანუ ქსელში შესაძლებელია ჩართულ იქნას **256 – 2** (192.168.14.0 და 192.168.14.255 დარეზერვებულია) =**254 კვანძი** (Host)

თუ ნიღაბში ჰოსტის ბიტს შევცვლით ქსელის ბიტით მივიღებთ:

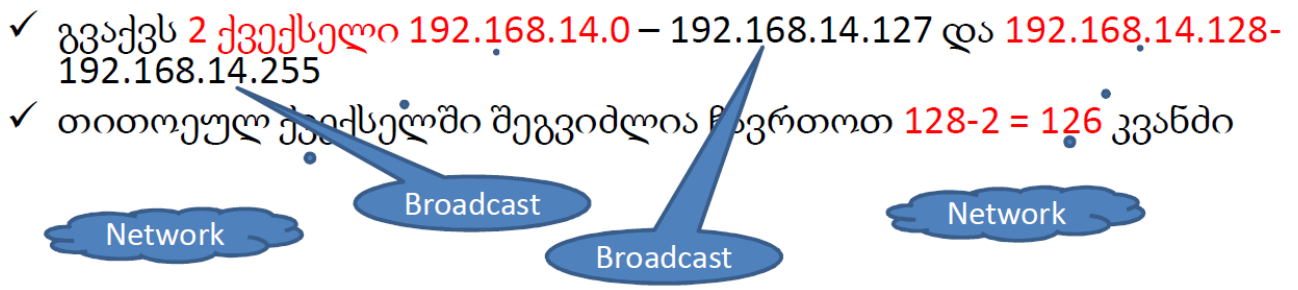
გვქონდა - 11111111.11111111.11111111.00000000 ანუ 255.255.255.0

მივიღეთ - 11111111.11111111.11111111.10000000 ანუ 255.255.255.128

ანუ

- ✓ **ქსელს** ეკუთვნის **25** ბიტი და **კვანძს 7**

ე.ი. მივიღეთ 2 ქვექსელი (იხ. სურ)




*სურ.2.3. 2*

ჩვენ განვიხილეთ შემთხვევა სადაც ქსელს უკეთვნოდა 25 ბიტი ან როგორც აღინიშნება /25

თუ წინა განხილულ მაგალითში კვალავ გავზრდით ქსელების ბიტს მივიღებთ შემდეგ ქვექსელის ნიღაბის მისამართებს:

- /26 - 11111111.11111111.11111111.11000000 ანუ 255.255.255.192
- /27 - 11111111.11111111.11111111.11100000 ანუ 255.255.255.224
- /28 - 11111111.11111111.11111111.11110000 ანუ 255.255.255.240

და ა.შ

- /26- ის შემთხვევაში ქვექსელის დიაპაზონი იქნება 256-192=64 ანუ გვექნება 256 / 64= 4 ქვექსელი, თითოეულში 64-2=62 კვანძით(Host)
- /27- ის შემთხვევაში ქვექსელის დიაპაზონი იქნება 256-224=32 ანუ გვექნება 256 / 32= 8 ქვექსელი, თითოეულში 32-2=30 კვანძით(Host)
- /28- ის შემთხვევაში ქვექსელის დიაპაზონი იქნება 256-240=16 ანუ გვექნება 256 / 16= 16 ქვექსელი, თითოეულში 16-2=14 კვანძით(Host)

და ა.შ.

**/16 - /24**

**255.255.0.0 -255.255.255.0**

ზემოთ მოცემულ დიაპაზონში ქსელის ბიტების შეცვლა გამოიწვევს ქვექსელის ნიღაბის მისამართის შეცვლას მე-3 ოქტეტში

- /17 11111111.11111111.10000000.00000000 ანუ 255.255.128.0 ამ შემთხვევაში გვაქვს 2 ქვექსელი თითოეულში (256- 128=128)*256-2 კვანძით
- /18 11111111.11111111.11000000.00000000 ანუ 255.255.192.0 ამ შემთხვევაში გვაქვს 4 ქვექსელი თითოეულში (256- 192=64)*256-2 კვანძით
- /19 11111111.11111111.11100000.00000000 ანუ 255.255.224.0 ამ შემთხვევაში გვაქვს 8 ქვექსელი თითოეულში (256- 224=32)*256-2 კვანძით

  და ა.შ

**/8 - /16**

**255.0.0.0 -255.255.0.0**

ზემოთ მოცემულ დიაპაზონში ქსელის ბიტების შეცვლა გამოიწვევს ქვექსელის ნიღაბის მისამართის შეცვლას მე-2 ოქტეტში

- /9 11111111.10000000.00000000.00000000 ანუ 255.128.0.0 ამ შემთხვევაში გვაქვს 2 ქვექსელი თითოეულში (256- 128=128)*256*256-2 კვანძით
- /10 11111111.11000000.00000000.00000000 ანუ 255.192.0.0 ამ შემთხვევაში გვაქვს 4 ქვექსელი თითოეულში (256- 192=64)*256*256-2 კვანძით
- /11 11111111.11100000.00000000.00000000 ანუ 255.224.0.0 ამ შემთხვევაში გვაქვს 8 ქვექსელი თითოეულში (256- 224=32)*256*256-2 კვანძით

  და ა.შ

**<u>გვახსოვდეს</u>**

• ქვექსელის ნიღაბი მოცემული IP მისამართის მიხედვით გაგვაგებინებს შესაბამისი ქსელის დიაპაზონს (ქსელის მისამართს(Network), ფართომაუწყებლობით მისამართს(Broadcast) და ქსელში კვანძების რაოდენობას(Host) )

- ✓ მაგ.: გვაქვს 192.168.14.14 /25 ე.ი S/M 255.255.255.128
  - Network Address 192.168.14.0
  - Broadcast Address 192.168.14.127
  - Host Numbers 126

✓ მაგ.: გვაქვს 140.168.14.14 /20 ე.ი S/M 255.255.192.0

– I Network Address 140.168.0.0
– Broadcast Address 140.168.63.255
– Host Numbers 62

✓ მაგ.: გვაქვს 10.168.14.14 /11 ე.ი S/M 255.224.0.0

– Network Address 10.160.0.0
– Broadcast Address 10.191.255.255
– Host Numbers 30

## სასარგებლო ბმულები

- http://en.wikipedia.org/wiki/Subnetwork
- http://www.subnet-calculator.com/
- http://www.youtube.com/results?search_query=subnet%20Mask&search=Search&sa=X&oi=spell&resnum=0&spell=1

*ტესტის ნიმუში*

*მოცემულია შემდეგი IP მისამართი 112.79.79.158 /18, ჩაწერეთ შესაბამისი ფართომაუწყებლობითი მისამართი (Broadcast Address)*

*რამდენი კვანძის(Host) დამისამართებაა შესაძლებელი, შემდეგი ქვექსელის ნიღაბით? -* ***255.255.255.192***

- 64
- 66
- 62
- 68

*ქვექსელის ნიღაბში(Subnet Mask) ქსელს ეკუთვნის 16 ბიტი. ჩაწერეთ შესაბამისი ქვექსელის ნიღაბის მისამართი -*

*მოცემულია შემდეგი ქსელის მისამართი (Network Address) 204.237.128.0 საჭიროა მოცემული ქსელი დაიყოს 9 ქვექსელად, ჩაწერეთ II ქვექსელის პირველი კვანძის(Host) მისამართი*

*მოცემულია შემდეგი ქსელის მისამართი (Network Address) 133.140.0.0 საჭიროა მოცემული ქსელი დაიყოს 53 ქვექსელად, ჩაწერეთ შესაბამი ქვექსელის ნიღაბის(Subnet Mask) მისამართი*

*მოცემულია შემდეგი IP მისამართი 112.79.79.158 /28, ჩაწერეთ შესაბამისი ქსელის მისამართი (Network Address)*

*ჩამოთვლილთაგან რომელია ქსელის (Network) მისამართი?*

- 157.254.45.192/26
- 98.253.160.255/24
- 117.214.145.0/20
- 97.141.244.63/26

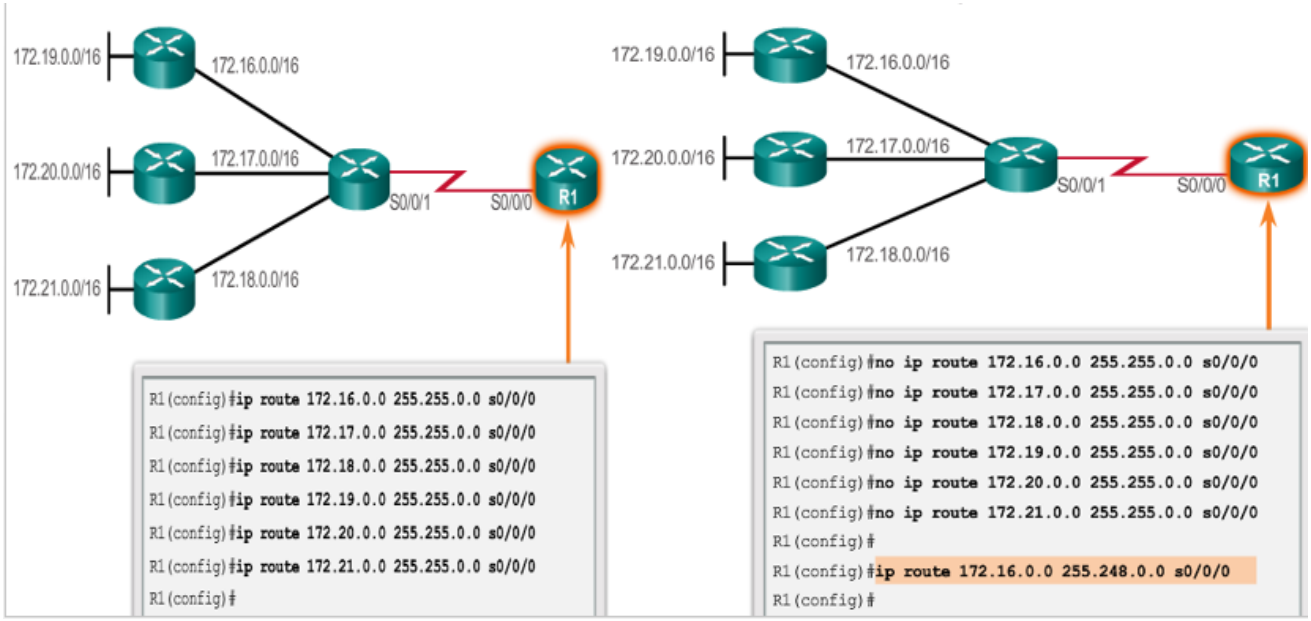
*ჩამოთვლილთაგან რომელი არ შეიძლება მიენიჭოს კვანძს(Host)?*

- 17.12.15.89 /24
- 192.168.14.16 /24
- 10.0.0.255 /16
- 172.0.255.0 /24

*მოცემულია შემდეგი ქსელის მისამართი (Network Address) 133.140.0.0 საჭიროა მოცემული ქსელი დაიყოს 53 ქვექსელად, ჩაწერეთ რამდენი კვანძის (Host) ჩართვა იქნება შესაძლებელი ამგვარად დაყოფის შედეგად მიღებულ თითოეულ ქვექსელში*

## 2.4. ქვექსელების შეჯამება - Summarization

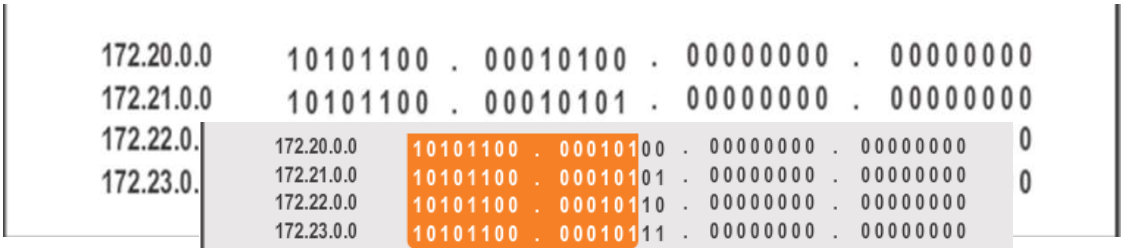
ჯამური მარშრუტი საშუალებას იძლევა ერთი ჩანაწერის სახით გავაერთიანოთ რამოდენიმე სტატიკური მარშრუტის შესაბამისი ჩანაწერი




*სურ.2.4. 1*

**ჯამური მარშრუტის გამოთვლის წესი:**

**ნაბიჯი 1.** წარმოვადგინოთ ქსელის მისამართები ორობით ფორმატში



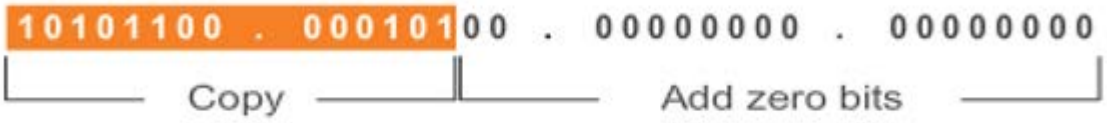

*სურ.2.4. 2*

**ნაბიჯი 2.** ქვექსელის ნიღაბის მისაღებად(Subnet mask) დავითვალოთ მარცხენა ნაწილში იდენტური სიმბოლოების(ბიტების) რიცხვი

*სურ.2.4. 3*

ზემოთ მოცემული სურათის მიხედვით მივიღეთ **14 იდენტური სიმბოლო** ანუ **/14** (14 ბიტი), რისი შესაბამისი ქვექსელის ნიღაბიც იქნება **255.252.0.0**

**ნაბიჯი 3.** ჯამური ქსელის მისამართის მისაღებად ავიღოთ მარცხენა ნაწილში იდენტური სიმბოლოები და დანარჩენი ბიტები შევავსოთ ნოლებით




*სურ.2.4. 4*

ზემოთ მოცემული სურათის მიხედვით 14 იდენტური სიმბოლოს დავუმატეთ ნოლები და მივირეთ ჯამური ქსელის მისამართი 172.20.0.0

**ზემოთ აღწერილი ნაბიჯებით მივიღეთ <u>ჯამური მარშრუტი</u>**

172.20.0.0 255.252.0.0

**რომელიც თავის თავში მოიცავს შემდეგ ქსელურ მისამართებს**

172.20.0.0 255.255.0.0

172.21.0.0 255.255.0.0

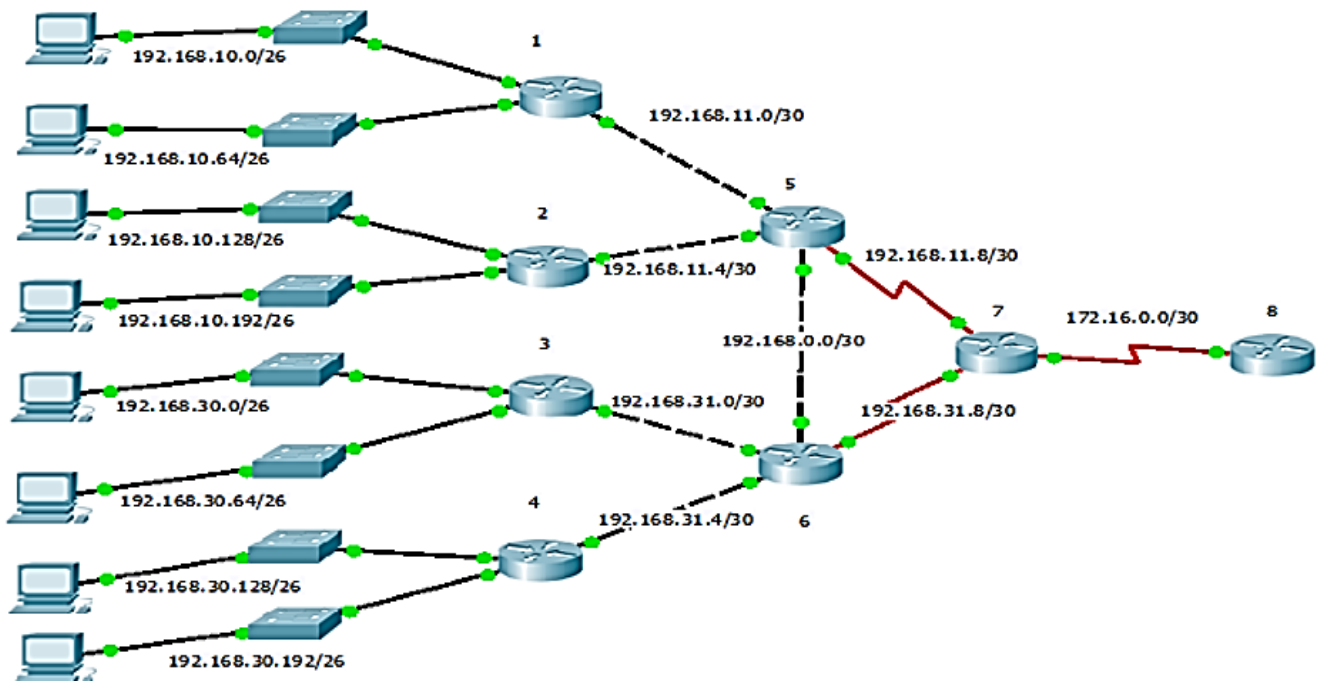
172.22.0.0 255.255.0.0

172.23.0.0 255.255.0.0

**დამოუკიდებელი სამუშაოების ჩამოსატვირთად გააქტიურეთ ქვემოთ მოცემული ბმულები**

- http://1drv.ms/1lSXIfm
- http://1drv.ms/1lSYlFT

*პრაქტიკული სამუშაო*




1. შექმენით მოცემულის შესაბამისი ლოკალური ქსელი და ინტერფეისები დაამისამართეთ შესაბამისად

2. I, II, III და IV მარშრუტიზატორებზე დანიშნეთ Default მარშრუტები

3. მე-5 მარშრუტიზატორზე დანიშნეთ ჯამური მარშრუტები: I როუტერის ქსელებზე; II როუტერის ქსელებზე;

4. მე-5 მარშრუტიზატორზე დანიშნეთ 2 ალტერნატიული ერთიანი ჯამური მარშრუტი მე-3 და მე-4 როუტერის ქსელებზე;

5. მე-5 მარშრუტიზატორზე დანიშნეთ 2 ალტერნატიული სტატიკური მარშრუტი მე-8 მარშრუტიზატორის ქსელზე

6. მე-6 მარშრუტიზატორზე დანიშნეთ ჯამური მარშრუტები: III როუტერის ქსელებზე; IV როუტერის ქსელებზე;

7. მე-6 მარშრუტიზატორზე დანიშნეთ 2 ალტერნატიული ერთიანი ჯამური მარშრუტი I და მე-2 როუტერის ქსელებზე;

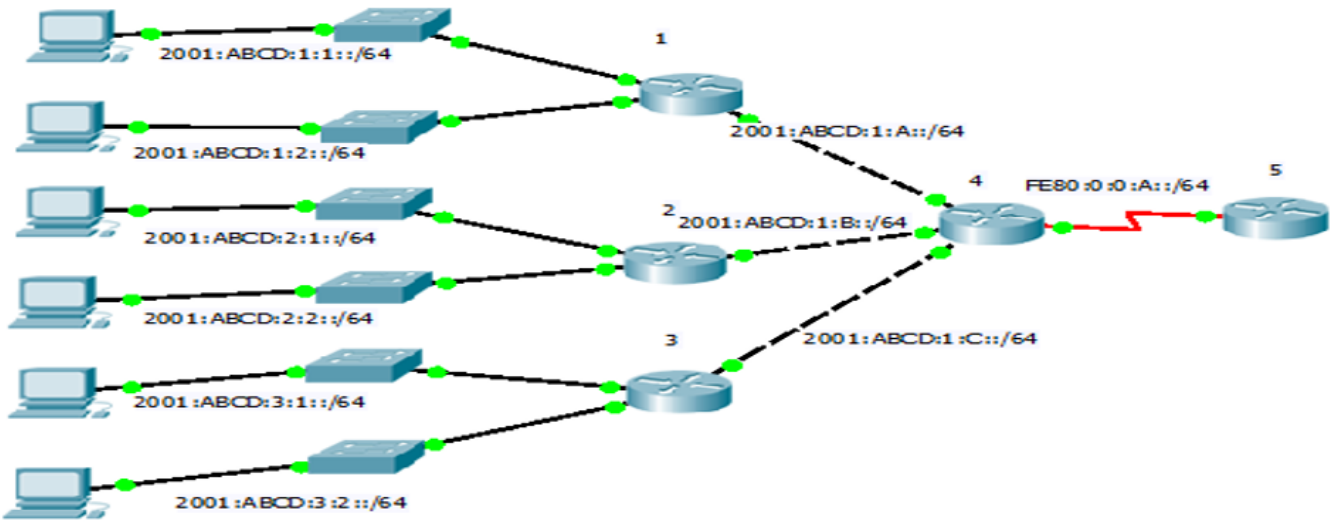
8. მე-6 მარშრუტიზატორზე დანიშნეთ 2 ალტერნატიული სტატიკური მარშრუტი მე-8 მარშრუტიზატორის ქსელზე

9. მე-7 მარშრუტიზატორზე დანიშნეთ 2 ალტერნატიული სტატიკური მარშრუტი მე-3 და მე-4 როუტერის ქსელებზე;

10. მე-7 მარშრუტიზატორზე დანიშნეთ 2 ალტერნატიული სტატიკური მარშრუტი I და მე-2 როუტერის ქსელებზე;

11. მე-8 მარშრუტიზატორზე დანიშნეთ 2 ალტერნატიული სტატიკური მარშრუტი ყველა სხვა ქსელზე;

*პრაქტიკული სავარჯიშო*




12. შექმენით მოცემულის შესაბამისი ლოკალური ქსელი და ინტერფეისები დაამისამართეთ შესაბამისად

13. I, II და III მარშრუტიზატორებზე დანიშნეთ Default მარშრუტები

14. მე-4 მარშრუტიზატორზე დანიშნეთ ჯამური მარშრუტები: I როუტერის ქსელებზე; II როუტერის ქსელებზე; III როუტერის ქსელებზე;

15. მე-5 მარშრუტიზატორზე დანიშნეთ სტატიკური ჯამური მარშრუტი ყველა სხვა ქსელზე;

პროცესზე დაკვირვება

შემფასებელი აკვირდება შესაფასებელ პირის მუშაობას პროფესიული სტანდარტით(პროგრამით / მოდულით ) განსაზღვრული ამოცანების შესრულების პროცესში . დაკვირვება ხორციელდება სამუშაო ადგილზე ან სამუშაო პირობებში, თუმცა დასაშვებია ჩატარდეს კომპიუტერებით აღჭურვილ ლაბორატორიაში, სადაც შესაფასებელი პირი პრაქტიკულ საქმიანობას ეწევა. შემფასებელმა წინასწარ უნდა დაგეგმოს დაკვირვების პროცესი, იმის დასადგენად, თუ რამდენად სწორად იყენებს შესაფასებელი პირი ცოდნას, უნარებსა და ყველა რესურსს შედეგის მისაღწევად.შეფასება დადებითია იმ შემთხვევაში თუ სტუდენტი ყველა კითხვას სწორად გასცემს პასუხს.

*პრაქტიკული სავარჯიშო -*

**IPv4 და Ipv6 მისამართების შერჩევა და მათი გამოყენება**

| სწავლის შედეგი | N | დასახელება | შეფასება | |
|---|---|---|---|---|
| | | | კი | არა |
| IPv4 დამისამართება | 1. | სწორად შეარჩია IPv4 მისამართები. | | |
| | 2. | სწორად მოახდინა IPv4 მისამართების გაწერა მოწყობილობებზე | | |
| IPv6 დამისამართება | 3. | სწორად შეარჩია IPv6 მისამართები. | | |
| | 4. | სწორად მოახდინა IPv6 მისამართების გაწერა მოწყობილობებზე | | |
| ქსელების ქვექსელებად დაყოფა - Subnetting. | 1. | სწორად შეარჩევს ქვექსელის ნიღაბს(Network Masks) | | |
| | 2. | მოთხოვნილი შესაბამისად სწორად მოახდინა ქსელის ქვექსელებათ დაყოფას | | |
| ქსელების შეჯამება - Summarization. | 1. | სწორად შეარჩია შეჯამებისთვის საჭირო ქსელები | | |
| | 2. | მოთხოვნის შესაბამისად სწორად შეაჯამა ქსელები | | |

# 3. მეორე დონის პროტოკოლების და ტექნოლოგიების საფუძვლები

## 3.1. Ethernet ტექნოლოგიისა და სტანდარტების ფუნდამენტური პრინციპების გარჩევა

**Ethernet ტექნოლოგია**

Ethernet – არის დღევანდელ დღეს ყველაზე გავრცელებული ლოკალური ქსელის სტანდარტი. ამ სტანდარტით აგებულია ათეულობით მილიონი ლოკალური ქსელი. მსოფლიოში პირველი ლოკალური ქსელი იყო Ethernet-ის ორიგინალური ვერსია.

**ისტორია**

30-ზე მეტი წლის წინ რობერტ მეტკალფმა და მისმა კოლეგებმა Ethernet ქსელი ფირმა "ქსეროქსში" დააპროექტეს. პირველი Ethernet ტექნოლოგიის სტანდარტი იქნა გამოქვეყნებული 1980 წელს კონსორციუმის მიერ, რომელიც შედგებოდა Intel-ისაგან, Xerox-ისაგან და Digital Equipment Corporation-ისგან (DIX). მეტკალფს უნდოდა რომ Ethernet ყოფილიყო განაწილებული სტანდარტი, ამიტომ ის იქნა გამოშვებული როგორც ღია სტანდარტი. პირველი პროდუქტები რომლებიც შეიქმნა Ethernet სტანდარტიდან გაყიდვაში გამოჩნდა XX საუკუნის 80-იან წლებში. 1985 წელს, ელექტრონიკისა და ელექტროობის ინსტიტუტის (IEEE) სტანდარტების კომიტეტმა გამოაქვეყნა ლოკალური და ქალაქის ზომის ქსელების სტანდარტები, ციფრებით 802. Ethernet-ის სტანდარტია 802.3. ინსტიტუტს უნდოდა, რომ მათი სტანდარტი შეთავსებულიყო საერთაშორისო სტანდარტების ორგანიზაციასთან (ISO) და OSI მოდელთან. იმისათვის, რომ ეს მომხდარიყო IEEE802.3 სტანდარტებს უნდა დაეკმაყოფილებინა OSI მოდელის პირველი დონისა და მეორე დონის ქვედა ნაწილის მოდელის მოთხოვნები. ამის შედეგად პატარა ცვლილებები განიცადა ორიგინალურმა Ethernet სტანდარტმა (802.3).

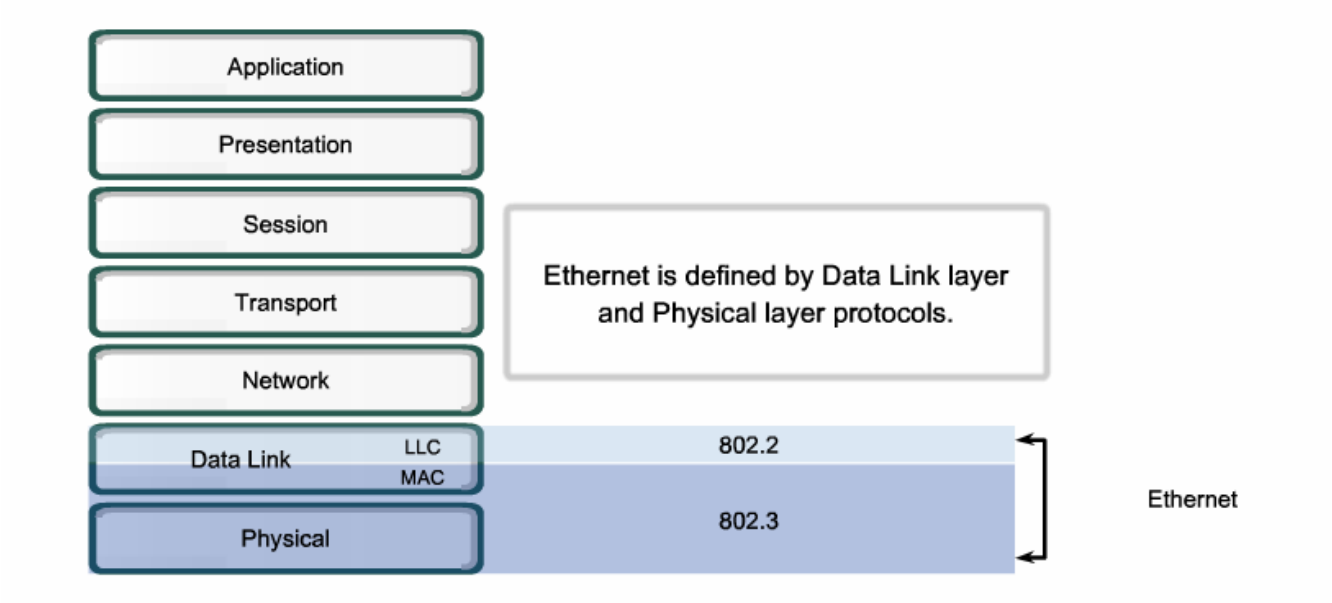
Ethernet მოქმედებს OSI მოდელის ორ ქვედა: არხის და ფიზიკურ დონეზე. რადგანაც OSI მოდელი გამოიყენება მხოლოდ წარმოსახვისთვის (ახსნისათვის), ამიტომ ის ყოველთვის ზუსტად ვერ აღწერს ყველა ტექნოლოგიას და პროტოკოლს რომელიც გამოიყენება კომპიუტერულ ქსელებში. არხის დონე გაყოფილია ორ ქვედონედ. პირველი ქვედონე (ზედა - Logical Link Control (LLC), ხოლო მეორე ქვედონე ე.წ. Media Access Control

(MAC - მედიაზე წვდომის კონტროლი. რეალურად Ethernet მოქმედებს არხის დონის მეორე (ქვედა - MAC) ქვედონეზე და ფიზიკურ დონეზე., რადგანაც (LLC) საერთოა ბევრი ტექნოლოგიისათვის Ethernet პირველ (ფიზიკურ) დონეზე შეიცავს:

- სიგნალებს;
- ბიტთა ნაკადებს, რომლებიც მოგზაურობენ მედიაში;
- ფიზიკურ კომპონენტებს, რომლებიც ათავსებენ სიგნალებს მედიაზე;
- სხვადასხვა ტოპოლოგიებს.

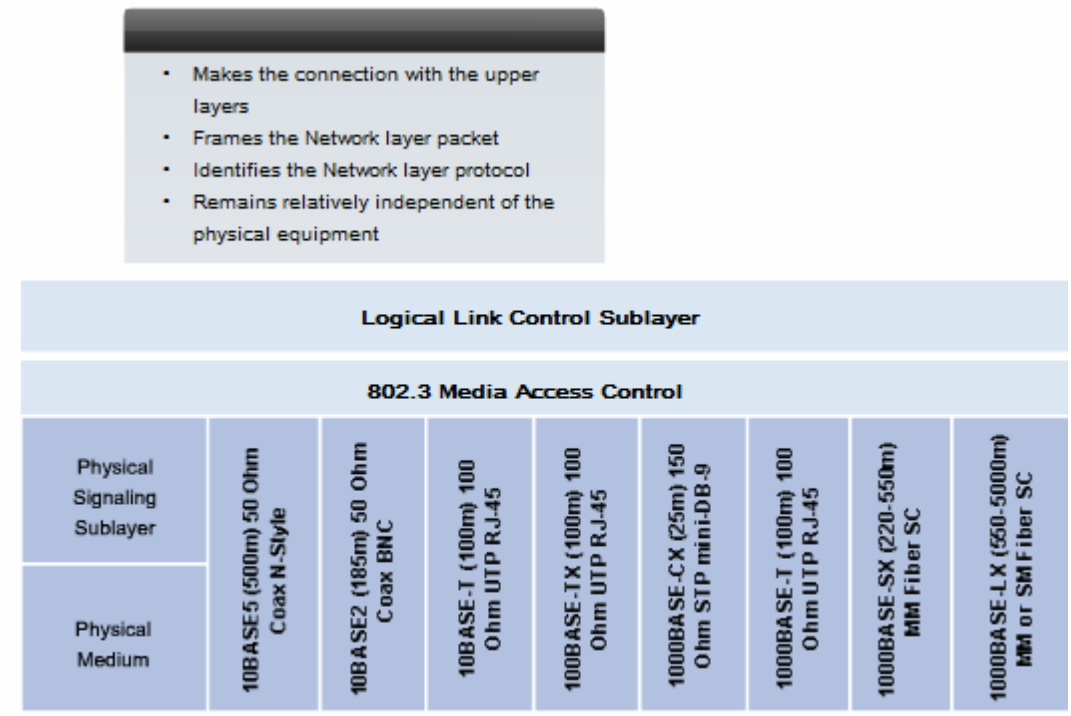
Ethernet-ის პრიველ დონეს გადამწყვეტი როლი უკავია მოწყობილობებს შორის კავშირის განხორციელებაში, თუმცა მის ყოველ ფუნქციას აქვს შეზღუდვები. Ethernet-ის მეორე (არხის) ქვედონეები, მნიშვნელოვან როლს ასრულებენ ტექნოლოგიურ თავსებადობაში და კომპიუტერულ კომუნიკაციაში. MAC ქვედონე დაკავშორებულია ფიზიკური კომპონენტებითან, რომლებიც გამოყენება ინფორმაციის დასაკავშირებლად და ის ამზადებს მონაცემებს მედიაზე გადასაცემად.

Logical Link Control (LLC) - ლოგიკური არხის კონტროლის ქვედონე, შედარებით დამოუკიდებელი რჩება ფიზიკურ მოწყობილობებბისაგან, რომლებსაც იყენებენ კომპიუტერულ ქსელებში.




*სურ.3.1. 1 Ethernet*

როგორც ზემოთ ავნიშნეთ Ethernet ყოფს მონაცემთა არხის დონის ფუნქციებს ორ განსხვავებულ ქვედონედ: LLC და MAC ქვედონეები. ფუნქციები რომლებიც იყო აღწერილი OSI მოდელში მონაცემთა არხის დონისთვის მოიცავს ამ ორ ქვედონეს. ამ ორი ქვედონის გამოყენება ხელს უწყობს თავსებადობას სხვადასხვა მოწყობილობებს შორის. Ethernet-ისთვის IEEE 802.2 სტანდარტმა აღწერა LLC ქვედონის ფუნქციები და 802.3 სტანდარტმა MAC ქვედონის და ფიზიკური დონის ფუნქციები. LLC ქვედონის ფუნქცია არის კავშრის უზრუნველყოფა ზედა დონეებთან და ქსელურ პროგრამებთან, ხოლო ქვედა ქვედონის ფუნქცია არის კავშრის უზრუნველყოფა (MAC) - აპარატურასთან. LLC ქვედონე იღებს ქსელური პროტოკოლის მონაცემებს, რომელიც ჩვეულებრივ არის IPv4 პაკეტი და ამატებს მმართველ ინფორმაციას იმისათვის, რომ დაეხმაროს პაკეტს დანიშნულების ადგილის მისაღწევად. მეორე დონე უკავშირდება ზედა დონეებს LLC-ის გამოყენებით. LLC წარმოდგენილია პროგრამულ უზრუნველყოფის სახით და ის დამოუკიდებელია ფიზიკურ მოწყობილობებზე. კომპიუტერში LLC შეგვიძლია წარმოვიდგინოთ როგორც ქსელური ადაპტერის დრაივერი. ეს არის პროგრამა, რომელიც შუამავლების გარეშე ურთიერთქმედებს აპარატურასთან ქსელურ ადაპტერში, რათა გადასცეს მონაცემი მედიასა და MAC ქვედონეს შორის.




*სურ.3.1. 2 LLC - ლოგიკური არხის კონტროლის ქვედონე*

MAC - არის არხის დონის (Ethernet) ქვედა ქვედონე. მას აქვს ორი ძირითადი სამუშაო:

- მონაცემთა ენკაპსულაცია - Data Encapsulation;
- მედიაზე წვდომის კონტროლი - Media Access Control.

მონაცემთა ენკაპსულაცია გვაძლევს სამ ძირითად ფუნქციას:

- შეცდომების აღმოჩენა;
- დამისამართება;
- კადრების განსაზღვრა.

მონაცემთა ენკაპსულაციის პროცესი შეიცავს ფრეიმების აწყობას მონაცემთა გადაცემამდე და ფრეიმების გარჩევას მონაცემთა მიღების შემდეგ. ფრეიმის ჩამოყალიბებაში, MAC ქვედონე ამატებს თავსართს და ბოლოსართს, მესამე დონის "პაკეტის მონაცემთა ერთეულზე" (PDU). ფრეიმების გამოყენება გვეხმარება ბიტების გადაცემაში როცა ისინი გადაიცემინ მედიაზე და მათ აწყობაში როცა ხდება მათი მიმღება. ფრეიმის შექმნის პროცესი გვაწვდის მნიშვნელოვან მსაზღვრელებს, რომლებიც გამოიყენებიან ბიტების ჯგუფის იდენტიფიცირებისთვის რომლისგანაც შედგება ფრეიმი. ეს პროცესი ახდენს სინქრონიზაციას გადამცემ და მიმღებ მხარეებს შორის. ენკაპსულაციის პროცესი

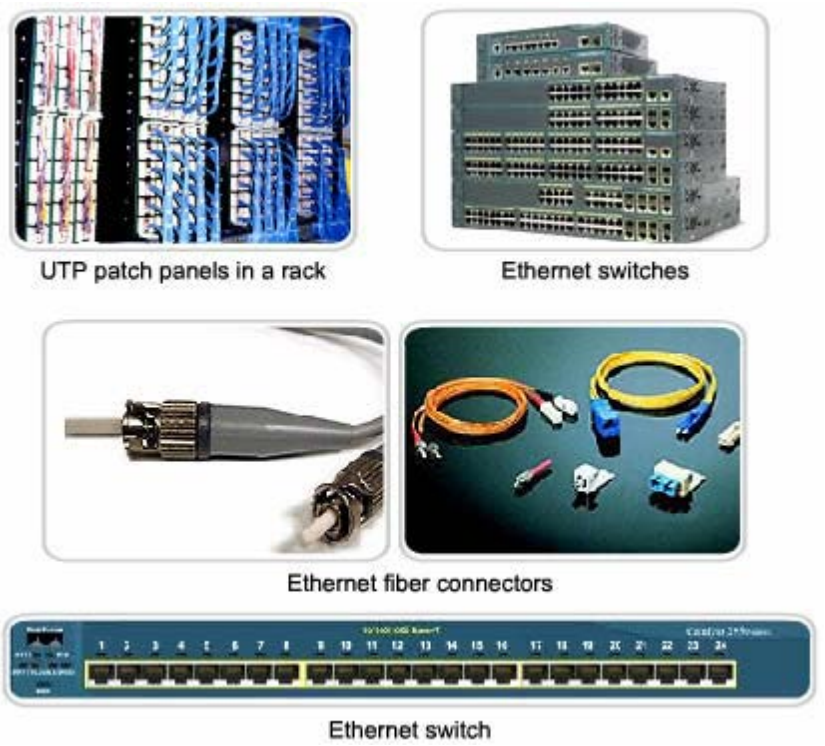
ასევე გვაძლევს მონაცემთა არხის დონის დამისამართებას. თითოეული Ethernet თავსართი, რომელიც ემატება ფრეიმს, შეიცავს ფიზიკურ მისამართს (MAC Address), რომელიც მას აძლევს საშუალებას მიაღწიოს დანიშნულების ადგილამდე. ენკაპსულაციის დამატებითი ფუნქცია არის შეცდომების აღმოჩენა. თითოეული Ethernet ფრეიმი შეიცავს ბოლოსართს, რომელიც შედგება ციკლური ნამატის შემოწმების სისტემისაგან (CRC). სანამ ფრეიმი გაიგზავნება ქსელში, ფრეიმის ფორმირებისას ხდება გარკვეული მათემატიკური ოპერაცია, შედეგი კი მიეწერება ფრეიმს ბოლოში ციკლური ნამატის სახით. ფრეიმის მიღების შემდეგ, მიმღები მხარე ქმნის იგივე პრინციპით CRC-ს და შემდგომ მას ადარებს მიღებულ კადრში განთავსებულ CRC-თან. თუ ეს ორი CRC ერთმანეთს ემთხვევა, შეგვიძლია ჩავთვალოთ რომ ფრეიმი მიღებულია უშეცდომოდ. MAC ქვედონე აკონტროლებს მედიაზე კადრების განთავსებას და აგრეთვე მათ მოშორებას. როგორც მისი სახელიდან ჩანს ის მართავს მედიაზე დაშვებას, ეს შეიცავს ფრეიმის გადაცემის დაწყებას და კოლიზიის აღმოჩენის შემთხვევაში მის ხელახალ გადაცემას. Ethernet-ის საფუძველს წარმოადგენს ლოგიკური ტოპოლოგია “მრავალი-დაშვების პრინციპი მედიაზე” (Multi Access Bus). ეს ნიშნავს რომ ყველა მხარე (მოწყობილობა) ქსელში ინაწილებს “მედიას (გამტარს)”. ეს კი თავისმხრივ ნიშნავს, რომ ყველა მხარე იღებს ყველა ფრეიმს იმისდა მიუხედავათ არის ის განკუთვნილი მისთვის. ამის გამო თითოეულმა მიმღებმა უნდა გაარკვიოს არის ეს ფრეიმი მისთვის გამოგზავნილი თუ არა. ამისათვის ხდება მისამართების შემოწმება კადრში, რომელიც წარმოდგენილია MAC მისამართის სახით. Ethernet გვაწვდის მეთოდს რათა დავადგინოთ თუ როგორ ხდება მედიაზე წვდომის განაწილება მხარეთა შორის. კლასიკურ Ethernet-ში ამას ანხორციელებს პროტოკოლი(CSMA/CD) „ინფორმაციის გადაცემის აღმოჩენა მრავალჯერადი შეღწევა და კოლიზიის აღმოჩენა“ (Carrier Sense Multiple Access with Collision Detection). ინტერნეტში ტრაფიკის უდიდესი ნაწილი იწყება და მთავრდება Ethernet კავშირებით. მისი დასაბამიდან 70-წლებში, Ethernet-მა განიცადა ცვლილებები, რათა ეპასუხა გაზრდილ მოთხოვნილებაზე, შექმნილიყო სწრაფი ლოკალური ქსელები. როდესაც ოპტიკურ ბოჭკოვანი კაბელი შემოვიდა ხმარებაში, Ethernet-მა გაიარა ამ ახალ ტექნოლოგიასთან ადაპტაცია და გამოიყენა მისი უპირატესობები, როგორიც არხის მაღალი გამტარობა და შეცდომების მცირე რაოდენობა. დღესდღეობით იგივე პროტოკოლს,

რომელსაც გადაჰქონდა ინფორმაცია 3მბ/წმ სიჩქარით, შეუძლია გადაიტანოს ის 10გბ/წმ სიჩქარით.

Ethernet წარმატება შემდეგმა პირობებმა გამოიწვია:

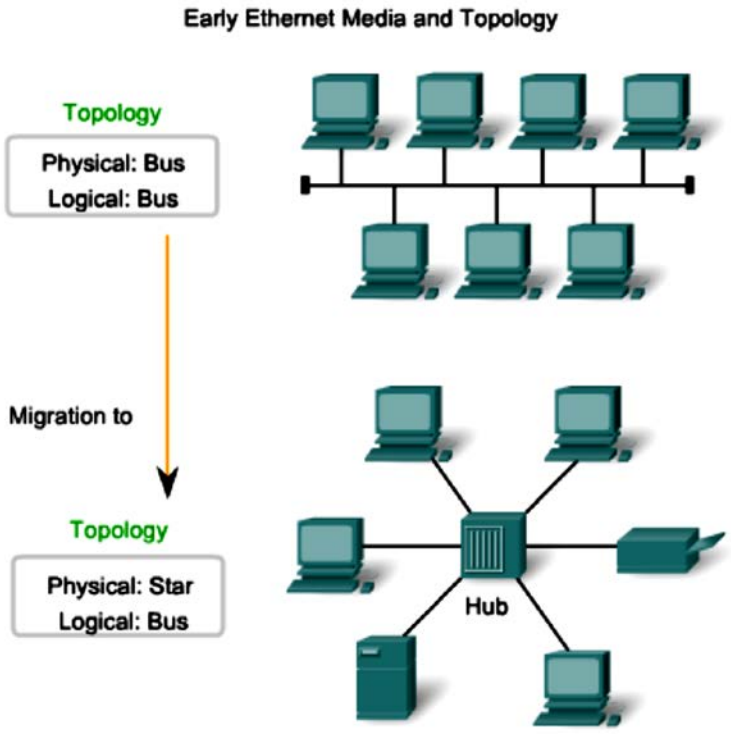
- სიმარტივე და ადვილი მომსახურეობა;
- საშუალება შთანთქას ახალი ტექნოლოგიები;
- საიმედოობა;
- ინსტალაციის და გაუმჯობესების დაბალი ფასი.

Ethernet ტექნოლოგიისთვის საფუძველი პირველად 1970 წელს შეიქმნა და პროგრამას ერქვა Alohanet. ეს იყო ციფრული რადიო ქსელი, შემუშავებული ისე რომ გადაეცა ინფორმაცია განაწილებულ რადიო სიხშირეზე ჰავაის კუნძულებს შორის. ამ ქსელისთვის საჭირო იყო ყველა მონაწილე მხარე დამორჩილებოდა პროტოკოლს, რომელშიც არადადასტურებული (Unacknowledged) გადაცემა უნდა განმეორებულიყო დროის პატარა მონაკვეთის შემდეგ. განაწილებული მედიის გამოყენების გზები იქნა შემდგომში გამოყენებული კაბელურ გამტარებში Ethernet ფორმით. Ethernet დიზაინი შექმნილი იყო ისე, რომ განეთავსებინათ რამდენიმე ურთიერთდაკავშირებული კომპიუტერი განაწილებულ სალტურ ტოპოლოგიაზე. Ethernet-ის პირველ ვერსიაში დაშვების მეთოდი იყო ცნობილი როგორც Carrier Sense Multiple Access with Collision Detection (CSMA/CD). ის მართავდა პრობლემებს რომლებიც წარმოიქმნებოდნენ მაშინ, როდესაც რამდენიმე მოწყობილობა ერთდროულად შეეცდებოდა კავშირს განაწილებულ ფიზიკურ მედიაზე.




*სურ.3.1. 3 Ethernet ტექნოლოგიის ფიზიკური მოწყობილობები*

Ethernet-ის პირველი ვერსიები სალტურ ტოპოლოგიასთან დასაკავშირებლად იყენებდნენ კოაქსიალურ კაბელს. თითოეული კომპიუტერი პირდაპირ იყო შეერთებული “მაგისტრალურ არხთან” (Backbone). ეს ვერსიები იყო ცნობილი როგორც Thicknet (10BASE5) და Thinnet (10BASE2).




*სურ.3.1. 4 Ethernet ტექნოლოგიის პირველი ვერსიები*

10BASE5 იყენებდა სქელ კოაქსიალურ კაბელს, რომელიც იძლეოდა საშუალებას 500 მეტრამდე კაბელის გაყვანას, სანამ დასჭირდებოდა განმეორებელი (Repeater). 10BASE2 კი იყენებდა თხელ კოაქსიალურ კაბელს, თუმცა ის უფრო ელასტიური იყო და მისი გაყვანა შეიძლებოდა 185 მეტრამდე. ორიგინალური Ethernet-ის მოღწევამ დღევანდელ დღემდე გამოიწვია შემდეგმა ფაქტორმა, მეორე დონის ფრეიმის სტრუქტურა პრაქტიკულად შეუცვლელი რჩება. ფიზიკური მედია, მედიაზე დაშვება და მედია კონტროლი განვითარდა და აგრძელებენ განვითარებას. თუმცა Ethernet-ის თავსართი და ბოლოსართი შეუცვლელი დარჩა. Ethernet ადრინდელ ვარიანტებში იყო გამოყენებული დაბალი გამტარობის ლოკალური ქსელების გარემოში, სადაც დაშვებას განაწილებულ მედიაზე მართავდა CSMA, ხოლო შემდგომ CSMA/CD. იმასთან ერთად რომ ის იყო ლოგიკური სალტური ტოპოლოგია მონაცემთა არხის დონეზე, ის ასევე იყო სალტური ტოპოლოგია ფიზიკურ დონეზეც. ეს ტოპოლოგია გახდა უფრო პრობლემატური, როდესაც ლოკალური ქსელები გაიზარდნენ და მათი მომსახურეობებიც მომრავლდა. კოაქსიალური კაბელები ჩაანაცვლეს UTP კაბელების ადრეულმა ვარიანტებმა. კოაქსიალურთან შედარებით ეს კაბელები მსუბუქი და იაფია.

მათთან მუშაობა ბევრად უფრო მარტივი იყო. ფიზიკური ტოპოლოგიაც შეიცვალა ვარსკვალურ ტოპოლოგიად კონცენტრატორების (Hub) გამოყენებით. ისინი კავშირებს აკონცენტრირებდნენ, სხვა სიტყვებით რომ ვთქვათ, მას ინდივიდუალური კაბელებით უკავშირდება ყველა ქსელში ჩართული მოწყობილობა და ეს საშუალებას იძლევა ქსელი აღქმულ იქნას როგორც ერთი განაწილებული მედია. როდესაც ფრეიმი შემოვა ერთ პორტზე, მისი გადაკოპირება ხდება ყველა დანარჩენ პორტზე გარდა იმ პორტისა საიდანაც შემოვიდა ფრემი, შესაბამისად ქსელის ყველა სეგმენტი იღებს ფრეიმს. კონცენტრატორის გამოყენებამ სალტურ ტოპოლოგიაში მას შემატა საიმედოობა და ერთი კონკრეტული კაბელის მწყობრიდან გამოსვლის შემთხვევაში არ ითიშება მთელი ქსელი. თუმცა ყველა დანარჩენი პორტისთვის ფრეიმის გამეორებამ არ გადაწყვიტა კოლიზიების პრობლემა.

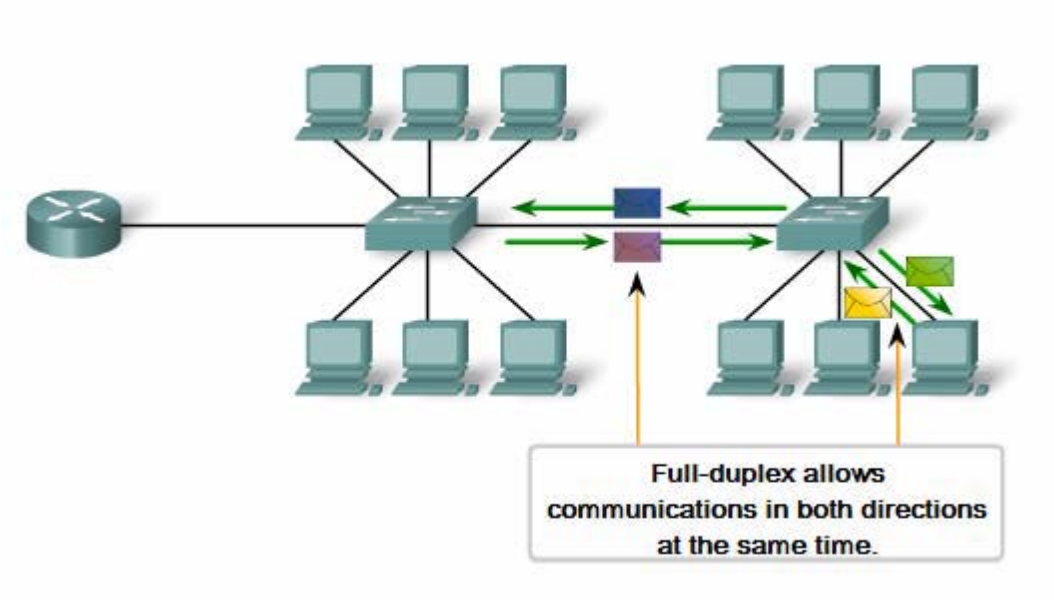
**კლასიკური Ethernet**

ტიპიურად 10BASE-T ქსელებში, ცენტრალური წერტილი ქსელის სეგმენტისა იყო კონცენტრატორი. ამან წარმოშვა განაწილებული მედია. იმის გამო, რომ მედია არის განაწილებული, მხოლოდ ერთ მხარეს შეეძლო წარმატებით ინფორმაციის გადაცემა ნებისმიერ მოცემულ დროის მონაკვეთში. ამ კავშირს ეწოდება ნახევარ-დუპლექსური კომუნიკაცია. მეტი მოწყობილობების დამატებასთან ერთად, ქსელში კადრების კოლიზიების რიცხვი მატულობდა. მაშინ, როდესაც კოლიზიების რიცხვი დაბალი იყო CSMA/CD მართვის შედეგად მომხმარებლისთვის არ იგრძნობოდა დისკომფორტი. თუმცა მათი რიცხვის გაზრდასთან ერთად შეიქმნა დისკომფორტიც. მსგავსი სიტუაციია იქნება როდესაც დილით ადრე მივემგზავრებით სადმე, გზა თავისუფალია და მასზე ცოტა მანქანები მოზრაობენ, თუმცა საღამოთი როდესაც გზებზე ბევრი მანქანა იწყებს მოძრაობას, იქმნება საცობები და მოძრაობა შენელებულია.

**დღევანდელი Ethernet**

ლოკალური ქსელების განვითარების ერთერთი უმნიშვნელოვანესი ეტაპი იყო სვიჩების (კომუტატორების) გამოჩენა, რომლებმაც ჩაანაცვლეს კონცენტრატორები. ეს დროში ახლოს მოხდა 100BASE-TX Ethernet-ის შექმნასთან. სვიჩებს შეუძლიათ აკონტროლონ მონაცემთა ნაკადი და გააგზავნონ ფრეიმი მხოლოდ იმ პორტზე, რომლისთვისაც არის ის განკუთვნილი. სვიჩი ამცირებს იმ მოწყობილობების რაოდენობას,

რომლებიც იღებენ კონკრეტულ ფრეიმს, რადგანაც სვიჩი ანხორციელებს მიზანმიმართულ გადაცემას პორტიდან პორტზე და ამით ამცირებს კოლიზიების რაოდენობას. ზემოთანიშნული ფუნქციის და შემდგომ სრული-დუპლექსის კომუნიკაციის გამოჩენამ (ერთდროულად გადაცემის და მიღების საშუალება) გამოიწვია 1გბიტ/წამში და უფრო სწრაფის Ethernet შექმნა .

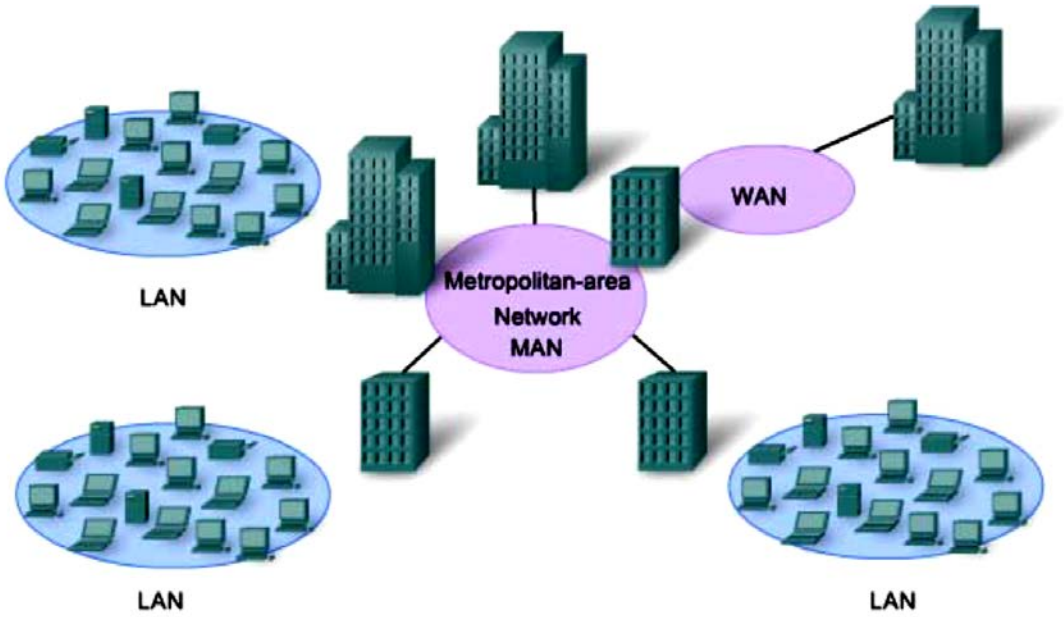



*სურ.3.1. 5 სვიჩის გამოყენებით აგებული ლოკალური ქსელი*

თანამედროვე მულტიმედიური პროგრამები, რომლებიც იყენებენ კომპიუტერულ ქსელს, ყოველდღიურად ტვირთავენ ყველაზე სწრაფ ქსელებსაც კი. მაგალითად, VoIP ტექნოლოგიის და მულტიმედიური გამოყენების ზრდამ, საჭირო გახადა უფრო სწრაფ კავშირები, ვიდრე არის 100მბიტ/წამში Ethernet. გიგაბიტ Ethernet გამტარობა შეადგენს 1000 მბიტ/წმ. ეს მიიღება სრული-დუპლექსის და UTP ან ოპტიკურ ბოჭკოვანი ტექნოლოგიების გამოყენებით. როდესაც ხდება ქსელის განახლება 100მბიტ/წამის გამტარობიდან 1გბიტ/წამამდე ან მეტი, განსხვავება საგრძნობია. ქსელის განახლება 1გბ/წმ-მდე ყოველთვის არ ნიშნავს მთელი ქსელის ინფრასტრუქტურის გამოცვლას. ზოგიერთი მოწყობილობა თანამედროვე ქსელებში შეიძლება ძალიან პატარა დანახარჯებით ამუშავდეს უფრო მაღალ სიჩქარეებზე.


*სურ.3.1. 6 ახალი მოწყობილობები და სერვისები რომლებიც ითხოვენ სწრაფ კომუნიკაციას*

ოპტიკურ-ბოჭკოვანი კაბელის შემოსვლასთან ერთად ზღვარი ლოკალურ ქსელსა და ფართო არის ქსელთან წაიშალა. Ethernet თავდაპირველად შემოისაზღვრებოდა ერთი შენობით და შემდეგ გავრცელდა შენობათა შორის. დღეს ის შეიძლება მთელს ქალაქს ფარავდეს და მას ეწოდებოდეს საქალაქო ქსელი (Metropolitan Area Network (MAN)).




*სურ.3.1. 7 გიგაბიტ Ethernet*

## 3.2. Switch-ის საბაზისო კონფიგურაცია

### *3.2.1 კომუტატორის ჩატვირთვის თანმიმდევრობა*

Cisco კომუტატორის ჩართვის შემდეგ, ის გადის შემდეგ ჩატვირთვის ეტაპებს:

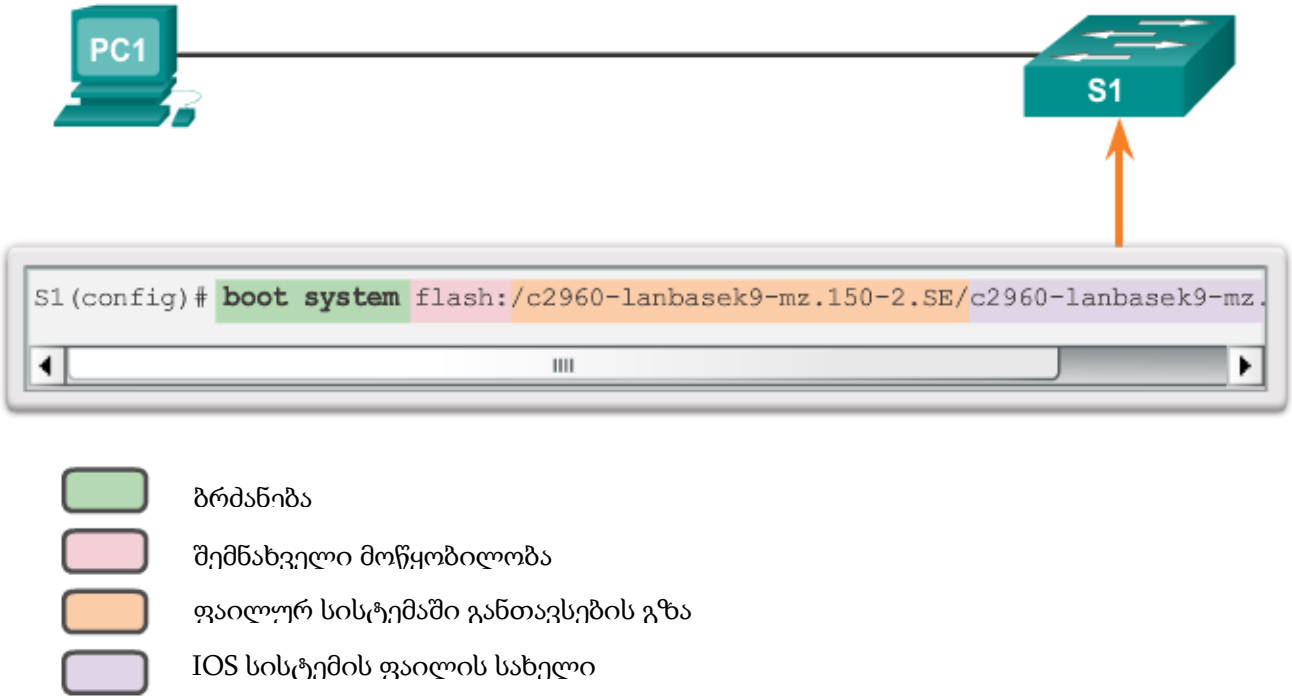
1. პირველი, კომუტატორი ტვირთავს თვითტესტირების პროგრამას (POST), რომელიც ინახება ROM მეხსიერებაში. POST ამოწმებს CPU-ს ქვესისტემას. ის ტესტირებას უტარებს ცენტრალურ პროცესორს (CPU), ოპერატიულ მეხსიერებას და ფლეშ მეხსიერების იმ ნაწილს, რომელიც შეადგენს ფლეშ ფაილურ სისტემას.
2. შემდეგ კომუტატორი ტვირთავს სისტემური ჩამტვირთავის (Boot Loader) პროგრამულ უზრუნველყოფას. სისტემური ჩამტვირთავი არის პატარა პროგრამა, რომელიც მოთავსებულია ROM მეხსიერებაში და ის მაშინვე ეშვება, POST პროცედურის წარმატებით შესრულების შემდეგ.
3. სისტემური ჩამტვირთავი ასრულებს დაბალი დონის ცენტრალური პროცესორის ინიციალიზაციას. ის ინიციალიზაციას ახდენს პროცესორის რეგისტრების, რომელიც აკონტროლებს თუ სად განთავსდება ფიზიკური მეხსიერება, ასევე მეხსიერების რაოდენობის და სისწრაფის.
4. სისტემური ჩამტვირთავი ინიციალიზაციას უკეთებს ფლეშ ფაილურ სისტემას სისტემურ პლატაზე.
5. ბოლოს სისტემური ჩამტვირთავი განათავსებს და ტვირთავს IOS ოპერაციული სისტემის იმიჯს ოპერატიულ მეხსიერებაში და კომუტატორი კონტროლს მთლიანად გადასცემს IOS ოპერაციულ სისტემას.

მთავარი ჩამტვირთავი Cisco IOS სისტემის იმიჯს კომუტატორზე ეძებს შემდეგნაირად: კომუტატორი ცდილობს ავტომატურად ჩატვირთვას BOOT ცვლად გარემოში არსებული ინფორმაციის გამოყენებით. თუ ეს ცვლადი არ არის დაყენებული, კომუტატორი ცდილობს ჩატვირთოს და გამოიყენოს პირველი შემსრულებელი ფაილი, რომელიც შეუძლია რეკურსიული შესრულებით, სიღრმეში ძებნით (Depth-First Search) მთელს ფლეშ ფაილურ სისტემაში. კატალოგების სიღრმისეული ძებნის დროს, თითოეული ქვეკატალოგი სრულად იძებნება, სანამ გაგრძელდება ძებნა ორიგინალ კატალოგში. Catalyst 2960 სერიის

კომუტატორებზე იმიჯ-ფაილი შედის იმ კატალოგში, რომელსაც აქვს იგივე სახელი რაც იმიჯ-ფაილს (გარდა .bin გაფართოების ფაილებისა).

IOS ოპერაციული სისტემა შემდეგ ახორციელებს ინტერფეისების ინიციალიზაციას Cisco IOS-ის ბრძანებების გამოყენებით, რომელსაც პოულობს NVRAM-ში შენახული კონფიგურაციის ფაილსა და საწყის კონფიგურაციაში.

მიმდინარე IOS ჩამტვირთავი ფაილის სანახავად გამოიყენეთ show bootvar (ძველ IOS-ის ვერსიებში show boot) ბრძანება.



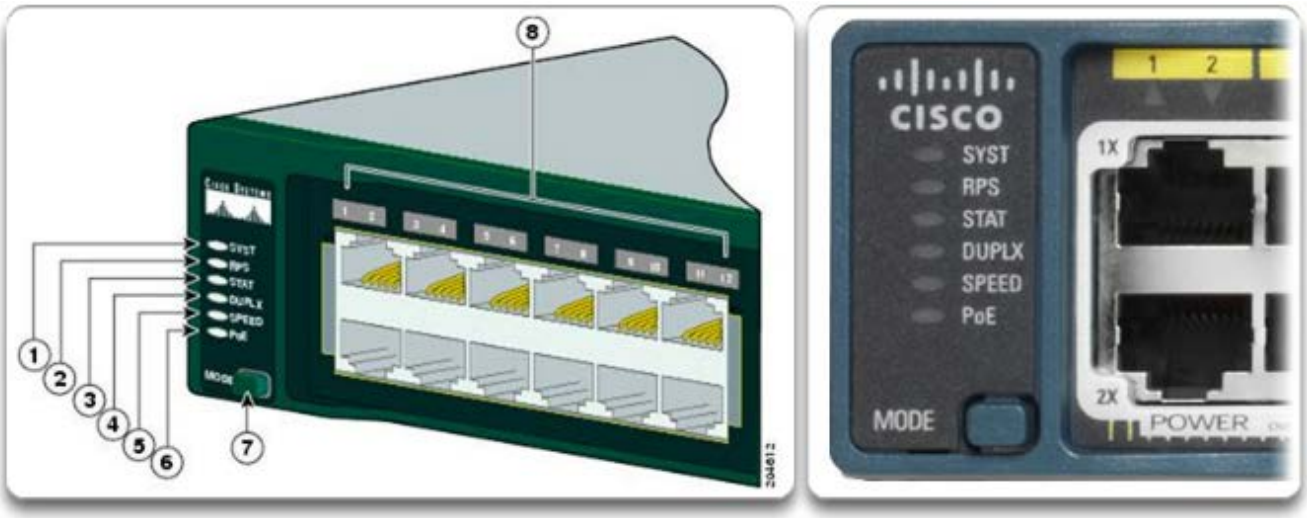

*სურ.3.2.1. 1 BOOT ცვლადი გარემოს კონფიგურაცია*

### *3.2.2 კომუტატორის LED ინდიკატორები*

Cisco Catalyst კომუტატორებს აქვთ LED მაჩვენებლების ნათების რამდენიმე სტატუსი. კომუტატორის LED ნათურები შესაძლოა გამოყენებულ იქნას კომუტატორის აქტიურობისა და წარმადობის მონიტორინგისთვის. სხვადასხვა მოდელის და მახასიათებლების კომუტატორებს შეიძლება ჰქონდეთ განსხვავებული LED ნათურები და წინა პანელზე მათი განლაგებაც შეიძლება განსხვავებული იყოს.

3.2.2.1 სურათზე ნაჩვენებია Cisco Catalyst 2960 კომუტატორის LED ნათურები და რეჟიმის ღილაკი (Mode Button). რეჟიმის ღილაკი გამოიყენება პორტის მდგომარეობის, დუპლექსის, სიჩქარის და PoE (თუ აქვს მხარდაჭერა) პორტის LED ნათურების სტატუსის გადასართველად. ქვემოთ აღწერილია LED ინდიკატორების დანიშნულება და მათი ფერების მნიშვნელობა:

- **System LED** - აჩვენებს იღებს თუ არა სისტემა კვებას და ფუნქციონირებს თუ არა ნორმალურად. თუ LED ნათურა გათიშულია, ნიშნავს რომ სისტემა არაა ჩართული. თუ ნათურა მწვანეა, ე.ი სისტემა ნორმალურად ფუნქციონირებს. თუ LED ნათურა ქარვისფერია, სისტემა იღებს კვებას, მაგრამ არ ფუნქციონირებს ნორმალურად.
- **Redundant Power System (RPS) LED** - აჩვენებს RPS სტატუსს. თუ LED ნათურა გამორთულია, RPS არ არის ჩართული ან არასწორადაა შეერთებული. თუ LED ნათურა მწვანეა, RPS დაკავშირებულია და მზადაა მიაწოდოს სარეზერვო კვება. თუ მწვანე LED ნათურა ციმციმებს, RPS დაკავშირებულია, მაგრამ მიუწვდომელია, რადგან ის აწვდის კვებას სხვა მოწყობილობას. თუ ნათურა ქარვისფერია, RPS იმყოფება ლოდინის რეჟიმში ან დაზიანებულ მდგომარეობაში. თუ ნათურა ქარვისფრად ციმციმებს, მაშინ კომუტატორის შიდა კვების ბლოკი დაზიანებულია და RPS აწვდის ძაბვას.
- **Port Status LED** - მიუთითებს, რომ პორტის მდგომარეობის რეჟიმია არჩეული, როცა LED ნათურა არის მწვანე. ეს არის ნაგულისხმევი რეჟიმი. როცა არჩეულია ეს რეჟიმი, პორტის LED ნათურები გამოხატავენ ფერებს განსხვავებული მნიშვნელობებით. თუ ნათურა არ ანთია, ე.ი არ არსებობს კავშირი ან პორტი გათიშულია. თუ LED ნათურა მწვანეა, კავშირი არსებობს. თუ ნათურა მწვანედ ციმციმებს, ე.ი პორტი აქტიურია და აგზავნის ან იღებს მონაცემებს. თუ ნათურა იცვლება მწვანე-ქარვისფერში, მაშინ დაზიანებულია კავშირი. თუ LED ნათურა ქარვისფერია, პორტი დაბლოკილია რაც გვატყობინებს, რომ ციკლი არ არსებობს გადაგზავნის დომეინში და არ ხდება მონაცემების გადაგზავნა (როგორც წესი, პორტი ასეთ მდგომარეობაში იმყოფება გააქტიურების შემდეგ პირველი 30 წამი). თუ LED ნათურა ქარვისფრად ციმციმებს, პორტი არის დაბლოკილი, რათა თავიდან იქნას აცილებული ციკლის შესაძლებლობა გადასაგზავნ დომეინში.




| Catalyst 2960 Switch LEDs | | | |
|---|---|---|---|
| 1 | The system LED | 5 | The port speed LED |
| 2 | The RPS LED (if RPS is supported on the switch) | 6 | The PoE status LED (if PoE is supported on the switch) |
| 3 | The port status LED (This is the default mode.) | 7 | The Mode button |
| 4 | The port duplex mode LED | 8 | The port LEDs |

*სურ.3.2.2. 1. კომპუტატორის LED ნათურები*

- **Port Duplex LED** - მიუთითებს, რომ პორტის დუპლექსის რეჟიმია არჩეული, მაშინ როცა LED ნათურა არის მწვანე. როცა არჩეულია ეს რეჟიმი, იმ პორტის ნათურები, რომლებიც გამორთულია, იმყოფებიან ნახევარ-დუპლექს რეჟიმში. თუ პორტის ნათურა არის მწვანე, ეს ნიშნავს რომ ის იმყოფება სრული-დუპლექსის რეჟიმში.
- **Port Speed LED** - მიუთითებს, რომ პორტის სიჩქარის რეჟიმია არჩეული. როცა არჩეულია ეს რეჟიმი, პორტის LED ნათურები გამოსახავენ ფერებს სხვადასხვა მნიშვნელობებით.როცა ნათურა გათიშულია, პორტი ფუნქციონირებს 10მგ/წმ სიჩქარით. თუ ნათურა არის მწვანე, მაშინ - 100მგ/წმ. თუ LED ნათურა მწვანედ ციმციმებს, ე.ი პორტი ფუნქციონირებს 1000მგ/წმ სიჩქარით.

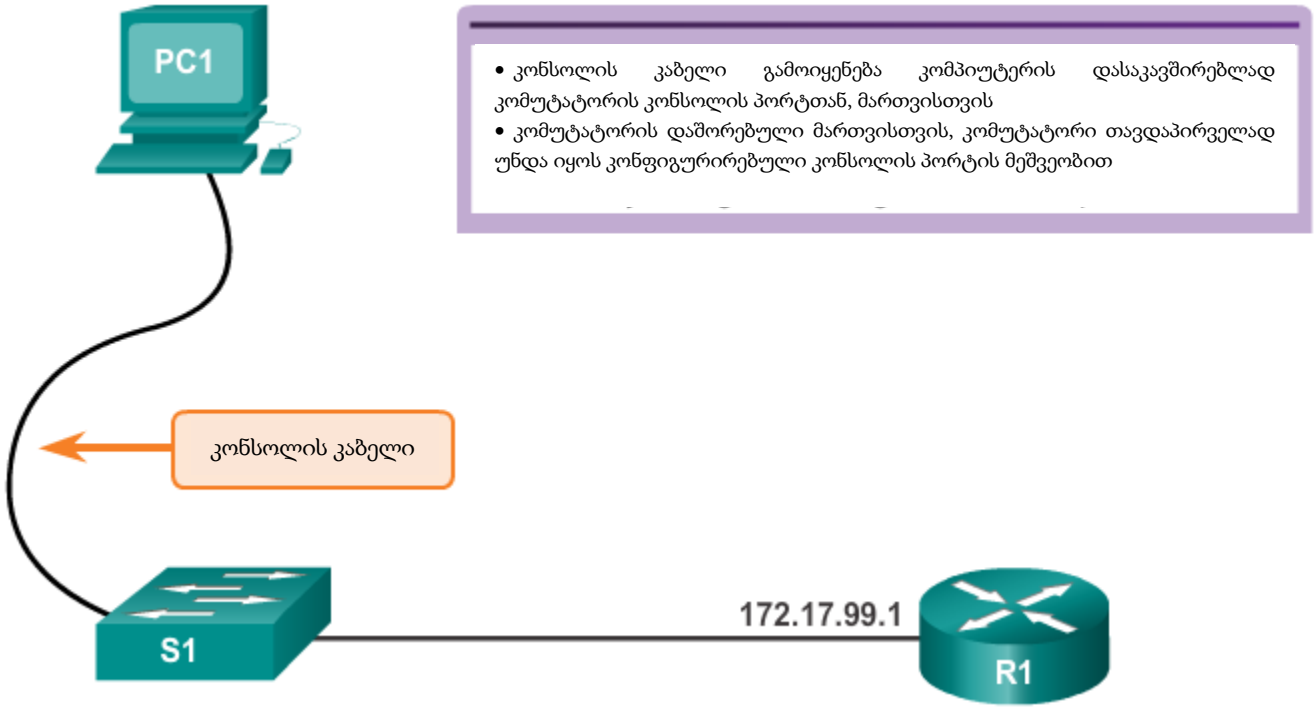
- **Power over Ethernet (PoE) Mode LED** - თუ კომუტატორს აქვს PoE-ს მხარდაჭერა, მაშინ PoE რეჟიმი იქნება აქტიური. თუ LED ნათურა გამორთულია, ის მიუთითებს, რომ PoE რეჟიმი არ არის არჩეული და რომ არცერთმა პორტმა არ თქვა უარი ძაბვაზე ან გადავიდა შეცდომის მდგომარეობაში. თუ ნათურა ციმციმებს ქარვისფრად, PoE რეჟიმი არაა არჩეული, მაგრამ მინიმუმ ერთმა პორტმა უარი თქვა ძაბვაზე, ან PoE რეჟიმი დაზიანდა. თუ LED ნათურა მწვანეა, ის მიუთითებს, რომ PoE რეჟიმი არჩეულია და პორტის ნათურები აჩვენებენ ფერებს სხვადასხვა მნიშვნელობით. თუ პორტის LED ნათურა არ ანთია, PoE გამორთულია. თუ პორტი არის მწვანე, PoE ჩართულია. თუ პორტის LED ნათურები იცვლება მწვანედან ქარვისფერში, PoE უარყოფილია, რადგან კვებით უზრუნველყოფილი მოწყობილობების კვების მოთხოვნამ გადააჭარბა კომუტატორის კვების მოცულობას. თუ LED ნათურა ქარვისფრად ციმციმებს, PoE გამორთულია შეცდომის გამო. თუ LED ნათურა ქარვისფერია, PoE გათიშულია პორტზე.

### *3.2.3. მომზადება კომუტატორის გაზისური მართვისთვის*

კომუტატორის დაშორებული მართვის წვდომისთვის მოსამზადებლად, კომუტატორი უნდა იყოს კონფიგურირებული IP მისამართით და ქვექსელის ნიღაბით. გაითვალისწინეთ, რომ დაშორებული ქსელიდან კომუტატორის მართვისთვის, საჭიროა რომ ის კონფიგურირებული იყოს ნაგულისხმევი გასასვლელით (Default Gateway). ეს პროცესი ძალიან გავს ჰოსტ მოწყობილობაზე IP მისამართის ინფორმაციის კონფიგურირებას. 3.2.3 სურათზე, S1 კომუტატორის ვირტუალურ ინტერფეისს (SVI) უნდა ჰქონდეს მინიჭებული IP მისამართი. SVI არის ვირტუალური ინტერფეისი და არა ფიზიკური პორტი კომუტატორზე.

SVI არის ცნება, რომელიც დაკავშირებულია VLAN-ებთან. VLAN-ები არის დანომრილი ლოგიკური ჯგუფები, რომელთაც შეიძლება ჰქონდეთ მინიჭებული ფიზიკური პორტები. კონფიგურაციები და პარამეტრები, რომლებიც გამოყენებულია VLAN-სთვის, ასევე ვრცელდება ყველა იმ პორტზე, რომელიც მინიჭებულია VLAN-ზე.

ნაგულისხმევად, კომუტატორი დაკონფიგურებულია ისე, რომ ჰქონდეს მართვა კონტროლირებად კომუტატორზე VLAN 1-ის საშუალებით. ნაგულისხმევად ყველა პორტი მინიჭებულია VLAN 1-ზე. უსაფრთხოების მიზნით საუკეთესო პრაქტიკაა VLAN 1-სგან განსხვავებული VLAN-ის გამოყენება VLAN-ის მართვისთვის.




*სურ.3.2.3. 1 მომზადება დაშორებული მართვისთვის*

მიაქციეთ ყურადღება, რომ IP მისამართის ეს პარამეტრები გამოიყენება მხოლოდ დაშორებული წვდომისთვის კომუტატორის სამართავად. IP პარამეტრები არ აძლევს კომუტატორს მესამე დონის პაკეტების მარშრუტიზაციის შესაძლებლობას.

### *3.2.4 კომუტატორის ბაზისური წვდომის მართვის კონფიგურაცია IPv4-ით*

**პირველი ეტაპი. მართვადი ინტერფეისის კონფიგურაცია**

კომუტატორისთვის IP მისამართი და ქვექსელის ნიღაბი დაკონფიგურებულია მართვად SVI-ზე, VLAN ინტერფეისის კონფიგურაციის რეჟიმიდან. როგორც ნაჩვენებია 3.2.4.1 სურათზე, interface vlan 99 ბრძანება გამოიყენება ინტერფეისის კონფიგურაციის რეჟიმში შესასვლელად. ip address ბრძანება გამოყენებულია IP მისამართის კონფიგურაციისთვის. no shutdown ბრძანებით აქტიურდება ინტერფეისი. მოცემულ მაგალითში VLAN 99 დაკონფიგურებულია 172.17.99.11 IP მისამართით. VLAN ის შესაქმნელად vlan_id ნომრით 99 და ინტერფეისთან დაკავშირებისთვის, გამოიყენეთ შემდეგი ბრძანებები:

S1(config)# **vlan** *vlan_id*

S1(config-vlan)# **name** *vlan_name*

S1(config-vlan)# **exit**

S1(config)# **interface***interface_id*

S1(config-if)# **switchport access vlan** *vlan_id*

| Cisco კომუტატორის IOS სისტემის ბრძანებები | |
|---|---|
| საერთო კონფიგურაციის რეჟიმში შესვლა | `S1# configure terminal` |
| ინტერფეისის კონფიგურაციის რეჟიმში შესვლა SVI-სთვის | `S1(config)# interface vlan 99` |
| მართვის ინტერფეისის IP მისამართის კონფიგურაცია | `S1(config-if)# ip address 172.17.99.11 255.255.255.0` |
| მართვის ინტერფეისის ჩართვა | `S1(config-if)# no shutdown` |
| პრივილეგირებულ EXEC რეჟიმში დაბრუნება | `S1(config-if)# end` |
| Save the running config to the startup config. | `S1# copy running-config startup-config` |

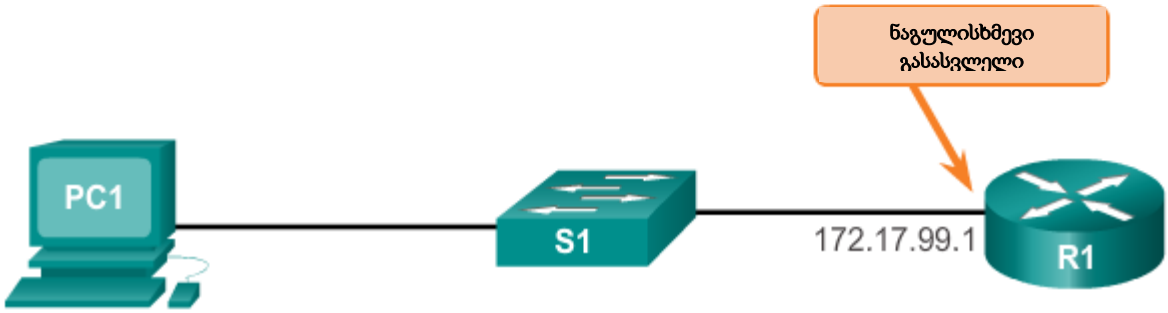
სურ. 3.2.4.1. კომუტატორის მართვის ინტერფეისის კონფიგურაცია

**მეორე ეტაპი. ნაგულისხმევი გასასვლელის (Default Gateway) კონფიგურაცია**

კომუტატორი შეიძლება კონფიგურირებული იყოს ნაგულისხმევი გასასვლელით თუ ის იმართება იმ ქსელებიდან, რომლებიც პირდაპირ არ არიან დაკავშირებული მასთან. ნაგულისხმევი გასასვლელი არის მარშრუტიზატორი, რომელზეც მიერთებულია კომუტატორი. კომუტატორი ადრესატის IP მისამართებთან ერთად გადაუგზავნის თავის IP პაკეტებს ნაგულისხმევ გასასვლელს ლოკალური ქსელის გარეთ. როგორც 3.2.4.2 სურათზეა მოცემული R1 არის S1-ის ნაგულისხმევი გასასვლელი. R1-ის ინტერფეისს, რომელზეც დაკავშირებულია კომუტატორი აქვს 172.17.99.1 IP მისამართი. ეს მისამართი არის ნაგულისხმევი გასასვლელი S1-სთვის.

კომუტატორის ნაგულისხმევი გასასვლელის კონფიგურაციისთვის გამოიყენეთ ip default-gateway ბრძანება. შეიყვანეთ ნაგულისხმევი გასასვლელის IP მისამართი. ნაგულისხმევი გასასვლელი არის მარშრუტიზატორის ინტერფეისის IP მისამართი, რომელზეც დაკავშირებულია კომუტატორი. გამოიყენეთ copy running-config startup-config ბრძანება, კონფიგურაციის სარეზერვო ასლის შესანახად.

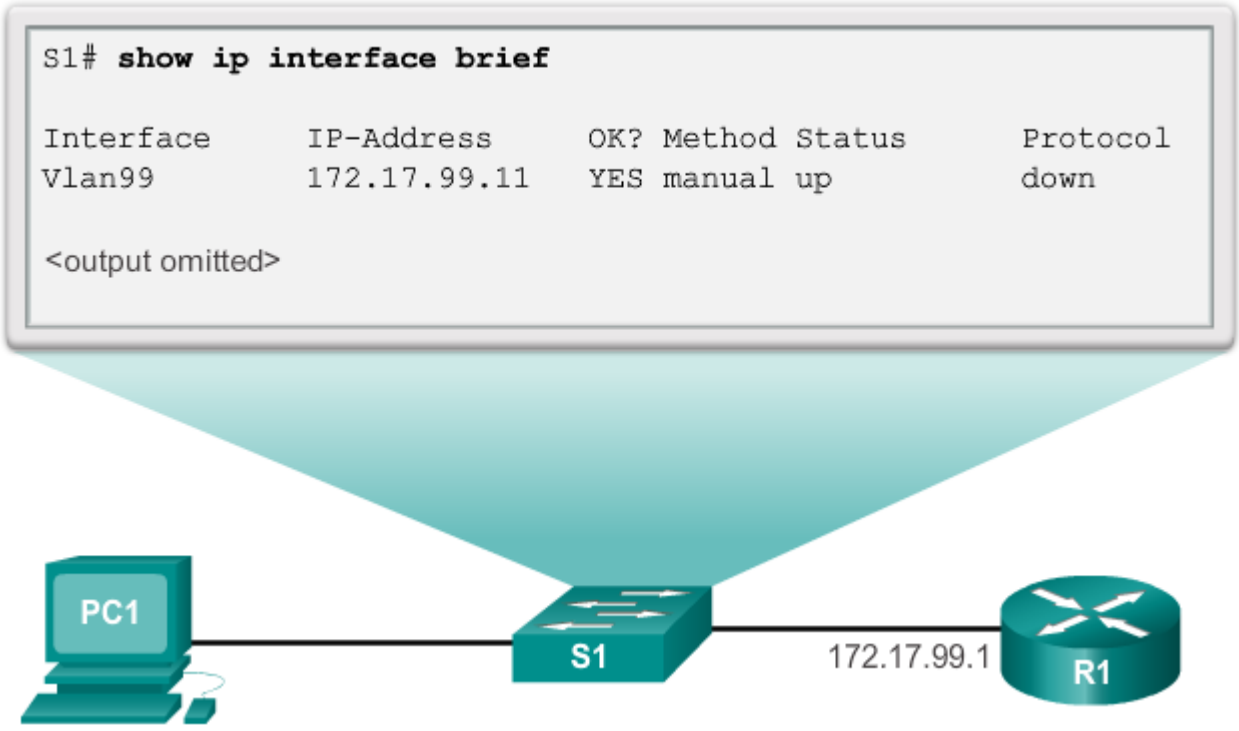
| Cisco კომუტატორის IOS სისტემის ბრძანებები | |
|---|---|
| საერთო კონფიგურაციის რეჟიმში შესვლა | `S1# configure terminal` |
| კომუტატორის ნაგულისხმევი გასასვლელის კონფიგურაცია | `S1(config)# ip default-gateway 172.17.99.1` |
| პრივილეგირებულ EXEC რეჟიმში დაბრუნება | `S1(config)# end` |
| გაშვებული კონფიგურაციის შენახვა სასტარტო კონფიგურაციაში | `S1# copy running-config startup-config` |




სურ. 3.2.4.2. კომუტატორის ნაგულისხმევი გასასვლელის კონფიგურაცია

### მესამე ეტაპი. კონფიგურაციის შემოწმება

როგორც 3.2.4.3 სურათზეა ნაჩვენები, show ip interface brief ბრძანებაა საჭირო ფიზიკური და ვირტუალური ინტერფეისების მდგომარეობის განსაზღვრისთვის. გამოტანილი ინფორმაცია ადასტურებს, რომ VLAN 99 ინტერფეისი დაკონფიგურებულია IP მისამართით და ქვექსელის ნიღაბით და რომ ის ფუნქციონირებს.




სურ. 3.2.4.3. კომუტატორის მართვის ინტერფეისის კონფიგურაციის შემოწმება

პრაქტიკული სამუშაო - კომუტატორის (Switch) ბაზისური პარამეტრების კონფიგურაცია

**ტოპოლოგია:**


**მისამართების ცხრილი:**

| მოწყობილობა | ინტერფეისი | IP მისამართი | ქვექსელის ნიღაბი | ნაგულისხმევი გასასვლელი |
|---|---|---|---|---|
| S1 | VLAN 99 | 192.168.1.2 | 255.255.255.0 | 192.168.1.1 |
| PC-A | NIC | 192.168.1.10 | 255.255.255.0 | 192.168.1.1 |

**დავალებები:**

**ნაწილი №1: ქსელთან მიერთება და კომუტატორის ნაგულისხმევი კონფიგურაციის შემოწმება**

**ნაწილი №2: ქსელური მოწყობილობის ბაზისური პარამეტრების კონფიგურაცია**

- კომუტატორის ბაზისური პარამეტრების მომართვა
- **PC-A** კომპიუტერის **IP** მისამართი მომართვა

**ნაწილი №3: შეამოწმეთ და გამოსცადეთ ქსელური შეერთება**

- მოწყობილობის კონფიგურაციის დათვალიერება
- **Ping** ბრძანებით საბოლოო მოწყობილობასთან კავშირის შემოწმება
- დაშორებული მართვის შესაძლებლობების შემოწმება, **Telnet**-ის გამოყენებით
- კომუტატორის გაშვებული კონფოგურაციის ფაილის შენახვა

## ნაწილი №4: MAC მისამართების ცხრილის მართვა

- ჰოსტის **MAC** მისამართის ჩაწერა
- კომუტატორის მიერ შესაწავლილი **MAC** მისამართების განსაზღვრა
- **show mac address-table** ბრძანების პარამეტრების ჩამოთვლა
- სტატიკური **MAC** მისამართის დაყენება

## ზოგადი ინფორმაცია / სცენარი

**Cisco** კომუტატორები შეიძლება კონფიგურირებული იყვნენ სპეციალური **IP** მისამართით, ცნობილი როგორც კომუტატორის ვირტუალური ინტერფეისი (**SVI**). **SVI** ან მართვის მისამართი შეიძლება გამოყენებულ იქნას კომუტატორთან დაშორებული წვდომისათვის, პარამეტრების დასათვალიერებლად ან დასაკონფიგურებლად. თუ **VLAN 1 SVI**-ს აქვს მინიჭებული **IP** მისამართი, ნაგულისხმევად, **VLAN 1**-ში ყველა პორტს აქვს წვდომა **SVI** მართვის **IP** მისამართთან.

მოცემულ ლაბორატორიულ დავალებაში, თქვენ უნდა ააწყოთ მარტივი ტოპოლოგია **Ethernet LAN** კაბელის გაყვანის საშუალებით და მოახდინოთ წვდომა Cisco კომუტატორთან კონსოლის ან დაშორებული მართვის მეთოდების გამოყენებით. კომუტატორის ბაზისური პარამეტრების მომართვამდე თქვენ უნდა გამოიკვლიოთ კომუტატორის ნაგულისხმევი კონფიგურაცია. კომუტატორის ბაზისური პარამეტრები მოიცავს: მოწყობილობის სახელს, ინტერფეისის აღწერას, ლოკალურ პაროლებს, დღის შეტყობინების (**MOTD**) ბანერს, **IP** დამისამართებას, სტატიკური **MAC** მისამართის დაყენებას და დაშორებული კომუტატორის მართვისთვის, მართვის **IP** მისამართის გამოყენების დემოსტრირებას. ტოპოლოგია შედგება ერთი კომუტატორისა და ერთი ჰოსტისაგან, რომლებიც იყენებენ მხოლოდ **Ethernet** და კონსოლის პორტებს.

**შენიშვნა:** კომუტატორად გამოყენებულია **Cisco Catalyst 2960** მოდელი, **Cisco IOS Release 15.0(2)** (**lanbasek9** იმიჯი) ოპერაციული სისტემით. შესაძლებელია სხვა კომუტატორების და სხვა **Cisco IOS** ვერსიების გამოყენებაც. კომუტატორის მოდელის და **Cisco IOS**

ვერსიების მიხედვით, ხელმისაწვდომი ბრძანებები და წარმოებული შედეგები შეიძლება განსხვავებული იმისგან, რაც ნაჩვენებია ამ ლაბორატორიულ დავალებაში.

**შენიშვნა:** დარწმუნდით რომ კომუტატორის პარამეტრები წაშლილია და არ არსებობს საწყისი გამშვები კონფიგურაცია. იხილეთ დანართი **A**, მოწყობილობების ინიციალიზაციისა და ხელახლა ჩატვირთვის პროცედურებისთვის.

**მოთხოვნილი რესურსები:**

- ერთი კომუტატორი (**Cisco 2960** მოდელი **Cisco IOS Release 15.0(2) lanbasek9** იმიჯის ვერსიით ან მსგავსი)
- ერთი პერსონალური კომპიუტერი (**Windows 7**, **Vista** ან **XP** ოპერაციული სისტემა, ტერმინალის ემულაციის პროგრამასთან ერთად, **Tera Term** და **Telnet** შესაძლებლობების ჩათვლით)
- კონსოლის კაბელი, რომელიც საჭიროა კონსოლის პორტის დახმარებით Cisco IOS მოწყობილობის კონფიგურაციისთვის
- **Ethernet** კაბელი, ისეთი როგორიც ნაჩვენებია ტოპოლოგიაში.

**ნაწილი №1: ქსელთან მიერთება და კომუტატორის ნაგულისხმევი კონფიგურაციის შემოწმება**

პირველ ნაწილში, თქვენ უნდა გამართოთ ქსელური ტოპოლოგია და შეამოწმოთ კომუტატორის ნაგულისხმევი კონფიგურაცია.

**პირველი ეტაპი : ქსელთან მიერთება ტოპოლოგიაზე ნაჩვენები სქემის მიხედვით**

ა. ტოპოლოგიაზე მოცემული სქემის მიხედვით შექმენით კონსოლ შეერთება. ჯერჯერობით არ დააკავშიროთ PC-A Ethernet კაბელი.

**შენიშვნა:** თუ იყენებთ Netlab-ს, თქვენ შეგიძლიათ გამორთოთ F0/6 ინტერფეისი S1-ზე, რომელსაც ექნება იგივე ეფექტი, რაც PC-A-ს S1-თან არ დაკავშირების შემთხვევაში.

ბ. შექმენით კონსოლ შეერთება PC-A-დან კომუტატორთან, Tera Term ან სხვა ტერმინალის ემულაციის პროგრამით.

რატომ იყენებთ კონსოლის შეერთებას, კომუტატორის საწყისი კონფიგურაციისთვის? რატომ არ არის შესაძლებელი კომუტატორთან დაკავშირება Telnet-ის ან SSH-ის გამოყენებით?

_____

_____

## მეორე ეტაპი: კომუტატორის ნაგულისხმევი კონფიგურაციის შემოწმება

ამ ეტაპზე თქვენ უნდა გამოიკვლიოთ კომუტატორის ნაგულისხმევი პარამეტრები, კომუტატორის მიმდინარე კონფიგურაციის, IOS-ის ინფორმაციის, ინტერფეისის თვისებების, VLAN-ის ინფორმაციის და ფლეშ მეხსიერების ჩათვლით.

თქვენ გაქვთ წვდომა კომუტატორის IOS-ის ყველა ბრძანებასთან, პრივილეგირებულ EXEC რეჟიმში. პრივილეგირებულ EXEC რეჟიმთან წვდომა შეიძლება შეზღუდული იყოს პაროლის დაცვით, რათა აკრძალულ იქნას არაავტორიზებული გამოყენება, იმიტომ რომ ის იძლევა პირდაპირ წვდომას გლობალური კონფიგურაციის რეჟიმთან და ბრძანებებთან, რომლებიც გამოიყენება საოპერაციო პარამეტრების კონფიგურაციისთვის. მოგვიანებით თქვენ დააყენებთ პაროლებს მოცემულ ლაბორატორიულ დავალებაში.

პრივილეგირებული EXEC რეჟიმის ბრძანების მომართვა მოიცავს იმ ბრძანებებს, რომელიც შედის მომხმარებლის EXEC რეჟიმში, ისევე როგორც **configure** ბრძანება, რომლის საშუალებითაც ხდება წვდომის მოპოვება დანარჩენი რეჟიმების ბრძანებასთან. გამოიყენეთ enable ბრძანება პრივილეგირებულ EXEC რეჟიმში შესასვლელად.

ა. დავუშვათ რომ კომუტატორს არ ჰქონდა კონფიგურაციის ფაილი შენახული ენერგოდამოუკიდებელ შემთხვევითი წვდომის მეხსიერებაში (NVRAM), თქვენ

მოხვდებით კომუტატორის მომხმარებლის EXEC რეჟიმის ბრძანებათა ზოლში Switch> ბრძანებით. გამოიყენეთ **enable** ბრძანება პრივილეგირებულ EXEC რეჟიმში შესასვლელად.

Switch> **enable**

Switch#

აღსანიშნავია, რომ კონფიგურაციაში შეიცვალა ბრძანებათა ზოლი, პრივილეგირებული EXEC რეჟიმის ასახვისთვის.

შეამოწმეთ სუფთა კონფიგურაციის ფაილი პრივილეგირებული EXEC რეჟიმის **show running-config** ბრძანებით. თუ ადრე მოხდა კონფიგურაციის ფაილის შენახვა, ის უნდა წაიშალოს. კომუტატორის მოდელის და IOS ვერსიის მიხედვით, თქვენი კონფიგურაცია შეიძლება გამოიყურებოდეს ოდნავ განსხვავებულად. თუმცა, შეიძლება არ იყოს კონფიგურირებული პაროლები და IP მისამართი. თუ თქვენს კომუტატორს არ აქვს ნაგულისხმევი კონფიგურაცია, წაშალეთ და ხელახლა ჩატვირთეთ კომუტატორი.

შენიშვნა: დანართი A დეტალებად ყოფს მოწყობილობის ინიციალიზაციისა და ხელახლა ჩატვირთვის ეტაპებს.

ბ. შეამოწმეთ მიმდინარე გაშვებული კონფიგურაცია.

Switch# **show running-config**

რამდენი FastEthernet ინტერფეისი აქვს 2960 კომუტატორს? _____

რამდენი Gigabit Ethernet ინტერფეისი აქვს 2960 კომუტატორს? _____

vty ხაზებისთვის, რა არის მნიშვნელობების დიაპაზონი? _____

გ. შეამოწმეთ გამშვები კონფიგურაცია NVRAM-ში.

Switch# **show startup-config**

startup-config is not present

რატომ გამოვიდა ასეთი შეტყობინება? _____

დ. შეამოწმეთ SVI-ის მახასიათებლები, VLAN 1-სთვის.

Switch# **show interface vlan1**

მინიჭებულია თუ არა IP მისამართი VLAN 1-ზე?

რა არის SVI-ის MAC მისამართი? პასუხი იქნება სხვადასხვა _____

ინტერფეისი არის თუ არა ჩართული? _____

ე. დაადგინეთ SVI VLAN 1-ის IP თვისებები.

Switch# **show interface vlan1**

რა შედეგს ხედავთ? _____

_____

ვ. შეაერთეთ PC-A Ethernet კაბელი კომუტატორის მე-6 პორტზე და გამოიკვლიეთ SVI VLAN 1-ის IP თვისებები. ადროვეთ კომუტატორს და პერსონალურ კომპიუტერს შეათანხმონ დუპლექსის და სიჩქარის პარამეტრები.

**შენიშვნა:** თუ იყენებთ Netlab-ს, ჩართეთ F0/6 ინტერფეისი S1 კომუტატორზე.

Switch# **show interface vlan1**

რა შედეგს ხედავთ? _____

_____

ზ. გამოიტანეთ კომუტატორის Cisco IOS ვერსიის ინფორმაცია.

Switch# **show version**

Cisco IOS-ის რომელი ვერსიაა გაშვებული კომუტატორზე? _____

რა არის სისტემური იმიჯის სახელი? _____

რა არის კომუტატორის ბაზისური MAC მისამართი? _____

თ. შეამოწმეთ FastEthernet ინტერფეისის ნაგულისხმევი თვისებები, რომელსაც იყენებს PC-A.

Switch# **show interface f0/6**

ინტერფეისი ჩართულია თუ გამორთული? _____

რა მოქმედებითაა შესაძლებელი ინტერფეისის ჩართვა? _____

ინტერფეისის MAC მისამართი _____

ინტერფეისის დუპლექსისა და სიჩქარის პარამეტრები _____

ი. კომუტატორზე შეამოწმეთ ნაგულისხმევი VLAN-ის პარამეტრები.

Switch# **show vlan**

რა არის VLAN 1-ის ნაგულისხმევი სახელი? _____

რომელ პორტებს მოიცავს VLAN-ი? _____

აქტიურია თუ არა VLAN 1? _____

რომელი ტიპის VLAN-ია ნაგულისხმევი VLAN-ი? _____

კ. გამოიკვლიეთ ფლეშ მეხსიერება.

გამოიყენეთ მოცემულ ბრძანებათაგან ერთ-ერთი, ფლეშ დირექტორიის შემცველობის სანახავად.

Switch# **show flash**

Switch# **dir flash:**

სახელის ბოლოს ფაილებს გააჩნიათ გაფართოება, ისეთი როგორიცაა .bin. დირექტორიებს არ აქვთ ფაილის გაფართოება.

რა არის Cisco IOS იმიჯის ფაილის სახელი? _____

## ნაწილი №2: ქსელური მოწყობილობის ბაზისური პარამეტრების კონფიგურაცია

მეორე ნაწილში თქვენ დააკონფიგურებთ კომუტატორის და პერსონალური კომპიუტერის ბაზისურ პარამეტრებს.

### პირველი ეტაპი : კომუტატორის ბაზისური პარამეტრების კონფიგურაცია: ჰოსტის სახელი, ლოკალური პაროლები, MOTD ბანერი, მართვის მისამართი და Telnet-თან წვდომა.

ამ ეტაპზე თქვენ მოგიწევთ პერსონალური კომპიუტერის და კომუტატორის ბაზისური პარამეტრების კონფიგურაცია, როგორიცაა ჰოსტის სახელი და კომუტატორის მართვის SVI IP მისამართი. IP მისამართის მინიჭება კომუტატორზე, ეს არის მხოლოდ პირველი ეტაპი. როგორც ქსელის ადმინისტრატორმა, თქვენ უნდა განსაზღვროთ, თუ როგორ მოხდება კომუტატორის მართვა. Telnet და SSH არის ორი ყველაზე თანამედროვე მართვის მეთოდი. თუმცა Telnet-ი არ არის დაცული პროტოკოლი. ყველა ინფორმაცია, რომელიც მოძრაობს ორ მოწყობილობას შორის იგზავნება ღია ტექსტით. პაროლები და სხვა მნიშვნელოვანი ინფორმაცია მარტივად შეიძლება იქნას დათვალიერებული, თუ გამოჭერილ იქნა პაკეტის სნიფერის მიერ.

ა. ვივარაუდოთ რომ კომუტატორს არ აქვს შენახული კონფიგურაციის ფაილი NVRAM-ში, დარწმუნდით რომ იმყოფებით პრივილეგირებულ EXEC რეჟიმში. შეიყვანეთ **enable** ბრძანება, თუ ბრძანებათა სტრიქონი დაუბრუნდა Switch> მდგომარეობას.

Switch# **enable**

Switch#

ბ. შედით გლობალური კონფიგურაციის რეჟიმში.

Switch# **configure terminal**

Enter configuration commands, one per line. End with CNTL/Z.

Switch(config)#

ბრძანებათა სტრიქონი ისევ შეიცვალა გლობალური კონფიგურაციის რეჟიმის ასახვისთვის.

გ. კომუტატორზე სახელის მინიჭება.

Switch(config) # **hostname S1**

S1(config)#

დ. პაროლის შიფრაციის კონფიგურაცია

S1(config)# **service password-encryption**

S1(config)#

ე. **class** საიდუმლო პაროლის მინიჭება პრივილეგირებულ EXEC რეჟიმთან წვდომისთვის

S1(config)# **enable secret class**

S1(config)#

ვ. არასასურველი DNS ძიების აკრძალვა.

S1(config)# **no ip domain-lookup**

S1(config)#

ზ. MOTD ბანერის კონფიგურაცია

S1(config)# **banner motd #**

Enter Text message. End with the character '#'.

**Unauthorized access is strictly prohibited. #**

თ. შეამოწმეთ თქვენი წვდომის პარამეტრები რეჟიმებს შორის გადასვლით.

S1(config)# **exit**

S1#

*Mar 1 00:19:19.490: %SYS-5-CONFIG_I: configured from console by console

S1# **exit**

S1 con0 is now available

**Press Return to get started**

Unauthorized access is strictly prohibited

S1>

რომელი კლავიშთა კომბინაცია გამოიყენება გლობალური კონფიგურაციის რეჟიმიდან პრივილეგირებულ EXEC რეჟიმში სწრაფად გადასასვლელად? _____

ი. მომხმარებლის EXEC რეჟიმიდან დაბრუნდით პრივილეგირებულ EXEC რეჟიმში. როცა მოთხოვნილი იქნება შეიყვანეთ **class** პაროლი.

S1> **enable**

password:

S1#

**შენიშვნა:** პაროლი შეყვანის დროს არ ჩანს.

კ. შედით გლობალური კონფიგურაციის რეჟიმში, კომუტატორის SVI IP მისამართის დასაყენებლად. ეს საშუალებას მოგვცემს დაშორებულად ვმართოთ კომუტატორი.

სანამ შეგიძლიათ S1-ის მართვა დაშორებულად PC-A-დან, თქვენ უნდა მიანიჭოთ კომუტატორს IP მისამართი. კომუტატორში ნაგულისხმევი კონფიგურაცია არის ის, რომ მისი მართვა შესაძლებელია VLAN 1-დან.

თუმცა, კომუტატორის ბაზისური კონფიგურაციისთვის საუკეთესო პრაქტიკაა, მართვადი VLAN-ის შეცვლა VLAN 1-სგან განსხვავებული VLAN-ით.

მართვის მიზნებისთვის, გამოიყენეთ VLAN 99. VLAN 99 არჩევა არის თვითნებური და არანაირად არ გულისხმობს, რომ თქვენ ყოველთვის უნდა გამოიყენოთ VLAN 99.

პირველ რიგში, შექმენით ახალი VLAN 99 კომუტატორზე. შემდეგ მომართეთ კომუტატორის IP მისამართი 192.168.1.2-ით, 255.255.255.0 ქვექსელის ნიღაბთან ერთად, VLAN 99 შიდა ვირტუალურ ინტერფეისზე.

S1# **configure terminal**

S1(config)# **vlan 99**

S1(config-vlan)# **exit**

S1(config)# **interface vlan99**

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan99, changed state to down

S1(config-if)# **ip address 192.168.1.2 255.255.255.0**

S1(config-if)# **no shutdown**

S1(config-if)# **exit**

S1(config)#

აღსანიშნავია, რომ VLAN99 ინტერფეისი გათიშულ მდგომარეობაშია, სანამ არ გაუშვებთ ბრძანებას no shutdown. ინტერფეისი ახლა გათიშულია, რადგან არცერთი კომუტატორის პორტი არ არის მინიჭებული VLAN 99-თან.

ლ. ყველა მომხმარებლის პორტის მინიჭება VLAN 99-თან.

S1(config)# **interface range f0/1 – 24, g0/1 – 2**

S1(config-if-range)# **switchport access vlan 99**

S1(config-if-range)# **exit**

S1(config)#

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan99, changed state to up

ჰოსტსა და კომუტატორს შორის კავშირის დასამყარებლად, პორტები, რომლებსაც ჰოსტი იყენებს უნდა იყოს იგივე VLAN-ში, რომელშიც კომუტატორი. ყურადღება მიაქციეთ ზედა შედეგს, VLAN 1 გადადის გათიშულ მდგომარეობაში იმიტომ რომ, არცერთი პორტი არაა მინიჭებული VLAN 1-თან. რამდენიმე წამის შემდეგ, VLAN 99 გადადის UP-ში რადგან მინიმუმ ერთი აქტიური პორტი (F0/6 დაკავშირებული PC-A-სთან) უკვე მინიჭებულია VLAN 99-თან.

მ. გაუშვით show vlan brief ბრძანება იმის შესამოწმებლად ყველა მომხმარებლის პორტი არის თუ არა VLAN 99-ში.

S1# **show vlan brief**

| VLAN | Name | Status | Ports |
|---|---|---|---|
| 1 | default | active | |
| 99 | VLAN0099 | active | Fa0/1, Fa0/2, Fa0/3, Fa0/4<br>Fa0/5, Fa0/6, Fa0/7, Fa0/8<br>Fa0/9, Fa0/10, Fa0/11, Fa0/12<br>Fa0/13, Fa0/14, Fa0/15, Fa0/16<br>Fa0/17, Fa0/18, Fa0/19, Fa0/20<br>Fa0/21, Fa0/22, Fa0/23, Fa0/24<br>Gi0/1, Gi0/2 |
| 1002 | fddi-default | act/unsup | |
| 1003 | token-ring-default | act/unsup | |
| 1004 | fddinet-default | act/unsup | |
| 1005 | trnet-default | act/unsup | |

ნ. დააკონფიგურეთ S1-ის ნაგულისხმევი გასასვლელი. თუ ნაგულისხმევი გასასვლელი არ არის დაყენებული, ვერ მოხერხდება კომუტატორის მართვა დაშორებული ქსელიდან,

იმიტომ რომ მოშორებით არის ერთზე მეტი როუტერი. ის ვერ უპასუხებს დაშორებული ქსელიდან მომავალ Ping ბრძანებებს. იმის გამო, რომ მოცემული დავალება არ მოიცავს გარე IP გასასვლელს, ვივარაუდოთ, რომ თქვენ საბოლოოდ დააკავშირეთ LAN ქსელი მარშრუტიზატორთან, გარე წვდომისათვის. ვივარაუდოთ, რომ LAN ინტერფეისი მარშრუტიზატორზე არის 192.168.1.1, მომართეთ ნაგულისხმევი გასასვლელი კომუტატორისთვის.

S1(config)# ip default-gateway 192.168.1.1

S1 (config) #

ო. შეიძლება ასევე აკრძალული იყოს კონსოლის პორტთან წვდომა. ნაგულისხმევი კონფიგურაცია არის ის, რომ დაშვებულ იქნას ყველა კონსოლის შეერთება, რომელიც პაროლს არ საჭიროებს. წყვეტის ბრძანებებიდან კონსოლის შეტყობინებების თავიდან ასაცილებლად, გამოიყენეთ **logging synchronous** ვარიანტი.

S1(config) # **line con 0**

S1( config-line) # **password cisco**

S1( config-line) # **login**

S1( config-line) # **logging synchronous**

S1( config-line) # **exit**

S1( config) #

პ. კომუტატორისთვის მომართეთ ვირტუალური ტერმინალის (vty) ხაზები, Telnet-თან წვდომის დასაშვებად. თუ არ დააკონფიგურებთ vty პაროლს, ვერ შეძლებთ კომუტატორთან Telnet-ით დაკავშირებას.

S1(config) # **line vty 0 15**

S1(config-line) # **password cisco**

S1(config-line) # **login**

S1(config-line) # **end**

S1#

*Mar 1 00:06:11.590: %SYS-5-CONFIG_I: Configured from console by console

რატომ არის **login** ბრძანება მოთხოვნილი? _____

**მეორე ეტაპი: PC-A კომპიუტერზე IP მისამართის კონფიგურაცია.**

მიანიჭეთ IP მისამართი და ქვექსელის ნიღაბი პერსონალურ კომპიუტერს, ისე როგორც ნაჩვენებია მისამართების ცხრილში. პროცედურის შემოკლებული ვერსია მოცემულია ქვემოთ. ნაგულისხმევი გასასვლელი არ არის მოთხოვნილი ამ ტოპოლოგიაში; თუმცა თქვენ მაინც უნდა შეიყვანოთ 192.168.1.1, S1-ის მარშრუტიზატორთან კავშირის სიმულაციისთვის.

1) დააჭირეთ Windows-ის **Start** ღილაკს > შემდეგ **Control Panel**.

2) ჩამოშალეთ **View By** მენიუ და აირჩიეთ **Small icons**.

3) აირჩიეთ **Network and Sharing Center**> **Change adapter settings**.

4) მონიშნეთ **Local Area Network Connection** განყოფილება, დააჭირეთ მაუსის მარჯვენა ღილაკს და აირჩიეთ **Properties**.

5) აირჩიეთ **Internet Protocol Version 4** (**TCP/IPv4**) > **Properties**.

6) დააჭირეთ **Use the following IP address** ღილაკს და შეიყვანეთ IP მისამართი და ქვექსელის ნიღაბი.

**ნაწილი №3: ქსელის კავშირის შემოწმება და ტესტირება**

მესამე ნაწილში თქვენ უნდა მოახდინოთ კომუტატორის კონფიგურაციის შემოწმება და დოკუმენტირება, ჩაუტარეთ ტესტირება PC-A კომპიუტერის და S1 კომუტატორს შორის შეერთებას, ასევე დატესტეთ კომუტატორის დაშორებული მართვის შესაძლებლობა.

**პირველი ეტაპი: კომუტატორის კონფიგურაციის გამოტანა.**

PC-A-ზე თქვენი კონსოლის კავშირით, გამოიტანეთ და შეამოწმეთ თქვენი კომუტატორის კონფიგურაცია. **Show run** ბრძანებას გამოაქვს მთელი გაშვებული კონფიგურაცია ერთ გვერდზე. გამოიყენეთ Space ღილაკი გადაფურცვლისთვის.

ა. ქვემოთ ნაჩვენებია მარტივი კონფიგურაცია. პარამეტრები, რომელიც თქვენ დააკონფიგურეთ ყვითლად არის მონიშნული. დანარჩენი კონფიგურაცია არის IOS ოპერაციული სისტემის ნაგულისხმევი პარამეტრები.

S1# **show run**

Building configuration…

Current configuration : 2206 bytes

!

version 15.0

no service pad

service timestamps debug datetime msec

service timestamps log datetime msec

service password-encryption

!

hostname S1

!

boot-start-marker

boot-end-marker

!

enable secret 4 06YFDUHH61wAE/kLkDq9BGho1QM5EnRtoyr8cHAUg.2

!

no aaa new-model

system mtu routing 1500

!

!

```
no ip domain-lookup
!
<output omitted>
!
interface FastEthernet0/24
 switchport access vlan 99
!
interface GigabitEthernet0/1
 switchport access vlan 99
!
interface GigabitEthernet0/2
 switchport access vlan 99
!
interface Vlan1
 no ip address
 no ip route-cache
!
interface Vlan99
 ip address 192.168.1.2 255.255.255.0
 no ip route-cache
!
ip default-gateway 192.168.1.1
ip http server
ip http secure-server
!
banner motd ^C
Unauthorized access is strictly prohibited. ^C
!
```

```
line con 0
 password 7 104D000A0618
 logging synchronous
 login
line vty 0 4
 password 7 14141B180F0B
 login
line vty 5 15
 password 7 14141B180F0B
 login
!
end
S1#
```

ბ. შეამოწმეთ მართვადი VLAN 99-ის პარამეტრები.

```
S1# show inrterface vlan 99
Vlan99 is up, line protocol is up
  Hardware is EtherSVI, address is 0cd9.96e2.3d41 (bia 0cd9.96e2.3d41)
  Internet address is 192.168.1.2/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
       reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:06, output 00:08:45, output hang never
  Last clearing of “show interface” counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
```

5 minute output rate 0 bits/sec, 0 packets/sec

175 packets input, 22989 bytes, 0 no buffer

Received 0 broadcasts (0 IP multicast)

0 runts, 0 giants, 0 throttles

0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored

1 packets output, 64 bytes, 0 underruns

0 output errors, 0 interface resets

0 output buffer failures, 0 output buffers swapped out

რამდენია გამტარობა ამ ინტერფეისზე? _____

როგორია VLAN 99-ის მდგომარეობა? _____

როგორია ხაზის (line) პროტოკოლის მდგომარეობა? _____

**მეორე ეტაპი: სრული კავშირის შემოწმება Ping-ის გამოყენებით.**

ა. PC-A კომპიუტერის ბრძანებათა ველიდან, პირველ რიგში „დაპინგეთ“ თქვენი PC-A კომპიუტერის IP მისამართი.

C:\Users\User1> **ping 192.168.1.10**

ბ. PC-A კომპიუტერის ბრძანებათა ველიდან, „დაპინგეთ“ S1-ის SVI მართვადი მისამართი.

C:\Users\User1> **ping 192.168.1.2**

იმის გამო, რომ PC-A კომპიუტერს სჭირდება დაადგინოს S1-ის MAC მისამართი ARP-ს საშუალებით, პირველი პაკეტი შეიძლება შეყოვნდეს. თუ Ping ბრძანების შედეგები გაგრძელდა წარუმატებლად, გაასწორეთ მოწყობილობის ბაზისური კონფიგურაცია. თუ აუცილებელია, თქვენ უნდა შეამოწმოთ როგორც ფიზიკური შეერთება, ისე ლოგიკური დამისამართება.

**მესამე ეტაპი: S1-ის დაშორებული მართვის გამოცდა და შემოწმება**

თქვენ ახლა გამოიყენებთ Telnet-ს კომუტატორთან დაშორებული წვდომისთვის. მოცემულ ლაბორატორიულ დავალებაში, PC-A და S1 იმყოფებიან გვერდიგვერდ. წარმოდგენილ ქსელში, კომუტატორი შეიძლება იყოს ზედა სართულის სამონტაჟო კარადაში, როცა თქვენი სამართავი კომპიუტერი მოთავსებულია პირველ სართულზე. მოცემულ ეტაპზე თქვენ გამოიყენებთ Telnet-ს S1 კომუტატორთან დაშორებული წვდომისთვის, მისივე SVI მართვის მისამართის გამოყენებით. Telnet-ი არ არის დაცული პროტოკოლი; თუმცა თქვენ მას გამოიყენებთ დაშორებული წვდომის შესამოწმებლად. Telnet-ის გამოყენებით ყველა ინფორმაცია, პაროლების და ბრძანებების ჩათვლით, მთელი სესიის დროს გაიგზავნება ღია ტექსტით. მომდევნო ლაბორატორიულ სამუშაოში, ქსელურ მოწყობილობებთან დაშორებული წვდომისთვის თქვენ გამოიყენებთ SSH-ს.

**შენიშვნა:** თუ თქვენ იყენებთ Windows 7 ოპერაციულ სისტემას, ადმინისტრატორს შესაძლოა დასჭირდეს Telnet პროტოკოლის ჩართვა. Telnet კლიენტის დასაინსტალირებლად, გახსენით ბრძანებათა სტრიქონი (CMD) და აკრიფეთ **pkgmgr /iu:"TelnetClient"**. ქვემოთ ნაჩვენებია ინსტალაციის მაგალითი:

C:\Users\User1> **pkgmgr /iu:"TelnetClient"**

ა. CMD-ს ფანჯარა ისევ გახსნილია PC-A-ზე, გაუშვით Telnet ბრძანება S1-თან დასაკავშირებლად, SVI სამართავი მისამართის დახმარებით. პაროლი არის **cisco**.

C:\Users\User1> **telnet 192.168.1.2**

ბ. cisco პაროლის შეყვანის შემდეგ, თქვენ იქნებით მომხმარებლის EXEC რეჟიმის ბრძანებათა სტიქონზე. გადადით პრივილეგირებულ EXEC რეჟიმში.

გ. Telnet სესიის დასასრულებლად აკრიფეთ **Exit** ბრძანება.

## მეოთხე ეტაპი: კომუტატორის გაშვებული კონფიგურაციის ფაილის შენახვა.

შეინახეთ კონფიგურაცია.

S1# **copy running-config startup-config**

Destination filename [startup-config]? [**Enter**]

Building configuration…

[OK]

S1#

## ნაწილი №4: MAC მისამართების ცხრილის მართვა

მეოთხე ნაწილში თქვენ განსაზღვრავთ MAC მისამართს, რომელიც კომუტატორმა შეისწავლა, მომართავთ სტატიკურ MAC მისამართს კომუტატორის ერთ ინტერფეისზე და შემდეგ მოხსნით სტატიკურ MAC მისამართს ამ ინტერფეისიდან.

### პირველი ეტაპი: ჰოსტის MAC მისამართის ჩაწერა.

PC-A კომპიუტერის ბრძანებათა ველიდან გაუშვით ipconfig /all ბრძანება, რათა დაადგინოთ და ჩაიწეროთ პერსონალური კომპიუტერის ქსელის ადაპტერის მეორე დონის (ფიზიკური) მისამართები. _____

### მეორე ეტაპი: კომუტატორის მიერ შესწავლილი MAC მისამართების დადგენა.

გამოიტანეთ MAC მისამართების სია **show mac address-table** ბრძანების გამოყენებით.

S1# **show mac address-table**

რამდენი დინამიური მისამართია გამოტანილი? _____

გამოტანილი MAC მისამართების ჯამური რაოდენობა? _____

ემთხვევა თუ არა დინამიური MAC მისამართი PC-A-ს MAC მისამართს? _____

### მესამე ეტაპი: ჩამოშალეთ show mac-address-table პარამეტრები.

ა. გამოიტანეთ MAC მისამართის ცხრილის პარამეტრები.

S1# **show mac address-table ?**

რამდენი პარამეტრია ხელმისაწვდომი show mac address-table ბრძანებით? _____

ბ. გაუშვით show mac address-table dynamic ბრძანება, მხოლოდ იმ MAC მისამართების გამოსატანად, რომლებიც დინამიურად იქნა შესწავლილი.

S1# **show mac address-table dynamic**

რამდენი დინამიური მისამართია გამოტანილი? _____

გ. დაათვალიერეთ PC-A კომპიუტერის MAC მისამართი. MAC მისამართის ფორმატი ბრძანებისთვის არის xxxx.xxxx.xxxx.

S1# show mac address-table address <აქ იწერება PC-A-ს MAC მისამართი>

## მეოთხე ეტაპი: სტატიკური MAC მისამართის მომართვა

ა. MAC მისამართების ცხრილის გასუფთავება.

არსებული MAC მისამართების წასაშლელად გამოიყენეთ clear mac address-table dynamic ბრძანება პრივილეგირებული EXEC რეჟიმიდან.

S1# **clear mac address-table dynamic**

ბ. დარწმუნდით რომ MAC მისამართების ცხრილი წაშლილია.

S1# **show mac address-table**

რამდენი სტატიკური MAC მისამართია მოცემული ცხრილში? _____

რამდენი დინამიური MAC მისამართია მოცემული ცხრილში? _____

გ. ხელახლა შეამოწმეთ MAC ცხრილი.

სავარაუდოა რომ აპლიკაციამ, რომელიც გაშვებულია თქვენს პერსონალურ კომპიუტერზე, უკვე გააგზავნა ფრეიმი ქსელის ადაპტერიდან S1 კომუტატორისკენ. დაათვალიერეთ MAC მისამართების ცხრილი ხელახლა პრივილეგირებულ EXEC რეჟიმში, იმის სანახავად კომუტატორმა ხელახლა შეისწავლა თუ არა PC-A კომპიუტერის MAC მისამართი.

S1# **show mac address-table**

რამდენი დინამიური MAC მისამართია მოცემული ცხრილში? _____

რატომ არის ეს ბოლო ცვლილება ნაჩვენები? _____

თუ S1-ს ჯერ ხელახლა არ შეუსწავლია PC-A კომპიუტერის MAC მისამართი, PC-A-დან „დაპინგეთ“ კომუტატორის VLAN 99-ის IP მისამართი და შემდეგ გაიმეორეთ **show mac address-table** ბრძანება.

დ. სტატიკური MAC მისამართის მომართვა.

იმის მისათითებლად თუ ჰოსტის რომელ პორტებთანაა შესაძლებელი დაკავშირება, ერთი ვარიანტია ჰოსტის MAC მისამართის სტატიკური რუქის (Mapping) შექმნა პორტთან.

მომართეთ სტატიკური MAC მისამართი F0/6-ზე, იმ IP მისამართის გამოყენებით, რომელიც ჩაწერილ იქნა PC-A-სთვის ნაწილი №4-ის პირველ ეტაპზე. 0050.56BE.6C89 MAC მისამართი გამოყენებულია მხოლოდ მაგალითისთვის. თქვენ შეგიძლიათ გამოიყენოთ თქვენი PC-A კომპიუტერის MAC მისამართი, რომელიც განსხვავებულია მაგალითში მოცემული MAC მისამართისაგან.

S1 (config)# **mac address-table static 0500.56BE.6C89 vlan 99 interface fastethernet 0/6**

ე. MAC მისამართების ცხრილის მონაცემების შემოწმება.

S1 # **show mac address-table**

ჯამურად რამდენი MAC მისამართია მოცემული? _____

რამდენი სტატიკური მისამართია მოცემული?_____

ვ. წაშალეთ სტატიკური MAC ჩანაწერი. შედით გლობალური კონფიგურაციის რეჟიმში და გააუქმეთ ბრძანება **no**-ს ჩამატებით ბრძანებათა რიგის თავში.

**შენიშვნა:** 0050.56BE.6C89 MAC მისამართი გამოყენებულია მხოლოდ მაგალითისთვის. გამოიყენეთ MAC მისამართი თქვენი PC-A კომპიუტერისთვის.

S1(config)# **no mac address-table static 0050.56BE.6C89 vlan 99 interface fastethernet 0/6**

ზ. შეამოწმეთ წაიშალა თუ არა სტატიკური MAC მისამართი.

S1# **show mac address-table**

ჯამურად რამდენი MAC მისამართია მოცემული? _____

**ასახვა**

1. რატომ უნდა დააკონფიგუროთ კომუტატორის vty ხაზები?

_____

2. რატომ ცვლით ნაგულისხმევ VLAN 1-ს განსხვავებული VLAN-ის ნომრით?

_____

3. როგორ შეიძლება პაროლების ღია ტექსტით გაგზავნისაგან თავის აცილება?

_____

4. რატომ აკონფიგურებთ სტატიკურ MAC მისამართს ინტერფეისის პორტზე?

_____

**დანართი A: მარშრუტიზატორის და კომუტატორის ინიციალიზაცია და ხელახლა ჩატვირთვა**

**პირველი ეტაპი: მარშრუტიზატორის ინიციალიზაცია და ხელახლა ჩატვირთვა**

ა. კონსოლის კაბელით დაკავშირება მარშრუტიზატორთან და პრივილეგირებულ EXEC რეჟიმში შესვლა

Router> **enable**

Router#

ბ. შეიყვანეთ **erase startup-config** ბრძანება NVRAM-დან საწყისი კონფიგურაციის ფაილის წასაშლელად.

Router# **erase startup-config**

Erasing the nvram filesystem will remove all configuration files! Continue? [confirm]

[OK]

Erase of nvram: complete

Router#

გ. გაუშვით **Reload** ბრძანება მეხსიერებიდან ძველი კონფიგურაციის წასაშლელად. როდესაც შემოთავაზებული იქნება გაგრძელდეს თუ არა გადატვირთვა?, დააჭირეთ Enter-ს. (ნებისმიერ სხვა ღილაკზე დაჭერა აუქმებს ხელახლა ჩატვირთვის პროცესს).

Router# **reload**

Proceed with reload? [confirm]

*Nov 29 18:28:09.923: %SYS-5-RELOAD: Reload requested by console. Reload Reason:

Reload Command.

**შენიშვნა:** შეიძლება მიიღოთ შემახსენებელი შეტყობინება, რომელიც გვთავაზობს გაშვებული კონფიგურაციის შენახვას, მარშრუტიზატორის ხელახლა ჩატვირთვის წინ. უპასუხეთ **no**-ს აკრეფით და დააჭირეთ Enter-ს.

System configuration has been modified. Save? [yes/no] : **no**

დ. მარშტურიზატორის გადატვირთვის შემდეგ, შემოთავაზებული იქნება საბაზისო კონფიგურაციის დიალოგურ ფანჯარაში შესვლა. შეიყვანეთ **no** და დააჭირეთ Enter ღილაკს.

Would you like to enter the initial configuration dialog? [yes/no] : **no**

ე. სხვა შემოთავაზება გვატყობინებს ავტომატური ინსტალაციის დასრულებას. უპასუხეთ **yes**-ის აკრეფით და დააჭირეთ Enter-ს.

Would you like terminate autoinstall? [yes] : **yes**

## მეორე ეტაპი: კომუტატორის ინიციალიზაცია და ხელახლა ჩატვირთვა

ა. კონსოლის კაბელით დაკავშირება კომუტატორთან და პრივილეგირებულ EXEC რეჟიმში შესვლა

Switch> enable

Switch#

ბ. გამოიყენეთ **Show flash** ბრძანება იმის დასადგენად შექმნილია თუ არა კომუტატორზე რაიმე VLAN-ი.

Switch# **show flash**

Directory of flash:/

| | | | | |
|---|---|---|---|---|
| 2 | -rwx | 1919 | Mar 1 1993 00:06:33 +00:00 | private-config.text |
| 3 | -rwx | 1632 | Mar 1 1993 00:06:33 +00:00 | config.text |
| 4 | -rwx | 13336 | Mar 1 1993 00:06:33 +00:00 | multiple-fs |
| 5 | -rwx | 11607161 | Mar 1 1993 02:37:06 +00:00 | c2960-lanbasek9-mz.150-2.SE.bin |
| 6 | -rwx | 616 | Mar 1 1993 00:07:13 +00:00 | vlan.dat |

32514048 bytes total (20886528 bytes free)

Switch#

გ. თუ მოიძებნა წაშალეთ **vlan.dat** ფაილი

Switch# **delete vlan.dat**

Delete filename [vlan.dat]?

დ. სისტემა შეგახსენებთ სახელის შემოწმებას. თუ თქვენ სახელი სწორად გაქვთ შეყვანილი დააჭირეთ Enter-ს; წინააღმდეგ შემთხვევაში თქვენ შეგიძლიათ ფაილის სახელის შეცვლა.

ე. სისტემა შეგახსენებთ, დარწმუნებული ხართ თუ არა წაიშალოს მოცემული ფაილი. თანხმობისთვის დააჭირეთ Enter ღილაკს.

Delete flash:/vlan.dat? [confirm]

Switch#

ვ. გამოიყენეთ **erase startup-config** ბრძანება, NVRAM-დან კონფიგურაციის ფაილის წასაშლელად. სისტემა შეგეკითხებათ, გსურთ თუ არა კონფიგურაციის ფაილის წაშლა. თანხმობისთვის დააჭირეთ Enter ღილაკს.

Switch# **erase startup-config**

Erasing the nvram filesystem will remove all configuration files! Continue? [confirm]

[OK]

Erase of nvram: complete

Switch#

ზ. გადატვირთეთ კომუტატორი მეხსიერებიდან ნებისმიერი ძველი კონფიგურაციის ინფორმაციის წასაშლელად. სისტემა შეგეკითხებათ გსურთ თუ არა კომუტატორის გადატვირთვა. პროცესის დასაწყებად დააჭირეთ Enter ღილაკს.

Switch# **reload**

Proceed with reload? [confirm]

შენიშვნა: შეიძლება მიიღოთ გამაფრთხილებელი შეტყობინება, რომელიც გვთავაზობს გაშვებული კონფიგურაციის შენახვას, კომუტატორის ხელახლა ჩატვირთვის წინ. უპასუხეთ no-ს აკრეფით და დააჭირეთ Enter-ს.

System configuration has been modified. Save? [yes/no] : **no**

თ. კომუტატორის გადატვირთვის შემდეგ, შემოთავაზებული იქნება საბაზისო კონფიგურაციის დიალოგურ ფანჯარაში შესვლა. შეიყვანეთ no და დააჭირეთ Enter ღილაკს.

Would you like to enter the initial configuration dialog? [yes/no] : **no**

Switch>

### *3.2.5 დაშორებული წვდომის უსაფრთხოება*

#### *3.2.5.1 SSH ფუნქცია*

Secure Shell (SSH) არის პროტოკოლი, რომელიც უზრუნველყოფს დაშორებული მოწყობილობის დაცულ (შიფრირებულ) მართვად კავშირს. მართვადი კავშირისთვის SSH-ი ცვლის Telnet-ს. Telnet არის ძველი პროტოკოლი, რომელიც იყენებს დაუცველ, ღია ტექსტით გადაცემას შესვლის აუთენტიფიკაციის (მომხმარებლის სახელი და პაროლი) და ურთიერთმოქმედ მოწყობილობებს შორის მონაცემების გადაცემის დროს. SSH იძლევა დაშორებული კავშირის უსაფრთხოებას ძლიერი შიფრირების უზრუნველყოფით, როდესაც მოწყობილობა არის ავტორიზებული (მომხმარებლის სახელი და პაროლი) და ასევე დაკავშირებულ მოწყობილობებს შორის მონაცემთა გადაცემის დროს. SSH-ს მინიჭებული აქვს TCP-ის 22 პორტი ნომერი, ხოლო Telnet-ს - TCP 23 პორტის ნომერი.

Catalyst 2960 კომუტატორზე SSH-ის ჩასართავად, კომუტატორი უნდა იყენებდეს IOS სისტემის იმ ვერსიას, რომელიც შეიცავს დაშიფვრის ფუნქციებს და შესაძლებლობებს. იმის სანახავად თუ რომელი IOS ოპერაციული სისტემაა გაშვებული, გამოიყენეთ show version ბრძანება კომუტატორზე. ასევე ამ ბრძანებით შეგვიძლია ვნახოთ ფაილის სახელი, რომელიც მოიცავს „K9”, შიფრაციის მხარდაჭერის ფუნქციებისა და შესაძლებლობების კომბინაციას.

```
S1> show version
Cisco IOS Software, C2960 Software (C2960-LANBASEK9-M),
Version 15.0(2)SE, RELEASE SOFTWARE (fc1)

<output omitted>
```

*სურ.3.2.5.1. 1*

#### *3.2.5.2 SSH-ის კონფიგურაცია*

SSH-ის კონფიგურირებამდე, კომუტატორ უნდა იყოს მინიმალურად კონფიგურირებული უნიკალური სახელით და სწორი ქსელის კავშირის პარამეტრებით.

**პირველი ეტაპი. SSH მხარდაჭერის შემოწმება.**

გამოიყენეთ show ip ssh ბრძანება, კომუტატორზე SSH მხარდაჭერის შესამოწმებლად. თუ კომუტატორზე არ არის გაშვებული ის IOS ოპერაციული სისტემა, რომელიც მხარს უჭერს შიფრაციის ფუნქციებს, მოცემული ბრძანება სისტემისთვის იქნება გაუგებარი.

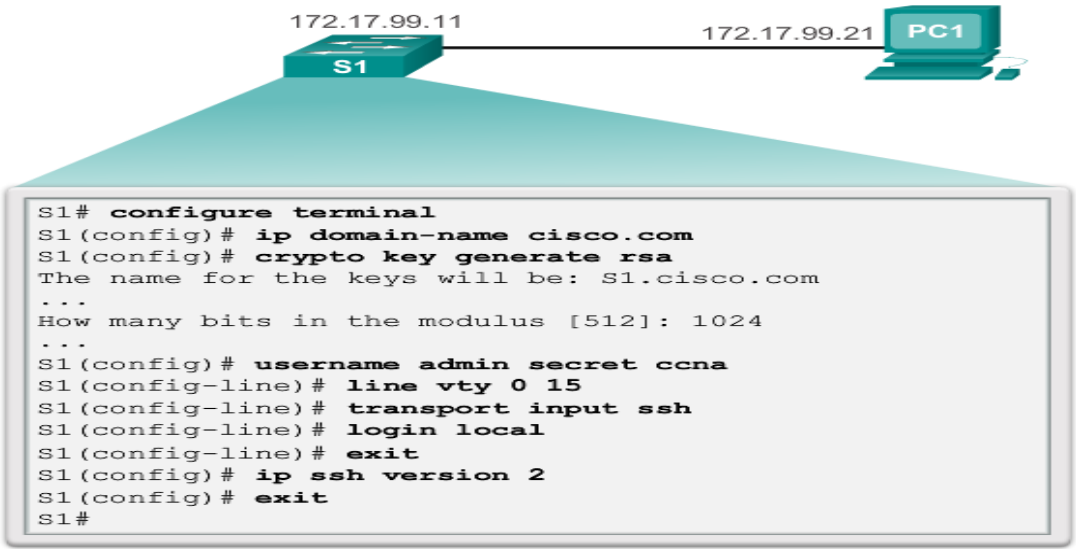
**მეორე ეტაპი. IP დომეინის კონფიგურაცია.**

დააკონფიგურეთ ქსელის IP დომეინის სახელი **ip domain-name** *domain-name* ბრძანების გამოყენებით, საერთო კონფიგურაციის რეჟიმში. 3.2.5.2,1 სურათზე *domain-name* მნიშვნელობა არის cisco.com.

**მესამე ეტაპი. RSA წყვილი გასაღების გენერაცია**

არა ყველა ვერსიის IOS ოპერაციულ სისტემაში, ნაგულისხმევად SSHv2-ში, და SSHv1-ში არის უსაფრთხოების ნაცნობი ხარვეზები. SSH ვერსია 2-ის კონფიგურაციისთვის გაუშვით ip ssh version 2 საერთო კონფიგურაციის რეჟიმის ბრძანება. RSA წყვილი გასაღების გენერაცია ავტომატურად რთავს SSH-ს. გამოიყენეთ crypto key generate rsa საერთო კონფიგურაციის რეჟიმის ბრძანება კომუტატორზე SSH სერვერის ჩასართავად და RSA წყვილი გასაღების გენერაციისთვის. როდესაც ხდება RSA გასაღებების გენერაცია, ადმინისტრატორმა უნდა მიუთითოს მოდულის სიგრძე. Cisco იძლევა რეკომენდაციას, რომ მინიმალური მოდულის სიგრძე იყოს 1024 ბიტი (იხილეთ მარტივი კონფიგურაცია 3.2.5.1 სურათზე). რაც მეტია მოდულის სიგრძე, მივიღებთ მეტ უსაფრთხოებას, მაგრამ მის გენერაციას და გამოყენებას მიაქვს დიდი დრო.

**შენიშვნა:** RSA წყვილი გასაღების წასაშლელად, გამოიყენეთ crypto key zeroize rsa საერთო კონფიგურაციის რეჟიმის ბრძანება. RSA წყვილი გასაღების წაშლის შემდეგ ავტომატურად ითიშება SSH სერვერი.




*სურ.3.2.5.2. 1 SSH-ის კონფიგურაცია დაშორებული მართვისთვის*

**მეოთხე ეტაპი. მომხმარებლის აუთენტიფიკაციის კონფიგურაცია**

SSH სერვერს შეუძლია მომხმარებლების აუთენტიფიკაცია ლოკალურად ან აუთენტიფიკაციის სერვერის გამოყენებით. ლოკალური აუთენტიფიკაციის მეთოდის გამოყენებისთვის შექმენით მომხმარებლის სახელისა და პაროლის წყვილი **username** *username* **secret** *password* საერთო კონფიგურაციის რეჟიმის ბრძანების გამოყენებით. მაგალითზე admin მომხმარებელს მინიჭებული აქვს ccna პაროლი.

**მეხუთე ეტაპი. vty ხაზების კონფიგურაცია**

ჩართეთ SSH პროტოკოლი vty ხაზებზე transport input ssh ხაზის კონფიგურაციის რეჟიმის ბრძანების გამოყენებით. Catalyst 2960 კომუტატორს აქვს vty ხაზების ზღვარი 0-დან 15-მდე. მოცემული კონფიგურაცია კრძალავს არა-SSH (Telnet-ის ჩათვლით) შეერთებებს და შეზღუდვას უწესებს კომუტატორს, რომ დაეთანხმოს მხოლოდ SSH კავშირებს. გამოიყენეთ line vty საერთო კონფიგურაციის რეჟიმის ბრძანება და შემდეგ login local ხაზის კონფიგურაციის რეჟიმის ბრძანება, რათა მოთხოვნილ იქნას ლოკალური აუთენტიფიკაცია SSH კავშირებისთვის, ლოკალურ მომხმარებელთა მონაცემთა ბაზიდან.

**მეექვსე ეტაპი. SSH ვერსია 2-ის ჩართვა**

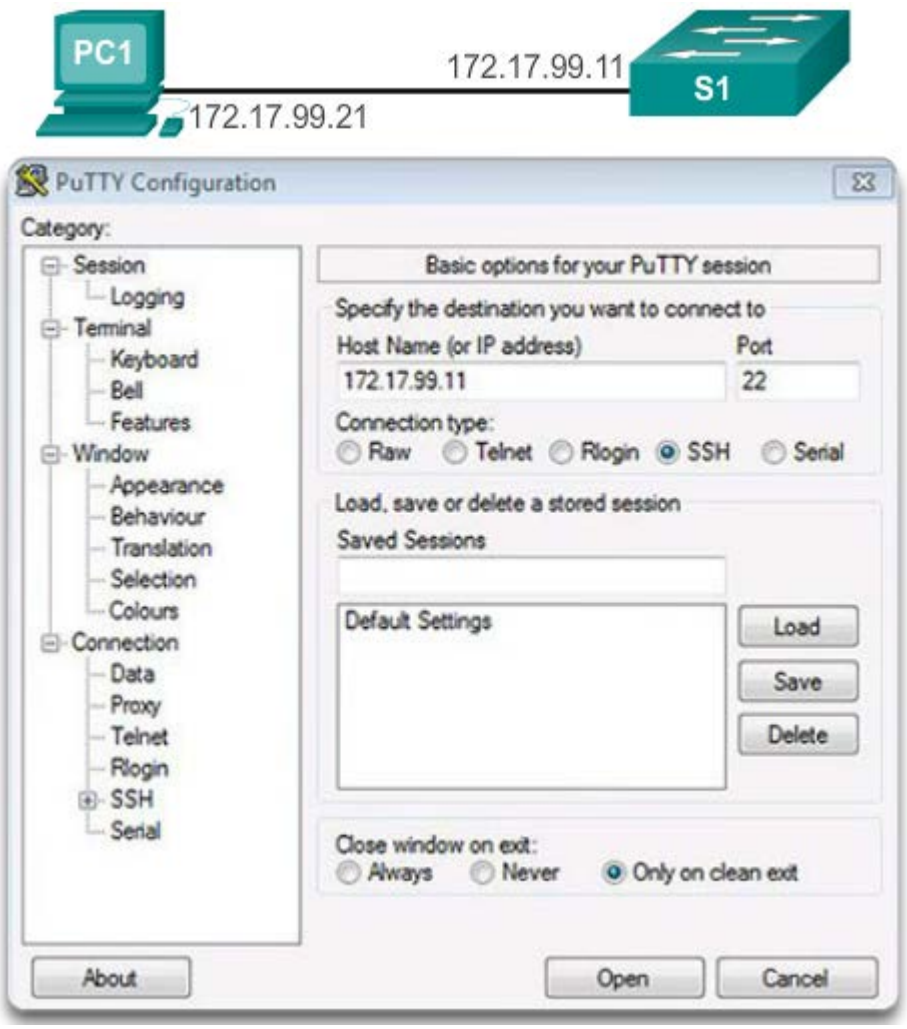
ნაგულისხმევად SSH მხარს უჭერს ორივე ვარიანტს: ვერსია 1 და ვერსია 2. როცა მხარდაჭერილია ორივე ვერსია, ეს ნაჩვენებია show ip ssh ბრძანების გაშვების შედეგზე, როგორც მხარდაჭერილი 1.99 ვერსია. ვერსია 1 ცნობილია თავისი ხარვეზებით, ამიტომ რეკომენდებულია მხოლოდ ვერსია 2-ის ჩართვა. ჩართეთ SSH ვერსია 2 ip ssh version 2 საერთო კონფიგურაციის ბრძანების გამოყენებით.

*3.2.5.3 SSH-ის შემოწმება*

კომპიუტერზე, SSH კლიენტი, PuTTY-ს ჩათვლით, გამოიყენება SSH სერვერთან დასაკავშირებლად. მაგალითისთვის სურათებზე, მომართულია შემდეგი:

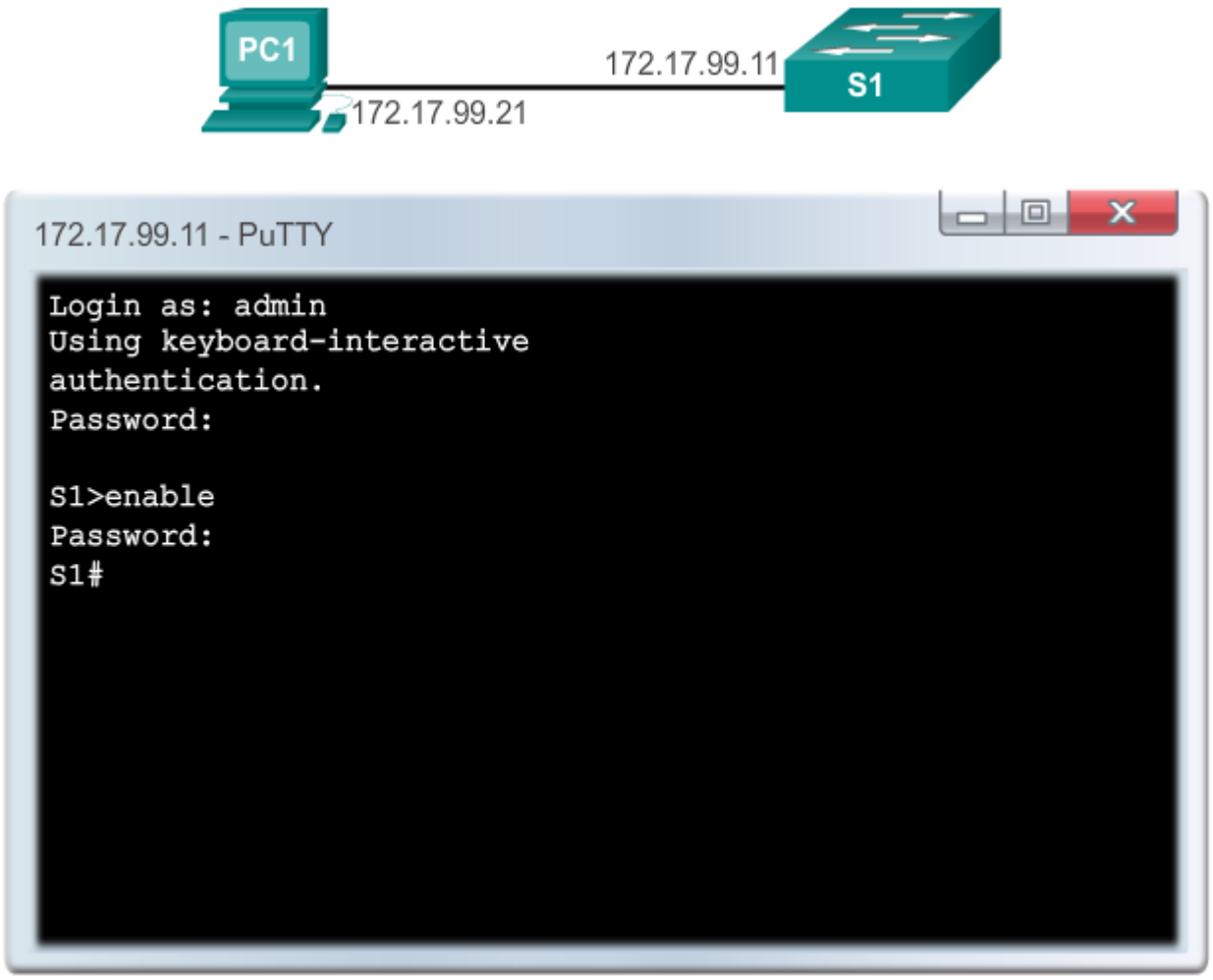
- SSH ჩართულია S1 კომუტატორზე
- VLAN 99 (SVI) ინტერფეისი 172.17.99.11 IP მისამართით S1 კომუტატორზე
- PC1 IP მისამართით 172.17.99.21

3.2.5.3.1 სურათზე PC1 კომპიუტერი ინიციალიზაციას უკეთებს SSH შეერთებას, S1 კომუტატორის SVI VLAN IP მისამართთან.



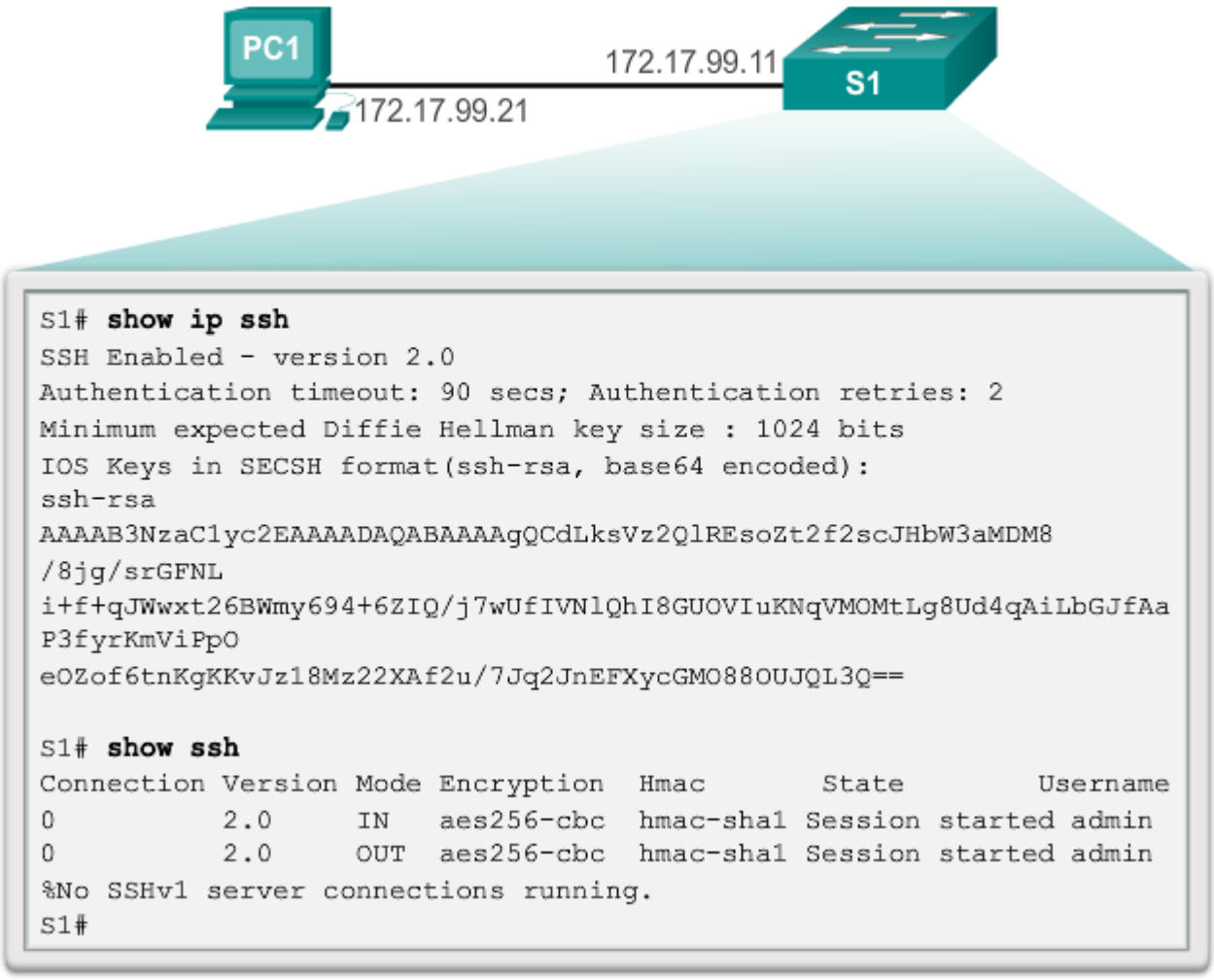

*სურ.3.2.5.3. 1 PuTTY SSH კლიენტის შეერთების პარამეტრების კონფიგურაცია*

3.2.5.3.2 სურათზე მომხმარებელმა უნდა შეიყვანოს მომხმარებლის სახელი და პაროლი. გამოიყენეთ წინა მაგალითში მოცემული კონფიგურაცია: მომხმარებლის სახელი admin და პაროლი ccna. სწორი კომბინაციის შეყვანის შემდეგ, მომხმარებელი SSH-ით დაკავშირდება Catalyst 2960 კომუტატორის CLI ინტერფეისთან.




*სურ.3.2.5.3. 2 დაშორებული მართვის SSH შეერთება*

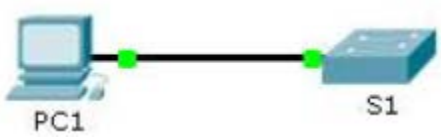
SSH-ის ვერსიის და კონფიგურაციის მონაცემების სანახავად იმ მოწყობილობაზე, რომელიც დავაკონფიგურეთ როგორც SSH სერვერი, გამოიყენეთ show ip ssh ბრძანება. მაგალითზე ჩართულია SSH ვერსია 2. მოწყობილობასთან SSH კავშირის შესამოწმებლად, გამოიყენეთ show ssh ბრძანება (იხ. სურ. 3.2.5.3.3):




*სურ.3.2.5.3. 3 SSH-ის მდგომარეობისა და პარამეტრების შემოწმება*

პრაქტიკული სამუშაო - SSH-ის კონფიგურაცია

**ტოპოლოგია**




**მისამართების ცხრილი:**

| მოწყობილობა | ინტერფეისი | IP მისამართი | ქვექსელის ნიღაბი |
|---|---|---|---|
| S1 | VLAN 1 | 10.10.10.2 | 255.255.255.0 |
| PC1 | NIC | 10.10.10.10 | 255.255.255.0 |

**შესასრულებელი ამოცანები:**

**ნაწილი №1: პაროლების უსაფრთხოება**

**ნაწილი №2: შეერთებების შიფრაცია**

**ნაწილი №3: SSH დანერგვის შემოწმება**

**ზოგადი ინფორმაცია**

**SSH**-მა უნდა შეცვალოს **Telnet**-ი შეერთებების მართვისთვის. **Telnet**-ი იყენებს დაუცველ ღია ტექსტით კომუნიკაციას. **SSH** იძლევა დაცვას დაშორებული შეერთებებისთვის, მოწყობილობებს შორის ყველა გადასაცემი ინფორმაციის რთული შიფრაციის უზრნველყოფით. მოცემულ დავალებაში თქვენ დაიცავთ დაშორებულ კომუტატორს პაროლის შიფრაციით და **SHH**-ით.

**ნაწილი №1: პაროლების უსაფრთხოება**

ა. **PC1**-ის ბრძანებათა სტრიქონის გამოყენებით, დაკაშირდით **Telnet**-ით **S1** კომუტატორთან. მომხმარებლის **EXEC** და პრივილეგირებული **EXEC** პაროლი არის **cisco**.

ბ. შეინახეთ მიმდინარე კონფიგურაცია, რადგან ნებისმიერი შეცდომა, რომლებიც თქვენ შეიძლება დაუშვათ შეიძლება გაუქმებულ იქნას **S1** კომუტატორის კვების გადამრთველის საშუალებით.

გ. დაათვალიერეთ მიმდინარე კონფიგურაცია და შენიშნავთ რომ პაროლები მოცემულია ღია ტექსტით. დაწერეთ ბრძანება, რომელიც დაშიფრავს ღია ტექსტის პაროლებს.

---

დ. დარწმუნდით, რომ პაროლები დაშიფრულია

**ნაწილი №2: კომუნიკაციების შიფრაცია**

**პირველი ეტაპი: IP დომეინის სახელის მომართვა და დაცული გასაღებების შექმენა.**

როგორც წესი **Telnet**-ის გამოყენება არ არის უსაფრთხო, იმიტომ რომ მონაცემები გადაიცემიან ღია ტექსტით. აქედან გამომდინარე გამოიყენეთ **SSH** როცა ის ხელმისაწვდომია.

ა. მომართეთ დომეინის სახელი ისე რომ იყოს **netacad.pka**

---

ბ. მონაცემების დასაშიფრად საჭიროა დაცული გასაღებები. შექმენით **RSA** გასაღებები **1024** ბიტიანი სიგრძის გამოყენებით

---

**მეორე ეტაპი: SSH მომხმარებლის შექმნა და VTY ხაზების ხელახალი კონფიგურაცია მხოლოდ SSH-ის წვდომისთვის.**

ა. **Administrator** მომხმარებელის შექმნა **cisco** უსაფრთხო პაროლთან ერთად.

---

ბ. **VTY** ხაზების დაკონფიგურება ლოკალური მომხმარებლის სახელების მონაცემთა ბაზის შესამოწმებლად, სააღრიცხვო მონაცემების შეყვანისა და მხოლოდ **SSH**-ის დაშვებისთვის, დაშორებული წვდომის უზრუნველყოფისთვის. გააუქმეთ არსებული **VTA** ხაზის პაროლი.

---

---

---

**ნაწილი №3: SSH რეალიზაციის შემოწმება**

ა. დახურეთ **Telnet** სესია და სცადეთ შესვლა ისევ **Telnet**-ის გამოყენებით. მცდელობა უნდა დასრულდეს წარუმატებლად

ბ. სცადეთ შესვლა **SHH**-ის გამოყენებით. აკრიფეთ **SSH** და დააჭირეთ **Enter** ღილაკს ყოველგვარიპარამეტრების გარეშე ბრძანებების გამოყენების ინსტრუქციების გამოსატანად. შეგახსენებთ: **-l** პარამეტრი არის სიმბოლო **„L”** და არა ციფრი **1**.

გ. წარმატებული შესვლის შემდეგ გადადით პრივილეგირებულ **EXEC** რეჟიმში და შეინახეთ კონფიგურაცია. თუ ვერ მოხერხდა **S1** კომუტატორთან წარმატებული წვდომა, გადატვირთეთ კვება და დაიწყეთ ისევ ნაწილი №1-დან.

## 3.3. Vlan-ების და Trunk-ების კონფიგურაცია

შესავალი

ქსელის წარმადობა მნიშვნელოვანი ფაქტორია ორგანიზაციის პროდუქტიულობაში. ერთ-ერთი ტექნოლოგია, რომელიც გამოიყენება ქსელის წარმადობის გასაზრდელად არის დიდი ფართომაუწყებლობითი დომენების დაყოფა პატარა ნაწილებად. როგორც წესი, მარშრუტიზატორები ბლოკავენ ფართომაუწყებლობით ტრაფიკს ინტერფეისზე. ამასთან, მარშრუტიზატორებს აქვთ შეზღუდული რაოდენობის ლოკალური ქსელის (LAN) ინტერფეისები. მარშრუტიზატორის მთავარი როლი არის ინფორმაციის გადატანა ქსელებს შორის, და არ უზრუნველყოფს საბოლოო მოწყობილობების ქსელთან წვდომის საშუალებას.

ლოკალურ ქსელში წვდომის უზრუნველყოფა, როგორც წესი ეკუთვნის წვდომის დონის კომუტატორს. მსგავსად მესამე დონის მოწყობილობისა, ვირტუალური ლოკალური ქსელის (VLAN) შექმნა შესაძლებელია მეორე დონის კომუტატორზეც, ფართომაუწყებლობითი დომენების ზომების შესამცირებლად. ვირტუალური ლოკალური ქსელები როგორც წესი ჩართულნი არიან ქსელის დაგეგმვაში, რაც აადვილებს ორგანიზაციის მიზნების განხორციელებას ქსელთან მიმართებაში. ვირტუალური ლოკალური ქსელები ძირითადად გამოიყენება კომუტირებულ ლოკალურ ქსელებში, მაგრამ ვირტუალური ქსელების თანამედროვე რეალიზაცია საშუალებას აძლევს მათ იმუშაონ MAN და WAN ქსელებშიც.

### *3.3.1. ვირტუალური ლოკალური ქსელების (VLAN) მიმოხილვა*

მომხმარებლის პროდუქტიულობა და ქსელთან ადაპტაცია არის აუცილებელი ბიზნესის ზრდისა და წარმატებისთვის. VLAN-ები აადვილებს ქსელის შექმნას, მთელი ორგანიზაციის მხარდაჭერისთვის. ვირტუალური ლოკალური ქსელების გამოყენების მთავარი უპირატესობებია:

- უსაფრთხოება - ჯგუფებს, რომელთაც აქვთ მნიშვნელოვანი ინფორმაცია, გამოყოფილნი არიან დანარჩენი ქსელის ნაწილისაგან, რაც ამცირებს

კონფიდენციალური ინფორმაციის დაზიანების ალბათობას. როგორც 3.3.3 სურათზეა ნაჩვენები, ფაკულტეტის კომპიუტერები არიან VLAN 10-ში და სრულად გამოყოფილნი არიან სტუდენტის და სტუმრის მონაცემთა ტრაფიკისაგან.

- ხარჯების შემცირება -
- უკეთესი წარმადობა -
- ფართომაუწყებლობითი დომეინების შემცირება -
- გაუმჯობესებული IT თანამშრომლების ეფექტურობა -
- მარტივი დაგეგმვა და აპლიკაციების მართვა

### *3.3.2. VLAN-ების ტიპები*

არსებობს განსხვავებული ტიპის VLAN-ების მთელი რიგი, რომელიც გამოყენებულია თანამედროვე ქსელებში. ზოგიერთი VLAN-ის ტიპი განსაზღვრულია ტრაფიკის კლასებით. სხვა ტიპის VLAN-ები განისაზღვრებიან კონკრეტული ფუნქციებით, რომლითაც ისინი ემსახურებიან ქსელს.

**მონაცემთა VLAN**

მონაცემთა VLAN არის VLAN, რომელიც დაკონფიგურებულია მომხმარებლის მიერ დაგენერირებული ტრაფიკის გადასაგზავნად. VLAN ქსელს, რომელსაც გადააქვს ხმა ან მართავს ტრაფიკს, არ იქნება მონაცემთა VLAN-ის ნაწილი. ეს არის ჩვეულებრივი პრაქტიკა, რათა გაყოფილ იქნას ხმა და მართვის ტრაფიკი მონაცემთა ტრაფიკისაგან. მონაცემთა VLAN-ს ზოგჯერ მოიხსენიებენ როგორც მომხმარებლის VLAN-ს. მონაცემთა VLAN გამოიყენება ქსელის დასაყოფად მომხმარებლების ჯგუფებად ან მოწყობილობებად.

**ნაგულისხმევი VLAN**

კომუტატორის ყველა პორტი ხდება ნაგულისხმევი VLAN-ის ნაწილი მას შემდეგ რაც კომუტატორის საწყისი ჩატვირთვა გაუშვებს ნაგულისხმევ კონფიგურაციას. კომუტატორის პორტები, რომლებიც მონაწილეობენ ნაგულისხმევ VLAN-ში, იგივე ფართომაუწყებლობითი დომეინის ნაწილი არიან. ეს საშუალებას აძლევს ნებისმიერ

მოწყობილობას, რომლებიც შეერთებულნი არიან კომუტატორის ნებისმიერ პორტთან, დააკავშიროს სხვა მოწყობილობები სხვა კომუტატორის პორტებთან. Cisco კომუტატორებისთვის ნაგულისხმევი VLAN არის VLAN 1. 3.3.4.1 სურათზე ნაგულისხმევი კონფიგურაციით მართვად კომუტატორზე გაშვებულია show vlan brief ბრძანება. აღსანიშნავია, რომ ნაგულისხმევად ყველა პორტი მიკუთვნებულია VLAN 1-ზე.

```
Switch# show vlan brief

VLAN Name                Status    Ports
---- ------------------- --------- -----------------------
1    default             active    Fa0/1,  Fa0/2,  Fa0/3,  Fa0/4
                                   Fa0/5,  Fa0/6,  Fa0/7,  Fa0/8
                                   Fa0/9,  Fa0/10, Fa0/11, Fa0/12
                                   Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                   Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                   Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                   Gi0/1,  Gi0/2
1002 fddi-default        act/unsup
1003 token-ring-default  act/unsup
1004 fddinet-default     act/unsup
1005 trnet-default       act/unsup
```

- All p ნაგულისხმევად ყველა პორტი ეკუთვნის VLAN 1-ს, მონაცემთა გადასაგზავნად
- ნაგულისხმევად ადგილობრივი VLAN-ი არის VLAN 1
- ნაგულისხმევად მართვადი VLAN-ი არის VLAN 1
- გამორიცხულია VLAN 1-ის სახელის შეცვლა ან წაშლა

*სურ.3.3.2. 1 VLAN 1*

VLAN 1-ს აქვს ნებისმიერი VLAN-ის ყველა მახასიათებელი, უბრალოდ გამორიცხულია მისი წაშლა ან სახელის შეცვლა. ნაგულისხმევად ყველა მეორე დონის ტრაფიკის კონტროლი არის დაკავშირებული VLAN 1-ით.

**ადგილობრივი VLAN**

ადგილობრივი VLAN არის მიკუთვნებული 802.1Q trunk პორტთან. Trunk პორტები არის ბმულები კომუტატორებს შორის, რომელიც მხარს უჭერს ტრაფიკის გადაცემას ერთზე მეტ დაკავშირებულ VLAN-თან. 802.1Q trunk პორტი მხარს უჭერს ტრაფიკს, რომელიც

მიღებულია მრავალი VLAN-დან (დანიშნული ტრაფიკი), ასევე ტრაფიკს, რომელიც არ მოდის VLAN-დან (დაუნიშნავი ტრაფიკი). 802.1Q trunk პორტი განათავსებს დაუნიშნავ ტრაფიკს ადგილობრივ VLAN-ზე, რომელიც ნაგულისხმევად არის VLAN 1.

საუკეთესო პრაქტიკაა, რომ დაკონფიგურდეს ადგილობრივი VLAN-ი, როგორც გამოუყენებელი VLAN-ი, რომელიც განსხვავდება VLAN 1-სა და სხვა VLAN-ებისგან. ფაქტობრივად ის არ არის გამოუყენებელი ფიქსირებული VLAN-ის გამოსაყოფად, რათა შეასრულოს ადგილობრივი VLAN-ის როლი ყველა trunk პორტისთვის, კომუტირებულ დომეინში.

### მართვადი VLAN

მართვადი VLAN არის ნებისმიერი VLAN, რომელიც დაკონფიგურებულია კომუტატორის მართვის შესაძლებლობებზე წვდომისთვის. ნაგულისხმევად VLAN 1 არის მართვადი VLAN-ი. მართვადი VLAN-ის შესაქმნელად, კომუტატორის ვირტუალური ინტერფეისი (SVI), რომლის VLAN-საც მინიჭებული აქვს IP მისამართი და ქვექსელის ნიღაბი, საშუალებას აძლევს კომუტატორს იყოს მართული HTTP, Telnet, SSH ან SNMP-ს საშუალებით. იმის გამო, რომ Cisco კომუტატორების სტანდარტულ კონფიგურაციაში VLAN 1 არის ნაგულისხმევი VLAN-ი, VLAN 1 შეიძლება იყოს ცუდი არჩევანი მათვადი VLAN-ისთვის.

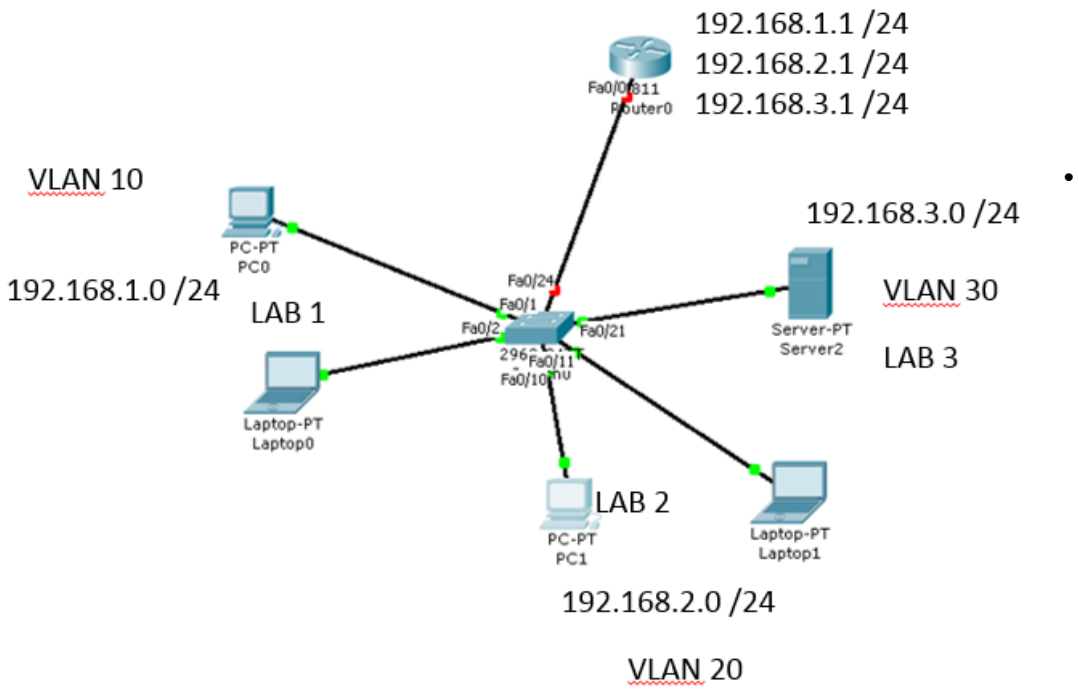
ადრე, მართვადი VLAN-ი 2960 კომუტატორისთვის იყო მხოლოდ ერთი აქტიური SVI. Catalyst 2960 სერიის კომუტატორების Cisco IOS-ის 15.x ვერსიებში, შესაძლებელია გვქონდეს ერთზე მეტი აქტიური SVI. Cisco IOS 15.x-ით, კონკრეტული აქტიური SVI, რომელიც განკუთვნილია დაშორებული მართვისთვის, უნდა იყოს დოკუმენტირებული. სანამ თეორიულად კომუტატორს შეუძლია ჰქონდეს ერთზე მეტი მართვადი VLAN, ერთზე მეტის არსებობა ზრდის ქსელური თავდასხმების რისკს.

3.3.2.1 სურათზე ყველა პორტი მინიჭებულია ნაგულისხმევ VLAN 1-ზე. ადგილობრივი VLAN-ი აშკარად არაა მინიჭებული და სხვა VLAN-ებიც არაა აქტიური. მაშასადამე ქსელი შექმნილია ადგილობრივი VLAN-ით, ისე როგორც მართვადი VLAN-ი. ეს განიხილება, როგორც უსაფრთხოები რისკი.

*პრაქტიკული სავარჯიშო: Vlan-ის შექმნა*

**ამოცანის მიზანი:** ერთ კომუტატორზე(Switch) მიერთებული კვანძებისთვის(Host) სხვადასხვა ვირტუალური ქსელის(Vlan) შექმნა; მარშრუტიზატორზე (Router) ფიზიკური ინტერფეისის ქვეინტერფეისების შექმნა და Vlan-ებთან დაკავშირება

ქსელის ფიზიკური და ლოგიკური მოდელი (იხ. სურ)




**ნაბიჯი 1**. VLAN-ის შექმნა და სახელის მინიჭება

- Switch>enable
- Switch#config t
- Switch(config)#vlan 10
- Switch(config-vlan)#name **lab1**
- Switch(config-vlan)#exit
- Switch(config)#vlan 20
- Switch(config-vlan)#name **lab2**
- Switch(config-vlan)#exit
- Switch(config)#vlan 30

- Switch(config-vlan)#name **lab3**
- Switch(config-vlan)#end

**მოცემული ბრძანებების კონფიგურირებით შექმნილია 3 vlan ვირტუალური ქსელი (VLAN 10 ; VLAN 20 და VLAN 30) და მათ მინიჭებული აქვს შესაბამისი სახელები : LAB1; LAB2 და LAB3**

მივიღებთ ამგვარ სურათს - ე.ი. VLAN-ები შექმნილია, მაგრამ მასში არ არის გაწევრიანებული ინტერფეისები

**Switch#show vlan**

| VLAN | Name | Status | Ports |
|---|---|---|---|
| 1 | default | active | Fa0/1, Fa0/2, Fa0/3, Fa0/4<br>Fa0/5, Fa0/6, Fa0/7, Fa0/8<br>Fa0/9, Fa0/10, Fa0/11, Fa0/12<br>Fa0/13, Fa0/14, Fa0/15, Fa0/16<br>Fa0/17, Fa0/18, Fa0/19, Fa0/20<br>Fa0/21, Fa0/22, Fa0/23, Fa0/24<br>Gig1/1, Gig1/2 |
| **10** | **lab1** | **active** | |
| **20** | **lab2** | **active** | |
| **30** | **lab3** | **active** | |
| 1002 | fddi-default | act/unsup | |
| 1003 | token-ring-default | act/unsup | |
| 1004 | fddinet-default | act/unsup | |
| 1005 | trnet-default | act/unsup | |

**ნაბიჯი 2.** ინტერფეისების VLAN-ში გაწევრიანება

Switch(config)#interface range fastEthernet 0/1-10

Switch(config-if-range)#switchport mode access

Switch(config-if-range)#switchport access vlan 10

**fastEthernet 0/1- დან fastEthernet 0/10-ის ჩათვლით გაწევრიანდა VLAN 10-ში**

Switch(config)#interface range fastEthernet 0/11-20

Switch(config-if-range)#switchport mode access

```
Switch(config-if-range)#switchport access vlan 20
Switch(config-if-range)#exit

Switch(config)#interface range fastEthernet 0/21-23
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 30
Switch(config-if-range)#exit
Switch(config)#
```

მივიღებთ შემდეგ სურათს:

```
Switch#show vlan

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/24, Gig1/1, Gig1/2
10   lab1                              active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                 Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                 Fa0/9, Fa0/10
20   lab2                              active    Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                                 Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                                 Fa0/19, Fa0/20
30   lab3                              active    Fa0/21, Fa0/22, Fa0/23
1002 fddi-default                      act/unsup
1003 token-ring-default                act/unsup
1004 fddinet-default                   act/unsup
1005 trnet-default                     act/unsup
```

**ნაბიჯი 3.** გარე გამავალ ინტერფეისზე ”Trunk Mode”-ის გააქტიურება

```
Switch>enable
Switch#config t
Switch(config)#interface fastEthernet 0/24
Switch(config-if)#switchport mode trunk
```

**ნაბიჯი 4.** მარშრუტიზატორზე ქვეინტერფეისების შექმნა და დამისამართება

```
Router(config)#interface fastEthernet 0/0.1
Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip address 192.168.1.1 255.255.255.0
Router(config-subif)#no shut

Router(config-subif)#exit
Router(config)#interface fastEthernet 0/0.2
Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip address 192.168.2.1 255.255.255.0
Router(config-subif)#no shut

Router(config-subif)#exit
Router(config)#interface fastEthernet 0/0.3
Router(config-subif)#encapsulation dot1Q 30
Router(config-subif)#ip address 192.168.3.1 255.255.255.0
Router(config-subif)#no shut
```

Show runn ბრძანებით მივიღეთ შემდეგ სურათს:

```
interface FastEthernet0/0.1
 encapsulation dot1Q 10
 ip address 192.168.1.1 255.255.255.0
!
interface FastEthernet0/0.2
 encapsulation dot1Q 20
 ip address 192.168.2.1 255.255.255.0
!
interface FastEthernet0/0.3
 encapsulation dot1Q 30
 ip address 192.168.3.1 255.255.255.0
```

შეამოწმეთ კავშირი VLAN-ის წევრებს შორის

PC>ping 192.168.2.2

Reply from 192.168.2.2: bytes=32 time=110ms TTL=127

Reply from 192.168.2.2: bytes=32 time=125ms TTL=127

Reply from 192.168.2.2: bytes=32 time=125ms TTL=127

Reply from 192.168.2.2: bytes=32 time=110ms TTL=127

PC>ping 192.168.3.2

Reply from 192.168.3.2: bytes=32 time=125ms TTL=127

Reply from 192.168.3.2: bytes=32 time=109ms TTL=127

Reply from 192.168.3.2: bytes=32 time=125ms TTL=127

Reply from 192.168.3.2: bytes=32 time=125ms TTL=127

*პრაქტიკული სამუშაო:*

Switch#show vlan

| VLAN | Name | Status | Ports |
|---|---|---|---|
| 1 | default | active | |
| 10 | lab10 | active | Fa0/1, Fa0/2, Fa0/3, Fa0/4<br>Fa0/5, Fa0/6, Fa0/7, Fa0/8, Fa0/9 |
| 20 | lab20 | active | Fa0/10, Fa0/11, Fa0/12, Fa0/13, Fa0/14 |
| 30 | lab30 | active | Fa0/15, Fa0/16, Fa0/17, Fa0/18, Fa0/19, Fa0/20 |
| 99 | Management | active | Fa0/21, Fa0/22, Fa0/23, Fa0/24, Gig1/1, Gig1/2 |

- შექმენით მოცემულის შესაბამისი VLAN ქსელები და გააწევრიანეთ შესაბამისი ინტერფეისები
- შექმენით სათანადო ქსელის მოდელი და დაამისამართეთ თქვენი შეხედულებისამებრ (სხვადასხვა VLAN-ში გაწევრიანებულ ჰოსტებს მიანიჭეთ სხვადასხვა ქსელის მისამართები)
- ქსელში ჩართეთ მარშრუტიზატორი და დააკონფიგურირეთ იმგვარად, რომ სხვადასხვა VLAN-ის კვანძებს შორის არსებობდეს ლოგიკური კავშირი

## პროცესზე დაკვირვება

შემფასებელი აკვირდება შესაფასებელ პირის მუშაობას პროფესიული სტანდარტით(პროგრამით / მოდულით ) განსაზღვრული ამოცანების შესრულების პროცესში . დაკვირვებახორციელდება სამუშაო ადგილზე ან სამუშაო პირობებში, თუმცა დასაშვებია ჩატარდესკომპიუტერებით აღჭურვილლაბორატორიაში,სადაც შესაფასებელი პირი პრაქტიკულ საქმიანობას ეწევა. შემფასებელმა წინასწარ უნდა დაგეგმოს დაკვირვების პროცესი, იმის დასადგენად, თუ რამდენად სწორად იყენებს შესაფასებელი პირი ცოდნას, უნარებსა და ყველა რესურსს შედეგის მისაღწევად.შეფასება დადებითია იმ შემთხვევაში თუ სტუდენტი ყველა კითხვას სწორად გასცემს პასუხს.

*პრაქტიკული სამუშაო*

## კომუტატორის საბაზისო კონფიგურაცია

| სწავლის შედეგი | დასახელება | შეფასება | |
|---|---|---|---|
| | | კი | არა |
| კომუტატორის (Switch) საბაზისო კონფიგურაცია | სწორად განსაზღვრა კომუტატორის კონფიგურაციის რეჟიმებს | | |
| | სწორად მოახდინა კომუტატორის საბაზისო კონფიგურაცია | | |
| | სწორად შეინახა კონფიგურაცია, კონფიგურაციის ასლი და შეძლო მის აღდგენა. | | |
| Vlan-ების და Trunk-ების მუშაობის პრინციპის და ტიპების კონფიგურაცია | სწორად დააკონფიგურირა Vlan-ები და Trunk-ები | | |

# 4. ფიზიკური ქსელის დაგეგმვა და განხორციელება

## 4.1. პასიური ქსელური ინფრასტრუქტურის სტანდარტების და ტიპების გარჩევა, ფიზიკური ტოპოლოგიის ნახაზის შედგენა

### ქსელის დაპროექტება

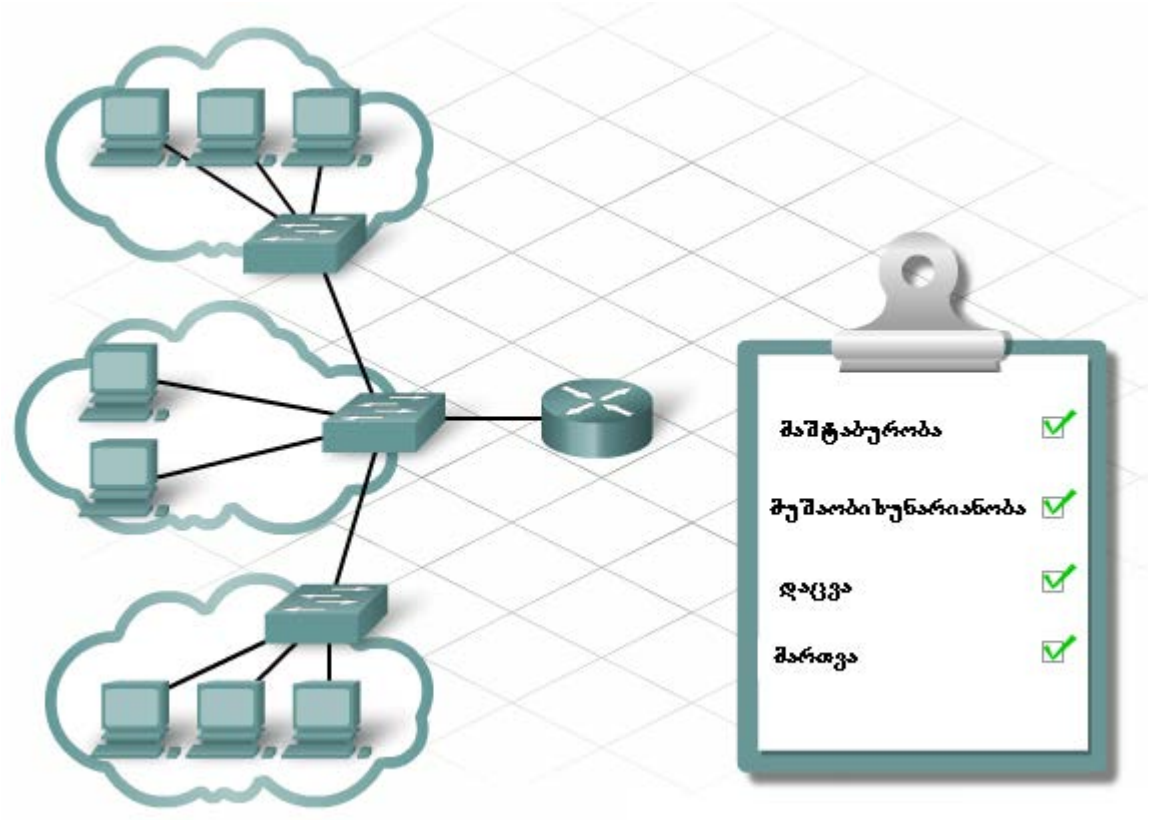
კომპიუტერული და საინფორმაციო ქსელები დიდ როლს თამაშობენ მცირე თუ მსხვილი ბიზნესის განვითარებაში. ისინი უზრუნველყოფენ სერვისებით და ბიზნესისათვის აუცილებელი რესურსებით. ყოველდღიური მოთხოვნების დასაკმაყოფილებლად კომპიუტერული ქსელები უფრო კომლექსური ხდება.M

ქსელისადმი წაყენებული მოთხოვნებია:

- ქსელი არ უნდა გამოვიდეს მწყობრიდან არც ერთ შემთხვევაში არც კავშირის არხების და არც მოწყობილობების მწყობრიდან გამოსვლისას და არც მისი გადატვირთვის შემთხვევაში.
- ქსელმა უნდა უზრუნველყოს ინფორმაციის საიმედო გადაცემა ჰოსტიდან ჰოსტამდე.
- ქსელი უნდა იყოს საიმედოდ დაცული. მან უნდა უზრუნველყოს მასში გადასაცემი მონაცემების და აგრეთვე შენახული ინფორმაციის საიმედოდ შენახვა.
- ქსელი უნდა იყოს მოქნილი ანუ ადვილად შესაძლებელი მისი მოდიფიკაცია და ადაპტირება ქსელის გაფართოებისადმი.
- ქსელის ხარვეზების გამოჩენის შემთხვევაში გაადვილდეს გაუმართაობების აღმოჩენა.

### დაპროექტების ფუნდამენტური მიზნები

- მაშტაბურობა
- მუშაობისუნარიანობა
- დაცვა
- მართვა



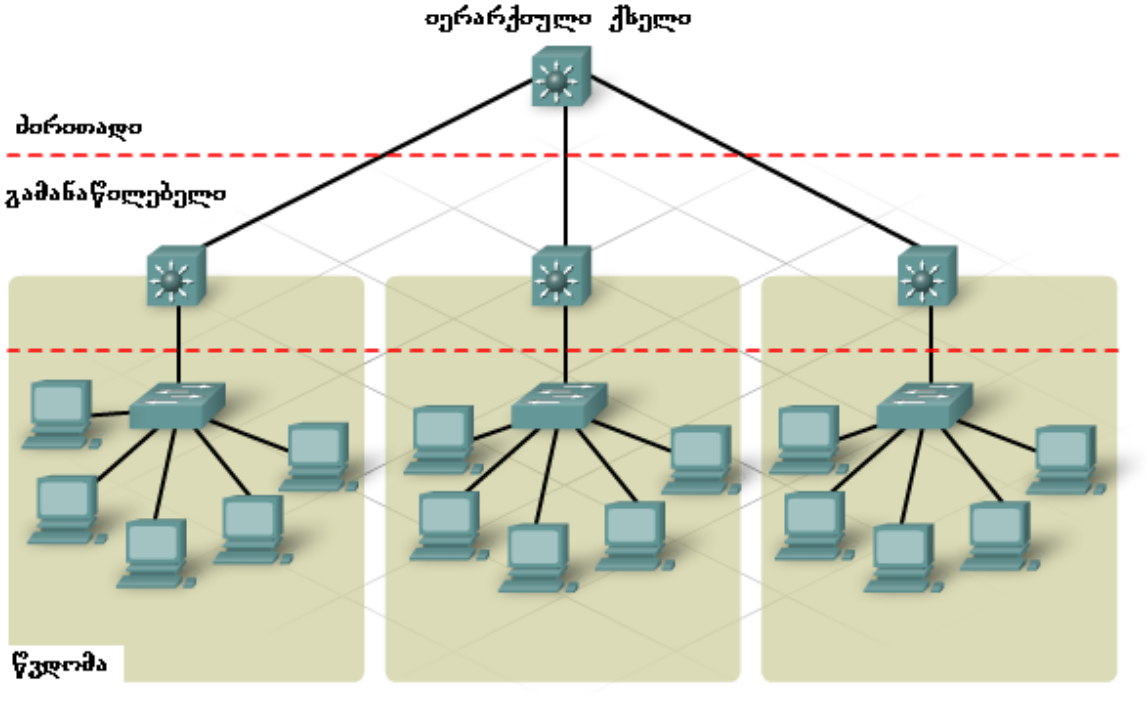

*სურ.4.1. 1*

### ქსელის იერარქიული დაპროექტების ღირებულებები

იმისათვის, რომ დაკმაყოფილდეს დაპროექტების ოთხივე ფუნდამენტური მოთხოვნა, ქსელის არქიტექტურა უნდა იყოს მოქნილი და შესაძლებელი იყოს მისი გაფართოება.

ქსელების იერარქიული არქიტეტქურა გულისხმობს მოწყობილობების გაერთიანებას სხვადასხვა ქსელებში. ქსელების ორგანიზაცია გულისხმობს მის დონეებად დაყოფას. ქსელის იერარქიული დაპრორქტების მოდელი შედგება სამი ძირითადი დონისაგან:

- **ძირითადი დონე (Core Layer)** – აკავშირებს გამანაწილებელი დონის მოწყობილობებს
- **გამანაწილებელი დონე (Distribution Layer)** – აკავშირებს მცირე ლოკალურ ქსელებს
- **წვდომის დონე (Access Layer)** – უზრუნველყოფს კავშირს ჰოსტებთან და საბოლოო მოწყობილობებთან



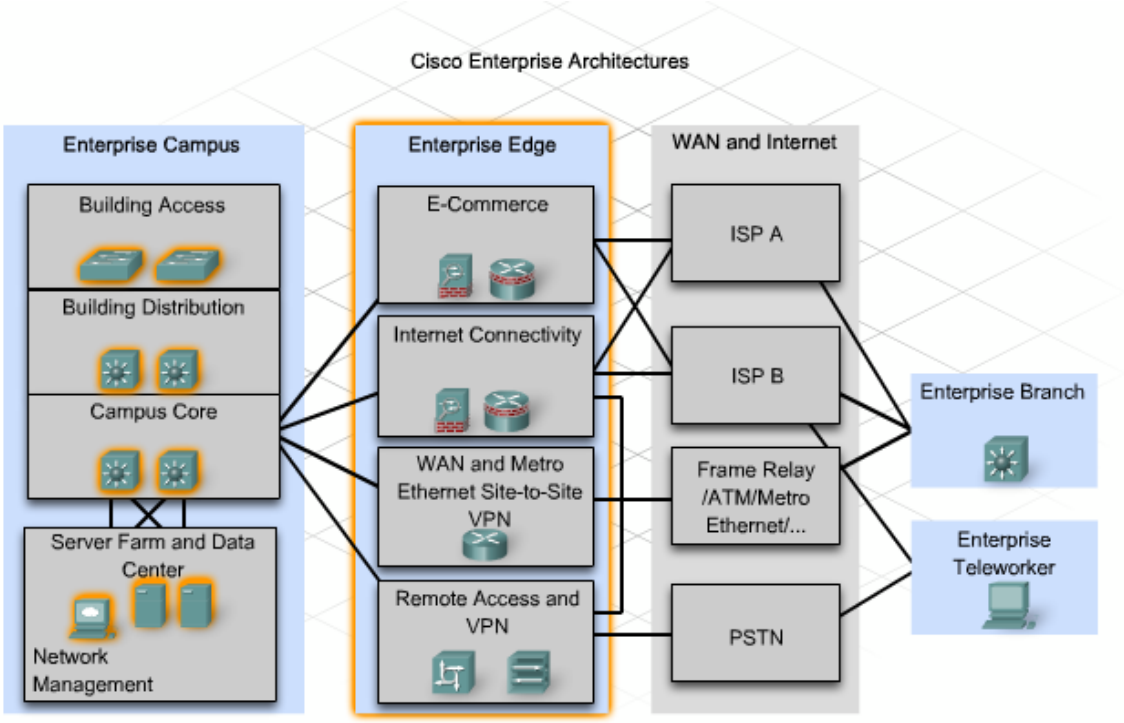

*სურ.4.1. 2*

### ქსელის იერარქიული დაპროექტების უპირატესობები

ცისკოს კორპორატიული არქიტექტურები იყენებს სამ დონიან იერარქიულ დიზაინს, რომელიც იყოფა მოდულებად. Mმოდულები წარმოადგენენ არეებს განსხვავებული ლოგიკური და ფიზიკური კავშირებით. Mმოდულებად დაყოფა ქსელის დიზაინს ხდის უფრო მოქნილს. რომელიც აადვილებს დანერგვასა და გაუმართაობების აღმოჩენას. Mმოდულურ ქსელის სამი ძირითადი არეებია:

- **კორპორატიული კამპუსი** – ეს არე შეიცავს ქსელის ელემენტებს დამოუკიდებელი მუშაობისათვის ცალკეული კამპუსის ან განყოფილების ფარგლებში.
- **სერვერების ჯგუფი** – კორპორატიული კამპუსის კომპონენტია. საინფორმაციო ცენტრის სერვერების ჯგუფი იცავს სერვერების რესურსებს და უზრუნველყოფს მათ დუბლირებული, საიმედო მაღალ-სიჩქარიანი კავშირით.
- **კორპორატიული საზღვარი** – ეს არე ფილტრავს გარედან შემოსულ ტრაფიკს და ამისამართებს კორპორატიულ ქსელში. ის შეიცავს ყველა ელემენტს ეფექტური და საიმედო

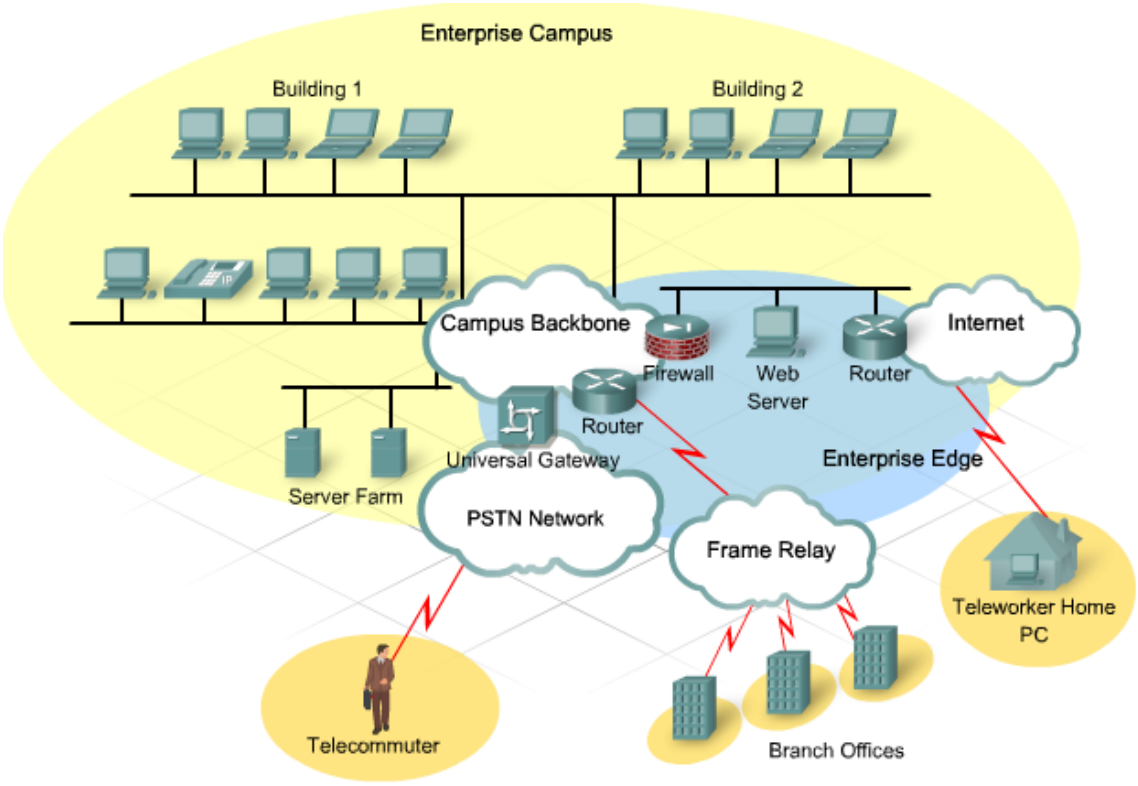
კავშირისათვის კორპორატიულ კამპუსსა და დაშორებულ ადგილმდებარეობებს შორის, დაშორებულ მომხმარებლებს შორის და ინტერნეტთან.




*სურ.4.1. 3*

ცისკოს კორპორატიული არქიტექტურის მოდულურ სტრუქტურას გააჩნია შემდეგი უპირატესობები:

- შესაძლებელია დეტერმინირებული ქსელის აგება მოდულებს შორის მკაფიოდ გამოკვეთილი საზღვრებით. ის უზრუნველყოფს მკაფიო დემარკაციის წერტილებს, რის მიხედვითაც ქსელის დიზაინერები განასხვავებენ სად წარმოიქმნება ტრაფიკი და რა მიმართულებით მიედინება ის.
- ცალკეულ მოდულებად დაყოფა აადვილებს დაპროექტების ამოცანას. რადგან ცალკეული არეების საჭიროებების განსაზღვრა ადვილია დამპროექტებლისთვის.
- უზრუნველყოფს მაშტაბურობას, რომელიც საშუალებას აძლევს კორპორაციას გაუადვილდეს მოდულების დამატება.
- საშუალებას აძლევს დამპროექტებლებს დაამატონ სერვისები და გადაწყვეტილებები ქსელის არქიტექტურის ცვლილებების გარეშე




*სურ.4.1. 4*

## ქსელის დაპროექტების მეთოდოლოგია

მსხვილმაშტაბიანი ქსელის დაპროექტება მოიცავს სამ ძირითად ეტაპს:

**1 ეტაპი:** ქსელური მოთხოვნების განსაზღვრა.

**2 ეტაპი:** არსებული ქსელის დახასიათება.

**3 ეტაპი:** ქსელის ტოპოლოგიისა და ამოცანების განსაზღვრა.

### ქსელური მოთხოვნების განსაზღვრა.

ქსელის დამპროექტებელი ითვალისწინებს მომხმარებლების მოთხოვნებს. მოთხოვნები შეიძლება დაიყოს ორ კატეგორიად:

- ბიზნეს მოთხოვნები – მიმართულია თუ როგორ გახადოს ქსელმა ბიზნესი უფრო წარმატებული
- ტექნიკური მოთხოვნები – ფოკუსირებულია იმაზე თუ რა სახის ტექნოლოგია ინერგება ქსელში.

### არსებული ქსელის დახასიათება

ხორციელდება არსებული ქსელის სერვისების ანალიზი. აუცილებლად უნდა მოხდეს შედარება არსებული ქსელის ფუნქციონიერებასა და ახალი განსაზღვრული ქსელის პროექტს შორის. დამპროექტებლები განსაზღვრავენ თუ რომელი არსებული აპარატურა, ინფრასტრუქტურა და პროტოკოლები შეიძლება ხელახლა გამოყენებულ იქნას, და რა ახალი აპარატურა და პროტოკოლებია საჭირო იმისათვის, რომ დასრულდეს პროექტი.

### ქსელის ტოპოლოგიის განსაზღვრა

ქსელის დაპროექტების მთავარი სტრატეგიაა ზემოდან - ქვემოთ პრინციპი (top-daun), რომლის მიხედვითაც განისაზღვრება ქსელური პროცედურები და სერვისის მოთხოვნები, ხოლო შემდგომ დაპროექტებისას ქსელმა უზრუნველყოს ყველა ეს მოთხოვნა.

პროექტის დასრულების შემდეგ მიმდინარეობს შექმნილი ქსელის პროტოტიპის ტესტირება. რის შედეგადაც მოწმდება დაპროექტებული ქსელის ფუნქციონირება მის საბოლოო დანერგვამდე.

### ძირითადი ქსელი

ძირითად ქსელს ხშირად უწოდებენ ქსელის ხერხემალს (Network Backbone). როუტერები და კომუტატორები ამ დონეზე უზრუნველყოფენ მაღალ სიჩქარიან კავშირს. კორპორატიულ ლოკალურ ქსელში ძირითადი დონე აერთიანებს მრავალ შენობასა და

საიტებს, აგრეთვე უზრუნველყოფს კავშირს სერვერების ჯგუფთან. ძირითადი დონე უზრუნველყოფს კავშირს ინტერნეტთან, ვირტუალურ კერძო ქსელთან (VPN), ექსტრანეტთან და წვდომას გლობალურ ქსელთან (WAN).

ძირითადი ქსელის დანერგვა ამცირებს ქსელის კომპლექსურობას და აადვილებს მართვას და ამასთანავე გაუმართაობების აღმოჩენას (troubleshoot).

### ძირითადი დონის ამოცანები

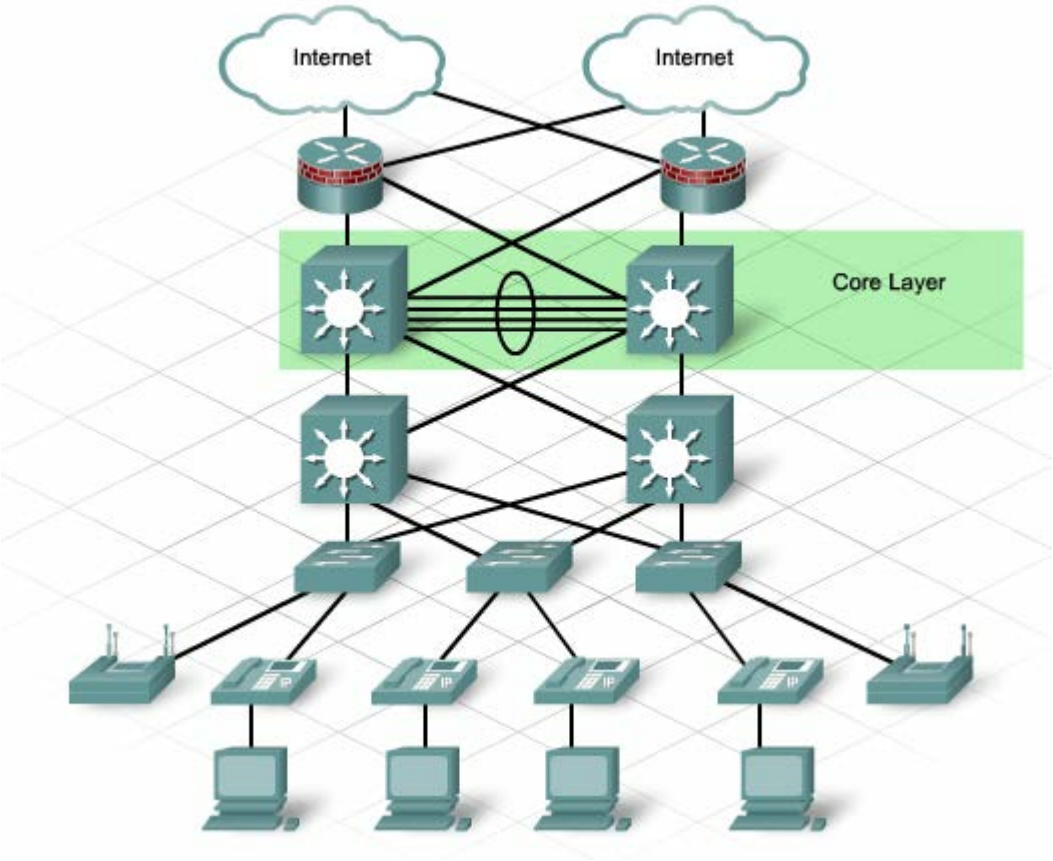
ძირითადი დონის სტრუქტურა საშუალებას იძლევა უზრუნველყოს მონაცემთა სწრაფ და ეფექტურ გადაცემას ერთი სეგმენტიდან მეორეში. ძირითად დონეზე დაპროექტების მთავარი მიზნებია:

- უზრუნველყოს წვდომის დრო 100%-ით;
- მაქსიმალური გამტარუნარიანობა;
- ქსელის გაფართოება

### ძირითადი ქსელის ტექნოლოგიები

ტექნოლოგიები, რომლებიც გამოიყენება ძირითად დონეზე, მოიცავენ:

- როუტერებს ან მრავალდონიან კომუტატორებს (Switches), რომლებშიც კომბინერებულია, როგორც მარშუტიზაციის ასევე კომუტაციის პროცედურები ერთ და იმავე მოწყობილობაში
- დარეზერვება და ბალანსირება;
- მაღალ-სიჩქარიანი და აგრეგირებული ლინკები
- მარშუტიზაციის პროტოკოლები: Enhanced Interior Gateway Routing Protocol (EIGRP) , Open Shortest Path First (OSPF)




*სურ.4.1. 5*

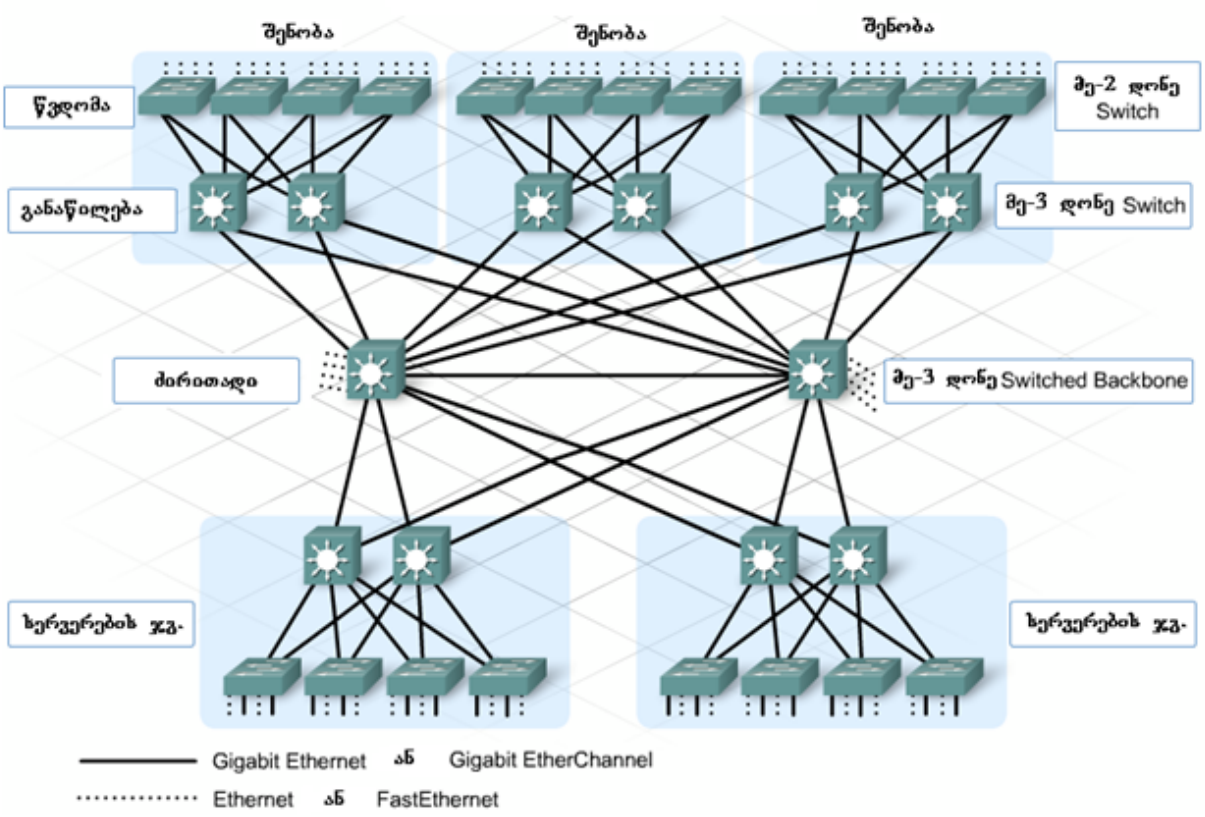
### სარეზერვო ლინკები

ძირითად დონეზე სარეზერვო ლინკების არსებობა საშუალებას იძლევა მოიძებნოს მონაცემთა გადაცემის ალტერნატიული მარშუტი გაუმართაობების აღმოჩენის შემთხვევაში. როცა მე-3 დონის მოწყობილობები განთავსებულია ძირითად დონეზე, სარეზერვო ლინკებს შეუძლიათ შეასრულონ დამატებით ბალანსის როლი და ამასთანავე უზრუნველყონ ბეკაპირება.

### ბადისებრი ტოპოლოგია

ძირითადი დონე უმეტე შემთხვევაში იყენებს სრულ ან ნაწილობრივ ბადისებრ ტოპოლოგიას. სრული ბადისებრი ტოპოლოგიის შემთხვევაში ერთი მოწყობილობა

უკავშირდება ყველა დანარჩენს. ერთის მხრივ სრული ბადის ტოპოლოგიას გააჩნია უპირატესობა, რომელიც უზრუნველყოფს სრულ სარეზეერვო ქსელს,ა მაგრამ მეორეს მხრივ, ძალიან რთულია და ძვირადღირებულიც. Dიდი ქსელების შემთხვევაში გამოიყენება მოდიფიცირებული ნაწილობრივი ბადის ტოპოლოგია, რომლის შემთხვევაში ყოველი მოწყობილობა დაკავშირებულია მინიმუმ ორ სხვა მოწყობილობასთან, რომელიც საკმაოდ ოპტიმალურ რეზერვს ზედმეტი კომპლექსურობის გარეშე სრული ბადის ტოპოლოგიისგან განსხვავებით.




*სურ.4.1. 6*

## დარეზერვება ბადისებრ ტოპოლოგიაში

### ქსელური ტრაფიკის პრიორიტიზაცია

ქსელის დამპროექტებელმა უნდა უზრუნველყოს ქსელის მდგრადობა გაუმართაობებისადმი და მათი წარმოქნის შემთხვევაში სწრაფად უნდა აღმოიფხვრას ისინი. მთავარი როუტერები და კომუტატორები უნდა შეიცავდნენ:

- ორმაგ დენის წყაროს და ვენტილატორებს
- მოდულურ არქიტექტურას
- დამატებით მართვის მოდულებს

სარეზერვო კომპონენტები ზრდიან ღირებულებას, მაგრამ ამავე დროს ზრდის ინვესტიციებსაც.

### გამანაწილებელი დონე

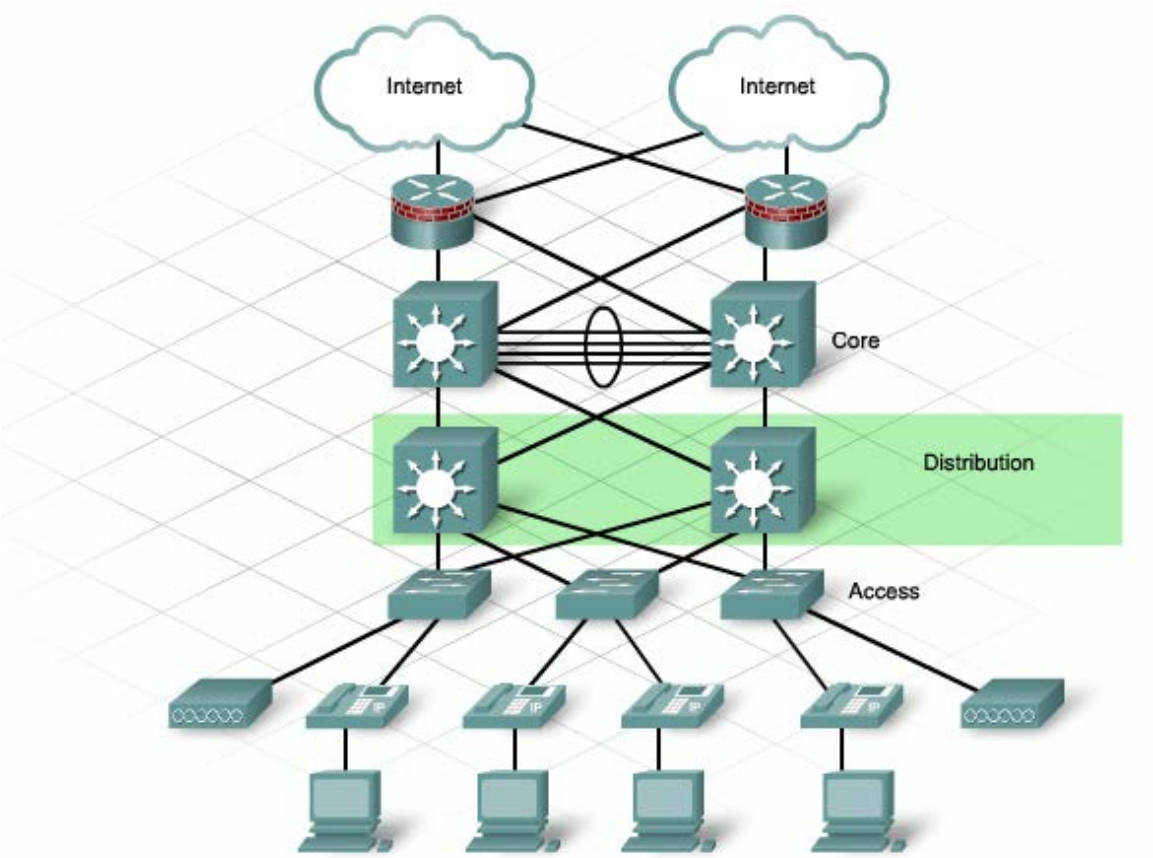
გამანაწილებელი დონე წარმოადგენს მარშუტიზაციის საზღვარს წვდომისა და ძირითად დონეებს შორის. Uუზრუნველყოფს კავშირს დაშორებულ საითებსა და ძირითად დონეს შორის.

### გამანაწილებელი ქსელის მარშუტიზაცია

წვდომის დონე ჩვეულებრივ იყენებს მეორე დონის ტექნოლოგიას. Gგამანაწილებელი დონე კი იყენებს მე-3 დონის მოწყობილობებს. როუტერები ან მრავალდონიანი კომუტატორები (switches), რომლებიც განთავსებულია გამანაწილებელ დონეზე, ასრულებენ სხვადასხვა ფუნქციებს, რომლებიც კრიტიკულია ქსელის არქიტექტურის მოთხოვნების მიმართ. ეს მოთხოვნებია:

- ტრაფიკის ფილტრაცია და მართვა;
- წვდომის მართვა;
- მარშუტების შეჯამება ძირითად დონეზე გაშვებამდე
- ძირითადი დონის იზოლირება წვდომის დონის გაუმართაობებისაგან
- მარშუტიზაცია წვდომის დონის ვირტუალურ ლოკალური ქსელებს (VLAN) შორის

გამანაწილებელი ქსელის მოწყობილობები გამოიყენება აგრეთვე რიგების მართვისათვის და ტრაფიკის პრიორიტიზაციისათვის, სანამ გაიგზავნება კამპუსის ძირიტად დონეზე.




*სურ.4.1. 7*

### მაგისტრალური არხი

მაგისტრალური არხი ხშირად კონფიგურირდება ძირითადი და გამანაწილებებლი დონის მოწყობილობებს შორის. ისინი გამოიყენება ტრაფიკი გადასაცემად ვირტუალური ლოკალური ქსელების მოწყობილობებს შორის. ქსელის დამპროექტებელი განიხილავს

მთლიანად VLAN-ის სტრატეგიას და ქსელის ტრაფიკის მოდელს მაგისტრალური კავშირის ხაზების დაპროექტებისას.

### სარეზერვო არხები

გამანაწილებელ დონეზე სარეზერვო არხების არსებობის შემთხვევაში მოწყობილობების დაკონფიგურირება შესაძლებელია დატვირთვის დაბალანსებით. რაც ზრდის გამტარიანობის ზოლს გამოყენებითი პროგრამებისათვის.

### გამანაწილებელი დონის ტოპოლოგია

გამანაწილებელი დონის ქსელები იყენებენ ნაწილობრივი ბადის ტოპოლოგიას. ეს ტოპოლოგია უზრუნველყოფს ქსელს საკმარისი სარეზერვო გზებით, რის შედეგადაც ქსელის ფუნქციონირება არ წყდება მოწყობილობის თუ არხის მწყობრიდან გამოსვლის შემთხვევაში. როცა გამანაწილებელი დონის მოწყობილობები იმყოფებიან ერთ და იგივე საინფორმაციო ცენტრში, მაშინ ისინი უკავშირდებიან ერთმანეთს გიგაბიტიანი არხებით. მაგრამ თუ ისინი არიან განთავსებულნი შორს ერთმანეთისაგან, მაშინ გამოიყენება ოპტიკურ-ბოჭკოვანი კაბელი. კომუტატორები მრავალჯერადი მაღალ სიჩქარიანი ოპტიკურ-ბოჭკოვანი კავშირებით ჯდება ძალიან ძვირი და ამიტომ აუცილებელია დაპროექტებისას ფრთხილად შეირჩეს საკმარისი ოპტიკური პორტების რიცხვი, რათა მიღწეულ იქნას საჭირო გამტარიანობის ზოლი და დუბლირება.

## 4.2. ლოკალური ქსელის მისამართები

გარკვეული მისამართები აქტუალურია შიდა ლოკალურ ქსელში, ეს მისამართები „მუშაობს“ მხოლოდ შიდა ლოკალურ ქსელში და მოცემულ კონკრეტულ ქსელში უნდა იყოს უნიკალური, გლობალურ ქსელში ის გარდაიქმნება (NAT კონფიგურირებით) გარე გლობალურ მისამართად, რომელიც თავის მხრივ უნიკალურია მსოფლიო მასშტაბით

**A კლასში** - (1 ქსელი)

• 10.0.0.0 ქსელი (10.0.0.1 -დან 10.255.255.254-ის ჩათვლით)

**B კლასში** - (16 ქსელი)

• 172.16.0.0 ქსელი (172.16.0.1 -დან 172.16.255.254-ის ჩათვლით)

• 172.17.0.0 ქსელი (172.17.0.1 -დან 172.17.255.254-ის ჩათვლით)

და ა.შ.

• 172.31.0.0 ქსელი (172.31.0.1 -დან 172.31.255.254-ის ჩათვლით)

**C კლასში** - (256 ქსელი)

• 192.168.0.0 ქსელი (192.168.0.1 -დან 192.168.0.254-ის ჩათვლით)

• 192.168.1.0 ქსელი (192.168.1.1 -დან 192.168.1.254-ის ჩათვლით)

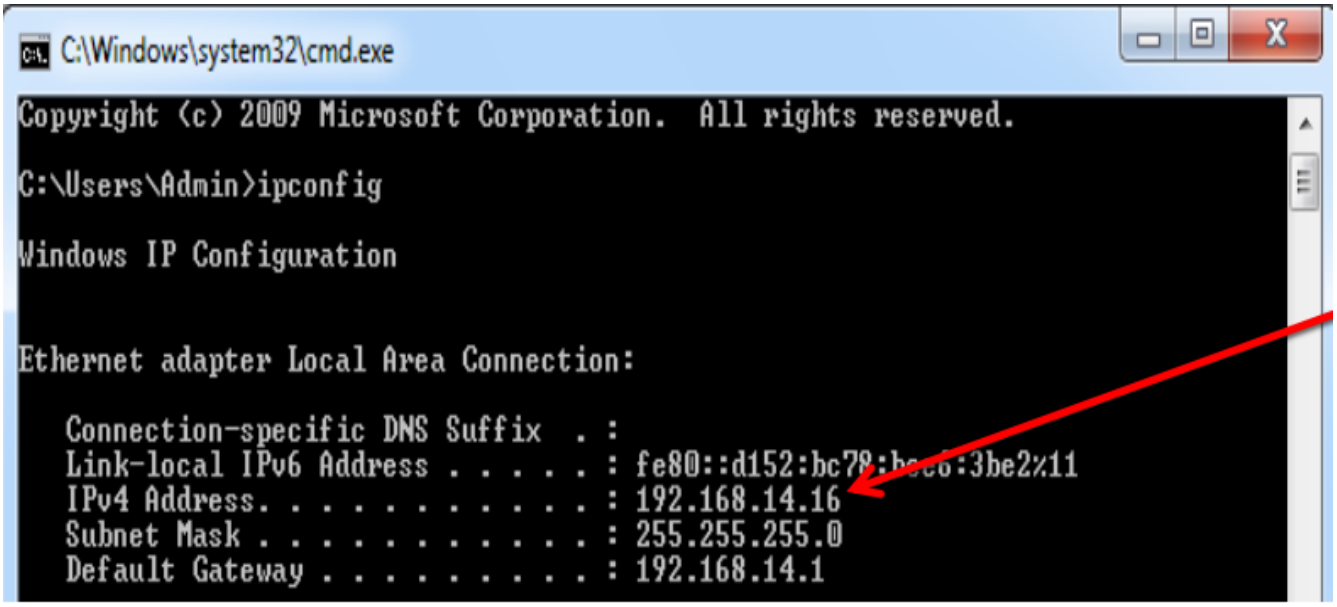
და ა.შ.

• 192.168.255.0 ქსელი (192.168.255.1 -დან 192.168.255.254-ის ჩათვლით)

**უნდა გვახსოვდეს**

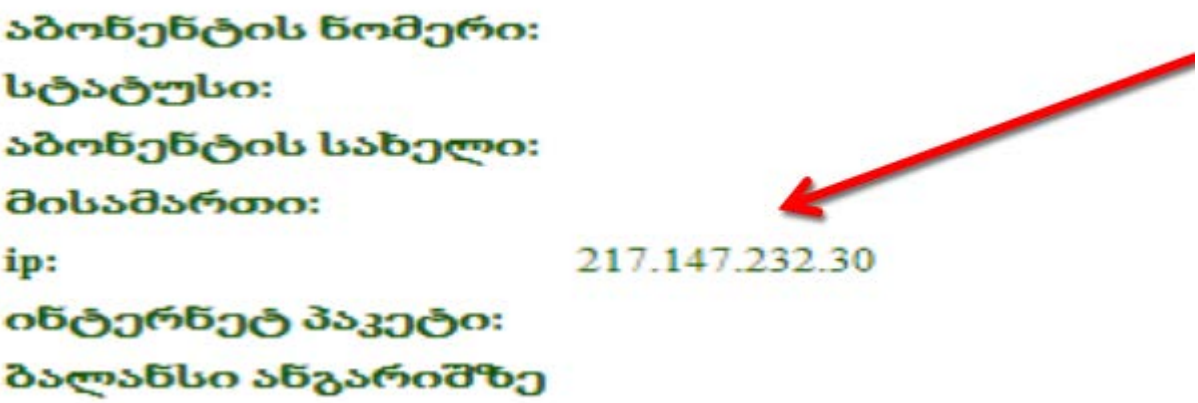
- ✓ ჩვენს კომპიუტერს აქვს ლოკალური მისამართი, რომელიც უნიკალურია მოცემულ კონკრეტულ ქსელში:

Start – (Run XP-ის შემთხვევაში) CMD – IPCONFIG



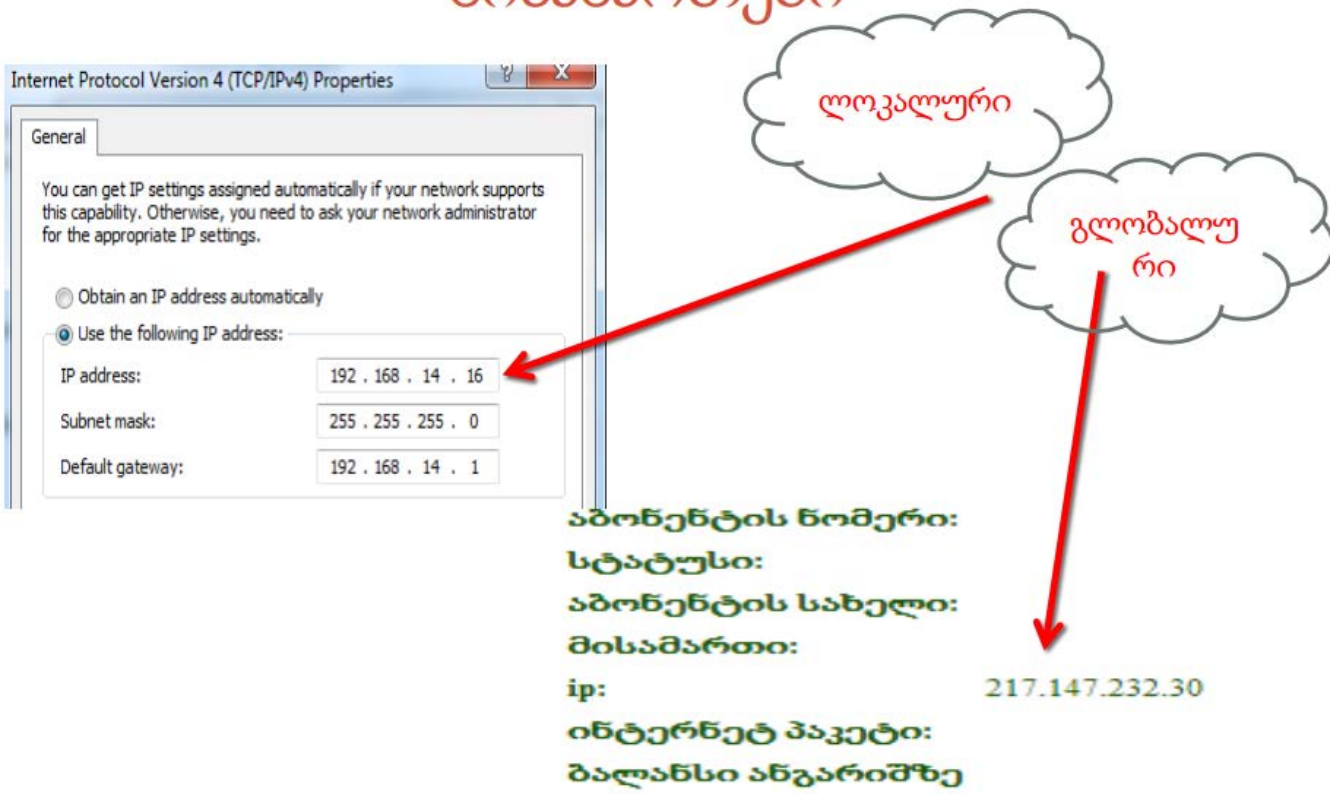

*სურ.4.2. 1*

- ✓ ჩვენს კომპიუტერს აქვს ლოკალური მისამართი, რომელიც უნიკალურია მოცემულ კონკრეტულ ქსელში, ხოლო გლობალურ ქსელში მას აქვს შესაბამისი უნიკალური გლობალური მისამართი




*სურ.4.2. 2*

# ლოკალური და გლობალური მისამართები




*სურ.4.2. 3*

**პროცესზე დაკვირვება**

შემფასებელი აკვირდება შესაფასებელ პირის მუშაობას პროფესიული სტანდარტით(პროგრამით / მოდულით ) განსაზღვრული ამოცანების შესრულების პროცესში . დაკვირვებახორციელდება სამუშაო ადგილზე ან სამუშაო პირობებში, თუმცა დასაშვებია ჩატარდესკომპიუტერებით აღჭურვილლაბორატორიაში,სადაც შესაფასებელი პირი პრაქტიკულ საქმიანობას ეწევა. შემფასებელმა წინასწარ უნდა დაგეგმოს დაკვირვების პროცესი, იმის დასადგენად, თუ რამდენად სწორად იყენებს შესაფასებელი პირი ცოდნას, უნარებსა და ყველა რესურსს შედეგის მისაღწევად.შეფასება დადებითია იმ შემთხვევაში თუ სტუდენტი ყველა კითხვას სწორად გასცემს პასუხს.

*პრაქტიკული სავარჯიშო -*

**ფიზიკური ტოპოლოგიის ნახაზის და დოკუმენტაციის შედგენა**

| სწავლის შედეგი | დასახელება | შეფასება | |
|---|---|---|---|
| | | კი | არა |
| ფიზიკური ტოპოლოგიის ნახაზის და დოკუმენტაციის შედგენა | სწორად განსხვავა ლოგიკური მისამართების კლასები შიდა და გარე დამისამართების დიაპაზონები | | |
| | სწორად დახაზა ფიზიკურ ტოპოლოგია მოთხოვნების ან/და სიტუაციების შესაბამისად | | |
| | სწორად დაგეგმა ფიზიკური ტოპოლოგიის აწყობისთვის საჭირო სამუშაოები | | |

# 5. მონიტორინგის და ინციდენტების აღმოჩენის სერვისები, უსაფრთხოების საფუძვლები

## 5.1. SNMP პროტოკოლის კონფიგურირება.

SNMP განვითარდა იმისათვის, რომ ადმინისტრატორებს ჰქონდეთ საშუალება მართონ სერვერები, სამუშაო სადგურები (Workstations), მარშრუტიზატორები, კომუტატორები და უსაფრთხოების ტექნიკა, IP ქსელში. ის საშუალებას აძლევს ქსელის ადმინისტრატორებს მართონ ქსელის წარმადობა, იპოვონ და აღმოფხვრან ქსელური პრობლემები და დაგეგმონ ქსელის გაზრდა.

SNMP არის გამოყენებითი დონის პროტოკოლი, რომელიც უზრუნველყოფს შეტყობინების ფორმატს, მენეჯერებსა და აგენტებს შორის კომუნიკაციისთვის. SNMP სისტემა შედგება სამი ელემენტისაგან:

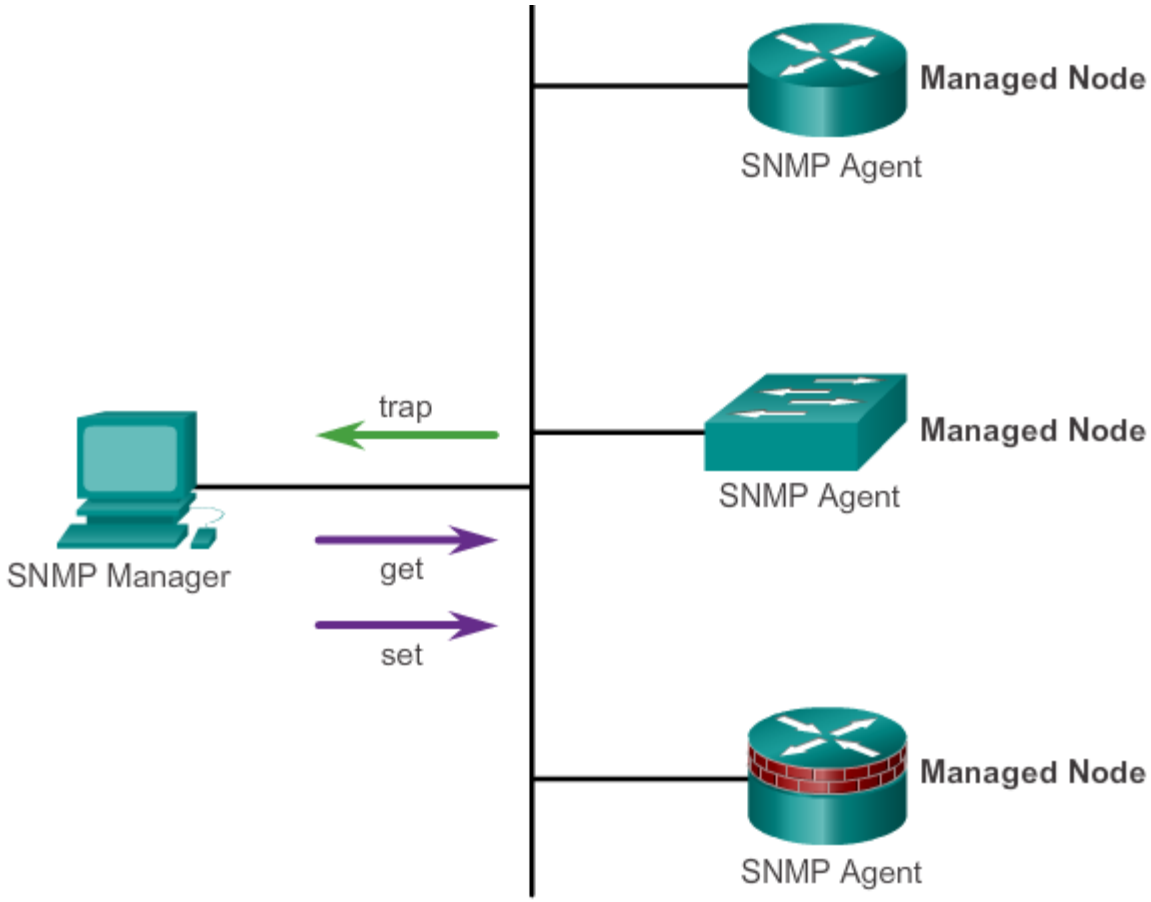
- SNMP მენეჯერი
- SNMP აგენტი (მართვადი კვანძი)
- მართვის საინფორმაციო ბაზა (MIB)

ქსელურ მოწყობილობაზე SNMP-ს კონფიგურაციისთვის, პირველ რიგში აუცილებელია განისაზღვროს ურთიერთობა მენეჯერსა და აგენტს შორის.

SNMP მენეჯერი არის ქსელის მართვის სისტემის (NMS) ნაწილი. SNMP მენეჯერი უშვებს SNMP-ის მართვის პროგრამულ უზრუნველყოფას. როგორც 5.1 სურათზეა მოცემული, SNMP მენეჯერს შეუძლია შეაგროვოს ინფორმაცია SNMP აგენტებიდან „get (მიღება)“ მოქმედების გამოყენებით და შეუძლი კონფიგურაციის შეცვლა აგენტზე „Set (გაშვება)“ მოქმედების გამოყენებით. დამატებით, SNMP აგენტებს შეუძლიათ ინფორმაციის გადაგზავნა პირდაპირ ქსელის მართვის სისტემასთან (NMS), „traps (მახეები)”-ის გამოყენებით.

SNMP აგენტი და მართვის საინფორმაციო ბაზა (MIB) მიეკუთვნება ქსელური მოწყობილობების კლიენტებს. ის ქსელური მოწყობილობები, რომელთა მართვაც შეიძლება,

კომუტატორების, მარშრუტიზატორების, სერვერების, ფაიერვოლების და სამუშაო სადგურების ჩათვლით, აღჭურვილია SNMP აგენტი პროგრამული უზრუნველყოფის მოდულით. მართვის საინფორმაციო ბაზა (MIB) ინახავს მონაცემებს მოწყობილობის მუშაობის შესახებ და განკუთვნილია იმისთვის რომ იყოს ხელმისაწვდომი ავტორიზებული დაშორებული მომხმარებლებისთვის. SNMP აგენტი პასუხისმგებელია ლოკალური მართვის საინფორმაციო ბაზის წვდომის უზრუნველყოფაზე ობიექტებთან, რომლებიც ასახავენ რესურსებსა და საქმიანობას.




სურ. 5.1. მარტივი ქსელის მართვის პროტოკოლი (SNMP)

SNMP განსაზღვრავს თუ როგორ იცვლება სამართავი ინფორმაცია ქსელის მართვის აპლიკაციებსა და მართვად აგენტებს შორის. SNMP იყენებს UDP პროტოკოლს, პორტის ნომრით 162, რათა მიიღოს და გააგზავნოს მართვის ინფორმაცია.

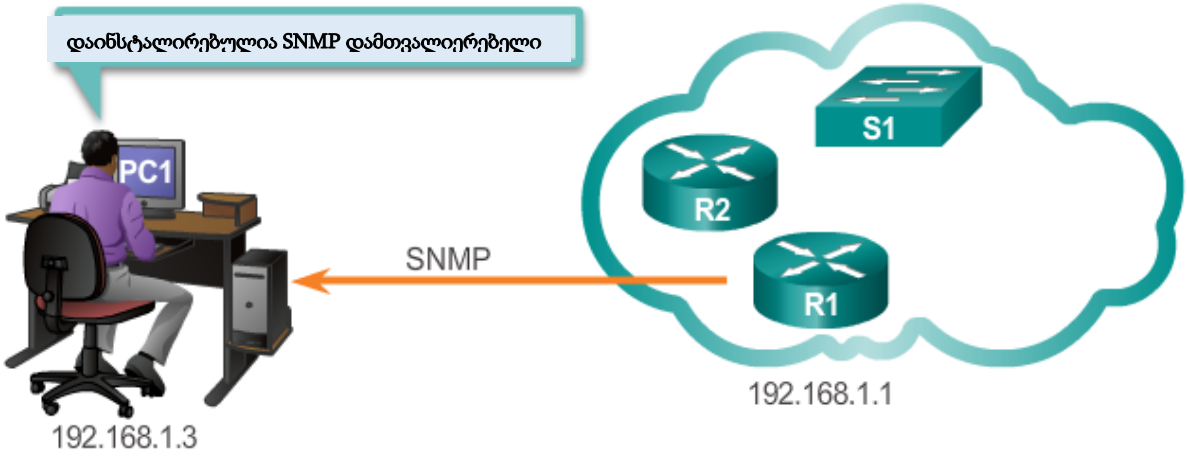
5.1.1 SNMP კონფიგურაციის ეტაპები

ქსელის ადმინისტრატორს შეუძლია SNMPv2-ის კონფიგურაცია ქსელური მოწყობილობებიდან ქსელის ინფორმაციის მისაღებად. როგორც 5.1.1 სურათზეა ნაჩვენები, SNMP-ს კონფიგურაციის ყველა ბაზისური ეტაპი არის საერთო კონფიგურაციის რეჟიმში.

პირველი ეტაპი. (აუცილებელი) დააკონფიგურეთ მწკრივების ერთობა (Community String) და დაშვების დონე (მხოლოდ ნახვა ან ნახვა-ჩაწერა) **snmp-server community** *string* **ro | rw** ბრძანებით.

მეორე ეტაპი. (დამატებითი) მოახდინეთ მოწყობილობის ადგილმდებარეობის დოკუმენტირება **snmp-server** location *text* ბრძანების გამოყენებით.

მესამე ეტაპი. (დამატებითი) მოახდინეთ სისტემური კონტაქტების დოკუმენტირება **snmp-server** contact *text* ბრძანებით.




```
R1(config)# snmp-server community batonaug ro SNMP_ACL
R1(config)# snmp-server location NOC_SNMP_MANAGER
R1(config)# snmp-server contact Wayne World
R1(config)# snmp-server host 192.168.1.3 version 2c batonaug
R1(config)# snmp-server enable traps
R1(config)# ip access-list standard SNMP_ACL
R1(config-std-nacl)# permit 192.168.1.3
```

სურ. 5.1.1. SNMP მენეჯერის კონფიგურაციის მხარაჭერა

მეოთხე ეტაპი. (დამატებითი) აკრძალეთ SNMP-ს წვდომა ქსელის მართვის სისტემის (NMS) ჰოსტებთან (SNMP მენეჯერები), რომლებიც დაშვებულნი არიან ACL-ის მიერ: განსაზღვრეთ ACL და შემდეგ მიუთითეთ ACL **snmp-server community** *string access-list-number-or-name* ბრძანების გამოყენებით. მოცემული ბრძანება გამოიყენება როგორც მწკრივების მისათითებლად, ისე SNMP წვდომის აკრძალვისთვის ACL-ების მეშვეობით. სურვილის შემთხვევაში პირველი და მეოთხე ეტაპი შეიძლება გაერთიანდეს ერთ ეტაპად. Cisco ქსელური მოწყობილობა აერთიანებს ორ ბრძანებას ერთში, თუ ისინი შეტანილია ცალ-ცალკე.

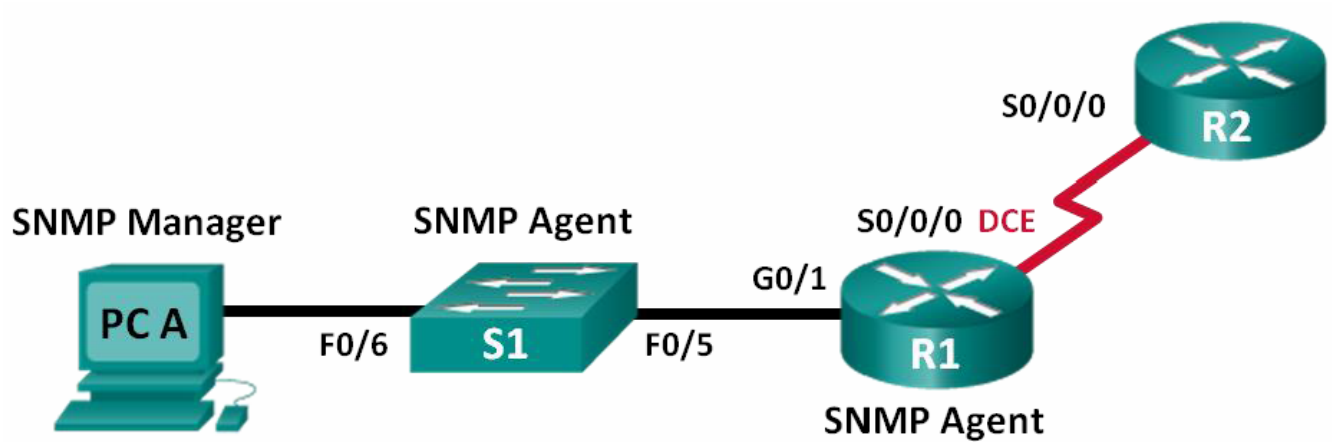
მეხუთე ეტაპი. (დამატებითი) მიუთითეთ SNMP trap ოპერაციების მიმღები **snmp-server host** *host-id* [version {1 | 2c | 3 [auth | noauth | priv]}] *community-string* ბრძანების გამოყენებით. ნაგულისხმევად trap მენეჯერ არ არის მითითებული.

მეექვსე ეტაპი. (დამატებითი) ჩართეთ traps (მახეები) SNMP აგენტზე **snmp-server enable traps** *notification-types* ბრძანებით. თუ მოცემულ ბრძანებაში არცერთი trap შეტყობინების ტიპი არ არის მითითებული, მაშინ ყველა ტიპის trap-ი იქნება გაგზავნილი. ამ ბრძანების განმეორებითი გამოყენება მოითხოვება, მაშინ თუ განსაზღვრული ტიპის trap ქვეჯგუფებია სასურველი.

**შენიშვნა:** ნაგულისხმევად, SNMP-ს არ აქვს არანაირი trap-ები მომართული. ამ ბრძანების გარეშე, SNMP მენეჯერებს შეუძლიათ ამოირჩიონ ყველა მართებული ინფორმაცია.

*5.1.1.1 - ლაბორატორიული სამუშაო - SNMP-ს კონფიგურაცია*

**ტოპოლოგია:**




*სურ.5.1.1.1. 1*

**მისამართების ცხრილი:**

| მოწყობილობა | ინტერფეისი | IP მისამართი | ქვექსელის ნიღაბი | ნაგულისხმევი გასასვლელი |
|---|---|---|---|---|
| R1 | G0/1 | 192.168.1.1 | 255.255.255.0 | N/A |
| | S0/0/0 | 192.168.2.1 | 255.255.255.252 | N/A |
| R2 | S0/0/0 | 192.168.2.2 | 255.255.255.252 | N/A |
| S1 | VLAN 1 | 192.168.1.2 | 255.255.255.0 | N/A |
| PC-A | NIC | 192.168.1.3 | 255.255.255.0 | 192.168.1.1 |

**შესასრულებელი დავალებები:**

**ნაწილი №1: ქსელის აწყობა და მოწყობილობის ბაზისური კონფიგურაცია**

**ნაწილი №2: SNMP მმართველისა და აგენტის კონფიგურაცია**

**ნაწილი №3: OID კოდების კონვერტაცია Cisco SNMP Object Navigator-თან**

## ზოგადი ინფორმაცია / სცენარი

**Simple Network Management Protocol (SNMP)** არის ქსელის მართვის პროტოკოლი და **IETF** სტანდარტი, რომელიც შეიძლება გამოყენებულ იქნას ქსელში კლიენტების მონიტორინგისა და მართვისათვის. **SNMP** შეიძლება გამოყენებულ იქნას ცვლადების მიღებისა და დაყენებისათვის, რომლებიც დაკავშირებულია ქსელური ჰოსტების მდგომარეობასა და კონფიგურაციაზე, როგორიცაა მარშრუტიზატორები და კომუტატორები, ასევე კლიენტი კომპიუტერების ქსელი. **SNMP** მმართველმა შეიძლება შეაგროვოს **SNMP** აგენტები მონაცემებისათვის, ან მონაცემები შეიძლება ავტომატურად იქნას გაგზავნილი **SNMP** მმართველთან, **SNMP** აგენტებზე **trap**-ების კონფიგურაციით.

ამ ლაბორატორიულ დავალებაში თქვენ გადმოიწერთ, დააინსტალირებთ და დააკონფიგურებთ **SNMP** მართვის პროგრამულ უზრუნველყოფას **PC-A**-ზე. თქვენ ასევე დააკონფიგურებთ **Cisco** მარშრუტიზატორებს და **Cisco** კომუტატორებს, როგორც **SNMP** აგენტებს. **SNMP** აგენტიდან მოსული **SNMP** შეტყობინების დაჭერის შემდეგ, თქვენ უნდა მოახდენთ **MIB/Object ID** კოდების კონვერტაციას, შეტყობინების დეტალების შესასწავლად **Cisco SNMP Object Navigator**-ის გამოყენებით.

**შენიშვნა:** მარშრუტიზატორები, რომლებიც გამოიყენება **CCNA**-ს პრაქტიკული სამუშაოებისთვის, არის **Cisco 1941** ინტეგრირებული სერვისების მარშრუტიზატორები **(ISRs) Cisco IOS Release 15.2(4)M3 (universalk9 image)** ვერსიასთან ერთად. გამოყენებული კომუტატორები არის **Cisco Catalyst 2960s** ვერსია, **Cisco IOS Release 15.0(2) (lanbasek9 image)** ოპერაციული სისტემით. შესაძლოა გამოყენებულ იქნას სხვა მარშრუტიზატორები, კომუტატორები და **Cisco IOS** ვერსიებიც. მოდელისა და **Cisco IOS** ვერსიის მიხედვით ხელმისაწვდომი ბრძანებები და მიღებული შედეგები შეიძლება იყოს განსხვავებული იმისგან, რაც ნაჩვენებია ამ ლაბორატორიულ სამუშაოში. მიაქციეთ ყურადღება მარშრუტიზატორის ინტერფეისის შემაჯამებელ ცხრილს ამ დავალების ბოლოში, სწორი ინტერფეისის იდენტიფიკატორებისათვის.

**შენიშვნა:** დარწმუნდით, რომ მარშრუტიზატორები და კომუტატორები წაშლილია და არ აქვთ საწყისი კონფიგურაციები. თუ არ ხართ დარწმუნებული დაუკავშირდით თქვენს ინსტრუქტორს.

შენიშვნა: **snmp-server** ბრძანებები ამ ლაბორატორიულ დავალებაში გამოიწვევს **Cisco 2960** კომუტატორზე გამაფრთხილებელი შეტყობინების გაშვებას, კონფიგურაციის ფაილის **NVRAM**-ში შენახვის დროს. გამაფრთხილებელი შეტყობინების თავიდან ასაცილებლად შეამოწმეთ კომუტატორი იყენებს თუ არა **lanbase-routing** შაბლონს. **IOS** შაბლონი იმართება კომუტატორის მონაცემთა ბაზის მმართველის (**Switch Database Manager - SDM**)-ის მიერ. სასურველი შაბლონის შეცვლის შემდეგ ახალი შაბლონი გამოყენებული იქნება გადატვირთვის შემდეგ, მაშინაც კი თუ კონფიგურაცია არ არის შენახული.

S1# **show sdm prefer**

გამოიყენეთ ქვემოთ მოცემული ბრძანებები lanbase-routing შაბლონის ნაგულისხმევ SDM შაბლონად მითითებისთვის.

S1# **configure terminal**

S1 (config) # **sdm prefer lanbase-routing**

S1 (config) # **end**

S1 # **reload**

**მოთხოვნილი რესურსები:**

- ორი მარშრუტიზატორი (**Cisco 1941 Cisco IOS Release 15.2(4)M3** უნივერსალი იმიჯით ან მსგავსით)
- ერთი კომუტატორი (**Cisco 2960 Cisco IOS Release 15.0(2) lanbasek9** იმიჯით ან მსგავსით)
- ერთი პერსონალური კომპიუტერი (**Windows** ოპერაციული სისტემა ტერმინალის ემულაციის პროგრამასთან ერთად, როგორიცაა **Tera Term**)

- ერთი პერსონალური კომპიუტერი (**Windows** ოპერაციული სისტემა ინტერნეტთან წვდომით)
- კონსოლის კაბელები **Cisco IOS** მოწყობილობების კონსოლის პორტებით კონფიგურაციისათვის.
- **Ethernet** და სერიალური კაბელები, როგორც ნაჩვენებია ტოპოლოგიაზე
- **SNMP** მართვის პროგრამული უზრუნველყოფა (**PowerSNMP** უფასო მმართველი **Dart Communication**-სგან, ან **SolarWinds Kiwi Syslog Server**-ის 30 დღიანი საცდელი ვერსია).

**ნაწილი №1: ქსელის აწყობა და მოწყობილობის ბაზისური პარამეტრების კონფიგურაცია**

ამ ნაწილში თქვენ მომართავთ ქსელის ტოპოლოგიას და დააკონფიგურებთ მოწყობილობას ბაზისური პარამეტრებით.

**პირველი ეტაპი: ქსელის კაბელებით დაკავშირება, როგორც ნაჩვენებია ტოპოლოგიაზე.**

**მეორე ეტაპი: PC ჰოსტის კონფიგურაცია**

**მესამე ეტაპი: მარშრუტიზატორებისა და კომუტატორის ინიციალიზაცია და ხელახლა ჩატვირთვა აუცილებლობის შემთხვევაში**

**მეოთხე ეტაპი: ბაზისური პარამეტრების კონფიგურაცია მარშრუტიზატორებისა და კომუტატორისთვის.**

ა. გათიშეთ **DNS lookup**

ბ. მომართეთ მოწყობილობების სახელები ისე როგორც ნაჩვენებია ტოპოლოგიაზე

გ. დააკონფიგურეთ **IP** მისამართები მისამართების ცხრილის მიხედვით. (ჯერჯერობით არ დააკონფიგუროთ S0/0/0 ინტერფეისი R1 მარშრუტიზატორზე.)

დ. დააყენეთ **cisco** კონსოლისა და **vty**-ის პაროლად და ჩართეთ შესვლა (**login**).

ე. პრივილეგირებული **EXEC** რეჟიმის შიფრირებულ პაროლად დააყენეთ **class**

ვ. დააკონფიგურეთ **logging Synchronous**, რათა აკრძალულ იქნას კონსოლის შეტყობინებები წყვეტის ბრძანებების ჩანაწერიდან.

ზ. **Ping** ბრძანების გაშვებით შეამოწმეთ **LAN** მოწყობილობებს შორის წარმატებული კავშირი.

თ. გადაიტანეთ გაშვებული კონფიგურაციის ასლი საწყის კონფიგურაციაში.

## ნაწილი №2: SNMP მმართველისა და აგენტების კონფიგურაცია

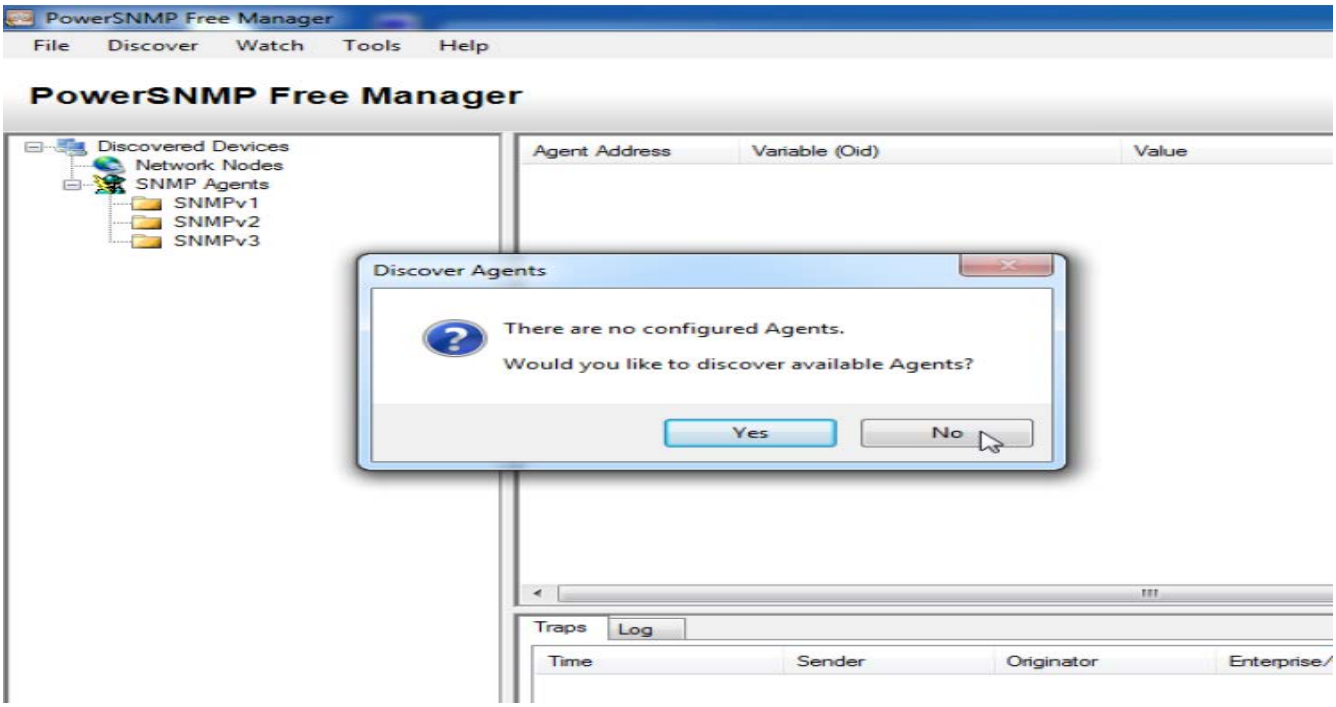
მეორე ნაწილში, **PC-A**-ზე მოხდება **SNMP** მართვის პროგრამული უზრუნველყოფის ინსტალაცია და კონფიგურაცია, ასევე **R1** და **S1** იქნება დაკონფიგურებული, როგორც **SNMP** აგენტები.

### პირველი ეტაპი: SNMP მართვის პროგრამის ინსტალაცია

ა. გადმოწერეთ და დააინსტალირეთ **Dart Communications**-ის მიერ გამოშვებული **PowerSNMP Free Manager** პროგრამა ქვემოთ მოცემული მისამართიდან: http://www.dart.com/snmp-free-manager.aspx.

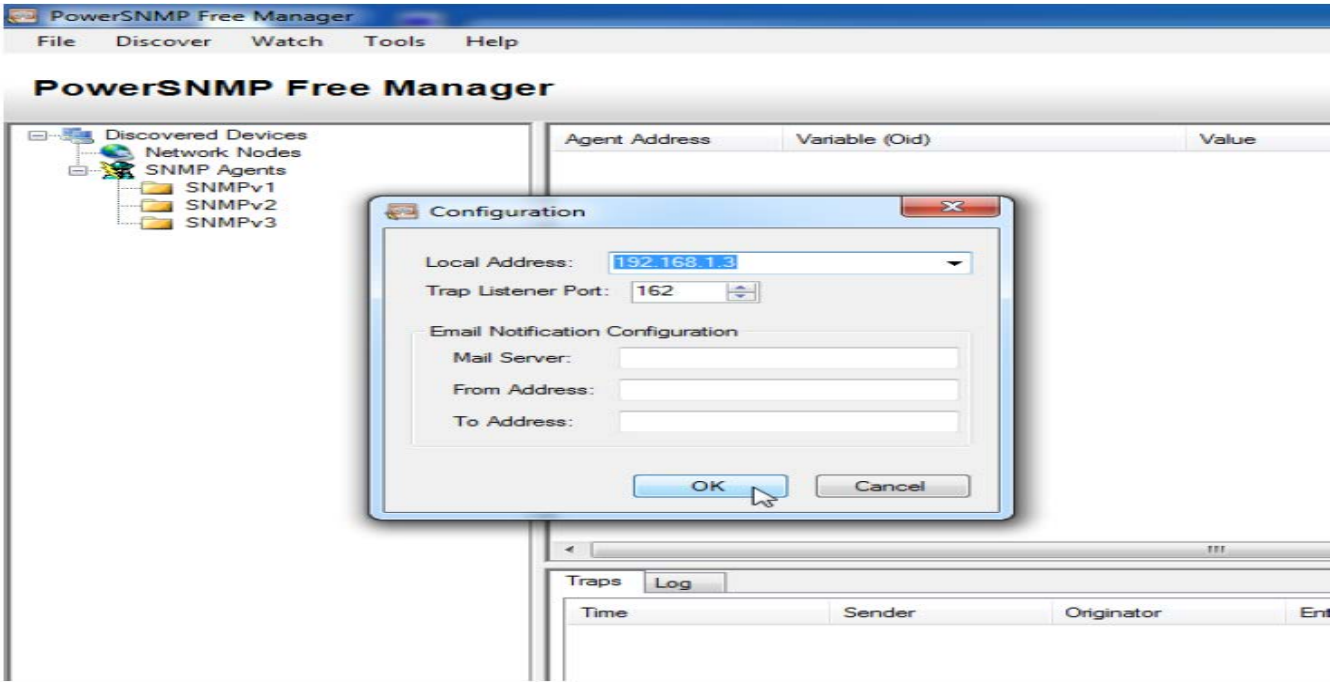
ბ. გაუშვით **PowerSNMP Free Manager** პროგრამა.

გ. დააჭირეთ **No** ღილაკს თუ შემოთავაზებულ იქნა ხელმისაწვდომი **SNMP** აგენტების აღმოჩენა. თქვენ აღმოაჩენთ **SNMP** აგენტებს, **R1** მარშრუტიზატორზე **SNMP**-ს კონფიგურაციის შემდეგ. **PowerSNMP Free Manager** მხარს უჭერს **SNMP**-ს 1, 2 და 3 ვერსიებს. მოცემულ ლაბორატორიულ სამუშაოში გამოყენებულია **SNMPv2**.




*სურ.5.1.1.1. 2*

დ. გამოტანილ კონფიგურაციის ფანჯარაში (თუ ფანჯარა არ გამოჩნდა, მაშინ გადადით **Tools** განყოფილებაში და აირჩიეთ **configuration** ბრძანება), მომართეთ ლოკალური **IP** მისამართი **192.168.1.3**, მოსმენისათვის და დააჭირეთ **OK** ღილაკს.




*სურ.5.1.1.1. 3*

**შენიშვნა:** თუ შემოთავაზებულ იქნა ხელმისაწვდომი **SNMP** აგენტების აღმოჩენა, დააჭირეთ **No** ღილაკს და გადადით ამ დავალების შემტებ ეტაპზე.

მეორე ეტაპი: **SNMP** აგენტის კონფიგურაცია

ა. **R1** მარშრუტიზატორზე გლობალური კონფიგურაციის რეჟიმიდან შეიყვანეთ ქვემოთ მოცემული ბრძანებები, მარშრუტიზატორის როგორც **SNMP** აგენტის კონფიგურაციისათვის. პირველ სტრიქონზე **SNMP** რიგების ერთობა (**Community string**) არის **ciscolab**, მხოლოდ დათვალიერების პრივილეგიებით და **SNMP_ACL** სახელის მქონე წვდომის სია, რომელიც განსაზღვრავს თუ რომელი ჰოსტები არიან დაშვებული რომ მიიღონ **SNMP** ინფორმაცია **R1** მარშრუტიზატორიდან. მეორე და მესამე სტრიქონებზე **SNMP** მმართველის **location** და **contact** ბრძანებები იძლევიან აღწერილობით საკონტაქტო ინფორმაციას. მეოთხე სტრიქონი განსაზღვრავს ჰოსტის **IP** მისამართს, რომელიც მიიღებს **SNMP** შეტყობინებებს, **SNMP** ვერსიას და რიგების ერთობას (**Community string**). მეხუთე სტრიქონი რთავს ყველა ნაგულისხმევ **SNMP trap**-ს, ხოლო მე-6 და მე-7 სტრიქონები ქმნიან დასახელებულ წვდომის სიას, რათა აკონტროლოს თუ რომელი ჰოსტები არიან დაშვებული მარშრუტიზატორიდან **SNMP** ინფორმაციის მისაღებად.

R1 (config) # **snmp-server community ciscolab ro SNMP_ACL**

R1 (config) # **snmp-server location snmp_manager**

R1 (config) # **snmp-server contact ciscolab_admin**

R1 (config) # **snmp-server host 192.168.1.3 version 2c ciscolab**

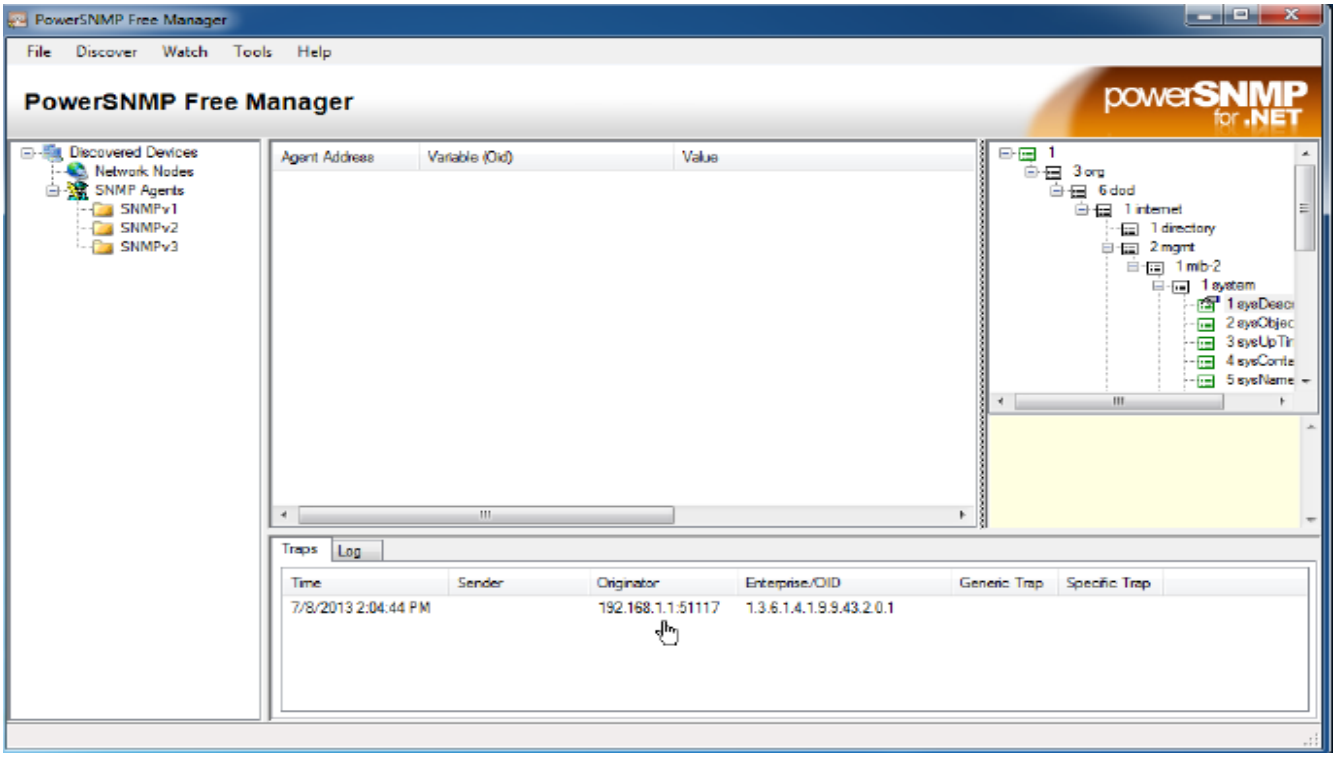
R1 (config) # **snmp-server enable traps**

R1 (config) # **ip access-list standard SNMP_ACL**

R1 (config-std-nacl) # **permit 192.168.1.3**

ბ. ამ ეტაპზე თქვენ შეიძლება გაფრთხილებულ იქნათ, რომ **PowerSNMP Free Manager** იღებს შეტყობინებებს **R1** მარშრუტიზატორიდან. თუ ასე არაა, მაშინ თქვენ

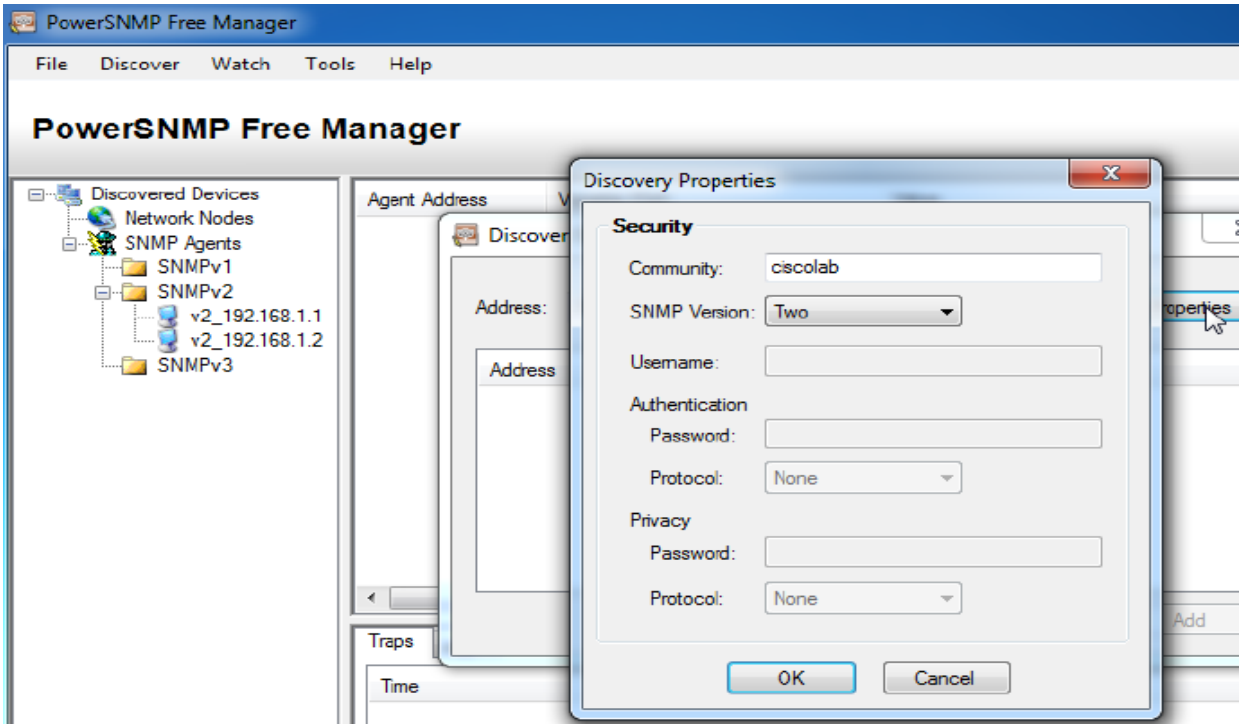
შეგიძლიათ აიძულოთ **SNMP** შეტყობინება, რომ გაგზავნილ იქნეს **copy run start** ბრძანების შეყვანით **R1** მარშრუტიზატორზე. წარუმატებლობის შემთხვევაში გადადით შემდეგ ეტაპზე.



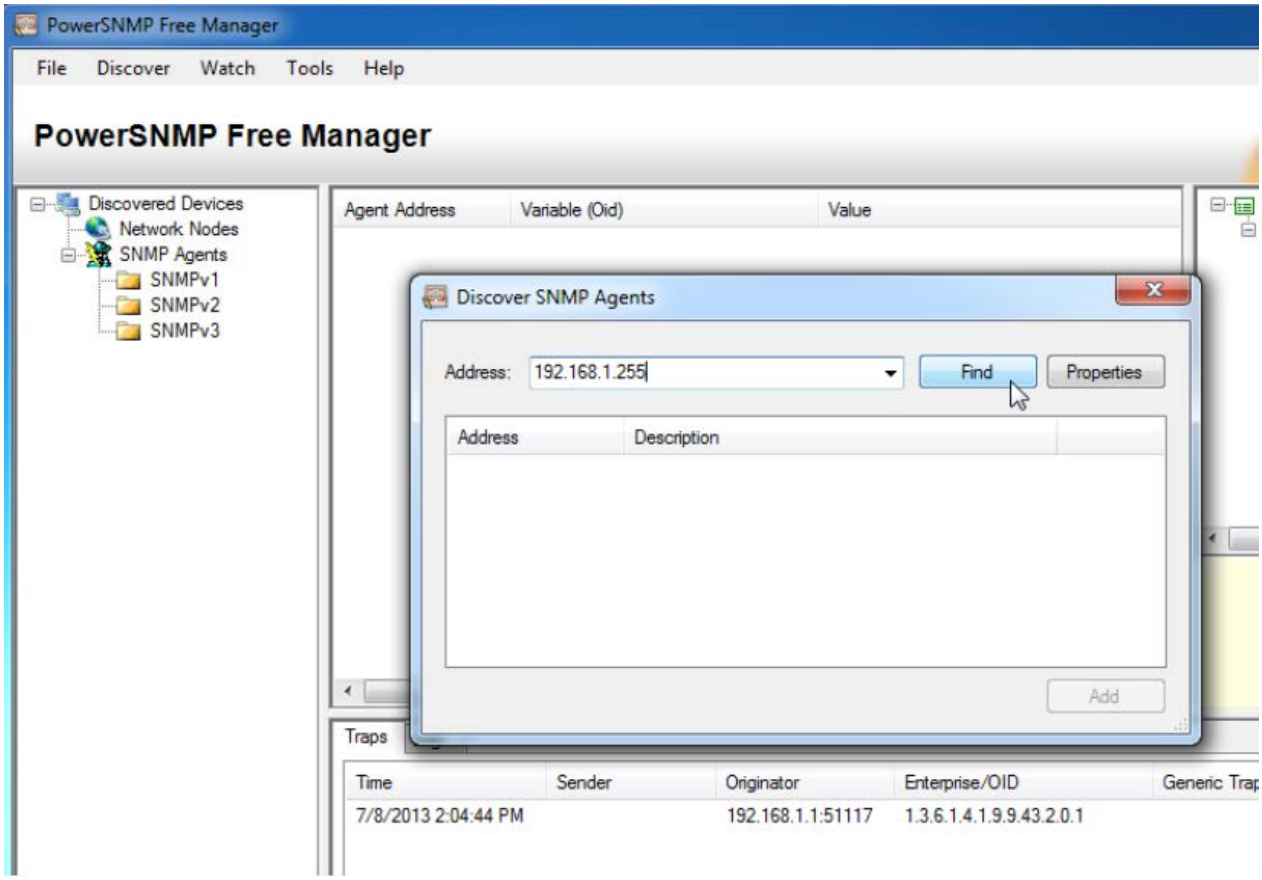

*სურ.5.1.1.1. 4*

**მესამე ეტაპი: SNMP აგენტის აღმოჩენა**

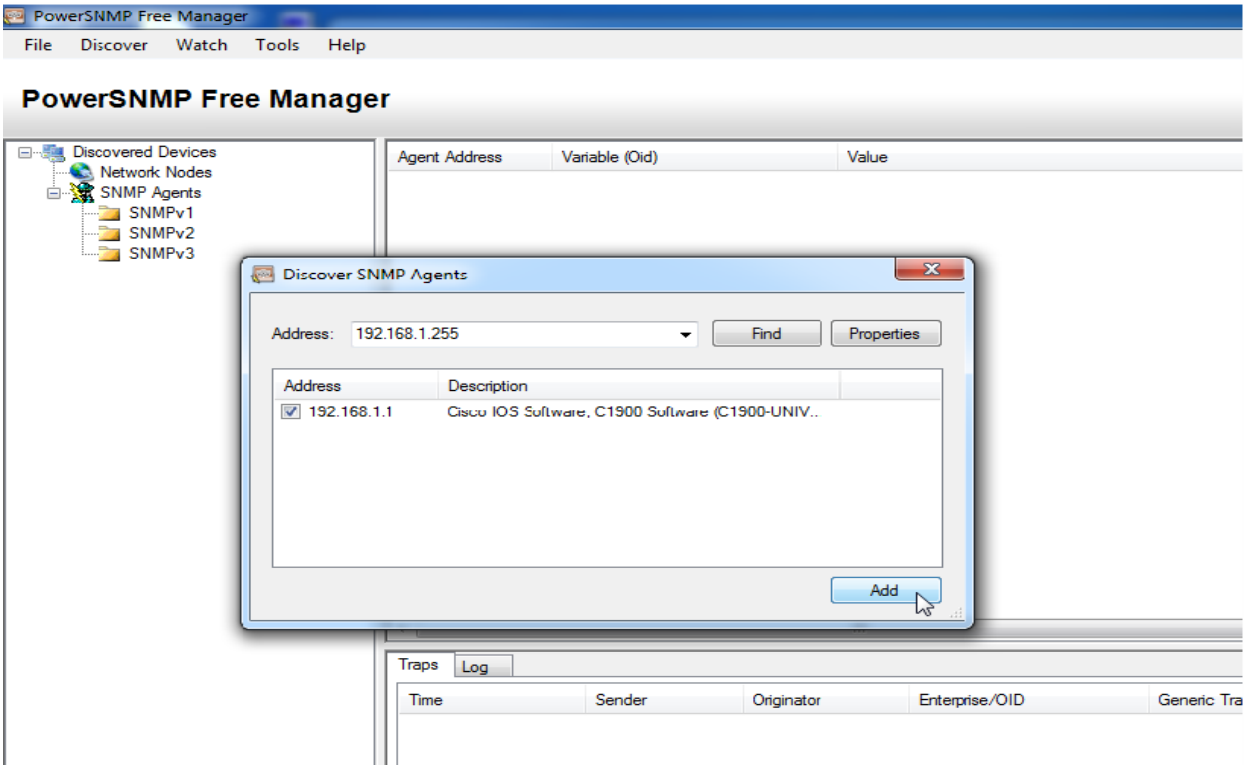
ა. **PC-A**-ზე დაყენებული **PowerSNMP Free Manager** პროგრამიდან, გახსენით **Discover** > **SNMP Agents** ფანჯარა. შეიყვანეთ **IP** მისამართი **192.168.1.255**. იგივე ფანჯარაში დააჭირეთ Properties ღილაკს და მომართეთ **ciscolab community** და **SNMP** ვერსია **ორი**, შემდეგ დააჭირეთ **OK** ღილაკს. ახლა თქვენ შეგიძლიათ დააწვეთ **Find** ღილაკს, ყველა **SNMP** აგენტის აღმოსაჩენად **192.168.1.0** ქსელში. **PowerSNMP Free Manager**-მა შეიძლება იპოვოს **R1** მარშრუტიზატორი **192.168.1.1**-ზე. დააწექით მონიშვნას და შემდეგ **Add** ღილაკს, **R1** მარშრუტიზატორის როგორც **SNMP** აგენტის დასამატებლად.




*სურ.5.1.1.1. 5*

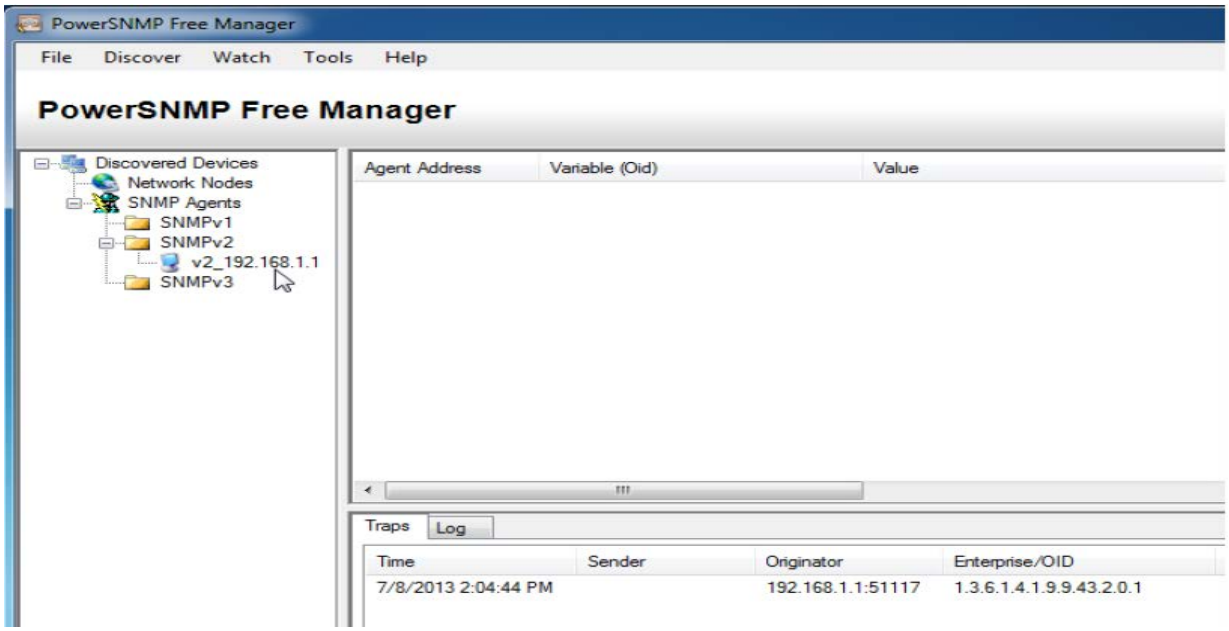



*სურ.5.1.1.1. 6*




*სურ.5.1.1.1. 7*

ბ. **PowerSNMP Free Manager**-ში, **R1** მარშრუტიზატორი დაემატა **SNMPv2** ხელმისაწვდომი აგენტების სიაში.




*სურ.5.1.1.1. 8*

გ. დააკონფიგურეთ **S1**, როგორც **SNMP** აგენტი. თქვენ შეგიძლიათ გამოიყენოთ იგივე **snmp-server** ბრძანებები, რომლებიც გამოიყენეთ **R1** მარშრუტიზატორის კონფიგურაციისას.

დ. **S1**-ის კონფიგურაციის შემდეგ, **SNMP** შეტყობინებები **192.168.1.2**-დან ნაჩვენებია **PowerSNMP Free Manager** პროგრამის **trap**-ების ფანჯარაში. **PowerSNMP Free Manager**-ში დაამატეთ **S1**, როგორც **SNMP** აგენტი იგივე პროცესის გამოყენებით, რომელიც გამოიყენეთ **R1**-ის აღმოჩენის დროს.

**ნაწილი №3: OID კოდების კონვერტაცია Cisco SNMP Object Navigator-ის საშუალებით**

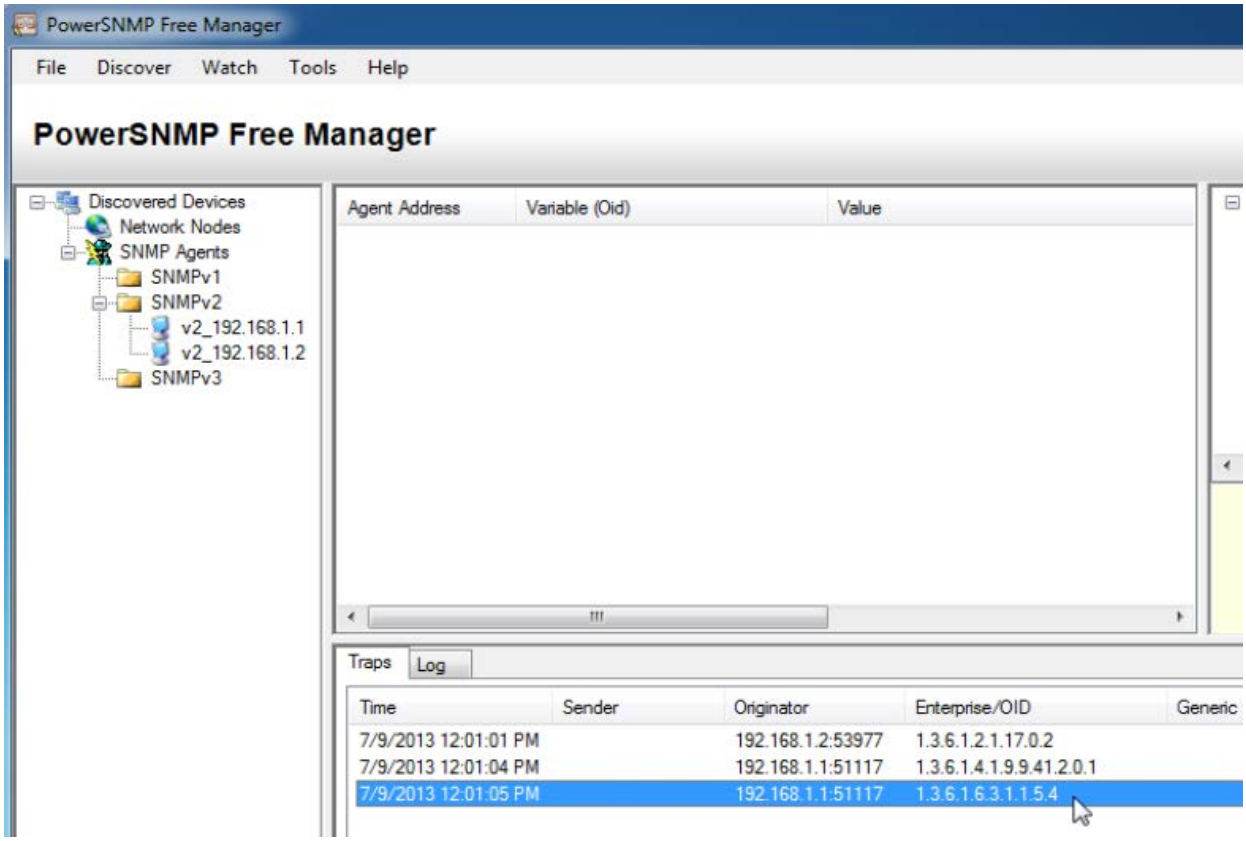
მესამე ნაწილში თქვენ აიძულებთ **SNMP** შეტყობინებებს რომ გაიგზავნონ **SNMP** მმართველზე, რომელიც განთავსებულია **PC-A**-ზე. შემდეგ თქვენ უნდა მოახდინოთ მიღებული **OID** კოდების კონვერტაცია სახელებით, რათა შესწავლილ იქნას შეტყობინებების „ბუნება“. **MIB/OID** კოდები შეიძლება მარტივად დაკონვერტირდეს **Cisco SNMP Object Navigator**-ის გამოყენებით, რომელიც განთავსებულია http://www.cisco.com -ზე

**პირველი ეტაპი: მიმდინარე SNMP შეტყობინებების წაშლა.**

**PowerSNMP Free Manager** პროგრამაში, მარჯვენა ღილაკით დააჭირეთ **Traps** ფანჯარას და აირჩიეთ **Clear** ბრძანება, **SNMP** შეტყობინებების წასაშლელად.

**მეორე ეტაპი: SNMP trap-სა და შეტყობინების შექმნა.**

**R1** მარშრუტიზატორზე დააკონფიგურეთ **S0/0/0** ინტერფეისი ლაბორატორიული სამუშაოს თავში მოცემული მისამართების ცხრილის მიხედვით. განახორციელეთ წვდომა გლობალური კონფიგურაციის რეჟიმში და **SNMP trap** შეტყობინებების შექმნისათვის ჩართეთ ინტერფეისი, რათა გაგზავნილ იქნას ისინი **SNMP** მმართველთან **PC-A**-ზე. გაითვალისწინეთ **Enterprise/OID** კოდური ნომრები, რომლის ნახვაც შესაძლებელია **traps** ფანჯარაში.




*სურ.5.1.1.1. 9*

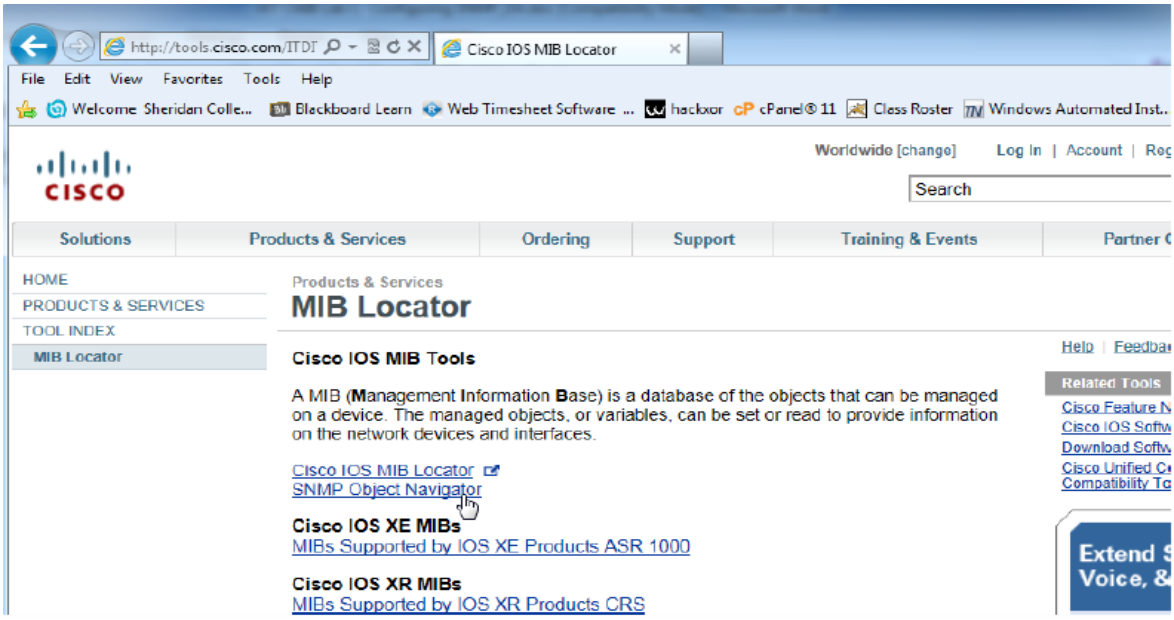
**მესამე ეტაპი: SNMP MIB/OID შეტყობინებების დეკოდირება**

ინტერნეტიანი კომპიუტერიდან გახსენით ბრაუზერი და გადადით http://www.cisco.com ვებ-გვერდზე.

ა. ფანჯრის ზედა ნაწილში არსებული ძებნის უტილიტის დახმარებით, მოძებნეთ **SNMP Object Navigarot**-ი.

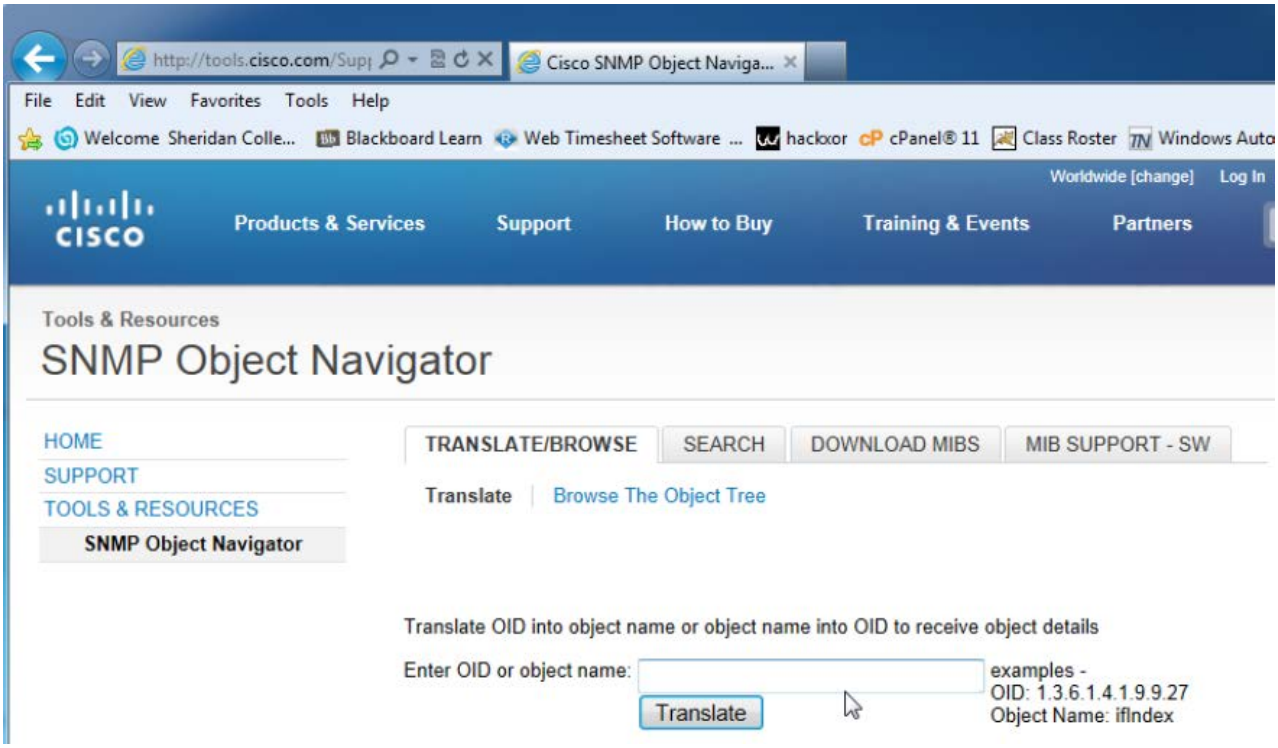
ბ. მიღებული შედეგიდან აირჩიეთ **SNMP Object Navigator MIB Download MIBs OID OIDs**.

გ. გადადით **MIB Locator** გვერდზე. დააჭირეთ **SNMP Object Navigrator** ბმულს




*სურ.5.1.1.1. 10*

დ. **SNMP Object Navigator** გვერდის გამოყენებით, მოახდინეთ **OID** კოდური ნომრების დეკოდირება **PowerSNMP Free Manager**-დან, რომელიც შექმნილია მესამე ნაწილის მეორე ეტაპზე. შეიყვანეთ **OID** კოდური ნომერი და დააწექით **Translate** ღილაკს.




*სურ.5.1.1.1. 11*

ე. ჩაწერეთ ქვემოთ მოცემულ ველებში **OID** კოდური ნომრები და მათი შესაბამისი შეტყობინების თარგმანი.

_____

_____

_____

**ასახვა (Reflection)**

1. რა პოტენციური უპირატესობები გააჩნია ქსელის **SNMP**-თი მონიტორინგს? _____

_____

_____

2. რატომ არის სასურველი მხოლოდ წაკითხვის წვდომის რეჟიმის გამოყენება **SNMPv2**-თან მუშაობის დროს? _____

_____

_____

**მარშრუტიზატორის ინტერფეისის შემაჯამებელი ცხრილი:**

| მარშრუტიზატორის ინტერფეისის შეჯამება | | | | |
|---|---|---|---|---|
| მარშრუტიზატორის მოდელი | Ethernet interface №1 | Ethernet interface №2 | Serial interface №1 | Serial interface №2 |
| 1800 | FastEthernet 0/0 (F0/0) | FastEthernet 0/1 (F0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |
| 1900 | GigabitEthernet 0/0 (G0/0) | GigabitEthernet 0/1 (G0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |
| 2801 | FastEthernet 0/0 (F0/0) | FastEthernet 0/1 (F0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |
| 2811 | FastEthernet 0/0 (F0/0) | FastEthernet 0/1 (F0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |
| 2900 | GigabitEthernet 0/0 (G0/0) | GigabitEthernet 0/1 (G0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |

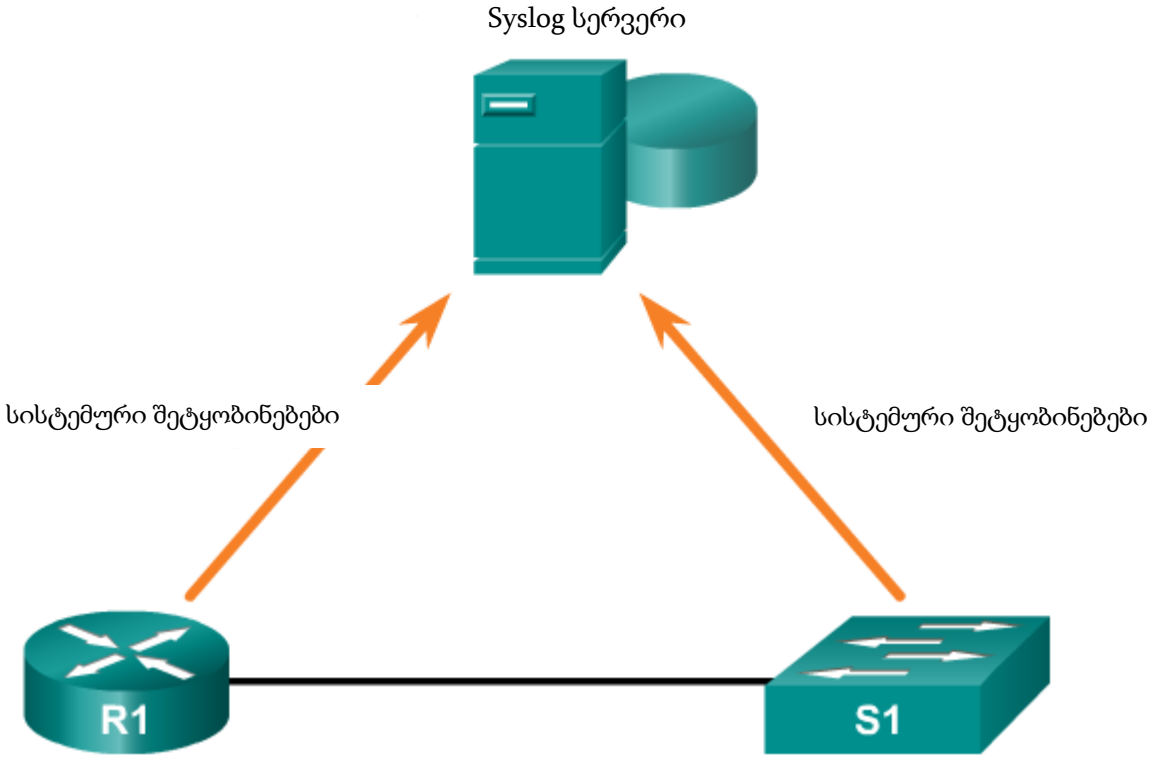
**შენიშვნა:** თუ გინდათ მოძებნოთ თუ როგორაა მარშრუტიზატორი კონფიგურირებული, დაათვალიერეთ ინტერფეისები რათა მოახდინოთ მარშრუტიზატორის იდენტიფიკაცია და დაადგინოთ რამდენი ინტერფეისი აქვს მას. აქ არ არის იმის საშუალება, რომ ჩამოიწეროს თითოეული მარშრუტიზატორის კონფიგურაციების კომბინაციები. მოცემული ცხრილი მოიცავს მოწყობილობის **Ethernet** და **Serial** ინტერფეისების შესაძლო კომბინაციების იდენტიფიკატორებს. ეს ცხრილი არ მოიცავს ნებისმიერი სხვა ტიპის ინტერფეისს, რადგან კონკრეტული მარშრუტიზატორი შეიძლება შეიცავდეს მხოლოდ ერთს. ამის მაგალითი შეიძლება იყოს **ISDN BRI** ინტერფეისი. ფრჩხილებში მოცემული ცვლადი არის ლეგალური აბრევიატურა, რომელიც შეიძლება გამოყენებულ იქნას **Cisco IOS** ბრძანებებში, ინტერფეისის წარმოსადგენად.

## 5.2. Syslog, NTP, Netflow პროტოკოლის კონფიგურირება.

### *5.2.1. Syslog-ის გაცნობა*

როდესაც რაიმე მოვლენა ხდება ქსელში, ქსელის მოწყობილობებს აქვთ სანდო მექანიზმი ადმინისტრატორისთვის დეტალური სისტემური შეტყობინებების გასაცნობად. ეს შეტყობინებები შეიძლება იყოს ან არაკრიტიკული ან მნიშვნელოვანი. ქსელის ადმინისტრატორებს აქვთ შენახვის, განმარტების და ჩვენების შეტყობინებების სხვადასხვა ვარიანტები, ასევე მიმდინარეობს გაფრთხილება იმ შეტყობინებებით, რომელთაც შეიძლება დიდი გავლენა იქონიონ ქსელის ინფრასტრუქტურაზე.

ყველაზე გავრცელებული მეთოდი, იმ სისტემურ შეტყობინებებზე წვდომისთვის, რომელთაც იძლევიან ქსელის მოწყობილობები, არის Syslog პროტოკოლის გამოყენება.




სურ. 5.2.1. Syslog

Syslog არის ტერმინი, რომელიც გამოიყენება სტანდარტის აღსაწერად. ის ასევე გამოყენებულია იმ პროტოკოლის აღსაწერად, რომელიც განვითარდა ამ სტანდარტისთვის. Syslog პროტოკოლი განვითარდა UNIX სისტემებისთვის 1980 წელში, მაგრამ პირველად მისი

დოკუმენტირება IETF-ის მიერ, როგორც RFC 3164, 2001 წელს. Syslog იყენებს UDP პორტის ნომერს 514 მოვლენის შესახებ შეტყობინების გასაგზავნად მთელს IP ქსელებში, მოვლენების შეტყობინების შემგროვებლისათვის, ისე როგორც ნაჩვენებია 5.2.1 სურათზე:

ბევრი ქსელური მოწყობილობა უჭერს მხარს Syslog-ს, მარშრუტიზატორების, კომუტატორების, აპლიკაციების სერვერების, ფაიერვოლების და სხვა ქსელური მოწყობილობების ჩათვლით. Syslog პროტოკოლი საშუალებას აძლევს ქსელურ მოწყობილობებს გააგზავნონ თავისი სისტემური შეტყობინებები მთელს ქსელში, Syslog სერვერებისთვის. შესაძლებელია სპეციალური out-of-band (OOB) ქსელის აწყობა ამ მიზნებისთვის.

არსებობს რამდენიმე განსხვავებული Syslog სერვერის პროგრამული უზრუნველყოფის პაკეტები Windows და UNIX სისტემებისათვის. ბევრი მათგანი არის უფასო.

Syslog აღრიცხვის ჟურნალის სერვისი უზრუნველყოფს სამ მთავარ ფუნქციას:

- ჩანაწერების ინფორმაციის შეგროვების შესაძლებლობა მონიტორინგისა და პრობლემის მოძებნისთვის;
- რეგისტრირებული ჩანაწერების ინფორმაციიდან ამორჩევის შესაძლებლობა;
- რეგისტრირებული Syslog შეტყობინებების ადრესატების განსაზღვრის შესაძლებლობა.

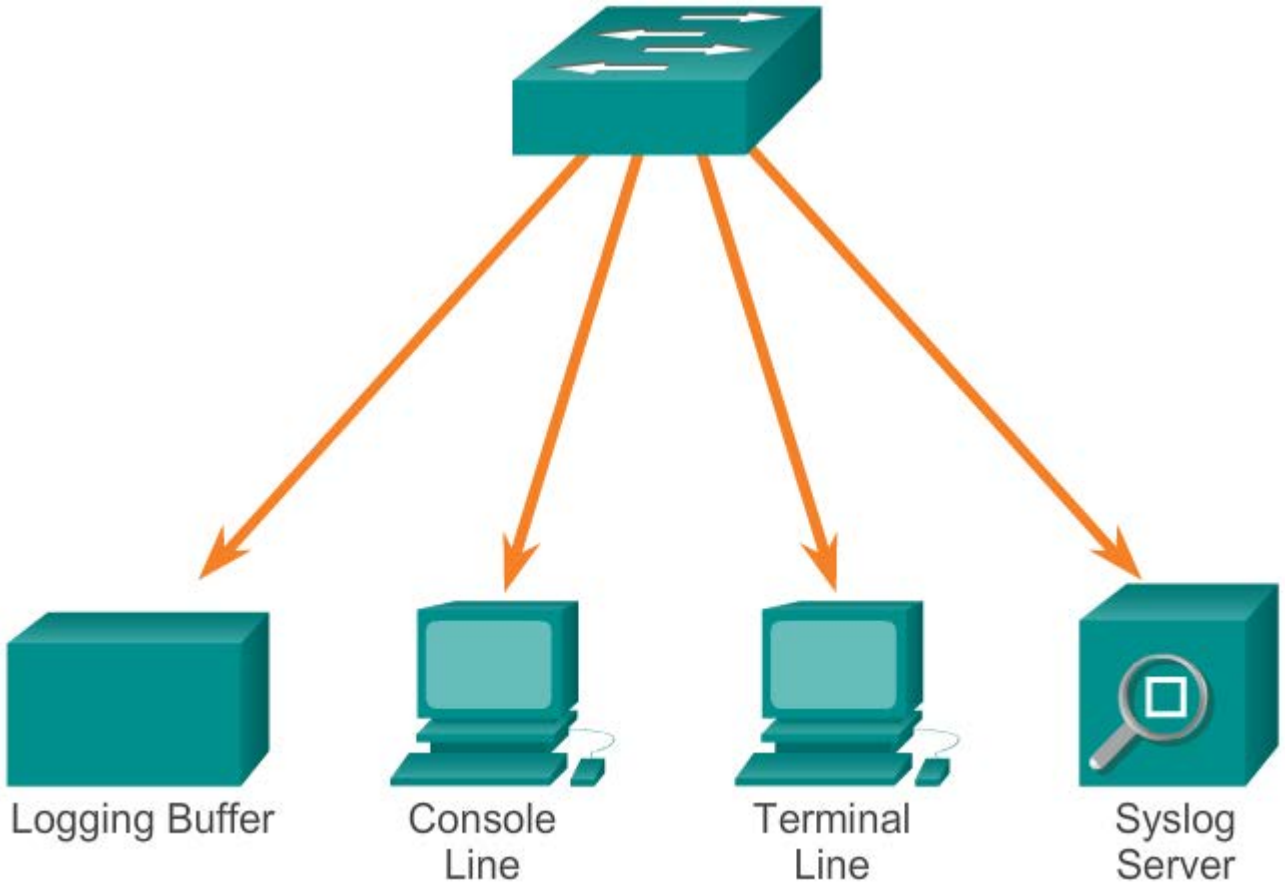
### *5.2.2. Syslog ოპერაცია*

Cisco ქსელურ მოწყობილობებზე Syslog პროტოკოლი იწყება სისტემური შეტყობინებების გაგზავნით და debug შედეგით ლოკალური რეგისტრაციის პროცესიდან, მოწყობილობის შიგნით. თუ როგორ მართავს ლოგირების პროცესი ამ შეტყობინებებს და შედეგებს დამოკიდებულია მოწყობილობის კონფიგურაციაზე. მაგალითად, Syslog შეტყობინებები შეიძლება გაგზავნილ იქნას მთელი ქსელის მასშტაბით, გარე Syslog სერვერზე. ეს შეტყობინებები შეიძლება მიღებულ იქნას მოქმედ მოწყობილობასთან წვდომის საჭიროების გარეშე. აღრიცხვის (Log) შეტყობინებები და შედეგები, რომლებიც

ინახება გარე სერვერზე, შეიძლება გადმოტანილ იქნას სხვადასხვა ანგარიშებში, უფრო მარტივად წაკითხვისთვის.

ალტერნატიულად, Syslog შეტყობინებები შეიძლება გაგზავნილ იქნას შიდა ბუფერში. შიდა ბუფერში გაგზავნილი შეტყობინებების ნახვა შესაძლებელია მხოლოდ მოწყობილობის CLI-დან.

ბოლოს, ქსელის ადმინისტრატორს შეუძლია მიუთითოს, რომ მხოლოდ განსაზღვრული ტიპის სისტემური შეტყობინებები იგზავნება დანიშნულების სხვადასხვა პუნქტებში. მაგალითად, მოწყობილობა შეიძლება იყოს კონფიგურირებული ისე, რომ გადააგზავნოს ყველა სისტემური შეტყობინება გარე syslog სერვერზე. თუმცა, გამართვის დონის (Debug-level) შეტყობინებები იგზავნება შიდა ბუფერში და ადმინისტრატორს მასთან წვდომა შეუძლია მხოლოდ CLI-ით.




სურ. 5.2.2. Syslog შეტყობინების დანიშნულების ადგილის ვარიანტები

როგორც 5.2.2 სურათზეა მოცემული, Syslog შეტყობინებების პოპულარული დანიშნულების ადგილები მოიცავს:

- აღრიცხვის ბუფერი (მარშრუტიზატორის ან კომუტატორის ოპერატიული მეხსიერების შიდა ნაწილი)
- კონსოლის ხაზი
- ტერმინალის ხაზი
- Syslog სერვერი

შესაძლებელია სისტემური შეტყობინებების დაშორებულად მონიტორინგი, ლოგების დათვალიერებით Syslog სერვერზე, ან მოწყობილობასთან წვდომით Telnet-ის, SSH-ის ან კონსოლის პორტის გამოყენებით.

### *5.2.3. Syslog-ის შეტყობინების ფორმატი*

Cisco-ს მოწყობილობები იძლევა Syslog შეტყობინებებს ქსელის რაიმე მოვლენის მიხედვით. თითოეული Syslog შეტყობინება შეიცავს  მკაცრი წესების დონეს და შესაძლებლობებს.

| Severity Name | Severity Level | Explanation |
|---|---|---|
| Emergency | Level 0 | System Unusable |
| Alert | Level 1 | Immediate Action Needed |
| Critical | Level 2 | Critical Condition |
| Error | Level 3 | Error Condition |
| Warning | Level 4 | Warning Condition |
| Notification | Level 5 | Normal, but Significant Condition |
| Informational | Level 6 | Informational Message |
| Debugging | Level 7 | Debugging Message |

სურ. 5.2.3.1 Syslog  დონეები

მცირე ნომრიანი დონე არის მეტად კრიტიკული Syslog-ის განგაშის სიგნალი. შეტყობინებების მკაცრი წესების დონე შეიძლება დაყენებულ იქნას კონტროლისთვის, თუ სად უნდა გამოჩნდეს თითოეული ტიპის შეტყობინება (ანუ კონსოლზე თუ სხვა დანიშნულების ადგილზე). Syslog დონეების სრული სია მოცემულია 5.2.3.1 სურათზე.

Syslog-ის თითოეულ დონეს აქვს თავისი მნიშვნელობა:

- გაფრთხილების დონე (Warning Level) - გადაუდებელი დონე (Emergency Level) - მოცემული შეტყობინებები არის შეცდომის შეტყობინებები პროგრამული ან ტექნიკური უზრუნველყოფის ცუდათ ფუნქციონირების შესახებ; ამ ტიპის შეტყობინებები ნიშნავს იმას, რომ მოწყობილობა არის დაზიანებული. პრობლემის სირთულე განსაზღვრავს გამოყენებული Syslog-ის დონეს.
- გამართვის დონე (Debugging Level) - მოცემული დონე მიუთითებს, რომ შეტყობინებები შედეგი, რომელიც გენერირებულია სხვადასხვა გამომავალი debug ბრძანებებიდან.
- შეტყობინების დონე (Notification Level) - შეტყობინების დონე არის მხოლოდ ინფორმაციისთვის, მოწყობილობის ფუნქციონირება არ არის დაზიანებული. ინტერფეისის up-ში ან down-ში გადასვლები, სისტემის გადატვირთვის შეტყობინებები ჩნდება შეტყობინების დონეზე.

დამატებით, სიმკაცრის (Severity) მისათითებლად, Syslog შეტყობინებები ასევე შეიცავს ინფორმაციას შესაძლებლობებზე. Syslog-ის შესაძლებლობები არის სერვისის იდენტიფიკატორები, რომელიც ამოიცნობს და კატეგორიებად ყოფს სისტემის მდგომარეობის მონაცემებს შეცდომებისა და მოვლენების შეტყობინების ანგარიშებისთვის. ხელმისაწვდომი ლოგირების შესაძლებლობის პარამეტრები ინდივიდუალურია ქსელური მოწყობილობისათვის. მაგალითად, Cisco 2960 კომუტატორზე, რომელზეც გაშვებულია Cisco IOS Release 15.0(2) და Cisco 1941 მარშრუტიზატორზე, რომელზეც გაშვებულია Cisco IOS Release 15.2(4) სისტემა, მხარს უჭერენ 24 შესაძლებლობის პარამეტრს, რომელიც დაყოფილია კატეგორიებად 12 ტიპის შესაძლებლობის მიხედვით.

ზოგიერთი გავრცელებული Syslog შეტყობინების შესაძლებლობები, რომლებიც ამოღებულია Cisco IOS მარშრუტიზატორებიდან, მოიცავს:

- IP
- OSPF პროტოკოლს
- SYS ოპერაციულ სისტემას
- IP Security (IPsec)
- Interface IP (IF)

ნაგულისხმევად, syslog შეტყობინებების ფორმატი Cisco IOS პროგრამულ უზრუნველყოფაზე არის შემდეგნაირი: seq no: timestamp: %facility-severity-MNEMONIC: description.

ველები, რომლებსაც მოიცავს Cisco IOS პროგრამული უზრუნველყოფაში, Syslog შეტყობინება, აღწერილია 5.2.3.2 სურათზე:

| Field | Explanation |
|---|---|
| seq no | Stamps log messages with a sequence number only if the **service sequence-numbers** global configuration comman is configured. |
| timestamp | Date and time of the message or event, which appears only if the **service timestamps** global configuration command is configured. |
| facility | The facility to which the message refers. |
| severity | Single-digit code from 0 to 7 that is the severity of the message. |
| MNEMONIC | Text string that uniquely describes the message. |
| description | Text string containing detailed information about the event bein reported. |

სურ. 5.2.3.2. Syslog-ის შეტყობინების ფორმატი

მაგალითად, მარტივი შედეგი Cisco კომუტატორზე EtherChannel ხაზის მდგომარეობის შეცვლა up-ში, არის: 00:00:46: %LINK-3-UPDOWN: Interface Port-channell, changed state to up.

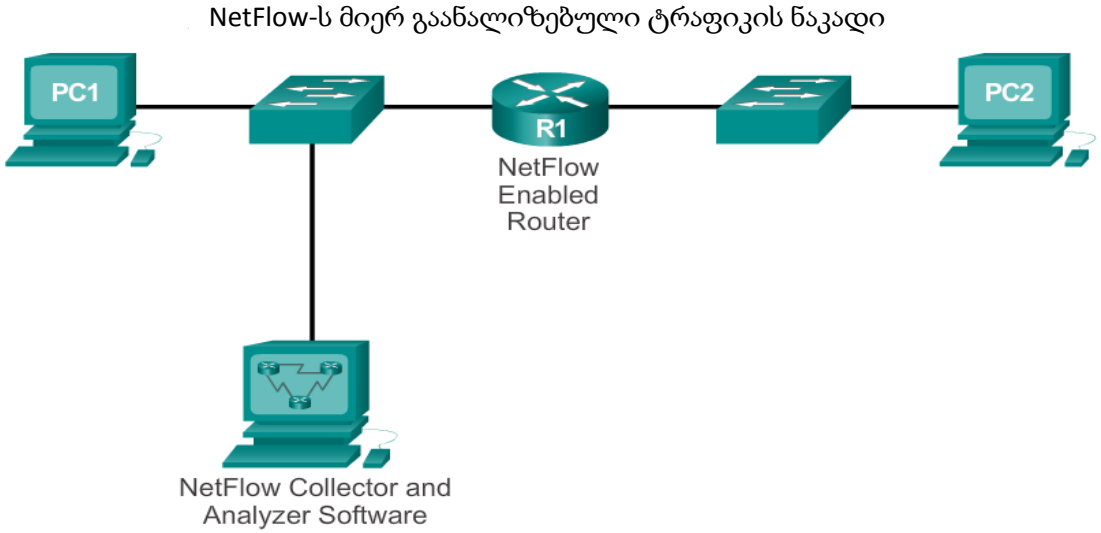
აქ შესაძლებლობა არის LINK და სიმკაცრის დონე არის 3, UPDOWN MNEMONIC-ით.

ყველაზე გავრცელებული შეტყობინებები არის ხაზის up და down მდგომარეობაში გადასვლის შეტყობინებები, და შეტყობინებები, რომლებიც მოწყობილობას გამოაქვს, როდესაც ხდება კონფიგურაციის რეჟიმიდან გამოსვლა. თუ ACL ლოგირება კონფიგურირებულია, მაშინ მოწყობილობა ახდენს Syslog შეტყობინებების გენერაციას, როცა პაკეტები ემთხვევა პარამეტრის პირობებს.

5.2.4. NetFlow-ს მიმოხილვა

NetFlow არის Cisco IOS ტექნოლოგია, რომელიც იძლევა სტატისტიკას პაკეტების ნაკადზე, Cisco მარშრუტიზატორის ან მრავალდონიანი კომუტატორის საშუალებით. NetFlow არის სტანდარტი IP ქსელებიდან IP ექსპლუატაციის მონაცემების შესაგროვებლად.

ისტორიულად, NetFlow ტექნოლოგია განვითარდა იმიტომ, რომ ქსელის პროფესიონალებს სჭირდებოდათ მარტივი და მოხერხებული მეთოდი ქსელში TCP/IP ნაკადების თვალყურის დევნების, და SNMP არ იყო საკმარისი ამ მოთხოვნებისთვის. მაშინ, როცა SNMP ცდილობს ქსელის მართვის მახასიათებლებისა და პარამეტრების ძალიან ფართო დიაპაზონის მოწოდებას, NetFlow ამ დროს ფოკუსირებულია IP პაკეტების ნაკადების სტატისტიკის მოწოდებაზე, ქსელური მოწყობილობების დახმარებით.




სურ. 5.2.4.1 NetFlow ქსელში

NetFlow გვაწვდის ინფორმაციას, რაც  იძლევა ქსელის და უსაფრთხოების მონიტორინგის, ქსელის დაგეგმვის, ტრაფიკის ანალიზის და IP ანგარიშის შესაძლებლობას. მაგალითად 5.2.4.1 სურათზე PC1 უკავშირდება PC2-ს აპლიკაციის დახმარებით, როგორიცაა HTTPS. NetFlow-ს შეუძლია ამ აპლიკაციის კავშირის მონიტორინგი, აპლიკაციის ინდივიდუალური ნაკადის ბაიტებისა და პაკეტების დათვლა. შემდეგ ის უშვებს სტატისტიკას გარე სერვერზე, რომელსაც NetFlow კოლექტორი ეწოდება.

NetFlow გახდა მონიტორინგის სტანდარტი და ახლა ის ფართოდაა მხარდაჭერილი ქსელურ ინდუსტრიაში.

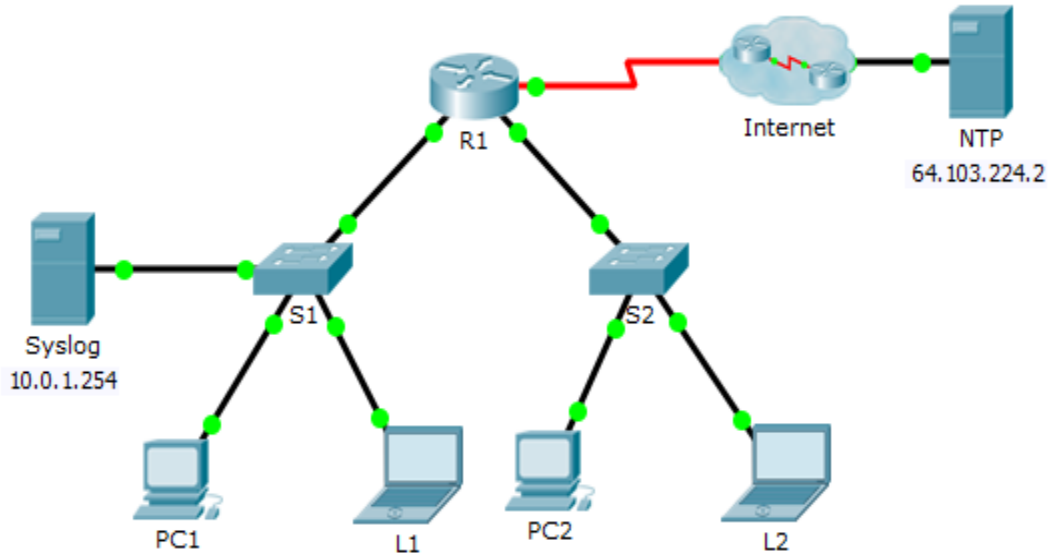
მოქნილი NetFlow არის NetFlow-ს ბოლო ტექნოლოგია. მოქნილი NetFlow აუმჯობესებს „ორიგინალ NetFlow“-ს, ტრაფიკის პარამეტრების მომართვის შესაძლებლობების დამატებით, რომელიც გამოიყენება ქსელის ადმინისტრატორის კონკრეტული მოთხოვნებისთვის.

მოქნილი NetFlow ამარტივებს  უფრო რთული ტრაფიკის ანალიზის და მონაცემთა ექსპორტის კონფიგურაციის შექმნას, მრავალჯერადი კონფიგურაციის კომპონენტების გამოყენებით.

მოქნილი NetFlow იყენებს მე-9 ვერსიის ექსპორტის ფორმატს. მე-9 ვერსიის NetFlow-ს ექსპორტის ფორმატის განმასხვავებელი მახასიათებელი არის ის, რომ ის არის შაბლონზე დაფუძნებული. შაბლონები იძლევიან ჩაწერის ფორმატის გაფართოებულ დიზაინს, ფუნქცია, რომელიც უზრუნველყოფს  NetFlow მომსახურების მომავალ გაუმჯობესებას, არ საჭიროებს ერთდროულ ცვლილებას ჩაწერის ნაკადის ძირითად ფორმატში. აღსანიშნავია, რომ მოქნილი NetFlow ბევრი საჭირო ბრძანება შეტანილ იქნა Cisco IOS-ის 15.1 ვერსიაში.

პრაქტიკული სამუშაო  - Syslog-ის და NTP-ს კონფიგურაცია

**ტოპოლოგია:**




**შესასრულებელი ამოცანები:**

**ნაწილი №1: Syslog სერვისის კონფიგურაცია**

**ნაწილი №2: რეგისტრირებული ღონისძიებების შექმნა**

**ნაწილი №3: კომუტატორის საათების (Switch Clocks) ხელით მომართვა**

**ნაწილი №4: NTP სერვისის კონფიგურაცია**

**ნაწილი №5: დაფიქსირებული სარეგისტრაციო ჟურნალისების შემოწმება**

**სცენარი**

მოცემულ ლაბორატორიულ დავალებაში თქვენ ჩართავთ და გამოიყენებთ **Syslog** და **NTP** სერვისებს, იმისათვის, რომ ქსელის ადმინისტრატორმა უფრო ეფექტურად შეძლოს ქსელის მონიტორინგი.

## ნაწილი №1: Syslog სერვისის კონფიგურაცია

### პირველი ეტაპი: Syslog სერვისის ჩართვა

ა. დააჭირეთ **Syslog**-ს, შემდეგ გადადით **Services** ჩანართში.

ბ. ჩართეთ **Syslog** სერვისი და გადაადგილეთ ფანჯარა ისე, რომ შეძლოთ საქმიანობის მონიტორინგი.

### მეორე ეტაპი: შუალედური მოწყობილობების კონფიგურაცია Syslog სერვისის გამოსაყენებლად

ა. **R1** მარშრუტიზატორის კონფიგურაცია ღონისძიებების ჟურნალის **(log events)** გასაგზავნად **Syslog** სერვერზე.

R1(config)# **logging 10.0.1.254**

ბ. დააკონფიგურეთ **S1** კომუტატორი ღონისძიებების ჟურნალის გასაგზავნად **Syslog** სერვერზე.

გ. დააკონფიგურეთ **S2** კომუტატორი ღონისძიებების ჟურნალის გასაგზავნად **Syslog** სერვერზე.

## ნაწილი №2: რეგისტრირებული ღონისძიებების (Logged Events) შექმნა

### პირველი ეტაპი: ინტერფეისების სტატუსის შეცვლა ღონისძიებების ჟურნალის შესაქმნელად.

ა. დააკონფიგურეთ **Loopback 0** ინტერფეისი **R1** მარშრუტიზატორზე, შემდეგ გათიშეთ.

ბ. გათიშეთ **PC1** და **PC2**. შემდეგ ხელახლა ჩართეთ

### მეორე ეტაპი: Syslog ღონისძიებების შესწავლა

ა. დაათვალიერეთ **Syslog** ღონისძიებები. შენიშვნა: ჩაწერილ იქნა ყველა ღონისძიება; თუმცა დროის შტამპები **(Time stamps)** არასწორია.

ბ. გაასუფთავეთ სარეგისტრაციო ჟურნალები (logs) შემდეგი ნაწილის დაწყებამდე.

## ნაწილი №3: კომუტატორის საათების ხელით დაყენება

### პირველი ეტაპი: საათების ხელით მომართვა კომუტატორებზე.

**S1** და **S2** კომუტატორებზე ხელით მომართეთ საათი მიმდინარე თარიღით და მიახლოებითი დროით. ქვემოთ მოცემულია მაგალითი:

S1# **clock set 11:47:00 July 10 2013**

### მეორე ეტაპი: ჟურნალირების დროითი შტამპების (logging timestamp) სერვისის დაყენება კომუტატორებზე.

დააკონფიგურეთ **S1** და **S2** კომუტატორი იმისათვის, რომ გააგზავნონ თავიანთი დროითი შტამპები სარეგისტრაციო ჟურნალებთან ერთად, რომლებსაც გზავნიან **Syslog** სერვერთან.

S1 (config) # **service timestamps log datetime msec**

## ნაწილი №4: NTP სერვისის კონფიგურაცია

### პირველი ეტაპი: NTP სერვისის ჩართვა

ამ დავალებაში უნდა ვივარაუდოთ რომ **NTP** სერვისი გამართულია საერთო ინტერნეტ სერვერზე. თუ **NTP** სერვერი არის კერძო, შესაძლოა გამოყენებულ იქნას აუთენტიფიკაციაც.

ა. გახსენით **NTP** სერვერის **Services** ჩანართი

ბ. ჩართეთ **NTP** სერვისი და ჩაინიშნეთ ის დრო და თარიღი, რომელიც ნაჩვენებია ეკრანზე.

### მეორე ეტაპი: საათის ავტომატური მომართვა მარშრუტიზატორზე

მომართეთ **R1** მარშრუტიზატორის საათი თარიღსა და დროსთან ერთად, **NTP** სერვერის შესაბამისად.

R1 (config) # **ntp server 64.103.224.2**

**მესამე ეტაპი: ჟურნალირების დროითი შტამპების ჩართვა მარშრუტიზატორზე**

დააკონფიგურეთ **R1** მარშრუტიზატორი იმისათვის, რომ გააგზავნონ თავიანთი დროითი შტამპები სარეგისტრაციო ჟურნალებთან (Log) ერთად, რომლებსაც გზავნიან **Syslog** სერვერთან.

**ნაწილი №5: დროით მარკირებული სარეგისტრაციო ჟურნალების (Timestamped Logs) შემოწმება**

ა. ხელახლა ჩართეთ და შემდეგ გამორთეთ **Loopback 0** ინტერფეისი **R1** მარშრუტიზატორზე

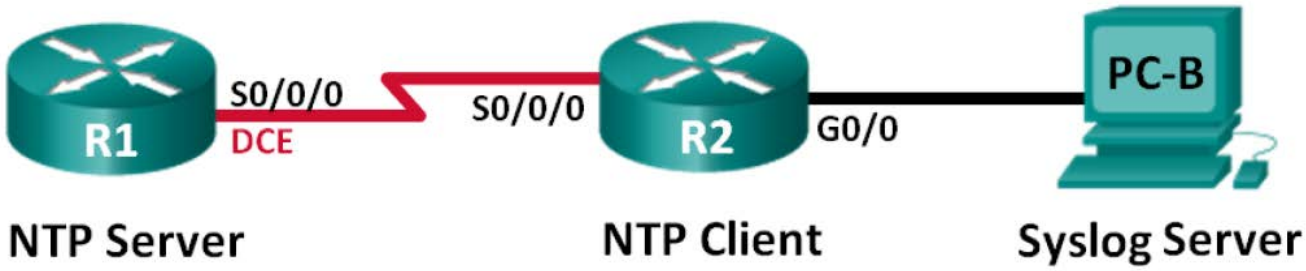
ბ. გათიშეთ **L1** და **L2** ლეპტოპი. შემდეგ ხელახლა ჩართეთ

**მეორე ეტაპი: შეისწავლეთ Syslog ღონისძიებები**

დაათვალიერეთ **Syslog** ღონისძიებები. **შენიშვნა:** ყველა ღონისძიება ჩაწერილია და დროითი შტამპებიც სწორია, კონფიგურაციის შესაბამისად. **შენიშვნა: R1** იყენებს საათის პარამეტრებს **NTP** სერვერიდან, ხოლო **S1** და **S2** იყენებენ საათის პარამეტრებს თქვენს მიერ **ნაწილი №3**-ის კონფიგურაციის შესაბამისად.

პრაქტიკული სამუშაო - Syslog-ის და NTP-ს კონფიგურაცია

**ტოპოლოგია**




**მისამართების ცხრილი:**

| მოწყობილობები | ინტერფეისი | IP მისამართი | ქვექსელის ნიღაბი | ნაგულისხმევი გასასვლელი |
|---|---|---|---|---|
| R1 | S0/0/0 (DCE) | 10.1.1.1 | 255.255.255.252 | N/A |
| R2 | S0/0/0 | 10.1.1.2 | 255.255.255.252 | N/A |
| | G0/0 | 172.16.2.1 | 255.255.255.0 | N/A |
| PC-B | NIC | 172.16.2.3 | 255.255.255.0 | 172.16.2.1 |

**შესასრულებელი დავალებები:**

**ნაწილი №1: მოწყობილობის ბაზისური პარამეტრების კონფიგურაცია**

**ნაწილი №2: NTP-ს კონფიგურაცია**

**ნაწილი №3: Syslog-ის კონფიგურაცია**

**ზოგადი ინფორმაცია / სცენარი**

**Syslog** შეტყობინებები, რომლებიც შექმნილია ქსელური მოწყობილობების მიერ, შეიძლება შეგროვდეს და დაარქივდეს Syslog სერვერზე. ინფორმაცია შესაძლოა გამოყენებულ იქნას მონიტორინგის, გამართვისა (**Debugging**) და პრობლემის მოძიებისა და აღმოფხვრის მიზნებისათვის. ადმინისტრატორს შეუძლია აკონტროლოს თუ სად არის შენახული და ნაჩვენები შეტყობინებები. **Syslog** შეტყობინებები შეიძლება იყოს დროით-მარკირებული ქსელური ღონისძიებების თანმიმდევრობის ანალიზისთვის. ამიტომ

აუცილებელია საათის სინქრონიზაცია ქსელური მოწყობილობებს შორის ქსელური დროის პროტოკოლის (**NTP**) სერვერით.

ამ ლაბორატორიულ დავალებაში თქვენ დააკონფიგურებთ **R1** მარშრუტიზატორს, როგორც **NTP** სერვერი და **R2** მარშრუტიზატორს, როგორც **Syslog** და **NTP** კლიენტი. **Syslog** სერვერის აპლიკაცია, **Tftp32d**-ის ან მსგავსი პროგრამის ჩათვლით, გაშვებულ იქნება **PC-B** კომპიუტერზე. გარდა ამისა თქვენ აკონტროლებთ სარეგისტრაციო ჟურნალების შეტყობინებების „სირთულის“ დონეს, რომელიც შეგროვებულია და დაარქივებული **Syslog** სერვერზე.

**შენიშვნა:** მარშრუტიზატორები, რომლებიც გამოიყენება **CCNA**-ს პრაქტიკული სამუშაოებისთვის, არის **Cisco 1941** ინტეგრირებული სერვისების მარშრუტიზატორები **(ISRs) Cisco IOS Release 15.2(4)M3 (universalk9 image)** ვერსიასთან ერთად. შესაძლოა გამოყენებულ იქნას სხვა მარშრუტიზატორები და **Cisco IOS** ვერსიებიც. მოდელისა და **Cisco IOS** ვერსიის მიხედვით ხელმისაწვდომი ბრძანებები და მიღებული შედეგები შეიძლება იყოს განსხვავებული იმისგან, რაც ნაჩვენებია ამ ლაბორატორიულ სამუშაოში. მიაქციეთ ყურადღება მარშრუტიზატორის ინტერფეისის შემაჯამებელ ცხრილს ამ დავალების ბოლოში, სწორი ინტერფეისის იდენტიფიკატორებისათვის.

**შენიშვნა:** დარწმუნდით, რომ მარშრუტიზატორებს არ აქვთ საწყისი კონფიგურაციები. თუ არ ხართ დარწმუნებული დაუკავშირდით თქვენს ინსტრუქტორს.

**მოთხოვნილი რესურსები:**

- 2 მარშრუტიზატორი **(Cisco 1941 with Cisco IOS Release 15.2.(4)M3** უნივერსალი ან მსგავსი იმიჯით)
- ერთი პერსონალური კომპიუტერი (**Windows** ოპერაციული სისტემით ტერმინალის ემულაციის პროგრამასთან ერთად, **Tera Term**-ის ჩათვლით და Syslog პროგრამული უზრუნველყოფით, **tftpd32**-ის ჩათვლით)
- კონსოლის კაბელი **Cisco IOS** მოწყობილობების კონსოლის პორტებით კონფიგურაციისათვის

- ტოპოლოგიაზე ნაჩვენები **Ethernet** და სერიალური კაბელები

**ნაწილი №1: მოწყობილობილობის ბაზისური პარამეტრების კონფიგურაცია**

პირველ ნაწილში თქვენ მომართავთ ქსელის ტოპოლოგიას და დააკონფიგურებთ ბაზისურ პარამეტრებს, ინტერფეისის **IP** მისამართების, მარშრუტიზაციის, მოწყობილობებთან წვდომის და პაროლების ჩათვლით.

**პირველი ეტაპი: კაბელების შეერთება ისე, როგორც ნაჩვენებია ტოპოლოგიაზე**

**მეორე ეტაპი: მარშრუტიზატორის ინიციალიზაცია და ხელახლა ჩატვირთვა, აუცილებლობის შემთხვევაში**

**მესამე ეტაპი: თითოული მარშრუტიზატორის ბაზისური პარამეტრების კონფიგურაცია**

ა. გათიშეთ **DNS lookup**

ბ. დააკონფიგურეთ მოწყობილობის სახელი

გ. დაშიფრეთ ღია ტექსტის პაროლები

დ. შექმენით დღის შეტყობინების (**MOTD**) გამაფრთხილებელი ბანერი მომხმარებლებისათვის, რომლებიც ახორციელებენ არაავტორიზებულ წვდომას.

ე. დააყენეთ **class**, როგორც დაშიფრული პრივილეგირებული **EXEC** რეჟიმის პაროლი

ვ. დანიშნეთ **cisco** როგორც კონსოლის და **vty** პაროლი და ჩართეთ შესვლა (**login**)

ზ. მომართეთ კონსოლის ჟურნალირება სინქრონული რეჟიმისთვის

თ. მისამართების ცხრილის მიხედვით მომართეთ **IP** მისამართები სერიალური და გიგაბიტური **Ethernet** ინტერფეისებისათვის და გააქტიურეთ ფიზიკური ინტერფეისები.

ი. დააყენეთ ტაქტური სიხშირე **128000**-ზე **DCE** სერიალ ინტერფეისისთვის

### მეოთხე ეტაპი: მარშრუტიზაციისკონფიგურაცია

ჩართეთ **single-area OSPF** მარშრუტიზატორებზე **process ID 1**-ით. დაამატეთ ყველა ქსელი **OSPF** პროცესში **area 0**-სთვის.

### მეხუთე ეტაპი: PC-B კომპიუტერის კონფიგურაცია

მისამართების ცხრილის მიხედვით დააკონფიგურეთ **PC-B** კომპიუტერის **IP** მისამართი და ნაგულისხმევი გასასვლელი.

### მეექვსე ეტაპი: სრული ციკლის კავშირის შემოწმება

შეამოწმეთ თითოეულ მოწყობილობას წარმატებით შეუძლია თუ არა **ping**-ის გაშვება ქსელის სხვა მოწყობილობებთან. თუ არა მოძებნეთ და აღმოფხვერით პრობლემა, სანამ არ მიიღწევა სრულკავშირიანი ციკლი.

### მეშვიდე ეტაპი: შეინახეთ გაშვებული კონფიგურაცია საწყის კონფიგურაციაში

## ნაწილი №2: NTP-ს კონფიგურაცია

მეორე ნაწილში თქვენ დააკონფიგურებთ **R1** მარშრუტიზატორს, როგორც **NTP** სერვერს და **R2** მარშრუტიზატორს, როგორც **R1**-ის **NTP** კლიენტს. სინქრონიზებული დრო აუცილებელია **Syslog** და **debug** ფუნქციებისათვის. თუ დრო არ არის სინქრონული, რთულია იმის განსაზღვრა რომელმა ქსელურმა ღონისძიებამ გამოიწვია შეტყობინება.

### პირველი ეტაპი: მიმდინარე თარიღის ჩვენება

გაუშვით **show clock** ბრძანება **R1** მარშრუტიზატორზე მიმდინარე თარიღის საჩვენებლად.

R1# **show clock**

*12:30:06.147 UTC Tue May 14 2013

ჩაიწერეთ მიმდინარე თარიღის ინფორმაცია ქვემოთ მოცემულ ცხრილში

| | |
|---|---|
| **თარიღი** | |
| **დრო** | |
| **სასაათო ზონა** | |

**მეორე ეტაპი: დროის დაყენება**

გამოიყენეთ clock set ბრძანება R1 მარშრუტიზატორზე დროის მოსამართად. ქვემოთ მოცემული არის თარიღისა და დროის პარამეტრების მაგალითი.

R1# **clock set 9:39:00 05 july 2013**

R1#

*Jul 5 09:39:00.000: %SYS-6-CLOCKUPDATE: System clock has been updated from 12:30:54 UTC Tue May 14 2013 to 09:39:00 UTC Fri Jul 5 2013, configured from console by console.

**შენიშვნა:** დრო შეიძლება ასევე დაყენებულ იქნას **clock timezone** ბრძანების გამოყენებით გლობალური კონფიგურაციის რეჟიმში. ამ ბრძანებასთან მიმართებაში მეტი ინფორმაციისთვის, მოძებნეთ **clock timezone** ბრძანება www.cisco.com საიტზე, თქვენი რეგიონის ზონის განსასაზღვრად.

**მესამე ეტაპი: NTP მმართველის კონფიგურაცია**

დააკონფიგურეთ **R1** როგორც **NTP** მმართველი **ntp master** *<შრის (stratum)-ნომერი>* ბრძანების გამოყენებით გლობალური კონფიგურაციის რეჟიმში. შრის ნომერი (**stratum number**) მიუთითებს **NTP** გადასვლების (**hops**) რიცხვს აუთენტური დროის წყაროდან. მოცემულ ლაბორატორიულ დავალებაში რიცხვი 5 არის ამ **NTP** სერვერის შრის დონე.

R1 (config) # **ntp master 5**

**მეოთხე ეტაპი: NTP კლიენტის კონფიგურაცია**

ა. გაუშვით **show clock** ბრძანება **R2** მარშრუტიზატორზე. ჩაიწერეთ **R2**-ზე ნაჩვენები მიმდინარე თარიღი, ქვემოთ მოცემულ ცხრილში.

| | |
|---|---|
| **თარიღი** | |
| **დრო** | |
| **სასაათო ზონა** | |

ბ. დააკონფიგურეთ **R2** მარშრუტიზატორი, როგორც **NTP** კლიენტი. გამოიყენეთ **ntp server** ბრძანება **NTP** სერვერის **IP** მისამართის ან ჰოსტის სახელის მისათითებლად. **ntp update-calendar** ბრძანება პერიოდულად განაახლებს კალენდარს **NTP** დროით.

R2 (config) # **ntp server 10.1.1.1**

R2 (config) # **ntp update-calendar**

**მეხუთე ეტაპი: NTP კონფიგურაციის შემოწმება**

ა. გამოიყენეთ **show ntp associations** ბრძანება იმის შესამოწმებლად აქვს თუ არა **R2**-ს **NTP** კავშირი **R1**-თან.

R2# **show ntp associations**

| address | ref clock | st | when | poll | reach | delay | offset | disp |
|---|---|---|---|---|---|---|---|---|
| *~10.1.1.1 | 127.127.1.1 | 5 | 11 | 64 | 177 | 11.312 | -0.018 | 4.298 |

*sys. peer, #selected, +candidate, -outlyer, x falseticker, ~configured

ბ. გაუშვით **show clock** ბრძანება **R1** და **R2** მარშრუტიზატორებზე, დროის მარკირების შესადარებლად.

**შენიშვნა:** რამდენიმე წუთია საჭირო **R2** მარშრუტიზატორის დროითი მარკირების სინქრონიზაციისთვის **R1**-თან.

R1# **show clock**

09:43:32.799 UTC Fri Jul 5 2013

R2# **show clock**

09:43:37.122 UTC Fri Jul 5 2013

**ნაწილი №3: Syslog-ის კონფიგურაცია**

ქსელური მოწყობილობებიდან **Syslog** შეტყობინებები შეიძლება შეგროვდეს და დაარქივდეს **syslog** სერვერზე. მოცემულ ლაბორატორიულ სამუშაოში, **Tftpd32** პროგრამული უზრუნველყოფა გამოიყენება როგორც **syslog** სერვერი. ქსელის

ადმინისტრატორს შეუძლია იმ შეტყობინებების ტიპების კონტროლი, რომლებიც შეიძლება გაიგზავნოს **syslog** სერვერზე.

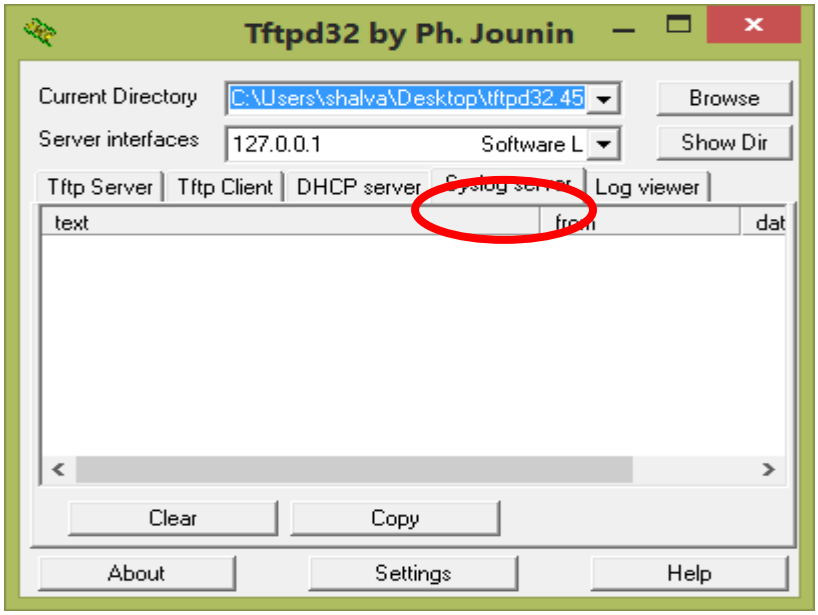
**პირველი ეტაპი: (არააუცილებელი) syslog სერვერის ინსტალაცია**

თუ **syslog** სერვერი ჯერ კიდევ არ არის ინსტალირებულის კომპიუტერზე, გადმოწერეთ და დააინსტალირეთ კომპიუტერზე **syslog** სერვერის ბოლო ვერსია, ისეთი როგორიცაა **Tftpd32**. **Tftpd32**-ის ბოლო ვერსია შეგიძლიათ გადმოწეროთ მოცემული ბმულიდან:

http://tftpd32.jounin.net/

**მეორე ეტაპი: syslog სერვერის გაშვება PC-B კომპიუტერზე**

**Tftpd32** აპლიკაციის გაშვების შემდეგ, დააჭირეთ **syslog server** ჩანართს.




**მესამე ეტაპი: შეამოწმეთ ჩართულია თუ არა დროითი მარკირების სერვისი R2 მარშრუტიზატორზე.**

გამოიყენეთ **show run** ბრძანება იმის შესამოწმებლად, რომ დროითი მარკირების სერვისი ჟურნალირებისთვის (**Logging**) ჩართულია თუ არა **R2** მარშრუტიზატორზე.

R2# **show run | include timestamp**

service timestamps debug datetime msec

service timestamps log datetime msec

თუ დროითი მარკირების სერვისი არაა ჩართული, გამოიყენეთ ქვემოთ მოცემული ბრძანება მის ჩასართავად.

R2 (config) # **service timestamps log datetime msec**

**მეოთხე ეტაპი: R2 მარშრუტიზატორის ჟურნალირების შეტყობინებების კონფიგურაცია syslog სერვერზე.**

დააკონფიგურეთ **R2** მარშრუტიზატორი **syslog** შეტყობინებების გასაგზავნად syslog სერვერზე, **PC-B. PC-B syslog** სერვერის **IP** მისამართი არის 172.16.2.3.

R2(config)# **logging host 172.16.2.3**

**მეხუთე ეტაპი: ნაგულისხმევი ჟურნალირების პარამეტრების ჩვენება**

გამოიყენეთ **show logging** ბრძანება, ნაგულისხმევი ჟურნალირების პარამეტრების გამოსატანად.

R2# **show logging**

Syslog logging: enabled (0 messages dropped, 2 messages rate-limited, 0 flushes, 0 overruns, xml disabled, filtering disabled)

No Activity Mesaage Discriminator.

No Inactive Message Discriminator.

Console logging: level debugging, 47 messages logged, xml disabled,
filtering disabled

Monitor logging: level debugging, 0 messages logged, xml disabled,
filtering disabled

Buffer logging: level debugging, 47 messages logged, xml disabled,
filtering disabled

Exception Logging: size (4096 bytes)

Count and timestamp logging messages: disabled

Persistent logging: disabled

No active filter modules.

Trap logging: level informational, 49 message lines logged

Logging to 172.16.2.3 (udp port 514, audit disabled,

link up),

6 message lines logged,

0 message lines rate-limited,

0 message lines dropped-by-MD,

xml disabled, sequence number disabled

filtering disabled

Logging Source-Interface: VRF Name:

რა არის **syslog** სერვერის **IP** მისამართი? _____

რომელ პორტს და პროტოკოლს იყენებს **syslog**-ი?_____

რომელ დონეზეა ჩართული **trap** ჟურნალირება (**Logging**)? _____

**მეექვსე ეტაპი: კონფიგურაცია და ჟურნალირების „სირთულის“ დონეების დაკვირვება R2 მარშრუტიზატორზე.**

ა. გამოიყენეთ **logging trap ?** ბრძანება სხვადასხვა **trap** დონეების ხელმისაწვდომობის განსასაზღვრად. როდესაც ვაკონფიგურებთ დონეს, **syslog** სერვერზე გაგზავნილი შეტყობინებები არის **trap** დონეზე და ნებისმიერ უფრო დაბალ დონეზე მომართული.

R2 (config) # **logging trap ?**

| <0-7> | Logging severity level | |
|---|---|---|
| alerts | Immediate action needed | {severity=1} |
| critical | Critical conditions | {severity=2} |
| debugging | Debugging messages | {severity=7} |
| emergencies | System is unusable | {severity=0} |

| | | |
|---|---|---|
| errors | Error conditions | {severity=3} |
| informational | Informational messages | {severity=6} |
| notifications | Normal but significant conditions | {severity=5} |
| warnings | Warning conditions | {severity=4} |
| <cr> | | |

თუ გაშვებულ იქნა **logging trap warnings** ბრძანება, როგორი სირთულის დონის შეტყობინებები იქნება ჟურნალირებული?_____

_____

ბ. ჟურნალირების (**Logging**) სირთულის დონის შეცვლა 4-მდე.

R2 (config) # **logging trap warnings**

or

R2 (config) # **logging trap 4**

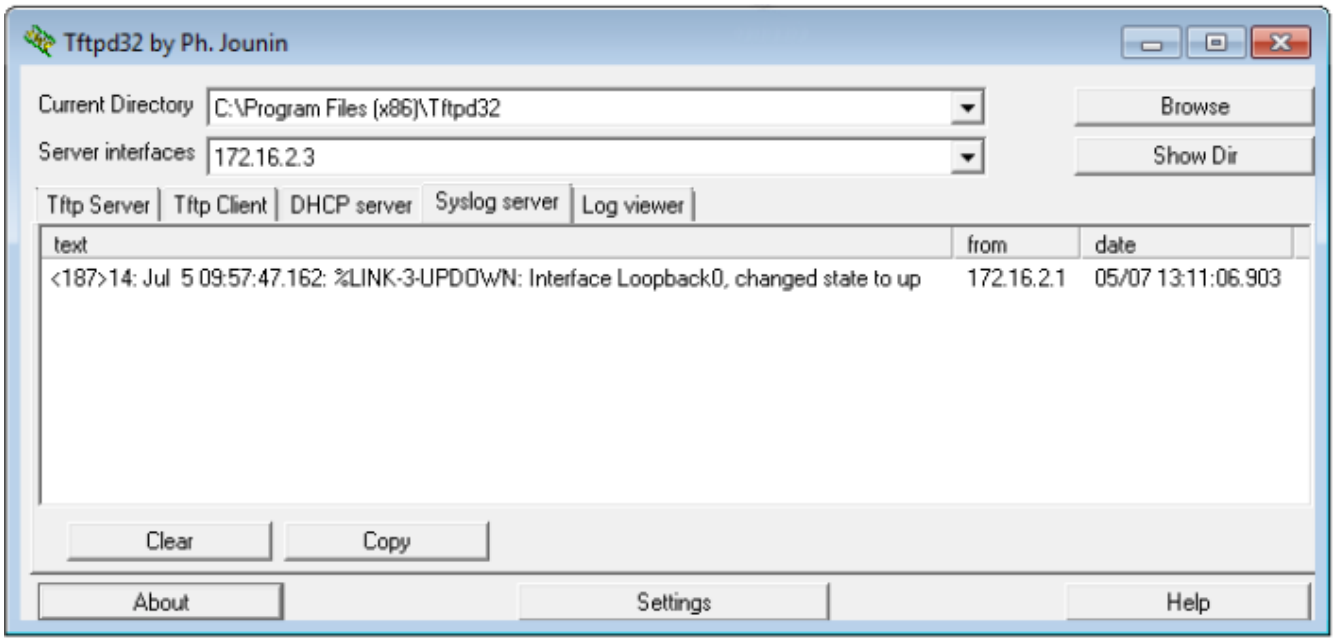
გ. **Loopback0** ინტერფეისის შექმნა **R2** მარშრუტიზატორზე და სარეგისტრაციო ჟურნალის შეტყობინებებზე დაკვირვება როგორც ტერმინალის ფანჯარაზე, ისე **PC-B** კომპიუტერის **syslog** სერვერის ფანჯარაზე.

R2 (config) # **interface lo 0**

R2 (config-if)#

Jul 5 09:57:47.162: %LINK-3-UPDOWN: Interface Loopback0, changed state to up

Jul 5 09:57:48.162: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up




დ. გააუქმეთ **Loopback0** ინტერფეისი **R2** მარშრუტიზატორზე და დააკვირდით სარეგისტრაციო ჟურნალის შეტყობინებებს.

R2 (config-if)# **no interface lo 0**

R2 (config) #

Jul 5 10:02:58.910: %LINK-5-CHANGED: Interface Loopback0, changed state to administratively down

Jul 5 10:02:59.910: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to down

სირთულის მეოთხე დონეზე, არის რაიმე სარეგისტრაციო ჟურნალის შეტყობინებები **syslog** სერვერზე? თუ რაიმე სარეგისტრაციო ჟურნალის შეტყობინება გამოჩნდა, ახსენით რა და რატომ გამოჩნდა. _____

_____

_____

ე. შეცვალეთ ჟურნალირების სირთულის დონე 6-მდე.

R2 (config) # **logging trap informational**

or

R2 (config) # **logging trap 6**

ვ. წაშალეთ syslog ჩანაწერები **PC-B** კომპიუტერზე. დააჭირეთ **Clear** ღილაკს **Tftpd32** დიალოგურ ფანჯარაში.

ზ. შექმენით **Loopback 1** ინტერფეისი **R2** მარშრუტიზატორზე.

R2 (config)# **interface lo 1**

Jul 5 10:05:46.650: %LINK-3-UPDOWN: Interface Loopback1, changed state to up

Jul 5 10:05:47.650: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to up

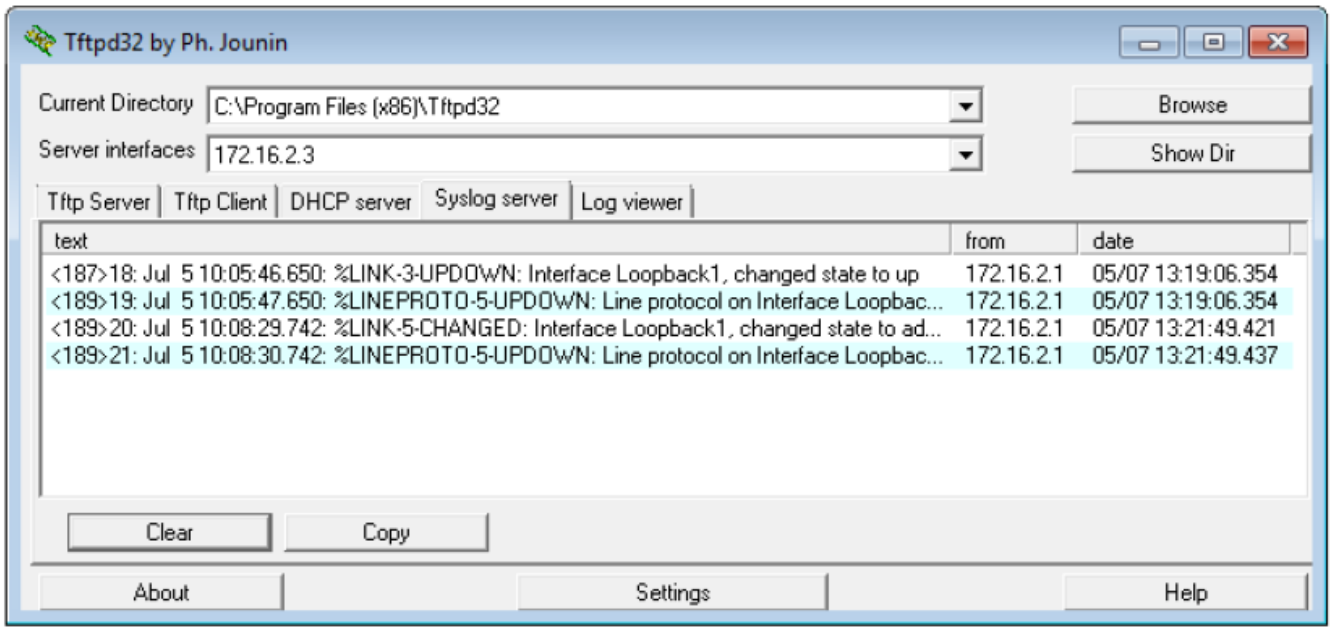
თ. გააუქმეთ **Loopback 1** ინტერფეისი **R2** მარშრუტიზატორზე

R2 (config-if) # **no interface lo 1**

R2 (config-if) #

Jul 5 10:08:29.742: %LINK-5-CHANGED: Interface Loopback1, changed state to administratively down

Jul 5 10:08:30.742: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to down




ი. დააკვირდით **syslog** სერვერის შედეგს. შეუდარეთ მოცემული შედეგი **trapping** მეოთხე დონის შედეგებს. თქვენი დასკვნა? _____

_____

_____

_____

## ასახვა (Reflection)

**Syslog**-სთვის რა არის პრობლემა, სირთულის დონის პარამეტრის ძალიან მაღალზე (დაბალი დონის რიცხვი) თუ ძალიან დაბალზე დაყენება (მაღალი დონის რიცხვი).

_____

_____

_____

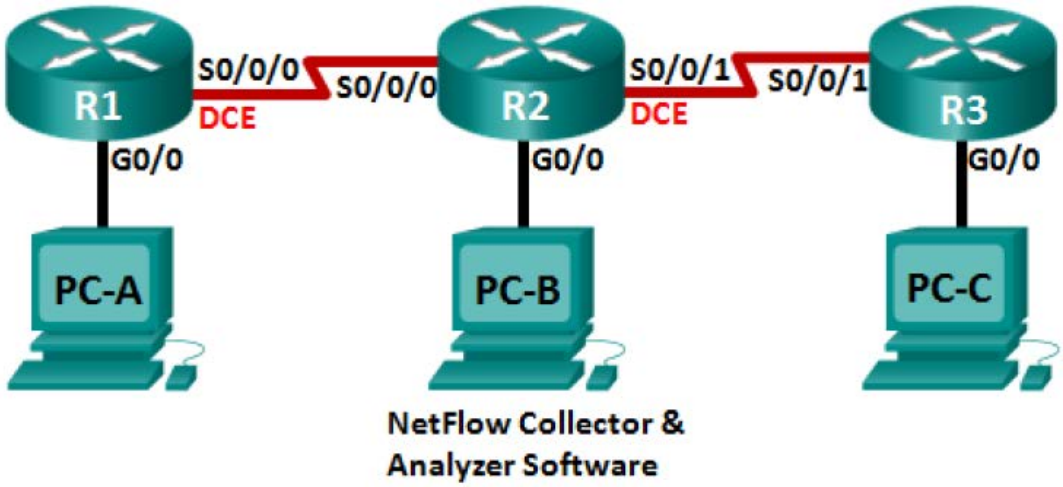
მარშრუტიზატორის ინტერფეისის შემაჯამებელი ცხრილი:

| მარშრუტიზატორის ინტერფეისის შეჯამება | | | | |
|---|---|---|---|---|
| მარშრუტიზატორის მოდელი | Ethernet interface №1 | Ethernet interface №2 | Serial interface №1 | Serial interface №2 |
| 1800 | FastEthernet 0/0 (F0/0) | FastEthernet 0/1 (F0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |
| 1900 | GigabitEthernet 0/0 (G0/0) | GigabitEthernet 0/1 (G0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |
| 2801 | FastEthernet 0/0 (F0/0) | FastEthernet 0/1 (F0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |
| 2811 | FastEthernet 0/0 (F0/0) | FastEthernet 0/1 (F0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |
| 2900 | GigabitEthernet 0/0 (G0/0) | GigabitEthernet 0/1 (G0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |

**შენიშვნა:** თუ გინდათ მოძებნოთ თუ როგორაა მარშრუტიზატორი კონფიგურირებული, დაათვალიერეთ ინტერფეისები რათა მოახდინოთ მარშრუტიზატორის იდენტიფიკაცია და დაადგინოთ რამდენი ინტერფეისი აქვს მას. აქ არ არის იმის საშუალება, რომ ჩამოიწეროს თითოეული მარშრუტიზატორის კონფიგურაციების კომბინაციები. მოცემული ცხრილი მოიცავს მოწყობილობის **Ethernet** და **Serial** ინტერფეისების შესაძლო კომბინაციების იდენტიფიკატორებს. ეს ცხრილი არ მოიცავს ნებისმიერი სხვა ტიპის ინტერფეისს, რადგან კონკრეტული მარშრუტიზატორი შეიძლება შეიცავდეს მხოლოდ ერთს. ამის მაგალითი შეიძლება იყოს **ISDN BRI** ინტერფეისი. ფრჩხილებში მოცემული ცვლადი არის ლეგალური აბრევიატურა, რომელიც შეიძლება გამოყენებულ იქნას **Cisco IOS** ბრძანებებში, ინტერფეისის წარმოსადგენად.

პრაქტიკული სამუშაო - Netflow მონაცემების შეგროვება და ანალიზი

**ტოპოლოგია:**




**მისამართების ცხრილი:**

| მოწყობილობა | ინტერფეისი | IP მისამართი | ნაგულისხმევი გასასვლელი |
|---|---|---|---|
| R1 | G0/0 | 192.168.1.1/24 | N/A |
| | S0/0/0 (DCE) | 192.168.12.1/30 | N/A |
| R2 | G0/0 | 192.168.2.1/24 | N/A |
| | S0/0/0 | 192.168.12.2/30 | N/A |
| | S0/0/1 (DCE) | 192.168.23.1/30 | N/A |
| R3 | G0/0 | 192.168.3.1/24 | N/A |
| | S0/0/1 | 192.168.23.2/30 | N/A |
| PC-A | NIC | 192.168.1.3 | 192.168.1.1 |
| PC-B | NIC | 192.168.2.3 | 192.168.2.1 |
| PC-C | NIC | 192.168.3.3 | 192.168.3.1 |

შესასრულებელი ამოცანები:

ნაწილი №1: ქსელის აწყობა და მოწყობილობის პარამეტრების ბაზისური კონფიგურაცია

ნაწილი №2: NetFlow-ს კონფიგურაცია მარშრუტიზატორზე

ნაწილი №3: NetFlow-ს ანალიზი CLI-ს გამოყენებით

ნაწილი №4: NetFlow კოლექტორის შესწავლა და ანალიზატორი პროგრამული უზრუნველყოფა

ზოგადი ინფორმაცია/ სცენარი

**NetFlow** არის **Cisco IOS** ტექნოლოგია, რომელიც იძლევა **Cisco** მარშრუტიზატორიდან ან მრავალდონიანი კომუტატორიდან წამოსული პაკეტების სტატისტიკას. **NetFlow** იძლევა ქსელისა და უსაფრთხოების მონიტორინგის, ქსელის დაგეგმვის, ტრაფიკის ანალიზის და **IP** ანგარიშის საშუალებას. მნიშვნელოვანია, რომ არ აურიოთ **NetFlow**-ს დანიშნულება და შედეგები, პაკეტების დაჭერის ტექნიკურ და პროგრამულ უზრუნველყოფასთან. პაკეტების დამჭერი იწერს ყველა შესაძლო ინფორმაციას, რომლებიც გამოდის ან შედის ქსელურ მოწყობილობაზე, შემდგომი ანალიზისათვის, **NetFlow**-ს კონკრეტული დავალებებია სტატისტიკური ინფორმაცია.

მოქნილი **NetFlow** არის **NetFlow**-ს ბოლო ტექნოლოგია. ის გაუმჯობესებულია ორიგინალ **NetFlow**-სთან შედარებით, რადგან დამატებული აქვს ტრაფიკის ანალიზის პარამეტრების მომართვის შესაძლებლობა. მოქნილი **NetFlow** იყენებს მე-9 ვერსიის ექსპორტის ფორმატს. დაწყებული **Cisco IOS Relase 15.1**-დან, მოქნილი **NetFlow**-ს მრავალი საჭირო ბრძანებაა მხარდაჭერილი.

მოცემულ ლაბორატორიულ სამუშაოში თქვენ დააკონფიგურებთ **NetFlow**-ს, როგორც შემომავალი ისე გამავალი პაკეტების დასაჭერად. თქვენ გამოიყენებთ **show** ბრძანებებს, იმის შესამოწმებლად, რომ **NetFlow** ფუნქციონირებს და იღებს სტატისტიკურ ინფორმაციას.

თქვენ ასევე შეისწავლით **NetFlow** შეგროვებისა და ანალიზის პროგრამული უზრუნველყოფის ხელმისაწვდომ პარამეტრებს.

**შენიშვნა:** მარშრუტიზატორები, რომლებიც გამოიყენება **CCNA**-ს პრაქტიკული სამუშაოებისთვის, არის **Cisco 1941** ინტეგრირებული სერვისების მარშრუტიზატორები **(ISRs) Cisco IOS Release 15.2(4)M3 (universalk9 image)** ვერსიასთან ერთად. შესაძლოა გამოყენებულ იქნას სხვა მარშრუტიზატორები და **Cisco IOS** ვერსიებიც. მოდელისა და **Cisco IOS** ვერსიის მიხედვით ხელმისაწვდომი ბრძანებები და მიღებული შედეგები შეიძლება იყოს განსხვავებული იმისგან, რაც ნაჩვენებია ამ ლაბორატორიულ სამუშაოში. მიაქციეთ ყურადღება მარშრუტიზატორის ინტერფეისის შემაჯამებელ ცხრილს ამ დავალების ბოლოში, სწორი ინტერფეისის იდენტიფიკატორებისათვის.

**შენიშვნა:** დარწმუნდით, რომ მარშრუტიზატორებს არ აქვთ საწყისი კონფიგურაციები. თუ არ ხართ დარწმუნებული დაუკავშირდით თქვენს ინსტრუქტორს.

**მოთხოვნილი რესურსები:**

- 3 მარშრუტიზატორი **(Cisco 1941 with Cisco IOS Release 15.2.(4)M3** უნივერსალი ან მსგავსი იმიჯით)
- სამი პერსონალური კომპიუტერი (**Windows** ოპერაციული სისტემით ტერმინალის ემულაციის პროგრამასთან ერთად, **Tera Term**-ის ჩათვლით)
- კონსოლის კაბელი **Cisco IOS** მოწყობილობების კონსოლის პორტებით კონფიგურაციისათვის
- ტოპოლოგიაზე ნაჩვენები **Ethernet** და სერიალური კაბელები

**ნაწილი №1: მოწყობილობილობის ბაზისური პარამეტრების კონფიგურაცია**

პირველ ნაწილში თქვენ მომართავთ ქსელის ტოპოლოგიას და დააკონფიგურებთ ბაზისურ პარამეტრებს **PC** ჰოსტებსა და მარშრუტიზატორებზე.

**პირველი ეტაპი: კაბელების შეერთება ისე, როგორც ნაჩვენებია ტოპოლოგიაზე**

**მეორე ეტაპი: მარშრუტიზატორის ინიციალიზაცია და ხელახლა ჩატვირთვა, აუცილებლობის შემთხვევაში**

**მესამე ეტაპი: თითოული მარშრუტიზატორის ბაზისური პარამეტრების კონფიგურაცია**

ა. გათიშეთ **DNS lookup**

ბ. დააკონფიგურეთ მოწყობილობის სახელი ისე როგორც ნაჩვენებია ტოპოლოგიაზე

გ. დააყენეთ **class**, როგორც დაშიფრული პრივილეგირებული **EXEC** რეჟიმის პაროლი

დ. დანიშნეთ **cisco** როგორც კონსოლის და **vty** პაროლი და ჩართეთ შესვლა (**login**)

ე. დაშიფრეთ ღია ტექსტის პაროლები

ვ. შექმენით დღის შეტყობინების (**MOTD**) გამაფრთხილებელი ბანერი მომხმარებლებისათვის, რომლებიც ახორციელებენ არაავტორიზებულ წვდომას.

ზ. მომართეთ **logging Synchronous** კონსოლის ხაზისთვის.

თ. დააყენეთ ტაქტური სიხშირე **128000**-ზე **DCE** სერიალ ინტერფეისისთვის

ი. დააკონფიგურეთ **IP** მისამართები მისამართების ცხრილის მიხედვით

კ. მომართეთ **OSPF Process ID 1**-ის გამოყენებით და შეატყობინეთ ყველა ქსელს. Ethernet ინტერფეისები უნდა იყოს პასიური

ლ. შექმენით ლოკალური მონაცემთა ბაზა **R3** მარშრუტიზატორზე, admin მომხმარებლის სახელითა და cisco პაროლით, პრივილეგირებულ დონე 15-ზე.

მ. **R3** მარშრუტიზატორზე ჩართეთ **HTTP** სერვისი და მოახდინეთ **HTTP** მომხმარებლების აუთენტიფიკაცია ლოკალური ბაზის გამოყენებით.

ნ. გადაიტანეთ გაშვებული კონფიგურაციის ასლი საწყის კონფიგურაციაში.

### მეოთხე ეტაპი: PC ჰოსტების კონფიგურაცია

### მეხუთე ეტაპი: ერთმანეთთან კავშირის შემოწმება

ყველა მოწყობილობას უნდა შეეძლოს ტოპოლოგიის სხვა მოწყობილობების დაპინგვა (**Ping**). აუცილებელია პრობლემის მოძიება და აღმოფხვრა, სანამ არ მიიღწევა სრული ციკლის კავშირი.

**შენიშვნა:** შეიძლება აუცილებელი იყოს პერსონალური კომპიუტერის ფაიერვოლის გათიშვა კომპიუტერებს შორის წარმატებული **ping**-სთვის.

## ნაწილი №2: NetFlow-ს კონფიგურაცია მარშრუტიზატორზე

მეორე ნაწილში თქვენ დააკონფიგურებთ **NetFlow**-ს **R2** მარშრუტზატორზე. **NetFlow** გამოიჭერს ყველა შემომავალ და გამავალ ტრაფიკს **R2** მარშრუტიზატორის სერიალურ ინტერფეისებზე და გაიტანს მონაცემებს **NetFlow** კოლექტორში, **PC-B**. **NetFlow** კოლექტორში ექსპორტირებისათვის გამოყენებული იქნება მოქნილი **NetFlow**-ს მე-9 ვერსია.

### პირველი ეტაპი: NetFlow გამოჭერის (Capture) კონფიგურაცია

მომართეთ **NetFlow** მონაცემთა დაჭერა ორივე სერიალურ ინტერფეისზე. დააკავეთ მონაცემები შემომავალი და გამავალი პაკეტებიდან.

R2 (config)# **interface s0/0/0**

R2 (config-if) # **ip flow ingress**

R2 (config-if)# **ip flow egress**

R2 (config-if) # **interface s0/0/1**

R2 (config-if) # **ip flow ingress**

R2 (config-if) # **ip flow egress**

**მეორე ეტაპი: NetFlow-ს მონაცემთა ექსპორტის კონფიგურაცია**

გამოიყენეთ **ip flow-export destination** ბრძანება **NetFlow** კოლექტორის **IP** მისამართისა და **UDP** პორტის იდენტიფიცირებისათვის, რომლითაც მარშრუტიზატორს შეუძლია **NetFlow**-ს მონაცემების ექსპორტი. ამ კონფიგურაციისათვის გამოყენებულ იქნება **9996 UDP** პორტის ნომერი.

R2 (config) # **ip flow-export destination 192.168.2.3 9996**

**მესამე ეტაპი: NetFlow-ს ექსპორტის ვერსიის კონფიგურაცია**

**Cisco** მარშრუტიზარტორები, რომელზეც გაშვებულია **IOS 15.1**, მხარს უჭერენ **NetFLow**-ს 1, 5 და 9 ვერსიებს. მე-9 ვერსია არის მოქნილი მონაცემთა ექსპორტის ფორმატი, მაგრამ არ არის თავსებადი წინამორბედ ძველ ვერსიებთან. გამოიყენეთ ip flow-export version ბრძანება NetFlow-ს ვერსიის მოსამართად.

R2 (config) # **ip flow-export version 9**

**მეოთხე ეტაპი: NetFlow-ს კონფიგურაციის შემოწმება**

ა. გაუშვით **show ip flow interface** ბრძანება **NetFlow** დაჭერის ინტერფეისის ინფორმაციის მიმოხილვისათვის.

R2 # **show ip flow interface**

Serial0/0/0

ip flow ingress

ip flow egress

Serial0/0/1

ip flow ingress

ip flow egress

ბ. გაუშვით **show ip flow export** ბრძანება, **NetFlow**-ს მონაცემთა ექსპორტის ინფორმაციის მიმოხილვისათვის.

R2# **show ip flow export**

Flow export v9 is enabled for main cache

Export source and destination details :

VRF ID : Default

Destination(1) 192.168.2.3 (9996)

Version 9 flow records

388 flow exported in 63 udp datagrams

0 flows failed due to lack of export packet

0 export packets were sent up to process level

0 export packets were dropped due to no fib

0 export packets were dropped due to adjacency issues

0 export packets were droped due to fragmentation failures

0 export packets were dropped due to encapsulation fixup failures

## ნაწილი №3: NetFlow-ს ანალიზი CLI-ს გამოყენებით

მესამე ნაწილში თქვენ შექმნით მონაცემთა ტრაფიკს **R1** და **R3** მარშრუტიზატორებს შორის, NetFlow-ს ტექნოლოგიაზე დაკვირვებისათვის.

### პირველი ეტაპი: მონაცემთა ტრაფიკის შექმნა R1 და R3 მარშრუტიზატორებს შორის.

ა. დაკავშირდით **Telnet**-ით **R1**-დან **R3**-ში 1**92.168.3.1 IP** მისამართის გამოყენებით. შეიყვანეთ პაროლი cisco მომხმარებლის **EXEC** რეჟიმში შესასვლელად. შეიყვანეთ პაროლი **class**, გლობალური **EXEC** რეჟიმის ჩასართავად. გაუშვით **show run** ბრძანება რაიმე **Telnet** ტრაფიკის შესაქმნელად. დატოვეთ **Telnet** სესია აქტიური.

ბ. **R3** მარშრუტიზატორიდან გაუშვით **ping 192.168.1.1 repeat 1000** ბრძანება, **R1 G0/0** ინტერფეისის „დასაპინგად“. ეს შექმნის **ICMP** ტრაფიკს **R2** მარშრუტიზატორზე.

გ. **PC-A**-დან დაათვალიერეთ **R3** მარშტურიზატორი 1**92.168.3.1 IP** მისამართის გამოყენებით. შედით როგორც **admin** მომხმარებელი **cisco** პაროლთან ერთად. დატოვეთ ბრაუზერი გახსნილი, მას შემდეგ რაც შეხვალთ **R3** მარშრუტიზატორში.

შენიშვნა: დარწმუნდით რომ **pop-up blocker** გათიშულია თქვენს ბრაუზერში.

**მეორე ეტაპი: NetFlow-ს შემაჯამებელი ანგარიშის სტატისტიკის ჩვენება.**

**R2** მარშრუტიზატორზე გაუშვით **show ip cache flow** ბრძანება, **NetFlow** შემაჯამებელი მონაცემების ცვლილებების სანახავად, პაკეტის ზომის განაწილების, **IP** ნაკადის ინფორმაციის, დაჭერილი პროტოკოლების და ინტერფეისის აქტივობის ჩათვლით. მიაქციეთ ყურადღება, რომ პროტოკოლები ახლა ნაჩვენებია შემაჯამებელ მონაცემებში.

R2# **show ip cache flow**

IP packet size distribution (5727 total ackets):

| 1-32 | 64 | 96 | 128 | 160 | 192 | 224 | 256 | 288 | 320 | 352 | 384 | 416 | 448 | 480 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| .000 | .147 | .018 | .700 | .000 | .001 | .001 | .001 | .001 | .011 | .009 | .001 | .002 | .000 | .001 |

| 512 | 544 | 576 | 1024 | 1536 | 2048 | 2560 | 3072 | 3584 | 4096 | 4608 |
|---|---|---|---|---|---|---|---|---|---|---|
| .001 | .001 | .097 | .000 | .000 | .000 | .000 | .000 | .000 | .000 | .000 |

IP Flow Switching cache, 278544 bytes

2 active, 4094 inactive, 114 added

1546 ager polls, 0 flow alloc failures

Active flows timeout in 15 seconds

IP Sub Flow Cache, 34056 bytes

0 active, 1024 inactive, 112 added, 112 added to flow

0 alloc failures, 0 force free

1 chunk, 1 chunk added

last clearing of statistics 00:07;35

| Protocol | total | Flows | Packets | Bytes | Packets | Active (sec) | Idle (sec) |
|---|---|---|---|---|---|---|---|
| --------- | Flows | /sec | /flow | /pkt | /Sec | /Flow | /Flow |
| TCP-Telnet | 4 | 0.0 | 27 | 43 | 0.2 | 5.0 | 15.7 |
| TCP-WWW | 104 | 0.2 | 14 | 275 | 3.4 | 2.1 | 1.5 |
| ICMP | 4 | 0.0 | 1000 | 100 | 8.8 | 27.9 | 15.4 |

| SrcIf | SrcIPaddress | DstIf | DstIPaddress | Pr SrcP DstP | Pkts |
|---|---|---|---|---|---|
| Total: | 112 | 0.2 | 50 146 12.5 | 3.1 | 2.5 |
| SrcIf | SrcIPaddress | DstIf | DstIPaddress | pr SrcP DstP | Pkts |
| Se0/0/0 | 192.168.12.1 | Null | 224.0.0.5 | 59 0000 0000 | 43 |
| Se0/0/1 | 192.168.23.2 | Null | 224.0.0.5 | 59 0000 0000 | 40 |

**მესამე ეტაპი: Telnet და ბრაუზერის სესიების დახურვა**

ა. გაუშვით **exit** ბრძანება **R1** მარშრუტიზატორზე, რათა გაითიშოს **R3** მარშრუტიზატორის **Telnet** სესიიდან გასათიშად.

ბ. დახურეთ ბრაუზერის სესია **PC-A** კომპიუტერზე

**მეოთხე ეტაპი: NetFlow სააღრიცხვო სტატისტიკის გასუფთავება**

ა. **R2** მარშრუტიზატორზე გაუშვით **clear ip flow stats** ბრძანება, **NetFlow** სააღრიცხვო სტატისტიკის გასასუფთავებლად.

R2# **clear ip flow stats**

ბ. ხელახლა გაუშვით **show ip cache flow** ბრძანება რათა შემოწმდეს **Netflow** სააღრცხვო სტატისტიკა განულდა თუ არა. მიაქციეთ ყურადღება, რომ თუ თქვენ აღარ შექმნით მონაცემებს **R2**-ის გავლით, მაშინ მონაცემები არჩეული იქნება **NetFLow**-ს მიერ.

ქვემოთ მოცემულ მაგალითში, ადრესატის მისამართი ამ ტრაფიკისათვის არის მულტიკასტმისამართი 224.0.0.5, ან **OSPF LSA** მონაცემი.

R2# **show ip cache flow**

IP packet size distribution (124 total packets) :

| 1-32 | 64 | 96 | 128 | 160 | 192 | 224 | 256 | 288 | 320 | 352 | 384 | 416 | 448 | 480 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| .000 | .000 | 1.00 | .000 | .000 | .000 | .000 | .000 | .000 | .000 | .000 | .000 | .000 | .000 | .000 |

| 512 | 544 | 576 | 1024 | 1536 | 2048 | 2560 | 3072 | 3584 | 4096 | 4608 |
|---|---|---|---|---|---|---|---|---|---|---|
| .000 | .000 | .000 | .000 | .000 | .000 | .000 | .000 | .000 | .000 | .000 |

IP Flow Switching cache, 278544 bytes

2 active, 4094 inactive, 2 added

1172 ager polls, 0 flow allow alloc failures

Active flows timeout in 30 minutes

Inactive flows timeout in 15 seconds

IP Sub Flow Cache, 34056 bytes

2 active, 1022 inactive, 2 added, 2 added to flow

0 alloc failures, 0 force free

1 chunk, 0 chunks added

last clearing of statistics 00:09:48

| Protocol --------- | total Flows | Flows /sec | Packets /flow | Bytes /pkt | Packets /Sec | Active (sec) /Flow | Idle (sec) /Flow |
|---|---|---|---|---|---|---|---|
| IP-other Total: | 2 | 0.0 | 193 | 79 | 0.6 | 1794.8 | 5.7 |
| | 2 | 0.0 | 193 | 79 | 0.6 | 1794.8 | 5.7 |

| SrcIf | SrcIPaddress | DstIf | DstIPaddress | Pr SrcP DstP | Pkts |
|---|---|---|---|---|---|
| Se0/0/0 | 192.168.12.1 | Null | 224.0.0.5 | 59 0000 0000 | 35 |

| SrcIf | SrcIPaddress | DstIf | DstIPaddress | pr SrcP DstP | Pkts |
|---|---|---|---|---|---|
| Se0/0/1 | 192.168.23.2 | Null | 224.0.0.5 | 59 0000 0000 | 33 |

**ნაწილი №4: NetFlow კოლექტორისა და ანალიზერის პროგრამული უზრუნველყოფის შესწავლა.**

ხელმისაწვდომია მრავალი მწარმოებლის **NetFlow** კოლექტორი და ანალიზერი პროგრამული უზრუნველყოფა. ზოგიერთი პროგრამული უზრუნველყოფა მოწოდებულია უფასოდ, სხვები არა. ქვემოთ მოცემული ბმული გვაწვდის **NetFlow**-ს ზოგიერთი უფასო პროგრამული უზრუნველყოფის შემაჯამებელ ვებ-გვერდს:

http://www.cisco.com/en/US/prod/iosswrel/ps6537/ps6555/ps6601/networking_solutions_products_genericcontent0900aecd805ff72b.html

დაათვალიერეთ მოცემული ვებ-გვერდი, რათა გაეცნოთ **NetFlow** კოლექტორისა და ანალიზატორის ზოგიერთ ხელმისაწვდომ პროგრამულ უზრუნველყოფას.

**ასახვა (Reflection)**

1. რა დანიშნულება აქვს **NetFlow** კოლექტორ პროგრამულ უზრუნველყოფას?

_____

_____

2. რა დანიშნულება აქვს **NetFlow** ანალიზატორ პროგრამულ უზრუნველყოფას?

_____

_____

3. ნაკადების გასარჩევად, რომელი შვიდი კრიტიკული ველი გამოიყენება ორიგინალი **NetFlow**-ს მიერ.

_____

_____

მარშრუტიზატორის ინტერფეისის შემაჯამებელი ცხრილი:

| მარშრუტიზატორის ინტერფეისის შეჯამება | | | | |
|---|---|---|---|---|
| მარშრუტიზატორის მოდელი | Ethernet interface №1 | Ethernet interface №2 | Serial interface №1 | Serial interface №2 |
| 1800 | FastEthernet 0/0 (F0/0) | FastEthernet 0/1 (F0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |
| 1900 | GigabitEthernet 0/0 (G0/0) | GigabitEthernet 0/1 (G0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |
| 2801 | FastEthernet 0/0 (F0/0) | FastEthernet 0/1 (F0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/1/1 (S0/1/1) |
| 2811 | FastEthernet 0/0 (F0/0) | FastEthernet 0/1 (F0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |
| 2900 | GigabitEthernet 0/0 (G0/0) | GigabitEthernet 0/1 (G0/1) | Serial 0/0/0 (S0/0/0) | Serial 0/0/1 (S0/0/1) |

**შენიშვნა:** თუ გინდათ მოძებნოთ თუ როგორაა მარშრუტიზატორი კონფიგურირებული, დაათვალიერეთ ინტერფეისები რათა მოახდინოთ მარშრუტიზატორის იდენტიფიკაცია და დაადგინოთ რამდენი ინტერფეისი აქვს მას. აქ არ არის იმის საშუალება, რომ ჩამოიწეროს თითოეული მარშრუტიზატორის კონფიგურაციების კომბინაციები. მოცემული ცხრილი მოიცავს მოწყობილობის **Ethernet** და **Serial** ინტერფეისების შესაძლო კომბინაციების იდენტიფიკატორებს. ეს ცხრილი არ მოიცავს ნებისმიერი სხვა ტიპის ინტერფეისს, რადგან კონკრეტული მარშრუტიზატორი შეიძლება შეიცავდეს მხოლოდ ერთს. ამის მაგალითი შეიძლება იყოს **ISDN BRI** ინტერფეისი. ფრჩხილებში მოცემული ცვლადი არის ლეგალური აბრევიატურა, რომელიც შეიძლება გამოყენებულ იქნას **Cisco IOS** ბრძანებებში, ინტერფეისის წარმოსადგენად.

## 5.3. DHCP პროტოკოლის კონფიგურირება.

ქსელში ჩართულ ყველა მოწყობილობას უნდა ჰქონდეს უნიკალური IP მისამართი. ქსელის ადმინისტრატორებს შეუძლიათ სტატიკურად(ხელით) დაუნიშნონ IP მისამართები ქსელურ მოწყობილობებს. ეს მეთოდი ეფექტურია როცა, როგორც წესი ქსელში ჩართული მოწყობილობების ადგილმდებარეობა არ იცვლება(როგორც ფიზიკურად ასევე ლოგიკურად) და მათ უნდა ჰქონდეს უცვლელი მისამართები. მისამართების სტატიკურად დანიშვნა ასევე უადვილებს ქსელის ადმინისტრატორებს, ჰქონდეთ წვდომა დაშორებულ ქსელურ მოწყობილობებთან, რამეთუ მათ ადვილად შეუძლიათ შესაბამისი მოწყობილობის IP მისამართის განსაზღვრა.

თუმცა, როცა ორგანიზაციაში მოხმარებლები და მათი კომპიუტერები ხშირად იცვლის ლოკაციას ფიზიკურად ან ლოგიკურად, ერთობ ძნელია ხშირი გადაადგილების შემთხვევაში ყოველ ჯერზე მისამართების სტატიკურად შეცვლა. შესაბამისად დგება IP მისამართების ავტომატურად მიღება-მინიჭების აუცილებლობა, რაც შესაძლებელია კვანძის დინამიურად კონფიგურირების DHCP(Dynamic Host Configuration Protocol) პროტოკოლის მეშვეობით. ცენტრალიზებული DHCP სერვერის გამოყენება, აძლევს ორგანიზაციას შესაძლებლობას მოახდინოს IP მისამართების დინამიური ადმინისტრირება, ერთი ცალკე აღებული სერვერიდან.

DHCP ხელმისაწვდომია როგორც IPv4 (DHCPv4) ასევე IPv6 (DHCPv6) სტანდარტის IP მისამართებისთვის.

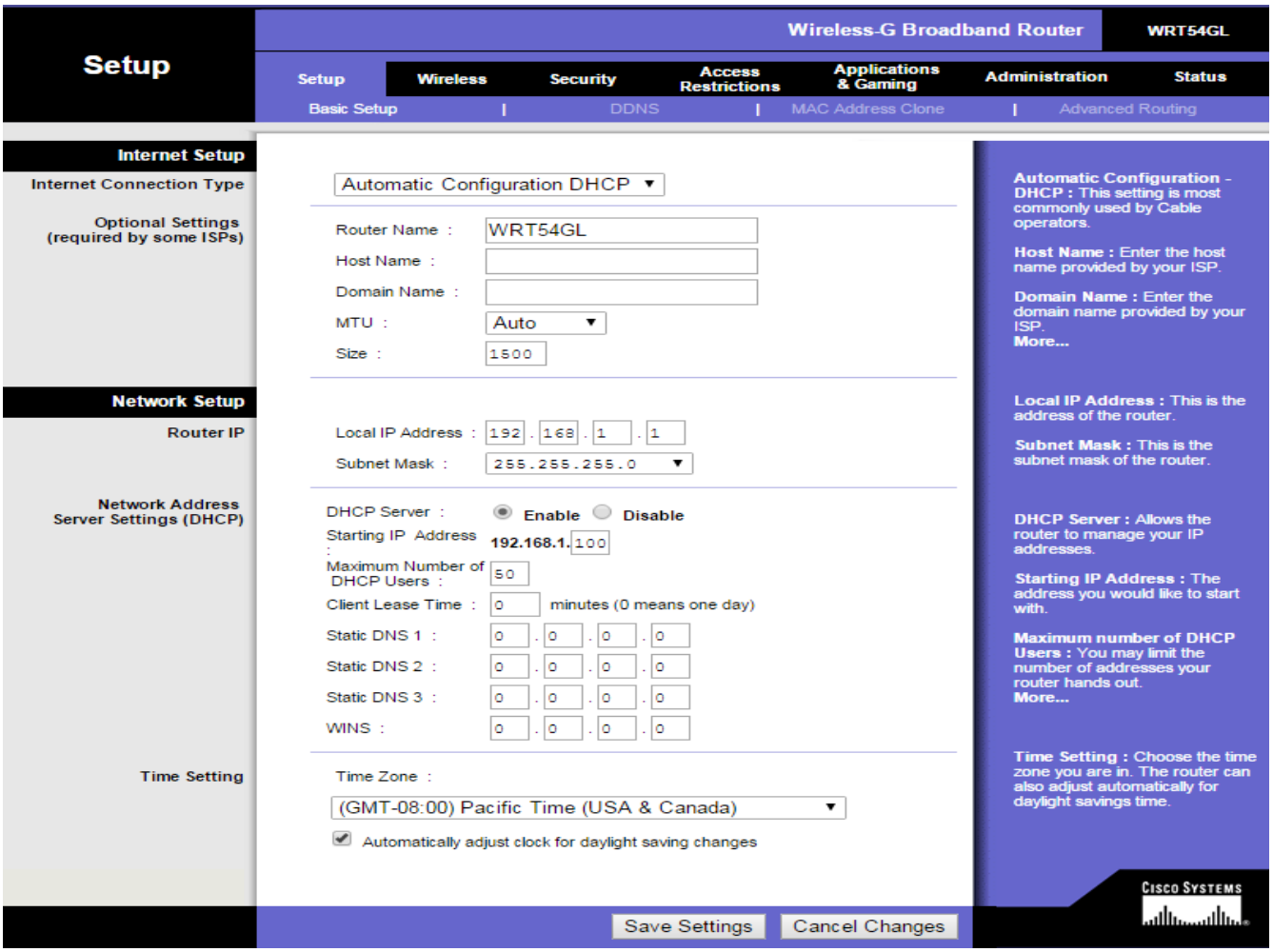
ქვემოთ მოცემული ბმულის მეშვეობით თქვენ შეგიძლიათ შეისწავლოთ DHCP კონფიგურირების მაგალითი Linksys wireless 54GL router მარშრუტიზატორის მაგალითზე:

http://ui.linksys.com/WRT54GL/4.30.0/Setup.htm

მოცემული ფანჯრის მიხედვით (იხ. სურ) გრაფიკულ რეჟიმში, თქვენ შეგიძლიათ კონფიგურირებით განსაზღვროთ:

- Host Name (ქსელური სახელი)
- მოცემული მოწყობილობის ლოკალური IP მისამართი და ქვექსელის ნიღაბი

- DHCP ჩართვა  ან გამორთვა
- საწყისი IP მისამართი
- მომხმარებელთა მაქსიმალური რიცხვი, რომელიც მიიღებს IP მისამართს DHCP პროტოკოლით
- მინიჭების დრო (Lease Time)




*სურ.5.3. 1*

პრაქტიკული სამუშაო

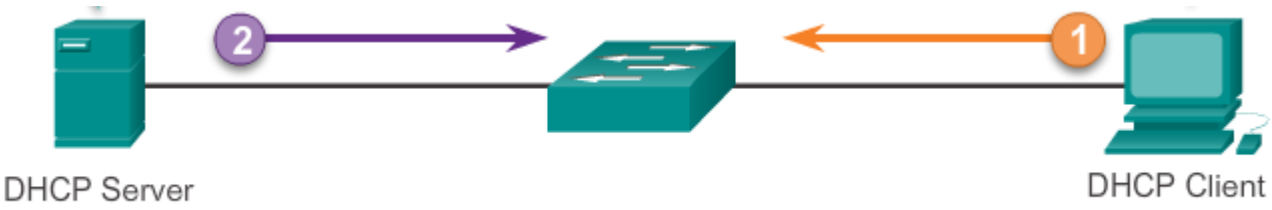
- მოცემულ ბმულით გახსენით კონფიგურირების ფანჯარა: http://ui.linksys.com/WRT54GL/4.30.0/Setup.htm
- შეურჩიეთ მოწყობილობას ქსელური სახელი(HostName) – თქვენი სახელი
- მოწყობილობის IP მისამართად(Local IP Address) განსაზღვრეთ - C კლასის ლოკალური ქსელის მისამართებიდან, ბოლო ქსელის პირველი შესაძლო ჰოსტის მისამართი
- შეარჩიეთ ქვექსელის ნიღაბი(Subnet Mask) - რომელიც იძლევა ქსელში 126 ჰოსტის ჩართვის საშუალებას
- გააქტიურეთ DHCP სერვერი
- საწყის IP მისამართად(Starting IP Address) შეარჩიეთ - საკუთრივ მოწყობილობის მისამართის შემდგომი IP მისამართი
- მომხმარებელთა მაქსიმალური რიცხვი - 100

DHCP სერვერის შემცველად, მცირე ზომის ქსელებში შესაძლებელია გამოყენებულ იქნას მარშრუტიზატორები, რომლებზედაც კონფიგურირებით მიიღწევა DHCP სერვისების გააქტიურება.

DHCPv4 მოიცავს მისამართების მინიჭების 3 განსხვავებულ მექანიზმს:

- Manual Allocation - ადმინისტრატორი წინასწარ განუსაზღვრავს კლიენტ კომპიუტერს IP მისამართს და ის მას მიეწოდება DHCPv4 სერვისის მეშვეობით
- Automatic Allocation – DHCPv4 ავტომატურად ანიჭებს მისამართებს მოწყობილობებს წინასწარ შერჩეული დიაპაზონიდან მუდმივი ვადით.
- Dynamic allocation - ყველაზე ხშირად გამოყენებული მექანიზმია, რომლის მიხედვითაც კლიენტი კომპიუტერი, გარკვეული - ადმინისტრირებით განსაზღვრული ვადით იღებს მისამართებს და ამ ვადის გავლის შემდგომ (Lease

Time), რომელიც შეიძლება იყოს 24 საათი, 1 კვირა და მეტი, ხდება ხელახალი მოთხოვნის ფორმირება, რომლის შესაბამისადაც ხდება ხელახალი დამისამართება, როგორც წესი იმავე მისამართებით.




*სურ.5.3. 2*

## DHCPv4 შეტყობინების ფორმატი

| 8 | 16 | 24 | 32 |
|---|---|---|---|
| OP Code (1) | Hardware Type (1) | Hardware Address Length (1) | Hops (1) |
| Transaction Identifier | | | |
| Seconds - 2 bytes | | Flags - 2 bytes | |
| Client IP Address (CIADDR) - 4 bytes | | | |
| Your IP Address (YIADDR) - 4 bytes | | | |
| Server IP Address (SIADDR) - 4 bytes | | | |
| Gateway IP Address (GIADDR) - 4 bytes | | | |
| Client Hardware Address (CHADDR) - 16 bytes | | | |
| Server Name (SNAME) - 64 bytes | | | |
| Boot Filename - 128 bytes | | | |
| DHCP Options - variable | | | |

*სურ.5.3. 3*

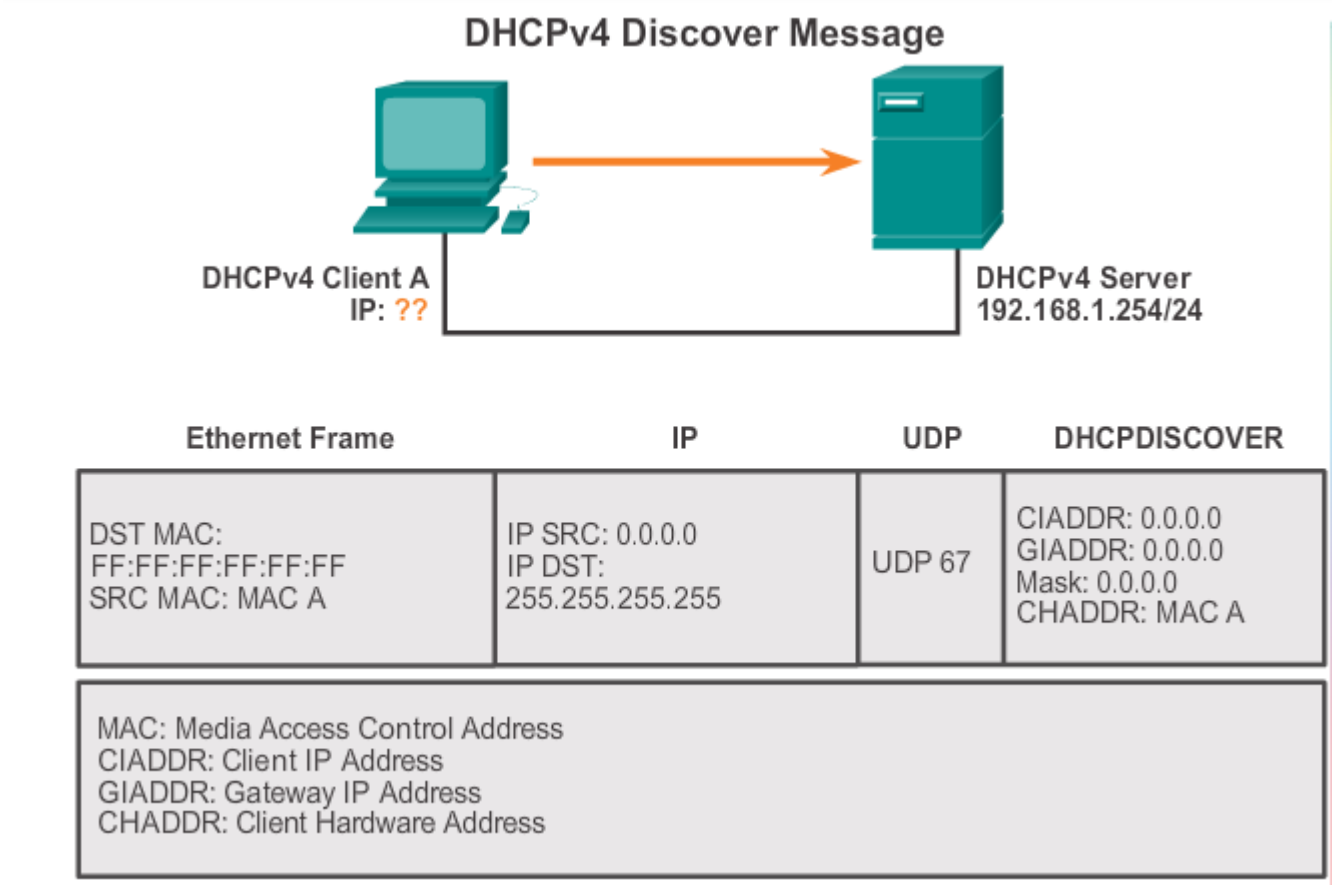
სურათზე 5.3.3 ნაჩვენებია DHCPv4 შეტყობინების ფორმატის მაგალითი, რომელშიც შესაბამისი ველები აღწერენ:

- Operation (OP) Code - მიუთითებს შეტყობინების ტიპს. მნიშვნელობა 1 - უჩვენებს შეტყობინება-მოთხოვნას, მნიშვნელობა 2 - საპასუხო შეტყობინებას

- Hardware Type - მიუთითებს ქსელში გამოყენებული მოწყობილობის ტიპს, მაგ.: 1 - Ethernet; 15 – Frame Relay; 20 – მიმდევრობითი Serial ხაზი
- Hardware Address Length - მიუთითებს მისამართების სიგრძეს
- Hops - მიუთითებს მისამართების გადგზავნის კონტროლზე.
- Transaction Identifier - გამოიყენება კლიენტის მიერ, სერვერიდან მიღებული შეტყობინების მოთხოვნასთან შესაბამისობის დასადგენად
- Seconds - განსაზღვრავს კლიენტის ახალი ან ხელახალი მოთხოვნიდან გასულ დროს. გამოიყენება DHCPv4 სერვერის მიერ ერთდროული მართვისას, პრიორიტეტის განსაზღვრის მიზნით.
- Flags - გამოიყენება კლიენტის მიერ, რომელმაც არ იცის საკუთარი მისამართი, როდესაც ის აგზავნის მოთხოვნას. მოცემული ველის მნიშვნელობა - 1 , მიუთითებს სერვერს, რომ პასუხი გაგზავნილ უნდა იქნას Broadcast - ფართომაუწყებლობითი გზავნილის სახით
- Client IP Address - გამოიყენება კლიენტის მიერ მისამართის ვადის განახლების დროს, მხოლოდ იმ შემთხვევაში თუ მოცემულ მომენტში მას აქვს ვალიდური მისამართი, წინააღმდეგ შემდხვევაში ამ ველში იწერება - 0
- Your IP Address - გამოიყენება სერვერის მიერ კლიენტისთვის Ipv4 მისამართის დანიშვნისთვის
- Server IP Address - გამოიყენება სერვერის მიერ იმ სერვერის მისამართის იდენტიფიკაციისთვის, რომელსაც კლიენტი გამოიყენებს ხელახალი ჩატვირთვისას
- Gateway IP Address - მოცემული მისამართი უზრუნველყოფს კომუნიკაციას DHCPv4 კლიენტსა და სერვერს შორის, როდესაც ისინი იმყოფება განსხვავებულ ქვექსელებსა თუ ქსელებში.
- Client Hardware Address - მიუთითებს კლიენტის ფიზიკურ დონეს.
- Server Name - გამოიყენება სერვერის მიერ DHCPOFFER ან DHCPACK გზავნილებისას

- Boot Filename - კლიენტის მხრიდან გამოიყენება განსაზღვრული ტიპის ფაილის მოთხოვნის დროს DHCPDISCOVER შეტყობინებისას. სერვერის მხრიდან DHCPOFFER გზავნილის დროს, რათა სრულად მიეთითოს ჩამოსათვირთი ფაილის დირექტორია და სახელი

როდესაც კლიენტი კომპიუტერი მომართულია Ipv4 მისამართების დინამიურ რეჟიმში მისაღებად და სურს ქსელში ჩართვა, ის ითხოვს მისამართებს შესაბამისი DHCP სერვერიდან. კლიენტი კომპიუტერი გადასცემს DHCPDISCOVER შეტყობინებას საკუთარ ლოკალურ ქსელში ჩატვირთვისას ან ქსელური კავშირების შესაძლებლობების დაფიქსირებისას. ვინაიდან კლიენტ კომპიუტერს არ აქვს საკუთარი ქსელის იდენტიფიცირების საშუალება, მოცემული შეტყობინება გახლავთ ფართომაუწყებლობითი (დანიშნულების მისამართია 255.255.255.255). ვინაიდან თვითონ კლიენტს არ აქვს მინიWებული IPv4 მისამართი, შეტყობინების წყაროს(Source) მისამართი იქნება 0.0.0.0




*სურ.5.3. 4*

## განვიხილოთ DHCPv4 სერვერის კონფიგურირების მაგალითი

## ეტაპი 1. მარშრუტიზატორის ბაზისური კონფიგურაცია

### მარშრუტიზატორის ძირითადი კონფიგურირება

*ბრძანებათა ველის რეჟიმები*

**სამომხმარებლო რეჟიმი (Line Console)**

– Router >

**პრივილეგირებული რეჟიმი (Exec Mode)**

– Router #

**პრივილეგირებულ რეჟიმში შესვლა**

– Router > enable

– Router #

**პრივილეგირებული რეჟიმიდან გამოსვლა**

– Router # exit (ან desible)

– Router >

**გლობალური კონფიგურაციის რეჟიმი**

– Router > enable

– Router # config t

– Router(config)#

**ინტერფეისის კონფიგურირების რეჟიმი**

– Router > enable

– Router # config t

– Router(config)#interface ტიპი ნომერი (მაგ.:interface FastEthernet 0/0)

– Router(config-if)#

### მოწყობილობისთვის სახელის მინიჭება

– Router > enable

– Router # config t

– Router(config)#Hostname სახელი (მაგ.: Hostname CiscoRouter)

– CiscoRouter(config)#

### პრივილეგირებული რეჟიმის პაროლით დაცვა

• არაშიფრირებული პაროლი

– Router > enable

– Router # config t

– Router(config)#enable password პაროლი

• დაშიფრული პაროლი

– Router > enable

– Router # config t

– Router(config)#enable secret პაროლი

### *ბრძანებათა კონსოლის დაცვა*

– Router > enable

– Router # config t

– Route(config)# line console 0

– Router(config)# password პაროლი

– Router(config)# login

*ვირტუალური ტერმინალის პორტების დაცვა*

– Router > enable

– Router # config t

– Route(config)# line vty 0 4

– Router(config)# password პაროლი

– Router(config)# login

**პაროლების დაშიფრვა**

– Router > enable

– Router # config t

– Route(config)# service password encryption

**მიმდინარე პარამეტრების საწყის-ჩამტვირთავ პარამეტრებად შენახვა**

– Router > enable

– Router# copy runn start

**ინტერფეისის კონფიგურირება**

ქვემოთ მოცემულია Fastethernet და Serial ინტერფეისებზე IP და ქვექსელის ნიღაბის მისამართის და ინტერფეისის გააქტიურების მაგალითი

```
Router(config)#interface fastethernet 0/0
Router(config-if)#description connection to Admin LAN
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/0
Router(config-if)#description connection to Router2
Router(config-if)#ip address 192.168.1.125 255.255.255.0
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown
```

*DHCP სერვისის კონფიგურირება*

**ნაბიჯი I (DHCP მისამართების სივრცის შექმნა)**

– Router > enable

– Router # config t

– Route(config)#ip dhcp pool სახელი

**ნაბიჯი II (ქსელის და ქვექსელის ნიღაბის მისამართების მითითება)**

– Router > enable

– Router # config t

– Router(config)#ip dhcp pool სახელი

– Router(dhcp-config)# network (მაგ.:) 192.168.1.0  255.255.255.0

**ნაბიჯი III (IP მისამართების გამორიცხვა-დარეზერვება)**

– Router > enable

– Router # config t

– Route(config)#ip dhcp excluded-address (მაგ.:192.168.1.1 192.168.1.49)

**ნაბიჯი IV (DNS სერვერის დამისამართება)**

– Router > enable

– Router # config t

– Router(config)#ip dhcp pool სახელი

– Router(dhcp-config)# network 192.168.1.0 255.255.255.0

– Router(dhcp-config)#dns-server (მაგ.:)192.168.1.10

**ნაბიჯი V (Gateway დამისამართება)**

– Router > enable

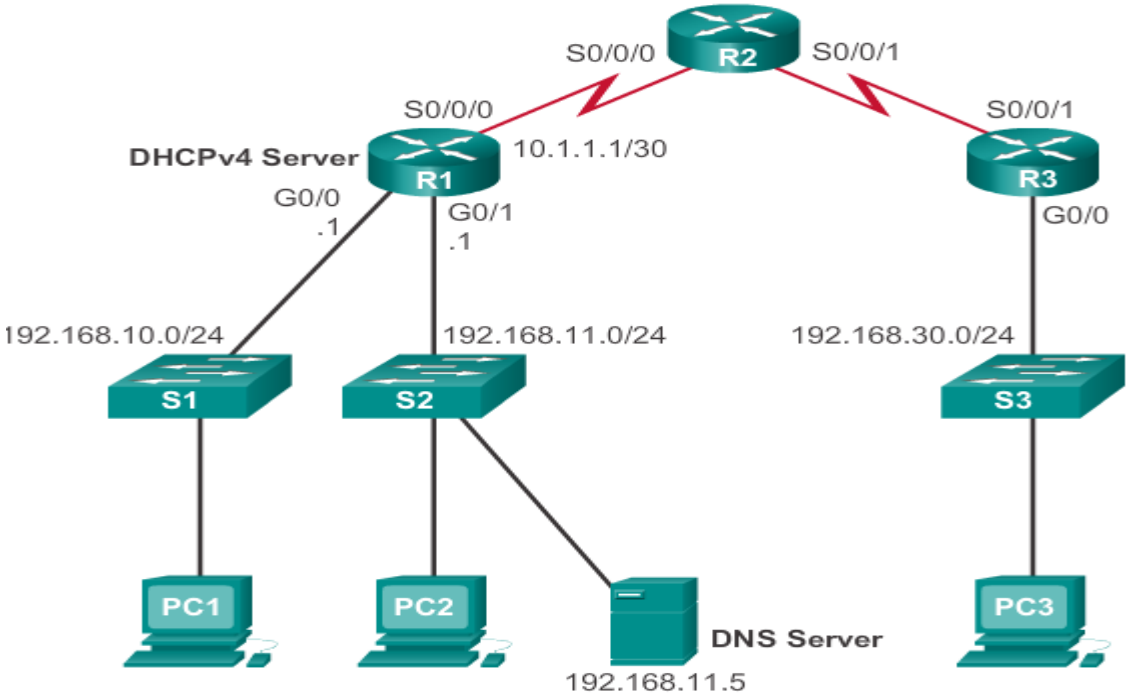
– Router # config t

– Router(config)#ip dhcp pool სახელი

– Router(dhcp-config)# network 192.168.1.0 255.255.255.0

– Router(dhcp-config)#default-router (მაგ.:) 192.168.1.1

*DHCPv4 კონფიგურირების შემოწმება*




*სურ.5.3. 5*

სურ.5.3.5-ზე მოცემულია ქსელის ფიზიკური და ლოგიკური მოდელი, R1 მარშრუტიზატორი კონფიგურირებულია როგორც DHCPv4 სერვერი. **show running-config | section dhcp** ბრძანება გამოგვიტანს შესაბამისი კონფიგურაციის ჩანაწერს, რომელსაც აქვს სურ.5.3.6-ზე მოცემულის შესაბამისი სახე

```
R1# show running-config | section dhcp
ip dhcp excluded-address 192.168.10.1 192.168.10.9
ip dhcp excluded-address 192.168.10.254
ip dhcp excluded-address 192.168.11.1 192.168.11.9
ip dhcp excluded-address 192.168.11.254
ip dhcp pool LAN-POOL-1
 network 192.168.10.0 255.255.255.0
 default-router 192.168.10.1
 dns-server 192.168.11.5
 domain-name example.com
ip dhcp pool LAN-POOL-2
 network 192.168.11.0 255.255.255.0
 default-router 192.168.11.1
 dns-server 192.168.11.5
 domain-name example.com
R1#
```

*სურ.5.3. 6*

სურ. 5.3.6-ზე მოცემული ჩანაწერების მიხედვით უპასუხეთ შემდეგ კითხვებს:

1. რა მისამართებია დარეზერვებული?
2. რამდენი მისამართების სივრცეა(POOL) შექმნილი და რა სახელწოდების?
3. რომელ სივრცეს(POOL) - რა ქსელის მისამართები შეესაბამება?
4. რომელია შესაბამისი სივრცის Gateway და DNS სერვერის მისამართები?

**show ip dhcp binding** ბრძანება გამოგვიტანს ინფორმაციას DHCPv4 სერვისით მინიჭებულ IP მისამართზე და შესაბამის ფიზიკურ MAC მისამართზე.

```
R1# show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address     Client-ID/          Lease expiration     Type
               Hardware address/
               User name
192.168.10.10  0100.e018.5bdd.35   May 28 2013 01:06 PM Automatic
192.168.11.10  0100.b0d0.d817.e6   May 28 2013 01:10 PM Automatic
```

*სურ.5.3. 7*

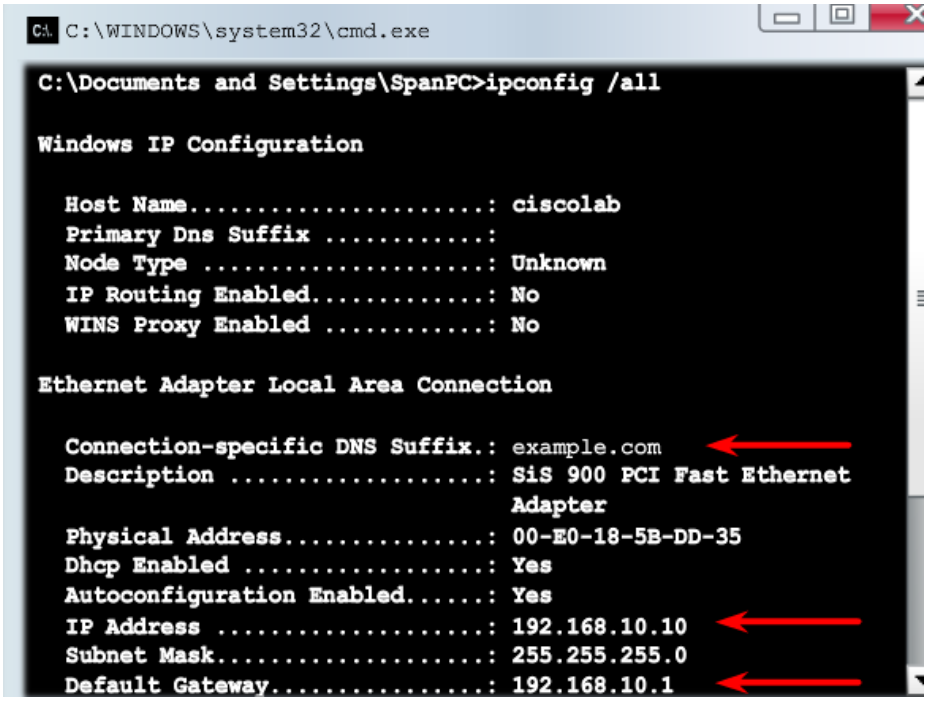
**Show ip dhcp server statistics** ბრძანება გამოგვიტანს ინფორმაციას მიღებულ და გაგზავნილ DHCPv4 შეტყობინებებზე

```
R1# show ip dhcp server statistics
Memory usage         25307
Address pools        2
Database agents      0
Automatic bindings   2
Manual bindings      0
Expired bindings     0
Malformed messages   0
Secure arp entries   0

Message              Received
BOOTREQUEST          0
DHCPDISCOVER         8
DHCPREQUEST          3
DHCPDECLINE          0
DHCPRELEASE          0
DHCPINFORM           0
```

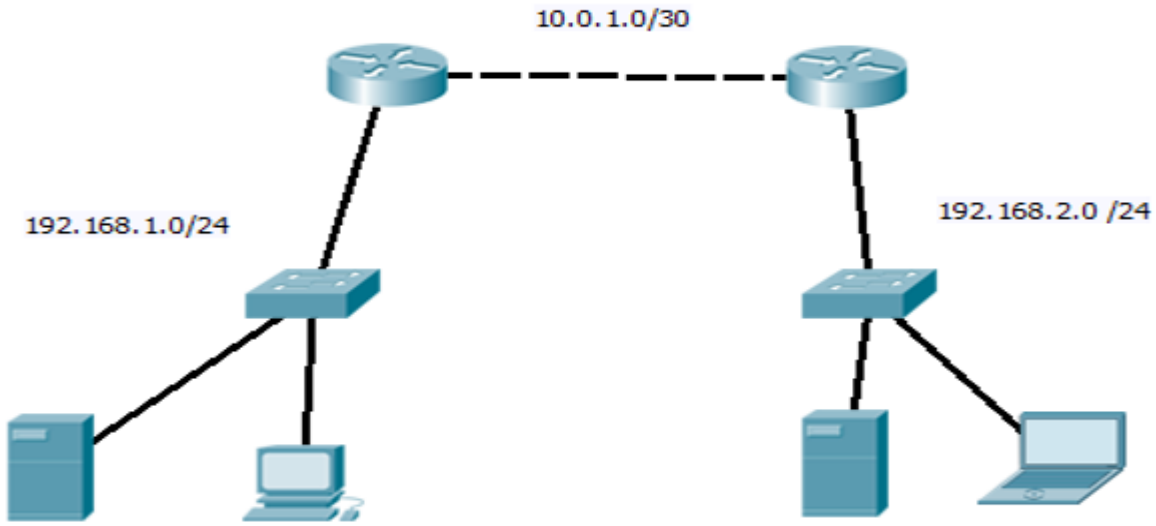
*სურ.5.3. 8*

სურათზე 5.3.9 ნაჩვენებია კლიენტ კომპიუტერზე TCP/IP პარამეტრები, რომელიც წარმოჩინდება ipconfig /all ბრძანების მითითებით.




*სურ.5.3. 9*

*პრაქტიკული სამუშაო*




*სურ.5.3. 10*

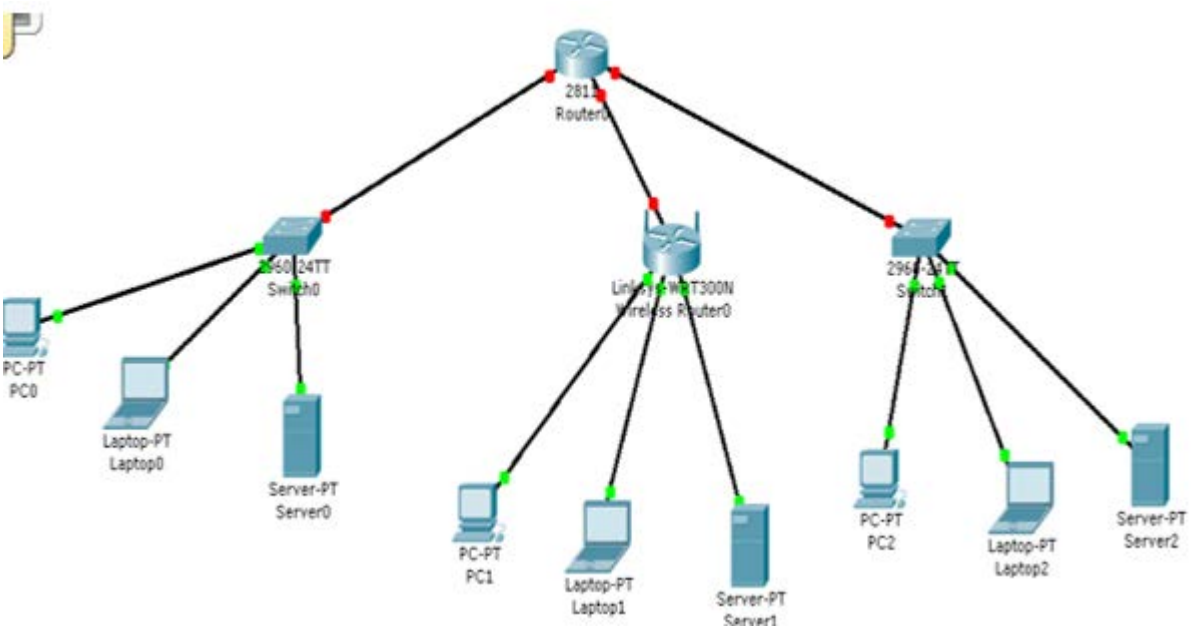
1. შექმენით 5.3.10-ზე მოცემულის შესაბამისი ქსელის ფიზიკური და ლოგიკური მოდელი

2. როუტერების შეურჩიეთ ქსელური სახელი - თქვენი სახელი და გვარი

3. როუტერებზე გააქტიურეთ(შექმენით) ბრძანებათა ველის(Line Console) და პრივილეგირებული რეჟიმის(Enable) პაროლები - თქვენი გვარი

4. მე-2 როუტერზე გააქტიურეთ DHCP კონფიგურაცია, რომელიც დაარიგებს შესაბამისი ქსელის ჰოსტებზე IP, Gateway და DNS მისამართებს

*პრაქტიკული სამუშაო*




*სურ.5.3. 11*

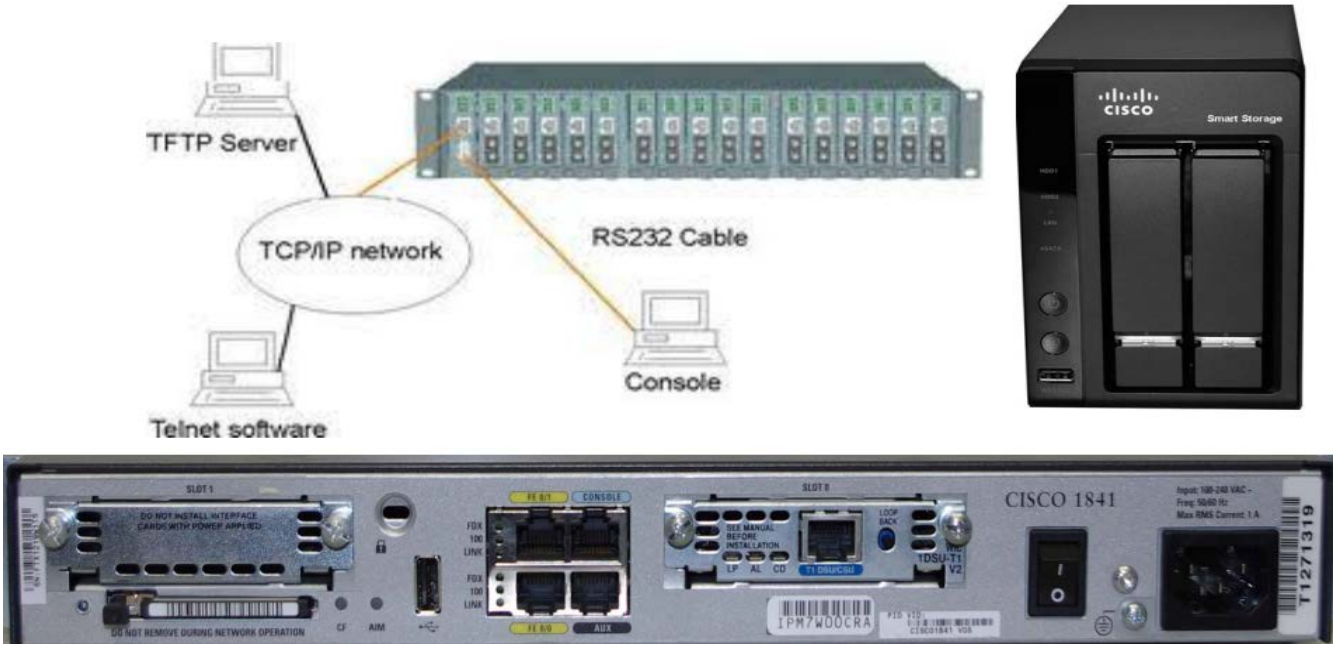
1. შექმენით სურ.5.3.11-ზე მოცემულის შესაბამისი ქსელის მოდელი
2. მარცხენა ნაწილში განლაგებულ ქსელის მოდელში, ჩართეთ 157.207.114.78 /27 ქსელის II ქვექსელის საწყისი კვანძები (Host)
3. მე-2 ქსელში უკაბელო კავშირის მოწყობილობას შეურჩიეთ 192.168.250.0 /26 ქსელის პირველი ჰოსტის მისამართი, რომელიც თავის მხრივ DHCP პროტოკოლით დაურიგებს IP მისამართებს მასთან მიერთებულ 3 კვანძს (Host)
4. მე-3 ქსელში 136.25.139.110 /19 მისამართის მქონე სერვერი DHCP უტილიტას მეშვეობით ავტომატურად მიანიჭებს შესაბამის ქსელის მისამართებს იმავე ქსელში ჩართულ 2 კვანძს (Host)
5. მოცემული ქსელები დააკავშირეთ ერთმანეთთან მარშრუტიზატორის (Router) მეშვეობით (თუ საჭირო იქნება ამისთვის მარშრუტიზატორს დაუმატეთ ეზერნეტ პორტი)
6. ყველა კვანძს Gateway მისამართად გაუწერეთ შესაბამისი ქსელის ბოლო კვანძის(Host) მისამართი, შესაბამისი მისამართი გაუწერეთ მარშრუტიზატორის სათანადო ინტერფეისებს...

7. დააკონფიგურირეთ მარშრუტიზატორი, კერძოდ შეურჩიეთ სახელი: თქვენი სახელი
8. პრივილეგირებული რეჟიმი დაიცავით პაროლით (პაროლი- თქვენი სახელი_TEST1)
9. ბრძანებათა ველის ინტერფეისი დაიცავით პაროლით (პაროლი-თქვენი სახელი_TEST1)
10. მიმდინარე კონფიგურირების პარამეტრები შეინახეთ როგორც საწყისი კონფიგურირების პარამეტრი

## 5.4. ქსელის ოპტიმიზაციის პროტოკოლების გამოყენება.

*TFTP სერვერი*

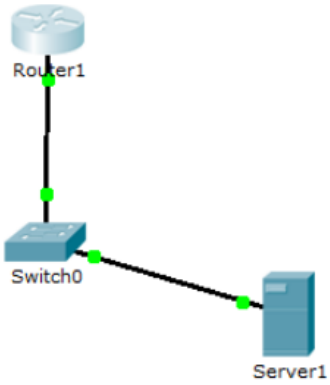
**TFTP სერვერზე კონფიგურირების ფაილის შენახვის აქტუალობა**



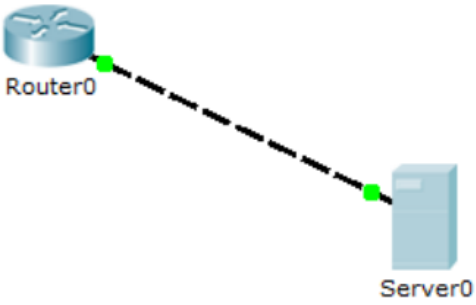

*სურ.5.4. 1*

ასლის შექმნა და შენახვა დაშორებულ კვანძზე (TFTP სერვერი), იძლევა საშუალებას, შემდგომი აუცილებლობის შემთხვევაში (კონფიგურაციის ფაილის წაშლა, ხარვეზები და. ა.შ.) - სწრაფად აღვადგინოთ გამართული კონფიგურაციის პარამეტრები

**ნაბიჯი I – ფიზიკურად დავაკავშიროთ Router-ი და Server-ი**




ან



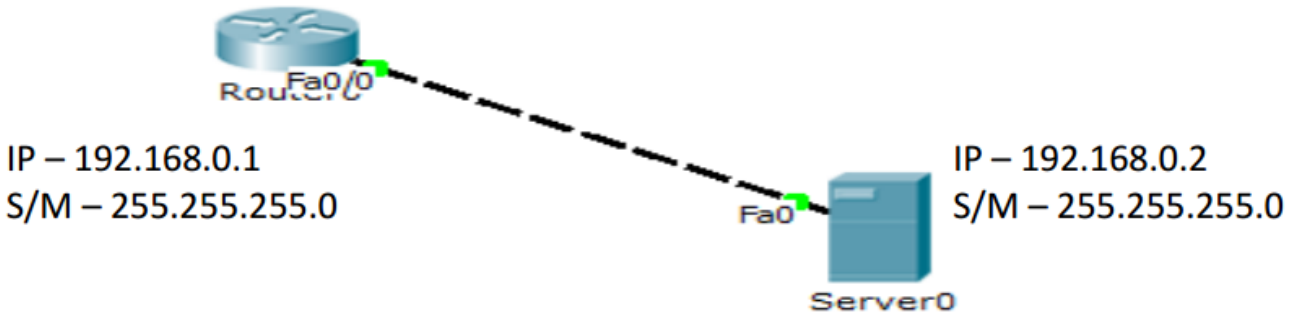

კავშირი Switch-ის მეშვეობით Straight კაბელით

პირდაპირი კავშირი Cross-Over კაბელით

*სურ.5.4. 2*

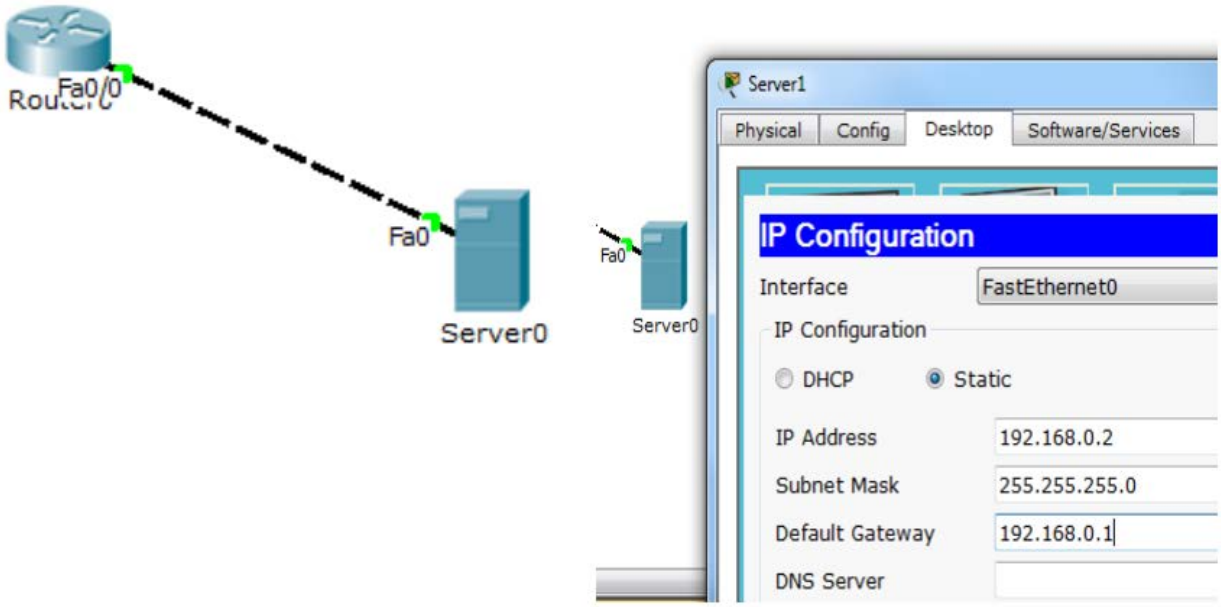
**ნაბიჯი II – ლოგიკურად დავაკავშიროთ Router-ი და Server-ი**

Router-ს და Switch-ს შესაბამის კავშირის ინტერფეისებზე უნდა ჰქონდეთ ერთი და იმავე ქსელის ლოგიკური მისამართები, მაგ.:




*სურ.5.4. 3*

```
Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 192.168.0.1 255.255.255.0
Router(config-if)#no shutdown
```



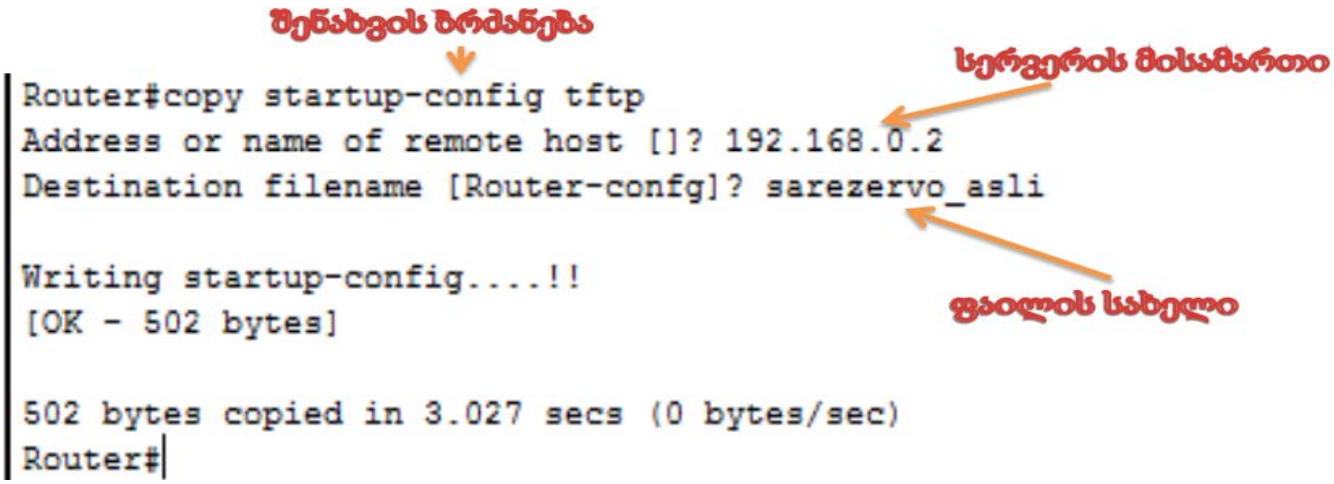

*სურ.5.4. 4*

**ნაბიჯი III– Router-ზე მიმდინარე კონფიგურირების პარამეტრების შენახვა საწყისი კონფიგურირების პარამეტრებად**

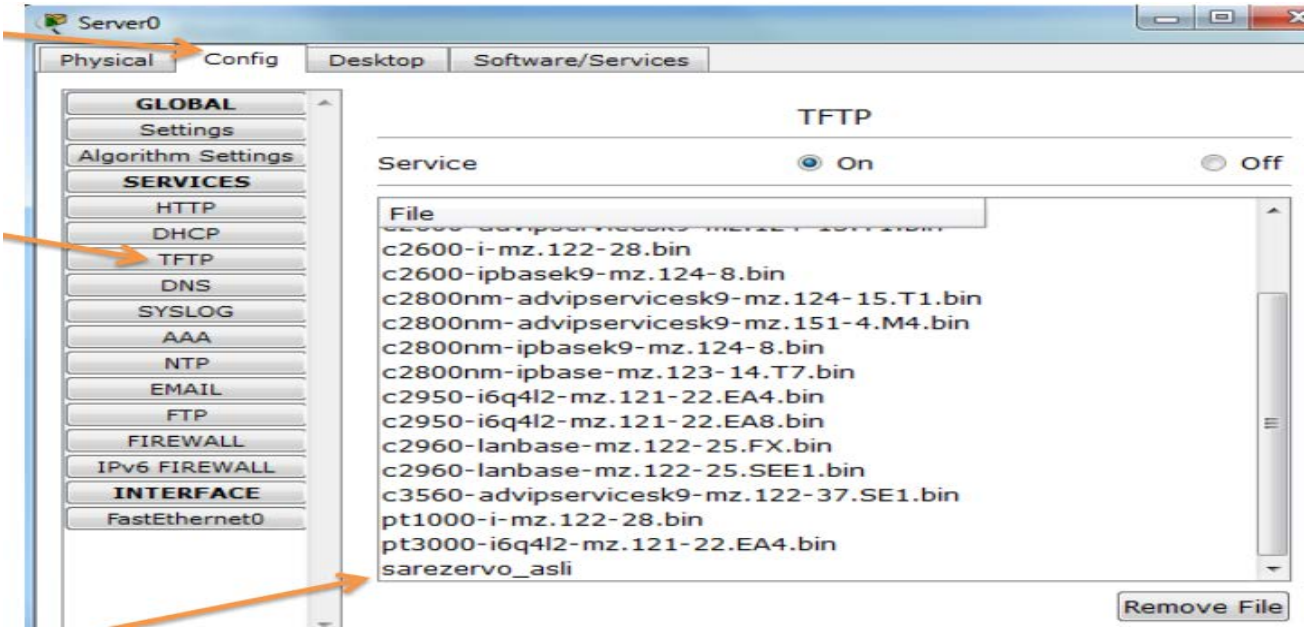
```
Router>ena
Router>enable
Router#copy runn
Router#copy running-config start
Router#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
```

*სურ.5.4. 5*

ნაბიჯი IV – Router-ზე არსებული საწყისი კონფიგურირების პარამეტრების შენახვა TFTP სერვერზე



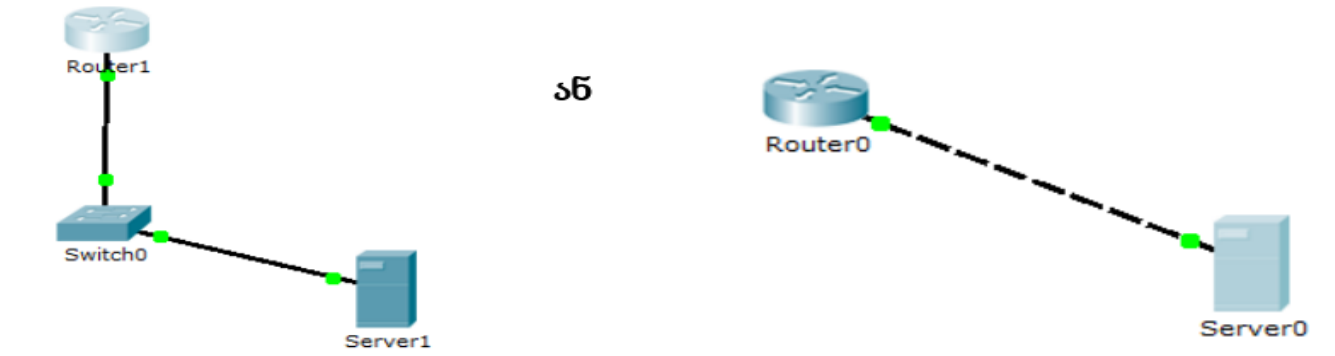

*სურ.5.4. 6*




*სურ.5.4. 7*

კონფიგურაციის პარამეტრების აღდგენა TFTP სერვერზე შენახული ასლიდან

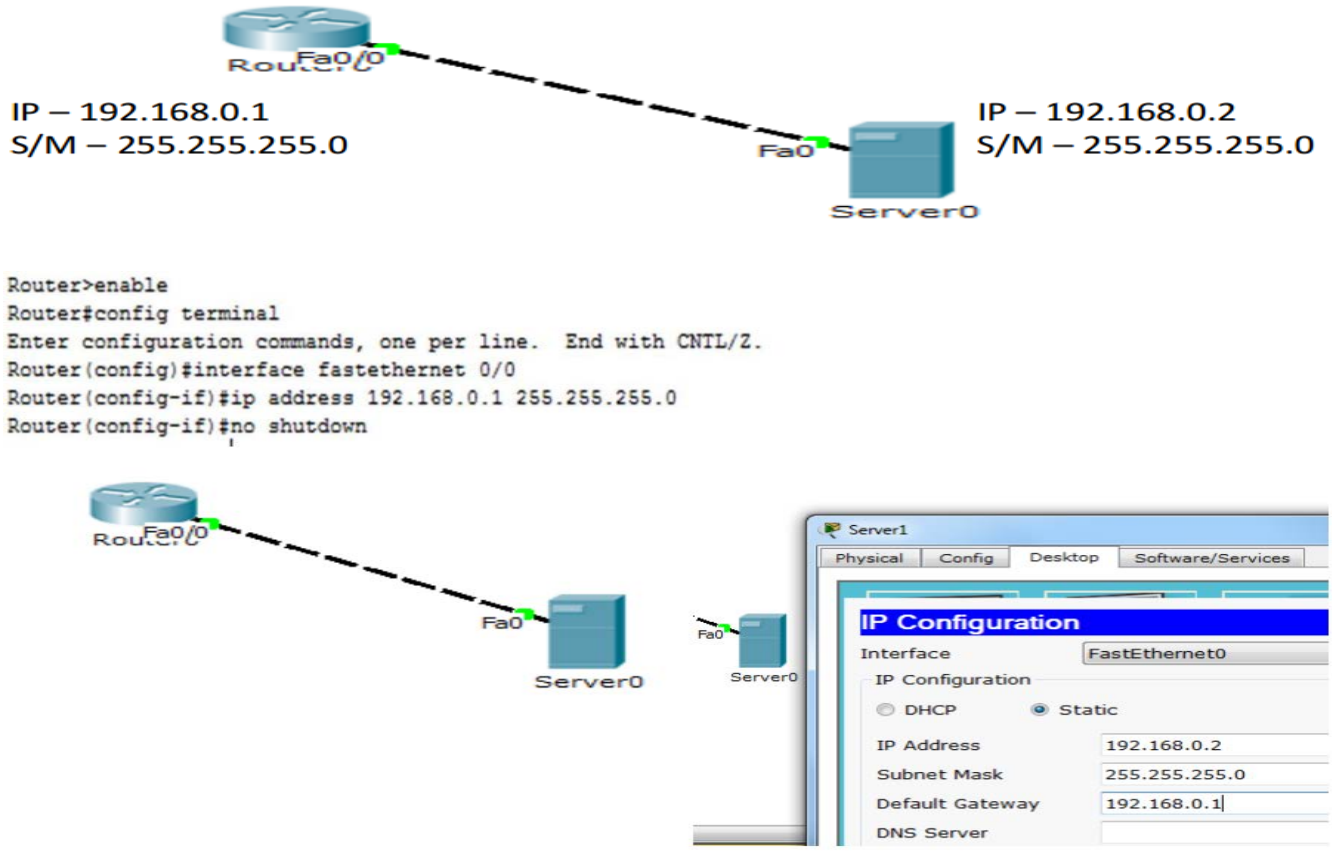
### ნაბიჯი I– ფიზიკურად დავაკავშიროთ Router-ი და Server-ი




*სურ.5.4. 8*

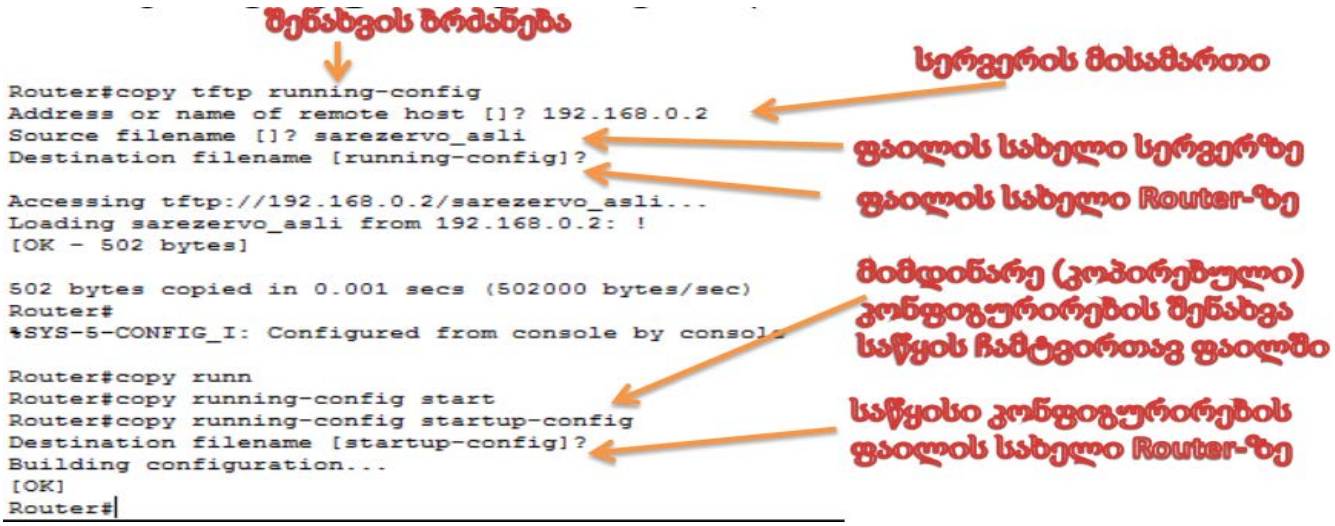
### ნაბიჯი II – ლოგიკურად დავაკავშიროთ Router-ი და Server-ი

Router-ს და Switch-ს შესაბამის კავშირის ინტერფეისებზე უნდა ჰქონდეთ ერთი და იმავე ქსელის ლოგიკური მისამართები, მაგ.:




*სურ.5.4. 9*

**ნაბიჯი III – Server-ზე არსებული კონფიგურირების ფაილის კოპირება Router-ის მიმდინარე კონფიგურირების ფაილში**




*სურ.5.4. 10*

**პროცესზე დაკვირვება**

შემფასებელი აკვირდება შესაფასებელ პირის მუშაობას პროფესიული სტანდარტით(პროგრამით / მოდულით ) განსაზღვრული ამოცანების შესრულების პროცესში . დაკვირვებახორციელდება სამუშაო ადგილზე ან სამუშაო პირობებში, თუმცა დასაშვებია ჩატარდესკომპიუტერებით აღჭურვილლაბორატორიაში,სადაც შესაფასებელი პირი პრაქტიკულ საქმიანობას ეწევა. შემფასებელმა წინასწარ უნდა დაგეგმოს დაკვირვების პროცესი, იმის დასადგენად, თუ რამდენად სწორად იყენებს შესაფასებელი პირი ცოდნას, უნარებსა და ყველა რესურსს შედეგის მისაღწევად.შეფასება დადებითია იმ შემთხვევაში თუ სტუდენტი ყველა კითხვას სწორად გასცემს პასუხს.

*პრაქტიკული სამუშაო*

- მონიტორინგის და ინციდენტების სერვისების კონფიგურერება

| სწავლის შედეგი | დასახელება | შეფასება | |
|---|---|---|---|
| | | კი | არა |
| SNMP პროტოკოლის კონფიგურირება. | სწორად დაკონფიგურირა SNMP-პროტოკოლის სხვადასხვა ვერსია. | | |
| | სწორად გამოიყენა SNMP პროტოკოლის მონიტორინგის პროგრამა | | |
| | დროულად დაადგენა და სწორად შაფასებს არსებულ პრობლემები. | | |
| Syslog ,NTP, Netflow პროტოკოლის კონფიგურირება. | სწორად დაკონფიგურირა Syslog ,NTP, Netflow. | | |
| | სწორად გარჩია Syslog პროტოკოლის მიერ მიღებულ ლოგები | | |
| | სწორად გამოიყენა Syslog, Netflow პროტოკოლის მონიტორინგის პროგრამები | | |
| | დროულად დაადგინა და სწრაფი რეაგირება მოახდინა კონფიგურაციაში არსებულ პრობლემებზე. | | |
| DHCP,DNS პროტოკოლების კონფიგურირება. | სწორად დაკონფიგურირა DHCP სერვერი და მისი პარამეტრები. | | |
| | სწორად დაკონგიგურერა DNS სერვერი. | | |
| | სწორად გამოიყენა TFTP პროტოკოლი. | | |

# 6. მესამე დონის მარშრუტიზაციის პროტოკოლების საფუძვლები (Static, RIP, EIGRP, OSPF)

## 6.1. მარშრუტიზაცის ტიპების გარჩევა და მათი გამოყენების მიზნები.

მარშრუტიზატორი იღებს გადაწყვეტილებას მარშრუტიზაციაზე, მის ცხრილში არსებული ინფორმაციის მიხედვით.

მარშრუტები შესაძლებელია დაინიშნოს ადმინისტრატორის მიერ სტატიკურად ან გამოეყოს მას დინამიურად სხვა მარშუტიზატორის მეშვეობით ან მარშრუტიზაციის პროგრამული პროტოკოლით.

მარშრუტი დგინდება 4 ძირითადი კომპონენტისაგან:

- მიმღების მისამართი
- ქვექსელის ნიღაბი
- კარიბჭის (Gateway ) მისამართი ან ინტერფეისის სახელი
- მარშრუტის ”ღირებულება” ან მეტრიკა

**მარშრუტიზაციის პროტოკოლები**

საკუთარი ინტერფეისისა და სხვა მარშრუტიზატორებისაგან მომავალი ინფორმაციის დინამიური მართვისათვის გამოიყენება მარშრუტიზაციის პროტოკოლები

დინამიური მარშრუტიზაცია ცვლის მარშრუტების სტატიკური დანიშვნის შრომატევად საქმიანობას, ქსელური ადმინისტრატორის ჩარევის გარეშე

**მარშრუტიზაციის ალგორითმი**

ინფორმაციის ანალიზი, ორი ძირითადი კრიტერიუმის საფუძველზე:

- მანძილი - რამდენად დაშორებულია მანძილი მოცემული მარშრუტიზატორიდან
- ვექტორი - რა მიმართულებითაა მიზანშეწონილი მოცემულ ქსელში პაკეტების გადამისამართება?

**მანძილი მარშრუტში** წარმოჩინდება ღირებულებით ან მეტრიკით, რომელიც შეიძლება ახასიათებს ერთ-ერთს შემდეგი პარამეტრებიდან:

- გადასვლების რიცხვი

- ადმინისტრაციული ირიბი ხარჯები
- გამტარობა
- გადაცემის სიჩქარე
- შეფერხების ალბათობა
- საიმედოობა
- პირდაპირი მარშრუტი

მარშრუტიზატორის ჩართვისთანავე სამუშაო რეჟიმში ერთვება მისი გამართული ინტერფეისები და მარშრუტიზაციის ცხრილში ინახება მასთან უშუალოდ მიერთებული ლოკალური ქსელის მისამართები, პირდაპირი მარშრუტების სახით.

Cisco-ს მარშრუტიზატორებში ამგვარი მარშრუტები აღინიშნება პრეფიქსით C. ისინი ავტომატურად ახლდება ხელახალი გამართვისას ან მარშრუტის გამორთვისას

**სტატიკური მარშრუტი**

ქსელის ადმინისტრატორს შეუძლია ხელით გამართოს სტატიკური მარშრუტი კონკრეტულ ქსელში. სტატიკური მარშრუტი არ იცვლება, მანამ სანამ ადმინისტრატორი არ შეცვლის მას.

მარშრუტიზაციის ცხრილში ეს მარშრუტი აღინიშნება პრეფიქსით S

**“Default” მარშრუტი**

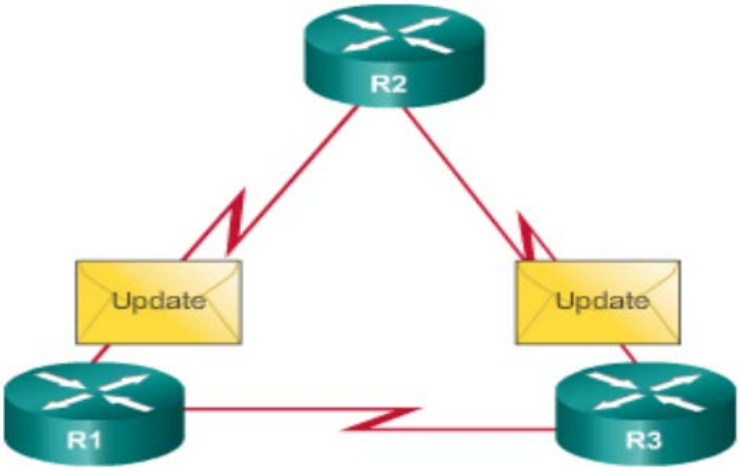
ქსელებისთვის, რომელთა გზაც არ არის ასახული მარშრუტიზაციის ცხრილში, გამოიყენება კარიბჭის მისამართი Default მითითებული . ჩვეულებრივ ეს არის შემდეგი მარშრუტიზატორი ISP-ისკენ გზაზე, თუ ქსელში მხოლოდ ერთი მარშრუტიზატორია, ავტომატურად ხდება მისი ამორჩევა

მარშრუტიზაციის ცხრილში ეს მარშრუტი აღინიშნება პრეფიქსით S*

**დინამიური (დინამიურად განახლებადი) მარშრუტები**

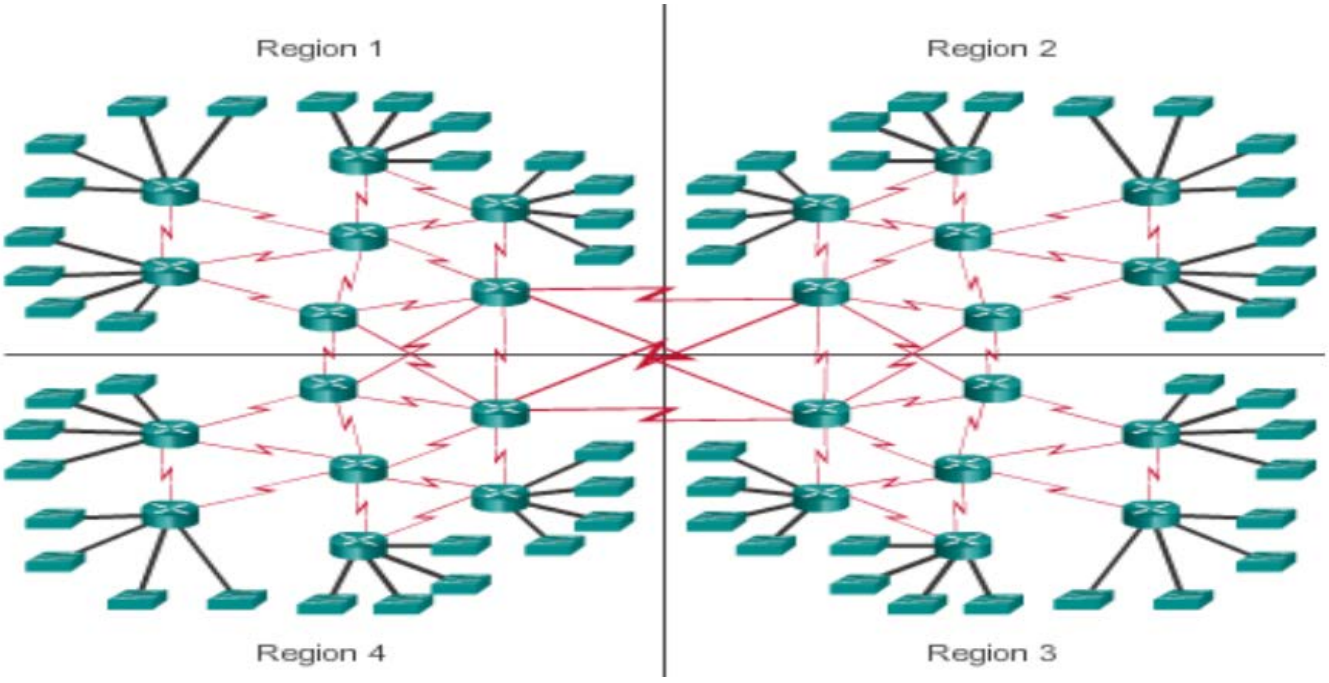
დინამიური მარშრუტები ავტომატურად იქმნება და ახლდება მარშრუტიზაციის პროტოკოლების მიერ. ეს მარშრუტები მარშრუტიზაციის ცხრილში გამოისახება წინსართით რომელიც ასახავს მარშრუტის შემქმნელ პროტოკოლის ტიპს. მაგ.: R აღნიშნავს მარშრუტის ინფორმაციის RIP პროტოკოლს.

მარშრუტიზაციის პროტოკოლების მეშვეობით მარშრუტიზატორები უზიარებენ ერთმანეთს განახლებებს და დინამიურად ქმნიან და ანახლებენ მარშრუტიზაციის ცხრილებს



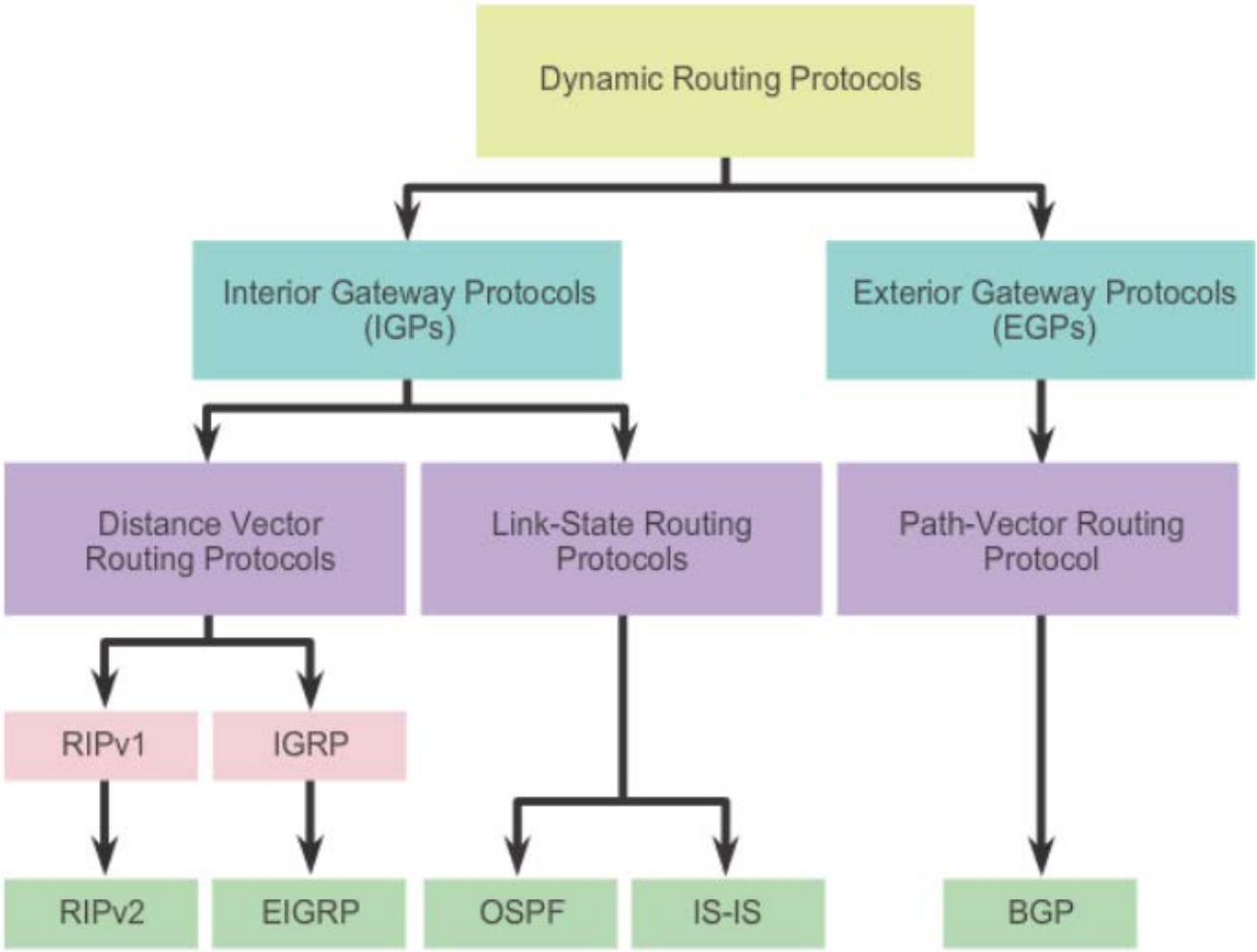

*სურ.6.1. 1*

დინამიური მარშრუტიზაცია საუკეთესო(შესაბამისობითი) არჩევანია დიდი მასშტაბის ქსელების შემთხვევაში




*სურ.6.1. 2*

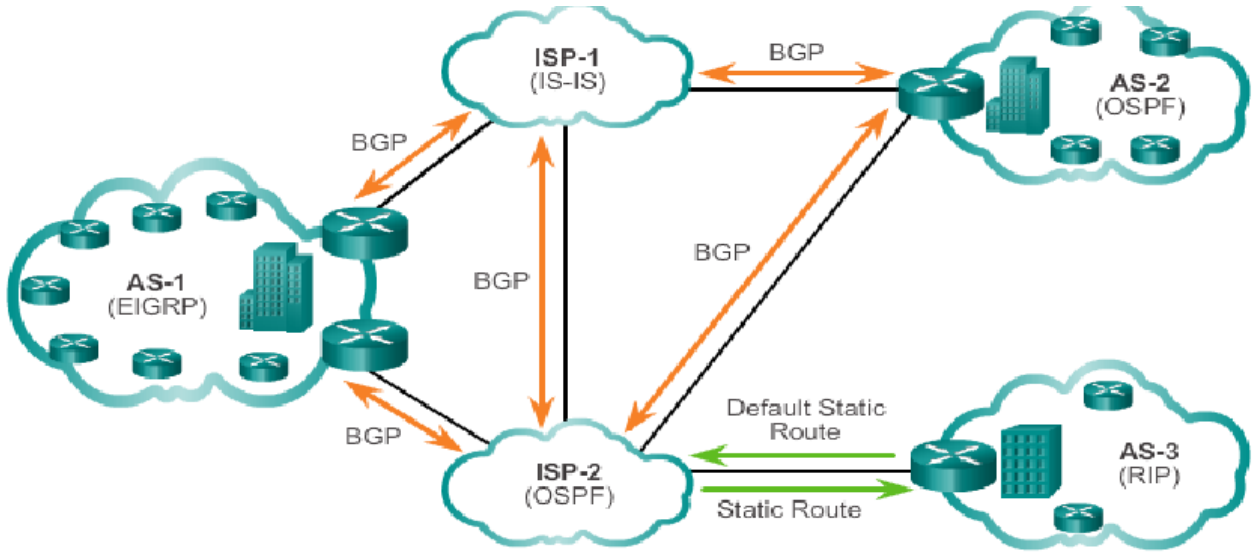
მარშრუტიზაციის პროტოკოლების ტიპები და კლასიფიკაცია




*სურ.6.1. 3*

**AS(ავტონომიური სისტემა)** – ერთი ადმინისტრირების ქვეშ მოქცეული მარშრუტიზატორების ერთობლიობა.

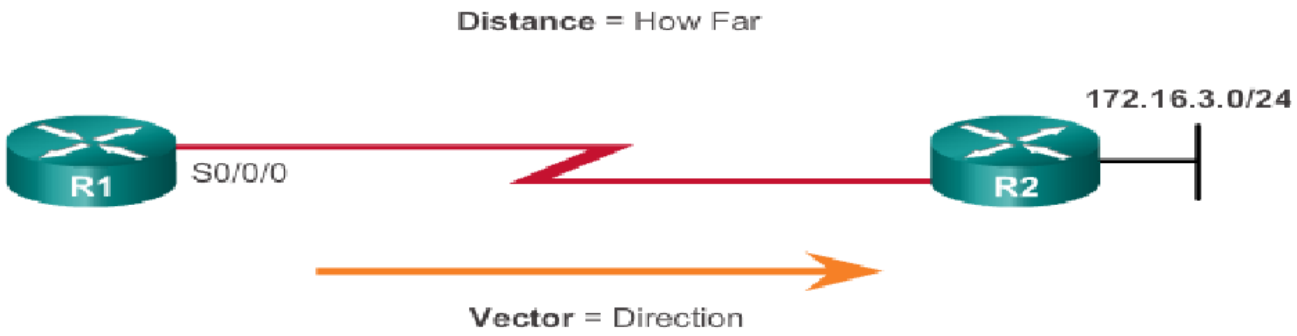
ინტერნეტი დაფუძნებულია ავტონომიური სისტემების AS კონცეპციაზე, ამიტომაც რეალიზებულია პროტოკოლების 2 ჯგუფი: **Interior Gateway Protocols (IGP)** - გამოიყენება ერთი ავტონომიური სისტემის შიგნით მარშუტიზაციისათვის, **Exterior Gateway Protocols (EGP)** - გამოიყენება ავტონომიურ სისტემებს შორის მარშუტიზაციისათვის




*სურ.6.1. 4*

**Distance vector(IPv4 IGPs: RIPv1 , RIPv2, IGRP, EIGRP)** პროტოკოლები ეფუძნება შემდეგ 2 მახასიათებელს:

- **Distance** - განსაზღვრავს დანიშნულების ქსელამდე სიშორეს და დაფუძნებულია მეტრიკაზე, როგორიცაა: გადასასვლელების(Hop) რაოდენობა, ღირებულება, გამტარუნარიანობა, დაყოვნება და სხვ.
- **Vector** - განსაზღვრავს შემდეგი გადასასვლელი მარშრუტიზატორის ან გასასვლელის ინტერფეისს

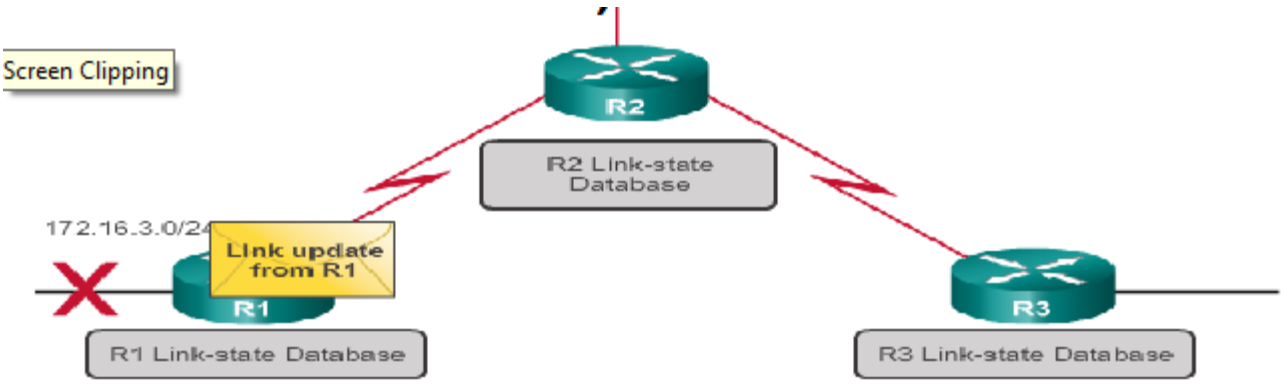



*სურ.6.1. 5*

**Link-State IPv4 IGPs OSPF; IS-IS** - მოცემული პროტოკოლებით კონფიგურირებულ მარშრუტიზატორებს შეუძლიათ შეიქმნან სრული წარმოდგენა ქსელის ტოპოლოგიაზე, ყველა სხვა მარშრუტიზატორიდან ინფორმაციის მიღების გზით.

მოცემული პროტოკოლები განსაკუთრებით ეფექტურია დიდი ზომის იერარქიულ ქსელებში, მაშინ როდესაც ქსელების სწრაფ კონვერგენციას აქვს გადამწყვეტი მნიშვნელობა

- RIP პროტოკოლებით კონფიგურირებული მარშრუტიზატორები პერიოდულად აგზავნიან განახლებებს მეზობელ როუტერებზე, მაშინ როდესაც Link-State განახლებები იგზავნება ქსელის ტოპოლოგიაში ცვლილების მოხდენის შემდეგ.




*სურ.6.1. 6*

მარშრუტიზაციის პროტოკოლები შეგვიძლია შევადაროთ შემდეგი მახასიათებლების მიხედვით:

**Speed of Convergence** - კონვერგენციის სიჩქარე განსაზღვრავს თუ რამდენად სწრაფად გაუზიარებენ მარშრუტიზატორები ქსელში მომხდარ ცვლილებებს

**Scalability(მასშტაბურობა)** - რამდენად მასშტაბური შეიძლება იყოს ქსელი

**Classful or Classless (Use of VLSM)** - პროტოკოლები მარშრუტიზაციისას ქვექსელის ნიღაბის მხარდაჭერით (Classful) და მის გარეშე (Classless)

**Resource Usage** - რესურსების გამოყენება ეხმიანება მარშრუტიზაციის პროტოკოლების მოთხოვნებს, მათ შორის - მეხსიერების(RAM) ზომა, პროცესორის სისწრაფე, გამტარუნარიანობა...

**Implementation and Maintenance** - რეალიზაცია და მომსახურება აღწერს ქსელის ადმინისტრატორის ცოდნის დონეს , რათა მან შეძლოს მარშრუტიზაციის მოცემული პროტოკოლების საფუძველზე, ქსელის სათანადო მართვა

### მარშრუტიზაციის პროტოკოლების შედარება

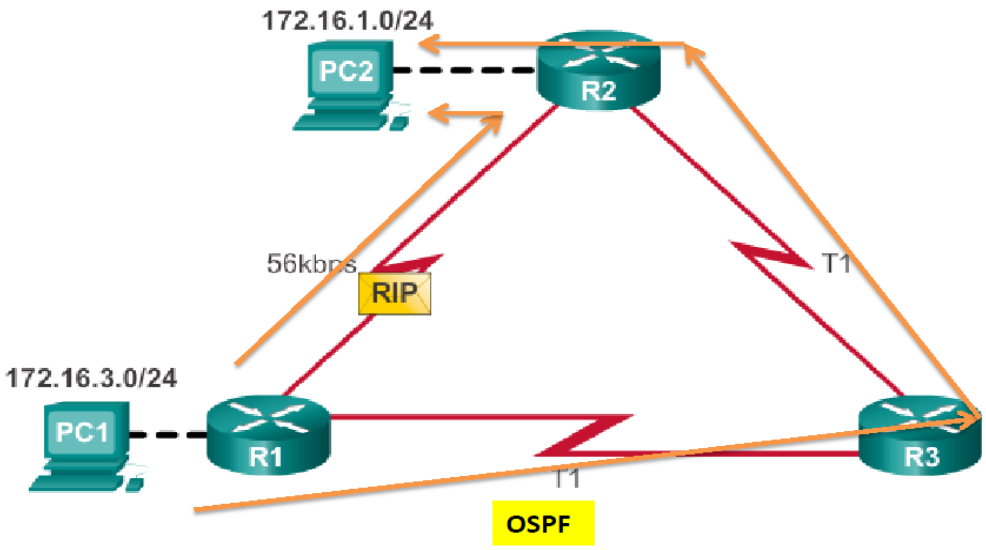
| | Distance Vector | | | | Link State | |
|---|---|---|---|---|---|---|
| | RIPv1 | RIPv2 | IGRP | EIGRP | OSPF | IS-IS |
| Speed Convergence | Slow | Slow | Slow | Fast | Fast | Fast |
| Scalability - Size of Network | Small | Small | Small | Large | Large | Large |
| Use of VLSM | No | Yes | No | Yes | Yes | Yes |
| Resource Usage | Low | Low | Low | Medium | High | High |
| Implementation and Maintenance | Simple | Simple | Simple | Complex | Complex | Complex |

*სურ.6.1. 7*

### მარშრუტიზაციის პროტოკოლების მეტრიკა

სხვადასხვა პროტოკოლები იყენებენ განსხვავებულ მეტრიკას დანიშნულების მისამართამდე მარშრუტის განსაზღვრისას

- მაგ.: RIP პროტოკოლი განსაზღვრავს მარშრუტს გადასასვლელების(Hop) რაოდენობის მიხედვით, მაშინ როდესაც OSPF პროტოკოლი ირჩევს მარშრუტს გამტარუნარიანობაზე (bandwidth) დაყრდნობით




*სურ.6.1. 8*

## 6.2. სტატიკური მარშრუტიზაციის საბაზისო კონფიგურაცია

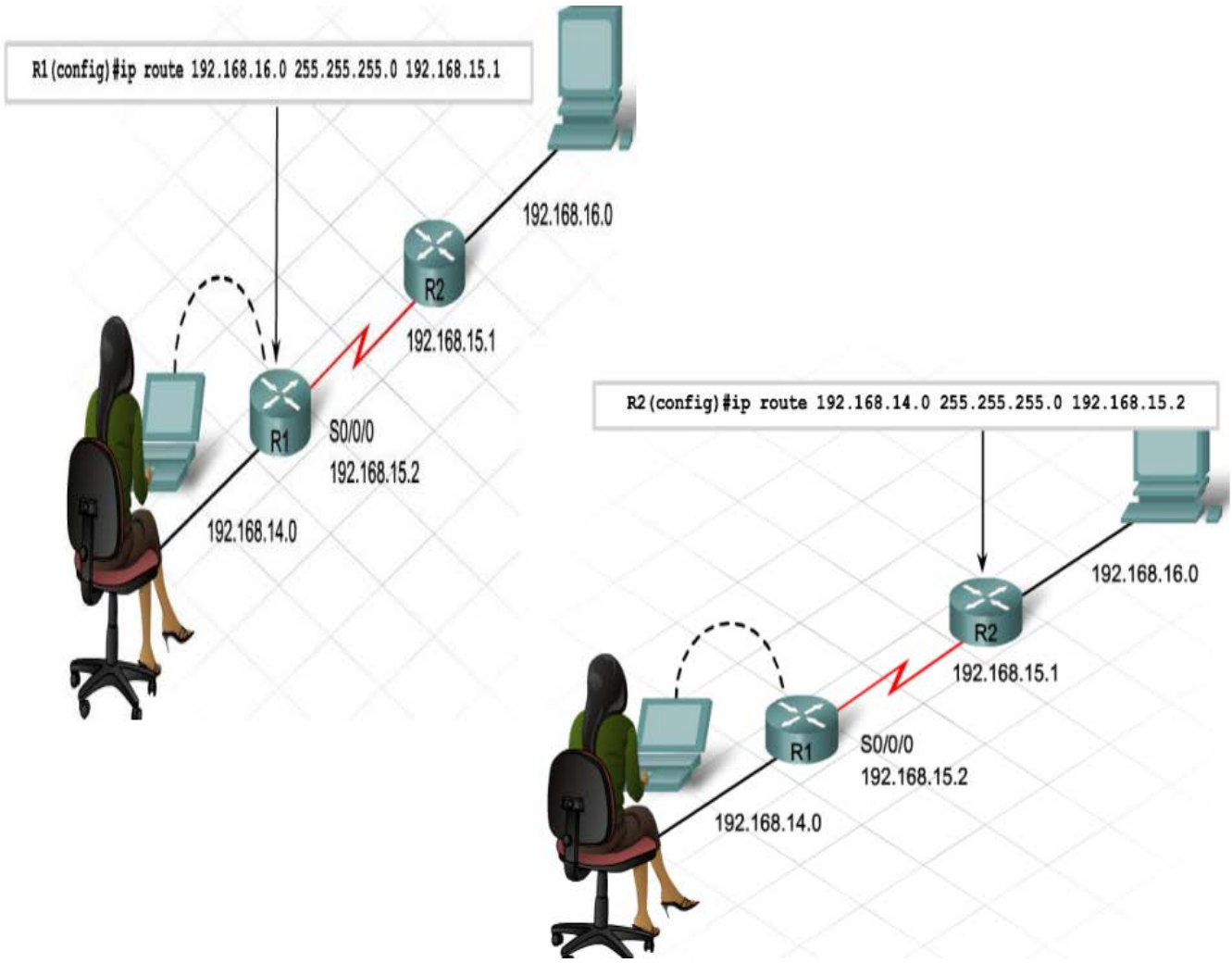
*პრაქტიკული სავარჯიშო*

დაინიშნოს 192.168.16.0/24 ქსელის კვანძებისთვის 192.168.15.1 მისამართის მქონე ინტერფეისი მარშრუტიზატორი, როგორც გამავალი ინტერფეისი

```
Router1>enable
Router1#config terminal
Router1(config)#ip route 192.168.16.0 255.255.255.0
192.168.15.1
```

ან

```
Router1(config)#ip route 192.168.16.0 255.255.255.0
S0/0/0
```




*სურ.6.2. 1*

*პრაქტიკული სამუშაო*

ქვემოთ მოცემული მარშრუტიზაციის ცხრილების მიხედვით შექმენით ქსელის ფიზიკური და ლოგიკური მოდელი

**Router 1**

Router#show ip route

Gateway of last resort is 192.168.0.2 to network 0.0.0.0

10.0.0.0/25 is subnetted, 2 subnets

C 10.0.0.0 is directly connected, FastEthernet0/0

C 10.0.0.128 is directly connected, FastEthernet0/1

192.168.0.0/30 is subnetted, 1 subnets

C 192.168.0.0 is directly connected, Serial0/0/0

S* 0.0.0.0/0 [1/0] via 192.168.0.2

**Router 2**

Router#show ip route

Gateway of last resort is not set

10.0.0.0/25 is subnetted, 2 subnets

S 10.0.0.0 [1/0] via 192.168.0.1

S 10.0.0.128 [1/0] via 192.168.0.1

172.16.0.0/25 is subnetted, 2 subnets

S 172.16.0.0 [1/0] via 192.168.0.6

S 172.16.0.128 [1/0] via 192.168.0.6

192.168.0.0/30 is subnetted, 2 subnets

C 192.168.0.0 is directly connected, Serial0/0/0

C 192.168.0.4 is directly connected, Serial0/0/1

**Router 3**

Router#SHOW IP ROute

Gateway of last resort is 192.168.0.5 to network 0.0.0.0

172.16.0.0/25 is subnetted, 2 subnets

C 172.16.0.0 is directly connected, FastEthernet0/0

C 172.16.0.128 is directly connected, FastEthernet0/1

192.168.0.0/30 is subnetted, 1 subnets

C 192.168.0.4 is directly connected, Serial0/0/1

S* 0.0.0.0/0 [1/0] via 192.168.0.5

## 6.3. მარშრუტიზაციის პროტოკოლი RIP -ის საბაზისო კონფიგურაცია

RIP ( Routing Information Protocol) პროტოკოლი

**RIP1**

- განახლებები იგზავნება 255.255.255.255 მისამართზე ყოველ 30 წამში
- მარშრუტი განისაზღვრება გადასასვლელების(Hop) რაოდენობით
- მაქსიმალური გადასასვლელების რაოდენობაა 15

**RIP2**

- განახლებები იგზავნება 224.0.0.9 მისამართზე
- აქვს უკლასო მარშრუტიზაციის VLSM და CIDR მხარდაჭერა
- აქვს ჯამური მარშუტიზაციის მხარდაჭერა
- უსაფრთხოების თვალსაზრისით აქვს აუტენტიფიკაციის მექანიზმის მხარდაჭერა

| Characteristics and Features | RIPv1 | RIPv2 |
|---|---|---|
| Metric | Both use hop count as a simple metric. The maximum number of hops is 15. | |
| Updates Forwarded to Address | 255.255.255.255 | 224.0.0.9 |
| Supports VLSM | ✗ | ✓ |
| Supports CIDR | ✗ | ✓ |
| Supports Summarization | ✗ | ✓ |
| Supports Authentication | ✗ | ✓ |

*სურ.6.3. 1*

IPv6 მხარდაჭერის პროტოკოლია RIPNG, მაქსიმალური 15 გადასასვლელითა და administrative distance 120-ით

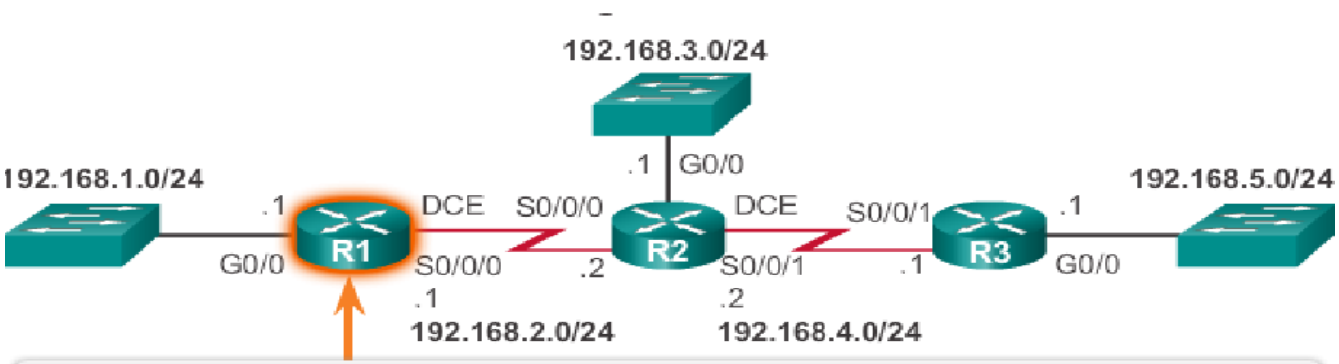
### *6.3.1. RIP კონფიგურირება*

ძირითადი მახასიათებლები:

- RIP იყენებს გადასასვლელების რიცხვს, როგორც მეტრიკას მარშრუტის არჩევისას
- 15-ზე მეტი გადასასვლელის შემთხვევაში განსაზღვრავს როგორც მიუღწეველ მარშრუტს
- აგზავნის მარშრუტიზაციის ცხრილის მონაცემებს ყოველ 30 წამში ;

**პროტოკოლის გამართვა**

- Router(config)#router rip
- Router (config-router)#version 2
- Router(config-router)#network [ქსელის მისამართი]




```
R1(config)# router rip
R1(config-router)# network 192.168.1.0
R1(config-router)# network 192.168.2.0
R1(config-router)#
```

*სურ.6.3.1. 1*

### პარამეტრების შემოწმება

```
R1# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 16 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip

  Default version control: send version 1, receive any version
    Interface              Send  Recv  Triggered RIP  Key-chain
    GigabitEthernet0/0     1     1 2
    Serial0/0/0            1     1 2

  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    192.168.1.0
    192.168.2.0

  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.2.2           120     00:00:15
  Distance: (default is 120)
```

1 2 3 4 5 6

*სურ.6.3.1. 2*

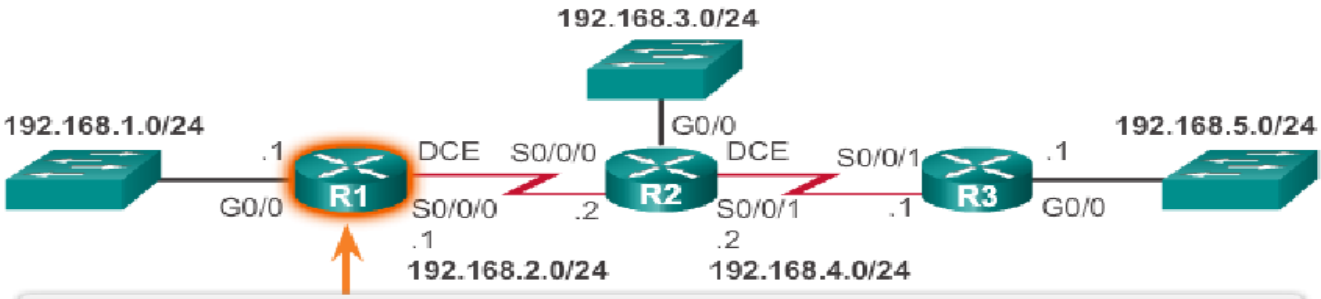
მარშრუტიზაციის ცხრილში RIP პროტოკოლით მიღებული ჩანაწერი R სიმბოლოთი აღინიშნება

```
R1# show ip route | begin Gateway
Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/0
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/24 is directly connected, Serial0/0/0
L        192.168.2.1/32 is directly connected, Serial0/0/0
R     192.168.3.0/24 [120/1] via 192.168.2.2, 00:00:24, Serial0/0/0
R     192.168.4.0/24 [120/1] via 192.168.2.2, 00:00:24, Serial0/0/0
R     192.168.5.0/24 [120/2] via 192.168.2.2, 00:00:24, Serial0/0/0
R1#
```

*სურ.6.3.1. 3*

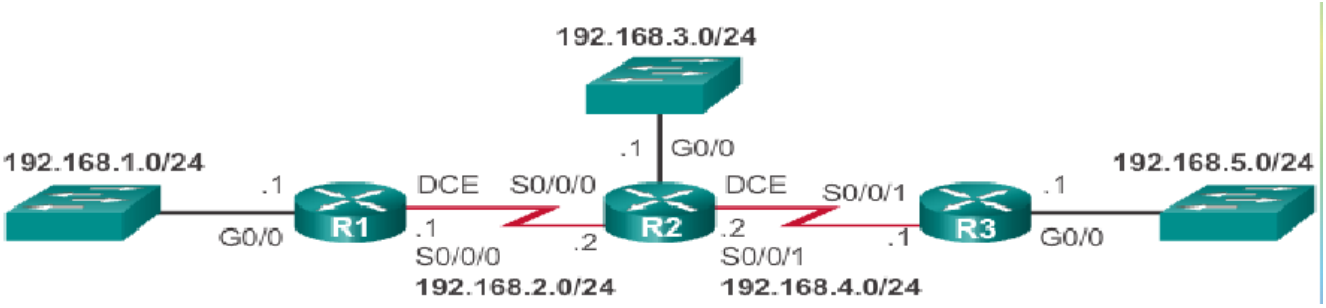
RIPv2-ის გააქტიურება




```
R1(config)# router rip
R1(config-router)# version 2
R1(config-router)# ^Z
R1#
R1# show ip protocols | section Default
  Default version control: send version 2, receive version 2
    Interface              Send  Recv  Triggered RIP  Key-chain
    GigabitEthernet0/0     2     2
    Serial0/0/0            2     2
R1#
```

*სურ.6.3.1. 4*

Passive ინტერფეისის დანიშვნა



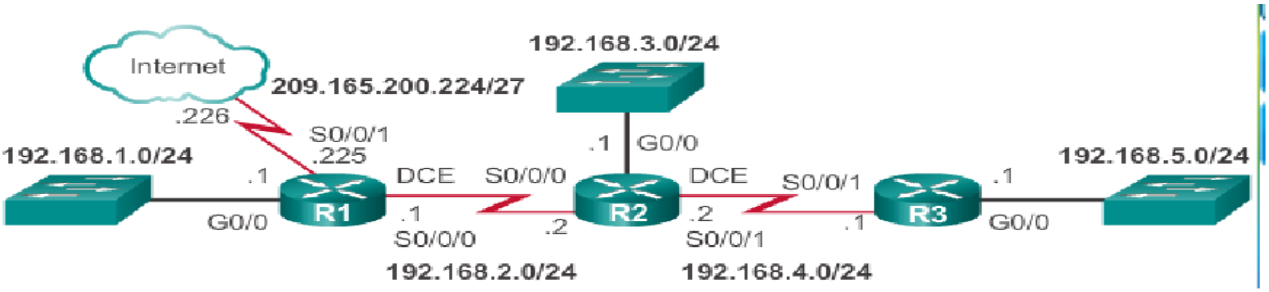

*სურ.6.3.1. 5*

მოცემულ სურათზე R1 მარშრუტიზატორის G0/0 ინტერფეისზე არ არის პროტოკოლებით ფორმირებული განახლებების დაგზავნის აუცილებლობა, რამეთუ მოცემულ ინტერფეისზე მიერთებულია LAN ქსელი, ამიტომაც მას უნდა მიენიჭოს Passive სტატუსი

```
R1(config)# router rip
R1(config-router)# passive-interface g0/0
R1(config-router)# end
```

*სურ.6.3.1. 6*

**Default მარშრუტის გავრცელება(შეხამება) RIP პროტოკოლთან**

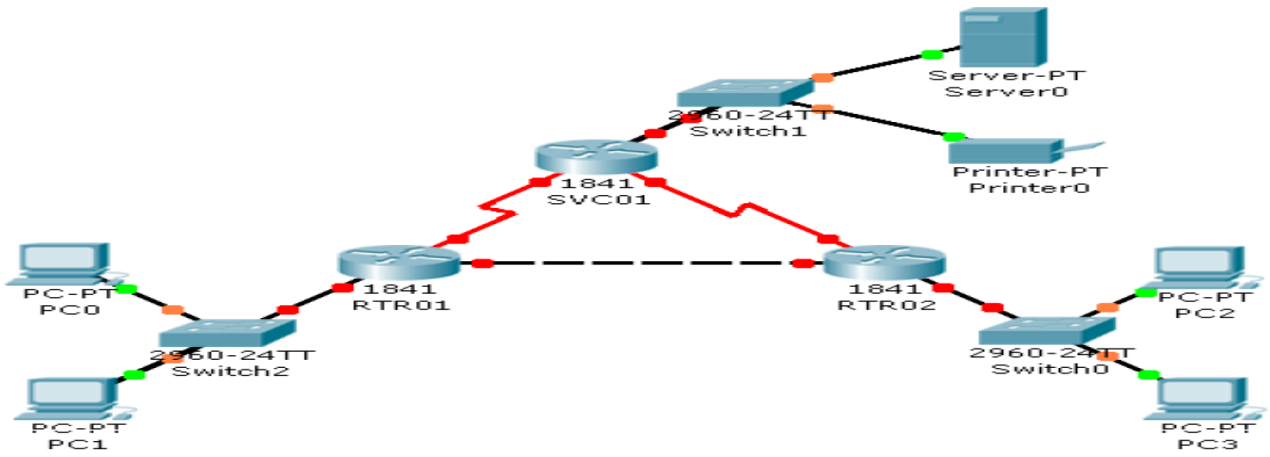



```
R1(config)# ip route 0.0.0.0 0.0.0.0 S0/0/1 209.165.200.226
R1(config)# router rip
R1(config-router)# default-information originate
R1(config-router)# ^Z
R1#
*Mar 10 23:33:51.801: %SYS-5-CONFIG_I: Configured from console by
console
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.200.226 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 209.165.200.226, Serial0/0/1
```

*სურ.6.3.1. 7*

*პრაქტიკული სამუშაო*




1. შექმენით სურათზე მოცემულის შესაბამისი ქსელის ფიზიკური მოდელი
2. ლოგიკური მისამართები შეარჩიეთ თქვენი სურვილისამებრ
3. მარშრუტიზატორებში გააქტიურეთ RIPv2 პროტოკოლი
4. შეამოწმეთ კავშირი სხვადასხვა ქსელის კვანძებს შორის

*პრაქტიკული სამუშაო*

**RIPv2 პროტოკოლის კონფიგურირება (სამუშაო სრულდება Packet Tracer-ში)**

სამუშაოს შესაბამისი ფაილის გადმოსაწერად გააქტიურეთ შემდეგი ბმული:

http://1drv.ms/1mUCEKm

*პრაქტიკული სამუშაო*

ქვემოთ მოცემული მარშრუტიზაციის ცხრილების მიხედვით შექმენით ქსელის ფიზიკური და ლოგიკური მოდელი

**თქვენი გვარი1#show ip route**

10.0.0.0/24 is subnetted, 1 subnets

C 10.0.0.0 is directly connected, Loopback0

172.16.0.0/28 is subnetted, 2 subnets

C 172.16.0.0 is directly connected, FastEthernet0/1

C 172.16.0.16 is directly connected, FastEthernet0/0

R 192.168.0.0/24 [120/1] via 172.16.0.2, 00:00:03, FastEthernet0/1

R 192.168.10.0/24 [120/1] via 172.16.0.18, 00:00:20, FastEthernet0/0

**თქვენი გვარი2#show ip route**

R 10.0.0.0/8 [120/1] via 172.16.0.17, 00:00:09, FastEthernet0/0

172.16.0.0/28 is subnetted, 2 subnets

R 172.16.0.0 [120/1] via 172.16.0.17, 00:00:09, FastEthernet0/0

C 172.16.0.16 is directly connected, FastEthernet0/0

R 192.168.0.0/24 [120/2] via 172.16.0.17, 00:00:09, FastEthernet0/0

C 192.168.10.0/24 is directly connected, Loopback0

თქვენი გვარი3#show ip route

R 10.0.0.0/8 [120/1] via 172.16.0.1, 00:00:26, FastEthernet0/0

172.16.0.0/28 is subnetted, 2 subnets

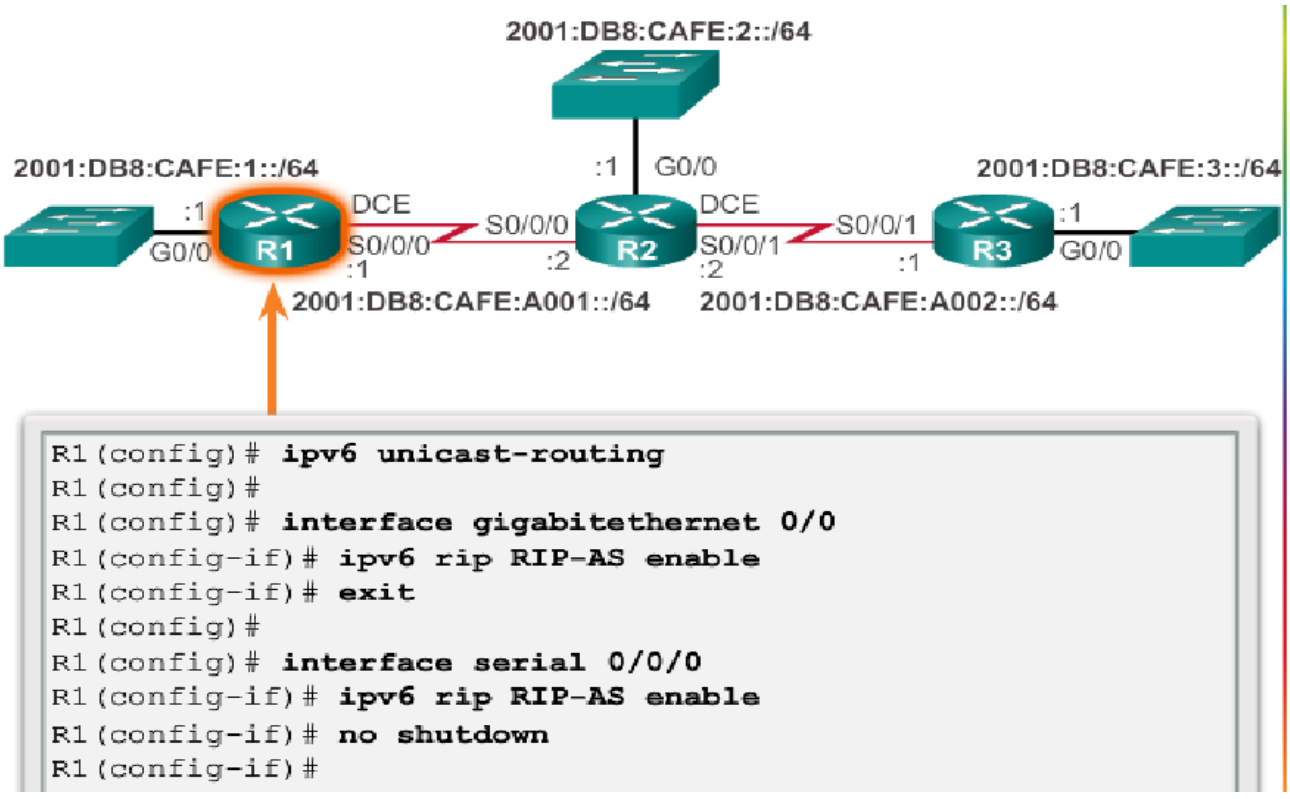
C 172.16.0.0 is directly connected, FastEthernet0/0

R 172.16.0.16 [120/1] via 172.16.0.1, 00:00:26, FastEthernet0/0

C 192.168.0.0/24 is directly connected, Loopback0

R 192.168.10.0/24 [120/2] via 172.16.0.1, 00:00:26, FastEthernet0/0

## *RIPng პროტოკოლის კონფიგურირება*




*სურ.6.3.1. 8*

RIPng პროტოკოლის შემოწმება

```
R1# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "rip RIP-AS"
  Interfaces:
    Serial0/0/0
    GigabitEthernet0/0
```

```
R1# show ipv6 route
C    2001:DB8:CAFE:1::/64 [0/0]
     via GigabitEthernet0/0, directly connected
L    2001:DB8:CAFE:1::1/128 [0/0]
     via GigabitEthernet0/0, receive
R    2001:DB8:CAFE:2::/64 [120/2]
     via FE80::FE99:47FF:FE71:78A0, Serial0/0/0
R    2001:DB8:CAFE:3::/64 [120/3]
     via FE80::FE99:47FF:FE71:78A0, Serial0/0/0
C    2001:DB8:CAFE:A001::/64 [0/0]
     via Serial0/0/0, directly connected
L    2001:DB8:CAFE:A001::1/128 [0/0]
     via Serial0/0/0, receive
R    2001:DB8:CAFE:A002::/64 [120/2]
     via FE80::FE99:47FF:FE71:78A0, Serial0/0/0
L    FF00::/8 [0/0]
     via Null0, receive
```

```
R1# show ipv6 route rip
R    2001:DB8:CAFE:2::/64 [120/2]
     via FE80::FE99:47FF:FE71:78A0, Serial0/0/0
R    2001:DB8:CAFE:3::/64 [120/3]
     via FE80::FE99:47FF:FE71:78A0, Serial0/0/0
R    2001:DB8:CAFE:A002::/64 [120/2]
     via FE80::FE99:47FF:FE71:78A0, Serial0/0/0
```

*სურ.6.3.1. 9*

*პრაქტიკული სამუშაო*

**RIPng პროტოკოლის კონფიგურირება (სამუშაო სრულდება Packet Tracer-ში)**

სამუშაოს შესაბამისი ფაილების გადმოსაწერად გააქტიურეთ შემდეგი ბმული:

- http://1drv.ms/1mUEtXD
- http://1drv.ms/1mXzgKy

## 6.4. მარშრუტიზაციის პროტოკოლი EIGRP -ის საბაზისო კონფიგურაცია

ძირითადი მახასიათებლები:

- მარშრუტის ღირებულების განსაზღვრა სხვადასხვა მეტრიკის საფუძველზე
- მაქსიმალური გადასვლების რიცხვი - 224
- RIP-გან განსხვავებით EIGRP პროტოკოლი არ შემოიფარგლება მხოლოდ საკუთარი მარშრუტიზაციის ცხრილით. ამ პროტოკოლისათვის იქმნება 2 ძირითადი მონაცემთა ბაზის ცხრილი: მეზობელი მარშრუტიზატორების ცხრილები და ტოპოლოგიის ცხრილი

| Characteristics and Features | IGRP | EIGRP |
|---|---|---|
| Metric | Both use a composite metric consisting of bandwidth and delay. Reliability and load can also be included in the metric calculation. | |
| Updates Forwarded to Address | 255.255.255.255 | 224.0.0.10 |
| Supports VLSM | ✗ | ✓ |
| Supports CIDR | ✗ | ✓ |
| Supports Summarization | ✗ | ✓ |
| Supports Authentication | ✗ | ✓ |

*სურ.6.4. 1*

*პრაქტიკული სამუშაო*

ქვემოთ მოცემული მარშრუტიზაციის ცხრილების მიხედვით შექმენით ქსელის ფიზიკური და ლოგიკური მოდელი

მარშრუტიზატორი 1

EQE1#sh ip route

172.16.0.0/24 is subnetted, 6 subnets

D 172.16.252.0 [90/2681856] via 172.16.250.2, 00:18:54, Ethernet0/0

C 172.16.250.0 is directly connected, Ethernet0/0

C 172.16.251.0 is directly connected, Ethernet0/1

D 172.16.50.0 [90/2195456] via 172.16.250.2, 00:18:54, Ethernet0/0

C 172.16.1.0 is directly connected, Loopback0

D 172.16.100.0 [90/2707456] via 172.16.250.2, 00:18:54, Ethernet0/0

C 192.168.1.0/24 is directly connected, Loopback1

მარშრუტიზატორი 2

EQE2#sh ip route

172.16.0.0/24 is subnetted, 6 subnets

C 172.16.252.0 is directly connected, Serial0

D 172.16.250.0 [90/2681856] via 172.16.252.1, 00:21:10, Serial0

C 172.16.251.0 is directly connected, Serial1

D 172.16.50.0 [90/2195456] via 172.16.252.1, 00:21:10, Serial0

D 172.16.1.0 [90/2707456] via 172.16.252.1, 00:15:36, Serial0

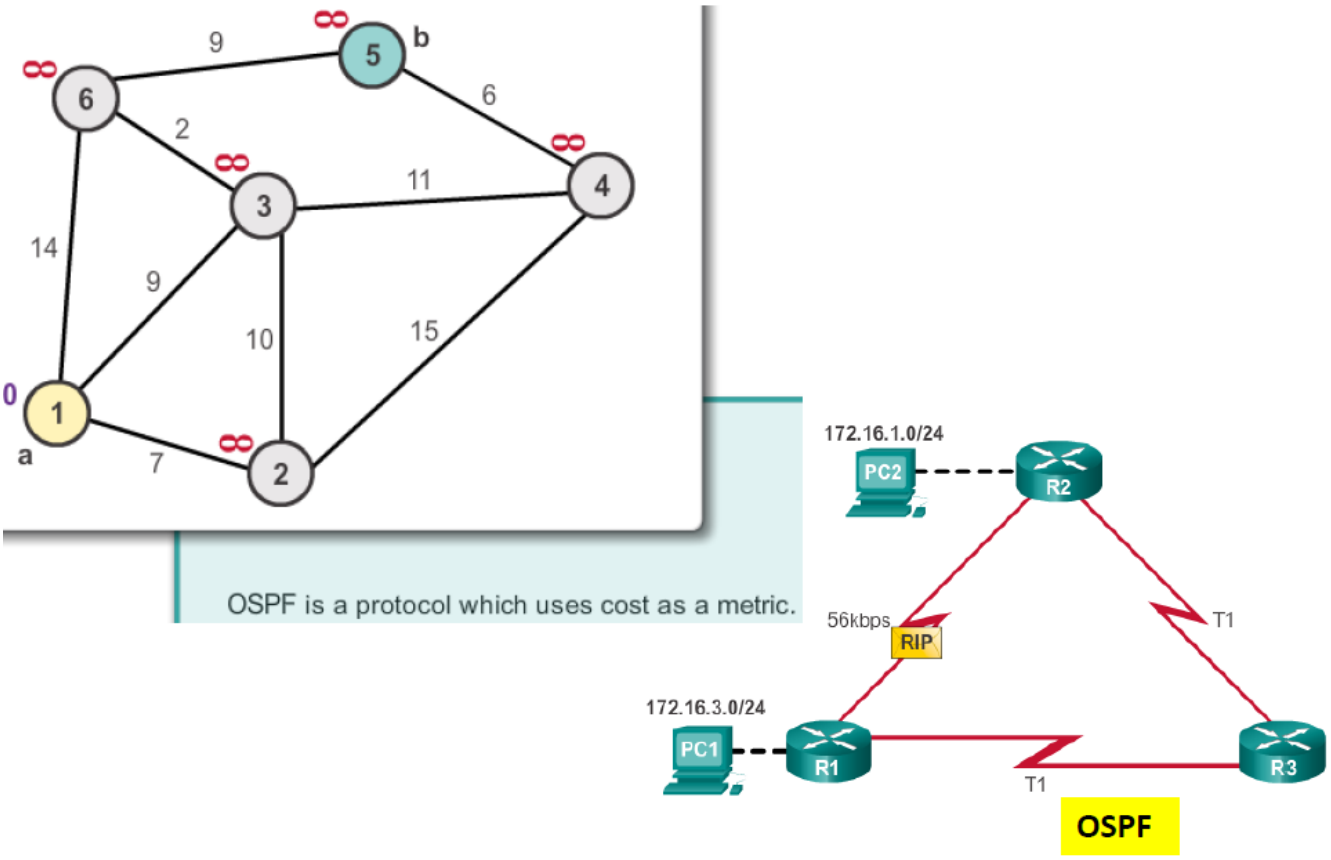
C 172.16.100.0 is directly connected, Ethernet0

## 6.5. მარშრუტიზაციის პროტოკოლი OSPF -ის საბაზისო კონფიგურაცია

სხვადასხვა პროტოკოლები იყენებენ განსხვავებულ მეტრიკას დანიშნულების მისამართამდე მარშრუტის განსაზღვრისას

**OSPF(Open Shortest Path First)** პროტოკოლი ირჩევს მარშრუტს გამტარუნარიანობაზე (bandwidth) დაყრდნობით

მარშრუტიზაციის პროტოკოლების მეტრიკა




*სურ.6.5. 1*

### *6.5.1. OSPF კონფიგურირების ძირითადი ბრძანებები*

*უშუალოდ მიერთებული ქსელის მითითება*

- ✓ **network „ქსელის მისამართი“ wildcard-mask area „არეალის ნომერი“**

  მაგ.: network 192.168.16.0 0.0.0.255 area 0

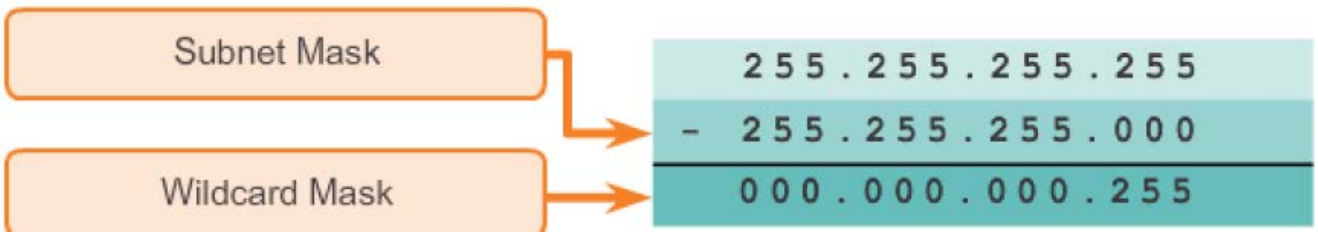
- 192.168.16.0 - უშუალოდ მიერთებული ქსელია
- 0.0.0.255 - შებრუნებული ქვექსელის ნიღაბი
- 0 – OSPF area

```
R1(config)# router ospf 10
R1(config-router)# network 172.16.1.0 0.0.0.255 area 0
R1(config-router)# network 172.16.3.0 0.0.0.3 area 0
R1(config-router)# network 192.168.10.4 0.0.0.3 area 0
R1(config-router)#
```

*სურ.6.5.1. 1*

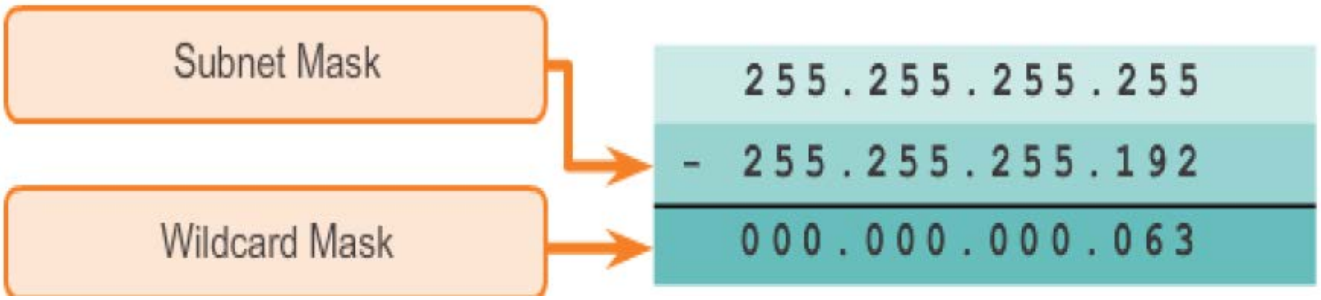
**შებრუნებული ქვექსელის ნიღაბის (Wildcard mask)-ის გამოთვლა**

მაგ.: გამოვითვალოთ 255.255.255.0 ანუ /24 – ის Wildcard mask




*სურ.6.5.1. 2*

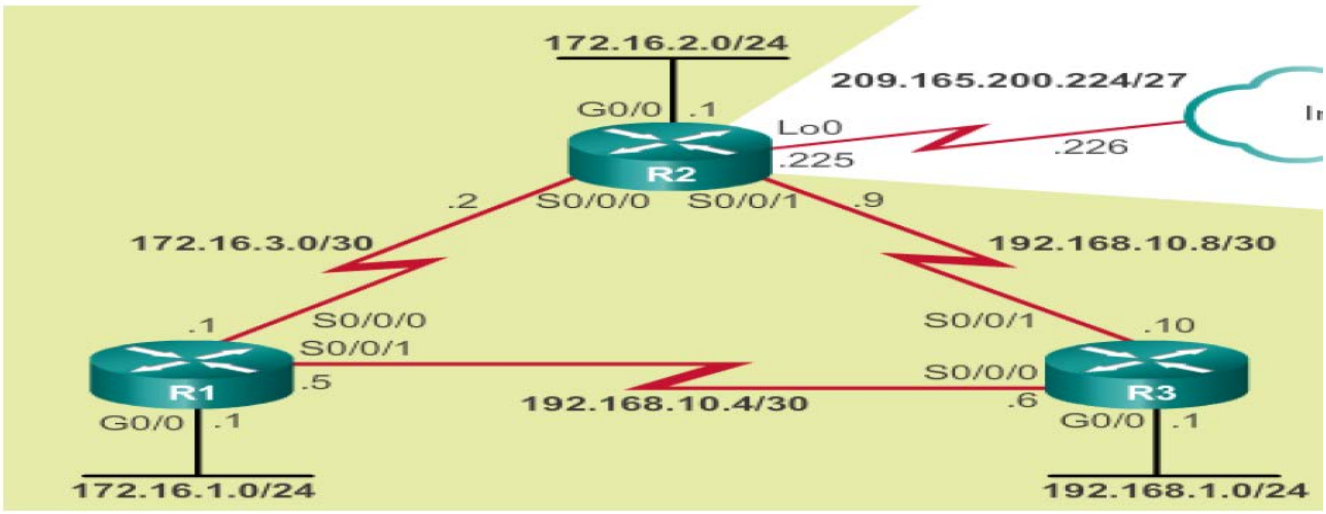
მაგ.: გამოვითვალოთ 255.255.255.192 ანუ /26 – ის Wildcard mask




*სურ.6.5.1. 3*

## Passive-interface- ის დანიშვნა

```
R1(config)# router ospf 10
R1(config-router)# passive-interface GigabitEthernet 0/0
R1(config-router)# end
R1#
```




*სურ.6.5.1. 4*

## OSPF Cost (ღირებულების) განსაზღვრა

Cost = reference bandwidth / interface bandwidth

Cost = 100,000,000 bps / interface bandwidth in bps

| Interface Type | Reference Bandwidth in bps | | Default Bandwidth in bps | Cost |
|---|---|---|---|---|
| 10 Gigabit Ethernet<br>10 Gbps | 100,000,000 | ÷ | 10,000,000,000 | 1 |
| Gigabit Ethernet<br>1 Gbps | 100,000,000 | ÷ | 1,000,000,000 | 1 |
| Fast Ethernet<br>100 Mbps | 100,000,000 | ÷ | 100,000,000 | 1 |
| Ethernet<br>10 Mbps | 100,000,000 | ÷ | 10,000,000 | 10 |
| Serial<br>1.544 Mbps | 100,000,000 | ÷ | 1,544,000 | 64 |
| Serial<br>128 kbps | 100,000,000 | ÷ | 128,000 | 781 |
| Serial<br>64 kbps | 100,000,000 | ÷ | 64,000 | 1562 |

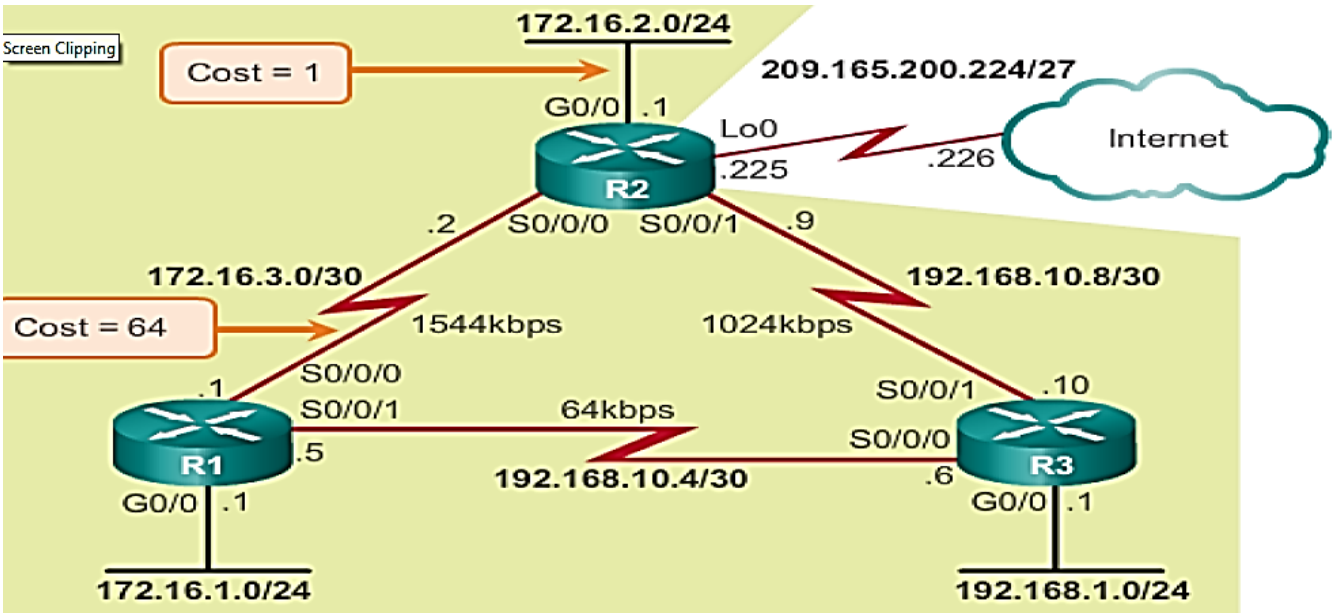
*სურ.6.5.1. 5*

მაგ.: სურათზე გამოსახულის მიხედვით განვსაზღვროთ R1-დან R2-ის 172.16.2.0/24 ქსელზე ჯამური ღირებულება

Serial link from R1 to R2 cost = 64

Gigabit Ethernet link on R2 cost = 1

Total cost to reach 172.16.2.0/24 = 65




*სურ.6.5.1. 6*

**auto-cost reference-bandwidth Mb/s**

Default პარამეტრებით მარშრუტიზატორებზე „reference bandwidth“ მაჩვენებელი 100 Mb/s-ია, ამიტომაც Cost მაჩვენებელი FastEthernet; GigabitEthernet და 10GigabitEthernet ინტერფეისებისთვის იქნება იდენტური.

| | | | | |
|---|---|---|---|---|
| **10 Gigabit Ethernet**<br>10 Gbps | 100,000,000 | ÷ | 10,000,000,000 | 1 |
| **Gigabit Ethernet**<br>1 Gbps | 100,000,000 | ÷ | 1,000,000,000 | 1 |
| **Fast Ethernet**<br>100 Mbps | 100,000,000 | ÷ | 100,000,000 | 1 |

*სურ.6.5.1. 7*

ამიტომაც უპრიანია GigabitEthernet და 10GigabitEthernet ინტერფეისების შემთხვევაში დავნიშნოთ ქვემოთ მოცემულის შესაბამისი „reference bandwidth“ მაჩვენებელი

Gigabit Ethernet - auto-cost reference-bandwidth 1000

10 Gigabit Ethernet - auto-cost reference-bandwidth 10000

**Cost მაჩვენებლის განსაზღვრის ალტერნატილი გზები :**

ინტერფეისზე შევცვალოთ Bandwidth მაჩვენებელი

```
R1(config)# int s0/0/1
R1(config-if)# bandwidth 64
R1(config-if)# end
R1#
*Mar 27 10:10:07.735: %SYS-5-CONFIG_I: Configured from console by c
R1#
R1# show interfaces serial 0/0/1 | include BW
  MTU 1500 bytes, BW 64 Kbit/sec, DLY 20000 usec,
R1#
R1# show ip ospf interface serial 0/0/1 | include Cost:
  Process ID 10, Router ID 1.1.1.1, Network Type
  POINT_TO_POINT, Cost: 15625
R1#
```

*სურ.6.5.1. 8*

ხელით შევცვალოთ(დავადგინოთ) ip ospf cost ღირებულება

```
R1(config)# int s0/0/1
R1(config-if)# no bandwidth 64
R1(config-if)# ip ospf cost 15625
R1(config-if)# end
R1#
R1# show interface serial 0/0/1 | include BW
    MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
R1#
R1# show ip ospf interface serial 0/0/1 | include Cost:
    Process ID 10, Router ID 1.1.1.1, Network Type POINT_TO_POINT,
    Cost: 15625
R1#
```

Screen Clipping

*სურ.6.5.1. 9*

**OSPF შემოწმება**

```
R1# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 10"
  Outgoing update filter list for all interfaces is not
  set
  Incoming update filter list for all interfaces is not
  set
  Router ID 1.1.1.1
  Number of areas in this router is 1. 1 normal 0 stub 0
  nssa
  Maximum path: 4
  Routing for Networks:
    172.16.1.0 0.0.0.255 area 0
    172.16.3.0 0.0.0.3 area 0
    192.168.10.4 0.0.0.3 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    2.2.2.2              110      00:17:18
    3.3.3.3              110      00:14:49
  Distance: (default is 110)
```

*სურ.6.5.1. 10*

```
Screen Clipping how ip ospf neighbor

Neighbor ID  Pri  State   Dead Time Address        Interface
3.3.3.3      0    FULL/-  00:00:37  192.168.10.6   Serial0/0/1
2.2.2.2      0    FULL/-  00:00:30  172.16.3.2     Serial0/0/0
R1#
```

```
R1# show ip ospf interface brief
Interface  PID  Area  IP Address/Mask  Cost  State  Nbrs F/C
Se0/0/1    10   0     192.168.10.5/30  15625 P2P    1/1
Se0/0/0    10   0     172.16.3.1/30    647   P2P    1/1
Gi0/0      10   0     172.16.1.1/24    1     DR     0/0
R1#
```

*სურ.6.5.1. 11*

*პრაქტიკული სამუშაო*

**OSPF პროტოკოლის კონფიგურირება (სამუშაო სრულდება Packet Tracer-ში)**

სამუშაოს შესაბამისი ფაილების გადმოსაწერად გააქტიურეთ შემდეგი ბმული:

- http://1drv.ms/1puiS5P
- http://1drv.ms/1puj1Ge
- http://1drv.ms/1jWt5HB

*6.5.2. OSPFv3 კონფიგურირება*

```
R1(config)# ipv6 unicast-routing
```

```
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ipv6 address fe80::1 link-local
R1(config-if)# exit
R1(config)# interface Serial0/0/0
R1(config-if)# ipv6 address fe80::1 link-local
R1(config-if)# exit
R1(config)# interface Serial0/0/1
R1(config-if)# ipv6 address fe80::1 link-local
R1(config-if)#
```

```
R1(config)# ipv6 router ospf 10
R1(config-rtr)#
*Mar 29 11:21:53.739: %OSPFv3-4-NORTRID: Process OSPFv3-1-
IPv6 could not pick a router-id, please configure manually
R1(config-rtr)#
R1(config-rtr)# router-id 1.1.1.1
R1(config-rtr)#
R1(config-rtr)# auto-cost reference-bandwidth 1000
% OSPFv3-1-IPv6: Reference bandwidth is changed. Please
ensure reference bandwidth is consistent across all routers
R1(config-rtr)#
R1(config-rtr)# end
R1#
R1# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "ospf 10"
  Router ID 1.1.1.1
  Number of areas: 0 normal, 0 stub, 0 nssa
  Redistribution:
    None
R1#
```

```
R1(config)# ipv6 unicast-routing
```

```
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)#
R1(config-if)# interface Serial0/0/0
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)#
R1(config-if)# interface Serial0/0/1
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)#
R1(config-if)# end
R1#
R1# show ipv6 ospf interfaces brief
Interface   PID   Area      Intf ID  Cost    State  Nbrs F/C
Se0/0/1     10      0         7      15625   P2P    0/0
Se0/0/0     10      0         6      647     P2P    0/0
Gi0/0       10      0         3       1      WAIT   0/0
R1#
```

## OSPFv3 შემოწმება

```
R1# show ipv6 ospf neighbor

OSPFv3 Router with ID (1.1.1.1) (Process ID 10)

Neighbor ID  Pri  State      Dead Time  Interface ID Interface
3.3.3.3        0  FULL/  -   00:00:39   6            Serial0/0/1
2.2.2.2        0  FULL/  -   00:00:36   6            Serial0/0/0
R1#
```

```
R1# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "ospf 10"
  Router ID 1.1.1.1
  Number of areas: 1 normal, 0 stub, 0 nssa
  Interfaces (Area 0):
    Serial0/0/1
    Serial0/0/0
    GigabitEthernet0/0
  Redistribution:
    None
R1#
```

```
R1# show ipv6 ospf interface brief
Interface    PID   Area            Intf ID    Cost  State Nbrs F/C
Se0/0/1      10    0               7          15625 P2P   1/1
Se0/0/0      10    0               6          647   P2P   1/1
Gi0/0        10    0               3          1     DR    0/0
R1#
```

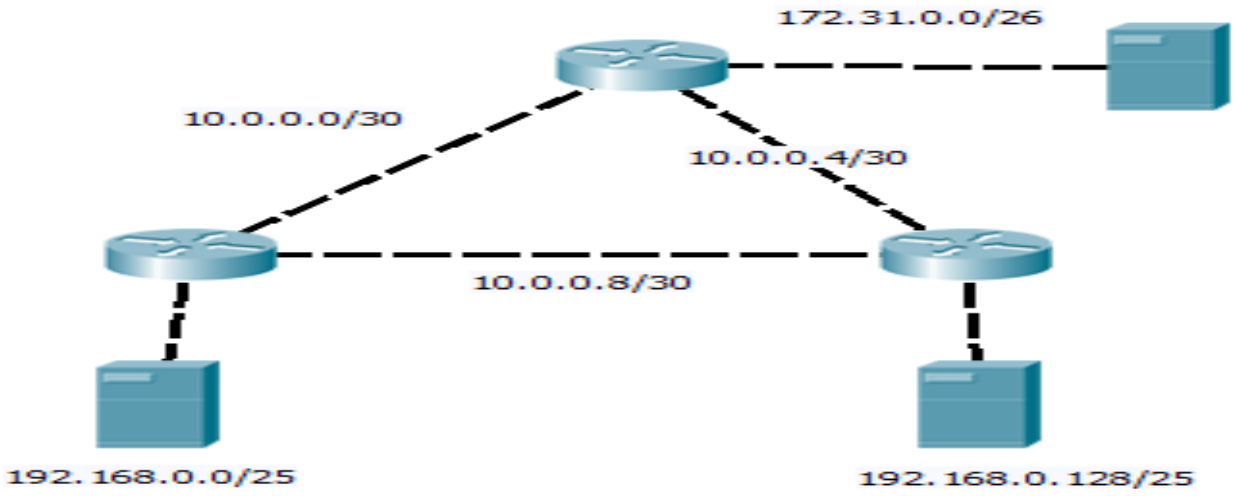
```
R1# show ipv6 route ospf
IPv6 Routing Table - default - 10 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user
Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS
summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND
Prefix, DCE - Destination
       NDr - Redirect, O - OSPF Intra, OI - OSPF Inter,
OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF
NSSA ext 2
O    2001:DB8:CAFE:2::/64 [110/657]
     via FE80::2, Serial0/0/0
O    2001:DB8:CAFE:3::/64 [110/1304]
     via FE80::2, Serial0/0/0
O    2001:DB8:CAFE:A002::/64 [110/1294]
     via FE80::2, Serial0/0/0
R1#
```

*პრაქტიკული სამუშაო*

**OSPFv3 პროტოკოლის კონფიგურირება (სამუშაო სრულდება Packet Tracer-ში)**

სამუშაოს შესაბამისი ფაილების გადმოსაწერად გააქტიურეთ შემდეგი ბმულები:

- http://1drv.ms/1nrWMUr
- http://1drv.ms/1nrWTiZ
- http://1drv.ms/1nrWZXC




1. შექმენით მოცემულის შესაბამისი ლოკალური ქსელი და ინტერფეისები დაამისამართეთ შესაბამისად
2. მარშრუტიზატორებზე გააქტიურეთ OSPF პროტოკოლი
    a. დანიშნეთ Router ID
    b. Refernce Bandwith განსაზღვრეთ 1000
    c. ლოკალური ინტერფეისებს მიანიჭეთ Passive სტატუსი

**პროცესზე დაკვირვება**

შემფასებელი აკვირდება შესაფასებელ პირის მუშაობას პროფესიული სტანდარტით(პროგრამით / მოდულით ) განსაზღვრული ამოცანების შესრულების პროცესში . დაკვირვებახორციელდება სამუშაო ადგილზე ან სამუშაო პირობებში, თუმცა დასაშვებია ჩატარდესკომპიუტერებით აღჭურვილლაბორატორიაში,სადაც შესაფასებელი პირი პრაქტიკულ საქმიანობას ეწევა. შემფასებელმა წინასწარ უნდა დაგეგმოს დაკვირვების პროცესი, იმის დასადგენად, თუ რამდენად სწორად იყენებს შესაფასებელი პირი ცოდნას, უნარებსა და ყველა რესურსს შედეგის მისაღწევად.შეფასება დადებითია იმ შემთხვევაში თუ სტუდენტი ყველა კითხვას სწორად გასცემს პასუხს.

*პრაქტიკული სამუშაო - მესამე დონის პროტოკოლების გამოყენებით საბაზისო მარშუტიზაციის განხორციელება*

| სწავლის შედეგი | დასახელება | შეფასება | |
|---|---|---|---|
| | | კი | არა |
| სტატიკური მარშუტიზაციის საბაზისო კონფიგურაცია | სწორად დააკონფიგა სტანდარტული მარშუტი(Default Route) | | |
| | დროულად დაადგინა და სწორად აღმოფხვრა სტატიკური მარშუტიზაციის პრობლემები. | | |
| მარშუტიზაციის პროტოკოლი RIP -ის საბაზისო კონფიგურაცია | სწორად მოახდინა მარშუტების განაწილება RIP მარშუტიზაციის პროტოკოლის საშუალებით | | |
| | დროულად დაადგინა და სწორად აღმოფხვრა მარშუტიზაციის პროცესში წარმოქმნილ პრობლემები | | |
| მარშუტიზაციის პროტოკოლი EIGRP -ის საბაზისო კონფიგურაცია | სწორად მოახდინა მარშუტების განაწილება EIGRP მარშუტიზაციის პროტოკოლის საშუალებით | | |
| | დროულად დაადგინა და სწორად აღმოფხვრა მარშუტიზაციის პროცესში წარმოქმნილ პრობლემები | | |
| მარშუტიზაციის პროტოკოლი OSPF -ის საბაზისო კონფიგურაცია | სწორად მოახდინა მარშუტების განაწილება OSPF მარშუტიზაციის პროტოკოლის საშუალებით | | |
| | სწორად მოახდინა მარშუტების განაწილება OSPF მარშუტიზაციის პროტოკოლის საშუალებით | | |

# დასკვნა

კომპიუტერული ქსელების მასობრიობისა და მათი მუდმივი გამოყენების ფონზე მეტადრე აქტუალურია ქსელის ადმინისტრირების საკითხები.

კომპიუტერული ქსელის ფიზიკური და ლოგიკური გამართვა კომპიუტერული ქსელის ადმინისტრატორის ძირითადი ამოცანაა.

თანამედროვე ქსელის ადმინისტრატორი კარგად უნდა ერკვეოდეს როგორც უშუალო პროფესიულ მოვალეობებში, არამედ ინფორმაციული ტექნოლოგიების სხვადასხვა მიმართულებებში, როგორიცაა კომპიუტერის არქიტექტურა, სერვერული ოპერაციული სისტემები.

ქსელის ტიპების წარმოდგენის ყველაზე თვალსაჩინო მაგალითია - ლოკალური და გლობალური ქსელები. მოცემული სახელმძღვანელო ძირითადად ეხმაურება ლოკალური ქსელების ტექნოლოგიებს, მასში განხილული საკითხები უთუოდ საინტერესო იქნება როგორც დამწყები, ასევე გარკვეული გამოცდილების მქონე ქსელის ადმინისტრირებით დაინტერესებული მკითხველისთვის.

მოცემული სახელმძღვანელო შექმნილია „კომპიუტერული ქსელის ადმინისტრირება“ პროფესიული სასწავლო პროგრამის სტუდენტებისათვის და მოიცავს I ეტაპზე სწავლებად მოდულებს.

გამოყენებული ლიტერატურა


1. CCNA R&S: Introduction to Networks (Cisco.netacad.com)
2. CCNA R&S: Routing and Switching Essentials (Cisco.netacad.com)
3. CCNA R&S: Scaling Networks (Cisco.netacad.com)
4. Cisco IT Essential 5.0 (Cisco.netacad.com)
5. Cisco CCNA Discovery (Cisco.netacad.com)
6. ვ. ადამია, ნ. არაბული, ზ. ცირამუა „კომპიუტერული ქსელები“ , საგამომცემლო სახლი „ტექნიკური უნივერსიტეტი“ ISBN 978-9941-14-646-6
7. ზ. ცირამუა, ვ. ოთხოზორია. „ინფორმაციული ტექნოლოგიების საფუძვლები - კომპიუტერული ტექნიკა და არქიტექტურა, ოპერაციული სისტემების საფუძვლები, კომპიუტერული ქსელების საფუძვლები“ თბილისი: გამომცემლობა „საუნჯე“ ; ISBN 978-9941-9260-3-7