დოქტორანტის ბიბლიოთეკა

რომან სამხარაძე, ლია გაჩეჩილაძე

კომპიუტერული ქსელების პრობლემები და მათი გადაწყვეტის მეთოდები

(სემინარის მეთოდიკური მითითებები)

„სტუ-ის IT კონსალთინგის სამეცნიერო ცენტრი“

სარჩევი

14. ციფრული ხელმოწერა. სიმეტრიული გასაღების გამოყენების მაგალითები. გარე გასაღებების გამოყენების მაგალითები. ღია გასაღებიანი სისტემების ინფრასტრუქტურის რეალიზების მაგალითების განხილვა. ვირტუალური კერძო ქსელების მაგალითები. უსადენო ქსელების მაგალითები.
15. აუტენტიფიკაციის პროტოკოლები. საერთო გასაღებზე დაფუძნებული აუტენტიფიკაციის მაგალითები. გასაღებების გამავრცელებელი ცენტრის დახმარებით, Kerberos პროტოკოლით და შიფრაციისას ღია გასაღების გამოყენებით აუტენტიფიკაციის მაგალითები. ელექტრონული მიმოწერის კონფიდენციალურობის მაგალითების განხილვა. ინტერნეტში ინფორმაციის დაცვის ამოცანები და მაგალითები. რესურსების უსაფრთხო სახელდების მაგალითების განხილვა. SSL დაცული სოკეტების პროტოკოლის გამოყენების მაგალითები.

პრაქტიკული სამუშაოს შესრულების ნიმუშები

სერვისის რეალიზება შეერთების დამყარების გარეშე

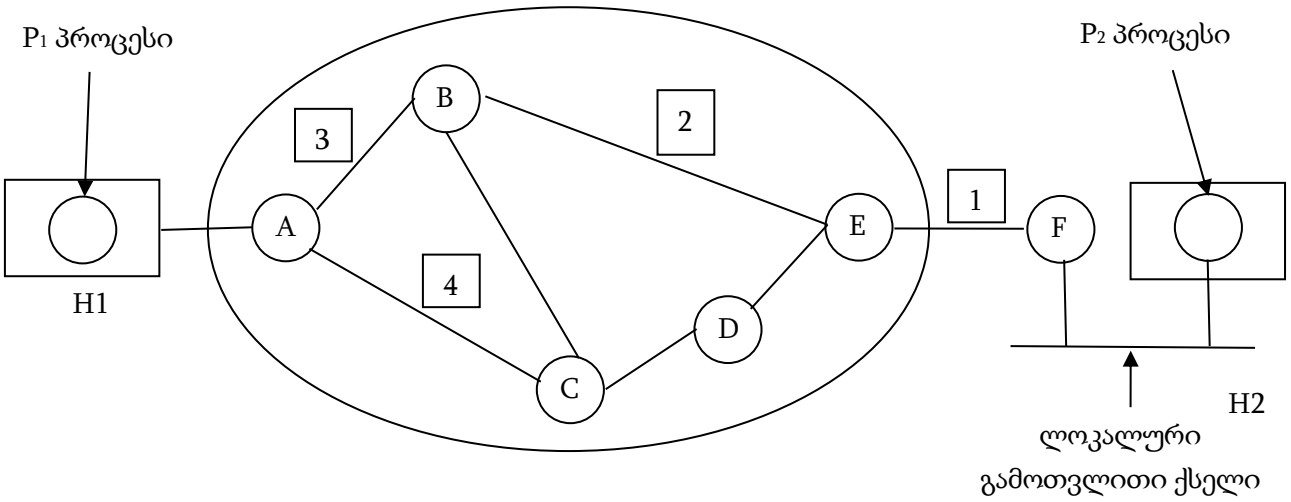
განვიხილოთ დეიტაგრამული ქვექსელების მუშაობის პრინციპი. დავუშვათ, P_1 პროცესი, რომელიც H_1 ჰოსთზე სრულდება, გრძელ შეტყობინებას უგზავნის P_2 პროცესს (ნახ. 1). P_1 პროცესი ამ შეტყობინებას გადასცემს სატრანსპორტო დონეს და ატყობინებს, რომ ეს შეტყობინება უნდა გადაეცეს P_2 პროცესს, რომელიც H_2 ჰოსთზე სრულდება. შეტყობინება სატრანსპორტო დონეს გადაეცემა სატრანსპორტო დონის სათაურთან ერთად, რომელიც შეტყობინების დასაწყისშია მოთავსებული.

დავუშვათ, შეტყობინება 4-ჯერ გრძელია პაკეტის მაქსიმალურ ზომაზე. ამიტომ, ქსელური დონე მას ჰყოფს 4 ნაწილად (1, 2, 3 და 4). ეს პაკეტები ეგზავნება A მარშრუტიზატორს ორწერტილიანი შეერთების რაიმე პროტოკოლის, მაგალითად PPP, გამოყენებით. თითოეულ მარშრუტიზატორს აქვს მარშრუტიზაციის ცხრილი, რომლის მიხედვით ის განსაზღვრავს პაკეტის შემდგომ გზას დანიშნულების თითოეული მისამართისთვის. ცხრილის თითოეული სტრიქონი ორი ველისგან შედგება: ადრესატი (დანიშნულება) და გამოსასვლელი ხაზი მოცემული ადრესატისთვის. მეორე ველში მითითებულია მხოლოდ ის არხები, რომლებიც უშუალოდაა მიერთებული მოცემულ მარშრუტიზატორთან. ნახ. 1-ზე A მარშრუტიზატორს აქვს ორი გამოსასვლელი ხაზი. ერთი უკავშირდება B-ს, მეორე C-ს. შესაბამისად, პაკეტები უნდა გადაეგზავნოს B ან C მარშრუტიზატორს, იმ შემთხვევაშიც კი, თუ ეს მარშრუტიზატორები არ არიან პაკეტების ადრესატები.

დავუშვათ, A-დან F-მდე გვინდა ამ ოთხი პაკეტის გაგზავნა. თავდაპირველად, 1, 2, 3 და 4 პაკეტები მიეწოდება A მარშრუტიზატორს და დროებით ინხება იქ, რადგან მოწმდება მათი კორექტული მოწოდება. შემდეგ, პირველი პაკეტი გადაეგზავნება B მარშრუტიზატორს ცხრილის მიხედვით. შემდეგ, B-დან E მარშრუტიზატორს და ბოლოს გადაეცემა F-ს. F მარშრუტიზატორზე პირველი პაკეტი გარდაიქმნება მონაცემების დონის კადრად და მიეწოდება H2 ჰოსთს ლოკალური ქსელით. ასეთივე მარშრუტით გადაიცემა მეორე და მესამე პაკეტები.

ახლა, დავუშვათ რომ, მე-4 პაკეტის გადაცემის მომენტში ABE გზაზე წარმოიქმნა საცობი, ამიტომ A მარშრუტიზატორმა ეს პაკეტი გადაუგზავნა C-ს. ამ დროს A აახლებს მარშრუტიზაციის ცხრილს (ნახ. 1-ზე „გაახლებული“ წარწერის ქვეშ). ახალი ცხრილიდან ჩანს, რომ E-თვის განკუთვნილი პაკეტები უნდა გადაეცეს C-ს.

ალგორითმს, რომელიც მართავს მარშრუტიზატორის ცხრილებს და იღებს გადაწყვეტილებებს, **მარშრუტიზაციის ალგორითმი** ეწოდება.



ნახ. 1. მარშრუტიზაცია დეიტაგრამულ ქვექსელში

ცხრილი 1

A მარშრუტიზატორის ცხრილი			
		გაახლებული	
A	-	A	-
B	B	B	B
C	C	C	C
D	C	D	C
E	B	E	C
F	B	F	C
ადრესატი	ხაზი	ადრესატი	ხაზი

ცხრილი 2

B მარშრუტიზატორის ცხრილი	
A	A
B	-
C	C
D	C
E	E
F	E
ადრესატი	ხაზი

ცხრილი 3

C მარშრუტიზატორის ცხრილი	
A	A
B	B
C	-
D	D
E	D
F	D
ადრესატი	ხაზი

ცხრილი 4

D მარშრუტიზატორის ცხრილი	
A	C
B	C
C	C
D	-
E	E
F	E
ადრესატი	ხაზი

ცხრილი 5

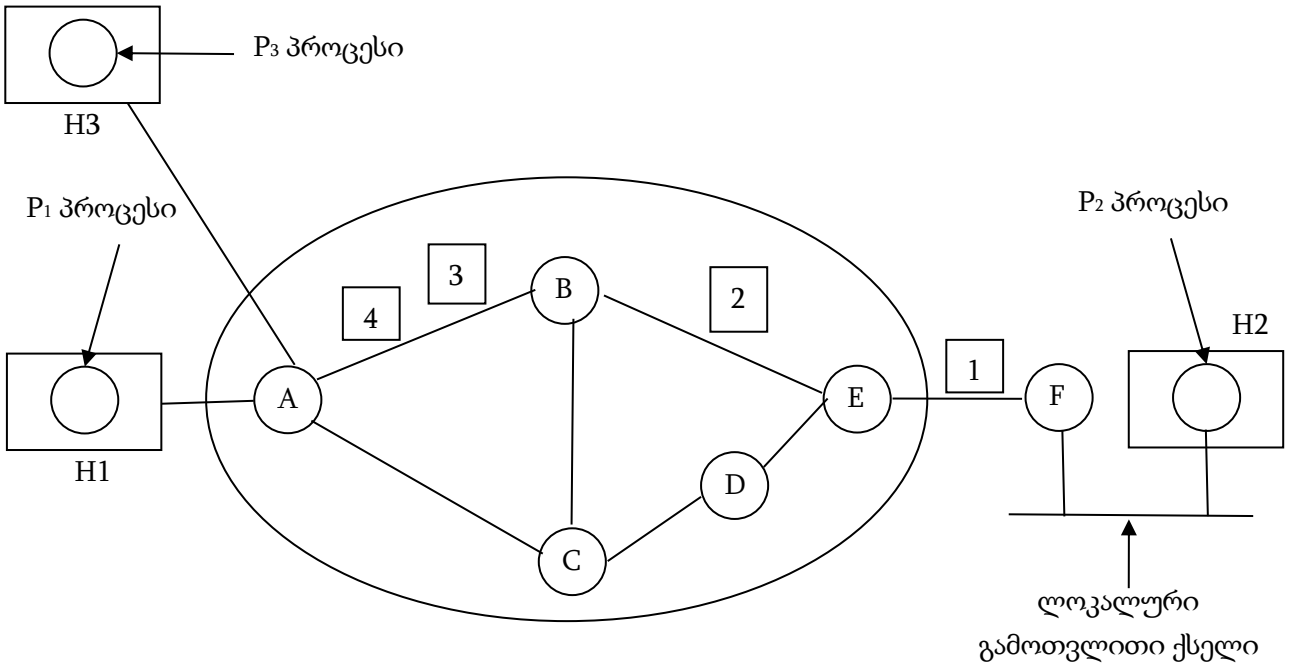
E მარშრუტიზატორის ცხრილი	
A	B
B	B
C	D
D	D
E	-
F	F
ადრესატი	ხაზი

სერვისის რეალიზება შეერთების დამყარებით

შეერთების დამყარებაზე ორიენტირებული სერვისები იყენებენ ვირტუალური არხის ქვექსელს. ამ დროს თითოეული პაკეტისათვის არ ხდება მარშრუტის არჩევა, არამედ მოცემული შეერთების ყველა პაკეტისთვის წინასწარ განისაზღვრება ფიქსირებული მარშრუტი. ეს მარშრუტი ინახება მარშრუტიზატორების ცხრილში. ასე მუშაობს, მაგალითად სატელეფონო სისტემა. ვირტუალური არხის გამოყენების შემთხვევაში თითოეული პაკეტი შეიცავს ვირტუალური არხის იდენტიფიკატორს.

განვიხილოთ მაგალითი (ნახ. 2). დავუშვათ, H1 ჰოსთი დაუკავშირდა H2 ჰოსთს. ეს შეერთება დაიმახსოვრება მარშრუტიზატორების ცხრილში პირველი სტრიქონის სახით. ასე, A მარშრუტიზატორის ცხრილის პირველი სტრიქონიდან ჩანს, რომ თუ H1 ჰოსთიდან მოსული პაკეტის იდენტიფიკატორია 1, მაშინ ის უნდა გადაიგზავნოს B მარშრუტიზატორზე იდენტიფიკატორით 1. ანალოგიურად, B მარშრუტიზატორის ცხრილის პირველი სტრიქონიდან ჩანს, რომ A-დან მოსული პაკეტი, რომლის იდენტიფიკატორია 1, უნდა გადაეგზავნოს E-ს იდენტიფიკატორით 1. დაბოლოს, E მარშრუტიზატორის ცხრილის პირველი სტრიქონიდან ჩანს, რომ B-დან მოსული პაკეტი, რომლის იდენტიფიკატორია 1, უნდა გადაეგზავნოს F მარშრუტიზატორს იდენტიფიკატორით 1.

ახლა, დავუშვათ, H3 ჰოსთმა მოინდომა შეერთების დამყარება H2 ჰოსთთან. ის ირჩევს შეერთების იდენტიფიკატორს 1, რადგან მოცემულ მომენტში ეს არის ერთადერთი არსებული შეერთება და ქვექსელს სთხოვს ვირტუალური არხის დაყენებას. მაგალითად, A მარშრუტიზატორის ცხრილს დაემატება მეორე სტრიქონი. აქ ადგილი აქვს კონფლიქტს. A მარშრუტიზატორს შეუძლია გაარჩიოს H1 და H3 ჰოსთებიდან მოსული პაკეტები, რომლებსაც ერთნაირი იდენტიფიკატორები აქვს (1). მაგრამ, B-ს არ შეუძლია მათი განსხვავება (B-ს ცხრილის მეორე სტრიქონს ექნებოდა ისეთივე სახე, როგორიც პირველს). ამიტომ, A მარშრუტიზატორი H3 ჰოსთიდან მოსულ პაკეტებს ანიჭებს ახალ იდენტიფიკატორს და ამით ქმნის მეორე შეერთებას. A-ს ცხრილის მეორე სტრიქონიდან ჩანს, რომ H3 ჰოსთიდან მოსული პაკეტი, რომლის იდენტიფიკატორია 1, უნდა გადაეგზავნოს B-ს იდენტიფიკატორით 2. B-ს ცხრილს დაემატება მეორე სტრიქონი, საიდანაც ჩანს, რომ A-დან მოსულ პაკეტს თუ აქვს იდენტიფიკატორი 2, ის უნდა გადაეგზავნოს E-ს იდენტიფიკატორით 2. ასევე, E-ს ცხრილს ემატება მეორე სტრიქონი, საიდანაც ჩანს, რომ B-დან მოსულ პაკეტს თუ აქვს ისენტიფიკატორი 2, ის უნდა გადაეგზავნოს F-ს იდენტიფიკატორით 2. სწორედ, ასეთი სახის კონფლიქტების თავიდან ასაცილებლად მარშრუტიზატორებს უნდა ჰქონდეს შეერთების იდენტიფიკატორების შეცვლის შესაძლებლობა გამოსასვლელ პაკეტებში. ამ პროცესს, ზოგჯერ **ჭდეების კომუტირებას** უწოდებენ.



ნახ. 2. მარშრუტიზაცია ვირტუალური არხის ქვექსელში

ცხრილი 6

A მარშრუტიზატორის ცხრილი				
H1	1		B	1
H3	1		B	2
შესასვ-ლელი			გამოსას-ვლელი	

ცხრილი 7

B მარშრუტიზატორის ცხრილი				
A	1		E	1
A	2		E	2
შესასვ-ლელი			გამოსას-ვლელი	

ცხრილი 8

E მარშრუტიზატორის ცხრილი				
B	1		F	1
B	2		F	2
შესასვ-ლელი			გამოსას-ვლელი	

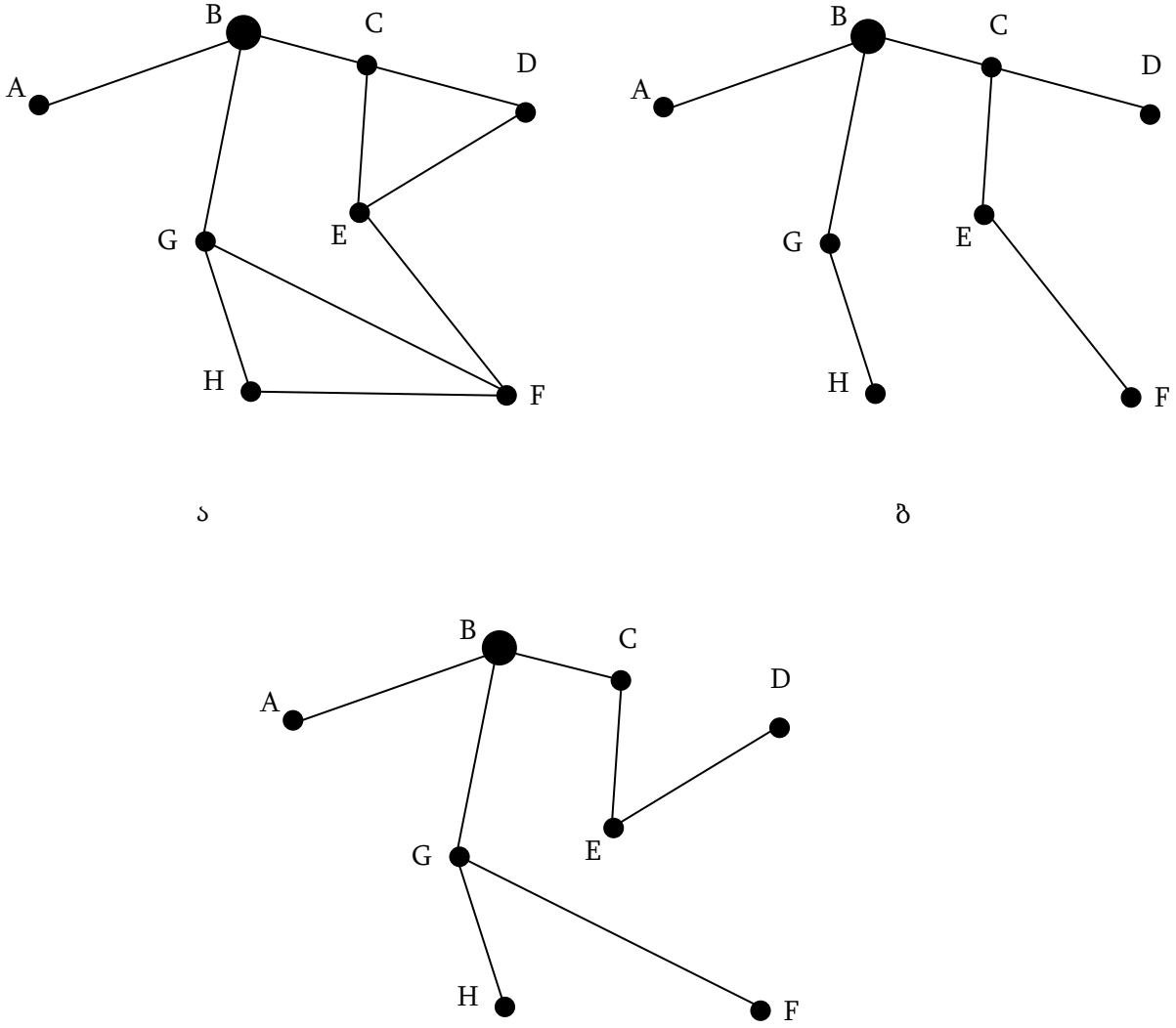
მარშრუტიზაციის ალგორითმები

ქსელური დონის მთავარი ფუნქციაა მარშრუტის არჩევა პაკეტისთვის საწყისი კვანძიდან საბოლოო კვანძამდე. თუ ქვექსელი იყენებს დეიტაგრამულ სამსახურს, მაშინ თითოეული პაკეტისთვის მარშრუტის არჩევა სრულდება ხელახლა, რადგან ოპტიმალური მარშრუტი შეიძლება შეცვლილი იყოს. თუ ქვექსელი იყენებს ვირტუალურ არხებს, მაშინ მარშრუტი აირჩევა მხოლოდ ვირტუალური არხის შექმნისას. ამის შემდეგ, ყველა პაკეტი გადაიცემა ამ მარშრუტით. პაკეტების ასეთ გადაცემას სეანსური მარშრუტიზაცია ეწოდება, რადგან მარშრუტი ძალაშია მომხმარებელის რეგისტრირებისას, ფაილის გადაცემისას და ა.შ.

მარშრუტიზაციასა და გადაგზავნას შორის შემდეგი განსხვავება არსებობს. **მარშრუტიზაცია** არის პროცესი, როცა სისტემა მარშრუტს ირჩევს, **გადაგზავნა** კი არის მოქმედება, რომელიც სრულდება პაკეტის მიღებისას. მარშრუტიზატორი ორ ფუნქციას ასრულებს. პირველია, შემოსული პაკეტების დამუშავება და მათთვის მარშრუტიზაციის ცხრილის საფუძველზე გამოსასვლელი ხაზის არჩევა. ამ პროცესს **გადაგზავნა** ეწოდება. მეორეა, მარშრუტიზაციის ცხრილების შევსება და გაახლება. სწორედ, ამ დროს მუშაობს მუშაობს მარშრუტიზაციის ალგორითმი.

მარშრუტის არჩევის ალგორითმებს უნდა ჰქონდეს შემდეგი თვისებები: კორექტულობა, სიმარტივე, საიმედოობა, მდგრადობა, სამართლიანობა და ოპტიმალურობა. ქსელების მუშაობის პროცესში მუდმივად აქვს ადგილი აპარატურის მტყუნებებს და შესაბამისად, ტოპოლოგიის ცვლილებებს. მარშრუტიზაციის ალგორითმებმა თავი უნდა გაართვას აღნიშნულ პრობლემებს ქსელის გადატვირთვისა და ჰოსთებზე მომუშავე პროცესების შეწყვეტის გარეშე. ჩამოთვლილი მიზნები, რიგ შემთხვევებში, ურთიერთწინააღმდეგობრივია.

მარშრუტიზაციის ალგორითმები შეიძლება ორ კლასად დავყოთ: ადაპტური და არაადაპტური. **არაადაპტური** ალგორითმები მარშრუტის არჩევისას არ ითვალისწინებენ ტოპოლოგიას, ქსელის მდგომარეობას და არ ზომავენ ტრაფიკს ხაზებზე. ისინი მარშრუტს წინასწარ ირჩევენ კვანძების თითოეული წყვილისთვის. მარშრუტების სია ჩაიტვირთება მარშრუტიზატორებში ქსელის ჩატვირთვის დროს. ამ პროცესს **სტატიკური მარშრუტიზაცია** ეწოდება. **ადაპტური** ალგორითმები ირჩევენ სხვადასხვა მარშრუტებს ქსელის ტოპოლოგიის ცვლილებისა და ხაზის დატვირთულობის მიხედვით.



ნახ. 5. ქვექსელი (ა); შესასვლელი ხეები B მარშრუტიზატორისთვის (ბ,გ)

უმოკლესი გზის არჩევა

მარშრუტის არჩევის არსი მდგომარეობს ქვექსელის გრაფის აგებასა და მასში უმოკლესი გზის პოვნაში. გრაფის თითოეული კვანძი შეესაბამება მარშრუტიზატორს, თითოეული რკალი კი - კავშირის ხაზს (რომელსაც კავშირსაც უწოდებენ). მარშრუტის არჩევისას ალგორითმი გრაფზე პოულობს უმოკლეს გზას ორ კვანძს (მარშრუტიზატორს) შორის.

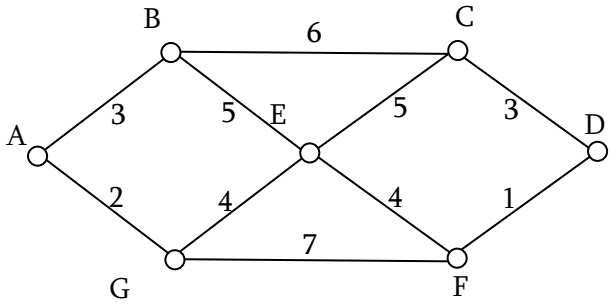
მანძილი ორ მარშრუტიზატორს შორის შეიძლება გაიზომოს ტრანზიტული უბნების რაოდენობით ან ხაზის ფიზიკური სიგრძით. თითოეულ რკალს შეიძლება შევუსაბამოთ რიგის საშუალო სიგრძე ან გადაგზავნის საშუალო დრო, რომელიც განისაზღვრება საათში ერთხელ სპეციალური პაკეტის გადაცემის გზით. თუ მანძილი იზომება ტრანზიტული უბნების რაოდენობით, მაშინ ABC და ABE გზების სიგრძეები ერთნაირია (ნახ. 3). თუ მანძილი იზომება კილომეტრებში, მაშინ ABC გზა გაცილებით გრძელია ABE გზაზე (ნახაზზე დაცული მასშტაბი). თუ მანძილი განისაზღვრება გადაგზავნის საშუალო დროის მიხედვით, მაშინ უმოკლეს გზად ჩაითვლება ყველაზე სწრაფი გზა.

ზოგადად, გრაფის რკალის პარამეტრები დამოკიდებულია მანძილზე, გამტარუნარიანობაზე, საშუალო დატვირთვაზე, რიგის საშუალო სიგრძეზე, კავშირის ღირებულებაზე, დაყოვნების სიდიდესა და სხვა ფაქტორებზე.

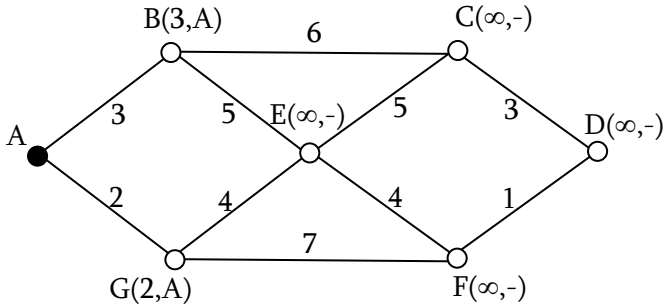
არსებობს გრაფის ორ კვანძს შორის უმოკლესი გზის გამოთვლის ალგორითმები. ერთ-ერთი მათგანი შეიმუშავა დეიქსთრამ 1959 წელს. მისი არსი შემდეგში მდგომარეობს. როგორც კი მუშა კვანძის ყველა მეზობელი კვანძი იქნება გამოკვლეული და დროებითი მონიშვნები მიწერილი (აუცილებლობის შემთხვევაში, დროებითი მონიშვნები შეიძლება შეიცვალოს), მთელ გრაფში უნდა მოიძებნოს კვანძი, რომელსაც აქვს ყველაზე მცირე დროებითი მონიშვნა, რომელიც მოინიშნება როგორც ძირითადი. შემდეგ, ამ კვანძისთვის ვიმეორებთ აღწერილ პროცესს. ეს პროცესი გრძელდება მანამ, სანამ არ მივადგებით საბოლოო კვანძს.

ალგორითმის მუშაობა განვიხილოთ წონადი არამიმართული გრაფის მაგალითზე (ნახ. 3, ა). დავუშვათ, წონის კოეფიციენტები შეესაბამება მანძილებს კვანძებს შორის. რკალებს მივაწეროთ ეს მანძილები. ერთი კვანძი შეესაბამება ერთ მარშრუტიზატორს. ჩვენი მიზანია ვიპოვოთ უმოკლესი გზა A მარშრუტიზატორიდან D-მდე.

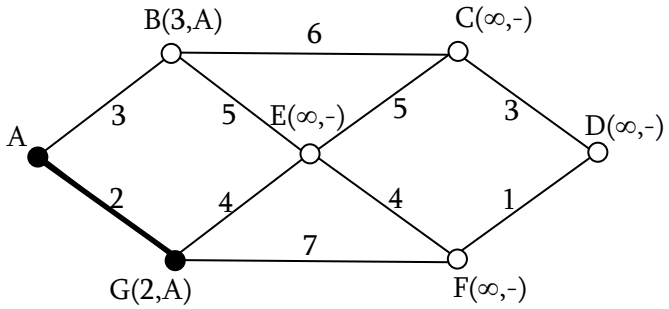
რადგან, A კვანძიდან ვიწყებთ უმოკლესი გზის ძებნას, ამიტომ ის შავი წრით მოვნიშნოთ, როგოც მუდმივი. შემდეგ, უნდა განვიხილოთ მისი მეზობელი კვანძები B და G. თითოეულ მათგანს მრგვალ ფრჩხილებში უნდა მივაწეროთ დროებითი მონიშვნები (ნახ. 3, ბ). მონიშვნა დროებითია, რადგან ის შემდგომში შეიძლება შეიცვალოს და მუდმივი გახდეს. დროებითი მონიშვნა ორი ელემენტისგან შედგება: მანძილი და კვანძი. მაგალითად, B კვანძს მიეწერება (3,A) მონიშვნა, სადაც პირველი ელემენტი მანძილია A კვანძამდე, მეორე კი - თვით კვანძი. ანალოგიურად, G კვანძს მიეწერება დროებითი (2, A) მონიშვნა. დანარჩენ წვეროებს მივაწეროთ მონიშვნა, რომელიც შედგება „∞“ და „-“ სიმბოლოებისგან, რადგან უცნობია გზა ამ კვანძებამდე და შესაბამისი მანძილები. მას შემდეგ, რაც A კვანძის ყველა მეზობელი კვანძი გამოკვლეულია და თითოეულ მათგანს მიწერილი აქვს დროებითი მონიშვნები, მთელ გრაფში უნდა მოიძებნოს კვანძი, რომელსაც აქვს ყველაზე მცირე დროებითი მონიშვნა. დროებითი მონიშვნა აქვს B და G კვანძებს. მათ შორის მინიმალური მონიშვნა აქვს G კვანძს. ამიტომ, ის მოინიშნება როგორც მუდმივი (მუქი წრე) და ხდება ახალი მუშა კვანძი (ნახ. 6, გ). ამ დროისთვის გვაქვს უმოკლესი მანძილი AG.



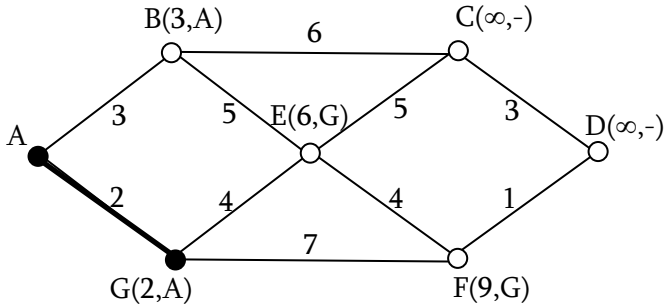
ა



ბ

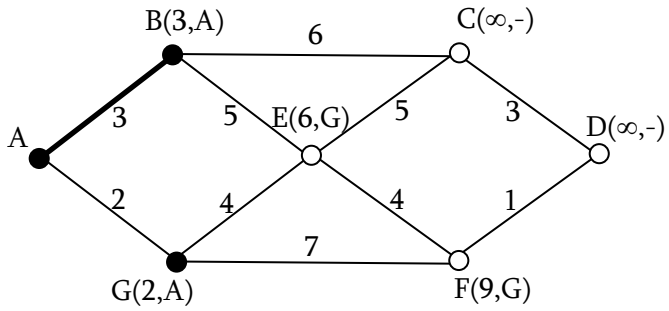


გ

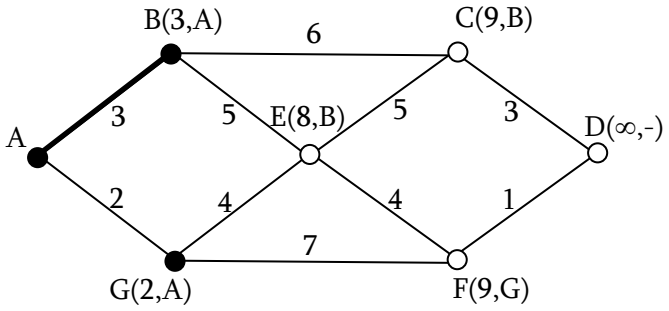


დ

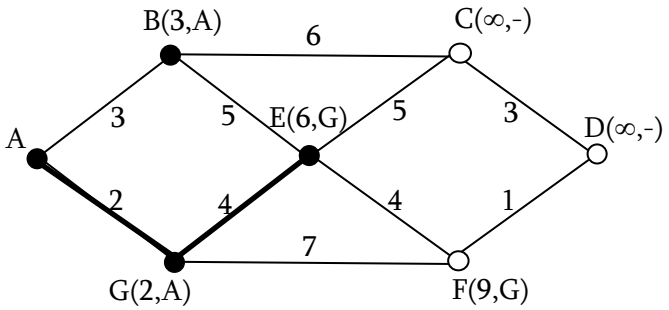
ნახ. 3. A-დან D-მდე უმოკლესი გზის გამოთვლის ბიჯები.



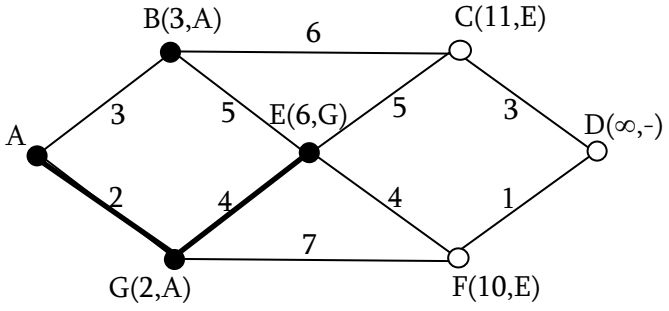
ე



ვ

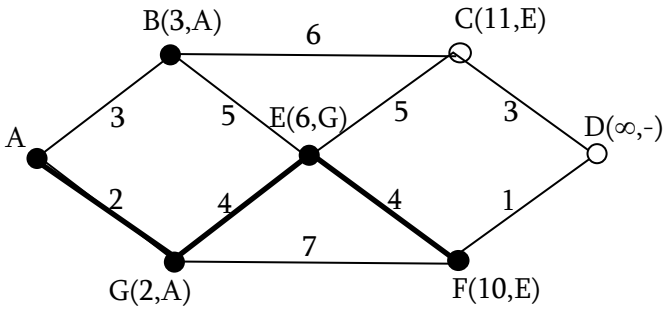


ზ

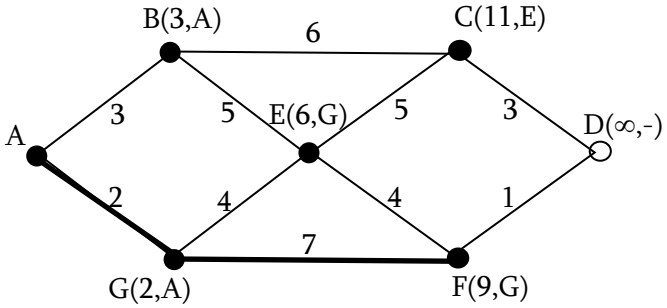


თ

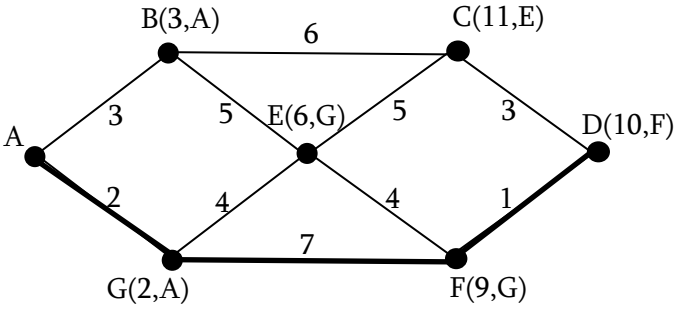
ნახ. 3. A-დან D-მდე უმოკლესი გზის გამოთვლის ბიჯები.



ი



ჰ



ლ

ნახ. 3. A-დან D-მდე უმოკლესი გზის გამოთვლის ბიჯები.

ზემოთ აღწერილი პროცესი გავიმეოროთ G კვანძისთვის. მისი მეზობლებია E და F. მივაწეროთ მათ დროებითი მონიშვნები. E კვანძის დროებითი მონიშვნის განსაზღვრისათვის G კვანძის დროებით მონიშვნას (2) უნდა მივუმატოთ G-დან E-მდე მანძილი (4). მივიღებთ 6-ს. შესაბამისად, E კვანძს მიეწერება (6, G) (ნახ. 3, დ). ანალოგიურად, F კვანძს მიეწერება მონიშვნა, რომელიც არის G წვეროს მონიშვნისა (2) და G-დან F-მდე მანძილის (7) ჯამი, ანუ F კვანძს მიეწერება დროებითი მონიშვნა (9,G). რადგან, G კვანძის ყველა მეზობელი კვანძი გამოკვლეულია და მიწერილი აქვს დროებითი მონიშვნები, მთელ გრაფში უნდა მოიძებნოს მინიმალური დროებითი მონიშვნის მქონე კვანძი. დროებითი მონიშვნის მქონე კვანძებია B, E და F. მათ შორის მინიმალური მონიშვნა აქვს B-ს. ამიტომ, ის მოინიშნება როგორც მუდმივი და ხდება ახალი მუშა წვერო (ნახ. 3, ე). ამ დროისთვის გვაქვს უმოკლესი გზა AB.

ზემოთ აღწერილი პროცესი გავიმეოროთ B კვანძისთვის. მას ორი მეზობელი ჰყავს C და E. C კვანძის დროებითი მონიშვნის განსაზღვრისთვის B კვანძის მონიშვნას (3) უნდა მივუმატოთ მანძილი B და C კვანძებს შორის (6). მივიღებთ (9,B) მონიშვნას (ნახ. 3, ვ). E კვანძის დროებითი მონიშვნის განსაზღვრისათვის B კვანძის მონიშვნას (3) უნდა დავუმატოთ მანძილი B და E კვანძებს შორის (5). მივიღებთ (8,B) მონიშვნას. დროებითი მონიშვნა აქვს C, E და F კვანძებს. B კვანძი უკვე გამოკვლეულია და მიწერილი აქვს მუდმივი მონიშვნა, ამიტომ მას არ განვიხილავთ. მათ შორის მინიმალური დროებითი მონიშვნა აქვს E კვანძს. მაგრამ, როგორც ვხედავთ E კვანძის მონიშვნა გაიზარდა: იყო 6 და გახდა 8. ეს იმას ნიშნავს, რომ მიღებულია უფრო გრძელი (არაოპტიმალური) გზა. ამიტომ, ვუბრუნდებით ნახ. 3, დ-ს და E კვანძს მოვნიშნავთ როგორც მუდმივს (ნახ. 3, ზ). ამ დროისთვის გვაქვს უმოკლესი გზა AGE.

ზემოთ აღწერილი პროცესი გავიმეოროთ E კვანძისთვის. მას ორი მეზობელი ჰყავს C და F. C კვანძის დროებითი მონიშვნა იქნება (11,E) (ნახ. 3, თ), რადგან E კვანძის მონიშვნას (6) დამატებული მანძილი E-დან C-მდე (5) არის 11. შესაბამისად, F წვეროს მიეწერება დროებითი მონიშვნა (10,E). ამის შემდეგ, მთელ გრაფში იძებნება მინიმალური დროებითი მონიშვნის მქონე კვანძი. დროებითი მონიშვნის მქონე კვანძებია C(11,E) და F(10,E). მათ შორის მინიმალური დროებითი მონიშვნა აქვს F კვანძს. ამიტომ, ის მოინიშნება როგორც მუდმივი (ნახ. 3, ი). ამ დროისთვის გვაქვს უმოკლესი გზა AGEF. მაგრამ, როგორც ვხედავთ, F კვანძის მონიშვნა გაიზარდა: იყო 9 და გახდა 10. ეს იმას ნიშნავს, რომ მიღებულია უფრო გრძელი (არაოპტიმალური) გზა. ამიტომ, ვუბრუნდებით ნახ. 3, ზ-ს და F კვანძს მოვნიშნავთ როგორც მუდმივს (ნახ. 3, კ). ამ დროისთვის გვაქვს უმოკლესი გზა AGF.

F კვანძს ერთი მეზობელი ჰყავს D, რომელიც საბოლოოა. მას მიეწერება მონიშვნა (10,F) (ნახ. 3, ლ). რადგან დროებითი მონიშვნის მქონე კვანძები აღარ გვაქვს და D კვანძი ბოლოა, ამიტომ, ალგორითმი მუშაობას ამთავრებს. ამ დროისთვის გვაქვს უმოკლესი გზა AGFD.

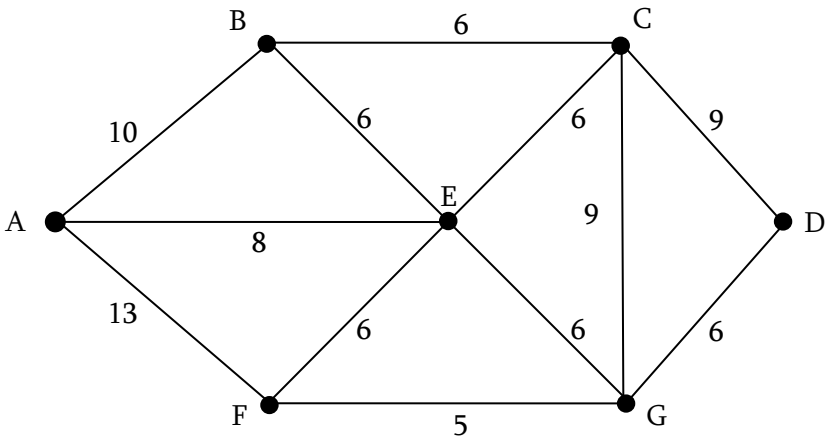
მარშრუტიზაცია მანძილის ვექტორის მიხედვით

არსებობს მარშრუტიზაციის სტატიკური და დინამიკური ალგორითმები. სტატიკური ალგორითმები არ ითვალისწინებენ ქსელის მიმდინარე დატვირთვას. ამიტომ, თანამედროვე კომპიუტერულ ქსელებში გამოიყენება დინამიკური ალგორითმები. მათგან გავრცელებულია ორი: მარშრუტიზაცია მანძილის ვექტორის მიხედვით და მარშრუტიზაცია არხების მდგომარეობის გათვალისწინებით.

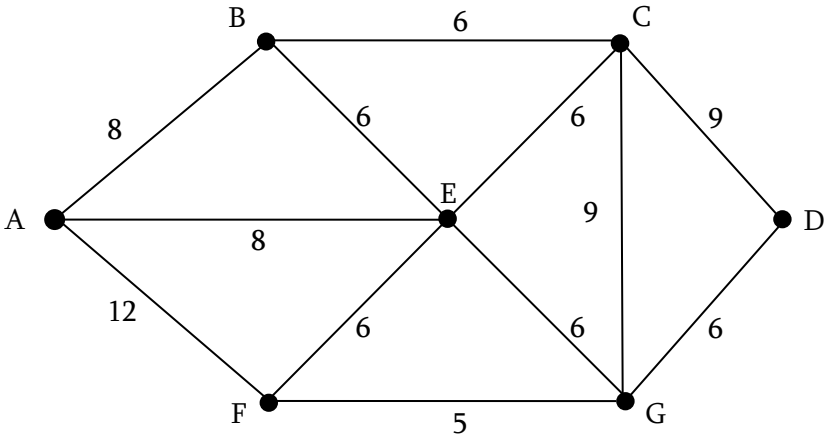
მანძილის ვექტორის მიხედვით მარშრუტიზაციის ალგორითმები იყენებენ მარშრუტიზატორების ცხრილებს, რომლებიც შეიცავენ საუკეთესო ცნობილ გზებს თითოეულ ადრესატამდე. თითოეული მარშრუტიზატორი თავის ცხრილებში მოთავსებული მონაცემების გაახლებისთვის ასრულებს ინფორმაციის გაცვლას მეზობელ მარშრუტიზატორებთან.

მანძილის ვექტორის მიხედვით მარშრუტიზაციის ალგორითმი ატარებს მისი შემქმნელების სახელებს: **ბელმან-ფორდის** განაწილებული ალგორითმი და **ფორდი-ფულკერსონის** ალგორითმი.

მანძილის ვექტორის მიხედვით მარშრუტიზაციისას ცხრილები შეიცავენ ჩანაწერებს ქვექსელის თითოეული მარშრუტიზატორის შესახებ. ჩანაწერი ორი ნაწილისაგან შედგება: ხაზის უპირატესი ნომერი მოცემული მიმღებისათვის და სავარაუდო მანძილი ან პაკეტის გავლის დრო ამ მიმღებამდე. ზომის ერთეულად შეიძლება გამოვიყენოთ ტრანზიტული უბნების რაოდენობა, მილიწამები, პაკეტების რაოდენობა, რომლებიც რიგში იცდიან მოცემული მიმართულებით და



ნახ. 4. ქვექსელი, რომლისთვისაც გამოითვლება მარშრუტი A-დან D-მდე.



ნახ. 5. ქვექსელი, რომლისთვისაც გამოითვლება მარშრუტი A-დან D-მდე.

ცხრილი 2.1.

	A მარშრუტიზა-ტორის ცხრილი		B მარშრუტიზა-ტორის ცხრილი		C მარშრუტიზა-ტორის ცხრილი		D მარშრუტიზა-ტორის ცხრილი		E მარშრუტიზა-ტორის ცხრილი		F მარშრუტიზა-ტორის ცხრილი		G მარშრუტიზა-ტორის ცხრილი	
მიმღები	ხაზის ნომერი	პაკეტის გავლის დრო	ხაზის ნომერი	პაკეტის გავლის დრო	ხაზის ნომერი	პაკეტის გავლის დრო	ხაზის ნომერი	პაკეტის გავლის დრო	ხაზის ნომერი	პაკეტის გავლის დრო	ხაზის ნომერი	პაკეტის გავლის დრო	ხაზის ნომერი	პაკეტის გავლის დრო
A	A	0	B	10	B	16	C	25	E	8	A	13	F	18
B	B	10	B	0	B	6	C	15	B	6	E	12	E	12
C	B	16	C	6	C	0	C	9	C	6	E	12	C	9
D	B	25	C	15	D	9	D	0	G	12	G	11	D	6
E	E	8	E	6	E	6	G	12	E	0	E	6	E	6
F	F	13	F	12	E	12	G	11	F	6	F	0	F	5
G	F	18	E	12	G	9	G	6	G	6	G	5	G	0

ცხრილი 2.2.

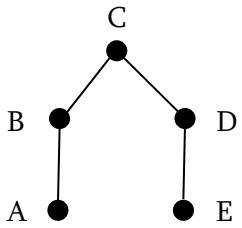
	A მარშრუტიზა-ტორის ცხრილი		B მარშრუტიზა-ტორის ცხრილი		C მარშრუტიზა-ტორის ცხრილი		D მარშრუტიზა-ტორის ცხრილი		E მარშრუტიზა-ტორის ცხრილი		F მარშრუტიზა-ტორის ცხრილი		G მარშრუტიზა-ტორის ცხრილი	
მიმღები	ხაზის ნომერი	პაკეტის გავლის დრო	ხაზის ნომერი	პაკეტის გავლის დრო	ხაზის ნომერი	პაკეტის გავლის დრო	ხაზის ნომერი	პაკეტის გავლის დრო	ხაზის ნომერი	პაკეტის გავლის დრო	ხაზის ნომერი	პაკეტის გავლის დრო	ხაზის ნომერი	პაკეტის გავლის დრო
A	A	0	B	8	B	16	C	25	E	8	A	12	F	18
B	B	8	B	0	B	6	C	15	B	6	E	12	E	12
C	B	16	C	6	C	0	C	9	C	6	E	12	C	9
D	B	25	C	15	D	9	D	0	G	12	G	11	D	6
E	E	8	E	6	E	6	G	12	E	0	E	6	E	6
F	F	12	F	12	E	12	G	11	F	6	F	0	F	5
G	E	14	E	12	G	9	G	6	G	6	G	5	G	0

უსასრულობამდე თვლის პრობლემა

მანძილის ვექტორის მიხედვით მარშრუტიზაციის ალგორითმს აქვს ნაკლი - უმოკლესი გზის ძებნას შეიძლება ძალიან დიდი დრო დასჭირდეს. ამის მიზეზია ის, რომ ალგორითმი საკმაოდ სწრაფად რეაგირებს ახალ კარგ ამბებზე (მაგალითად, გაფუჭებული მარშრუტიზატორი ამუშავდა, გამოჩნდა უფრო მოკლე გზა რომელიმე მარშრუტიზატორამდე და ა.შ.), ვიდრე ცუდ ახალ ამბებზე (მაგალითად, მარშრუტიზატორის მწყობრიდან გამოსვლა და ა.შ.). დავუშვათ, A მარშრუტიზატორიდან B მარშრუტიზატორამდე დაყოვნება დიდია და ვექტორების მორიგი გაცვლისას A მარშრუტიზატორს მისმა მეზობელმა C მარშრუტიზატორმა შეატყობინა, რომ მისგან (C-გან) B მარშრუტიზატორამდე არის უფრო მცირე დაყოვნება. ამ შემთხვევაში, A მარშრუტიზატორი პაკეტებს გაუგზავნის B მარშრუტიზატორს C მარშრუტიზატორის გავლით. ე.ი. კარგი ახალი ამბავი გავრცელდა ინფორმაციის ერთი გაცვლის განმავლობაში.

ზემოთ თქმული განვიხილოთ კონკრეტულ მაგალითზე. დავუშვათ, გვაქვს წრფივი ქვექსელი, რომელიც ხუთი მარშრუტიზატორისაგან შედგება (ნახ. 6). მარშრუტიზატორებს შორის მანძილის საზომად გამოვიყენოთ სატრანზიტო კვანძების რაოდენობა. დავუშვათ, აგრეთვე, რომ A მარშრუტიზტორი მწყობრიდანაა გამოსული და დანარჩენმა მარშრუტიზატორებმა იციან ამის შესახებ. ისინი თვლიან, რომ მანძილი A მარშრუტიზატორამდე უსასრულობის ტოლია.

როცა A მარშრუტიზტორი ამუშავდება, დანარჩენი მარშრუტიზატორები ამის შესახებ იგებენ ვექტორების გაცვლის გზით. პირველი გაცვლის შემდეგ B იგებს, რომ მის მარცხნივ მყოფ მეზობელს A-თან კავშირისას აქვს ნულოვანი დაყოვნება და B-თვის მარშრუტების ცხრილში მონიშნავს, რომ A იმყოფება მარცხნივ ერთი სატრანზიტო კვანძის მანძილზე. ამ დროისთვის, დანარჩენი მარშრუტიზატორები თვლიან, რომ A გამორთულია. A-თვის ცხრილებში მოთავსებული დაყოვნებების მნიშვნელობები ნაჩვენებია ნახ. 6, ა-ს მეორე სტრიქონზე. ინფორმაციის მომდევნო გაცვლის შემდეგ C იგებს, რომ B-ს აქვს გზა A-კენ, რომლის სიგრძეა 1. ამიტომ, C აახლებს თავის ცხრილს და A-მდე მანძილს უთითებს 2-ის ტოლად. ამ მომენტისთვის D და E-მ არ იციან, რომ A ჩართულია. ასე გრძელდება მანამ, სანამ ყველა მარშრუტიზატორი არ გაიგებს A-ს არსებობის შესახებ. როგორც ვხედავთ, კარგი ახალი ამბავი ვრცელდება სიჩქარით 1 ტრანზიტული უბანი ვექტორების 1 გაცვლისას. თუ ქვექსელში ყველაზე გრძელი გზა N ტრანზიტული უბნისაგან შედგება, მაშინ ქვექსელის ყველა მარშრუტიზატორს ჩართული მარშრუტიზატორებისა და ამუშავებული ხაზების შესახებ ეცოდინება ინფორმაციის N გაცვლის შემდეგ.



A	B	C	D	E	
	●	●	●	●	დასაწყისში
1	●	●	●	1 გაცვლის შემდეგ
1	2	●	●	2 გაცვლის შემდეგ
1	2	3	●	3 გაცვლის შემდეგ
1	2	3	4	4 გაცვლის შემდეგ

ა

A	B	C	D	E	
1	2	3	4	დასაწყისში
3	2	3	4	1 გაცვლის შემდეგ
3	4	3	4	2 გაცვლის შემდეგ
5	4	5	4	3 გაცვლის შემდეგ
5	6	5	6	4 გაცვლის შემდეგ
7	6	7	6	5 გაცვლის შემდეგ
7	8	7	8	6 გაცვლის შემდეგ

ბ

ნახ. 6. უსასრულობამდე თვლის პრობლემა

ამრიგად, პრობლემის არსი იმაში მდგომარეობს, რომ C აუწყებს B-ს, რომ აქვს რაღაც გზა, მაგრამ B-ს არ შეუძლია დაადგინოს თვით მასზე გადის თუ არა ეს გზა.

მანძილის ვექტორის მიხედვით მარშრუტიზაცია გამოიყენებოდა 1979 წლამდე. ამ ალგორითმის გამოყენებაზე უარი ითქვა შემდეგი მიზეზების გამო. პირველი, ალგორითმი გზის არჩევისას არ ითვალისწინებს გზის გამტარუნარიანობას. საქმე ისაა, რომ ადრე ყველა ხაზის გამტარუნარიანობა იყო 56 კბიტი/წმ. ამჟამად არსებობს ხაზები, რომელთა გამტარუნარიანობაა 230 კბიტი/წმ, 1544 კბიტი/წმ. ამიტომ, ხაზის გამტარუნარიანობის გათვალისწინება აუცილებელი გახდა. მეორე, ალგორითმი ძალიან დიდხანს მიდის მდგრად მდგომარეობამდე.

მარშრუტიზაცია ხაზის მდგომარეობის გათვალისწინებით

მანძილის ვექტორის მიხედევით მარშრუტიზაციის ალგორითმი მისი უარყოფითი მხარეების გამო 1979 წლიდან შეიცვალა ახალი ალგორითმით, რომელსაც ეწოდება **მარშრუტიზაცია ხაზის მდგომარეობის გათვალისწინებით**. ამჟამად, ფართოდ გამოიყენება ამ ალგორითმის სხვადასხვა ვარიანტები. ალგორითმს საფუძვლად უდევს 5 მოქმედება, რომელიც თითოეულმა მარშრუტიზატორმა უნდა შეასრულოს [3]:

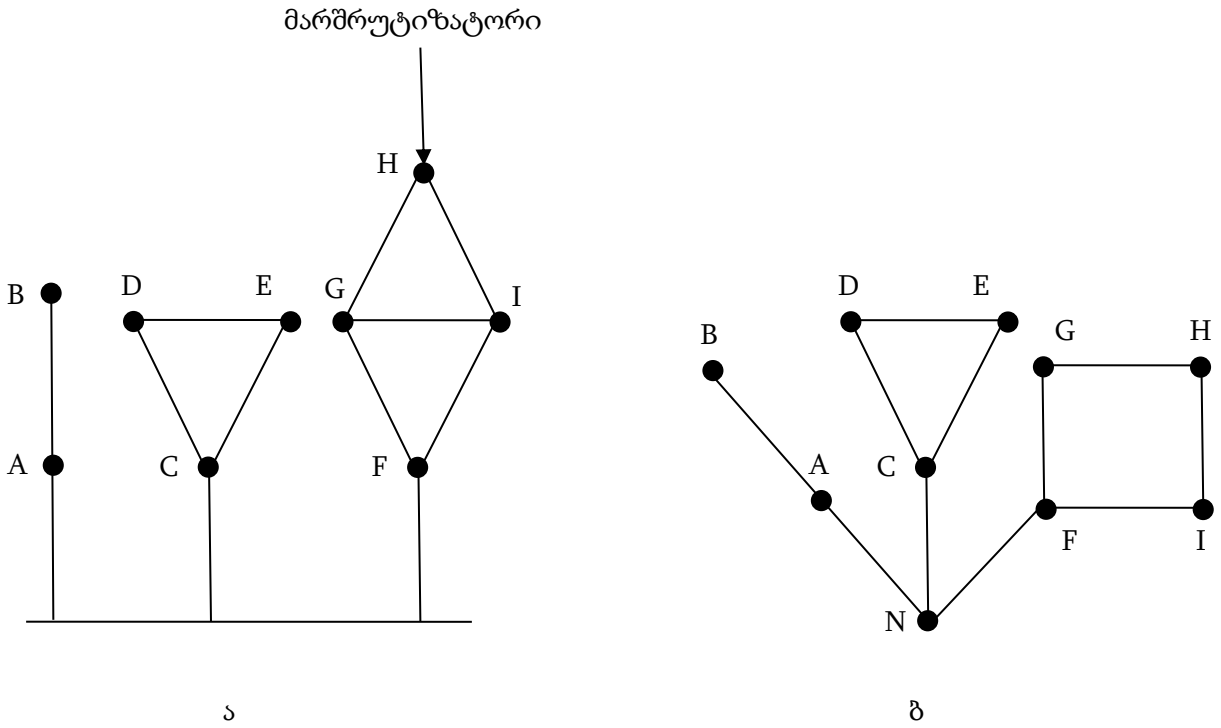
1. მეზობლების აღმოჩენა და მათი ქსელური მისამართების გაგება.
2. თითოეულ მეზობელთან დაყოვნების ან კავშირის ღირებულების გაზომვა.
3. პაკეტის ფორმირება, რომელიც შეიცავს წინა ორ ეტაპზე შეგროვებულ ინფორმაციას.
4. ფორმირებული პაკეტის გაგზავნა ყველა მარშრუტიზატორებისთვის.
5. ყველა მარშრუტიზატორისკენ უმოკლესი გზის განსაზღვრა.

ამ მოქმედებების შესრულების შედეგად თითოეულ მარშრუტიზატორს ეგზავნება ქსელის სრული ტოპოლოგია და დაყოვნებების მნიშვნელობები. ამის შემდეგ, უმოკლესი გზის საპოვნელად, თითოეული მარშრუტიზატორისაკენ შეგვიძლია დეიქსტრას ალგორითმის გამოყენება.

მეზობლების გაცნობა

როცა მარშრუტიზატორი იტვირთება, პირველ რიგში მან ინფორმაცია უნდა მიიღოს თავისი მეზობლების შესახებ. ამისთვის, ის ყველა ორწერტილიან ხაზში გზავნის სპეციალურ HELLO პაკეტს. პასუხად ხაზის მეორე ბოლოში მყოფმა მარშრუტიზატორმა უნდა გამოგზავნოს ინფორმაცია თავის შესახებ. მარშრუტიზატორების სახელები უნდა შედგებოდეს ლათინური ანბანის ასოებისა და ციფრებისაგან და უნდა იყოს უნიკალური, მაგალითად, და ROUTER_1, GTU-01 ა.შ.

როცა რამდენიმე მარშრუტიზატორი გაერთიანებულია ლოკალურ ქსელში, მაშინ სიტუაცია რთულდება. ნახ. 7, ა-ზე A, B და F მარშრუტიზატორები მიერთებულია როგორც ლოკალურ ქსელთან, ისე სხვა მარშრუტიზატორებთან. ასეთ შემთხვევაში ლოკალური ქსელი შეგვიძლია წარმოვადგინოთ გრაფის კვანძის სახით, რომელსაც უკავშირდება A, B და F მარშრუტიზატორები. თუ გვინდა პაკეტის გადაცემა A მარშრუტიზატორიდან C-მდე, მაშინ ისინი გაივლიან ANC გზას.

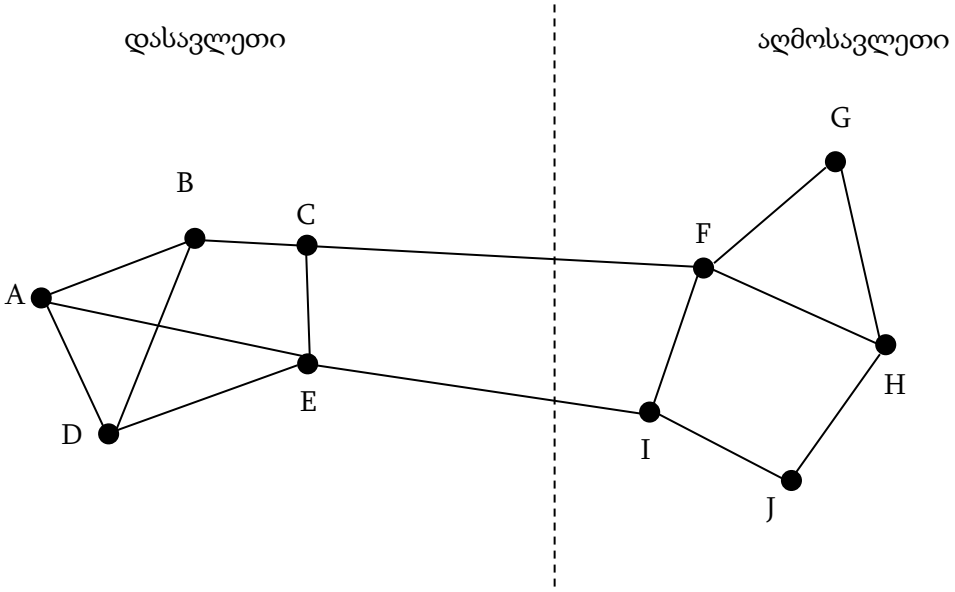


ნახ. 7. ცხრა მარშრუტიზატორი და ლოკალური ქსელი (ა); ამავე სისტემის გრაფული მოდელი (ბ)

ხაზის ღირებულების გაზომვა

ნებისმიერ მარშრუტიზატორს უნდა ჰქონდეს ინფორმაცია თითოეულ თავის მეზობლამდე ხაზის შესახებ, ან დაყოვნებების მნიშვნელობები. დაყოვნების გასაზომად გამოიყენება ECHO სპეციალური პაკეტი. A მარშრუტიზატორი ECHO პაკეტს უგზავნის B მარშრუტიზატორს. B ვალდებულია დაუყოვნებლივ უპასუხოს A-ს, ე.ი. პაკეტი მიდის A-დან B-მდე, შემდეგ B-დან A-მდე. როგორც კი პაკეტი მოვა B მარშრუტიზატორიდან გაზომილი დრო ორზე გაიყოფა. შედეგად, მიიღება დაყოვნების მნიშვნელობა A და B მარშრუტიზატორებს შორის. ორ მარშრუტიზატორს შორის უფრო ზუსტი დაყოვნების მისაღებად ზემოთ აღწერილი პროცესი უნდა გავიმეოროთ რამდენიმეჯერ და გამოვთვალოთ მიღებული მნიშნელობების საშუალო არითმეტიკული. აქ იგულისხმება, რომ დაყოვნებები A-დან B-მდე და B-დან A-მდე სიმეტრიულია, ანუ აქვთ ერთნაირი მნიშვნელობები.

დაყოვნებების გაზომვისას შეგვიძლია გავითვალისწინოთ ან არ გავითვალისწინოთ ხაზზე დატვირთვა. თუ გვინდა ხაზზე დატვირთულობის გათვალისწინება, მაშინ ECHO პაკეტის გაგზავნისას უნდა ჩავრთოთ ტაიმერი. ხაზის დატვირთულობის იგნორირებისათვის კი ტაიმერი უნდა ჩავრთოთ მაშინ, როცა ECHO პაკეტი მიაღწევს რიგის დასაწყისს. ამ ორივე მიდგომის სასარგებლოდ არსებობს არგუმენტები. თუ დაყოვნების გაზომვისას ვითვალისწინებთ ხაზის დატვირთულობას ანუ ტრაფიკს, მაშინ ორი ერთნაირი გამტარუნარიანობის მქონე ხაზიდან აირჩევა აირჩევა ის, რომელსაც ნაკლები დატვირთვა აქვს. ითვლება, რომ ასეთი ხაზი უფრო მოკლეა. შედეგად, აჩეული ხაზი უფრო მეტად დაიტვირთება, რაც ამაღლებს სისტემის მუშაობის ეფექტურობას.



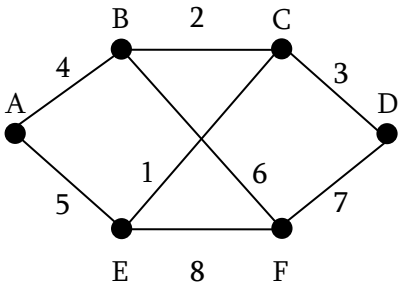
ნახ. 8. ქვექსელი, რომელშიც აღმოსავლეთი და დასავლეთი ნაწილები შეერთებულია ორი ხაზით

არსებობს არგუმენტი ხაზის დატვირთულობის იგნორირების სასარგებლოდ. განვიხილოთ ქვექსელი, რომელიც ორ ნაწილადაა გაყოფილი (ნახ. 8). პირველი ნაწილი მეორეს უკავშირდება CF და EI ხაზებით. დავუშვათ, პაკეტების ძირითადი ნაკადი გადაიცემა CF ხაზით. შედეგად, ერთი ხაზი აღმოჩნდება ძალიან დატვირთული და პაკეტები გადაიცემა დიდი დაყოვნებებით. დაყოვნებების გაზომვისას აღმოჩნდება, რომ EI ხაზი გაცილებით ნაკლებადაა დატვირთული და პაკეტები გადაიცემა ნაკლები დაყოვნებებით. ამიტომ, მარშრუტიზაციის ცხრილების გაახლების შემდეგ EI ჩაითვლება მოკლე მარშრუტად და მასზე გადაინაცვლებს პაკეტების ნაკადის დიდი ნაწილი. ამის შემდეგ, დაყოვნებების გაზომვისას აღმოჩნდება რომ CF ხაზი ნაკლებად დატვირთულია და ის ჩაითვლება მოკლე მარშრუტად. მარშრუტიზაციის ცხრილების გაახლების შემდეგ პაკეტების დიდი ნაწილი გადაინაცვლებს CF ხაზზე და ა.შ. შედეგად, ადგილი ექნება მუდმივ რხევებს, რაც გამოიწვევს სისტემის მუშაობის ეფექტურობის დაქვეითებას. ეს პრობლემა არ წამოიჭრება, თუ ხაზის დატვირთვებს არ გავითვალისწინებთ.

ამ პრობლემის ერთ-ერთი გადაწყვეტაა ის, რომ დატვირთვა, ჩვენი მაგალითის შემთხვევაში, შეგვიძლია გავანაწილოთ ორ ხაზს შორის, ზოგადად კი - რამდენიმე ხაზს შორის. მაგრამ, ამ შემთხვევაში, საუკეთესო გზა არ იქნება ეფექტურად გამოყენებული. ზოგად შემთხვევაში, უმჯობესია დატვირთვა განაწილდეს რამდენიმე ხაზს შორის (ანუ ტრაფიკის ცალკეული ნაწილები გადავანაწილოთ ამ ხაზზე), ვიდრე ადგილი ჰქონდეს სისტემის რხევებს.

ხაზის მდგომარეობის პაკეტების გავრცელება

ეს ყველაზე რთული ეტაპია. სირთულე იმაში მდგომარეობს, რომ მარშრუტიზატორები პაკეტების მიღებასთან ერთად იწყებენ თავიანთი მარშრუტების შეცვლას ანუ ცხრილების გაახლებას. შედეგად, სხვადასხვა მარშრუტიზატორებს ექნებათ ქსელის ტოპოლოგიის სხვადასხვა ვერსიები. ამან შეიძლება, გამოიწვიოს წინააღმდეგობები, მარშრუტებში ყულფებისა და მიუწვდომელი მანქანების გაჩენა და ა.შ.



ა

ხაზის მდგომარეობის პაკეტები

A	
რიგითი ნომერი	
ასაკი	
B	4
E	5

B	
რიგითი ნომერი	
ასაკი	
A	4
C	2
F	6

C	
რიგითი ნომერი	
ასაკი	
B	2
D	3
E	1

D	
რიგითი ნომერი	
ასაკი	
C	3
F	7

E	
რიგითი ნომერი	
ასაკი	
A	5
C	1
F	8

F	
რიგითი ნომერი	
ასაკი	
B	6
D	7
E	6

ბ

ნახ. 11. ქვექსელი (ა); ამ ქვექსელის ხაზის მდგომარეობის პაკეტები (ბ)

ხაზის მდგომარეობის პაკეტების გავრცელების ალგორითმი იყენებს ჩასხმის ალგორითმს. როგორც ვიცით, თითოეულ პაკეტში სათაურის შემდეგ მოთავსებულია რიგითი ნომერი, რომელიც ერთით იზრდება ყოველი მომდევნო პაკეტისთვის. როცა მარშრუტიზატორი იღებს ხაზის მდგომარეობის ახალ პაკეტს, ის თავის ცხრილში ეძებს გამომგზავნი მარშრუტიზატორის მისამართს და პაკეტის რიგით ნომერს. თუ ეს ახალი პაკეტია, მაშინ ის გადაიცემა ყველა ხაზზე, გარდა იმ ხაზისა, საიდანაც ეს პაკეტი მოვიდა. თუ მოსული პაკეტი დუბლიკატია, ანუ არსებობს პაკეტი მოსული იმავე მარშრუტიზატორისაგან და აქვს იგივე ნომერი, რაც უკვე მიღებულ პაკეტს, მაშინ ის წაიშლება. თუ მოსული პაკეტის რიგითი ნომერი ნაკლებია იმავე მარშრუტიზატორიდან უკვე მიღებული პაკეტის რიგით ნომერზე, მაშინ ეს პაკეტი წაიშლება როგორც მოძველებული.

ალგორითმის მუშაობისას ადგილი აქვს შემდეგ პრობლემებს:

1. როცა პაკეტის რიგითი ნომერი აღწევს მაქსიმალურ მნიშვნელობას, ის განულდება. ასეთი პაკეტი ჩაითვლება მოძველებულად და უარიყოფა როგორც მოძველებული. გამოსავალია რიგითი ნომრისთვის 4 ბაიტის გამოყენება.
2. როცა მარშრუტიზატორი მწყობრიდან გამოდის, მაშინ იკარგება პაკეტის ნომერი. თუ ეს პაკეტი ხელახლა გაიგზავნება ნულოვანი რიგითი ნომრით, მაშინ ის უარიყოფა როგორც მოძველებული.
3. როცა არასწორად იცვლება პაკეტის რიგითი ნომერი, მაშინ ზოგიერთი მარშრუტიზატორის მიერ ხდება პაკეტების გარკვეული ნაწილის იგნორირება. მაგალითად, თუ პაკეტის რიგითი ნომერი უნდა ყოფილიყო 32 და შეცდომით გახდა 1021, მაშინ პაკეტები, რომელთა რიგითი ნომრებია 33÷1021, უარიყოფა მარშრუტიზატორების მიერ როგორც მოძველებული.

ამ პრობლემების გადასაწყვეტად პაკეტში რიგითი ნომრის შემდეგ მოათავსეს პაკეტის ასაკი. პაკეტის ასაკი ერთით მცირდება ყოველი წამის შემდეგ. როცა მისი მნიშვნელობა განულდება, მაშინ ამ მარშრუტიზატორიდან მოსული ინფორმაცია იშლება. ჩვეულებრივ, ახალი პაკეტი მოდის, მაგალითად ყოველ 10 წამში ერთხელ, ამიტომ ინფორმაცია მარშრუტიზატორის შესახებ მოძველდება, როცა ის გამორთულია. ასაკის მნიშვნელობა ერთით მცირდება, აგრეთვე თითოეული მარშრუტიზატორის მიერ ჩასხმის პროცესის დაწყების დროს. ამით, გარანტირებულია ის, რომ პაკეტი არ დაიკარგება და არ იარსებებს (იცხოვრებს) მუდმივად.

ამ ალგორითმის საიმედოობის ამაღლების მიზნით შეიმუშავეს მიდგომა, რომელიც შემდეგში მდგომარეობს. როცა ხაზის მდგომარეობის პაკეტი მოდის მარშრუტიზატორზე ჩასხმისთვის, ის რიგში არ დგება მაშინვე გადასაცემად. პაკეტი გარკვეული დროის განმავლობაში ინახება ბუფერში (შუალედური შენახვის უბანში). თუ ამ პერიოდის განმავლობაში გამომგზავნი მარშრუტიზატორიდან მოდის კიდევ ერთი პაკეტი, მაშინ მარშრუტიზატორი ადარებს მათ რიგით ნომრებს. წაიშლება უფრო ძველი პაკეტი, წაიშლება, აგრეთვე დუბლიკატი თუ ნომრები ერთნაირია. მარშრუტიზატორებს შორის კავშირის ხაზებზე შეცდომებისაგან თავის დასაცავად ხაზის მდგომარეობის ყველა პაკეტის მიღება დასტურდება. როგორც კი ხაზი გათავისუფლდება, მარშრუტიზატორი ამოწმებს დროებითი შენახვის უბანს, საიდანაც პაკეტები ამოირჩევა გადასაცემად და დასადასტურებლად.

ნახ. 9-ზე ნაჩვენებია B მარშრუტიზატორის პაკეტების ბუფერი, რომელიც გამოიყენება B-ს მიერ ნახხ. 11, ა-ზე ნაჩვენებ ქვექსელთან მუშაობისთვის. ცხრილის თითოეული სტრიქონი შეესაბამება ახლად მიღებულ ხაზის მდგომარეობის პაკეტს, რომელიც ბოლომდე არ არის დამუშავებული. როგორც ნახაზიდან ჩანს, ცხრილი შეიცავს პაკეტის გამომგზავნი მარშრუტიზატორის მისამართს, რიგით ნომერს, ასაკს, დაგზავნის ალმებს B მარშრუტიზატორის სამი ხაზიდან თითოეულისთვის (A, C და F-კენ შესაბამისად), დადასტურების ალმებს B მარშრუტიზატორის სამი ხაზიდან თითოეულისთვის და მონაცემებს. თუ დაგზავნის ალამი „1“ მდგომარეობაშია, ეს იმას ნიშნავს, რომ პაკეტი უნდა გადაიგზავნოს შესაბამისი ხაზით. თუ დადასტურების ალამი „1“ მდგომარეობაშია, ეს იმას ნიშნავს, რომ უნდა დადასტურდეს პაკეტის მიღება შესაბამისი ხაზიდან [1-4].

ცხრილის (ნახ. 9) პირველი სტრიქონიდან ჩანს, რომ ხაზის მდგომარეობის პაკეტი მოვიდა A მარშრუტიზატორიდან. დაგზავნის ალმებიდან ჩანს, რომ პაკეტი უნდა გადაეგზავნოს C და F მარშრუტიზატორებს. დადასტურების ალმებიდან ჩანს, რომ პაკეტის მიღების შესახებ დადასტურება უნდა გაეგზავნოს A-ს. ანალოგიურად, ცხრილის მეორე სტრიქონიდან ჩანს, რომ ხაზის მდგომარეობის პაკეტი მოვიდა F-დან. დაგზავნის ალმები გვიჩვენებენ, რომ პაკეტი F-დან უნდა გაეგზავნოს A და C-ს. დადასტურების ალმები გვიჩვენებენ, რომ F-ს უნდა გაეგზავნოს დადასტურება.

ცხრილის მესამე სტრიქონიდან ჩანს, რომ მესამე პაკეტი მოვიდა E-დან. პაკეტი მიღებული იყო ორჯერ EAB და EFB ხაზებით. ამიტომ ის უნდა გავუგზავნოთ C-ს, ხოლო დადასტურება კი A და F მარშრუტიზატორებს.

			დაგზავნის ალმები			დადასტურების ალმები			
წყარო	რიგითი ნომერი	ასაკი	A	C	F	A	C	F	მონაცემები
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

ნახ. 9. B მარშრუტიზატორის პაკეტების ბუფერი ნახ. 8-დან

თუ ბუფერში ორიგინალი პაკეტის ყოფნისას მოვიდა მისი დუბლიკატი, მაშინ დაგზავნისა და დადასტურების ალმების ბიტების მნიშვნელობები უნდა შეიცვალოს. მაგალითად, თუ C მარშრუტიზატორის მდგომარეობის ასლი მოვა F-დან მანამ, სანამ გაგზავნილი იქნება ცხრილის მეოთხე სტრიქონი, მაშინ დაგზავნის ალმები მიიღებენ 100 მნიშვნელობას, დადასტურების ალმები კი - 011 მნიშვნელობებს. ეს კი იმას ნიშნავს, რომ ეს პაკეტი უნდა გაეგზავნოს A-ს (და არა F-ს), ხოლო C და F-თვის კი უნდა გაიგზავნოს დადასტურება. ცხრილის მეხუთე სტრიქონი მიიღებს სახეს:

C	20	60	1	0	0	0	1	1	

იერარქიული მარშრუტიზაცია

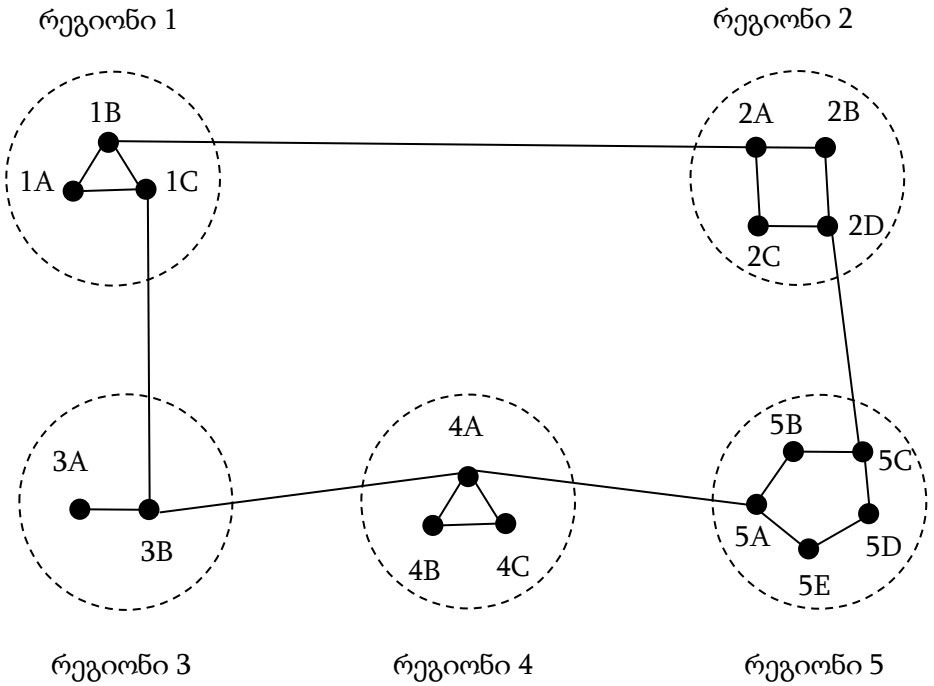
ქსელის ზომის ზრდასთან ერთად პროპორცილად იზრდება მარშრუტების ცხრილების ზომები. შესაბამისად, იზრდება მათი დამუშავებისთვის საჭირო ცენტრალური პროცესორის დრო. იზრდება, აგრეთვე სამომსახურეო პაკეტების ზომა. ეს კი ზრდის ხაზზე დატვირთვას, რადგან ამ პაკეტებს ერთმანეთს უგზავნიან მარშრუტიზატორები. თუ ქსელის ზომა ძალიან დიდია, მაშინ შეიძლება შეუძლებელი გახდეს მარშრუტიზატორებზე მარშრუტიზაციის ცხრილების შენახვა. გამოსავალი იმაშია, რომ დიდ ქსელებში მარშრუტიზაცია უნდა განხორციელდეს იერარქიულად.

იერარქიული მარშრუტიზაციის შემთხვევაში მარშრუტიზატორები იყოფა რეგიონებად. ერთი რეგიონის შიგნით თითოეულმა მარშრუტიზატორმა იცის მარშრუტი ამავე რეგიონის სხვა მარშრუტიზატორებამდე. მაგრამ, ერთი რეგიონის მარშრუტიზატორებმა არაფერი არ იციან სხვა რეგიონის სტრუქტურის შესახებ. რამდენიმე ქსელის გაერთიანებისას ისინი უნდა განვიხილოთ როგორც ცალკეული რეგიონები. ერთ რეგიონში შემავალმა მარშრუტიზატორებმა არ უნდა იცოდეს სხვა რეგიონების ტოპოლოგიები. ძალიან დიდი ზომის ქსელებში შეიძლება საჭირო გახდეს მრავალდონიანი იერარქია. კერძოდ, რეგიონები შეიძლება დაჯგუფდეს კლასტერებში, კლასტერები ზონებში, ზონები ჯგუფებში და ა.შ. [5-8].

ნახ. 10-ზე ნაჩვენებია ორდონიანი იერარქიის მაგალითიხუთი რეგიონით. 1A მარშრუტიზატორის ცხრილი შედგება 17 სტრიქონისგან, ხოლო იერარქიული მარშრუტიზაციის გამოყენების შემთხვევაში კი 7 სტრიქონისგან. ზოგად შემთხვევაში, რაც უფრო მსხვილია რეგიონები, მით უფრო მცირეა ცხრილის ზომა. ახლა განვიხილოთ ცხრილის სტრიქონები. როგორც ცხრილიდან ჩანს 1A-დან 2A-მდე პაკეტი გაივლის 1B ხაზს და შედეგად, 2 ტრანზიტულ უბანს. ეს მარშრუტი შეგვიძლია ასე ჩავწეროთ: 1A-1B-2A. ანალოგიურად, 1A-დან 2D-მდე პაკეტი გაივლის 1B ხაზს და შესაბამისად 4 ტრანზიტულ უბანს. ეს მარშრუტი შეგვიძლია ასე ჩავწეროთ:

1A-1B-2A-2B-2D. განვიხილოთ კიდევ ერთი მარშრუტი 1A-დან 5B-მდე. ეს გზა იყენებს 1C ხაზს სულ 5 ტრანზიტულ უბანს. მარშრუტს ექნება სახე: 1A-1C-3B-4A-5A-5B.

ასეთი მიდგომის ნაკლია ის, რომ ცხრილის ზომის შემცირებასთან ერთად იზრდება გზის სიგრძე. მაგალითად, საუკეთესო მარშრუტი 1A-დან 5C-მდე გადის მეორე რეგიონზე, მაგრამ იერარქიული მარშრუტიზაციის გამოყენებისას მთელი ტრაფიკი მეხუთე რეგიონში გადაიცემა მესამე რეგიონის გავლით. ასე უმჯობესია მეხუთე რეგიონის მარშრუტიზატორების უმრავლესობისათვის.



ა

სრული ცხრილი 1A-თვის		
დანიშნულება	ხაზი	ტრანზიტული უბანი
1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

ბ

იერარქიული ცხრილი 1A-თვის		
დანიშნულება	ხაზი	ტრანზიტული უბანი
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	3
4	1C	3
5	1C	4

გ

ნახ. 10. იერარქიული მარშრუტიზაცია

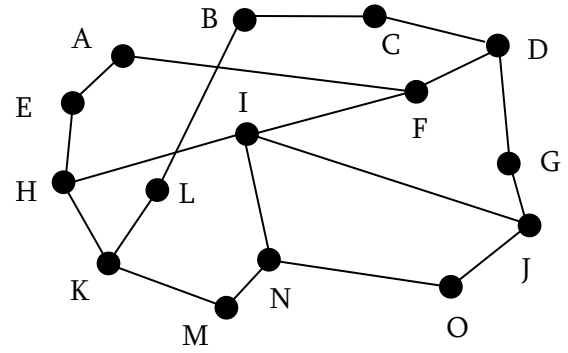
ბუნებრივია ისმის კითხვა: რამდენი იერარქიული დონე უნდა გვქონდეს? მაგალითად, თუ ქსელი შედგება 720 მარშრუტიზატორისგან და იერარქია არ გვაქვს, მაშინ თითოეულ მარშრუტიზატორს ექნება 720 სტრიქონიანი ცხრილი. თუ ქვექსელს დავყოფთ 24 რეგიონად თითოეულში 30 მარშრუტიზატორით, მაშინ თითოეული მარშრუტიზატორის ცხრილში იქნება 53 სტრიქონი. აქედან 30 სტრიქონი შეიცავს ინფორმაციას მოცემული რეგიონის 30 მარშრუტიზატორის შესახებ, დანარჩენი 23 კი - სხვა რეგიონების შესახებ. სამდონიანი იერარქიის არჩევისას, რომელიც შედგება 8 კლასტერისაგან, თითოეული კლასტერი 9 რეგიონისაგან, თითოეული რეგიონი - 10 მარშრუტიზატორისაგან, მაშინ თითოეული მარშრუტიზატორის ცხრილში იქნება 25 სტრიქონი. აქედან, 10 სტრიქონი ლოკალური მარშრუტიზატორებისთვის, 8 - მოცემული კლასტერის სხვა რეგიონებისთვის, 7 - სხვა კლასტერებისთვის. ზოგად შემთხვევაში, იერარქიის დონეების რაოდენობა ქვექსელისთვის, რომელიც შედგება N მარშრუტიზატორისგან lnN-ის ტოლია, ხოლო თითოეული მარშრუტიზატორის ცხრილის სტრიქონების რაოდენობა იქნება elnN-ის ტოლი. გარდა ამისა, ეფექტური იერარქიული მარშრუტიზაციით გამოწვეული საშუალო გზის ზომის ზრდა საკმაოდ მცირეა და ამიტომ, მისაღები.

ფართოსამაუწყებლო მარშრუტიზაცია

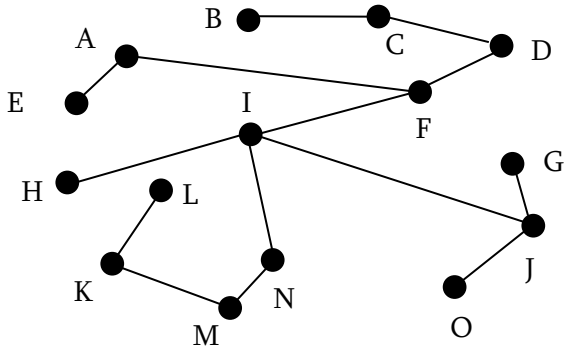
ფართომაუწყებლობა ეწოდება პაკეტების დაგზავნას დანიშნულების ყველა ჰოსთისკენ. ამ დროს ნებისმიერ დაინტერესებულ ჰოსთს შეუძლია დაგზავნილი მონაცემების მიღება. ფართოსამაუწყებლო დაგზავნის მაგალითებია ამინდის პროგნოზის გავრცელება, რადიოპროგრამების პირდაპირ ეთერში გავრცელება, ფასიანი ქაღალდების საბანკო კურსების გაახლება, სამხედრო საქმე და ა.შ.

ფართომაუწყებლობის რეალიზებისთვის სხვადასხვა მეთოდები გამოიყენება. ერთ-ერთია **ჩასხმის** მეთოდი. მაგრამ, მისი გამოყენება აქვეითებს ქსელის გამტარუნარიანობას. მეორე მეთოდია მრავალმისამართიანი მარშრუტიზაცია. მისი გამოყენებისას თითოეულ პაკეტში ათავსებენ ადრესატების სიას ან ბიტობრივ კარტას, რომლებიც შეიცავენ დანიშნულების უპირატეს ჰოსთებს. იღებს რა ასეთ პაკეტს, მარშრუტიზატორი განსაზღვრავს მომდევნო დაგზავნისთვის საჭირო გამოსასვლელი ხაზების ნაკრებს. მარშრუტიზატორი ქმნის პაკეტის ასლს თითოეული გამოყენებული გამავალი ხაზისთვის. მასში მოთავსდება მხოლოდ ის ადრესატები, რომლებთანაც მიმართვისათვის მოითხოვება მოცემული ხაზი. შედეგად, დაგზავნის სია ნაწილდება გამავალ ხაზებს შორის. დაგზავნების კარკვეული რაოდენობის შემდეგ პაკეტში დარჩება დანიშნულების მხოლოდ ერთი მისამართი და პაკეტი მიიღებს ჩვეულებრივ სახეს.

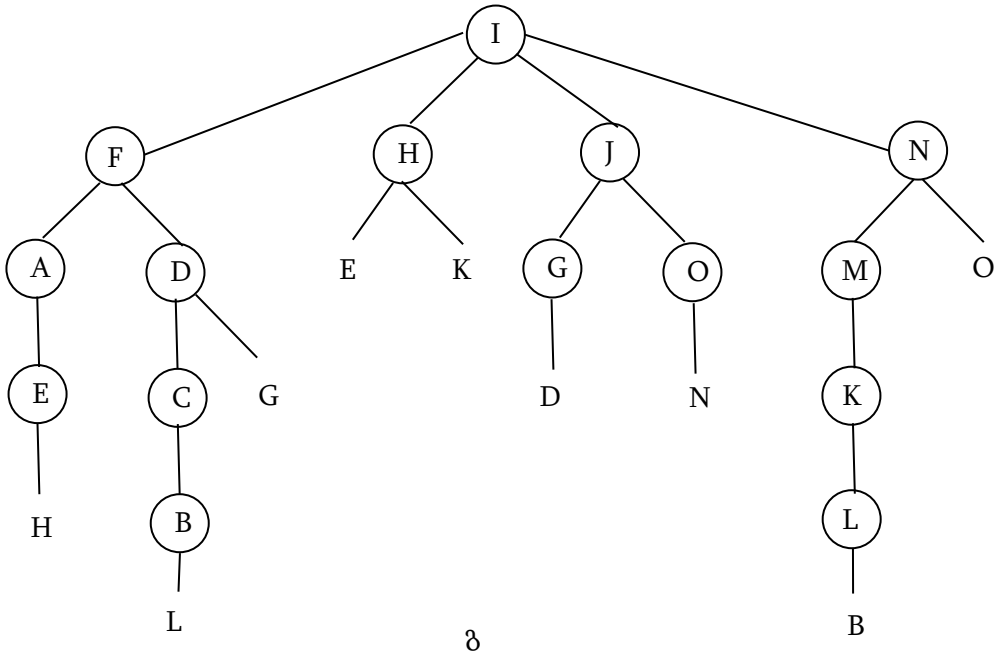
ფართომაუწყებლობის რეალიზების კიდევ ერთი მეთოდი იყენებს საყრდენ ან მაკავშირებელ ხეს. მაკავშირებელი ხე არის ქვექსელის ქვესიმრავლე, მოიცავს ყველა მარშრუტიზატორს და არ შეიცავს ჩაკეტილ გზებს. თითოეულმა მარშრუტიზატორმა იცის თუ მისი რომელი ხაზი ეკუთვნის მაკავშირებელ ხეს, მას შეუძლია პაკეტი გადასცეს მაკავშირებელი ხის ყველა ხაზით, იმ ხაზის გარდა, საიდანაც ეს პაკეტი მოვიდა. ამ მეთოდის დადებითი მხარეა ის, რომ ოპტიმალურად ხდება ქსელის გამტარუნარიანობის გამოყენება და წარმოიქმნება პაკეტების მინიმალური რაოდენობა. მეთოდის ნაკლია ის, რომ თითოეულ მარშრუტიზატორს უნდა ჰქონდეს ინფორმაცია მაკავშირებელი ხის შესახებ. ეს ინფორმაცია ხან მისაწვდომია, მაგალითად ხაზების მდგომარეობის გათვალისწინებით მარშრუტიზაციის შემთხვევაში, ხან არა, მაგალითდ მანძილების ვექტორების მიხედვით მარშრუტიზაციის შემთხვევაში.



ა



ბ



გ

ნახ. 5.11. შემხვედრი გზის გასწვრის წაწევა: ქვექსელი (ა); მაკავშირებელი ხე (ბ); ხე, აგებული შემხვედრი გზის გასწვრის წაწევის მეთოდით (გ)

მრავალმისამართიანი დაგზავნა

პრაქტიკაში, ხშირად გვხვდება დანართები, რომლებშიც პროცესები გაერთიანებული არიან ჯგუფებში. მაგალითად, პროცესების ჯგუფის სახით შეიძლება რეალიზებული იყოს მონაცემთა განაწილებული ბაზა. თუ ჯგუფის ერთ პროცესს სურს შეტყობინების გაგზავნა ამავე ჯგუფის სხვა წევრებთან და ჯგუფი მცირე ზომისაა, მაშინ საკმარისია ჯგუფის თითოეულ წევრს ცალკე შეტყობინება გავუგზავნოთ. თუ ჯგუფის ზომა დიდია, მაშინ ასეთი დაგზავნა აღმოჩნდება ძვირადღირებული.

ასეთი ჯგუფის წევრებისათვის შეტყობინების გადაცემას მრავალმისამართიანი დაგზავნა ეწოდება, ხოლო ამ ალგორითმს - **მრავალმისამართიანი მარშრუტიზაცია**.

ერთ-ერთი გადაწყვეტაა ფართომაუწყებლობის გამოყენება. თუმცა მისი გამოყენება ასობით ჰოსთის ინფორმირებისთვის მილიონი კვანძისგან შემდგარ ქსელში არაეფექტურია. ეს გამოწვეულია იმით, რომ ჰოსთების უმრავლესობა არ იქნება დაინტერესებული ამ შეტყობინების მიღებით ან არიან დაინტერესებული, მაგრამ საჭიროა მათგან ამ ინფორმაციის დამალვა. ამრიგად, საჭიროა შეტყობინებების დაგზავნის საშუალება მკაცრად განსაზღვრული ჯგუფებისთვის, რომლებიც რაოდენობრივად ძალიან დიდია, მაგრამ მცირეა მთელ ქსელთან შედარებით.

განვიხილოთ მრავალმისამართიანი მარშრუტიზაციის რეალიზების ერთ-ერთი საშუალება. მარშრუტიზატორებმა უნდა იცოდნენ თუ რომელი ჰოსთი რომელ ჯგუფს ეკუთვნის. ამისთვის, ჰოსთმა უნდა შეატყობინოს თავის მარშრუტიზატორებს ჯგუფის შემადგენლობაში ცვლილებების შესახებ ან მარშრუტიზატორებმა თვითონ პერიოდულად უნდა გამოკითხონ თავიანთო ჰოსთები. ასეთი გზით მარშრუტიზატორები იგებენ თუ მათი ჰოსთებიდან რომელი რომელ ჯგუფს ეკუთვნის. ამის შესახებ მარშრუტიზატორები აუწყებენ თავიანთ მეზობლებს. ასე ვრცელდება ეს ინფორმაცია მთელ ქსელში.

მრავალმისამართიანი დაგზავნის მიზნით თითოეული მარშრუტიზატორი ახდენს მაკავშირებელი ხის გათვლას, რომელიც ფარავს ქვექსელის ყველა დანარჩენ მარშრუტიზატორს. ნახ. 12, ა-ზე ნაჩვენებია ქვექსელი ორი ჯგუფით 1 და 2. ნახ. 12, ბ-ზე ნაჩვენებია მაკავშირებელი ხე ყველაზე მარცხენა მარშრუტიზატორისთვის. როცა პროცესი ჯგუფს უგზავნის მრავალმისამართიან პაკეტს, პირველი მარშრუტიზატორი სწავლობს თავის მაკავშირებელ ხეს და აჭრის მას ხაზებს. ნახ. 12, გ-ზე ნაჩვენებია ჩამოჭრილი მაკავშირებელი ხე პირველი ჯგუფისთვის, დ-ზე კი - მეორე ჯგუფისთვის. მრავალმისამართიანი პაკეტები იგზავნება მხოლოდ მათი ჯგუფების შესაბამისი ჩამოჭრილი მაკავშირებელი ხის გასწვრივ.

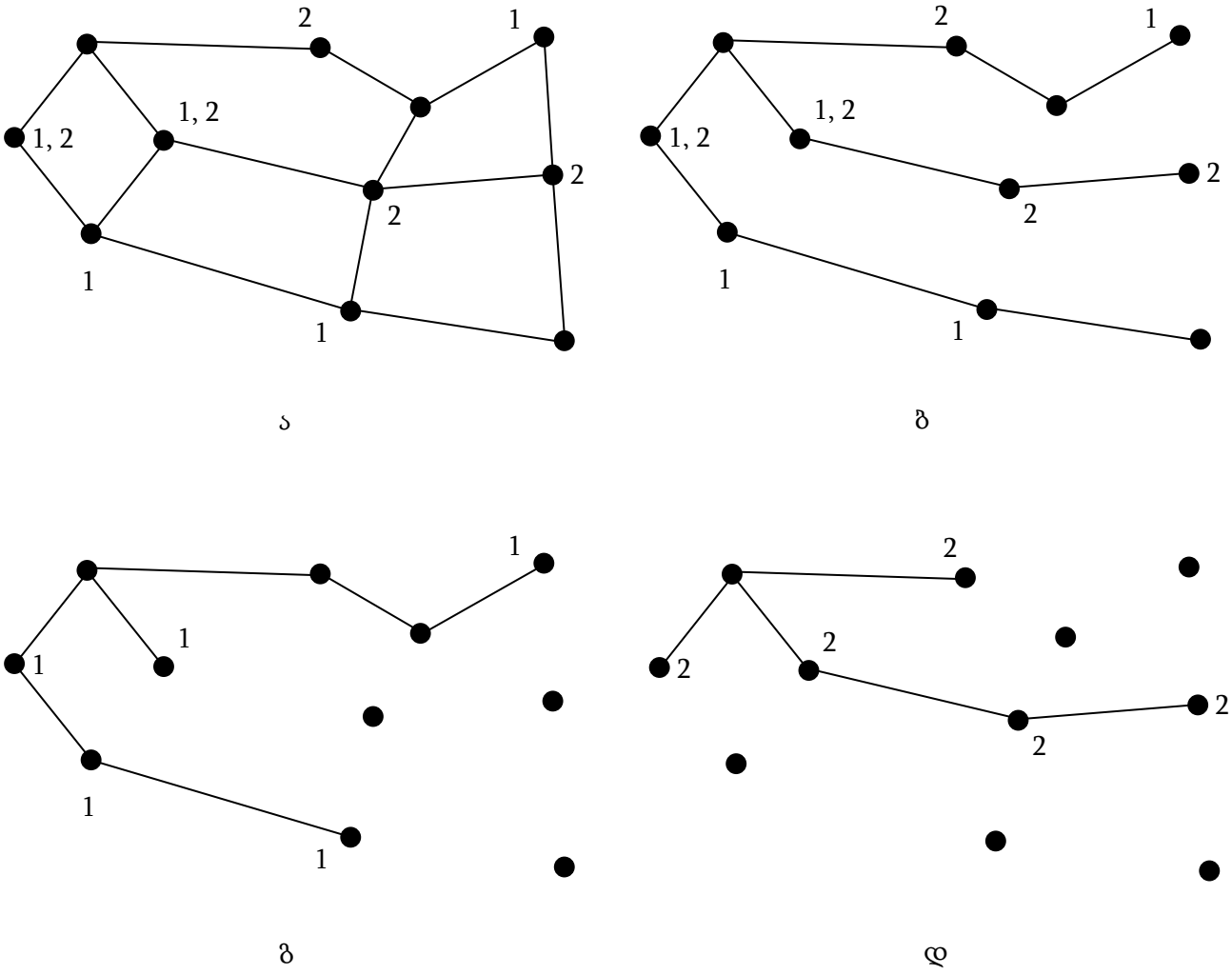
არსებობს მაკავშირებელი ხის ჩამოჭრის რამდენიმე საშუალება. ერთ-ერთი მათგანი გამოიყენება ხაზის მდგომარეობის გათვალისწინებით მარშრუტიზაციისას. ამ დროს, მაკავშირებელი ხიდან შეიძლება წაიშალოს მარშრუტიზატორები, რომლებიც არ ეკუთვნიან მოცემულ ჯგუფს, დაწყებული თითოეული გზის ბოლოდან ხის ძირამდე [12-15].

მდგომარეობების ვექტორების მიხედვით მარშრუტიზაციისას შეიძლება გამოყენებული იყოს ხის ჩამოჭრის სხვა სტრატეგია. მრავალმისამართიანი დაგზავნისათვის აქ გამოიყენება შემხვედრი გზის გასწვრივ წინ წაწევის ალგორითმი. თუ მრავალმისამართიან შეტყობინებას იღებს მარშრუტიზატორი, რომელსაც არ აქვს ჯგუფში შემავალი ჰოსთები და კავშირის ხაზები სხვა მარშრუტიზატორებთან, ის გასცემს PRUNE (ჩამოჭრა) შეტყობინებას. ამით ის გამომგზავნს ატყობინებს, რომ შეტყობინებები მოცემული ჯგუფისთვის მას არ უნდა გამოუგზავნოს. ასეთივე პასუხს გასცემს მარშრუტიზატორი, რომელსაც არ აქვს ჯგუფში შემავალი ჰოსთები, თუ მან

მიიღო მრავალმისამართიანი შეტყობინება ყველა ხაზით. შედეგად, შესრულდება ქვექსელის რეკურსიული შეკვეცა.

მოყვანილი ალგორითმი ცუდად მუშაობს დიდი ქსელებისთვის. თუ ქსელში არის m ჯგუფი, რომელთაგან თითოეული შედგება n კვანძისაგან, მაშინ თითოეული ჯგუფისთვის უნდა ინახებოდეს n ჩამოჭრილი შესასვლელი ხე, ანუ nm ხე მთელი ქსელისთვის. ამიტომ, თუ ჯგუფების რაოდენობა დიდია, მაშინ ყველა ხის შესანახად საჭირო იქნება დიდი მეხსიერება.

არსებობს კიდევ ერთი მეთოდი, რომელიც იყენებს ხეებს ფუძით შუაგულში. ამ მეთოდის თანახმად თითოეული ჯგუფისთვის გამოითვლება ერთიანი მაკავშირებელი ხე ფუძით (ბირთვით) ჯგუფის შუაგულის ახლოს. ჰოსთი უგზავნის მრავალმისამართიან შეტყობინებას ჯგუფის ბირთვს. ბირთვიდან ეს შეტყობინება იგზავნება მთელი მაკავშირებელი ხის გასწვრივ. მართალია, ერთიანი ხე არაა ოპტიმალური ყველა წყაროსთვის, მაგრამ ის ჯგუფისთვის ამცირებს მასში ინფორმაციის დანახარჯებს n-ჯერ.



ნახ. 5.12. ქვექსელი (ა); მაკავშირებელი ხე ყველაზე მარცხენა მარშრუტიზატორისთვის (ბ); მრავალმისამართიანი ხე მეორე ჯგუფისთვის

მარშრუტიზაციის ალგორითმები მობილური ჰოსთებისთვის

ამ განყოფილებაში ვნახავთ, თუ როგორ სრულდება მარშრუტიზაცია იმ შემთხვევაში, როცა მარშრუტიზატორები სტაციონარულია, ჰოსთები კი - მობილური.

სტაციონარული ეწოდება ჰოსთს, რომელიც არასოდეს არ გადაადგილდება. ის ქსელთან შეერთებულია სპილენძის გამტარით ან ოპტიკური კაბელით. **მიგრირებადი** ეწოდება ისეთ სტაციონარულ ჰოსთს, რომელიც დრო და დრო გადაადგილდება ერთი ადგილიდან მეორეზე და ქსელით მხოლოდ მაშინ სარგებლობს, როცა ფიზიკურად არის მასთან მიერთებული. **მობილურია ჰოსთი**, რომელსაც არ აქვს მუდმივი ადგილმდებარეობა და სჭირდება მუდმივი კავშირი ქსელთან სივრცეში გადაადგილების დროს.

ითვლება, რომ ყველა ჰოსთს აქვს სახლის მუდმივი მისამართი, რომელიც გამოიყენება სახლის ადგილმდებარეობის განსაზღვრისთვის. მარშრუტიზაციის მიზანი მობილურ ჰოსთებიან სისტემებში არის პაკეტების გადაცემის უზრუნველყოფა მობილური მომხმარებლებისთვის მათი სახლის მისამართების საშუალებით. აქ, რთული მომენტია მომხმარებლის პოვნა.

ქსელის კუთხით მსოფლიო დაყოფილია უბნებად (ნახ. 13). თითოეული მათგანი წარმოადგენს ლოკალურ ქსელს ან უკაბელო ფიჭას. თითოეულ უბანში მოქმედებს ერთი ან მეტი **გარე აგენტი**, რომლებიც თვალყურს ადევნებენ იმ მობილურ მომხმარებლებს, რომლებიც უბანში შემოვიდნენ. უბანში მოქმედებს, აგრეთვე **შიგა აგენტი**, რომელიც თვალყურს ადევნებს იმ მომხმარებლებს, რომლებმაც უბანი დატოვეს. აგენტი არის მარშრუტრიზატორი, რომელსაც ჩართული აქვს სპეციალური პროტოკოლი მობილური ჰოსთებისთვის.

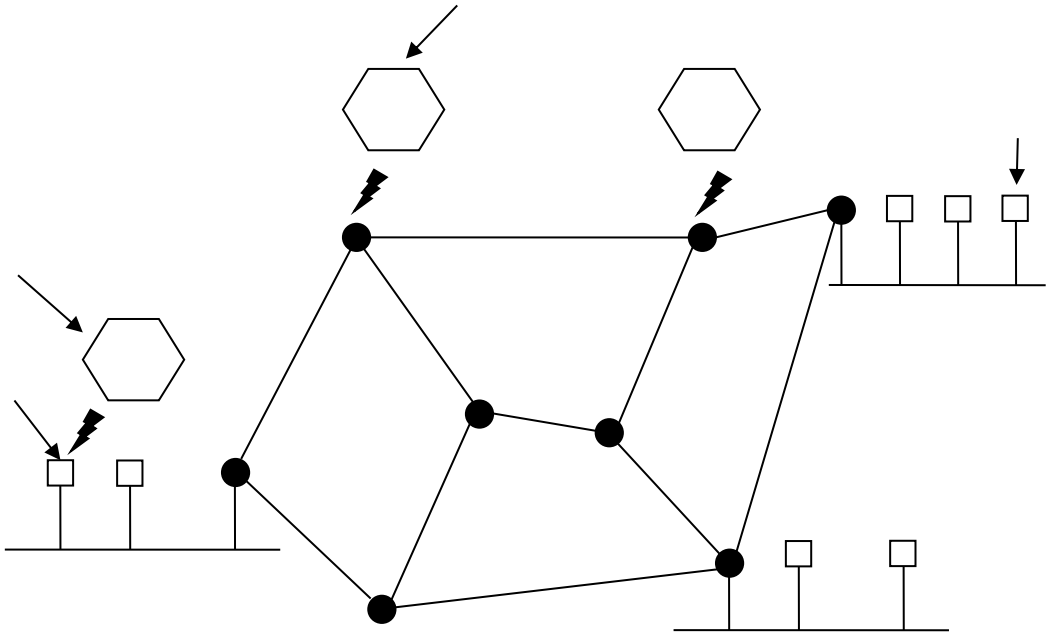
როცა უბანში ახალი მომხმარებელი ჩნდება, ანუ მომხმარებელი, რომელიც უერთდება ქსელს ან გადაადგილებული ფიჭაში, მისი კომპიუტერი უნდა დარეგისტრირდეს მოცემულ უბანში. ამისათვის, ის უნდა დაუკავშირდეს ამ უბნის გარე აგენტს. უბანში რეგისტრირების პროცესი შემდეგი მოქმედებებისაგან შედგება [3,16-18]:

1. პერიოდულად თითოეული გარე აგენტი გზავნის პაკეტს იმისათვის, რომ გამოაცხადოს თავისი არსებობისა და ადგილმდებარეობის შესახებ. უბანში მოსული მობილური ჰოსთი შეიძლება ელოდოს ამ პაკეტის მიღებას, ან თვითონ გაგზავნოს მოთხოვნა ამ უბანში გარე აგენტის საპოვნელად.
2. მოცემულ უბანში რეგისტრირებისას მობილური ჰოსთი ამ უბნის გარე აგენტს გადასცემს შემდეგ ინფორმაციას: თავის საშინაო მისამართს, მონაცემების გადაცემის დონის მიმდინარე მისამართს, ინფორმაციას, რომელიც ადასტურებს მის ნამდვილობას.
3. მოცემული უბნის გარე აგენტი უკავშირდება მობილური მომხმარებლის შიგა აგენტს და ატყობინებს მას იმის შესახებ, რომ მისი (შიგა აგენტის) ერთ-ერთი ჰოსთი იმყოფება გარე აგენტის უბანში. ეს შეტყობინება შეიცავს გარე აგენტის ქსელის მისამართს და ინფორმაციას, რომელიც ადასტურებს მობილური ჰოსთის ნამდვილობას. ეს შიგა აგენტს არწმუნებს იმაში, რომ მობილური ჰოსთი ნამდვილად აქ იმყოფება.
4. შიგა აგენტი ამოწმებს მობილური ჰოსთის მისთვის გადაცემულ უსაფრთხოების იდენტიფიკატორს. ის შეიცავს დროით შტამპს, რომელიც ადასტურებს, რომ იდენტიფიკატორი რამდენიმე წამის წინ შეიქმნა. თუ მობილური ჰოსთის ნამდვილობის შემოწმება წარმატებით ჩაივლის, მაშინ შიგა აგენტი გარე აგენტს კავშირის გაგრძელების ნებას რთავს.

5. ,

,

.



. 5. 13. ,

, ( )

, .

,

. ,

,

,

. 14- , ( )

, ( )

( ) . ,

,

(. 14, I). ( )

,

( ) ( ) ,

.

,

, - (. 14, II

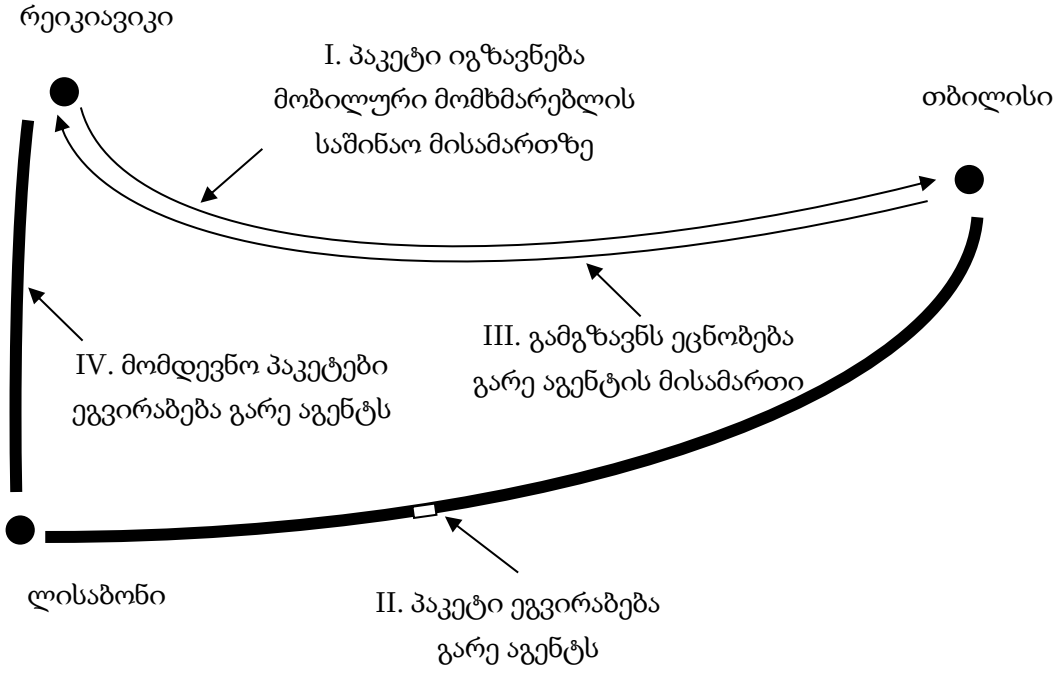
). . .

პაკეტის მიღების შემდეგ, გარე აგენტი მონაცემების ველიდან იღებს ორიგინალ პაკეტს და მას უგზავნის მობილურ ჰოსთს მონაცემების გადაცემის დონის კადრის სახით[19-20].

შემდეგ, შიგა აგენტი გამგზავნს (რეიკიავიკი) ატყობინებს, რომ მან აღარ უნდა გაუგზავნოს პაკეტები მობილურ ჰოსთს სახლის მისამართზე (კიევი), არამედ ისინი უნდა მოათავსოს იმ პაკეტების მონაცემების ველში, რომლებიც გარე აგენტს ეგზავნება (ნახ. 14, III ეტაპი). ამის შემდეგ, მომდევნო პაკეტები პირდაპირ გადაეგზავნება მობილურ მომხმარებელს გარე აგენტის საშუალებით (ნახ. 14, IV ეტაპი).

მობილური ჰოსთებისთვის მარშრუტიზაციის არსებული სქემები ერთმანეთისგან ოთხი ასპექტით განსხვავდებიან:

- განსხვავდებიან იმით, რომ თუ პროტოკოლის რომელი ნაწილი სრულდება მარშრუტიზატორების მიერ, და რომელი - ჰოსთების მიერ.
- ზოგიერთ სქემაში მარშრუტიზატორები იწერენ გარდაქმნილ მისამართებს. შედეგად, მათ შეუძლიათ ხელში ჩაიგდონ და გადაამისამართონ პაკეტები მანამდეც კი, სანამ ისინი მოასწრებენ მობილური მომხმარებლის საშინაო მისამართამდე მოსვლას.
- ზოგიერთ სქემაში თითოეულ მომსვლელს ეძლევა უნიკალური დროებითი მისამართი, ხოლო სხვა სქემაში დროებითი მისამართი მიმართავს აგენტს, რომელიც ამუშავებს ტრაფიკს ყველა მომსვლელისათვის.
- სქემები განსხვავდებიან პაკეტების გადამისამართების საშუალებებით. ერთი საშუალებაა პაკეტში მიმღების მისამართის შეცვლა და ისე გადაცემა. მეორეა, მთელი პაკეტის მოთავსება სხვა პაკეტის შიგნით, რომელიც იგზავნება დროებითი მისამართით.



ნახ. 5. 14. პაკეტების მარშრუტიზაცია მობილური ჰოსთებისკენ

ლიტერატურა

1. Сетевые операционные системы. / В.Г. Олифер, Н. А. Олифер. – СПб.: Питер, 2002. – 544 с.: ил.
2. Сети Net Ware 5. Руководство от Novell. Джефри Ф. Хьюз, Блейер В. Томас. – Вильямс, 2000.
3. Компьютерные сети. 4-е изд. / Э. Таненбаум. - СПб.: Питер, 2003. – 992 с.: ил. – (Серия «Классика computer science»).
4. Столлингс В. Беспроводные линии связи и сети.: Пер. с англ. – М.: Издательский
5. дом «Вильямс», 2003. – 640 с.
6. Компьютерные сети. Первый шаг. : Пер. с англ. — М. : Издательский дом "Вильяме", 2006. — 432 с.: ил. — Парал. тит. англ.
7. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 3-е изд. – Спб.: Питер, 2006. – 958 с.
8. Tanenbaum, Andrew S., Computer networks / Andrew S. Tanenbaum, David J. Wetherall. -- 5th ed. 2011. p.962. CD-5640
9. Peter L Dordal. An Introduction to Computer Networks. 2019. P. 882. CD-5640
10. Taha Hamdy A. Operations research. -8th edition. 2007. P. 838. CD-5640
11. Diarmuid O'Briain. Open Networks. 2015. P. 276. CD-5640
12. Nerijus Paulauskas, Eimantas Garsva. Computer Networks. 2012. P. 167. CD-5640
13. I. Marsic. Compter Networks. 2013. P. 595. CD-5640
14. Максим М. Безопасность беспроводных сетей / Мерит Максим, Дэвид Полино; Пер. с англ. Семенова А.В. – М.: Компания АйТи; ДМК Пресс, 2004.- 288с.
15. Вентцель Е.С. Исследование операций. М. «Советское радио», 1972, 552 с.,
16. Акулич И.Л. Математическое программирование в примерах и задачах: Учеб. пособие для студентов эеоном. спец. вузов. – М.: Высш. шк., 1986. – 319 с., ил.
17. Принципы маршрутизаци в Internet, 2-е изд.: Пер. с англ. М.: Издательский дом «Вильямс», 2001. - 448 с. : ил.
18. Х. Таха. Введение в исследовании операций: В 2-х книгах. Кн. 1, Пер. с англ. – М.: Мир, 1985. – 479 с., ил.
19. Frederick S. Hillier, Gerald J. Lieberman. Introduction to operations research. 7th ed. 2001. P. 1237. CD-5640
20. O. Bonaventure. Compter Networks. 2019. P. 272. CD-5640

კომპიუტერული უზრუნველყოფა ლ. გაჩეჩილაძის

იბეჭდება ავტორთა მიერ წარმოდგენილი სახით

გადაეცა წარმოებას 30.03.2021. ხელმოწერილია დასაბეჭდად
6.04.2021. ბეჭდვა ოფსეტური. პირობითი ნაბეჭდი თაბახი 2,3.
ტირაჟი 50 ეგზ.

Verba volant,
scripta manent

სტუ-ს "IT-კონსალტინგის სამეცნიერო ცენტრი", თბილისი, კოსტავას 77