საქართველოს უნივერსიტეტი

დასახელება: კომპიუტერული ქსელის დაპროექტება ლოგიკურ-ალბათური მეთოდების გამოყენებით

ავტორი: სულხან სულხანიშვილი

სადისერტაციო ნაშრომი შესრულებულია
საქართველოს უნივერსიტეტის
ინფორმატიკის, ინჟინერიისა და მათემატიკის სკოლის
დოქტორის აკადემიური ხარისხის მოსაპოვებლად

ხელმძღვანელი: სერგო ცირამუა

თბილისი

2015

საავტორო უფლებები

სადისერტაციო ნაშრომი: „კომპიუტერული ქსელის დაგეგმარება ლოგიკურ-ალბათური მეთოდების გამოყენებით“ .

ინფორმაციული ტექნოლოგიების დოქტორის ხარისხის მოსაპოვებლად.

ავტორი - სულხან სულხანიშვილი

სულხან სულხანიშვილი © თბილისი, 2015 წლის 15 ივლისი

მიმოხილვა

თანამედროვე პირობებში კომპიუტერული ქსელი ნებისმიერი ორგანიზაციის აუცილებელი კომპონენტია და მის გამართულად მუშაობაზე მნიშვნელოვნად არის დამოკიდებული მცირე, საშუალო თუ დიდი მასშტაბის ორგანიზაციების, მათ შორის სასწავლო დაწესებულებებისა და უნივერსიტეტების წარმატებული ფუნქციონირება. გლობალიზაციის პრიობებში ინფორმაციის გავრცელება და მისი მოპოვება დამოკიდებულია კომპიუტერულ ქსელებში ნებისმიერი სახის ინფორმაციის გადაცემის სისწრაფეზე, საიმედოობაზე და სერვისის ხარისხზე. კომპიუტერული ქსელების ფუნქციონირებისას უდიდესი მნიშვნელობა ენიჭება ინფორმაციის დაცულობას და უსაფრთხოებას.

ქსელური ტექნოლოგიების სწრაფი განვითარების პირობებში მნიშვნელოვანია ამ ტექნოლოგიების როგორც თეორიული, ასევე პრაქტიკული საკითხების გადაწყვეტა. ხშირ შემთხვევაში ორგანიზაციის განვითარებასთან, გაფართოებასთან და გაზრდასთან ერთად დინამიურად ვითარდება და იზრდება კომპიუტერული ქსელიც. კომპიუტერული ქსელი დინამიურია, იგი განვითარებადია დროსა და სივრცეში. დროში განვითარება გულისხმობს ახალი ქსელური ტექნოლოგიების დანერგვას, სივრცეში განვითარება კი - კომპიუტერული ქსელის ტერიტორიულ გაფართოებას და შესაბამისად მასში შემავალი კომპონენტების რაოდენობის გაზრდას. ასეთ ვითარებაში აქტუალურია კომპიუტერული ქსელის დაგეგმარება განხორციელდეს მისი ეფექტიანობის, საიმედოობისა და უსაფრთხოების გათვალისწინებით, რისი უზრუნველყოფაც შესაძლებელია მათემატიკური მოდელირების, კერძოდ კი ლოგიკურ-ალბათური მოდელირებისა და ზუსტი საინჟინრო გამოთვლების საფუძველზე. შესაბამისად ნაშრომში შემოთავაზებულია კომპიუტერული ქსელის განვითარების დინამიკის ანალიზი საქართველოს უნივერსიტეტის მაგალითზე და მაღალი ეფექტიანობის (სწრაფქმედების, საიმედოობის, უსაფრთხოების, სერვისის ხარისხის) მათემატიკური მოდელირებისა და საინჟინრო გამოთვლებზე დაფუძნებული საუნივერსიტეტო კომპიტერული ქსელის დაპროექტების საკითხები, რაც მეტად აქტუალურია თანამედროვე ქსელურ ტექნოლოგიებში.

ნაშრომში ჩატარებული კვლევის მიზანია კომპიუტერული ქსელის დაპროექტების მათემატიკურ მოდელირებაზე დაფუძნებული მეთოდოლოგიის შემუშავება, ქსელური ეფექტიანობის პარამეტრების შეფასების ლოგიკურ-ალბათური მოდელირება, შეფასება და ანალიზი.

დასახული მიზნიდან გამომდინარე შესრულდა შემდეგი ამოცანები: კომპიუტერული ქსელების ტოპოლოგიებისა და პროტოკოლების განხილვა, შედარებითი ანალიზის ჩატარება; საქართველოს უნივერსიტეტის კომპიუტერული ქსელის განვითარების დინამიკის განხილვა და ანალიზი; კომპიუტერული ქსელის ეფექტიანობის (საიმედოობის, უსაფრთხოების, სერვისის ხარისხის) მაჩვენებლების განსაზღვრა; კომპიუტერული ქსელის საიმედოობის ლოგიკურ-ალბათური მოდელირება, საიმედოობის რაოდენობრივი შეფასება ლოგიკურ-ალბათური მეთოდების გამოყენებით და ქსელის ოპტიმიზაცია.

ნაშრომის სამეცნიერო სიახლეს წარმოადგენს დინამიურად განვითარებადი კომპიუტერული ქსელის შედარებითი დროითი და სივრცითი ანალიზის მეთოდოლოგიის შემუშავება, ასევე კომპიუტერული ქსელის დაპროექტების ლოგიკურ-ალბათური მოდელების შემუშავება, ქსელის უსაფრთხოების (ინფორმაციის დაცულობისა და არასანქცირებული შეღწევის) სიტუაციური სცენარის შემუშავება და რაოდენობრივი შეფასების მეთოდოლოგიის შემუშავება.

კვლევის ობიექტს წარმოადგენს საქართველოს უნივერსიტეტის დინამიურად განვითარებადი კომპიუტერული ქსელი, მისი ეფექტიანობის შეფასების პარამეტრების ანალიზი დროსა და სივრცეში.

სარჩევი

მიმოხილვა 3
სარჩევი 5
ილუსტრაციების, ცხრილების და გრაფიკების ჩამონათვალი 6
გამოყენებული ტერმინოლოგია 6
შესავალი 11
თემის აქტუალურობა 11
ნაშრომის მიზანი 12
ამოცანები 12
თავი 1. კომპიუტერული ქსელების მიმოხილვა და მათი ტოპოლოგია 14
კომპიუტერული ქსელების მიმოხილვა 14
1.2. კომპიუტერული ქსელების ტოპოლოგია 17
1.3.კომპიუტერების დამისამართება ქსელში და ქსელის სტრუქტურიზაცია 23
1.4. ინფორმაციის გადაცემის გარემო 28
თავი 2. ღია სისტემების ურთიერთქმედების მოდელები (OSI) 37
2.1. OSI მოდელის ურთიერთქმედების დონეები 37
2.2. ქსელზე დამოკიდებული და ქსელისგან დამოუკიდებელი დონეები 40
2.3. პროტოკოლები 44
თავი 3. საქართველოს უნივერსიტეტის კომპიუტერული ქსელის განვითარების დინამიკა 55
თავი 4. კომპიუტერული ქსელის სტრუქტურის საიმედოობისა და უსაფრთხოების ლოგიკურ-ალბათური მოდელირება 70
4.1. სისტემის ქმედითუანრიანობის პირობის აღწერა ლოგიკურ-ალბათური ფუნქციების საშუალებით 70
4.2 სისტემის უმტყუნო მუშაობის ალბათობის გამოთვლა კვეთის ალგორითმის გამოყენებით. 84
4.3. სისტემის უმტყუნო მუშაობის ალბათობის გამოთვლის ცხრილური მეთოდი. 89
4.4. ორთოგონალიზაციის მეთოდით სისტემის უმტყუნო მუშაობის ალბათობის რაოდენობრივი შეფასება 94
4.5. ცალკეული ელემენტის როლი სისტემის საიმედოობაში 101
4.5.1. სისტემაში ელემენტის "წონის" რაოდენობრივი შეფასება 102
4.5.2. სისტემაში ელემენტის "მნიშვნელობის" შეფასება 106
4.6. საბაზო სტრუქტურისა და რეზერვირებული სისტემის უმტყუნო მუშაობის ალბათობის შედარებითი ანალიზი 110
მიღებული შედეგები 115
გამოყენებული ლიტერატურა 116

ილუსტრაციების, ცხრილების და გრაფიკების ჩამონათვალი

ილუსტრაცია N 1 - ერთრანგიანი ქსელი და ქსელი სერვერის ბაზაზე ---------------------------- 16
ილუსტრაცია N 2 - სალტე -- 19
ილუსტრაცია N 3 - ვარსკვლავი --- 19
ილუსტრაცია N 4 - წრე -- 19
ილუსტრაცია N 5 - ქსელიდან დაზიანების შემთხვევაში როგორ არ წყდება სხვა ომპიუტერების მუშაობა --- 21
ილუსტრაცია N 6 - ვარსკვლავი-სალტე ტოპოლოგია (star-bus) ------------------------------------ 22
ილუსტრაცია N 7 - კომპიუტერული ქსელი გამმეორებელის გამოყენებით -------------------- 26
ილუსტრაცია N 8 - ქსელის ლოგიკური სტრუქტურიზაცია ხიდის საშუალებით ----------- 27
ილუსტრაცია N 9 - კოაქსალური გამტარი --- 29
ილუსტრაცია N 10 - ხვეული წყვილი -- 29
ილუსტრაცია N 11 - ოპტიკური კაბელი -- 30
ილუსტრაცია N 12 - კონექტორი RG-45 --- 31
ილუსტრაცია N 13 - არამოდულირებული გადაცემა (ციფრული სიგნალი) ------------------- 33
ილუსტრაცია N 14 - IEEE 802.X სტანდარტის სტრუქტურა --- 49
ილუსტრაცია N 15 - OSI-ს სხვადასხვა დონეებზე გამოყენებული პროტოკოლები -------- 51
ილუსტრაცია N 16 - პაკეტის სქემა -- 54
ილუსტრაცია N 17 - საქართველოს უნივერსიტეტის კომპიუტერული ქსელის 2005 წლის ტოპოლოგია --- 56
ილუსტრაცია N 18 - Domain Forest სტრუქტურა -- 57
ილუსტრაცია N 19 - კომპიუტერული ქსელის სტრუქტურა --- 58
ილუსტრაცია N 20 - 2007 წლის სტრუქტურა --- 59
ილუსტრაცია N 21 - საქართველოს უნივერსიტეტის კომპიუტერული ქსელი სტრუქტურა - --- 60
ილუსტრაცია N 22 - მაღალი დონის დიზაინი საწარმოთა ქსელიში ------------------------------ 61
ილუსტრაცია N 23 - მაღალი დონის კამპუსის ქსელის იერარქიული მოდელის მაგალითი - --- 63
ილუსტრაცია N 24 - OSI მოდელის ურთიერთობა პროტოკოლებთან და ქსელთან ---------- 64
ილუსტრაცია N 25 - Layer 2 switching -- 65
ილუსტრაცია N 26 - Layer 3 Switching -- 65
ილუსტრაცია N 27 - პაკეტების გადაწერის შესაძლებლობა Layer 3 მარშრუტიზაციით, სადაც ორი კომპიუტერი უკავშირდება ერთმანეთს ICMP -- 67
ილუსტრაცია N 28 - კლიენტი/სერვერის ტრაფიკის ნაკადი --------------------------------------- 68
ილუსტრაცია N 29 - კომპიუტერული ქსელის სტრუქტურული სქემა ------------------------- 83
ილუსტრაცია N 30 - სტრუქტურული სქემა სარეზერვო და არხებით.

ცხრილი N 1 - ტოპოლოგიის ცხრილი --- 23
ცხრილი N 2 - გადამცემ და მიმღებ კომპიუტერებს შორის OSI -ის მოდელების ურთიერთქმედება --- 39
ცხრილი N 3 - ქსელისგან დამოუკიდებელი დონეები --- 43
ცხრილი N 4 - OSI მოდელს თავისი თვისებებით -- 66
ცხრილ N 5 - სისტემის უმტყუნო მუშაობის ალბათობები მნიშვნელობები, სისტემაში შემავალი ელემენტების საიმედოობის სხვადასხვა მნიშვნელობებისათვის ----------------- 98
ცხრილი N 6 - სისტემის ცალკეული ელემენტების „წონა“ -- 104

ცხრილი N 7 - სისტემის უმტყუნო მუშაობის (საიმედოობის) ალბათობის რიცხვითი მნიშვნელობების საფუძველზე აგებულია უმტყუნო მუშაობის ალბათობის გრაფიკი სისტემის ორივე ვარიანტისათვის --- 112

გრაფიკი N 1 - სისტემის უმტყუნო მუშაობის ალბათობის გრაფიკი ---------------------------- 98
გრაფიკი N 2 - სისტემების უმტყუნო მუშაობის ალბათობის გრაფიკი 1-საბაზო ვარიანტი, 2 - რეზერვირებით --- 115
გრაფიკი N 3 - ელემენტების "მნიშვნელობის" გარფიკი სისტემის სტრუქტურლ საიმედოობაში --- 107

გამოყენებული ტერმინოლოგია

API - (Application Program Interface) გამოყენებითი პროგრამული ინტერფეისი - ესაა შეთანხმებების ნაკრები, რომელიც უზრუნველყოფს ფუნქციის გამოძახების და პარამეტრების გადაცემის წესს გამოყენებითი პროგრამიდან.

ავტონომიური კომპიუტერი - კომპიუტერი რომელიც არ არის ჩართული ქსელში.

ანალოგური ხაზი - კავშირის ხაზი, რომელიც სიგნალებს გადასცემს ანალოგური ფორმით, მაგალითად, სატელეფონო ხაზი.

არამოდულირებული გადაცემა - მონაცემთა გადაცემის ხერხია, რომლის დროსაც მონაცემთა ყოველი ბიტი კოდირდება ცალკეული ელექტრული ან სინათლის იმპულსების სახით.

არხი – მონაცემთა ან სიგნალის გადაცემის გარემო ან გზაა.

აპარატურული უზრუნველყოფა - კომპიუტერის ფიზიკური კომპონენტები ე.ი ის რასაც შეიძლება ხელით შევეხოთ.

მეხსიერების საბაზო მისამართი - ოპერატიული მეხსიერების პარამეტრია, რომელიც განსაზღვრავს რომელიმე მოწყობილობისათვი მეხსიერების ფრაგმენტის დასაწყისს.

ბაიტი - ინფორმაციის ერთეულია, რომელიც შეესაბამება ერთ სიმბოლოს მაგალითად ასოს, ციფრს ან პუნკტუაციის ნიშანს, შედგება 8 ბიტისაგან. კომპიუტერის მეხსიერება იზომება კილობაიტებში -210 ბაიტი, მეგაბაიტებში - 220 ბაიტი, გიგაბაიტებში - 230 ბაიტი, პეტაბაიტი 250 ბაიტი ან ეკსაბაიტი 260 ბაიტი.

ბიტ/წმ - მონაცემთა გადაცემის სიჩქარე.

ბუფერი - მონაცემთა გადაცემის ან მიღების დროს დროებითი შესანახი ოპერატიული დამხსომებელი მოწყობილობაა (RAM).

გარსი – პროგრამული უზრუნველყოფაა, რომელიც უზრუნველყოფს მომხმარებლის ურთიერთობას ოპერაციულ სისტემასთან (მომხმარებლის ინტერფეისი).

განყოფილება – არის ფიზიკური დისკის ნაწილი, რომელიც ოპერაციული სისტემის მიერ აღიქმება, როგორც დამოუკიდებელი ლოგიკური მოწყობილობა.

გამმეორებელი – რეპეატერ- მოწყობილობა, რომელიც გამოიყენება სიგნალების აღდგენისათვის და კავშირის საერთო ხაზის სიგრძის გასაზრდელად. მუშაობს OSI - მოდელის ფიზიკურ დონეზე.

ერთრანგიანი ქსელი – ქსელი, სადაც არ არის გამოყოფილი სერვერი ან კომპიუტერებს შორის იერარქია. ყველა კომპიუტერი თანაბარუფლებიანია. ყოველი კომპიუტერი სერვერიც არის და კლიენტიც.

ეკრანი – მეტალური წნულია ან ცილინდრი, რომელიც იცავს გადასაცემ მონაცემებს, ამცირებს გარე ელექტრულ შეფერხებებს,(რომელსაც ხმაურს უწოდებენ)

ეკრანირებული ხვეულა წყვილი – ხვეულა წყვილი რომელიც დაფარულია მეტალური ფოლგით, რომელიც ეკრანის როლს ასრულებს,

WAN- Wide Area Network ფართო არის გამოთვლითი ქსელია, აერთიანებს ერთმანეთისაგან შორი მანძილით დაშორებულ კომპიუტერებს.

დრაივერი - არის პროგრამა, რომელიც საშუალებას აძლევს კომპიუტერს დაუკავშირდეს მოწყობილობას.

კადრი – OSI მოდელში არხის დონის ინფორმაციის ბლოკია.

კეში – cache – მეხსიერების სახეა, მასში თავსდება ხშირად გამოყენებადი მონაცემების კოპიო (ასლი). ის ძალიან სწრაფქმედია, მასთან ურთიერთობს პროცესორი.

კომპიუტერი დისკის გარეშე – კომპიუტერი მოქნილი და ხისტი დისკების გარეშე, იყენებს სპეციალურ მუდმივ დამხსომებელ მოწყობილობას (მდმ), რომლის პროგრამა უკეთებს რეგისტრაციას ქსელურ კლიენტს.

LAN – ლოკალური გამოთვლითი ქსელია, კომპიუტერები შეერთებულნი არიან ერთმანეთთან შემოსაზღვრულ არეზე.

მაგისტრალი - backbone ძირითადი კაბელია, რომლიდანაც კაბელები მოდის კომპიუტერებთან, გამმეორებლებთან და ხიდებთან.

მისამართი – უნიკალური იდენტიფიკატორია, რომელიც ენიჭება ქსელს ან ქსელურ მოწყობილობას, იმისათვის რომ სხვა ქსელებმა ან მოწყობილობებმა შეძლონ მისი გამოცნობა ინფორმაციის გაცვლის დროს.

მარშრუტიზატორი - router მოწყობილობაა, რომელიც აერთებს სხვადასხვა ტიპის ქსელებს, რომლებიც იყენებენ სხვადასხვა პროტოკოლებს და არქიტექტურას. ისინი მუშაობენ OSI -ს მოდელის ქსელურ დონეზე. ისინი მონაცემთა გადასაცემად ირჩევენ საუკეთესო გზას.

ოპერატიული მეხსიერება-RAM - ენერგოდამოკიდებული მეხსიერებაა, რომელიც გამოიყენება პროცესორის მიერ ჩასაწერად ან წასაკითხად.

ოპტიკური ბოჭკო - არის გარემო, რომელშიც სიგნალები გადაიცემა მოდულირებული სინათლის იმპულსების სახით.

პაკეტი - OSI -ს მოდელის ქსელური დონის ინფორმაციის ბლოკია, რომელიც გადაიცემა ქსელში.

პროგრამული უზრუნველყოფა – პროგრამების ნაკრებია, რომელიც მართავს კომპიუტერის აპარატურის მუშაობას.

პროტოკოლი - ესაა შეთანხმება ქსელში ჩართულ ორ კომპიუტერს შორის .

სამუშაო ჯგუფი – Workgroup კომპიუტერების ნაკრებია, რომლებიც გაერთიანებულნი არიან ერთ ლოკალურ ქსელში და შეთავსებით იყენებენ რესურსებს, როგორიცაა მონაცემები და პერიფერიული მოწყობილობები.

სექტორი – დისკური სივრცის ფრაგმენტია. დისკი იყოფა ზედა და ქვედა მხარედ. ბილიკი ეს არის კონცენტრირებული წრე ყოველ მხარეზე, სექტორი კი არის ამ წრის ნაწილი. დისკზე ფიზიკური მეხსიერების უმცირესი ელემენტია მას აქვს ფიქსირებული ზომა, ჩვეულებრივ მონაცემთა 512 ბაიტი.

სააღრიცხვო ინფორმაცია - ესაა ინფორმაცია ქსელური მომხმარებლის შესახებ: მისი სახელი, რეგისტრაციის პაროლი ქსელში შესასვლელად, ჯგუფი რომელსაც მიეკუთვნება ეს სააღრიცხვო ინფორმაცია, რესურსებთან დაკავშირების უფლება დასისტემასთან მუშაობის პრივილეგია.

ტოპოლოგია – კომპიუტერების და სხვა ქსელური კომპონენტების ქსელში შეერთების სქემა.

ქსელი სერვერის ბაზაზე - ქსელი, რომელშიც კომპიუტერები დიფერენცირებულია სერვერად და კლიენტად

ხიდი - Bridge - საშუალებას აძლევს სადგურებს დაუკავშირდეს სხვა ქსელის რესურსებს. შეიძლება გამოყენებული სიგრძის ან კვანძების რაოდენობის გასაზრდელად. ისინი მუშაობენ OSI -ს მოდელის არხის დონეზე.

წყვეტა - არის სიგნალი, რომელიც გაეგზავნება ცენტრალურ პროცესორს მოწყობილობიდან და საჭიროებს მის ჩარევას შექმნილ სიტუაციაში.

WAN – კომპიუტერული ქსელია, რომელიც აერთიანებს Aშორ მანძილზე დაშორებულ კომპიუტერებს.

შესავალი

თემის აქტუალურობა

კომპიუტერულ ქსელებს უმნიშვნელოვანესი როლი უჭირავს ინფორმაციულ-საკომუნიკაციო ტექნოლოგიების სფეროში. კომპიუტერული ქსელი ზოგადად ნებისმიერი თანამედროვე, თუნდაც მცირე მასშტაბის ორგანიზაციის აუცილებელი კომპონენტია. გლობალიზაციისა პრიობებში ინფორმაციის გავრცელება და მისი მოპოვება დამოკიდებულია კომპიუტერულ ქსელებში ნებისმიერი სახის ინფორმაციის გადაცემის სისწრაფეზე, საიმედოობაზე და სერვისის ხარისხზე. კომპიუტერული ქსელის გამართულად მუშაობაზე მნიშვნელოვნად არის დამოკიდებული ნებისმიერი მცირე, საშუალო თუ დიდი მასშტაბის ორგანიზაციების, მათ შორის სასწავლო დაწესებულებებისა და უნივერსიტეტების წარმატებული ფუნქციონირება. ასევე კომპიუტერული ქსელების ფუნქციონირებისას უდიდესი მნიშვნელობა ენიჭება ინფორმაციის დაცულობას და უსაფრთხოებას.

ქსელური ტექნოლოგიების სწრაფი განვითარების პირობებში მნიშვნელოვანია ამ ტექნოლოგიების როგორც თეორიული, ასევე პრაქტიკული საკითხების გადაწყვეტა. ხშირ შემთხვევაში ორგანიზაციის განვითარებასთან, გაფართოებასთან და გაზრდასთან ერთად დინამიურად ვითარდება და იზრდება კომპიუტერული ქსელიც. კომპიუტერული ქსელი დინამიურია, იგი განვითარებადია დროსა და სივრცეში. დროში განვითარება გულისხმობს ახალი ქსელური ტექნოლოგიების დანერგვას, სივრცეში განვითარება კი - კომპიუტერული ქსელის ტერიტორიულ გაფართოებას და შესაბამისად მასში შემავალი კომპონენტების რაოდენობის გაზრდას. ასეთ ვითარებაში აქტუალურია კომპიუტერული ქსელის დაგეგმარება განხორციელდეს მისი ეფექტიანობის, საიმედოობისა და უსაფრთხოების გათვალისწინებით, რისი უზრუნველყოფაც შესაძლებელია მათემატიკური მოდელირებისა და ზუსტი საინჟინრო გამოთვლების საფუძველზე. შესაბამისად ნაშრომში შემოთავაზებულია კომპიუტერული ქსელის განვითარების დინამიკის ანალიზი საქართველოს უნივერსიტეტის მაგალითზე და მაღალი ეფექტიანობის (სწრაფქმედების, საიმედოობის, უსაფრთხოების, სერვისის ხარისხის) მათემატიკური მოდელირებისა და საინჟინრო გამოთვლებზე დაფუძნებული საუნივერსიტეტო კომპიტერული ქსელის დაპროექტების საკითხები, რაც მეტად აქტუალურია თანამედროვე ქსელური ტექნოლოგიებში.

ნაშრომის მიზანი

ნაშრომში ჩატარებული კვლევის მიზანია კომპიუტერული ქსელის დაპროექტების მათემატიკურ მოდელირებაზე დაფუძნებული მეთოდოლოგიის შემუშავება, ქსელის სტრუქტურული საიმედოობის შეფასების ლოგიკურ-ალბათური მოდელირება და შეფასება.

ამოცანები

1. კომპიუტერული ქსელების მიმოხილვა, ტოპოლოგიებისა და პროტოკოლების განხილვა, შედარებითი ანალიზის ჩატარება
2. საქართველოს უნივერსიტეტის კომპიუტერული ქსელის განვითარების დინამიკის განხილვა და ანალიზი
3. კომპიუტერული ქსელის ეფექტიანობის (საიმედოობის, უსაფრთხოების, სერვისის ხარისხის) მაჩვენებლების განსაზღვრა
4. კომპიუტერული ქსელის სტრუქტურული საიმედოობის რაოდენობრივი შეფასების მეთოდოლოგიის დამუშავება ლოგიკურ-ალბათური მეთოდების ბაზაზე
5. კომპიუტერული ქსელის სტრუქტურული საიმედოობის ლოგიკურ-ალბათური მოდელირება
6. კომპიუტერული ქსელის სტრუქტურული საიმედოობის რაოდენობრივი შეფასება ლოგიკურ-ალბათური მეთოდების გამოყენებით და ქსელის ოპტიმიზაცია.

ნაშრომის სამეცნიერო სიახლეს წარმოადგენს დინამიურად განვითარებადი კომპიუტერული ქსელის შედარებითი დროითი და სივრცითი ანალიზის მეთოდოლოგიის შემუშავება, ასევე კომპიუტერული ქსელის დაპროექტების ლოგიკურ-ალბათური მოდელების შემუშავება და რაოდენობრივი შეფასების მეთოდოლოგიის შემუშავება.
კვლევის ობიექტს წარმოადგენს საქართველოს უნივერსიტეტის დინამიურად განვითარებადი კომპიუტერული ქსელი, მისი ეფექტიანობის პარამეტრების ანალიზი დროსა და სივრცეში.

სადისერტაციო ნაშრომის სტრუქტურა

დისერტაცია შედგება შესავლისაგან, ოთხი თავისაგან, დასკვნების, ლიტერატურის ჩამონათვალისაგან და დანართისგან.

შესავალი - აქტუალობა, მიზანი, ამოცანები

თავი 1. კომპიუტერული ქსელების მიმოხილვა და მათი ტოპოლოგია

1.1. კომპიუტერული ქსელების მიმოხილვა

1.2. კომპიუტერული ქსელების ტოპოლოგია

1.3. კომპიუტერების დამისამართება ქსელში და ქსელის სტრუქტურიზაცია

1.4. ინფორმაციის გადაცემის გარემო

თავი 2. ღია სისტემების ურთიერთქმედების მოდელები (OSI)

2.1. OSI მოდელის ურთიერთქმედების დონეები
2.2. ქსელზე დამოკიდებული და ქსელისგან დამოუკიდებელი დონეები
2.3. პროტოკოლები

თავი 3. საქართველოს უნივერსიტეტის კომპიუტერული ქსელის განვითარების დინამიკა

თავი 4. კომპიუტერული ქსელის სტრუქტურის საიმედოობისა და უსაფრთხოების ლოგიკურ-ალბათური მოდელირება

4.1. სისტემის ქმედითუანრიანობის პირობის აღწერა ლოგიკურ-ალბათური ფუნქციების საშუალებით

4.2 სისტემის უმტყუნო მუშაობის ალბათობის გამოთვლა კვეთის ალგორითმის გამოყენებით.

4.3. სისტემის უმტყუნო მუშაობის ალბათობის გამოთვლის ცხრილური მეთოდი.

4.4. სისტემის უმტყუნო მუშაობის ალბათობის გამოთვლა ორთოგონალიზაციის ალგორითმის გამოყენებით

4.5. ორთოგონალიზაციის მეთოდით სისტემის უმტყუნო მუშაობის ალბათობის რაოდენობრივი შეფასება

4.6. ცალკეული ელემენტის როლი სისტემის საიმედოობაში

4.6.1. სისტემაში ელემენტის "წონის" რაოდენობრივი შეფასება

4.6.2. სისტემაში ელემენტის "მნიშვნელობის" შეფასება

4.6.3 სისტემაში ელემენტის "წვლილის" შეფასება

4.7. საბაზო სტრუქტურისა და რეზერვირებული სისტემის უმტყუნო მუშაობის ალბათობის შედარებითი ანალიზი

თავი 1. კომპიუტერული ქსელების მიმოხილვა და მათი ტოპოლოგია

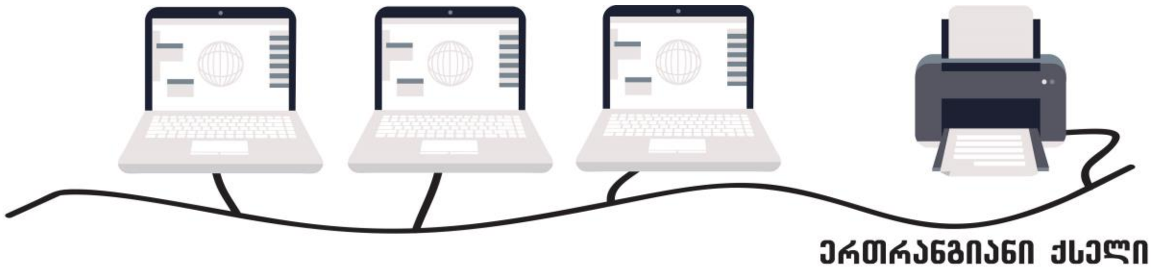
კომპიუტერული ქსელების მიმოხილვა

კომპიუტერული ქსელები არის კომპიუტერების და პერიფერიული მოწყობილობების ერთიანობა, რომლებსაც სპეციალური საკომუტაციო საშუალებების და პროგრამული უზრუნველყოფის საშუალებით შეუძლიათ ინფორმაციის გაცვლა.
კომპიუტერულ ქსელში კომპიუტერების რაოდენობა ორიდან რამდენიმე ათასამდე შეიძლება იცვლებოდეს. კომპიუტერების რაოდენობის და ქსელის ზომის მიხედვით არსებობს ლოკალური (LAN) და ფართო არის ქსელი (WAN).
ლოკალურ ქსელში კომპიუტერების და პერიფერიული მოწყობილობების რაოდენობა შეზღუდულია. ისინი განლაგებულნი არიან შემოსაზღვრულ არეზე.
ფართო არის ქსელში კომპიუტერების რაოდენობა რამდენიმე ათასამდე შეიძლება იცვლებოდეს. ისინი სხვადასხვა ქალაქებსა და სახელმწიფოებშიც კი შეიძლება იყვნენ განლაგებული. (ყველაზე გავრცელებული ფართო არის ქსელია ინტერნეტი). ფართო არის ქსელები აგებულნი არიან სატელეფონო ხაზების გამოყენებით.
კომპიუტერული ქსელების ძირითადი დანიშნულებაა ქსელში ჩართული ყველა კომპიუტერისათვის რესურსების შეთავსებით გამოყენება და მუდმივი კავშირი რეალურ დროში. რესურსები ესაა მონაცემები, დანართები და პერიფერიული მოწყობილობები.
ქსელები მათში გამოყენებული მოწყობილობების მიხედვით არსებობენ: ერთრანგიანი ქსელები, ქსელები სერვერის ბაზაზე და კომბინირებული ქსელები. მათ შორის პრინციპიალური განსხვავება არსებობს რადგან მათ სხვადასხვა შესაძლებლობები გააჩნიათ.
ქსელის დაპროექტებისას უნდა არჩეული იყოს ქსელის ტიპი. ქსელის ტიპის არჩევას ბევრი ფაქტორი განსაზღვრავს მაგალითად, ორგანიზაციის სიდიდე, ქსელური მომხმარებლის მოთხოვნა და ა.შ.
ერთრანგიან ქსელში ჩართული ყველა კომპიუტერი თანაბარი უფლებით სარგებლობს. ისინი ფუნქციონირებენ როგორც კლიენტი და როგორც სერვერი ე.ი არ არის ისეთი კომპიუტერი რომელიც “პასუხს აგებს” ქსელის მუშაობაზე. ისინი ფუნქციონირების თვალსაზრისით ძალიან მარტივნი არიან, ამიტომ მათ გააჩნიათ შედარებით დაბალი ღირებულება სერვერის მქონე ქსელებთან შედარებით. სერვერის მქონე ქსელის შემთხვევაში ქსელში ჩართულია გამოყოფილი სერვერი (*Олифер/ოლიფერ, 1999).*

ერთრანგიან ქსელებს უწოდებენ აგრეთვე მუშა ჯგუფებს. ამიტომ, ერთრანგიან ქსელებში კომპიუტერების რაოდენობა შეზღუდულია. ასეთი ქსელები შედარებით

მარტივია. ისინი სხვა ქსელებზე იაფია, რადგან არ მოეთხოვებათ მძლავრი ცენტრალური სერვერის და ქსელების სხვა კომპონენტების არსებობა. მაგრამ სასურველია, რომ მომხმარებელთა კომპიუტერები იყოს მძლავრი.

ილუსტრაცია N 1 ერთრანგიანი ქსელი და ქსელი სერვერის ბაზაზე



წყარო: (კვლევის მასალები)

ერთრანგიან ქსელში მოთხოვნილება წარმადობაზე და ქსელური პროგრამული უზრუნველყოფის დაცვის დონეზე დაბალია სხვა ქსელებთან შედარებით. ერთრანგიან ქსელებში კომპიუტერებზე დამატებითი პროგრამული უზრუნველყოფის დაყენება არ არის საჭირო. საკმარისია დაყენებული იქნას ოპერაციული სისტემები Windows 98, Windows 2000 ან Windows 7 და ა.შ, რომლებშიც ჩაშენებულია ერთრანგიანი ქსელის ფუნქციონირების უზრუნველყოფა. აღსანიშნავია, რომ კომპიუტერულ ქსელში გაერთიანებისთვის გამოიყენება უბრალო საკაბელო სისტემა. ერთრანგიან ქსელში ყოველი კომპიუტერი ფუნქციონირებს, როგორც კლიენტი და როგორც სერვერი, ამიტომ საჭიროა ყოველი მომხმარებელს ქონდეს ცოდნის საკმარის დონე, რომ მან იმუშაოს, როგორც მომხმარებელმა და როგორც ადმინისტრატორმა

თუ კომპიუტერული ქსელის აგების დროს მომხმარებელთა რიცხვი დიდია, მაშინ მიზანშეწონილია დავაპროექტოთ ქსელი სერვერის ბაზაზე. დღეისათვის კომპიუტერული

ქსელების უმრავლესობაში გამოიყენება გამოყოფილ სერვერი ან ქსელური მარშრუტიზატორი(ROUTER). გამოყოფილი სერვერი ეწოდება კომპიუტერს, რომელიც ფუნქციონირებს მხოლოდ როგორც სერვერი, მისი დანიშნულებაა ქსელის ფაილების და კატალოგების დაცვა.

ქსელის ზომების და ტრაფიკის მოცულობის გაზრდასთან ერთად საჭიროა მაღალი დონის მარშრუტიზატორი. ქსელის ამოცანების განაწილება სხვადასხვა სერვერებს შორის წარმოადგენს ქსელის ფუნქციონირების ეფექტიანობის ამაღლების გარანტიას. ამიტომ, დიდი ზომის ქსელებში, სადაც კომპიუტერების რაოდენობა ასეულობით და ათასეულობით გამოისახება, გამოიყენება როგორც მარშრუტიზატორები აგრეთვე სერვერები, რათა მოხდეს ქსელში მისამართების დარიგება და ინფორმაციის საიმედო გადაცემა, აგრეთვე სხვადასხვა შიდა/გარე კომპიუტერული ქსელის დაკავშირება ერთმანეთან.

სერვერის და მარშრუტიზატორების მიერ შესრულებული ამოცანების კლასი მრავალფეროვანია და რთული. ამიტომ თანამედროვე ქსელებში შეიძლება რამდენიმე მარშრუტიზატორი და სერვერი იყოს გამოყენებული. თითოეული სერვერი ასრულებს მასზე დაკისრებულ ფუნქციებს და მათ სპეციალიზირებულ სერვერებს უწოდებენ. ქსელში შეიძლება არსებობდეს ფაილების დაბეჭვდის სერვერი, დანართების სერვერი, საფოსტო სერვერი და ა.შ.

სერვერი და ოპერაციული სისტემა მუშაობს როგორც ერთიანი მთლიანი, ოპერაციული სისტემის გარეშე ყველაზე მძლავრი სერვერიც კი გამოუსადეგარია. სწორედ იგი იძლევა საშუალებას რეალიზაცია გაუკეთოს სერვერის აპარატურულ რესურსებს.

კომბინირებულ ქსელებში გათვალისწინებულია ერთრანგიანი ქსელებისა და სერვერის ბაზაზე არსებული ქსელების საუკეთესო თვისებები. ესაა ქსელების ყველაზე გავრცელებული ტიპი.

კომბინირებული ქსელების უპირატესობა მდგომარეობს იმაში, რომ ერთდროულად შეიძლება ფუნქციონირებდეს ორივე ტიპის ოპერაციული სისტემა: სერვერის ბაზაზე ქსელში, მაგალითად Windows Serveri-ი, ხოლო კომპიუტერ-კლიენტებზე, მაგალითად, Windows 7, 8 და ასე შემდეგ.

მონაცემთა გადაცემის ქსელების სწრაფმა განვითარებამ წარმოქმნა არა მხოლოდ ავტომატიზებული მართვა, ელექტრონული კომერცია, დისტანციური განათლება, არამედ ხელი შეუწყო მეცნიერების, ტექნიკის, კულტურის, ხელოვნებისა და სხვა სოციალური თუ ეკონომიკური სფეროების პროგრესს.

კომპიუტერული ქსელის ფუნქციონირების ეფექტიანობის ამაღლება, რომელიც ფასდება საიმედოობისა და მწარმოებლობის მაჩვენებლებით, ერთ-ერთი აქტუალური საკითხია. მონაცემთა გადაცემის ქსელის ძირითად ამოცანას წარმოადგენს გარკვეული მოცულობის ინფორმაცია, მისი აუცილებელი ხარისხის გათვალისწინებით, მიეწოდოს მომხმარებელს განსაზღვრულ დროში. ბუნებრივია, რომ რაც უფრო საპასუხისმგებლო და სწრაფად მიმდინარეა პროცესი, მით უფრო მკაცრია მოთხოვნა ინფორმაციის გადაცემის საიმედოობისადმი.

მონაცემთა გადაცემის ქსელების შექმნა და მათი წარმატებით ფუნქციონირება უშუალოდ დაკავშირებულია, როგორც რაციონალური სტრუქტურის მქონე სისტემის დაპროექტებასთან, ასევე მონაცემთა გადაცემის მართვის მეთოდების სრულყოფასთან. მონაცემთა გადაცემის ქსელში საიმედოობის ასამაღლებლად აქტუალურია ისეთი მათემატიკური აპარატის შექმნა, რომელიც საშუალებას მოგვცემს საფუძვლიანად შეირჩეს გამოყენებული ტექნიკური სისტემების საიმედოობაზე მოთხოვნები, მოხდეს მათი ანალიზი და შეფასება.

1.2. კომპიუტერული ქსელების ტოპოლოგია

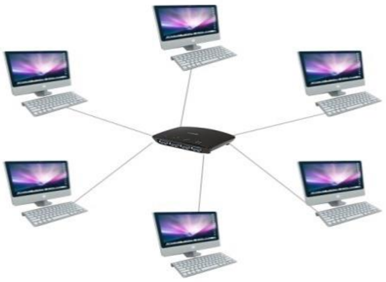
“ტოპოლოგია” ანუ “ქსელების ტოპოლოგია” ნიშნავს კომპიუტერების, კაბელების და სხვა კომპონენტების ფიზიკურ განლაგებას. ტოპოლოგიის არჩევა გავლენას ახდენს საჭირო ქსელური მოწყობილობების შემადგენლობაზე, ქსელის გაფართოების შესაძლებლობაზე და ა.შ.

ნებისმიერი ქსელის დაპროექტება წარმოებს სამი ძირითადი ტოპოლოგიის საფუძველზე: სალტე (linear bus) (ილუსტრაცია N1), ვარსკვლავი (star) (ილუსტრაცია N 2.), და წრე (ring) (ილუსტრაცია N 3).

ილუსტრაცია N 2 - სალტე

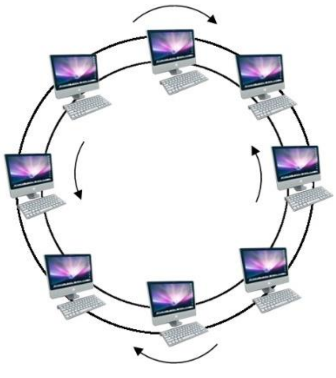
წყარო: (კვლევის მასალები)

ილუსტრაცია N 3 - ვარსკვლავი



წყარო: (კვლევის მასალები)

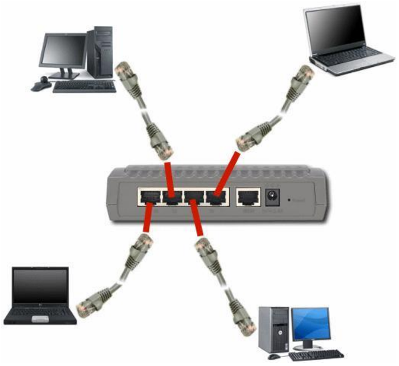
ილუსტრაცია N 4 - წრე



წყარო: (კვლევის მასალები)

"სალტე" ტოპოლოგიის შემთხვევაში კომპიუტერები ერთი კაბელის გასწვრივ არიან შეერთებული. "ვარსკვლავი" ტოპოლოგიის შემთხვევაში კომპიუტერები სხვადასხვა კაბელით არიან შეერთებული. ეს კაბელები კი ერთი წერტილიდან - კონცეტრატორიდან (HUB, SWITCH) გამოდიან. "წრე" ტოპოლოგიის შემთხვევაში კომპიუტერები ჩაკეტილ წრეზე არიან ჩართული.
"სალტე" ტოპოლოგიაში გამოიყენება ერთი კაბელი, რომელსაც მაგისტრალი ქვია.
"სალტე" ტოპოლოგია თავისი სიმარტივის გამო ერთ-ერთ ყველაზე გავრცელებულ ტოპოლოგიას წარმოადგენდა. ასეთი ტოპოლოგიის დროს ხდება ელექტრული სიგნალებით მონაცემების გადაცემა მთელ ქსელში, მაგრამ ინფორმაციას იღებს ის კომპიუტერი, რომლის მისამართიც შეესაბამება მომხმარებლის მიერ მითითებულ მისამართს (ადრესატს). დროის ყოველ მომენტში ინფორმაციის გადაცემა შეუძლია მხოლოდ ერთ კომპიუტერს. რადგან მონაცემთა გადაცემა შეიძლება მხოლოდ ერთი კომპიუტერის მიერ, მისი მწარმოებლურობა დამოკიდებულია სალტეზე მიერთებული კომპიუტერების რაოდენობაზე, რაც მეტია კომპიუტერების რაოდენობა, მით მეტი კომპიუტერი იმყოფება ლოდინის მდგომარეობაში და მით ნელია ქსელი. „სალტე" ტოპოლოგია პასიური ტოპოლოგიაა, რადგან კომპიუტერები მხოლოდ "უსმენენ" ქსელში გადაცემულ მონაცემებს და არ იღებენ მონაწილეობას მათ გამგზავნიდან მიმღებისკენ გადაცემაში. (აქტიური ტოპოლოგიის დროს კომპიუტერები აკეთებენ სიგნალების რეგენერაციას (აღდგენას) და გადასცემენ მათ ქსელში).
"სალტე" ტოპოლოგიის შემთხვევაში ელექტრონული სიგნალები ვრცელდება კაბელის თავიდან ბოლომდე. კაბელის ბოლოს მიღწეული სიგნალი აირეკლება და უკან ბრუნდება ქსელში. ეს კი იწვევს ქსელში გადაცემული ინფორმაციის დამახინჯებას. ამიტომ კაბელის ბოლოს მიღწეული სიგნალი აუცილებლად უნდა "ჩაქრეს" ამისთვის კაბელის ბოლოს აყენებენ ტერმინატორს (terminators), რომელიც ახშობს ამ სიგნალებს.
"ვარსკვლავი" ტოპოლოგიის შემთხვევაში კომპიუტერები კაბელის სეგმენტით უერთდება კონცენტრატორს. მისი საშუალებით ხდება დანარჩენ კომპიუტერებზე სიგნალების გადაცემა. თუ ერთი კომპიუტერი გამოვა მწყობრიდან (ან კაბელი რომელიც კონცენტრატორს და კომპიუტერს ერთმანეთთან აკავშირებს) მაშინ მხოლოდ მას არ შეუძლია გადასცეს ან მიიღის მონაცემები, სხვა დანარჩენი კომპიუტერები მუშაობენ ნორმალურად, მათ მუშაობაზე ეს დაზიანება გავლენას არ ახდენს. მაგრამ თუ დაზიანდა კონცეტრატორი, ქსელი წყვეტს მუშაობას. "ვარსკვლაურ" ტოპოლოგიაში კონცეტრატორი (HUB, SWITCH) არის ცენტრალური კვანძი.

ილუსტრცია N 5 - ქსელიდან გამოირთვება მხოლოდ დაზიანებული კომპიუტერი.



წყარო: (კვლევის მასალები)

არსებობს აქტიური და პასიური კონცენტრატორი. აქტიური კონცენტრატორი მიერთებულია კვების წყაროსთან, რის გამოც მას გააჩნია სიგნალის გაძლიერების და ფორმირების უნარი. აქტიური კონცენტრატორი ახდენს სიგნალის რეგენერაციას და გადაცემას ისევე, როგორც ამას აკეთებს რეპიტერი. ზოგჯერ მათ მრავალპორტიან რეპიტერებს უწოდებენ. კომპიუტერის მისაერთებლად მათ აქვთ რვიდან-თორმეტამდე პორტი.

პასიურ კონცენტრატორებს უწოდებენ სამონტაჟო პანელებს და საკომუნიკაციო ბლოკებს. ისინი, როგორც კომუნიკაციის კვანძები სიგნალებს მხოლოდ ატარებენ, არც აძლიერებენ და არც აღადგენენ. შესაბამისად, პასიური კონცენტრატორები წარმოადგენენ მარტივ მოწყობილობებს და არ საჭიროებენ კვების წყაროებს.

ჰიბრიდულს უწოდებენ კონცენტრატორს, რომელთანაც შესაძლებელია განსხვავებული ტიპის კაბელების მიერთება.

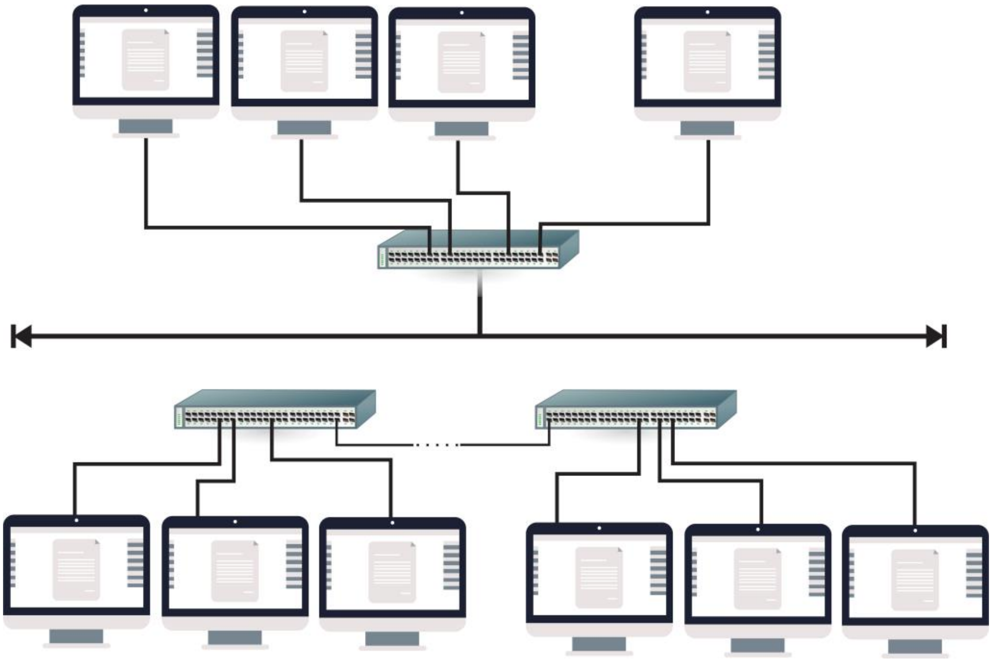
კონცენტრატორების გამოყენებას აქვს მრავალი უპირატესობა:

1. მარტივად სრულდება ქსელის შეცვლა ან გაფართოება: საკმარისია მივუერთოთ კიდევ ერთი კომპიუტერი ან კონცენტრატორი;
2. განსხვავებული პორტების გამოყენება სხვადასხვა ტიპის კაბელების მისაერთებლად.

ქსელის მუშაობის და ქსელის ტრაფიკის ცენტრალიზებული კონტროლი: აქტიურ კონცენტრატორებს გააჩნიათ დიაგნოსტიკის საშუალებები, რითაც განსაზღვრავენ შეერთებების საიმედოობას

“წრე” ტოპოლოგიაში კომპიუტერები შეერთებულნი არიან ერთმანეთთან ჩაკეტილ წრეზე, კაბელს არ აქვს თავისუფალი ბოლო. სიგნალები გადაიცემიან ერთი მიმართულებით და გადიან ყველა კომპიუტერზე. მაგრამ თუ მწყობრიდან გამოვიდა რომელიმე კომპიუტერი, ქსელი წყვეტს მუშაობას. ასეთი სახის ქსელებში მონაცემთა გადაცემა ხდება მარკერით. მარკერი მოძრაობს წრეზე (კომპიუტერიდან კომპიუტერზე) მანამ, სანამ მას არ მიიღებს ის კომპიუტერი, რომელსაც სურს მონაცემთა გაგზავნა. გადამცემი კომპიუტერი მარკერს მიამაგრებს გადასაცემ მონაცემებს და აგზავნის წრეზე. მონაცემები მოძრაობენ წრეზე მანამ, სანამ მას არ მიიღებს მიმღები კომპიუტერი (ანუ კომპიუტერი, რომლის მისამართიც ემთხვევა მიმღების მისამართს). ამის შემდეგ მიმღები კომპიუტერი გზავნის გადამცემ კომპიუტერთან შეტყობინებას მონაცემთა მიღების შესახებ. მიიღებს რა თანხმობას, გადამცემი კომპიუტერი ქმნის ახალ მარკერს და აბრუნებს ქსელში. პრაქტიკაში გვხვდება უფრო რთული კომბინაციები, რომლებიც ერთდროულად რამდენიმე ტოპოლოგიას აერთიანებს ე.წ. კომბინირებული ტოპოლოგიები: ვარსკვლავი-სალტე ტოპოლოგია (star-bus) (ილუსტრაცია N 6).

ილუსტრაცია N 6 ვარსკვლავი-სალტე ტოპოლოგია (star-bus)



წყარო: (კვლევითი მასალები)

ეს არის სალტის და ვარსკვლავი ტოპოლოგიის კომბინაცია. ჩვეულებრივ ასეთი ტოპოლოგიის დროს რამდენიმე ვარსკვლავი ტოპოლოგია არის გაერთიანებული

მაგისტრალური წრფივი სალტით. ასეთი ტოპოლოგიის დროს ერთი კომპიუტერის მწყობრიდან გამოსვლა არ ახდენს გავლენას ქსელის მუშაობაზე, კონცენტრატორის მწყობრიდან გამოსვლა გათიშავს მხოლოდ უშუალოდ მასთან შეერთებულ კომპიუტერებს.
ტოპოლოგიის არჩევისას უნდა გავითვალისწინოთ შემდეგი ფაქტორები (ცხრილი N 1):

ცხრილი N 1 - ტოპოლოგიის ცხრილი

ტოპოლოგია	უპირატესობანი	უარყოფითი
სალტე	კაბელის ეკონომიური დანახარჯი, ინფორმაციის გადაცემა შედარებით იაფია და მისი გაფართოება მარტივია	დიდი ზომის ტრაფიტის დროს მცირდება ქსელის შესაძლებლობანი. კაბელის მწყობრიდან გამოსვლის დროს ქსელი ითიშება.
წრე	ყველა კომპიუტერი თანასწორუფლე ბიანია, კომპიუტერების რიცხვი თითქმის არ ახდენს გავლენას ქსელის მწარმოებლურობაზე	ერთი კომპიუტერის მწყობრიდან გამოსვლა ქსელს თიშავს. ქსელის გაფართოებისათვის მთელი ქსელი უნდა გაითიშოს.
ვარსკვლავი	მარტივად ხდება ქსელის მოდიფიკაცია. ერთი კომპიუტერის მწყობრიდან გამოსვლა არ იწვევს ქსელის გათიშვას.	კომუნიკატორის გათიშვა იწვევს ქსელის გათიშვას.

წყარო: (კვლევითი მასალები)

კომპიუტერულ ქსელებში კომპიუტერების ერთმანეთთან შესაერთებლად გამოიყენება როგორც ინდივიდუალური (გამოყოფილი) ასევე საერთო (shared) ხაზები. საერთო ხაზების მაგალითია, მაგალითად, ტოპოლოგია "საერთო სალტე", რომელშიც ერთი კაბელი გამოიყენება ქსელში ჩართული ყველა კომპიუტერის მიერ მონაცემთა გადასაცემად. არცერთ კომპიუტერს არ შეუძლია ინდივიდუალურად გამოიყენოს კაბელი. "ვარსკვლავ" და "წრე" ტოპოლოგიაში კაბელის ინდივიდუალური გამოყენება შესაძლებელია. კომპიუტერების დიდი რაოდენობის არსებობის დროს, საერთო ქსელის შემთხვევაში ქსელი მუშაობს ძალიან ნელა, ვიდრე გამოყოფილი ხაზების შემთხვევაში (Mansfield & Antonakos/მანსფილდი & ანტონაკოსი, 2010).

1.3. *კომპიუტერების დამისამართება ქსელში და ქსელის სტრუქტურიზაცია*

კომპიუტერების ქსელის შექმნის პროცესში ერთ-ერთ პრობლემას წარმოადგენს დამისამართების პრობლემა. კომპიუტერის (ანუ კვანძის) მისამართი უნდა იყოს:

1. უნიკალური, რადგან შესაძლებელი იყოს მისი დარეგისტრირება ნებისმიერ ქსელში.
2. მას უნდა ქონდეს იერარქიული სტრუქტურა, რადგან დიდი ქსელების ასაგებად იყოს მოსახერხებელი. თუ მისამართს არ ექნება იერარქიული სტრუქტურა, მაშინ საკომუტაციო საშუალებებს ადრესატის მდებარეობის დასადგენად მოუწევთ უამრავი მისამართების ცხრილების გაანალიზება.
3. მისამართს უნდა ქონდეს სიმბოლური სახე, რადგან მომხმარებლისათვის იყოს ადვილად დასამახსოვრებელი. მაგალითად, www.ug.edu.ge.

ამ მოთხოვნების ერთი ტიპის მისამართებში გათვალისწინება შეუძლებელია. დღესდღეობით ძირითადად გავრცელებულია დამისამართების სამი სახე, ე.ი კომპიუტერს ერთდროულად აქვს რამდენიმე სახელი-მისამართი და საჭიროების შემთხვევაში იყენებს შესაბამის მისამართს. კომპიუტერი ცალსახად განისაზღვრება თავისი მისამართით. არსებობს სპეციალური პროტოკოლები (წესები), რომლებიც ერთი ტიპის მისამართების საშუალებით განსაზღვრავენ სხვა მისამართებს.
აპარატურული მისამართი მოწყობილობაშია ჩაშენებული ან ავტომატურად ფორმირდება მოწყობილობების დაყენებისას, მაგალითად, ქსელური ადაპტერის მისამართი. იგი წარმოდგენილია ორობითი ან თექვსმეტობითი კოდის სახით, მაგალითად, 0081005e24a8. ასეთი მისამართისათვის უარყოფითია ის, რომ მოწყობილობის შეცვლისას, მაგალითად, თუ შევცვლით ქსელურ ადაპტერს, კომპიუტერის მისამართიც შეიცვლება. ასეთი მისამართები შეიძლება გამოყენებული იყოს არა დიდი ზომის ქსელებში. აპარატურულ მისამართს არ აქვთ იერარქიული სტრუქტურა.
სიმბოლური მისამართი ანუ სახელი განკუთვნილია მომხმარებელთათვის, იგი გამოიყენება როგორც დიდ, ასევე არა დიდ ქსელებში, მათი ქსელში გადაცემა არაეკონომიურია.
რიცხვითი მისამართები გამოიყენება დიდ ქსელში კომპიუტერების დასამისამართებლად. თანამედროვე ქსელებში გამოიყენება ერთდროულად სამივე ტიპის დამისამართება. მომხმარებელი ამისამართებს თავის კომპიუტერს სიმბოლური სახელით, რომელიც ქსელში გადაცემისას ავტომატურად შეიცვლება რიცხვითი მისამართით, ამ რიცხვითი

მისამართით შეტყობინებები გადადის ქსელიდან ქსელში, ხოლო როდესაც კომპიუტერი მიმღები მიიღებს შეტყობინებას, მაშინ გამოიყენება კომპიუტერის აპარატურული მისამართი.
სხვადასხვა ტიპის მისამართებს შორის შესაბამისობის დადგენა ხდება ცენტრალიზებულად ან დამოუკიდებლად.
ცენტრალიზებული მეთოდის შემთხვევაში ქსელში გამოიყოფა ერთი კომპიუტერი (სერვერი) ან ქსელური მაღალი დონის მოწყობილობა (მარშრუტიზატორი, ფაირვოლი), რომელშიც ინახება ინფორმაცია სიმბოლურ და რიცხვითი მისამართების შესაბამისობის შესახებ და ყველა კომპიუტერი მიმართავს მას.
დამოუკიდებელი მეთოდის შემთხვევაში თითოეული კომპიუტერი თვითონ ადგენს მისამართებს შორის შესაბამისობას. მაგალითად, თუ მომხმარებელმა, რომელიც ქსელში გზავნის ინფორმაციას, მიუთითა რიცხვითი მისამართი, ქსელში მონაცემთა გადაცემის წინ გადამცემი კომპიუტერი ყველა კომპიუტერს გაუგზავნის შეტყობინებას (მისამართს). კომპიუტერები მიიღებენ რა შეტყობინებას, ადარებენ მას თავის საკუთარ მისამართს. ის კომპიუტერი, რომლის მისამართიც დაემთხვევა, უგზავნის პასუხს თავისი აპარატურული მისამართის მითითებით, რის შედეგადაც იწყება ლოკალურ ქსელში მონაცემთა გადაცემა. მაგრამ შეტყობინებების გადაცემამ კომპიუტერის მისამართის მოსაძებნად, შეიძლება ძალიან გადატვირთოს ქსელი. ასეთი მეთოდის გამოყენება შესაძლებელია პატარა ქსელებში. დიდი ქსელებისათვის გამოიყენება ცენტრალიზებული მეთოდი, მაგალითად, ინტერნეტში ასეთი მეთოდია Domein Name System (DNS).
კომპიუტერულ ქსელების ასაგებად, რომელთა ზომა დიდი არ არის (კომპიუტერების რიცხვი 10-30) ძირითადი ტოპოლოგიებიდან ერთ-ერთი გამოიყენება. ყველა კომპიუტერს მონაცემთა გადაცემისას მასში აქვს თანაბარი უპირატესობა. მაგრამ დიდი ზომის ქსელების ასაგებად, ეს არ არის საკმარისი, რადგან კვანძებს შორის მანძილი შეიძლება იყოს დიდი, ასევე კომპიტერების რაოდენობა შეიძლება იყოს ბევრი, ეს კი იწვევს ტრაფიკის ინტესიობის გაზრდას. ამ პრობლემების მოსახსნელად გამოიყენება ქსელების სტრუქტურიზაციის (კონფიგურაციის) მეთოდები და მოწყობილობები რომლებიც ცვლიან ქსელის სტრუქტურას. მათ ხშირად საკომუნიკაციო მოწყობილობებსაც უწოდებენ.
ფიზიკური ტოპოლოგია ესაა ქსელების კონფიგურაცია, რომელიც კაბელის სხვადასხვა სეგმენტების მიერ არის შექმნილი.

ლოგიკური სტრუქტურიზაცია ესაა კომპიუტერებს შორის ინფორმაციული ნაკადის გადაადგილება. ზევრ შემთხვევაში ქსელის ფიზიკური და ლოგიკური ტოპოლოგია ერთმანეთს ემთხვევა.

ჩვენ ზემოთ განვიხილეთ ქსელების ტოპოლოგიები, ყველა მათგანს გააჩნია თვისება, რომ ყველა კომპიუტერს აქვთ ერთნაირი წვდომა სხვა კომპიუტერებთან. თუ კომპიუტერები ინტესიურად ცვლიან ერთმანეთში ინფორმაციას, საჭიროა შემცირდეს ქსელში ჩართული კომპიუტერების რიცხვი. ამისთვის გამოიყენება: გამმეორებელი, კონცეტრატორი, ხიდები, კომუტატორი და მარშრუტიზატორი.

გამმეორებელი (repeator) (ილუსტრაცია N 7) – ყველაზე მარტივი საკომუნიკაციო მოწყობილობაა, რომლის საშუალებითაც ხდება ლოკალური ქსელის კაბელის სხვადასხვა სეგმენტების ფიზიკურად შეერთება. ამით კი ხდება კაბელის სიგრძის გაზრდა, ანუ იგი ცვლის ქსელის ფიზიკურ სტრუქტურას. გამმეორებელი გადასცემს სიგნალს ქსელის ერთი სეგმენტიდან სხვა სეგმენტში.

ილუსტრაცია N 7 კომპიუტერული ქსელი გამმეორებელის გამოყენებით

წყარო: (კვლევითი მასალები)

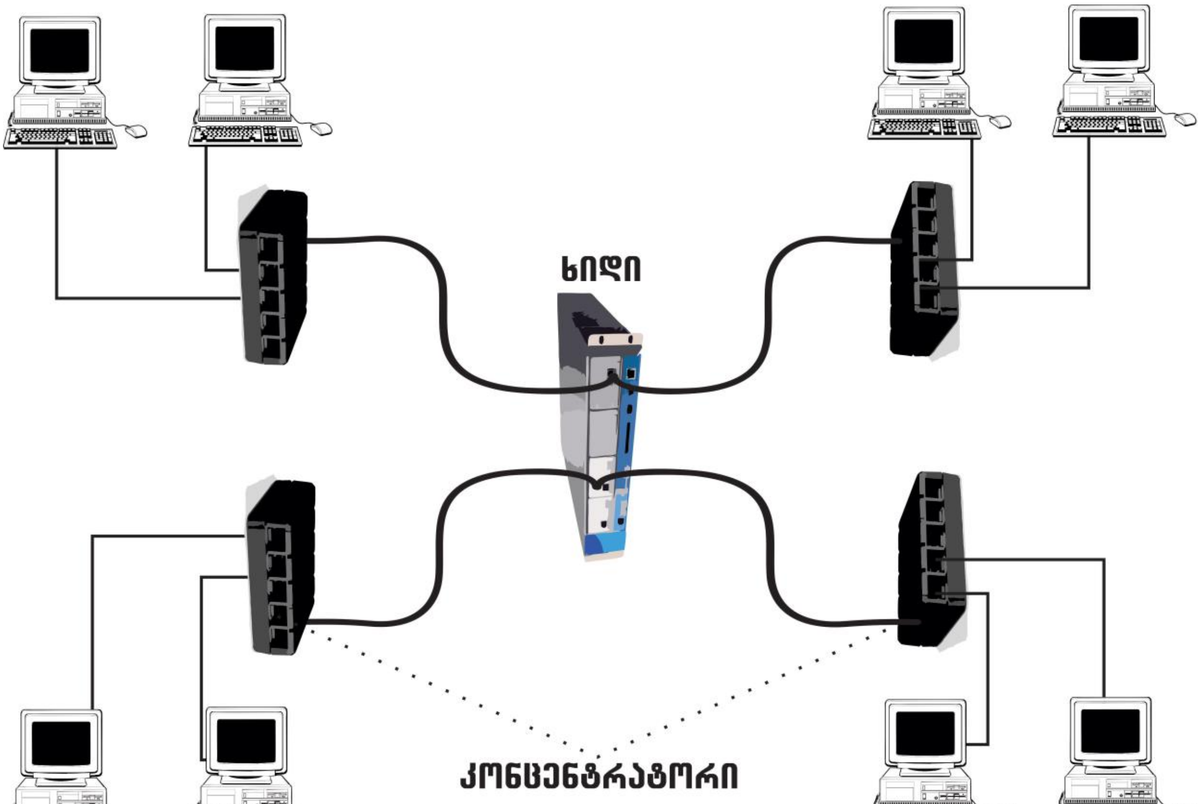
თუ გამმეორებელს აქვს რამდენიმე პორტი და აერთებს რამდენიმე ფიზიკურ სეგმენტს, მაშინ მას უწოდებენ კონცენტრატორს (concetrator). კონცენტრატორის ერთერთ პორტზე მოსულ სიგნალს იმეორებს მის სხვა პორტებზე.

ქსელების ლოგიკური სტრუქტურიზაციისთვის გამოიყენება ხიდები, მარშუტიზატორები და შლუზები.

ხიდი (bridge) გადაცემის გარემოს ყოფს ლოგიკურ სეგმენტებად, ინფორმაციის გადაცემა სეგმენტიდან სეგმენტში ხდება აუცილებლობის შემთხვევაში. ე.ი მაშინ თუ მიმღები

კომპიუტერი ეკუთვნის სხვა ქსელს. ხიდი ტრაფიკის ლოკალიზაციისთვის იყენებს კომპიუტერის აპარატურულ მისამართს. ამიტომ რთულია განისაზღვროს რომელ ლოგიკურ სეგმენტს ეკუთვნის კომპიუტერი. თვითონ მისამართიც არ შეიცავს ამის შესახებ არანაირ ინფორმაციას, ამიტომ ხიდი იმახსოვრებს თვითეული კომპიუტერიდან რომელ პორტზე მოვიდა კადრი და შემდეგ გადასცემს შესაბამის ინფორმაციას შესაბამის მისამართზე. ილუსტრაცია N 8-ზე ნაჩვენებია ქსელი, რომლის ქვექსელები აგებულნი არიან კონცენტრატორებზე, მათ გასაერთიანებლად კი გამოიყენება ხიდი. ქსელი 1 ,2, 3, 4 წარმოადგენენ დამოუკიდებელ ლოგიკურ სეგმენტს. ყოველი სეგმენტი აგებულია კონცენტრატორის ბაზაზე და აქვთ მარტივი სტრუქტურა

ილუსტრაცია N 8 ქსელის ლოგიკური სტრუქტურიზაცია ხიდის საშუალებით.



წყარო: (კვლევითი მასალები)

კომუტატორის (switch) მუშაობის პრინციპით თითქმის არ განსხვავდება ხიდისაგან. ძირითადი განსხვავება ხიდისაგან ისაა, რომ ის არის საკომუნიკაციო მულტიპროცესორი. მისი ყოველი პორტი ამუშავებს კადრებს ხიდის ალგორითმის მიხედვით. ამით კი კომუტატორის მწარმოებლურობა შედარებით იზრდება.

მარშრუტიზატორების (router) საშუალებით უფრო ეფექტურად ხდება ცალკეული ქსელების ერთმანეთისგან გამოყოფა. მარშრუტიზატორები ქმნიან ლოგიკურ სეგმენტებს. ისინი იყენებენ ციფრულ მისამართებს. ამ მისამართებში არის ქსელის ნომრის ველი, ყველა კომპიუტერი, რომლებსაც აქვთ ერთნაირი მნიშვნელობა, ამ ქსელში მიეკუთვნებიან ერთ სეგმენტს (ქვექსელს). გარდა ამისა, მარშრუტიზატორები ასრულებენ ბევრ სხვა ფუნქციას. მისი საშუალებით შესაძლებელი ხდება სხვადასხვა ქსელურ ტექნოლოგიებზე (მაგალითად, Ethernet და X.25) აგებული ქსელების გაერთიანება ერთ ქსელში.

შლუზები (gateway) ასევე ქსელების ცალკეული ნაწილების შესაერთებლად გამოიყენება. მათი ძირითადი დანიშნულებაა გააერთიანოს სხვადასხვა ტიპის ქსელების სისტემური და გამოყენებითი პროგრამული უზრუნველყოფა.

დიდი ქსელების ასაგებად ყოველთვის გამოიყენება ლოგიკური სტრუქტურიზაცია. ცალკეული ქვექსელების აგება ხდება ძირითადი ტოპოლოგიების საფუძველზე მათ გასაერთიანებლად კი გამოიყენება მოწყობილობები, რომლებიც იწვევენ ტრაფიკის ლოკალიზაციას.

1.4. *ინფორმაციის გადაცემის გარემო*

ინფორმაციის გადაცემის გარემო ესაა კომპიუტერების ერთმანეთთან დაკავშირების საშუალება, რომლითაც ხდება ინფორმაციის გაცლა. კომპიუტერულ ქსელებში გადაცემის გარემოდ გამოყენებულია კაბელები და უგამტარო კავშირები. არსებობს კაბელების სამი ძირითადი ტიპი: კოაქსიალური (ილუსტრაცია N 9) (coaxial cabel), ხვეული წყვილი (twisted pair) , (ილუსტრაცია N 10) და ოპტიკურბოჭკოვანი (ილუსტრაცია N 11) (fiber optic). კოაქსიალური (coaxial cabel) შედგება:

1) გამტარისაგან,

2) იზოლაციისაგან,

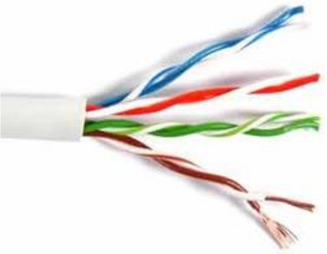
3) ეკრანისაგან,

4) გარე გარსისაგან (იზოლაცია)

ილუსტრაცია N 9 - კოაქსალური გამტარი

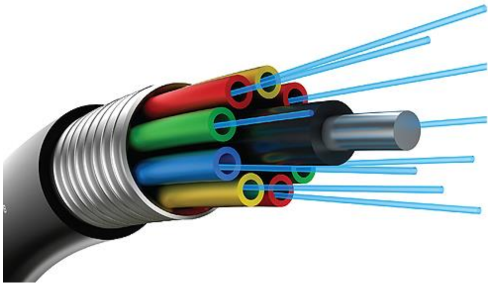
წყარო: (www.google.com)

ილუსტრაცია N 10 - ხვეული წყვილი



წყარო: (www.google.com)

ილუსტრაცია N 11 - ოპტიკური კაბელი



წყარო: (www.google.com)

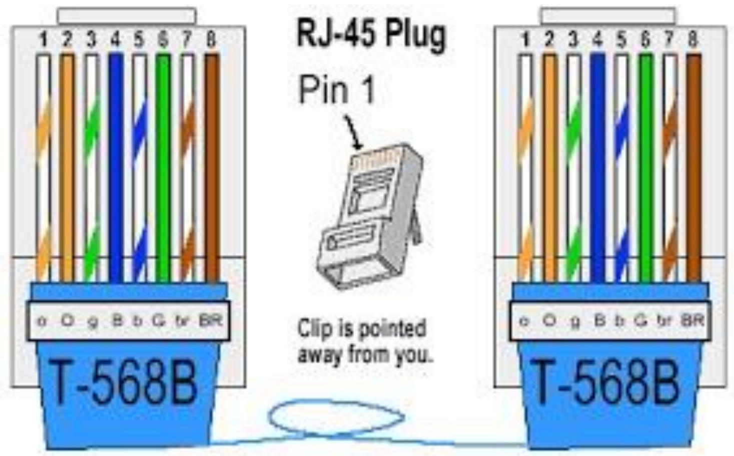
კოაქსიალური კაბელი ყველაზე მეტად გავრცელებულია მისი სიაფის, სიმსუბუქის, გამოყენებისთვის მოხერხებულობის გამო და ასევე მისი დაყენების სიმარტივის გამო. მარტივი კოაქსიალური კაბელი შედგება სპილენძის გამტარისაგან, იზოლაციისაგან, რომელიც ირგვლივ აქვს გამტარს, მეტალური წნულისაგან (ეკრანისაგან) და გარეთა გარსისაგან. ზოგჯერ მეტალური წნულის გარდა აქვს ფოლგის ფენა, მაშინ მას ქვია კაბელი ორმაგი ეკრანიზაციით. ძლიერი შეფერხებების დროს შეიძლება გამოყენებული იქნას კაბელი ოთხმაგი ეკრანიზაციით. იგი შედგება ფოლგის ორი ფენისაგან და მეტალური წნულის ორი ფენისაგან. ელექტრული სიგნალები გადაიცემა გამტარში, გამტარი მზადდება სპილენძისაგან. გამტარი გარშემორტყმულია დიალექტრიკული ფენით რომელიც მას მეტალური წნულისაგან გამოყოფს. წნული მიწის როლს ასრულებს და იგი იცავს გამტარს ელექტრული სიგნალისაგან და გადამკვეთი შეფერხებებისაგან. გადამკვეთი შეფერხებები ესაა ელექტრული დაზიანება, რომელსაც იწვევს მეზობელ გამტარებში სიგნალებს. გამტარი და მეტალის წნული ერთმანეთს არ უნდა ეხებოდეს, რადგან წარმოიქმნება მოკლე ჩართვა და მონაცემები დამახინჯდება. კოაქსიალური კაბელი შეფერხებების მიმართ უფრო მდგრადია, ვიდრე ხვეულა წყვილი. მასში სიგნალების მილევა უფრო მცირეა. სიგნალის მილევა – ეს არის კაბელში გავლისას სიგნალების შესუსტება.

კოაქსიალური კაბელის ორი ტიპი არსებობს: წვრილი კოაქსიალური კაბელი (thinnet) და მსხვილი კოაქსიალური კაბელი (thicknet). წვრილი კოაქსიალური კაბელი მოქნილი კაბელია დიამეტრით 0.5 სმ-დე, იგი გამოიყენება ნებისმიერი ტიპის ქსელისთვის და უშუალოდ ერთვება ქსელის ადაპტერის პლატას. ასეთ კაბელებს დაუმახინჯებლად

ინფორმაციის გადაცემა შეუძლია 185 მ-დე. სქელი კოაქსიალური კაბელი შედარებით ხისტი კაბელია დიამეტრით 1 სმ-მდე. რაც მეტია სასიგნალო გამტარის სისქე, მით მეტ მანძილზე შეუძლია მას სიგნალების გადაცემა დაუმახინჯებლად. სქელ კოაქსიალური კაბელს მონაცემთა გადაცემა დაუმახინჯებლად შეუძლია 500 მ-დე მანძილზე. ამიტომ მას იყენებენ, როგორც მაგისტრალი, რომელიც რამდენიმე პატარა ქსელს ერთმანეთთან აერთიანებს. ასეთი კაბელების მისაერთებლად ქსელის ადაპტერის პლატასთან გამოიყენება ტრანსივერი.

ყველაზე მარტივი ხვეულა წყვილი ეს არის ორი ერთმანეთისგან გამოყოფილი სასიგნალო გამტარი. არსებობს ხვეულა წყვილის ორი ტიპი: ეკრანირებული (STP) როდესაც სპილენძის გამტარები ერთმანეთისგან იზოლაციით (ეკრანით) არიან გამოყოფილი და არაეკრანირებული (UTP), როდესაც იზოლაცია არ არსებობს. ხვეულა წყვილის მისაერთებლად კომპიუტერთან გამოიყენება კონექტორი RG-45.

ილუსტრაცია N 12 - კონექტორი RG-45.



წყარო : (www.google.com)

ოპტიკურბოჭკოვან კაბელში მონაცემთა გადაცემა ხდება მოდულირებული სინათლის იმპულსების სახით. იგი მონაცემთა გადაცემის შედარებით დაცული ხერხია. ასეთი ტიპის ხაზები გამოიყენება დიდი მოცულობის მონაცემების გადასაცემად მაღალი სისწრაფით, (10 გიგაბაიტი და მეტი). მათში სიგნალების მილევა და დამახინჯება თითქმის არ ხდება. ოპტიკური ბოჭკო – წვრილი შუშის ცილინდრია (5-60 მიკრონი), რომელსაც ქვია

სასიგნალო გამტარი და რომელიც დაფარულია შუშის ფენით. ყოველი ოპტიკური ბოჭკო სიგნალს გადაცემს ერთი მიმართულებით, ამიტომ ყოველი კაბელი შედგება ორი ოპტიკური ბოჭკოსგან, რომლებსაც აქვთ დამოუკიდებელი კონექტორები ერთი გამოიყენება გადასაცემად, მეორე – მიმღებად. დღესდღეობით კომპიტერულ ქსელებში გამოიყენება სამივე ტიპის კაბელი, მაგრამ ყველაზე პერსპექტიულია ოპტიკურბოჭკოვანი კაბელი. იგი გამოიყენება მაგისტრალების ასაგებად.

ოპტიკურობოჭკოვანი კაბელით ინფორმაციის გადაცემის დროს მასზე არ მოქმედებს ელექტრული შეფერხებები, არ ხდება სიგნალის დამახინჯება და მილევა. ამიტო გადაცემა ხდება ძალიან მაღალი, წამში ასობით მეგაბიტი სიჩქარით, რომლის თეორიული ზღვარი 200000 მგბიტ/წმ-ის ტოლია. სინათლის იმპულსი დაუმახინჯებლად ვრცელდება 2 კმ მანძილზე.

უგამტარო შეერთებები გამოიყენება მონაცემთა გადასაცემად ლოკალურ გამოთვლით ქსელებში, გაფართოებულ ლოკალურ გამოთვლით ქსელებში და მობილურ ქსელებში. ტიპიური უგამტარო ქსელი მუშაობს ისე, როგორც კაბელური ქსელი. უგამტატო ადაპტერის პლატა ყენდება ყოველ კომპიუტერზე. მომხმარებლები მუშაობენ ისე, თითქოს კომპიუტერები შეერთებულნი არიან კაბელის საშუალებით.

უგამტარო ლოკალურ ქსელებში გამოყენებულია მონაცემთა გადაცემის ოთხი პრინციპი:

1. ინფრაწითელი გამოსხივება;
2. ბლუთუზი;
3. რადიო გადაცემა ვიწრო ზოლში;
4. რადიო გადაცემა ფართო ზოლში;

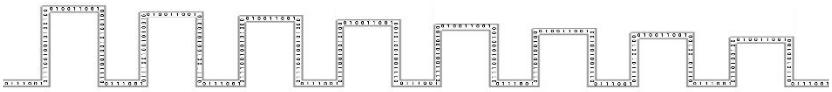
ინფრაწითელი და ბლუთუზური გადაცემები მოითხოვენ ობიექტის პირდაპირ ხედვას: გადამცემი და მიმღები უნდა „ხედავდნენ“ ერთმანეთს.

უგამტარო მობილურ ქსელებში გადაცემის გარემოდ გამოყენებულია სატელეფონი სისტემები, რომლებშიც ჩართულია პაკეტური რადიოგადაცემა, ფიჭური ქსელი და თანამგზავრის სადგურები.

სიგნალების გადაცემა. კაბელში კოდირებული სიგნალების გადაცემისათვის გამოიყენება ორი ტექნოლოგია - ვიწროზოლიანი გადაცემა და ფართოზოლიანი გადაცემა. ვიწროზოლიანი სისტემები მონაცემებს ერთი სიხშირის ციფრული სიგნალის სახით გადაცემენ. სიგნალები წარმოადგენენ დისკრეტული ფორმის ელექტრულ ან სხივურ იმპულსებს (ილუსტრატორი N 13). ამ ტექნოლოგიის დროს გადაცემის არხის მთელი

მოცულობა გამოყენებულია ერთი სიგნალის გადასაცემად, სხვა სიტყვებით - ციფრული სიგნალი მთლიანად იყენებს კაბელის გატარების ზოლს.
ვიწროზოლიანი გადაცემის ქსელებში ყოველი გადამცემი მოწყობილობა ორივე მიმართულებით გზავნის მონაცემებს, ზოგიერთი კი მათ გადასცემს და ღებულობს ერთდროულად.

ილუსტრაცია N 13 არამოდულირებული გადაცემა (ციფრული სიგნალი)



წყარო : (კვლევითი მეთოდები)

კაბელში გავრცელების დროს სიგნალი თანდათან მიილევა და მახინჯდება. თუ კაბელი ძალიან გრძელია ვიწროზოლიან სისტემებში გამოიყენება რეპიტერები. რეპიტერი აძლიერებს სიგნალს, აძლევს მას თავდაპირველ ფორმას (აფორმირებს) და გადასცემს შემდგომ სეგმენტში.
ფართოზოლიანი სისტემები მონაცემებს გადასცემენ ანალოგური სიგნალის სახით, რომელსაც გამოყენებული აქვს კაბელის გატარების ზოლის სიხშირეთა გარკვეული ინტერვალი. სიგნალი წარმოადგენს უწყვეტ (და არა დისკრეტულ) ელექტრომაგნიტურ ან ოპტიკურ ტალღებს. ტალღები ფიზიკურ გარემოში ვრცელდებიან ერთი მიმართულებით. ამ პრინციპით კაბელში ერთდროულად შეიძლება გადაიცეს რამოდენიმე სიგნალი, ისეთი, როგორც საკაბელო ტელეხედვა და მონაცემთა გადაცემა.
ყოველ გადამცემ სისტემას გამოყოფილი აქვს გატარების ზოლის ნაწილი. მოცემულ სისტემასთან დაკავშირებული ყველა მოწყობილობა (მაგ. კომპიუტერის მიმღებ-გადამცემი მოწყობილობა) ისე უნდა იყოს მომართული, რომ იმუშაოს გატარების ზოლის მისთვის განკუთვნილ ნაწილში.
სიგნალის აღდგენის მიზნით, თუ ვიწროზოლიან სისტემებში გამოიყენება რეპიტერები, ფართოზოლიანი სისტემები იყენებენ გამაძლიერებლებს (ამპლიფიერს).

ქსელური ადაპტერის პლატები. ქსელური ადაპტერის პლატა (ანუ ქსელური ადაპტერი) ამყარებს ფიზიკურ კავშირს კომპიუტერებსა და გადაცემის გარემოს შორის. პლატები იდგმება ქსელურ კომპიუტერებსა და სერვერებში. მათ აქვთ შემდეგი დანიშნულება:

1. კომპიუტერიდან მიღებულ მონაცემების მომზადება ქსელში გადასაცემად,
2. მონაცემების გადაცემა სხვა კომპიუტერზე,
3. კომპიუტერსა და კაბელს შორის მონაცემთა ნაკადების მართვა.

გარდა ამისა, ქსელური ადაპტერი იღებს მონაცემებს კაბელიდან და გადაჰყავს ისინი პროცესორისთვის გასაგებ ფორმატში.

ქსელური ადაპტერი შედგება აპარატურული ნაწილისაგან და ოპერატიულ დამახსოვრების მოწყობილობებში ჩაშენებული პროგრამებისაგან. ქსელში მონაცემთა გაგზავნამდე ქსელური ადაპტერი მონაცემებს კომპიუტერის გასაგები ფორმატიდან გადაიყვანს ისეთ ფორმატში რომლითაც ისინი გადაიცემიან კაბელში.

მონაცემთა მომზადება გადასაცემად. კომპიუტერის შიგნით მონაცემთა გადაცემა ხდება სალტეებით. სალტე ეს არის ერთმანეთის პარალელურად განლაგებული რამდენიმე გამტარი. სალტეებში ხდება მონაცემთა პარალელური გადაცემა, ქსელურ კაბელში კი მონაცემთა გადაცემა ხდება მიმდევრობით (ბიტების ნაკადებად). ქსელური ადაპტერი იღებს პარალელურ მონაცემებს და გარდაქმნის მათ თანმიმდევრულ მონაცემებად. შემდეგ კი ხდება კომპიუტერის ციფრული მონაცემების გადაყვანა ელექტრულ ან ოპტიკურ სიგნალად, რომელიც კაბელში გადაიცემა. ამ გარდაქმნას ახორციელებს ტრანსივერი.

ქსელური მისამართი. ქსელური ადაპტერის პლატას აქვს უნიკალური მისამართი, რათა მათი გარჩევა შეიძლებოდეს სხვა პლატებისაგან. ქსელური მისამართები იმყოფება IEEE (Institute of Electrikal and Electronics Engineers)   ეს კომიტეტი ქსელური ადაპტერი ყველა მწარმოებელს აწვდის მისამართების ინტერვალს. შემდეგ ყველა მწარმოებელი პლატის ოპერატიულ დამხსომებელ მოწყობილობებში წერს უნიკალურ ქსელურ მისამართს. მონაცემთა გადაცემა პროცესორიდან ქსელური ადაპტერში უფრო სწრაფად ხდება, ვიდრე ქსელური ადაპტერი მას გადასცემს ქსელში, ამიტომ დროებით მათი მოთავსება ხდება ბუფერში.

მონაცემთა გადაცემა და მართვა. ქსელში მონაცემების გაგზავნამდე ქსელური ადაპტერი მიმღებ კომპიუტერის ქსელურ ადაპტერთან მართავს “დიალოგს” შემდეგ საკითხებზე:

1. გადასაცემი მონაცემების ბლოკის მაქსიმალური ზომა
2. გადასაცემ მონაცემთა მოცულობა

3. მონაცემთა ბლოკების გადაცემებს შორის ინტერვალი
4. მონაცემთა მოცულობა რომელიც შეიძლება მიიღოს ყოველმა პლატამ ბუფერის გადავსების გარეშე
5. გადაცემის სიჩქარე: თუ უფრო სწრაფ პლატას უხდება მონაცემთა გადაცემა შედარებით ნელ პლატასთან, ისინი უნდა შეთანხმდენ გადაცემის სიჩქარეზე. ამ პარამეტრების განსაზღვრის შემდეგ ისინი იწყებენ მონაცემთა გაცვლას.

ქსელური ადაპტერის პარამეტრები. წყვეტის ნომერი (IRQ), შეტანა-გამოტანის პორტის საბაზო მისამართი (In/Out Port, Address), მეხსიერების საბაზო მისამართი და ტრანსივერის ტიპი უნდა იყვნენ კორექტულად დაყენებული.

წყვეტა. წყვეტის მოთხოვნის ხაზები ფიზიკური ხაზებია, რომლითაც სხვადასხვა ძირითად და პერიფერიულ მოწყობილობას (კლავიატურას, შეტანა/გამოტანის პორტებს, დისკზე დამგროვებლებს, ქსელურ ადაპტერს) შეუძლია კომპიუტერის მიკროპროცესორს გაუგზავნოს სიგნალი (მოთხოვნა) მომსახურებაზე ანუ წყვეტაზე.

წყვეტის მოთხოვნის ხაზები კომპიუტერის აპარატურაში არის ჩამონტაჟებული. მათ სხვადასხვა პრიორიტეტები აქვთ, რაც პროცესორს აძლევს საშუალებას განსაზღვროს მოთხოვნის ხარისხი. კომპიუტერის ყველა მოწყობილობა ვალდებულია ისარგებლოს განსხვავებული წყვეტის მოთხოვნის ხაზებით ანუ წყვეტით (IRQ). წყვეტის მოთხოვნის ხაზისდადგენა ხდება მოწყობილობის აწყობის დროს.

უმეტეს შემთხვევაში ქსელური ადაპტერის პლატას შეუძლია აირჩიოს IRQ3 და IRQ5 წყვეტის ხაზები, თუ ისინი დაკავებული არ არიან. წინააღმდეგ შემთხვევაში ქსელური ადაპტერის პლატისათვის შეიძლება შეირჩეს IRQ10 და IRQ11-დან ერთ-ერთი. ამ ორი ხაზის დაკავება არ შეუძლია მოცემული კომპიუტერის არც ერთ სხვა მოწყობილობას.

შეტანა/გამოტანის საბაზო პორტი. შეტანა/გამოტანის საბაზო პორტი (base i/o port) განსაზღვრავს არხს, რომლითაც ხდება მონაცემების გადაცემა კომპიუტერის მოწყობილობაში, მაგალითად, ქსელური ადაპტერის პლატასა და ცენტრალურ პროცესორს შორის. ცენტრალური პროცესორისათვის პორტი ნიშნავს მოწყობილობის მისამართს. სისტემის ყოველ მოწყობილობას უნდა ჰქონდეს შეტანა/გამოტანის საბაზო პორტის უნიკალური მისამართი. პორტების მისამართები მოცემულია სპეციალურ ცხრილში (თექვსმეტობით ფორმატში). აქედან მისამართი, რომელიც დაკავებული არ არის, შეიძლება გამოიყოს ქსელური ადაპტერის პლატისათვის.

მეხსიერების საბაზო მისამართი. მეხსიერების საბაზო მისამართი მუთითებს კომპიუტერის ძირითადი მეხსიერების ოპერატიულ დამახსოვრების მოწყობილობების იმ

არეს, რომელიც გამოყოფილია ქსელური ადაპტერის პლატის მიერ, ბუფერად მონაცემთა შემომავალი და გამავალი კადრების დროებით დამახსოვრებისათვის. ამ მისამართს ზოგჯერ ოპერატიული დამახსოვრების მოწყობილობების საწყისს მისამართს უწოდებენ. ზოგიერთი პლატები საშუალებას გვაძლევენ გამოიყოს 16კბ, 32კბ ან მეტი მოცულობის მეხსიერება. რაც მეტ მეხსიერებას გამოვყოფთ ბუფერისათვის, მით უფრო სწრაფი იქნება ქსელი, მაგრამ მით ნაკლები მეხსიერება დარჩება სხვა ამოცანების შესასრულებლად. ქსელური ადაპტერის პლატებს, რომლებსაც აქვთ საკუთარი მეხსიერება და არ იყენებენ სისტემის მეხსიერებას ეს პარამეტრი, მეხსიერების საბაზო მისამართი, არ აქვთ.

ტრანსივერის შერჩევა. ქსელური ადაპტერის პლატას შეიძლება ჰქონდეს დამატებითი პარამეტრები, რომელთა მოცემა კონფიგურაციის დროს აუცილებელია. მაგალითად, ქსელური ადაპტერის პლატა შეიძლება დამზადებული იყოს გარე და ჩაშენებული ტრანსივერით. კონფიგურაციაში აუცილებელია მიეთითოს ის ტრანსივერი, რომელიც იქნება გამოყენებული. ტრანსივერის არჩევა ხშირად ხდება გადამრთველების, საშუალებით. (გადამრთველები აერთებს ორ გამოსასვლელს და განსაზღვრავს რომელი კავშირის ხაზი იმუშავებს).

მონაცემთა სალტე. კომპიუტერი და ქსელი რომ იყოს თავსებადი ქსელური ადაპტერის პლატა უნდა აკმაყოფილებდეს მოთხოვნებს: უნდა შეესაბამებოდეს კომპიუტერის შიდა სტრუქტურას და უნდა ქონდეა კაბელის ტიპის შესაბამისი შემაერთებელი.

ქსელური ადაპტერის არჩევისას უნდა იყოს გათვალისწინებული:

1. კომპიუტერში არსებული მონაცემთა სალტის ტიპი - ISA,VESA, PCI ან სხვა რომელიმე, მაგალითად, ძველი კომპიუტერები 286, 386 აქვთ მხოლოდ ISA სალტე, 486-ს აქვს როგორც ISA ისევე PCI, ხოლო Pentium III_დან უკვე აღარ აქვთ ISA-სალტე).
2. ქსელის ტიპი, რომელზეც ხდება კომპიუტერის ჩართვა. თუ მაგალითად კომპიუტერს რთავთ ქსელში კოაქსიალურ კაბელზე, მაშინ აუცილებლად უნდა აირჩიოთ ქსელური ადაპტერი შესაბამისი გასართებით.
3. შეუძლია თუ არა იმუშაოს ქსელურ ადაპტერს სხვადასხვა ოპერაციულ სისტემასთან.

ქსელური ადაპტერის პლატა თავისი პარამეტრებით გავლენას ახდენს ქსელის მწარმოებლურობაზე. პლატის დაბალი სიჩქარის დროს, დაბალია ქსელში მონაცემთა გადაცემის სიჩქარეც და დაბალია ქსელის წარმადობაც. ქსელში მონაცემთა გადაცემის სიჩქარეს განსაზღვრავენ შემდეგი ფაქტორები:

პირდაპირი წვდომა მეხსიერებასთან. ქსელური ადაპტერის პლატის ბუფერიდან, მონაცემები, პირდაპირ გადაეცემა კომპიუტერის ოპერატიულ მეხსიერებას ცენტრალური პროცესორის ჩარევის გარეშე.

ადაპტერის დაყოფადი მეხსიერება. ქსელური ადაპტერის პლატას აქვს საკუთარი ოპერატიული მეხსიერება, რომელსაც იყენებს კომპიუტერთან ერთად. კომპიუტერი ამ მეხსიერებას, აღიქვამს როგორც საკუთარს.

დაყოფადი სისტემური მეხსიერება. ქსელური ადაპტერის პლატის პროცესორი, მონაცემთა დამუშავებისათვის იყენებს კომპიუტერის მეხსიერების ნაწილს.

სალტის მართვა. ქსელური ადაპტერის პლატას დროებით გადაეცემა სალტის მართვა. შესაბამისად, ცენტრალური პროცესორის გვერდის ავლით პლატა მონაცემებს გადასცემს კომპიუტერის სისტემურ მეხსიერებას. ამ პრინციპით იზრდება როგორც კომპიუტერის წარმადობა, რადგან პროცესორს შეუძლია ამ დროს სხვა ამოცანები ამოხსნას, ასევე ქსელის წარმადობა. მაგრამ ასეთი პლატები ძვირი ღირს.

მონაცემთა ბუფერში მოთავსება. ქსელური პლატის უმრავლესობისათვის ქსელში მონაცემთა გადაცემის სიჩქარის თანამედროვე დონე ძალიან მაღალია. ამიტომ ადაპტერში იდგმება ბუფერი (მეხსიერების მიკროსქემები). როდესაც პლატა იღებს იმაზე მეტ მონაცემებს რისი დამუშავებაც შეუძლია, ბუფერი ინახავს მათ, მანამ სანამ არ მოხდება მათი დამუშავება. ბუფერი ამაღლებს პლატის წარმადობას და მთლიანად სისტემის მოქნილობას.

ჩაშენებული მიკროპროცესორი. თუ ქსელური ადაპტერის პლატას ასეთი მირკოპროცესორი აქვს მას არ სჭირდება კომპიუტერის დახმარება მონაცემების დამუშავებისათვის. ასეთი მიკროპროცესორები ზრდიან ქსელური ოპერაციების შესრულების სიჩქარეს.

უგამტარო და ოპტიკურბოჭკოვანი ქსელებისათვის დამუშავებულია სპეციალიზირებული ქსელური ადაპტერის პლატები.

თავი 2. *ღია სისტემების ურთიერთქმედების მოდელები (OSI)*

2.1. *OSI მოდელის ურთიერთქმედების დონეები*

პროტოკოლი ესაა შეთანხმება ორ ურთიერთდაკავშირებულ ობიექტს შორის, ამ შემთხვევაში ქსელში ჩართულ ორ კომპიუტერს შორის. ქსელში მუშაობა ესაა მონაცემთა გადაცემა ერთი კომპიუტერიდან მეორეზე. ამ პროცესში შეიძლება რამდენიმე საკითხის გამოყოფა:

მონაცემთა გამოცნობა, მონაცემთა დაყოფა მმართველ ბლოკებად, თვითოეული ბლოკისათვის ინფორმაციის დამატება, რათა მივუთითოთ მონაცემთა მდებარეობა და მივუთითოთ მიმღები, დავუმატოთ სინქრონიზაციის და შეცდომების შესწორების შესახებ ინფორმაცია, მოვათავსოთ მონაცემები ქსელში და გავგზავნოთ ისინი მითითებული მისამართით. ამ ოპერაციების შესრულება ხდება პროტოკოლებით. არ არის აუცილებელობა, რომ ქსელში გამოყენებული პროტოკოლები იყოს სტანდარტული, მაგრამ ქსელების ექსპლოატაციის პრაქტიკაში მიღებულია რომ ისინი იყვნენ სტანდარტულნი. როგორიცაა საერთაშორისო სტანდარტი.

80-იან წლებში საერთაშორისო ორგანიზაციებმა ISO, ITU-T-მა დაამუშავეს OSI (Open System Interconnection) – ღია სისტემების ურთიერთქმედების მოდელი, რომელმაც დიდი როლი ითამაშა ქსელების განვითარებაში. ღია სისტემა შეიძლება ვუწოდოთ ნებისმიერ სისტემას - კომპიუტერს, ქსელს, ოპერაციულ სისტემას, პროგრამულ პაკეტს და სხვა აპარატურულ და პროგრამულ კომპონენტებს, რომლებიც აგებულია ღია სპეციფიკაციის შესაბამისად. სპეციფიკაციის ქვეშ იგულისხმება აპარატურული ან პროგრამული კომპონენტების ფუნქციონირების, სხვა კომპონენტებთან ურთიერთქმედების ფორმალური აღწერა. ღია სპეციფიკაციის ქვეშ იგულისხმება გამოქვეყნებული, ყველასთვის მისაწვდომი სპეციფიკაციები.

OSI მოდელში ურთიერთქმედების საშულებები იყოფა შვიდ დონედ: გამოყენებითი, წარმოდგენითი, სეანსური, სატრანსპორტო, ქსელური, არხული 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 ცხრილში N 2. რეალური კავშირი კი მხოლოდ მეზობელ დონეებს შორის ხორციელდება.

OSI -ს მოდელი აღწერს მხოლოდ სისტემურ საშუალებებს, რომელიც ოპერაციული სისტემისა და სისტემური უტილიტების მიერ არის რეალიზებული.

ცხრილი N 2 გადამცემ და მიმღებ კომპიუტერებს შორის OSI -ის მოდელების ურთიერთქმედება

OSI მოდელი			
	მონაცემების ერთეული	**დონე**	**ფუნქცია**
ჰოსტის დონეები	მონაცემები	პროგრამული	ქსელის მიწოდება პროგრამისათვის
		პრეზენტაციის	მონაცემების შიფრაცია და წარდგენა
		სესიის	კვანძთაშორისი კავშირი
	სეგმენტები	ტრანსპორტული	კავშირი ორ უკიდურეს წერტილს შორის და საიმედოობა
მატარებელი დონეები	პაკეტები	ქსელური	გეზის განსაზღვრა და ლოგიკური მისამართები (IP)
	კადრები	მონაცემთა არხი	ფიზიკური მისამართები (MAC და LLC)
	ბიტები	ფიზიკური	მატარებელი ხაზი, სიგნალი და ორობითი გადაცემა

წყარო: (კვლევითი მასალები)

დაუშვათ მივმართეთ მოთხოვნით გამოყენებით დონეს. ამ მოთხოვნის საფუძველზე გამოყენებითი დონის პროგრამული უზრუნველყოფა ჩამოაყალიბებს (აფორმირებს) სტანდარტული ფორმატის შეტყობინებას. ჩვეულებრივ ეს მოთხოვნა შედგება სათაურისაგან და მონაცემთა ველისაგან. სათაური შეიცავს სამომსახურო ინფორმაციას, რომელიც აუცილებლად უნდა გადაიცეს ქსელში მიმღები კომპიუტერის გამოყენებითი დონისათვის, რათა მიმღებ კომპიუტერს შეატყობინოს თუ რა სამუშაო უნდა შეასრულოს მან. მონაცემთა ველი შეიძლება იყოს ცარიელი ან შეიცავდეს რაიმე ინფორმაციას, მაგალითად, თუ რა უნდა ჩაიწეროს მიმღები კომპიუტერის ფაილში. ეს ინფორმაცია რომ დანიშნულებისამებრ გადაიცეს, ქვედა დონეების მიერ უნდა შესრულდეს ბევრი ამოცანა. გამოყენებითი დონე ჩამოაყალიბებს შეტყობინებას და გადასცემს მას წარმოდგენით

დონეს. წარმოდგენითი დონიდან მიღებული ინფორმაციის საფუძველზე პროტოკოლები ასრულებს საჭირო მოქმედებას და უმატებს შეტყობინებას საჭირო ინფორმაციას, რომელიც მიმღები კომპიუტერის შესაბამისი დონისთვისაა განკუთვნილი. შედეგად მიღებული ინფორმაცია გადაეცემა სეანსის დონეს, რომელიც დაუმატებს საჭირო ინფორმაციას და ა,შ. სანამ შეტყობინება არ მიაღწევს ქვედა ფიზიკურ დონეს, რომელიც გადაცემს მას ქსელში. ქსელში გადაცემულ ინფორმაციას კომპიუტერი-ადრესატი მიიღებს, ფიზიკურ დონეზე გაანალიზებს მისთვის განკუთვნილ შეტყობინებას (message), მოხსნის გამომგზავნი კომპიუტერის ფიზიკურ დონეზე დამატებულ ინფორმაციას და გადაცემს მის ზემდგომ დონეს. როდესაც პაკეტი მიაღწევს გამოყენებით დონეს მთლიანად ექნება მოცილებული სამისამართო ინფორმაცია და მონაცემები მიიღებს საწყის სახეს.

2.2. *ქსელზე დამოკიდებული და ქსელისგან დამოუკიდებელი დონეები*

OSI -ს მოდელების ყველა ფუნქცია მიეკუთვნება შემდეგ ორ ჯგუფს: ქსელის ტექნიკური რეალიზაციის ფუნქციებს ან დანართებთან მუშაობაზე ორიენტირებულ ფუნქციებს. ფიზიკური, არხული და ქსელური დონე ქსელზე დამოკიდებულნი არიან ე.ი ამ დონის პროტოკოლები განსაზღვრავენ ქსელების ტექნიკურ რეალიზაციას და გამოყენებულ საკომუტაციო მოწყობილობებს.

გამოყენებითი, წარმოდგენითი და სეანსის დონის პროტოკოლები ორიენტირებულნი არიან დანართებზე და თითქმის არ არიან დამოკიდებულნი ქსელის აგების ტექნიკურ საშუალებებზე, როგორც არ უნდა მოხდეს ქსელის ტექნიკური საშუალებების შეცვლა (აპარატურის ან სხვა ქსელური ტექნოლოგიის გამოყენება) ამ დონეების პროტოკოლებში იგი არ აისახება.

სატრანსპორტო დონე არის შუალედური, იგი გამოყოფს ქვედა დონეების ფუნქციონირების ყველა დეტალს ზედა დონეებისგან.

კომპიუტერი, რომელზეც ქსელური ოპერაციული სისტემაა დაყენებული ურთიერთობს სხვა კომპიუტერთან შვიდივე დონის პროტოკოლებით. ამ ურთიერთობებს კომპიუტერები ახორციელებენ სხვადასხვა საკომუტაციო მოწყობილობებით: კონცენტრატორებით, ხიდებით, მოდემებით, კომუტატორებით, მარშრუტიზატორებით. მოწყობილობის ტიპიდან გამომდინარე იგი შეიძლება მუშაობდეს ფიზიკურ დონეზე (გამმეორებელი), ფიზიკურ ან არხის დონეზე (ხიდი), ფიზიკურ, არხულ და ქსელურ (ზოგჯერ სატრანსპორტო) დონეზე (მარშრუტიზატორი).

როგორც ზემოთ ავღნიშნეთ ქსელის ფუნქციონირების საფუძველს წარმადგენს ორი კომპიუტერის კავშირის უზრუნველყოფა ინფორმაციის გაცვლის მიზნით. ამ მიზანს ემსახურება წარმოდგენილი ფენოვანი პროგრამული და აპარატურული უზრუნველყოფა. A-დან B-მომხმარებლისათვის ინფორმაციის გადაცემის დროს, მან უნდა გაიაროს გადამცემის ყველა დონე ზემოდან ქვემოთ და მიმღების ყველა დონე, ქვემოდან ზემოთ. ამ დროს, ერთ კომპიუტერში ყოველი დონე ისე მუშაობს, თითქოს ის უშუალოდ არის დაკავშირებული მეორე კომპიუტერის ასეთივე დონესთან. მაგრამ ეს კავშირი არის მხოლოდ ლოგიკური ანუ ვირტუალური. სინამდვილეში კავშირი მყარდება მხოლოდ ერთი კომპიუტერის მომიჯნავე დონეებს შორის. მაგალითად, სესიის დონე უშუალოდ ურთიერთქმედებს მხოლოდ წარდგენით და სატრანსპორტო დონეებთან. ყოველი დონის პროგრამული უზრუნველყოფა ასრულებს ამ დონის პროტოკოლით განსაზღვრულ ქსელურ ფუნქციას ან ფუნქციებს. მომიჯნავე დონეების ურთიერთქმედება ხორციელდება

შესაბამისი ინტერფეისით. ინტერფეისი განსაზღვრავს იმ მომსახურების ფორმას და მისი მიღების საშუალებას, რომელსაც ქვედა დონე სთავაზობს ზედა დონეს.
ყოველ დონეზე სრულდება რამოდენიმე მომსახურება (რამოდენიმე ოპერაცია), რომლებიც ამზადებენ მონაცემის გადაცემას მეორე კომპიუტერში ქსელის გავლით. ამიტომ ერთი კომპიუტერის ყოველ დონეს ,,ჰგონია", რომ ის უშუალოდ ურთიერთქმედებს მეორე კომპიუტერის ასეთივე დონესთან. სინამდვილეში ეს ურთიერთქმედება წინასწარ განსაზღვრულია პროტოკოლებით, რომლებშიც ჩამოყალიბებულია ერთნაირი დონეების მოქმედების წესები და გამოყენებული სიდიდეების პარამეტრები.
ამგვარად, ერთნაირ დონეთა ურთიერთქმედება განისაზღვრება პროტოკოლებით, მომიჯნავეთა კი - ინტერფეისით.
ქსელში გადაცემის წინ მონაცემები იყოფა პაკეტებად. პაკეტი არის ინფორმაციის ერთეული, რომელიც გადაიცემა ქსელის მოწყობილობებს შორის, როგორც ერთი მთლიანი მონაცემი. გადამცემი კომპიუტერის ყოველ დონეზე, ზემოდან ქვემოთ, პაკეტს ემატება გარკვეული სამისამართო ინფორმაცია მისი ფორმატირებისათვის, რაც აუცილებელია მონაცემთა გადაცემისათვის ქსელში. მიმღებ კომპიუტერში პაკეტი გადის ყველა დონეს საწინააღმდეგო მიმართულებით ქვევიდან ზევით. ყოველ დონეზე პროგრამული უზრუნველყოფა კითხულობს გამგზავნი კომპიუტერის ამავე დონის სამისამართო ინფორმაციას. შემდგომ ჩამოაშორებს ამ დონეზე დამატებულ ინფორმაციას და გადასცემს შემდგომ დონეს. როცა პაკეტი მიაღწევს გამოყენებით დონეს, ყველა სამისამართო ინფორმაცია მოშორებული იქნება და პაკეტი მიიღებს პირველყოფილ სახეს. შესაბამისად, ქსელური მოდელის არცერთ დონეს არ შეუძლია ინფორმაციის უშუალოდ გადაცემა მეორე კომპიუტერის შესაბამის დონეზე. ინფორმაციის (პაკეტის) გადაცემა ხდება მხოლოდ ფიზიკური გარემოს (კაბელის) საშუალებით. მაგალითად, ქსელური დონის სამისამართო ინფორმაცია A კომპიუტერიდან არხის და ფიზიკური დონეების გავლით ჩამოდის ქსელურ კაბელამდე, მისი გავლით მოხვდება B კომპიუტერში, სადაც ფიზიკური და არხის დონეების საშუალებით მიაღწევს მიმღების ქსელურ დონეს. ეს ინფორმაცია იქნება მისამართის და შეცდომების კონტროლის, რომელსაც ამ დონის პროტოკოლის შესაბამისად აღიქვამს B კომპიუტერი.
შვიდდონიანი ეტალონური მოდელის პროტოკოლების იერარქიაში ყველაზე დაბალი, ფიზიკური დონე არის საბაზო დონე. ფიზიკური დონე ქმნის ზედა, არხის დონის ინტერფეისს. ყოველი შემდგომი დონე კი წარმადგენს მის ზედა დონის უზრუნველყოფას.

ქსელის ძირითადი დანიშნულების განხორციელება ხდება ყველაზე ზედა, გამოყენებით დონეზე. როგორც ზემოთ იყო აღნიშნული, ეს დონე არის ქსელურ მომსახურებასთან გამოყენებითი პროცესების შეღწევის ფანჯარა. იგი უზრუნველყოფს მომხმარებლის (ქსელის კლიენტის) შეღწევას ფაილების გადაცემის პროგრამულ უზრუნველყოფასთან, მონაცემთა ბაზების და ელექტრონული ფოსტის გამოყენებას და სხვა მრავალ მომსახურებას.

პროტოკოლების სტეკი (protocol stack) - ეს არის პროტოკოლების კომბინაცია. ყოველი დონე განსაზღვრავს პროტოკოლების ნაკრებს მოცემული დონის კავშირის ფუნქციის განსახორციელებლად. ყოველ დონეს შეესაბამისება საკუთარი წესების ნაკრები ანუ პროტოკოლები.

ცხრილი N 3 ქსელისგან დამოუკიდებელი დონეები

გამოყენებითი დონე	ინიციალიზაცია ან მოთხოვნის მიღება
პროგრამული დონე	დაფორმატიზების, გარდასახვის და დაშიფვრის ინფორმაციის დამატება პაკეტში
სეანსის დონე	ტრაფიკის შესახებ ინფორმაციის დამატება პაკეტის გაგზავნის მომენტის ჩვენებით
ტრასპორტირების დონე	შეცდომის დამუშავების ინფორმაციის დამატება
ქსელური დონე	სამისამართო და გადაცემული პაკეტების თანმიმდევრობაში მოცემული პაკეტის ადგილის ინფორმაციის დამატება
არხის დონე	შეცდომების შემოწმების ინფორმაციის დამატება და მონაცემთა მომზადება ფიზიკურ შეერთებაში გადასაცემად
ფიზიკური დონე	პაკეტის, როგორც ბიტების ნაკადის გადაცემა არხში

წყარო : (კვლევითი მასალები)

პროტოკოლების სტეკის ქვედა სამი დონე აღწერს სხვადასხვა მწარმოებლის მიერ დამზადებული მოწყობილობების (ქსელური ადაპტერის პლატის, კონცენტრატორის, ხიდის, მარშრუტიზატორის და სხვ.) ურთიერთქმედების წესებს. ზედა დონეები კი აღწერენ კავშირის სეანსების ჩატარების წესებს და გამოყენებითი ნაწილის ინტერპრეტაციას (რეალიზაციას).

სტეკების პროტოკოლები ასრულებენ თავისი დონისათვის დამახასიათებელ სპეციფიკურ სამუშაოს. ქსელის კომუნიკაციური ამოცანების შესაბამისად პროტოკოლები იყოფა სამ ტიპად: გამოყენებითი; სატრანსპორტო; ქსელური.

პროტოკოლების სტანდარტულ მოდელებს წარმოადგენენ:

1. ISO/OSI პროტოკოლების ნაკრები;
2. IBM System Network Architecture (SNA);
3. Digital DEC net;
4. Novell Net Ware;
5. Apple Apple Talk;
6. ინტერნეტის პროტოკოლების ნაკრები TCP/IP.

გამოყენებითი პროტოკოლები მუშაობენ OSI მოდელის ზედა სამ დონეზე. ისინი უზრუნველყოფენ გამოყენებითი ნაწილების ურთიერთქმედებას და მათ შორის მონაცემთა გაცვლას.

სატრანსპორტო პროტოკოლები უზრუნველყოფენ კავშირის სეანსებს კომპიუტერებს შორის და ქმნიან მათ შორის მონაცემთა გაცვლის საიმედო გარანტიას.

ქსელური პროტოკოლები უზრუნველყოფენ კავშირების მომსახურებას, რომლებიც ქვედა სამი დონით არის განსაზღვრული. ეს პროტოკოლები მართავენ მონაცემთა შემდეგ ტიპებს: ადრესაციას, მარშრუტიზაციას, შეცდომების შემოწმებას და მოთხოვნებს განმეორებითი გადაცემის შესახებ.

სტანდარტიზაციის ორგანიზაციების მიერ დამუშავებულია OSI მოდელის დონეების შესაბამისი პროტოკოლები.

2.3. პროტოკოლები

როგორც ზემოთ იყო აღნიშნული პროტოკოლი (Protokols) ესაა პროცედურების და წესების ერთობლიობა, რომელთა საშუალებითაც ხორციელდება კომპიუტერებს შორის ურთიერთკავშირი. პროტოკოლები ფუნქციონირებენ OSI მოდელების სხვადასხვა დონეზე. მაგალითად, თუ პროტოკოლი მუშაობს ფიზიკურ დონეზე ეს ნიშნავს, რომ ის უზრუნველყოფს ქსელურ ადაპტერზე პაკეტის გავლას და მის მოხვედრას ქსელურ კაბელში. ერთდროულად შეიძლება მუშაობდეს რამდენიმე პროტოკოლი. მაშინ ამბობენ რომ ისინი ქმნიან პროტოკოლების სტეკს.

ტექნიკური თვალსაზრისით ქსელში მონაცემთა გადაცემა იყოფა რამდენიმე თანმიმდევრულ ბიჯად, თითოეულ ამ საკითხს თავისი წესები და პროცედურა ან პროტოკოლი შეესაბამება. ამ ბიჯების თანმიმდევრობა მკაცრად დაცულია და ქსელში ყველა კომპიუტერზე ერთნაირია. გადამცემ კომპიუტერზე ეს მოქმედება სრულდება ზევიდან ქვევით, მიმღებზე კი პირიქით ქვევიდან – ზევით.

გადამცემი კომპიუტერი პროტოკოლების შესაბამისად ასრულებს შემდეგ მოქმედებებს:

1. მონაცემებს ყოფს ბლოკებად, პაკეტებად, რომლებთანაც მუშაობს პროტოკოლი.
2. პაკეტებს უმატებს მიმღები კომპიუტერის მისამართებს.
3. ამზადებს მონაცემებს ქსელური ადაპტერის საშუალებით კაბელში გადასაცემად.

კომპიუტერი მიმღები პროტოკოლების შესაბამისად ასრულებს იგივე მოქმედებას ოღონდ შებრუნებული თანმიმდევრობით:

1. იღებს მონაცემებს კაბელიდან.
2. ქსელური ადაპტერის საშუალებით მიღებულ პაკეტს გადასცემს კომპიუტერს.
3. კომპიუტერი მოხსნის გადამცემის მიერ დამატებულ ყველა ინფორმაციას.
4. პაკეტიდან მიღებულ მონაცემებს ათავსებს ბუფერში, რათა მოახდინოს საწყის ბლოკებად გაერთიანება.
5. მონაცემთა ბლოკს გადასცემს დანართს შესაბამის ფორმატში.

კომპიუტერმა-გადამცემმა და კომპიუტერმა-მიმღებმა თითოეული მოქმედება უნდა შეასრულოს ერთნაირად, რათა ქსელიდან მიღებული მონაცემები დაემთხვას საწყისს მონაცემებს. თუ მონაცემთა გადაცემა ხდება ერთი ლოგიკური ქსელიდან მეორეში ერთ-ერთი შესაძლო მარშუტით ვუწოდებთ მარშრუტირებადს. ხოლო პროტოკოლებს, რომლებიც უზრუნველყოფენ მონაცემთა გადაცემას ქსელებს შორის, ქვიათ მარშრუტირებადი პროტოკოლები. ასე რომ სამარშრუტო პროტოკოლებს გამოყენება შეიძლება რამდენიმე ლოკალური ქსელის გასაერთიანებლად გლობალურ ქსელში.

გამოყენებითი პროტოკოლები. გამოყენებითი პროტოკოლები OSI-ის მოდელის ზედა დონეზე მოქმედებენ. ისინი უზრუნველყოფენ დანართების ურთიერთობას და მათ შორის მონაცემთა გაცვლას.

ყველაზე უფრო მეტად არის გავრცელებული შემდეგი პროტოკოლები:

FTAM (File Transfer Access and Management) – OSI-ის ფაილებთან წვდომის პროტოკოლი;

X.400 – CCITT-ის პროტოკოლი ელექტრონული ფოსტის საერთაშორისო გაცვლისათვის;

X.500 – CCITT-ის ფაილებისა და კატალოგების სამსახურის პროტოკოლი რამოდენიმე სისტემაზე;

SMTP (Simple Mail Transfer Protocol) – Internet გლობალური ქსელის ელექტრონული ფოსტის გაცვლის პროტოკოლი;

FTP (File Transfer Protocol) – Internet გლობალური ქსელის ფაილების გაცვლის პროტოკოლი;

SNMP (Simple Network Management Protocol) – ქსელის მონიტორინგის, ქსელური კომპონენტების მუშაობის კოტროლისა და მათი მართვის პროტოკოლი;

Telnet – Internet გლობალური ქსელის პროტოკოლი დაშორებულ სერვერებზე რეგისტრაციისა და მასზე მონაცემების დამუშავებისათვის;

Microsoft SMBs (Server Message Blocks, სერვერის შეტყობინებათა ბლოკები) და კლიენტური გარსები ანუ Microsoft ფირმის რედირექტორიები;

NCP (Novell NetWare Core Protocol) და კლიენტური გარსები ანუ Novell ფირმის რედირექტორიები;

სატრანსპორტო პროტოკოლები. სატრანსპორტო პროტოკოლები უზრუნველყოფენ კომპიუტერებს შორის კავშირს და მონაცემთა დაუმახინჯებლად გაცვლას. უფრო მეტად პოპულარულია შემდეგი სატრანსპორტო პროტოკოლები:

TCP (Transmission Control Protocol) – TCP/IP პროტოკოლების ნაკრების ნაწილი, რომელიც უზრუნველყოფს მონაცემების გარანტირებულ მიწოდებას, დაყოფილს

თანმიმდევრულად ფრაგმენტებად;
SPX – IPX/SPX (Internetwork Packet Exchange/Sequential Packet Exchange) პროტოკოლების ნაკრების ნაწილი, რომელიც უზრუნველყოფს მონაცემების გარანტირებულ მიწოდებას, დაყოფილს თანმიმდევრულად ფრაგმენტებად, შემოთავაზებულს ყოველლ კომპანიის მიერ;
NWLink – Microsoft კომპანიის IPX/SPX პროტოკოლის რეალიზაცია;
NetBEUI – (NetBIOS Extended User Interface, NetBIOS-ის გაფართოებული ინტერფეისი) – ამყარებს სეანსებს (Netbios)–ის კომპიუტერებს შორის და სთავაზობს ტრანსპორტულ მომსახურებას ზედა დონეებს.
ქსელური პროტოკოლები. ქსელური პროტოკოლები უზრუნველყოფენ კავშირს. მათ ურთიერთობა აქვთ სამისამართო და სამარშრუტო ინფორმაციასთან, შეცდომების აღმოჩენასთან და მათ შესწორებასთან. უფრო პოპულარულია შემდეგინ პტოტოკოლები:
IP (Internet Protocol) – TCP/IP-პროტოკოლი ანუ პაკეტების არაგარანტირებული გადაცემის პროტოკოლი კავშირების დამყარების გარეშე;
IPX (Internetwork Packet Exchange) – NetWare კომპანიის პროტოკოლი ანუ პაკეტების გადაცემისა და მარშუტიზაციის არაგარანტირებული პროტოკოლი:
NWLink – Microsoft კომპანიის IPX/SPX პროტოკოლის რეალიზაცია;
NETBEUI – სატრანსპორტო პროტოკოლი, რომელიც უზრუნველყოფს მონაცემთა ტრანსპორტირების სერვისებს Netbios-ის სენსებისა და პროგრამებისათვის.
განვიხილოთ ყველაზე გავრცელებული პროტოკოლები. OSI მოდელი იყენებს ორ ძირითად მეთოდს აბონენტების ურთიეთქმედებისა ქსელში:

1. ურთიერთქმედების მეთოდი ლოგიკური კავშირის გარეშე (ანუ დეტაგრამების მეთოდი);
2. ურთიერთქმედების მეთოდი ლოგიკური კავშირის მეშვეობით.

დეიტაგრამების მეთოდი – ეს უმარტივესი მეთოდია, რომელშიც ყოველი პაკეტი განიხილება, როგორც დამოუკიდებელი ობიექტი.
ამ მეთოდის შემთხვევაში პაკეტი გადაეცემა ლოგიკური არხის გარეშე, ე.ი. საინფორმაციო პაკეტების წინასწარი გაცვლის გარეშე, რომელიც შეიცავს ინფორმაციას მიმღების მზადყოფნაზე, აგრეთვე ლოგიკური არხის ლიკვიდაციის გარეშე, ე.ი. პაკეტის გარეშე, რომელიც შეიცავს ინფორმაციას გადაცემის დამთავრების დადასტურებაზე. პაკეტი მიიღო თუ არა მიმღებმა - უცნობია (პაკეტის მიღების შემოწმება ევალება უფრო მაღალ დოენეებს).

დეიტაგრამების მეთოდის დროს აპარატურას წაეყენება მაღალი მოთხოვნები, რადგანაც მიმღები ყოველთვის მზად უნდა იყოს მიღებაზე. ამ მეთოდის უპირატესობა იმაში მდგომარეობს, რომ გადამცემი და მიმღები ფუნქციონირებენ ერთმანეთისაგან დამოუკიდებლად, ამასთანავე პაკეტები გროვდება ბუფერში, შემდგომ გადაეცემა ერთდროულად, ასევე შეიძლება გამოვიყენოთ ფართომაუწყებლობითი გადაცემა ანუ პაკეტი ერთდოულად რამოდენიმე აბონეტს გადაეცემა. ამ მეთოდის ნაკლოვანებებია – პაკეტის დაკარგვის შესაძლებლობა, აგრეთვე ქსელის გადატვირთვა პაკეტებით იმ შემთხვევაში თუ მიმღები მზად არ არის.

სტანდარტიზაციის ორგანიზაციების მიერ დამუშავებულია OSI მოდელის დონეების შესაბამისი პროტოკოლები. მაგალითად IEEE კომიტეტმა ფიზიკური დონისათვის განსაზღვრა პროტოკოლები: IEEE 802.X.

IEEE 802.X სტანდარტი. 70-იანი წლების ბოლოს განისაზღვრა ლოკალური ქსელების სტანდარტები. გამოშვებული იქნა სტანდარტი 802.X (802 ნიშნავს გამოშვების წელს და თვეს ე.ი 1980 წელი თებერვალი). 802.X სტანდარტი ადგენს ქსელის ფიზიკური კომპონენტების სტანდარტებს (კაბელური სისტემები და ქსელური ადაპტერი). იგი ეხება OSI-ს მოდელის ფიზიკურ და არხის დონეს. ფიზიკურ და არხულ დონეზე დგინდება თუ როგორ უნდა გამოიყენოს ერთდროულად ქსელი რამდენიმე კომპიუტერმა, რომ არ შეუშალონ ხელი ერთმანეთს. ეს ორი ქვედა დონე ამყარებს წესებს იმის შესახებ თუ როგორ უნდა გამოიყენოს რამოდენიმე კომპიუტერმა ქსელი ერთდროულად ისე, რომ ხელი არ შეუშალონ ერთმანეთს. ამ პროექტით არხის დონე გაიყო ორ ქვედონედ:

1. მონაცემთა ლოგიკური გადაცემა (Logical Link Control, LLC)
2. გარემოსთან კავშირის მართვა (Media Access Control, MAC).

LLC დონე უზრუნველყოფს კადრების გადაცემას კვანძებს შორის იგი ამყარებს კავშირს და წყვეტს მას.

MAC დონე უზრუნველყოფს გადაცემის გარემოსთან კავშირს, კადრების საზღვრების დადგენას, კადრებზე მისამართების ამოცნობას.

LLC და MAC დონის პროტოკოლები ურთიერთდამოუკიდებელნი არიან, MAC დონის ყოველი პროტოკოლი შეიძლება გამოყენებული იყოს LLC დონის ნებისმიერი პროტოკოლის მიერ და პირიქით.

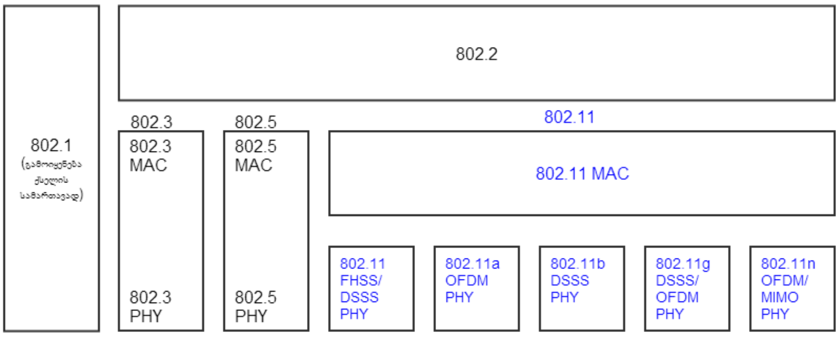
გარემოში შეღწევის მართვის ქვედონე MAC უზრუნველყოფს საერთო გარემოს კორექტულად გამოყენებას. ეს ქვედონე პირდაპირ კავშირშია ქსელურ ადაპტერთან, რომელიც დგას უშუალოდ კომპიუტერში, ადგენს მის ქსელურ მისამართს და პასუხს

აგებს ორ კომპიუტერს შორის მონაცემთა გადაცემაზე. მას შემდეგ რაც კავშირი დამყარდება, გამოიყენება უფრო მაღალი დონე LLC, რომელიც უზრუნველყოფს მონაცემთა ლოგიკური ერთეულების გადაცემას.

კატეგორიები 802.3, 802.4, 802.5 და 802.12 განსაზღვრავენ სტანდარტებს როგორც გარემოში შეღწევის მართვის ქვედონისათვის, ასევე OSI მოდელის პირველი (ფიზიკური) დონისათვის.

IEEE 802 სტანდარტს აქვს ილუსტრაცია N 14-ზე წარმოდგენილი სტრუქტურა:

ილუსტრაცია N 14 IEEE 802.X სტანდარტის სტრუქტურა.



წყარო: (კვლევითი მასალები)

Project 802-ის დამუშავების დასაწყისშივე IEEE-კომიტეტმა გამოიტანა დასკვნა, რომ ერთი სტანდარტით შეუძლებელია ქსელური არქიტექტურის და მოწყობილობების მთელი მრავალსახეობის აღწერა. ამიტომ ჩამოყალიბდა თორმეტი ქვეკომიტეტი, რომლებმაც შექმნეს შესაბამისი სტანდარტები ქსელური სფეროების მიხედვით:

802.1 _ ქსელების გაერთიანება.

802.2 _ ლოგიკური კავშირის მართვა.

ლოკალური ქსელი მარკერის გადაცემით (Token Ring)

802.3 (*Ethernet)*

ეს არის ,,ლოგიკური სალტე“, მონაცემთა გადაცემის სიჩქარე - 10/100მგ ბიტ/წმ. მონაცემები კომპიუტერებს შორის გადაიცემა კაბელის საშუალებით, მაგრამ ღებულობს მხოლოდ ერთი, ვისი მისამართითაც არის მონაცემები გაგზავნილი. CSMA/CD პროტოკოლი არეგულირებს ქსელის ტრაფიკს: გადაცემის ნებართვას იძლევა მაშინ, როცა კაბელი დაკავებული არ არის და სხვა კომპიუტერი არ გადასცემს ინფორმაციას.

802.4 (მარკერის გადაცემა)

ეს არის ,,სალტის" ტოპოლოგია, რომელიც იყენებს მარკერის გადაცემის სქემას. მარკერი გადაეცემა ერთი კომპიუტერიდან მეორეს და განსაზღვრავს იმ კომპიუტერს, რომელსაც მიენიჭება გადაცემის უფლება. ყველა კომპიუტერი ღებულობს მონაცემებს, რომლებიც ქსელში გადაიცემა, მაგრამ რეაგირებს მხოლოდ ის, ვისი მისამართითაც იგზავნება.

802.5 (*Token Ring)*

ეს არის ,,ლოგიკური რგოლი'', მონაცემთა გადაცემის სიჩქარე - 4 ან 16 მგ ბიტი/წმ. რგოლის ტოპოლოგია სრულდება MAU კონცენტრატორის საშუალებით, ამიტომ გარეგნულად გამოიყურება როგორც ვარსკვლავი. მარკერი, რომელიც გადაეცემა რგოლში, განსაზღვრავს იმ კომპიუტერს, რომელიც ჩაიჭერს მას და მიიღებს გადაცემის უფლებას.

არხის დონის IEEE პროტოკოლები ამყარებენ კავშირს გარემოში შეღწევის მართვის ქვედონეზე. გარემოში შეღწევის მართვის პროტოკოლი განსაზღვრავს კერძოდ, რომელ კომპიუტერს შეუძლია გამოიყენოს კაბელი, თუ რამოდენიმე კომპიუტერი ერთდროულად ცდილობს კაბელის ხელში ჩაგდებას. CSMA/CD პროტოკოლი აღმოაჩენს კოლიზიას (collision) და კრძალავს გადაცემას, სანამ კაბელი არ განთავისუფლდება. შემდეგ, დროის შემთხვევითი ინტერვალის ბოლოს, თითოეული კომპიუტერი კვლავ ცდილობს დაიწყოს გადაცემა.

802.6 _ ქალაქის მაშტაბის ქსელი (Metropolitan Area Network, MAN).

802.7 _ ფართოზოლიანი გადაცემის ტექნოლოგიის საკონსულტაციო საბჭო (Broadcast Technical Advisory Group).

802.8 _ ოპტობოჭკოვანი ტექნოლოგიის საკონსულტაციო ჯგუფი (Fიბერ-Oპტიც თეცჰნიცალ Aდვისორყ Gროუპ).

802.9 _ ხმის და მონაცემთა გადაცემის ინტეგრირებული ქსელი (Integrated Voice / Data Networks).

802.10 _ ქსელების უსაფრთხოება.

802.11 _ უგამტარო ქსელები.

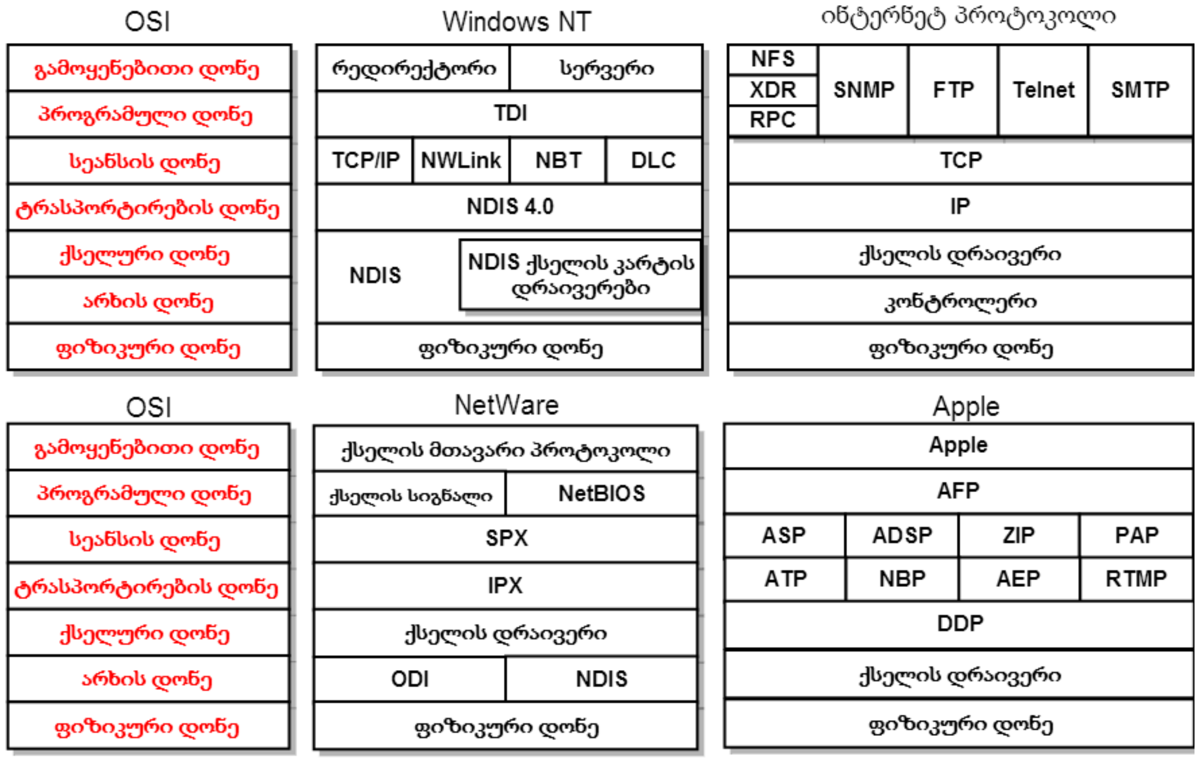
802.12 _ ლოკალური ქსელი შეღწევის პრიორიტეტის მოთხოვნით (Demand Priority Access LAN 100 baseVG _ Anylan).

ამგვარად OSI და Project 802 მოდელები განსაზღვრავენ სტანდარტულ პროტოკოლებს, რომლებსაც იყენებს პროგრამული და აპარეტურული უზრუნველყოფა. ქსელში მონაცემთა გადაცემის აპარატურის არქიტექტურა და პარამეტრები ემყარება ამ

მოდელების მოთხოვნებს. შესაბამისად საწარმოების და ფირმების პროდუქცია უშუალოდ ეფუძნება ამ სტანდარტებს მათი დამუშავების დროს.

ქვემოთ მოცემულ ილუსტრაცია N 15-ში მოცემულია ინფორმაცია OSI-ის თითოეულ დონეზე გამოყენებული პროტოკოლების შესახებ.

ილუსტრაცია N 15 OSI-ს სხვადასხვა დონეებზე გამოყენებული პროტოკოლები



წყარო: (კვლევითი მასალები)

ფართოდ გავრცელებული პროტოკოლები :

დღეისათვის ფართოდ არის გავრცელებული შემდეგი პროტოკოლები:

TCP (Transmission Control Protocol/გადაცემის მართვის პროტოკოლი) პროგრამებს შორის კომუნიკაცია. UDP (User Datagram Protocol) უბრალო კომუნიკაცია პროგრამებს შორის. IP (Internet Protocol/ინტერნეტ პროტოკოლი) კომპიუტერებს შორის კომუნიკაცია. ICMP (Internet Control Message Protocol/ინტერნეტის მართვის შეტყობინების პროტოკოლი) შეცდომებისთვის და სტატისტიკისთვის.

DHCP (Dynamic Host Configuration Protocol/დინამიკური ჰოსტის კონფიგურაციების პროტოკოლი) დინამიკური მიმართვისათვის. TCP/IP ახორციელებს მარშუტიზაციას, და ამიტომ გამოიყენება როგორც ქსელებს შორის პროტოკოლი.
X.25 არის პროტოკოლების ნაკრები პაკეტური კომუტაციის ქსელებისათვის. იგი გამოიყენებოდა დაშორებული ტერმინალების დასაკავშირებლად მეინფრეიმებთან.
OSI პროტოკოლების ნაკრები - არის პროტოკოლების სტეკი, სადაც თითოეული პროტოკოლი შეესაბამება OSI მოდელის კონკრეტულ დონეს. ნაკრები შეიცავს IEEE Project 802 სერიის მარშუტიზაციის და ტრანსპორტის პროტოკოლებს.
Apple Talk არის Appl Computer ფირმის პროტოკოლების საკუთარი სტეკი, რომელიც Appl Macintosh კომპიუტერებს აძლევს ფაილების და პრინტერების ერთობლივი გამოყენების საშუალებას.
DECnet არქიტექტურა განსაზღვრავს Ethernet ლოკალური ქსელის ბაზაზე აგებულ ქსელებს. DECnet -ს შეუძლია გამოიყენოს, როგორც TCP/IP და OSI პროტოკოლები, ასევე თავისი საკუთარი.
დრაივერი და კონტროლერი. კომპიუტერსა და პერიფერიულ მოწყობილობას შორის კავშირი ხორციელდება აპარატურული და პროგრამული საშუალებებით: პერიფერიული მოწყობილობების კონტროლერით და დრაივერით. კონტროლერი იღებს მონაცემებს ან ბრძანებებს პროცესორიდან თავის ბუფერში (მას ხშირად უწოდებენ რეგისტრს ან პორტს) შემდეგ ახორციელებს შესაბამის გარდაქმნებს და გადასცემს საჭირო მიმართულებით. დრაივერი ანუ მოწყობილობათა დრაივერი არის პროგრამული უზრუნველყოფა, რომლის საშუალებითაც კომპიუტერი მუშაობს სხვა მოწყობილობებთან. მოწყობილობა შეიძლება მიერთებული იყოს კომპიუტერთან, მაგრამ იგი ვერ იმუშავებს თუ არ იქნება დაყენებული ამ მოწყობილობის შესაბამისი დრაივერი. კომპიუტერში არსებობს თითქმის ყველა მოწყობილობის შესაბამისი დრაივერები, მაგალითად მაუსის, დისკის კონტროლერების, მულტიმედიის მოწყობილობების, პრინტერების და ა.შ. დრაივერები მოყვება შესაბამისი მოწყობილობას დისკით ან შეიძლება მათი ინტერნეტიდან გადმოწერა.
კონტროლერი ასრულებს მარტივ ბრძანებებს პერიფერიული მოწყობილობების სამართავად, ხოლო დრაივერი იყენებს ამ ბრძანებებს, რომ მოწყობილობამ შეასრულოს რაიმე ალგორითმის შესაბამისად უფრო რთული მოქმედებები. მაგალითად, პრინტერის კონტროლერი ასრულებს ბრძანებას "სიმბოლოს ბეჭვდა", "ტექსტის გადატანა", და ა.შ. პრინტერის დრაივერი ამ ბრძანებებით ბეჭდავს მთელ სტრიქონს, მთელ გვერდს და ასრულებს სხვა ოპერაციებს.

ქსელური დრაივერები უზრუნველყოფენ კავშირს ქსელურ ადაპტერსა და რედირექტორს შორის. რედირექტორი ასევე პროგრამული უზრუნველყოფის ნაწილია, რომელიც იღებს ქსელში შეტანა/გამოტანის მოთხოვნას და გადასცემს მას სხვა კომპიუტერს. დრაივერების დასაყენებლად გამოიყენება სპეციალური უტილიტა.

ქსელში მონაცემთა გადაცემა. როგორც ავღნიშნეთ ქსელში მონაცემთა გადასაცემად იქმნება პაკეტები ანუ კადრები. ესაა მონაცემთა პატარა ბლოკი, რომელთა გადაცემა ქსელში გაცილებით მარტივია და შეცდომის აღმოჩენის შემთხვევაში მისი შესწორებაც გაცილებით მარტივია.

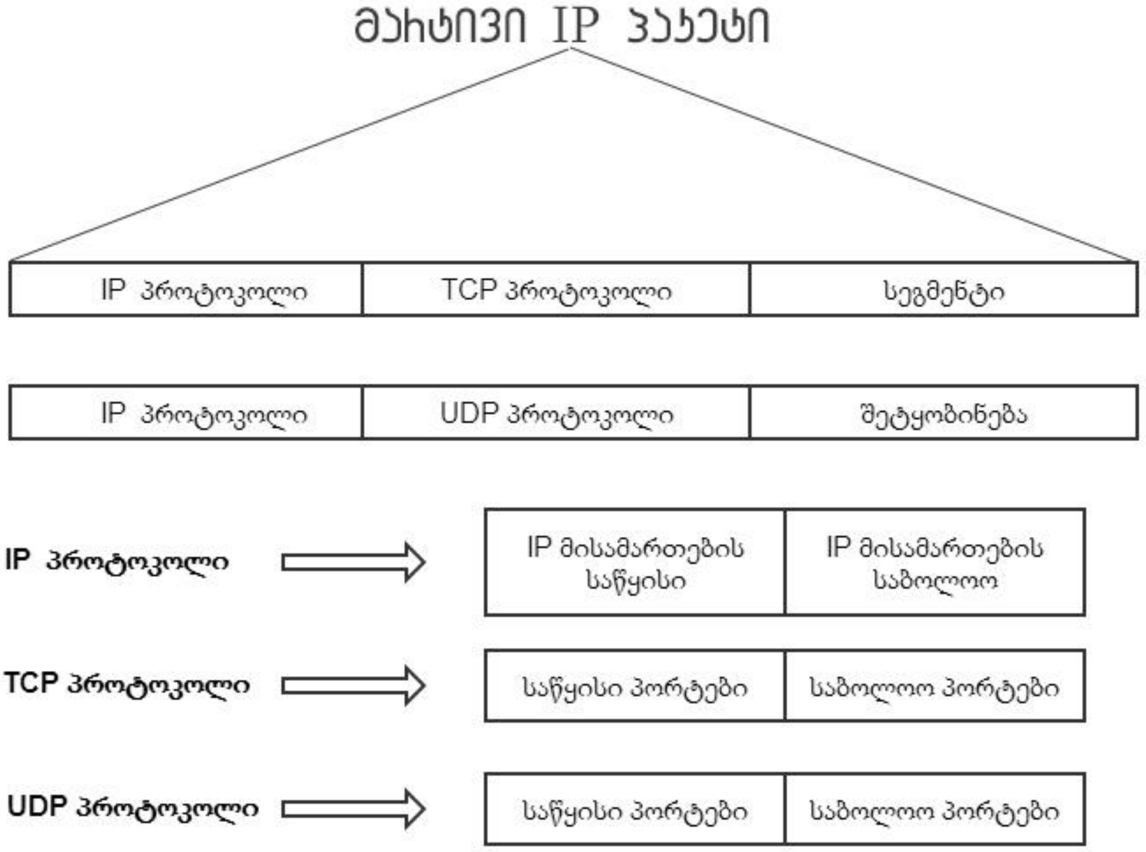
მონაცემთა პაკეტებად დაყოფის შემდეგ ქსელური ოპერაციული სისტემა უმატებს თითოეულ პაკეტს სპეციალურ მმართველ ინფორმაციას, ამ ინფორმაციის დანიშნულებაა:

1. საწყისი მონაცემების გადაცემა არა დიდი ზომის ბლოკებად
2. გადაცემული მონაცემების შეკრება (მათი მიღებისას)
3. მიღებული მონაცემების შემოწმება შეცდომების არსებობაზე.

პაკეტისთვის ქსელში გადასაცემად აუცილებელია შემდეგი სახის ინფორმაცია:

1. გადამცემი კომპიუტერის მისამართი
2. გადასაცემი მონაცემები
3. მიმღები კომპიუტერის მისამართი
4. ინფორმაცია იმის შესახებ თუ რა თანმიმდევრობითააწყოს მიმღებმა კომპიუტერმა მიღებული ინფორმაცია
5. ინფორმაცია შეცდომების შესწორებაზე.

ილუსტრაცია N 16 პაკეტის სქემა



წყარო: (კვლევითი მასალები)

პაკეტის კომპონენტები ჯგუფდება სამ განყოფილებად: სათაური, მონაცემები, და ტრეილერი. სათაური შეიცავს სიგნალს იმის შესახებ, რომ გადაიცემა მონაცემები, საწყისი კომპიუტერის მისამართს, მიმღები კომპიუტერის მისამართს და გადაცემის სინქრონიზაციის შესახებ ინფორმაციას. მონაცემები ესაა უშუალოდ გადასაცემი მონაცემები. ტრეილერი ესაა ინფორმაცია შეცდომების შესწორებისთვის. პაკეტის ფორმირება იწყება გამოყენებით დონეზე და მთავრდება ფიზიკურ დონეზე. როგორც უკვე აღნიშნული გვქონდა მას თვითოეულ დონეზე ემატება შესაბამისი ინფორმაცია.

მონაცემთა გადაცემის ძირითადი მეთოდები ქსელში. პაკეტების გადაცემის დროს თავიდან რომ იქნას აცილებული მონაცემთა დამახინჯება და კოალიზია, აუცილებელია ქსელში ტრაფიკის მართვა. ქსელში მონაცემთა გადაცემის მეთოდი – ესაა წესების ერთობლიობა, რომელიც უზრუნველყოფს იმას თუ როგორ უნდა გაიგზავნოს და როგორ

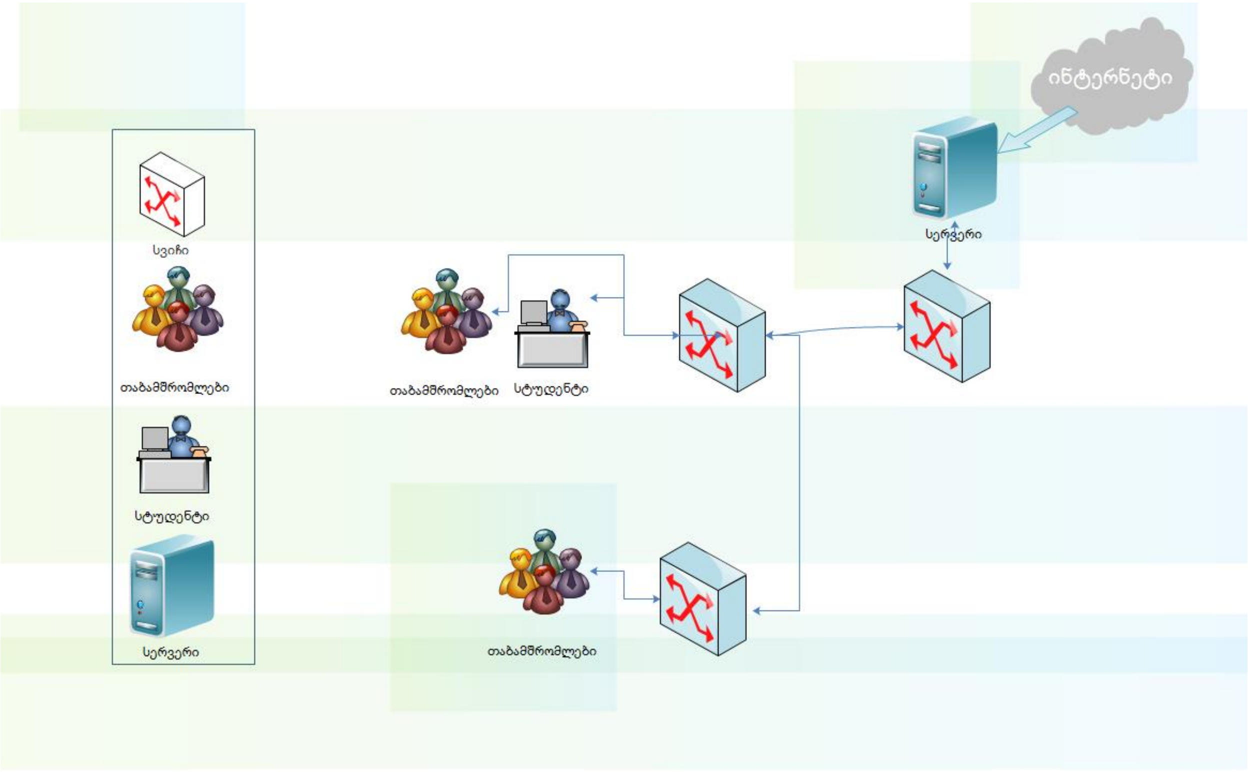
უნდა მიიღოს მონაცემები ქსელურ კაბელში. ამ წესების საშუალებით შესაძლებელი ხდება ერთდროულად რამდენიმე კომპიუტერის მიერ ქსელში მონაცემთა გადაცემა. არსებობს ქსელში მონაცემთა გადაცემთა ძირითადი მეთოდები:

1. ერთდროული შეღწევა გადამტანის (სიხშირის) კონტროლით:
2. კოლიზიის აღმოჩენით;
3. კოლიზიის თავიდან აცილებით.
4. შეღწევა მარკერის გადაცემით.
5. მონაცემების გადაცემა შეუძლია მხოლოდ იმ კომპიუტერს, რომელიც მიიღებს მარკერს.
6. შეღწევა მოთხოვნის პრიორიტეტით.

თავი 3. საქართველოს უნივერსიტეტის კომპიუტერული ქსელის განვითარების დინამიკა

საქართველოს უნივერსიტეტში სტუდენტების მატებასთან ერთად ყოველწლიურად ვითარდებოდა კომპიუტერული ქსელის სტრუქტურა. მაგალითისთვის, 2005 წელს იყო მცირე ზომის ქსელი, სადაც მომხმარებლის რაოდენობა არ აღემატებოდა 40 მომხმარებელს, დღეისათვის კი მომხმარებლების რაოდენობა აჭარბებს 600-ს. თანმიმდევრობით განვიხილოთ საქართველოს უნივერსიტეტის კომპიუტერული ქსელის ტოპოლოგიის განვითარება და მისი უსაფრთხოების უზრუნველყოფის საშუალებები.

ილუსტრაცია N 17 საქართველოს უნივერსიტეტის კომპიუტერული ქსელის 2005 წლის ტოპოლოგია



წყარო: (კვლევითი მეთოდები)

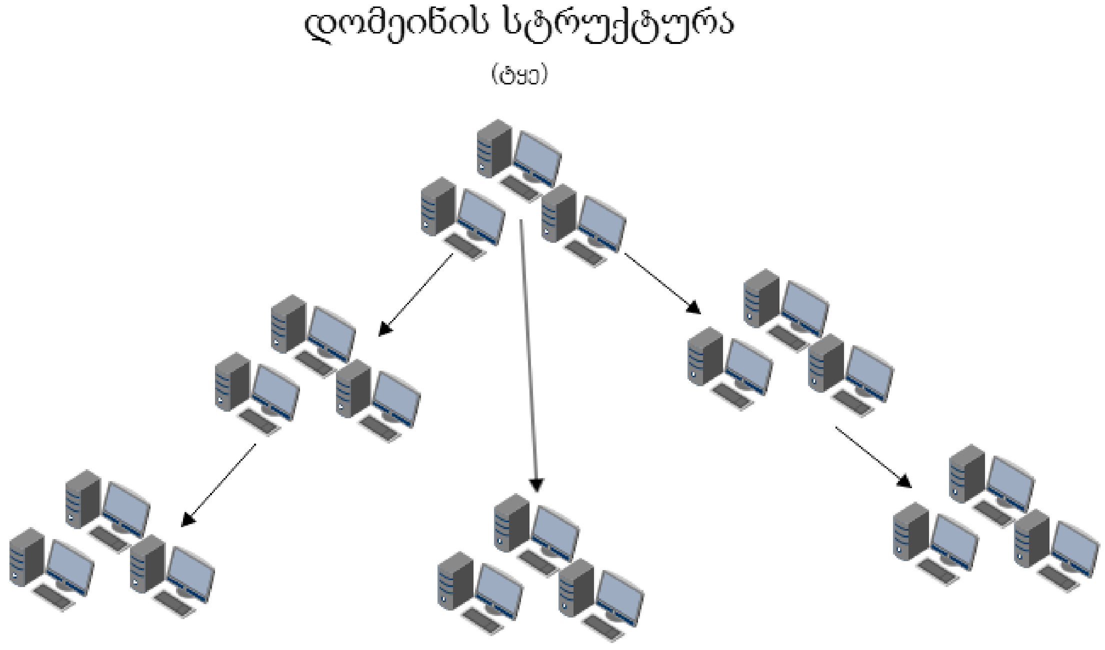
2005 წლისთვის ქსელის ტოპოლოგია აწყობილი იყო სერვერ-კლიენტის რეჟიმში, Windows Server 2003_ის დახმარებით.

Windows Server 2003_ის დახმარებით გვქონდა გამართული Active Directory, რაც ხელს უწყობდა ქსელის საიმედოობას.

რა არის Active Directory(AD)?

AD არის Microsoft-ის ნაწარმი, რომელიც შეიქმნა ქსელური სერვისების უზრუნველსაყოფად. იყენებს ერთ საერთო მონაცემთა ბაზას, სადაც ინახება ინფორმაცია კომპიუტერების, რომლებიც არიან გაწევრიანებული ერთ დომეინში.
AD-ის სტრუქტურას მოიხსენებენ როგორც Forest. იგი მოიცავს ყველა ობიექტს, რომლის ქვემოთ არის ერთი ან რამოდენიმე ხე, რომლებიც არიან დომეინები.

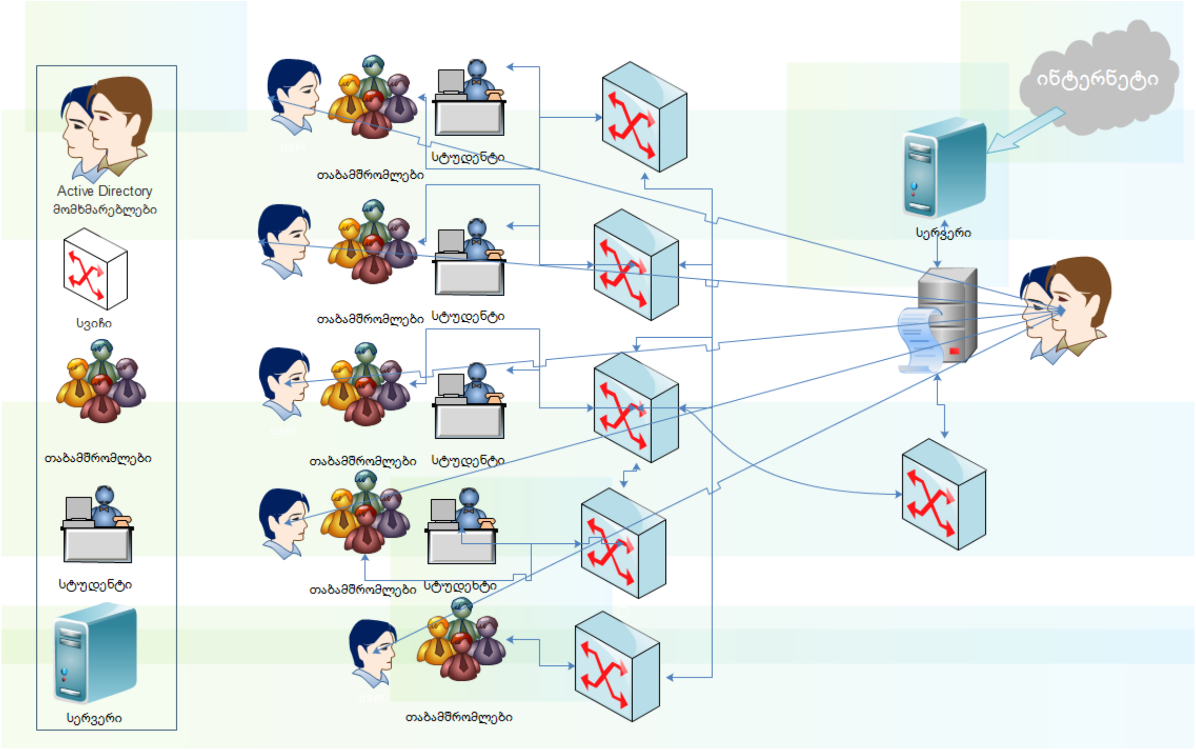
ილუსტრაცია N 18 Domain Forest სტრუქტურა



წყარო: (კვლევითი მასალები)

AD-ის სტრუქურის მთავარ ნაწილს წარმოადგენს დომენი. დომენის სახელის დარქმევაში შეზღუდული არ ვართ, შეგვიძლია დავარქვათ ინტერნეტში გავრცელებული სახელებიც (blogger.com, news.com და ა.შ.)
რეკომენდირებულია ერთ დომეინში შევქმნათ მრავალი ორგანიზაციული ერთეული და ამის მეშვეობით მთლიანი სტრუქტურა გავხადოთ მდგრადი და მოქნილი.

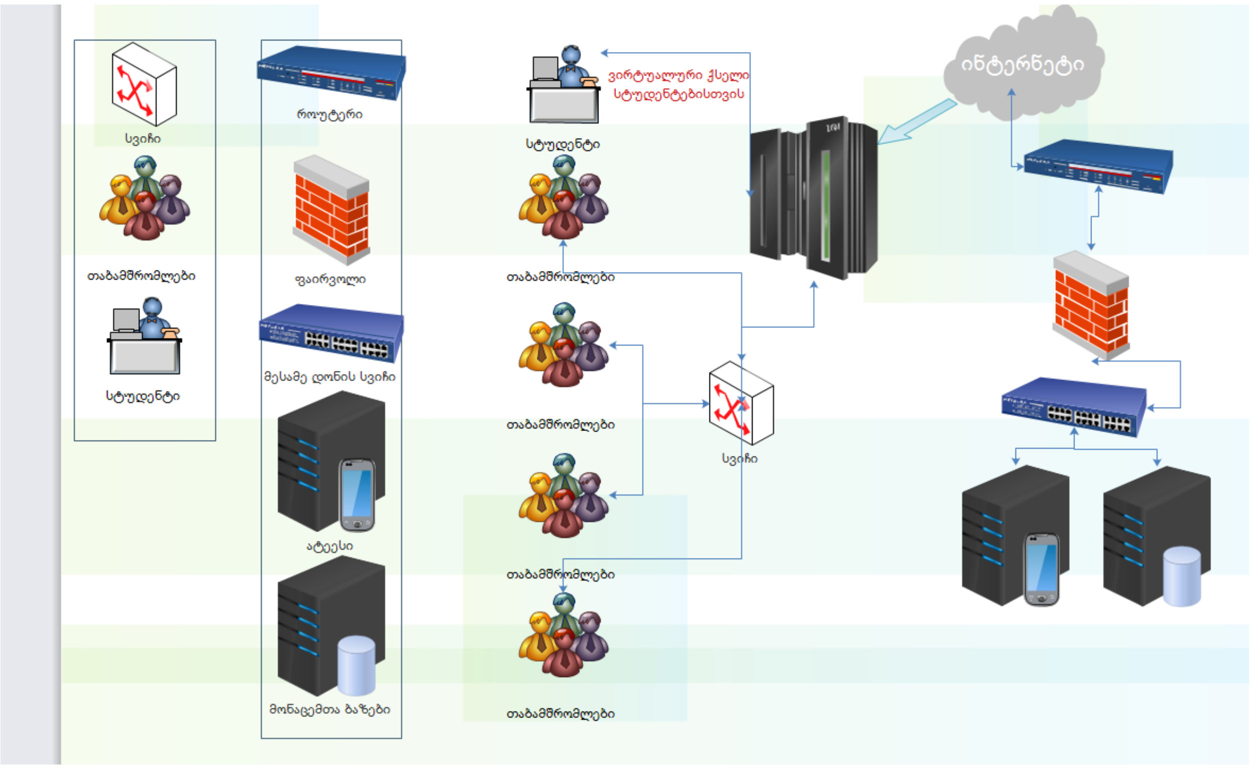
ილუსტრაცია N 19 კომპიუტერული ქსელის სტრუქტურა



წყარო: (კვლევითი მასალები)

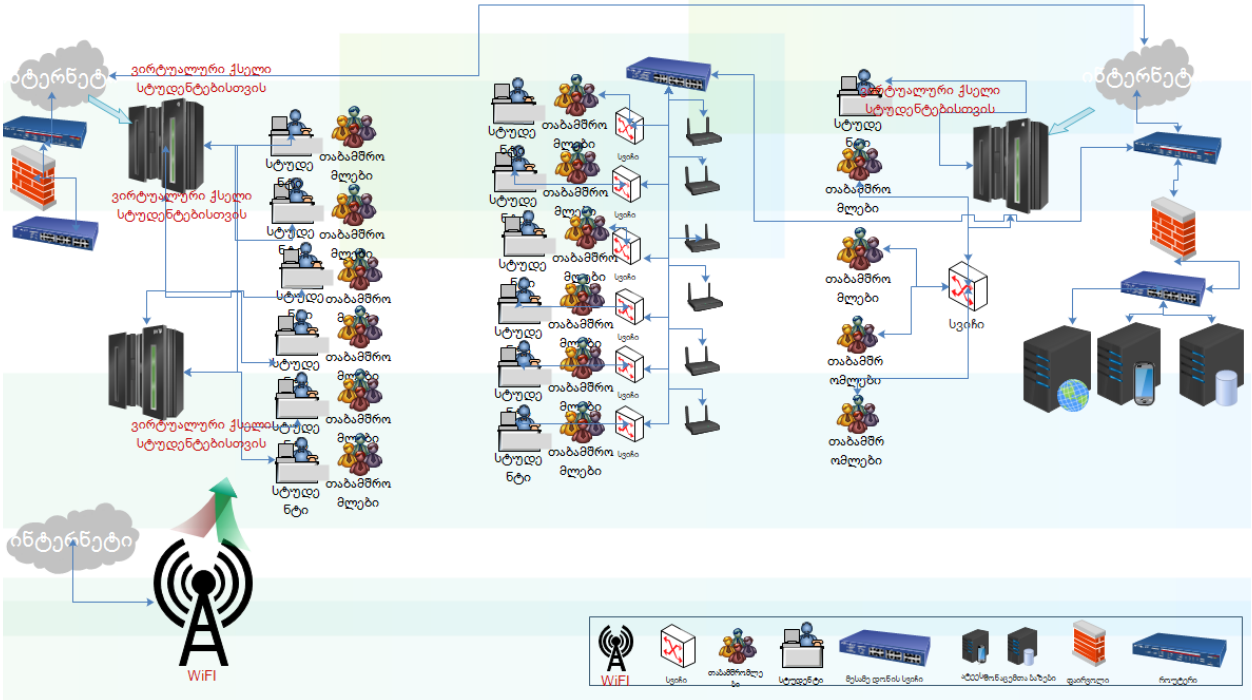
2007-2008 წლებში მოხდა კომპიუტერული ქსელის თავიდან გადაწყობა CISCO-ს აპარატურაზე, როგორიცაა ROUTER, SWITCH, ASA, აგრეთვე დაემატა რამდენიმე სერვერი (WEB Server, MC SQL Server, Elastix და ა.შ.).

ილუსტრაცია N 20 2007 წლის სტრუქტურა



წყარო: (კვლევითი მასალები)

ილუსტრაცია N 21 საქართველოს უნივერსიტეტის კომპიუტერული ქსელი სტრუქტურა



წყარო: (კვლევითი მასალები)

უკანასკნელი ნახევარი საუკუნის განმავლობაში ბიზნესის განვითარებამ დიდ დონეებს მიაღწია, რაშიც დიდი წვლილი კომუნიკაციას და კომპიუტერულ ტექნოლოგიებს აქვს. უნივერსიტეტის ზრდასთან ერთად იზრდება კომპიუტერული ქსელის ინფრასტრუქტურაც. დღესდღეისობით აღარ არის ერთი რაიმე განსაკუთრებული მოდელი, იგი მორგებული უნდა იყოს ბიზნესსზე. ქსელის დიზაინი გვაძლევს ბალანსს ხელმისაწვდომობის, უსაფრთხოებისა, მოქნილობის და მართვას შორის.

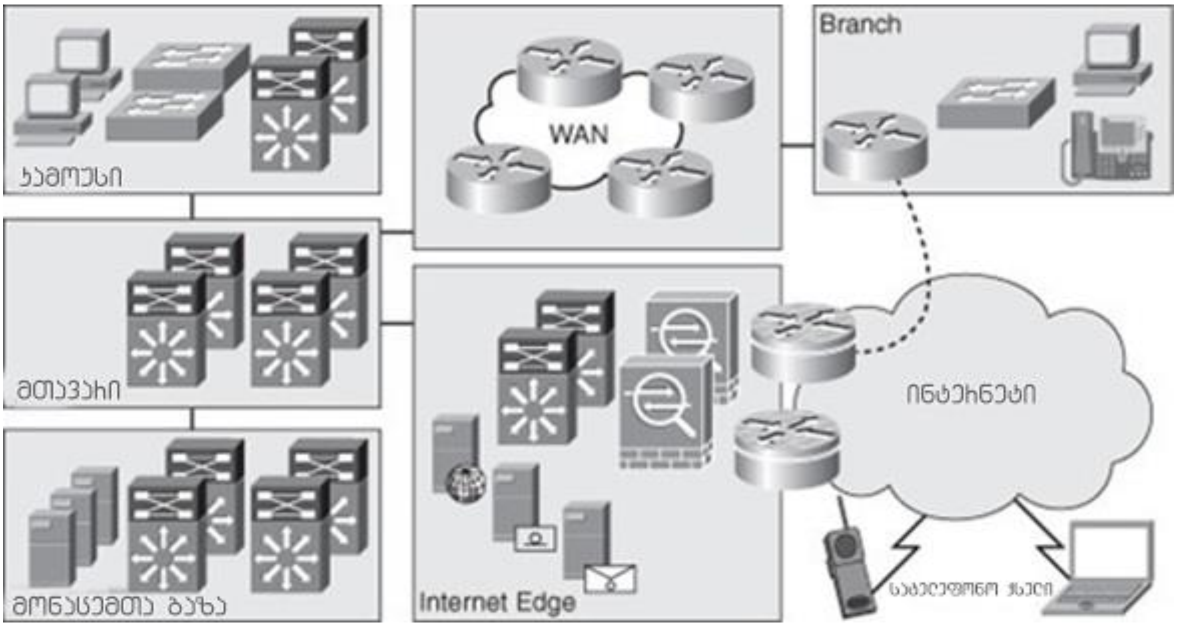
შესავალი საწარმო კამპუსის ქსელის დიზაინში

Cisco აქვს რამდენიმე სხვადასხვა მოდელები ქსელის დიზაინის:

1. Core ხერხემალი
2. კამპუსის მონაცემთა ცენტრი
3. ფილიალი / WAN

4. ინტერნეტ საზღვრები

ილუსტრაცია N 22 მაღალი დონის დიზაინი საწარმოთა ქსელიში.



წყარო: (კვლევითი მასალები)

კამპუსი არის ქსელის ნაწილი. მას განიხილავენ როგორც ინფრასტრუქტურას, რომელიც აკავშირებს სერვისებს და რესურსებს მომხმარებელთან. კამპუსი შეიძლება იყოს ერთ სართულზე, ერთ ოთახში ან დიდი შენობების ჯგუფში, რომელიც განლაგებულია სხვადასხვა ადგილას. მას შეუძლია იყოს ძირითადი ხერხემალი ქსელში და ასევე აკავშირებს მონაცემთა ცენტრს გარე ქსელთან.

მონაცემების ცენტრი, როგორც ნაწილი საწარმოს ქსელის, ზოგადად მიჩნეულია, რომ დაწესებულებაში გამოიყენება სახლის გამოთვლითი სისტემებისთვის, როგორიცაა სერვერები, რომელიც წვდომის საშვალებას გვაძლევს მეილთან, მონაცემთა ბაზასთან და სხვადასხვა აპლიკაციებთან.

ფილიალი / WAN არის ნაწილი, რომელიც შეიცავს როუთერებს, სვიჩებს და ა.შ. და აკავშირებს მთავარ ოფისს სხვადასხვა ოფისებთან. გავითვალისწინოთ, რომ ბევრ დიდ კომპანიას გააჩნია ბევრი კამპუსი და მონაცემთა ცენტრი, რომელიც დაკავშირებულია ერთმანეთთან. ხშირად დიდი კომპანიის ქსელები დაკავშირებულია სხვადასხვა

კომპანიის მონაცემთა ცენტრებთან, რომელიც მოითხოვს მაღალ გამტარობას შესაერთებლად, რომელიც თავის მხრივ მოითხოვს დამატებით routing თვისებებს და უმაღლეს სიჩქარეს.

ინტერნეტი არის ნაწილი საწარმოს ქსელის, რომელიც მოიცავს როუტერებს, სვიჩებს,ფაირვოლებს და ქსელის სხვადასხვა მოწყობილობებს, რათა დააკავშიროს ცენტრი ინტერნეტთან.

იერარქიული მოდელები - კამპუსის დიზაინი. განვიხილოთ ღია სისტემის ურთიერთკავშირი (OSI) მინიშნება მოდელი, რომელიც არის ფენიანი მოდელი. OSI მოდელის გამოყენებით მარტივდება კომუნიკაცია ორ კომპიუტერს შორის.

კამპუსის დიზაინი ასევე იყენებს ფენებს არქიტექტურის გასამარტივებლად თითოეული ფენა მიმართულია რაიმე კონკრეტული ფუნქციისკენ რომელიც უმარტივებს ქსელის დიზაინერს სწორი სისტემის აწყობას. კამპუსის არქიტექტურა შედგება access, distribution და core layers თავისი მახასიათებლებით.

Access layer კამპუსის დიზაინში. Access layer არის სვიჩი, რომელსაც აქვს პორტები რათა დაუკავშირდეს workstations, servers, printers, wireless access points. ეს ფენა არის ყველაზე მდიდარი, რადგან მას შეუძლია security, access control, filters, management მართვა.

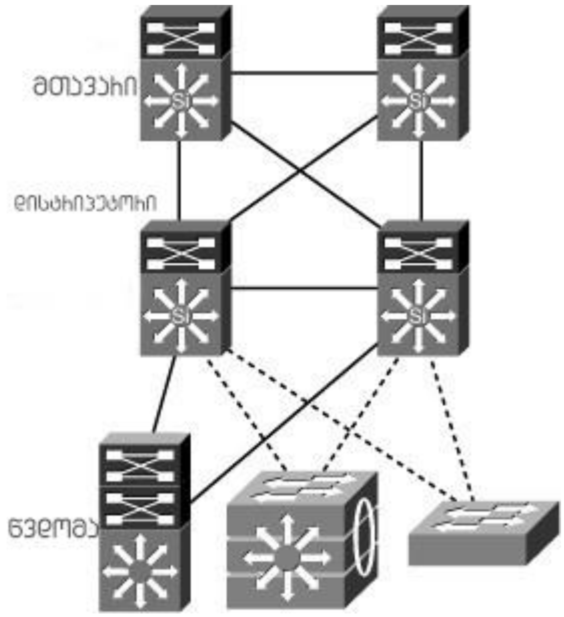
Distribution layer: გააჩნია ნაკლები უსაფრთხოება, ძირითად გამოიყენება access და core layers სერვისების გასაცვლელად.

Core ფენა (ასევე მოიხსენიება, როგორც ხერხემალი): მაღალი სიჩქარის მაგისტრალური, რომელიც განკუთვნილია პაკეტების უსწრაფესად გადასაცემად. თუ კარგად არის დიზაინი აწყობილი, იგი უკავშირდება სვიჩებს 10 Gigabit Ethernet-ით. რადგან იგი კრიტიკულია კავშირისთვის, მან უნდა შეძლოს და განახორციელოს მაღალ დონეზე ცვლილებები.

ეს მოდელი არაა ახალი და უკეთესია სხვა მოდელებთან განსხვავებით შემდეგი მიზეზების გამო:

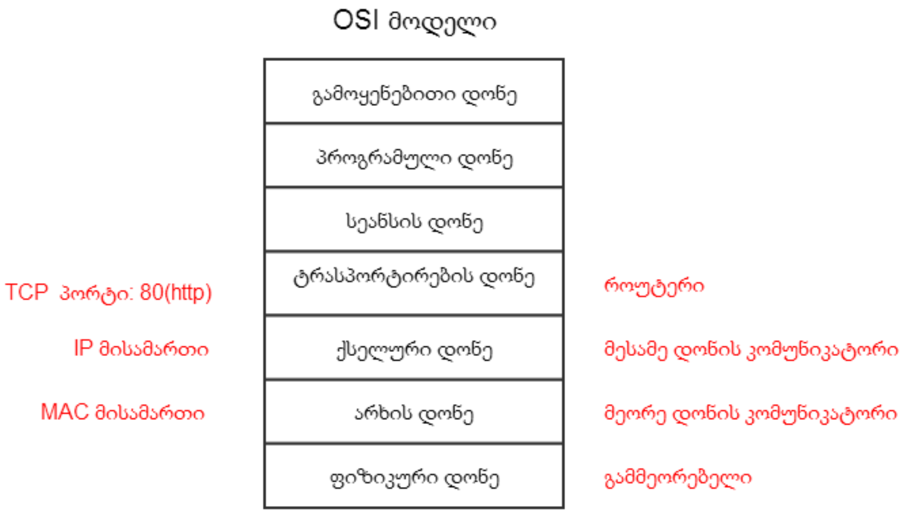
1. უზრუნველყოფს მოდულარობას.
2. ადვილია იმის გაგება
3. ზრდის მოქნილობას
4. აადვილებს ზრდას და მასშტაბს
5. უზრუნველყოფს ქსელის პროგნოზირებადობას
6. ამცირებს troubleshooting სირთულის

ილუსტრაცია N 23 მაღალი დონის კამპუსის ქსელის იერარქიული მოდელის მაგალითი.



წყარო: (Cisco.com)

ილუსტრაცია N 24 OSI მოდელის ურთიერთობა პროტოკოლებთან და ქსელთან



წყარო: (კვლევითი მასალები)

Layer 2 switches-ებში პაკეტების გადაცემა დაფუძნებულია MAC addresses-ებზე. Layer 2 switches ზრდის ქსელის გამტარობას და პორტების სიხშირეს. Layer 2 switching ნიშნავს რომ frame-ების შეცვლა შეუძლებელია. ასევე Layer 2 switch-ს გააჩნია რამდენიმე შესაძლებლობა Layer 3-ისა, როგორიცაა პაკეტების კლასიფიცირება და ქსელში წვდომა IP მისამართებით. მხოლოდ Layer 2-ს არ შეუძლიათ ჩარჩოების გადაცემა IP მისამართით და დაფუძნებულია მხოლოდ MAC მისამართებზე.

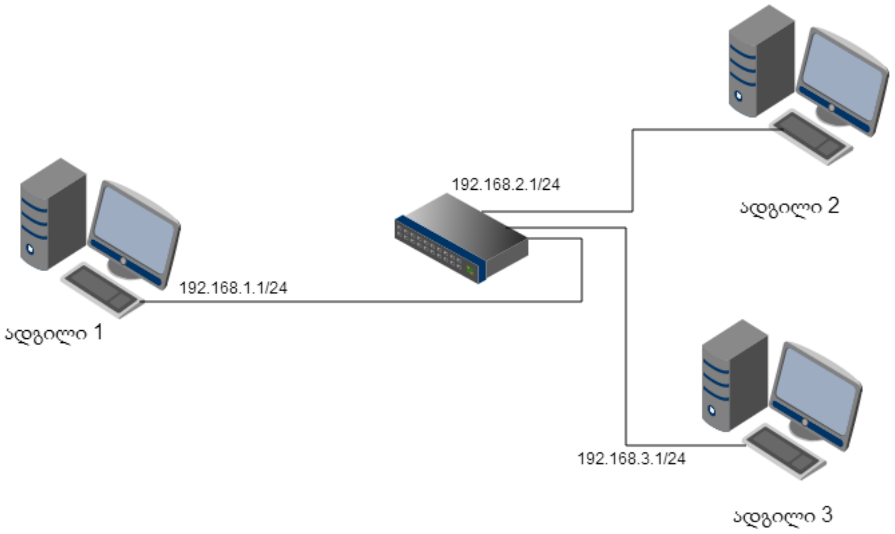
მემკვიდრე Layer 2 switches ქსელის ყველა მოწყობილობა უნდა იყოს კომპიუტერული ქსელის ერთ სეგმეტში, რათა პაკეტები გაცვალონ ერთმანეთში. ქსელის მოწყობილობები ერთად შეჯგუფებულები, რომლებმაც უნდა გადაცვალონ პაკეტები ქმნიან დომეინს. Layer 2 switches ძირავენ უცნობ unicast, multicast, და broadcast trafficს. ეს შესაძლებლობა იცავს ქსელის ტოპოლოგიას ზედმეტად გაზრდისგან, ვიდრე Layer 2 switches შეუძლია. ილუსტრატორი N 25 ასახავს Layer 2 switching-ს, რომელიც შეთავსებადია მხოლოდ MAC დამისამართებით.

ილუსტრაცია N 25. Layer 2 switching

წყარო: (cisco.com)

Layer 3 Switching. Layer 3 switches შეიცავს Layer 3 routing შესაძლებლობას. ბევრ ამ ფენის სვიჩებს აქვთ შესაძლებლობა BGP, RIP, OSPF, და EIGRP როუთინგის. მაგრამ მხოლოდ მცირეოდენ მათგანს აქვს BGP პროტოკოლის მხარდაჭერა, რადგან მას დიდი მეხსიერება სჭირდება და სვიჩებს არ აქვთ ამხელა მეხსიერება.

ილუსტრაცია N 26 Layer 3 Switching



წყარო: (კვლევითი მასალები)

Layer 4 and Layer 7 Switching. Layer 4 and Layer 7 Switching ტერმინოლოგია არაა ისეთივე მარტივი, როგორც წინა ორი. Layer 4 Switching დაფუძნებულია პროტოკოლებზე. მარტივად რომ ვთქვათ, იგი არა მარტო გამგზავნის და მიმღების IP მისამართებს იყენებს, არამედ IP სესიებსაც. ყველაზე განმასხვავებელი მეთოდი Layer 4 switching ის არის TCP და UDP port ნომრები. სერვერი იცავს ბალანსს Layer 4 and Layer 7 Switching-ს შორის. იგი ანსხვავებს ერთმანეთისგან IP trafficებს, როგორიცაა FTP, NTP, HTTP, SHTTP, SSH.
Layer 7 switching დაფუძნებულია აპლიკაციურ ინფორმაციაზე. Layer 7 switching გულისხმობს შიგთავსის გაშიფრვას როგორიცაა URLs, cookies, host headers,
ცხრილი N 4 გვიჩვენებს OSI მოდელს თავისი თვისებებით.

ცხრილი N 4: OSI მოდელს თავისი თვისებებით.

OSI Level	OSI Layer	PDU Type	Device Example	Address
1	Physical	Electrical signals	Repeater, transceiver	None
2	Data link	Frames	Switches	MAC address
3	Network	Packet	Router, multilayer switches	IP address
4	Transport	TCP or UDP data segments	Multilayer switch load balancing based on TCP port number	TCP or UDP port numbering
7	Application	Embedded application information in data payload	Multilayer switch using Network-Based Application Recognition (NBAR) to permit or deny traffic based on data passed by an application	Embedded information in data payload

წყარო: (cisco.com)

Layer 2 Switching სიღრმეში. ეს დონე დაფუძნებულია hardware-ზე bridging. ამ დონეს შეუძლია გამტარუნარიანობა გაზარდოს მაქსიმუმამდე ქსელში ზედმეტი მოწყობილობების დამატების გარეშე. ამ დონეზე frame მოდიფიცირება არ სჭირდება, როდესაც გადადის ქვედა დანოზე, როგორც Fast Ethernetდან 10 Gigabit Ethernetში.
Layer 2 switches შეიცავს შემდეგს:

1. აქვს მაღალი სიჩქარე
2. მცირე დაყოვნება

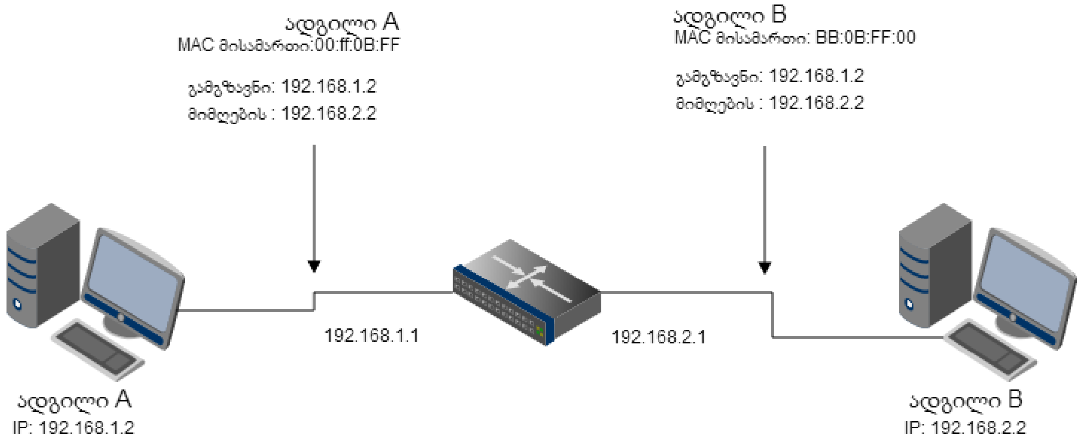
3. თავსებადია სხვადასხვა სვიჩ ტოპოლოგიაში როუთერებისა და Layer 3 switch
4. მხარს უჭერს Layer 3 switch ის სხვადასხვა ფუნქციებს.

Layer 3 Switching სიღრმეში. Layer 3 Switch უმკლავდებიან არაადეკვატურობას Layer 2 scalability და უზრუნველყოფს მარშრუტიზაციის დომენებს. Layer 3 Switch ასრულებს ყველაფერს პაკეტბთან დაკავშირებით, რომელიც ტრადიციული როუტერი აკეთებს, მათ შორის:

1. განსაზღვრავს, გადაცემის გზას დაფუძნებული Layer 3 ინფორმაციზე
2. განსაზღვრავს მთლიანობის Layer 3 პაკეტი header -ის მეშვეობით
3. ამოწმებს პაკეტის Time-To-Live (TTL) გასვლის
4. გადაწერა გამგზავნის და მიმღების MAC მისამართს IP rewrites დროს
5. მიმართავს უსაფრთხოების კონტროლს და საჭიროების შემთხვევაში კლასიფიკაციას უკეთებს მომსახურებას.

Layer 3 მარშრუტიზაცია მოითხოვს პაკეტების გადაწერის უნარს.
მთავარი განსხვავება პაკეტების გადაცემის ოპერაციაში როუთერსა და Layer 3 switch შორის არის ფიზიკური შესრულება. Layer 3 switch აქვს განსხვავებული hardware კომპონენტები და მეტი პორტი, ვიდრე ტრადიციულ როუტერებს.

ილუსტრაცია N 27 პაკეტების გადაწერის შესაძლებლობა Layer 3 მარშრუტიზაციით, სადაც ორი კომპიუტერი უკავშირდება ერთმანეთს ICMP

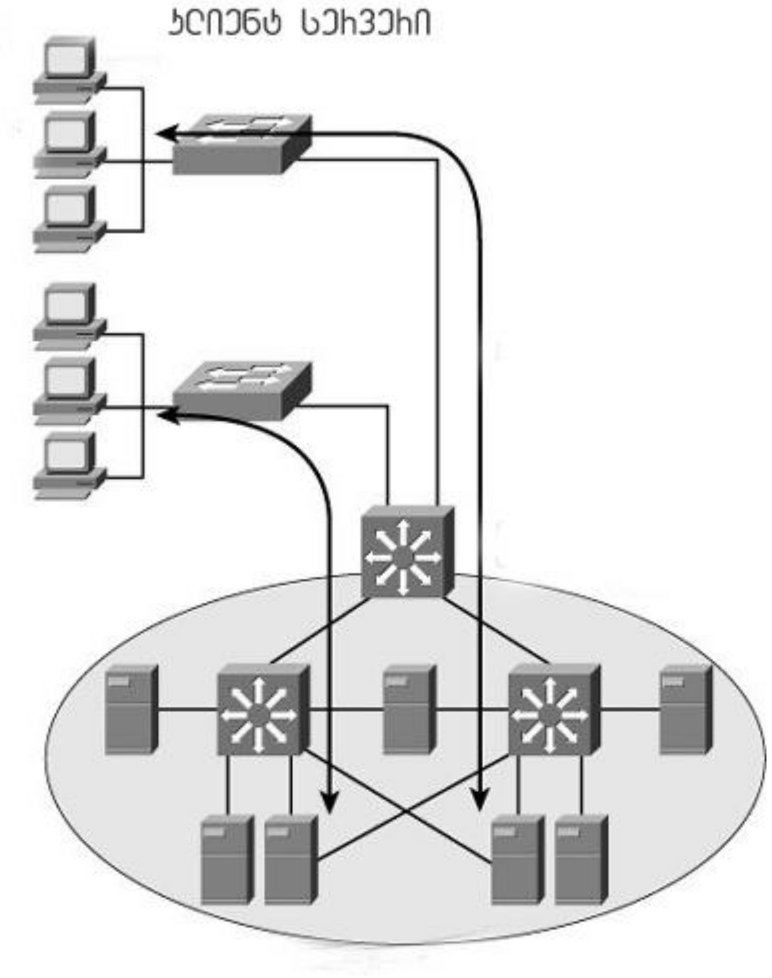


წყარო: (კვლევითი მასალები)

მრავალ დონიანი Switching-ი. Multilayer Switching აერთიანებს Layer 2 switching და Layer 3 routing. სინამდვილეში ქსელი იყენებს Layer 3 switch და მრავალდონიან სვიჩს, რომელსაც გააჩნია Layer 2 მხარდაჭერა Layer 3 switching. Layer 2 და Layer 3 ფუნქციების კომბინაცია მარტივ მართვადს ხდის ქსელის ტოპოლოგიას.

კლიენტი/სერვერის აპლიკაციები. ბევრი კომპანიიის ტრაფიკის ნაკადები დაფუძნებულია კლიენტი/სერვერის მოდელზე, სადაც კავშირი სერვერთან შესაძლოა გახდეს ვიწრო გასასვლელი. ქსელის სიჩქარეს იყო ძვირი, მაგრამ დღეს, ნაკლები ღირს აპლიკაციის მოთხოვნებთან შედარებით. მაგალითად, Gigabit Ethernet -ის ღირებულება და 10 Gigabit არის მომგებიანი, თუ შევადარებთ აპლიკაციის სიჩქარის მოთხოვნებს, რაც იშვიათად აღემატება 1 Gigabit Ethernet. სიტუაციებში, რომელშიც ყველაზე დაბალი შეყოვნებაა საჭირო.

ილუსტრაცია N 28 კლიენტი/სერვერის ტრაფიკის ნაკადი



წყარო: (cisco.com)

დიდ კომპანიებში, აპლიკაციების ტრაფიკმა შეიძლება გადაკვეთოს ერთზე მეტი გაყვანილობის კაბინეტი ან LAN-ი, იმისათვის რომ მიწვდეს მონაცემთა ცენტრის სერვერის ჯგუფში არსებულ აპლიკციებს. კლიენტი/სერვერის აპლიკაციების მოიცავენ შემდეგს:

1. ორგანიზაციული ფოსტის სერვერები
2. საერთო ფაილის სერვერები
3. საერთო მონაცემთა ბაზის სერვერები ორგანიზაციული

აპლიკაციებისთვის, როგორიცაა ადამიანური რესურსები, ინვენტარი, და გაყიდვების განცხადებები

მსხვილი საწარმოების მომხმარებლები მოითხოვენ სწრაფ, საიმედო და კონტროლირებად წვდომას კრიტიკულ აპლიკაციებთან.

თავი 4. *კომპიუტერული ქსელის სტრუქტურის საიმედოობისა და უსაფრთხოების ლოგიკურ-ალბათური მოდელირება*

4.1. *სისტემის ქმედითუუნარიანობის პირობის აღწერა ლოგიკურ-ალბათური ფუნქციების საშუალებით*

ლოგიკურ-ალბათური მეთოდების საფუძვლები. ლოგიკურ-ალბათური მეთოდი (ლამ) ეწოდება სისტემების ეფექტიანობის (საიმედოობის, სიცოცხლისუნარიანობის, უსაფრთხოების, რისკების) გაანაგრიშების მეთოდს, როდესაც სისტემის სტრუქტურა აღიწერება მათემატიკური ლოგიკის საშუალებით, ხოლო საიმედოობის რაოდენობრივი შეფასება წარმოებს ალბათობის თეორიის გამოყენებით.

შეგვიძლია გამოვყოთ ლამ-ის განვითარების სამი პერიოდი:

1. ლოგიკური ცვლადების პირდაპირი ჩანაცვლება ალბათური ცვლადებით და ლოგიკური ოპერაციების ჩანაცვლება მათემატიკური ოპერაციებით, რაც დასაშვებია მხოლოდ მარტივი სტრუქტურის სისტემებისათვის.
2. სპეციალიზირებული ალგორითმების დამუშავების პერიოდი, რომელთა საშუალებით შესაძლებელი ხდებოდა ნებისმიერი ტიპის ლოგიკის ალგებრის ფუნქციის გარდაქმნა და მათი ჩანაცვლება ალბათობით.
3. დიდი განზომილების რთული სტრუქტურის სისტემების ავტომატიზებული ლოგიკურ-ალბათური მოდელირების პერიოდი.

ძირითადი განმარტებები და აღნიშვნები. შემოვიტანოთ არგუმენტის ხარისხის ცნება $x_i^{a_i}$, სადაც a_i– ორობითი ცვლადია: x_i ცვლადებს და მათ უარყოფებს x_i' ($i=1, 2, \ldots, n$) ვუწოდებთ ასოით ცვლადებს (ასოებს), i-ს ცვლადების ნომრებს ან ინდექსებს.

$$x_1^{a_1} x_2^{a_2} \ldots x_r^{a_r}$$

ეწოდება r რანგის ელემენტარული კონუნქცია, რომელშიც ყველა ასო x განსხვავებულია, ვინაიდან ვიცით, რომ $x_i x_i'=0$ და $x_i x_i \ldots x_i=x_i$.

$$K_1 \vee K_2 \vee \ldots \vee K_s,$$

სადაც K_j – განსხვავებული რანგის ელემენტარული კონუნქციებია, ეწოდება *დიზუნქციური ნორმალური ფორმა* (დნფ). მაგალითად, ფუნქცია

$$f(x_1, \ldots, x_4) = x_1 x_2 \vee x_1 x_2 x_3' \vee x_1' x_3 x_4$$

ჩაწერილია დნფ სახით, რადგანაც სამივე შესაკრები წარმოადგენს ელემენტარულ კონიუნქციას.

განსაზღვრება 3. თუ ფუნქცია $f(x_1, x_2, \ldots, x_n)$ ჩაწერილია დნფ სახით და ყველა ელემენტარული კონუნქციის რანგი უდრის n-ს, მაშის ასეთ დნფ-ს ვუწოდებთ *სრულყოფილ დიზუნქციურ ნორმალურ ფორმას* (სდნფ), ხოლო შემადგენელ კონუნქციებს - სდნფ-ის წევრებს.

$$x_1^{a_1} \vee x_2^{a_2} \ldots \vee x_r^{a_r}$$

r რანგის ელემენტარული დიზუნქცია ეწოდება.

განსაზღვრება 5. ორ ელემენტარულ კონუნქციას ეწოდება *ორთოგონალური*, თუ მათი ნამრავლი 0-ის ტოლია. მაგალითად, ელემენტარული კონუნქციების x_1x_2და $x_1x_2'x_3x_4$ ნამრავლი 0-ის ტოლია, ვინაიდან პირველი კონუნქცია შეიცავს x_2-ს, ხოლო მეორე x_2'- ს, შესაბამისად ისინი ორთოგონალურია.

განსაზღვრება 6. დნფ-ს ეწოდება *ორთოგონალური დიზუნქციური ნორმალური ფორმა* (ოდნფ), თუ ყველა მისი წევრი წყვილ-წყვილად ორთოგონალურია.

ამ განსაზღვრებიდან გამომდინარეობს, რომ სდნფ წარმოადგენს ოდნფ-ს, ვინაიდან ყველა მისი წევრი წყვილ-წყვილად ორთოგონალურია. მაგრამ სდნფ ყველაზე არაეკონომიური ოდნფ-ია, ვინაიდან შეიცავს მაქსიმალური რაოდენობის ასოებს.

არაგანმეორებადი დიზუნქციური ნორმალური ფორმა (ადნფ) ეწოდება ისეთ დიზუნქციური ნორმალური ფორმას, რომელშიც ყველა ასოს გააჩნია განსხვავებული ნომერი (ინდექსი).

მაგალითად, x_i-ს და x_i'-ს ერთნაირი ინდექსები აქვთ, ამიტომ შეუძლებელია ისინი ერთდროულად შედიოდნენ ადნფ-ში.

ლოგიკის ალგებრის ფუნქციის (ლაფ) *არაგანმეორებადი ფორმა* ეწოდება ისეთ ფორმას, რომელშიც ყველა ასოს გააჩნია განსხვავებული ნომრები (ინდექსები). არაგანმეორებადი ლაფ-ს კერძო შემთხვევას წარმოადგენს ადნფ. მაგალითად, ფუნქცია

$$f(x_1, \ldots, x_8) = x_1 \begin{vmatrix} x_2 \\ x_3 \\ x_4' \end{vmatrix} \vee x_5 \begin{vmatrix} x_6 \\ x_7 x_8' \end{vmatrix}$$

ჩაწერილია არაგანმეორებადი ფორმით, ვინაიდან ყველა ცვლადი განსხვავებული ნომრითაა.

ალბათური ფუნქცია (აფ) ეწოდება ლოგიკის ალგებრის ფუნქციის ჭეშმარიტების ალბათობას

$$P\{f(x_1, \ldots, x_n) = 1\}.$$

ლოგიკის ალგებრის ფუნქციას, რომელიც უშვებს ალბათურ ფუნქციაზე გადასვლას ლოგიკური ცვლადების ალბათურით შეცვლითა და ლოგიკური ოპერაციების შესაბამისი არითმეტიკული ოპერაციებით შეცვლის გზით, ეწოდება *ჩანაცვლებაზე გადასვლის ფორმა.*

ალბათური ფუნქციის შერეული ფორმა ეწოდება ფუნქციის ფორმას, რომელიც მიიღება ლოგიკური ცვლადებისა და ოპერაციების ნაწილობრივი ჩანაცვლებით ალბათური ცვლადებითა და მათემატიკური ოპერაციებით და რომელიც ერთდროულად შეიცავს როგორც ლოგიკურ ცვლადებსა და ოპერაციებს, ასევე ალბათურ ცვლადებსა და მათემატიკურ ოპერაციებს.

ლაფ-ის ფორმა, რომელიც უშვებს ალბათური ფუნქციის შერეულ ფორმაზე გადასვლას ლოგიკური ცვლადებისა და ოპერაციების ნაწილობრივი ჩანაცვლებით ალბათური ცვლადებითა და მათემატიკური ოპერაციებით ეწოდება *ნაწილობრივი ჩანაცვლებაზე გადასვლის ფორმა.*

A და B გამონათქვამების *ექვივალენტობა* აღინიშნება „~ „ სიმბოლოთი და განისაზღვრება შემდეგი ჭეშმარიტების მნიშვნელობებით:

$$0 \sim 0 = 1;\ 0 \sim 1 = 0;\ 1 \sim 0 = 0;\ 1 \sim 1 = 1.$$

A და B გამონათქვამების *2-ის მოდულით შეკრება* განისაზღვრება შემდეგი ჭეშმარიტების მნიშვნელობებით:

$$0 \oplus 0 = 0;\ 0 \oplus 1 = 1;\ 1 \oplus 0 = 1;\ 1 \oplus 1 = 0.$$

2-ის მოდულით შეკრების ლოგიკურ ოპერაციისათვის ადგილი აქვს შემდეგ კანონებს:

$$A \oplus B = B \oplus A;$$
$$A \oplus (B \oplus C) = (A \oplus B) \oplus C;$$
$$A \wedge (B \oplus C) = (A \wedge B) \oplus (A \wedge C).$$

$$\left.\begin{array}{ll} A \oplus 1 = A'; & 34.\ A \oplus A = 0; \\ A \oplus 0 = A; & 35.\ A \oplus A' = 1. \end{array}\right\}$$

$$\left.\begin{aligned} A \vee B &= A \oplus B \oplus AB; \\ A \wedge B &= A \oplus AB'; \\ A \oplus B &= AB' \vee A'B. \end{aligned}\right\}$$

$f(x_1, x_2, \ldots, x_n)$ ფუნქციის *ბულის სხვაობა* (ლოგიკური სხვაობა) x_i არგუმენტით ეწოდება საწყისი ფუნქციისა და საწყის ფუნქციაში x_i არგუმენტის მისი უარყოფით შეცვლით მიღებულ ფუნქციის 2-ის მოდულით შეკრებით მიღებულ ფუნქციას

$$\Delta_{x_i} f(x_1, \ldots, x_n) = f(x_1, \ldots, x_i, \ldots, x_n) \oplus f(x_1, \ldots, x_i', \ldots, x_n).$$

$$f_{x_i'}(x_1, \ldots, x_n) = f(x_1, \ldots, x_i', \ldots, x_n)$$

ეწოდება *სიმეტრიული ფუნქცია* x_i არგუმენტით საწყისი $f(x_1, x_2, \ldots, x_n)$ ფუნქციის მიმართ.

განსაზღვრება 17. ფუნქციას, რომელიც მიიღება x_i არგუმენტის შეცვლით 1-ით და 0-ით, ეწოდებათ ერთეულოვანი და ნულოვანი ფუნქციები x_i არგუმენტის მიმართ:

$$f_1^{(i)}(x_1, \ldots, x_n) = f(x_1, \ldots, 1, \ldots, x_n);$$
$$f_0^{(i)}(x_1, \ldots, x_n) = f(x_1, \ldots, 0, \ldots, x_n).$$

$f(x_1, x_2, \ldots, x_n)$ ფუნქციას ეწოდება *მონოტონური*, თუ ნებისმიერი ლოგიკური არგუმენტებისათვის $(\alpha_1, \ldots, \alpha_n)$ და $(\beta_1, \ldots, \beta_n)$, რომლებიც აკმაყოფილებენ პირობას $\alpha_i \le \beta_i$, ადგილი აქვს შემდეგ თანაფარდობას:

$$f(\alpha_1, \ldots, \alpha_n) \le f(\beta_1, \ldots, \beta_n).$$

მატრიცის სახით ჩაწერილ ფუნქციას, რომელშიც კონუნქცია აღინიშნება სტრიქონებში ჩაწერილი ლოგიკური სიმბოლოებით, ხოლო დიზუნქცია სვეტებში ჩაწერილი ლოგიკური სიმბოლოებით, *ლოგიკური მატრიცა* ეწოდება.

ლოგიკური მატრიცისათვის გამოიყენება ლოგიკისალგებრის ყველა ცნობილი გარდაქმნა. მაგალითად, კონუნქციის გადანაცვლებადობის კანონი გულისხმობს სტრიქონში სიმბოლოების გადანაცვლების დაშვებას, დიზუნქციის გადანაცვლებადობის კანონი გულისხმობს სვეტებში სიმბოლოების გადანაცვლების დაშვებას.

ვთქვათ მოცემული გვაქვს შემდეგი ლაფ:

$$f(x_1, \dots, x_8) = \Big\{\big\{x_1 \wedge x_3 \wedge \big[x_5 \vee \big(x_4 \wedge x_6 \wedge x_8\big)\big]\big\} \vee$$
$$\vee \big\{x_2 \wedge x_4 \wedge \big[x_6 \vee \big(x_3 \wedge x_5 \wedge x_8\big)\big]\big\}\Big\} \wedge x_7.$$

მატრიცული ფორმით მოცემული ლოგიკური განტოლება ჩაიწერება შემდეგნაირად:

$$f(x_1, \dots, x_8) = \left|\begin{array}{l|l} x_1x_3 & x_5 \\ & x_4x_6x_8 \\ x_2x_4 & x_6 \\ & x_3x_5x_8 \end{array}\right| x_7 = \left|\begin{array}{l} x_1x_3x_5x_7 \\ x_1x_3x_4x_6x_8x_7 \\ x_2x_4x_6x_7 \\ x_2x_4x_3x_5x_8x_7 \end{array}\right|$$

საბოლოო შედეგი ჩაწერილია დნფ-ის სახით.

ინვერსიის კანონის გამოყენებით მივიღებთ:

$$f(x_1, \dots, x_8)' = \left|\begin{array}{l} x_1x_3x_5x_7 \\ x_1x_3x_4x_6x_8x_7 \\ x_2x_4x_6x_7 \\ x_2x_4x_3x_5x_8x_7 \end{array}\right| = \left|\begin{array}{l|l|l|l} x_1' & x_1' & x_2' & x_2' \\ x_3' & x_3' & x_4' & x_4' \\ x_5' & x_4' & x_6' & x_3' \\ x_7' & x_6' & x_7' & x_5' \\ & x_8' & & x_8' \\ & x_7' & & x_7' \end{array}\right|$$

წყარო: (ცირამუა, 2014)

ლოგიკის ალგებრისა და ალბათური ლოგიკის თეორემები. ალბათობის თეორიასა და მათემატიკურ ლოგიკას შორის მჭიდრო კავშირი დიდი ხანია შენიშნულია. თანამედროვე პირობებში მათემატიკური ლოგიკა და ალბათობის თეორია გაერთიანებულია ლოგიკურ-ალბათურ თეორიაში.

ალბათობის თეორიის საფუძველზე წარმოებს სისტემების საიმედოობისა და უსაფრთხოების რაოდენობრივი შეფასება, ხოლო მათი სტრუქტურა აღიწერება მათემატიკური ლოგიკის საშუალებით.

რთული სტრუქტურის სისტემებისათვის ძირითად სირთულეს წარმოადგენს ლაფ-ის გარდაქმნა სრული ჩანაცვლების ფორმაში. ამისათვის საჭირო შეიქნა ერთგვარი „ხიდის“ გადება ალბათობის თეორიასა და მათემატიკურ ლოგიკას შორის. განვიხილოთ ლოგიკის ალგებრისა და მათემატიკური ლოგიკის ზოგიერთი თეორემა მკაცრი დამტკიცების გარეშე.

n არგუმენტზე (*n≥1*) დამოკიდებული ნებისმიერი ლაფ შესაძლებელია წარმოდგენილი იქნას შემდეგნაირად:

$$f(x_1, \ldots, x_n) = x_i f_1^{(i)}(x_1, \ldots, x_n) \vee x_i' f_0^{(i)}(x_1, \ldots, x_n).$$

ეს გამოსახულება ცნობილია, როგორც *შენონის დაშლა.*

$$\begin{aligned} f(x_1, \ldots, x_n) &= x_i f_1^{(i)}(x_1, \ldots, x_n) \vee x_i' f_0^{(i)}(x_1, \ldots, x_n) = \\ &= x_i f_1^{(i)}(x_1, \ldots, x_n) \oplus x_i' f_0^{(i)}(x_1, \ldots, x_n) \oplus \\ &\oplus x_i x_i' f_1^{(i)}(x_1, \ldots, x_n) f_0^{(i)}(x_1, \ldots, x_n) = \\ &= x_i f_1^{(i)}(x_1, \ldots, x_n) \oplus x_i' f_0^{(i)}(x_1, \ldots, x_n). \end{aligned}$$

ნებისმიერი ლაფ-ის ბულის სხვაობა x_i არგუმენტით შეგვიძლია წარმოვადგინოთ შემდეგნაირად

$$\Delta_{x_i} f(x_1, \ldots, x_n) = f_1^{(i)}(x_1, \ldots, x_n) \oplus f_0^{(i)}(x_1, \ldots, x_n).$$

დამტკიცება:

$$\begin{aligned} \Delta x_i f(x_1, \ldots, x_n) &= f(x_1, \ldots, x_n) \oplus f_{x_i'}(x_1, \ldots, x_n) = \\ &= \left[x_i f_1^{(i)}(x_1, \ldots, x_n) \oplus x_i' f_0^{(i)}(x_1, \ldots, x_n)\right] \oplus \\ &\oplus \left[x_i' f_1^{(i)}(x_1, \ldots, x_n) \oplus x_i f_0^{(i)}(x_1, \ldots, x_n)\right] = \\ &= (x_i \oplus x_i') f_1^{(i)}(x_1, \ldots, x_n) \oplus (x_i' \oplus x_i) f_0^{(i)}(x_1, \ldots, x_n) = \\ &= f_1^{(i)}(x_1, \ldots, x_n) \oplus f_0^{(i)}(x_1, \ldots, x_n). \end{aligned}$$

ნებისმიერი ლაფ-ის ბულის სხვაობა x_i არგუმენტით შეგვიძლია წარმოვადგინოთ შემდეგნაირად

$$\Delta_{x_i} f(x_1, \ldots, x_n) = \begin{vmatrix} f_1^{(i)}(x_1, \ldots, x_n) f_0'^{(i)}(x_1, \ldots, x_n) \\ f_1'^{(i)}(x_1, \ldots, x_n) f_0^{(i)}(x_1, \ldots, x_n) \end{vmatrix}.$$

n არგუმენტზე (*n≥1*) დამოკიდებული ნებისმიერი ლაფ შესაძლებელია წარმოდგენილი იქნას შემდეგნაირად:

$$\begin{aligned} &f(x_1, \ldots, x_i, x_{i+1}, \ldots, x_n) = \\ &= \vee x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_i^{\alpha_i} f(\alpha_1, \alpha_2, \ldots, \alpha_i, x_{i+1}, \ldots, x_n). \end{aligned}$$

ყველა მონოტონური ლაფ-სათვის

$$\left\{(x_1, \ldots, x_n): f_0^{(i)}(x_1, \ldots, x_n) = 1\right\} \subset \left\{(x_1, \ldots, x_n): f_1^{(i)}(x_1, \ldots, x_n) = 1\right\}.$$

ყველა მონოტონური ლაფ-სათვის-დან გამომდინარეობს 5 შედეგი, რომელთა ცოდნა მნიშვნელოვნად აადვილებს მონოტონური ლაფ-ის გარდაქმნას.

1) $\left\{(x_1, \ldots, x_n): f_0^{(i)}(x_1, \ldots, x_n) = 1\right\} \subset \left\{(x_1, \ldots, x_n): f(x_1, \ldots, x_n)\right.$
$\subset \left\{(x_1, \ldots, x_n): f_1^{(i)}(x_1, \ldots, x_n) = 1\right\};$

2) $f_1^{(i)}(x_1, \ldots, x_n) \vee f_0^{(i)}(x_1, \ldots, x_n) \equiv f_1^{(i)}(x_1, \ldots, x_n);$

3) $f_1^{(i)}(x_1, \ldots, x_n) \wedge f_0^{(i)}(x_1, \ldots, x_n) \equiv f_0^{(i)}(x_1, \ldots, x_n);$

4) $f_1^{(i)}(x_1, \ldots, x_n) \wedge f_0'^{(i)}(x_1, \ldots, x_n) \equiv \Delta_{x_i} f(x_1, \ldots, x_n);$

5) $f_1'^{(i)}(x_1, \ldots, x_n) \wedge f_0^{(i)}(x_1, \ldots, x_n) \equiv 0.$

მონოტონური ლაფ-ის $f(x_1, x_2, \ldots, x_n)$ ჭეშმარიტების ალბათობის კერძო წარმოებული x_i არგუმენტის ჭეშმარიტების ალბათობით რიცხობრივად უდრის ამ ფუნქციის ბულის სხვაობას x_i არგუმენტით

$$\frac{\partial P\left\{f(x_1, \ldots, x_n) = 1\right\}}{\partial P\left\{x_i = 1\right\}} = P\left\{\Delta_{x_i} f(x_1, \ldots, x_n) = 1\right\}.$$

ოდნფ-ის სახით მოცემული ნებისმიერი ლაფ-ის ჭეშმარიტების ალბათობა უდრის მოცემული ლაფ-ის ორთოგონალური წევრების ჭეშმარიტების ალბათობების ჯამს:

$$P\left\{f(x_1, \ldots, x_n) = \bigvee_{i=1}^{s} \square_i = 1\right\} = \sum_{i=1}^{s} P\left\{\square_i = 1\right\},$$

ორთოგონალური არაგანმეორებადი ფორმების დიზუნქცია კონუნქცია-უარყოფის ბაზისზე წარმოადგენს სრული ჩანაცვლებაზე გადასვლის ფორმას.

თუ ლაფ წარმოდგენილია სრული ჩანაცვლების სახით, მაშინ ადგილი აქვს შემდეგ წესებს:

$$P\left\{x_i = 1\right\} = R_i,\ P\left\{x_i = 0\right\} = P\left\{x_i' = 1\right\} = Q_i = 1 - R_i;$$

$$P\left\{f(x_1, \ldots, x_7) = \left[(x_1x_2)'(x_3x_4)'(x_5(x_6'x_7')')'\right]' = 1\right\} =$$
$$= 1 - (1 - R_1R_2)(1 - R_3R_4)[1 - R_5(1 - Q_6Q_7)];$$

ლოგიკური ნამრავლისა და შეკრების ოპერაციები იცვლება მათემატიკური ნამრავლითა და შეკრებით.

მაგალითი: ვთქვათ მოცემულია ლოგიკური ფუნქცია

$$f(x_1, \ldots, x_8) = x_1(x_2 \vee x_3 \vee x_4') \vee x_5(x_6 \vee x_7 x_8'),$$

და საჭიროა ვიპოვოთ $P\{(x_1, \ldots, x_8) = 1\}$.

ვინაიდან ეს ფუნქცია არაგანმეორებადია (მიუხედავად იმისა, რომ არ წარმოადგენს დნფ-ს), მივიღებთ

$$f(x_1, \ldots, x_8) = \left\{ \left\{ x_1 \left[x_2' x_3' x_4 \right] \right\}' \left\{ x_5 \left[x_6' (x_7 x_8') \right] \right\}' \right\}' ;$$

$$P\left\{ f(x_1, \ldots, x_8) = 1 \right\} =$$

$$= 1 - \left\{ 1 - R_1 \left[Q_2 Q_3 R_4 \right] \right\} \left\{ 1 - R_5 \left[1 - Q_6 (1 - R_7 Q_8) \right] \right\}.$$

წყარო: (ცირამუა, 2014).

რთული სტრუქტურის სისტემების დაპროექტების საკითხები. თანამედროვე სისტემების სტრუქტურისა და ფუნქციონირების ალგორითმის სირთულე განაპირობებს მათემატიკური აპარატის გამოყენების აუცილებლობას დაპროექტების ადრეულ სტადიაზე და საიმედოობაზე მოთხოვნის ფორმირების პროცესში, რაც წარმოადგენს საკმაოდ რთულ ამოცანას. სისტემის ცალკეულ ნაწილებზე საიმედოობაზე მოთხოვნის ფორმირება შედარებით ადვილი ამოცანაა და წარმოადგენს სისტემის ოპტიმალური სინთეზის ამოცანას.

რთული სისტემების დაპროექტების საწყის ეტაპზე ძირითად ამოცანას წარმოადგენს რაციონალური სტრუქტურისა და ფუნქციონირების ალგორითმის განსაზღვრა. ამ ორივე ამოცანის გადაჭრა ხორციელდება კონსტრუქტორების მიერ მათი გამოცდილებისა და ინტუიციის საფუძველზე. მაგრამ როგორც წესი გამოცდილებაზე და ინტუიციაზე დაყრდნობით ვერ ხერხდება საუკეთესო ვარიანტის მიღება, ვინაიდან ამოცანას უამრავი ამოხსნა გააჩნია. ასეთ შემთხვევაში წარმოებს სხვადასხვა ვარიანტების განხილვა და შედარებითი ანალიზის ჩატარება შედარებით უკეთესი ვარიანტის შერჩევის მიზნით.

სისტემების დაპროექტების საწყის ეტაპზე საჭიროა რეალური სისტემის საკმაოდ მარტივი, მაგრამ ამავე დროს ზუსტი მათემატიკური მოდელის შექმნა. მათემატიკური მოდელის სიზუსტეს ნაწილობრივ განსაზღვრავს საწყისი მონაცემების სიზუსტე და

ობიექტურობა, თუმცა მათი არარსებობის შემთხვევაშიც არ უნდა ვთქვათ უარი საიმედოობისა და უსაფრთხოების მატემატიკურ მოდელირებაზე. რთული სისტემების საიმედობის ლოგიკურ–ალბათური მოდელირება საშუალებას მოგვცემს საწყისი მონაცემების გარეშეც მოვახდინოთ სისტემის სტრუქტურების შედარებითი ანალიზი და შევარჩიოთ ოპტიმალური ვარიანტი. ამგვარად ჩვენი განხილვის საგანი იქნება ძირითადად რთული სისტემის სტრუქტურის (სტრუქტურული) საიმედოობა და ფუნქციონირების პროცესების აღწერა.

ასევე მნიშვნელოვანია სისტემის უსაფრთხოების საკითხი. ავარიების მიზეზების ანალიზი გვიჩვენებს, რომ ისინი, მიუხედავად მათი ხასიათისა, დროისა თუ რეგიონისა, ექვემდებარებიან გარკვეულ კანონზომიერებებს. ავარიებს წინ უსწრებს სისტემების შემადგენელი ნაწილების (ელემენტების) ფუნქციონირების ნორმალური რეჟიმიდან გადახრის ფაზა. ამ ფაზის ხანგრძლივობა შეიძლება იყოს წამები, წუთები ან წლები. გარკვეულ მომენტამდე სისტემის ელემენტების ფუნქციონირების ნორმალური რეჟიმიდან გადახრა შესაძლებელია არ წარმოადგენდეს საფრთხეს, მაგრამ მათი დაგროვების რომელიღაც კრიტიკულ მომენტში ხდება ავარია, რასაც სავალალო შედეგებამდე მივყავართ. ელემენტების ფუნქციონირების ნორმალური რეჟიმიდან გადახრების დაგროვება გამოწვეულია იმით, რომ ყოველთვის ვერ ხერხდება სისტემის ცალკეულ ელემენტებზე დაკვირვება და დიაგნოსტიკა, პერსონალი ეჩვევა ასეთ გადახრებს.

მეორე ფაზა – მაინიცირებელი (მოულოდნელი, იშვიათი) ხდომილების ფაზა, როდესაც სისტემა უცბად (აფეთქების მსგავსად) ხვდება სახიფათო მდგომარეობაში. ასეთ სემთხვევასი პერსონალს არ რჩება არც დრო და არც საშუალებები ეფექტური ღონისძიებების გასატარებლად.

მესამე ფაზა – ავარიის განვითარების ფაზა, როდესაც ადამიანურ ფაქტორს შეუძლია ითამაშოს როგორც დადებითი, ასევე უარყოფითი როლი. პირველ შემთხვევაში პერსონალის სწორ და ეფექტურ მოქმედებას შეუძლია ზარალისა და დანაკარგების შემცირება, მეორე შემთხვევაში კი პირიქით.

სისტემების საიმედოობისა და უსაფრთხოების შეფასებისას არ უნდა მოხდეს ცნებების, მაგალითად, საიმედოობისა, სიცოცხლისუნარიანობის და უსაფრთხოების აღრევა. საიმედო სისტემა შეიძლება აღმოჩნდეს არასიცოცხლისუნარიანი და საფრთხის შემცველი. ამგვარად უსაფრთხოება არ უნდა იქნას განხილული, როგორც საიმედოობის ერთ–ერთი კომპონენტი და მისი შემადგენელი. უარი უნდა ვთქვათ „აბსოლუტურად“

უსაფრთხო სისტემის კონცეფციაზე. უნდა დამუშავდეს და განხორციელდეს ავარიის რისკების შეფასების მოდელები. ასევე აუცილებელია ავარიული სიტუაციების მეცნიერულად დასაბუთებული ახალი მეთოდებისა და ტექნიკური საშუალებების შემუშავება.

ელემენტების საიმედოობის გაზრდით, სტრუქტურული და დროითი სიჭარბის შექმნით, ასევე ურთიერთშენაცვლებისა და აღდგენადობის თვისებების საშუალებებით ჩვენ უზრუნველვყოფთ რთული სისტემის მტყუნებამდგრადობის ამაღლებას. მაგრამ სწორედ რთულ სისტემებს ახასიათებთ ელემენტთა ფუნქციონირების ნორმალური რეჟიმიდან მცირედი გადახვევები, რომლებიც დროთა განმავლობაში უახლოვდებიან კრიტიკულ ზღვარს და იწვევენ ავარიული სიტუაციების შექმნას.

რთული სისტემების საიმედოობის რეალური მოდელების განხილვისას, როგორც წესი მხედველობაში იღებენ ელემენტების შემადგენლობას, მათ შორის ურთიერთკავშირებს, სამუშაო რეჟიმს და ა.შ. უსაფრთხოების განხილვისას წინა პლანზე გამოდის ისეთი კომპონენტები, რომლებსაც არ ითვალისწინებს საიმედოობის თეორია: გარემო, რომელშიც ფუნქციონირებს სისტემა, დამცავი ნაგებობები და საშუალებები, გარემო ფაქტორების ზემოქმედება, ადამიანთა (პერსონალის) წინასწარგანზრახული ან უპასუხისმგებლო მოქმედება და სხვ.

მაღალი პასუხისმგებლობის რთული სისტემების დაპროექტებისას და მათი მართვის ალგორითმების შემუშავებისას აუცილებელია ძალიან დაწვრილებით იქნას განხილული და აღწერილი სახიფათო მდგომარეობების სიმრავლე და მათი წარმოქმნის ლოგიკა. აუცილებელია ზუსტად ვიცოდეთ, რა პირობებში წარმოიქმნება აფეთქების, ხანძრის, ნგრევის, გერმეტიზაციის დარღვევის ა და სხვა მსგავსი სახიფათო სიტუაციები. ეს საშუალებას მოგვცემს ერთის მხრივ წინასწარ მივიღოთ დაცვის ზომები, მეორეს მხრივ კი შევიმუშაოთ სისტემის უსაფრხო მართვის ალგორითმი.

შესაბამისად, უნდა შემუშავდეს ისეთი მათემატიკური მოდელი, რომელიც მოგვცემს საშუალებას კომპიუტერის დახმარბით გავათამაშოთ ყველა შესაძლო ავარიული სიტუაცია. სრული და ობიექტური საწყისი მონაცემების არარსებობის შემთხვევაში შესაძლებელია ავარიული სიტუაციის განვითარების ალბათობის სცენარის შემუშავება. შესაბამისად, სცენარი უნდა შემუშავდეს არა „ქვევიდან–ზემოთ“, როცა ვიწყებთ განხილვას თითოეული ელემენტის მტყუნებით, თუ რა შედეგამდე მიგვიყვანს ეს მტყუნება, არამედ „ზევიდან–ქვემოთ“, როცა განხილვას ვიწყებთ სახიფათო

მდგომარეობით და შემდეგ ვიკვლევთ მის გამომწვევ მიზეზებს (თუ რამ მიგვიყვანა ამ სახიფათო მდგომარეობამდე).

სისტემის ქმედითუანრიანობის პირობის აღწერა. კომპიუტერული ქსელის საიმედოობის, უსაფრთხოების, რისკის შეფასებისა და ანალიზისას, როგორც წესი, ყურადღება ექცევა სისტემის შემადგენელ ელემენტებს, კავშირებს მათ შორის, სისტემის მუშაობის რეჟიმებს და ა. შ. აქედან გამომდინარე, საჭიროა თავდაპირველად მოხდეს სისტემის ქმედითუნარიანობის პირობის აღწერა.

სისტემის ქმედითუნარიანობის პირობის აღწერა შეიძლება შესრულებული იქნას სხვადასხვა სახით:

1. სიტყვიერი ფორმით;
2. გრაფიკული (ე.წ. სისტემის სტრუქტურული სქემის დახმარებით);
3. ფორმალიზებული (მაგ. ლოგიკურ-ალგებრული ფუნქციბის დახმარებით);

სისტემის ქმედითუნარიანობის სიტყვიერი აღწერა ყველაზე მეტად გავრცელებული და მარტივი ფორმაა, მაგრამ საკმაოდ მოცულობითი და შრომატევადია, ამასთან არასაკმარისად თვალსაჩინო და ზუსტია. გრაფიკული აღწერა სისტემის სტუქტურული სქემის საშუალებით შედარებით თვალსაჩინოა, მაგრამ არასრულყოფილი და არაერთგვაროვანია. ფორმალიზებული აღწერა ლოგიკურ-ალგებრული ფუნქციების დახმარებით წარმოადგენს ყველაზე მკაფიო, სრულყოფილი და ერთგვაროვან ფორმას, მაგრამ ნაკლებად არის გავრცელებული.

კომპიუტერული ქსელი მიეკუთვნება რთული სტრუქტურის სისტემათა კლასს. რთული სისტემების სტრუქტურული საიმედოობის გამოკვლევისას მიზანშეწონილია სისტემის ქმედითუანრიანობის პირობის აღწერა ყველა ზემოთ მოცემული მეთოდის გამოყენებით, რათა ცალკეული მეთოდების ნაკლოვანებების ურთიერთ კომპესაცია მოხდეს.

სისტემის ქმედითუანრიანობა აღვწეროთ ლოგიკურ-ალბათური ფუნქციების დახმარებით, რადგან ეს მეთოდი ყველაზე მეტად მოსახერხებელია და მკაფიოდ აღწერს რთული სტრუქტურის მქონე სისტემის სტრუქტურას და მისი მუშაობის რეჟიმებს.

სისტემის შემადგენელი ყველა ელემენტი ავღნიშნოთ x_i -ურით, სადაც

$$x_i = \begin{cases} 1, \text{თუ } x_i \text{ ელემენტი ქმედითუნარიანია} \\ 0, \text{თუ } x_i \text{ ელემენტი მტყუნებულია (არაქმედითუნარიანია)} \end{cases}$$

ამ პირობის თანახმად, x_i აღნიშნავს i -ური ელემენტის ქმედითუნარიან მდგომარეობას, x_i' - i -ური ელენტის მტყუნებას.

ორობითი ცვლადების $x_1, x_2, \ldots, x_i, \ldots, x_m$ კონკრეტული მნიშვნელობები განსაზღვრავს ე.წ. სისტემის მდგომარეობის ვექტორს $x = (x_1, x_2, \ldots, x_i, \ldots, x_m)$.

ამასთანავე იგულისხმება, რომ სისტემის მოქმედება (მუშაობა) დეტერმინებულად დამოკიდებულია მისი შემადგენელი ელემენტების მოქმედებაზე. ლოგიკურ-ალგებრულ ფუნქციას, რომელის ელემენტების მდგომარეობას აკავშირებს სისტემის მდგომარეობასთან $y(x_1, x_2, \ldots, x_i, \ldots, x_m) = y(x)$ ეწოდება სისტემის ქმედითუანრიანობის ფუნქცია.

ისეთი სისტემებისათვის, სადაც მტყუნებული ელემენტის შეცვლა ქმედითუნარიანი ელემენტით არ იწვევს სისტემის მტყუნებას, ქმედითუნარიანობის ფუნქცია წარმოადგენს მონოტონურ ფუნქციას, ხოლო სისტემებს, რომლებიც აკმაყოფილებენ აღნიშნულ პირობას, ეწოდება მონოტონური სისტემები.

აღსანიშნავია, რომ ყველა ლოგიკურ-ალბათური ფუნქცია, რომელიც ჩაწერილია კონიუნქციის ან დიზიუნქციის (არა უარყოფით) საშუალებით, წარმოადგენს მონოტონურ ფუნქციას.

მონოტონური სტრუქტურის მქონე სისტემების ქმედითუნარიანობის ფუნქცია $y(x)$ შესაძლებელია ჩაიწეროს ე.წ. ფუნქციონირების უმოკლესი გზების და მტყუნების მინიმალური ხდომილებების საშუალებით.

სისტემის ფუნქციონირების უმოკლესი გზები წარმოადგენს მისი ელემენტების ისეთ კონიუნქციას, რომელთაგან არცერთი კომპონენტის ამოღება რა შეიძლება, ისე რომ არ დაირღვეს სისტემის ფუნქციონირების პირობა. ასეთი კონიუნქცია ლოგიკურ-ალბათური ფუნქციის დახმარებით ჩაიწერება შემდეგი სახით:

$$K_j = \underset{i \in \Pi_{K_j}}{\wedge} x_i \qquad (4.1)$$

სადაც K_{Π_l} სისტემის ფუნქციონიების უმოკლესი გზის ელემენტთა ნომრების სიმრავლეა. სხვაგვარად რომ ვთქვათ, სისტემის ფუნქციონირების უმოკლესი გზა აღწერს სისტემისადმი წაყენებული ამოცანებიდან ერთ-ერთ შესაძლებელ დამოუკიდებელ ვარიანტს, ქმედითუნარიანი ელემენტების მინიმალური რაოდენობის დახმარებით, რომელიც აუცილებელია რათა შესრულდეს სისტემისადმი წაყენებული ამოცანის მოცემული ვარიანტი.

სისტემის მტყუნების მინიმალური გზები წარმოადგენენ სისტემის მტყუნებული ელემენტების ისეთ კონიუნქციას, რომლთაგან არცერთი კომპონენტის ამოღება არ

შეიძლება ისე, რომ არ დაირღვეს სისტემის მწყობრიდან გამოსვლის პირობა. ასეთი კონიუნქციის ჩაწერა ლოგიკურ-ალბათური ფუნქციის დახმარებით შესაძლებელია შემდეგი სახით:

$$S_i = \underset{i \in K_{S_j}}{\wedge} x_i' \tag{4.2}$$

სადაც K_{s_j} სისტემის მტყუნების მინიმალური გზის ელემენტთა ნომრების სიმრავლეა. სხვაგვარად რომ ვთქვათ, სისტემის მტყუნების მინიმალური გზა აღწერს სისტემის ქმედითუნარიანობის დარღვევის ერთ-ერთ შესაძლო საშუალებას, მტყუნებული ელემენტების მინიმალური რაოდენობის დახმარებით.

რამდენადაც რთული არ უნდა იყოს სისტემა, ყველა მათგანს გააჩნია ფუნქციონირების უმოკლესი გზების $(l = 1,2,\ldots,d)$ და მტყუნების მინიმალური ხდომილებების $j = (1,2,\ldots,n)$ სასრული რიცხვი. ამ ცნებების გამოყენებით შესაძლებელია სისტემის ქმედითუნარიანობის ჩაწერა სახავდასხვა სახით:

1. ფუნქციონირების უმოკლესი გზების ყველა შესაძლო ვარიანტების დიზიუნქციის სახით

$$y(x_1,\ldots,x_m) = \overset{d}{\underset{l=1}{\vee}} \Pi_l = \overset{d}{\underset{l=1}{\vee}} \left[\underset{i \in K_l}{\wedge} x_i \right] \tag{4.3}$$

2. ყველა შესაძლო მტყუნების მინიმალური გზების უარყოფის კონიუნქციის სახით

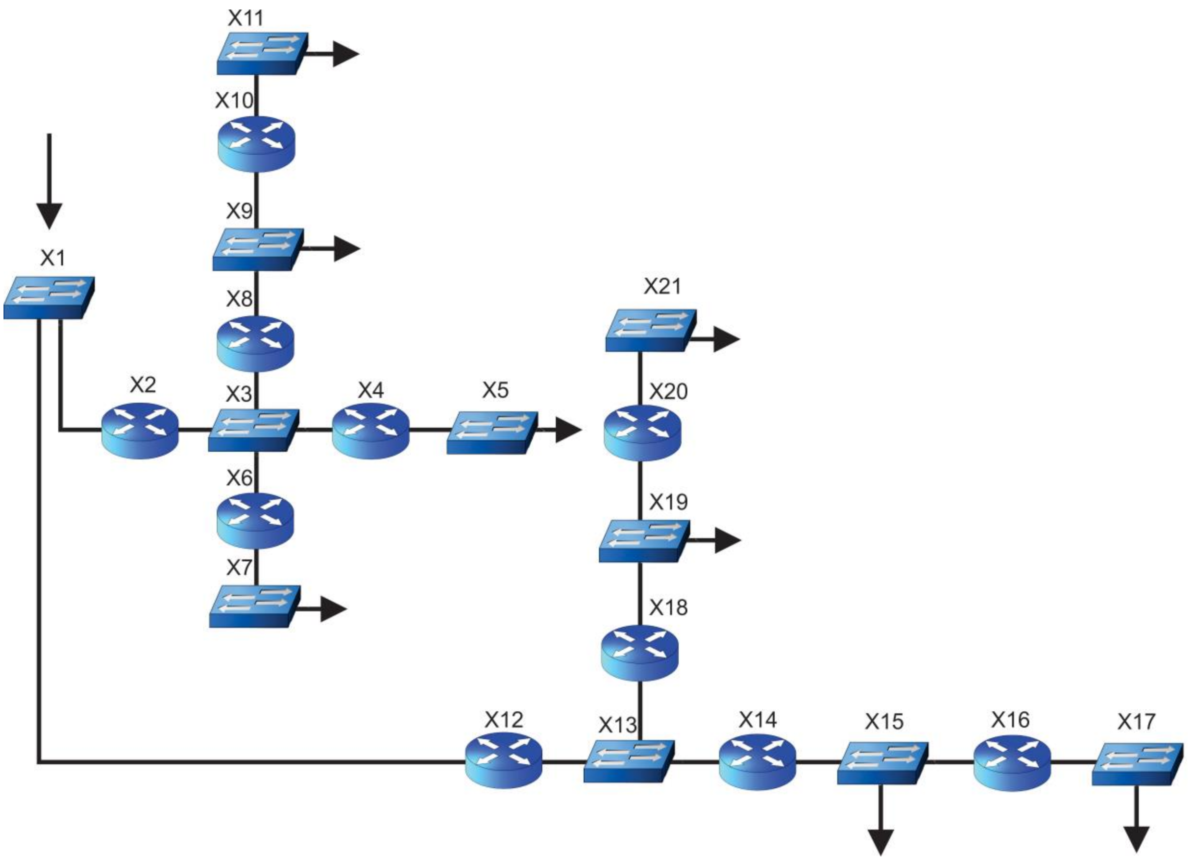
$$y(x_1,\ldots,x_m) = \overset{n}{\underset{j=1}{\wedge}} S_j' = \overset{n}{\underset{j=1}{\wedge}} \left[\underset{i \in K_{s_j}}{\vee} x_i \right] \tag{4.4}$$

ე.ი. რეალური სისტემის ქმედითუნარიანობის პირობა შეიძლება წარმოვადგინოთ ფუნქციონირების გზების პარალელური ან მტყუნების მინიმალური ხდომილებების მიმდევრობითი შეერთებების სტრუქტურის სახით.

ილუსტრაცია N 29_ზე მოცემული სისტემის ქმედითუნარიანობის პირობას, რომელიც აღწერილია ლოგიკურ-ალგებრული ფუნქციების, კერძოდ ფუნქციონირების უმოკლესი გზების დახმარებით მატრიცული ფორმით, აქვს შემდეგი სახე:

$$y(x_1,\ldots,x_{21}) = \begin{vmatrix} x_1x_2x_3x_4x_5 \\ x_1x_2x_3x_6x_7 \\ x_1x_2x_3x_8x_9 \\ x_1x_2x_3x_8x_9x_{10}x_{11} \\ x_1x_{12}x_{13}x_{14}x_{15} \\ x_1x_{12}x_{13}x_{14}x_{15}x_{16}x_{17} \\ x_1x_{12}x_{13}x_{18}x_{19} \\ x_1x_{12}x_{13}x_{18}x_{19}x_{20}x_{21} \end{vmatrix} = \begin{vmatrix} \Pi_1 \\ \Pi_2 \\ \Pi_3 \\ \Pi_4 \\ \Pi_5 \\ \Pi_6 \\ \Pi_7 \\ \Pi_8 \end{vmatrix} \quad (4.5)$$

ილუსტრაცია N 29. კომპიუტერული ქსელის სტრუქტურული სქემა.



წყარო: (კვლევითი მასალები)

მართლაც, (4.5) გამოსახულებებიდან თვალნათლივ ჩანს, რომ აღნიშნული სისტემა სრული ფუნქციონირების მდგომარეობაშია, როცა გამართულია ფუნქციონირების ყველა

გზა. თუ ერთ-ერთი გზა გამოვა მწყობრიდან სისტემა გადადის ნაწილობრივი ფუნქციონირების მდგომარეობაში, როცა არცერთი გზა არ ფუნქციონირებს სისტემა სრული მტყუნების მდგომარეობაშია.

4.2 *სისტემის უმტყუნო მუშაობის ალბათობის გამოთვლა კვეთის ალგორითმის გამოყენებით*

კომპიუტერული ქსელის საიმედოობისა და უსაფრთხოების ანალიზისა და შეფასებისას წარმოქმნილი უამრავი საკითხების გამოკვლევა და გადაჭრა შესაძლებელია ლოგიკურ-ალბათური მოდელების გამოყენებით.

ალბათობის თეორია რაოდენობრივად აფასებს სისტემის საიმედოობასა და უსაფრთხოებას, ხოლო მათემატიკური ლოგიკის საშუალებით ხდება სისტემის სტრუქტურის აღწერა.

კვეთის ალგორითმი ლოგიკური ფუნქციის დაშლის თეორემაზეა დაფუძვნებული, რომელის მდგომარეობს შემდეგში: n არგუმენტზე ($n \geq 1$) დამოკიდებული ნებისმიერი ლოგიკური ფუნქცია შეიძლება ჩაიწეროს შემდეგი სახით

$$f(x_1,...,x_i,x_{i+1},...,x_n) = \vee x_1^{a_1} x_2^{a_2} ... x_i^{a_i} f(a_1,a_2,...,a_i,x_{i+1},...,x_n) \qquad (4.6)$$

ამ თეორემას ეწოდება დაშლის თეორემა, ვინაიდან ის იძლევა ლოგიკური ფუნქციის ნებისმიერი i -ური არგუმენტით დაშლის საშუალებას. ამდენად, ლოგიკური ფუნქციის თითეული x_i -ური არგუმენტით დაშლის შემთხვევაში მივიღებთ:

$$f(x_1,...,x_n) = x_i f_1^{(i)}(x_1,...,x_n) \vee x_i' f_0^{(i)}(x_1,...,x_n) \qquad (4.7)$$

ფუნქციის ყველა n არგუმენტით დაშლის შემთხვევაში მივიღებთ:

$$f(x_1,...,x_n) = \underset{1}{\vee} x_1^{a_1} x_2^{a_2} ... x_n^{a_n} \qquad (4.8)$$

სიმბოლო $\underset{1}{\vee}$ აღნიშნავს, რომ დიზიუნქცია აიღება მხოლოდ იმ $< a_1, a_2, ..., a_n >$ ნაკრებისათვის, რომლებზედაც სრულდება ტოლობა:

$$f(a_1,a_2,...,a_n) = 1 \qquad (4.9)$$

დაშლის აღნიშნულ ფორმულებს ეწოდება შენონის დაშლის ფორმულები. ლოგიკური ფუნქციის დაშლის თეორემაზეა დაფუძვნებული სისტემის სტრუქტურული საიმედოობის შეფასების კვეთის ალგორითმი, რომელიც მდგომარეობს შემდეგში:

1. უნდა განისაზღვროს სისტემის ქმედითუნარიანობის ფუნქციაში $y(x_1,\ldots,x_n)$ შემავალი თითეული x_i ელემენტის რაოდენობა

$$(m_1, m_2, \ldots, m_n) = \{m_i\} \qquad (4.10)$$

2. მიღებული მნიშვნელობებიდან ვპოულობთ მაქსიმალური მნიშვნელობას და მის შესაბამის არგუმენტს ენიჭება თავდაპირველად 0, შემდეგ 1 და ორივე შემთხვევისათვის ცალცალკე იწერება $y(x_1,\ldots,x_n)$ ფუნქციაში შესბამისი კონსტანტების ჩასმით მიღებულ შედეგი.

მაგალითად, დავუშვათ $y(x_1,\ldots,x_n)$ ფუნქციაში მაქსიმალური რაოდენობით წარმოდგენილია x_1 ელემენტი, ამიტომ მოვახდინოთ ფუნქციის კვეთა ამ არგუმენტის მიხედვით. ვიხილავთ ორ შემთხვევას:

$$x_1 = 0 \quad y_0 = y(0, x_2, \ldots, x_n) = y_0(x_2, \ldots, x_n) \qquad (4.11)$$

$$x_1 = 1 \quad y_1 = y(1, x_2, \ldots, x_n) = y_1(x_2, \ldots, x_n) \qquad (4.12)$$

წყარო: (ცირამუა, 2014).

3. შემდეგ ეტაპზე უნდა მოვახდინოთ მიღებული y_0 და y_1 ფუნქციების გარდაქმნა შემდეგი ლოგიკური ოპერციების საფუძველზე

1. $x_i \wedge 1 = x_i$
2. $x_i \wedge 0 = 0$
3. $x_i \wedge x_i = x_i$
4. $x_i \wedge x_i' = 0$
5. $x_i \vee 1 = 1$
6. $x_i \vee 0 = x_i$
7. $x_i \vee x_i = x_i$
8. $x_i \vee x_i' = 1$

4. y_0 და y_1 ფუნქციების გარდაქმნებისა და გამარტივების შედეგად შეიძლება მივიღოთ რომ:

ა) y_0 ან y_1 ფუნქცია გარდაიქმნას მუდმივად (კონსტანტად);

ბ) y_0 ან y_1 ფუნქციამ შეიძლება მიიღოს ისეთი სახე, სადაც გარდაქმნების შედეგად დარჩენილი თითოეული არგუმენტი ფუნქციაში წარმოდგენილი იქნება არაუმეტეს ერთისა (არ მოხდება არგუმენტების განმეორება).

გ) გარდაქმნილ y_0 ან y_1 ფუნქციაში თუნდაც ერთი არგუმენტი წარმოდგენილ იქნას ერთზე მეტჯერ (მოხდეს არგუმენტების განმეორება).

ვამოწმებთ აღნიშნული სამი ვარიანტიდან (ა, ბ, გ) რომელ მდგომარეობაშია y_0 და y_1 ფუნქციები.

5. შემოწმების შედეგად თუ მივიღებთ რომ, ადგილი აქვს მესამე (გ) შემთხვევას, მაშინ შესაბამისი ფუნქციისათვის ისევ ვითვლით $\{m_i\}$ მნიშვნელობას ყველა დარჩენილი არგუმენტებისათვის და ვაწარმოებთ ფუნქციის კვეთას მაქსიმალური m_i -ურის შესაბამისი არგუმენტით.

მაგალითად, დავუშვათ ფუნქციაში მაქსიმალური რაოდენობით წარმოდგენილია x_2 ელემენტი და მოვახდინოთ ფუნქციის კვეთა ამ არგუმენტის მიხედვით.

თუ კვეთას საჭიროებს y_0 ფუნქცია, მივიღებთ:

$$y_{00} = y_0(0, x_3,\ldots,x_n) = y_{00}(x_3,\ldots,x_n)$$

$$y_{01} = y_0(1, x_3,\ldots,x_n) = y_{01}(x_3,\ldots,x_n)$$

თუ კვეთას საჭიროებს y_1 ფუნქცია, მაშინ

$$y_{10} = y_1(0, x_3,\ldots, x_n) = y_{10}(x_3,\ldots, x_n)$$

$$y_{11} = y_1(1, x_3,\ldots,x_n) = y_{11}(x_3,\ldots,x_n)$$

მოვახდენთ რა მიღებული ფუნქციების გარდაქმნას ლოგიკური ოპერციების საფუძველზე, ვიმეორებთ მე-4 და მე-5 ეტაპებს მანამ, სანამ კვეთის შედეგად მიღებული ფუნქციები არ დააკმაყოფილებენ მე-4 ეტაპზე მოცემულ ა) ან ბ) შემთხვევას. ე.ი. ისინი უნდა გარდაიქმნან მუდმივებად ან ისეთ ფუნქციად, სადაც თითეული არგუმენტი წარმოდგენილი იქნება არაუმეტეს ერთისა.

საბოლოოდ ყველა კვეთისგან მიღებული ფუნქციის სისტემის ქმედითუნარიანობის პირობის $y(x_1,\ldots,x_n)$ ფუნქციაში ჩასმით მივიღებთ:

$$y(x_1,\ldots,x_n)=\begin{vmatrix} x_1' y_0 \\ x_1 y_1 \end{vmatrix} = \begin{vmatrix} x_1' & \begin{vmatrix} x_2 y_{01} \\ x_2' y_{00} \end{vmatrix} \\ x_1 & \begin{vmatrix} x_2 y_{11} \\ x_2' y_{10} \end{vmatrix} \end{vmatrix} = \begin{vmatrix} x_1' & \begin{vmatrix} x_2 y_{01} \\ x_2' y_{00} \end{vmatrix} & \\ x_1 & x_2 y_{11} & \\ & x_2' & \begin{vmatrix} x_3 y_{101} \\ x_3' y_{100} \end{vmatrix} \end{vmatrix} \ldots \quad (4.13)$$

4.13 გამოსახულების გასამარტივებლად შესაძლებელი მისი ჩაწერა H_i შეუთავსებადი (ორთოგონალური) ჰიპოთეზების საშუალებით:

$$y(x_1,\ldots,x_n)=\begin{vmatrix} x_1' & \begin{vmatrix} x_2 y_{01} \\ x_2' y_{00} \end{vmatrix} & \\ x_1 & x_2 y_{11} & \\ & x_2' & \begin{vmatrix} x_3 y_{101} \\ x_3' y_{100} \end{vmatrix} \end{vmatrix} = \begin{vmatrix} x_1' x_2 y_{01} \\ x_1' x_2' y_{00} \\ x_1 x_2 y_{11} \\ x_1 x_2' x_3 y_{101} \\ x_1 x_2' x_3' y_{100} \end{vmatrix} = \begin{vmatrix} H_1 y_{01} \\ H_2 y_{00} \\ H_3 y_{11} \\ H_4 y_{101} \\ H_5 y_{100} \end{vmatrix} \quad (4.14)$$

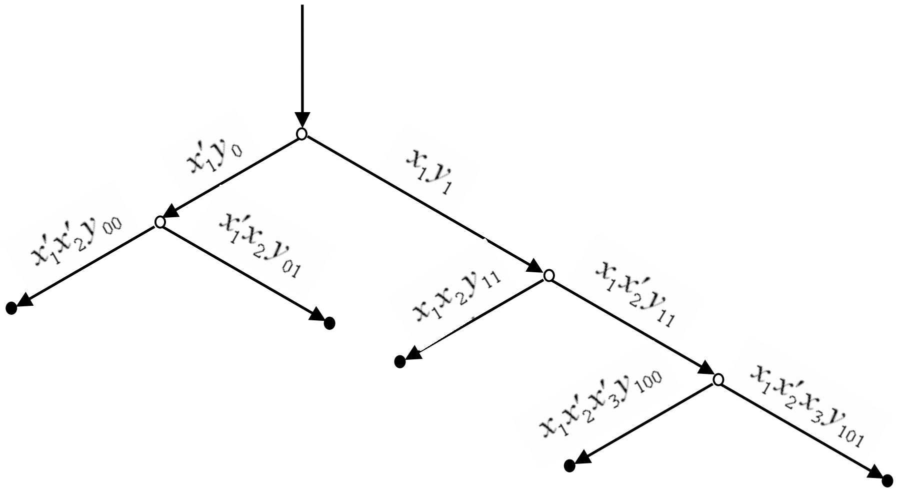
სისტემის უმტყუნო მუშაობის ალბათობა გამოითვლება შემდეგი ფორმულით:

$$P\{y(x_1,\ldots,x_n)=1\}=R_c=\sum_{i=1}^{m} P(H_i)P(y|H_i)$$

სადაც, H_i მოვლენები ქმნიან შეუთავსებადი (ორთოგონალური) ჰიპოთეზების მთლიან ჯგუფს, ხოლო $P(y|H_i)$ წარმოადგენს სისტემის გამოსწორებადი მდგომარეობის პირობით ალბათობას, ყოველი H_i ჰიპოთეზის შემთხვევაში.

კვეთის ალგორითმის გამოყენებით სისტემის საიმედოობის შეფასების დროს მიზანშეწონილია სისტემის მდგომარეობის გრაფის აგება, რომელიც უზრუნველყოფს ჩატარებული ოპერაციების თვალსაჩინოებასა და თანმიმდევრობას (ცირამუა, 2014.

ილუსტრაცია N 29. კვეთის ალგორითმის გამოყენების თანმიმდევრობა.



წყარო: (კვლევითი მეთოდები)

4.3. სისტემის უმტყუნო მუშაობის ალბათობის გამოთვლის ცხრილური მეთოდი

სისტემის უმტყუნო მუშაობის ალბათობის გამოთვლის ცხრილური მეთოდი დაფუძნებულია ერთობლივი მოვლენების ალბათობათა შეკრების თეორემებზე. ალბათობათა შეკრების თეორემის თანახმად ორი ნებისმიერი ხდომილების ჯამი გამოითვლება ფორმულით:

$$P(A+B)=P(A)+P(B)-P(AB) \qquad (4.15)$$

ხოლო ნებისმიერი $A_1, A_2, \ldots A_n$ ხდომილებისათვის ადგილი აქვს ალბათობათა შეკრების თეორემას

$$P(\sum_{i=1}^{n} A_i)=\sum_{i} P(A_i)-\sum_{i,j} P(A_i A_j)+\sum_{i,j,k} P(A_i A_j A_k)-\ldots+(-1)^{n-1}\cdot P\left(A_1 A_2 \ldots A_n\right) \qquad (4.16),$$

სადაც ჯამები გავრცელებულია სათითაოდ, წყვილ-წყვილად, სამ-სამად და ამრიგად აღებული სხვადასხვა $i, j, k, \ldots$ ინდექსების ყველა შესაძლო კომბინაციებზე. საინფორმაციო-საკომუნიკაციო სისტემების უმტყუნო მუშაობის ალბათობის გამოთვლის შემთხვევაში ერთობლივ მოვლენებს წარმოდგენენ სისტემის ქმედითუარიანობის ფუნქციის (რომელიც ფუნქციონირების უმოკლესი გაზების დახმარებით ჩაწერილია ნორმალური დიზიუნქციურ ფორმაში) ელემენტარული კონიუნქციები.

$$y\left(x_1, x_2, \ldots, x_n\right)=\bigvee_{l=1}^{d} \Pi_l \qquad (4.16)$$

ალბათობათა შეკრების თეორემისა და 4.16 გამოსახულების თანახმად კომპიუტერული ქსელის უმტყუნო მუშაობის ალბათობა გამოითვლება შემდეგი ფორმულით:

$$P\{y(x_1,\ldots,x_n=1)\}=R_c=P\left\{\bigvee_{l=1}^{d}\Pi_l\right\}=\sum_i P(\Pi_i)-\sum_i\sum_j P(\Pi_i\wedge\Pi_j)+$$
$$+\sum_i\sum_j\sum_k P(\Pi_i\wedge\Pi_j\wedge\Pi_k)-\ldots+(-1)^{d-1}P(\Pi_1\wedge\Pi_2\wedge\ldots\wedge\Pi_d) \qquad (4.17)$$

განხილული მეთოდის საშუალებით სისტემის საიმედოობის გამოთვლა საკმაოდ მარტივი და ადვილად კონტროლირებადია, თუ გამოთვლებს ვაწარმოებთ ცხრილური ფორმით, რაც განსაზღვრავს აღნიშნული მეთოდის სახელწოდებას.
აღნიშნული ცხრილური მეთოდის თანახმად აუცილებელია სპეციალური ცხრილის აგება (შექმნა), რომელიც შედგება n სტრიქონისაგან (სისტემაში შემავალი ელემენტების რიცხვის შესაბამისად) და C სვეტისაგან, ამასთან

$$C = C_d^1 + C_d^2 + ... + C_d^k + ... + C_d^d \qquad (4.17)$$

სადაც C_d^k არის d -დან k -ური შეერთებების რიცხვია.

ცხრილში სტრიქონის დასახელებად მიეთითება ელემენტის უმტყუნო მუშაობის ალბათობა R_{x_i} , ხოლო სვეტების დასახელებად იწერება Π_l კონიუნქციები. ყველა შესაძლო შეერთება აღებული, სათითაოდ, წყვილ-წყვილად, სამსამად და ა.შ. გარდა ამისა 4.17 ფორმულის თანახმად თანმიმდევრულად მიეთითება კონიუნქციების ალბათობათა ნიშანი ("+"ან "-"). აღნიშნული ცხრილის აგების შემდეგ საჭიროა მისი შევსება 1-ით ან 0-ით, სადაც 1-ით აღინიშნება იმ მოვლენების ალბათობა, რომლებიც შედიან მოცემულ კონიუნქციეში, 0-ით იმ მოვლენების ალბათობა, რომლებიც არ არის შესაბამის კონიუნქციაში.

ცხრილის შევსების შემდეგ შესაძლებელია სისტემის უმტყუნო მუშაობის ალბათობის გამოთვლა. გადავამრავლოთ თოთეულ სვეტში ის ალბათობები R_{x_i} რომლებიც აღნიშნულია 1-ით და სისტემის უმტყუნო მუშაობის ალბათობის გამოსათვლელ ფორმულაში შევიტანოთ შესაბამისი ნიშნით. ყველა ელემენტის ერთნაირი საიმედოობის შემთხვევაში თითეულ სვეტში 1-იანების რაოდენობა წარმოადგენს ელემენტის უმტყუნო მუშაობის ალბათობის (R -ის) ხარისხს.

4.4.

R_{x_i}	$П_1$	$П_2$	…	$П_n$	$П_1 П_2$	$П_1 П_3$	…	$П_1 П_n$	$П_2 П_3$	$П_2 П_4$	…	$П_2 П_n$	…	$П_{n-1} П_n$	$П_1 П_2 П_3$	…	$П_1 П_2 П_n$	$П_1 П3 П_4$	…	$П_1 П3 П_n$	…	$П_1 … П_i … П_n$
	„+“				„-“										„+“							
R_{x_1}	1	1		1	1	1		1	1	1		1		1	1		1	1		1		1
R_{x_2}	1	1		0	1	0		1	0	1		1		0	1		1	1		1		1
R_{x_3}	0	1		0	1	0		0	0	0		1		0	1		1	0		0		1
R_{x_4}	0	0		1	0	1		1	0	0		1		1	1		0	0		1		1
R_{x_5}	1	0		1	1	1		1	1	0		1		1	1		1	1		1		1
…																						
R_{x_n}	0	0		1	0	0		0	1	1		1		0	0		0	1		1		1
R_C																						

წყარო: (ცირამუა, 2014).

სისტემის უმტყუნო მუშაობის ალბათობის გამოთვლა ორთოგონალიზაციის ალგორითმის გამოყენებით

ორთოგონალიზაციის ალგორითმი ეფუძვნება ლოგიკურ-ალბათური ფუნქციის გარდაქმნას ორთოგონალური დიზიუნქციის ნორმალური ფორმით. ავღნიშნოთ, რომ ორ ელემენტარულ კონიუნქციას ეწოდება ორთოგონალური, თუ მათი ნამრავლი ნულის ტოლია. ხოლო დიზიუნქციის ნორმალურ ფორმას ეწოდება ორთოგონალური, თუ მისი ყველა წევრი წყვილ-წყვილად ორთოგონალურია.

ორთოგონალური დიზიუნქციის ნორმალური ფორმის მიღების მიზნით შემუშავებულია სპეციალური ალგორითმი, რომლის აღწერაც მოცემულია შემდეგ ორ განმარტების სახით:

r თანრიგის მქონე ელემენტარული კონიუნქციის $K_i = x_1^{a_1} x_2^{a_2} \dots x_r^{a_r}$ უარყოფა ექვივალენტურია დიზიუნქციის, რომლის წევრებიც წყვილ-წყვილად ორთოგონალურია.

$$K_i' = x_1^{a'_1} \vee x_1^{a_1} x_2^{a'_2} \vee \dots \vee x_1^{a_1} x_2^{a_2} \dots x_{r-1}^{a_{r-1}} x_r^{a'_r} \quad (4.18)$$

კერძო შემთხვევაში, როცა ელემენტარული კონიუნქცია არ შეიცავს უარყოფას, (4.18) ფორმულა შეიძლება ჩავწეროთ შემდეგი სახით:

$$(x_1 x_2 \dots x_r)' = \begin{vmatrix} x_1' \\ x_2' \\ x_3' \\ \dots \\ x_r' \end{vmatrix} = \begin{vmatrix} x_1' \\ x_1 x_2' \\ x_1 x_2 x_3' \\ \dots \\ x_1 x_2 x_3 x_4 \dots x_{r-1} x_r' \end{vmatrix} \quad (4.19)$$

$f(x_1, x_2, \dots, x_m)$ ბულის ფუნქცია, რომელიც წარმოდგენილია დიზიუნქციურ ნორმალური ფორმის სახით

$$f(x_1, x_2, \dots, x_m) = \bigvee_{i=1}^{n} K_i \left(i \le 2^m\right) \quad (4.20)$$

ექვივალენტურია ფუნქციის

$$f(x_1, x_2, \dots, x_m) = K_1 \vee K_1' K_2 \vee K_1' K_2' K_3 \vee \dots \vee K_1' K_2' \dots K_{n-1}' K_n \quad (4.21)$$

(4.20) და (4.21) გამოსახულებების მატრიცული ფორმით ჩაწერა შესაძლებელია შემდეგი სახით:

$$f(x_1,x_2,...,x_m)=\begin{vmatrix}K_1\\K_2\\K_3\\...\\K_n\end{vmatrix}=\begin{vmatrix}K_1\\K_1'K_2\\K_1'K_2'K_3\\...\\K_1'K_2'K_3'...K_{n-1}'K_n\end{vmatrix} \tag{4.22}$$

თუ 4.22 გამოსხულებაში თითოეული $K_i'(i \le n)$ წევრის მაგივრად ჩავსვამთ მათ შესაბამის 4.18-ში მოცემულ მნიშვნელობებს, საბოლოოდ მივიღებთ ბულის $f(x_1,x_2,...,x_m)$ ფუნქციის ორთოგონალური დიზიუნქციის ნორმალურ ფორმას.

ზემოთ მოყვანილი განმარტებების საფუძველზე შეგვიძლია ჩამოვაყალიბოთ $y(x_1,x_2,...,x_m)$ ფუნქციის ორთოგონალური დიზიუნქციის ნორმალური ფორმით წარმოდგენის ალგორითმი, რომელსაც აქვს შემდეგი სახე:

1. თავდაპირველად $y(x_1,x_2,...,x_m)$ ფუნქცია უნდა გარდავქმნათ დიზიუნქციურ ნორმალურ ფორმად;

2. ვაწარმოოთ მიღებული დიზიუნქციის გადანომვრა 1-დან n -მდე $(n<2^m)$, ამასთავავე დიზიუნქციის მაღალი თანრიგის წევრს მივანიჭოთ დაბალი ნომერი.

3. განვსაზღვროთ $y(x_1,x_2,...,x_m)$ ფუნქციის ორთოგონალური დიზიუნქციის ნორმალური ფორმა 4.22 გამოსახულების დახმარებით.

შესასრულებელი ოპერაციების რიცხვის მინიმიზაციისათვის მიზანშეწონილია $K_1'K_2'...K_{i-1}'K_i'$ კონიუნქციაში შევასრულოთ შემდეგი გამარტივებები:

ა) დიზიუნქციის $K_j(j \le i-1)$ წევრები, რომლებიც K_i წევრის ორთოგონალურია გაუტოლოთ ნულს;

ბ) კონიუნქციის უარყოფითი ელემენტები $K_j'(j \le i-1)$, რომლებიც ორთოგონალურია K_i -ს გაუტოლოთ ნულს;

სისტემის ქმედითუნარიანობის პირობის ორთოგონალური დიზიუნქციის ნორმალური ფორმით წარმოდგენის შემდეგ შეგვიძლია სისტემის უმტყუნო მუშაობის ალბათობის გამოთვლა, რომელიც ამ ფუნქციის ყველა ორთოგონალური წევრების ჭეშმარიტობის ალბათობების ჯამის ტოლია.

$$P\{y(x_1,...,x_m)=1\}=R_c=\sum_{i=1}^{S}P(O_i) \tag{4.23}$$

სადაც O_i - ორთოგონალური დიზიუნქციის ნორმალური ფორმით ჩაწერილი $y(x_1, x_2, ..., x_m)$ ფუნქციის ორთოგონალური წევრებია.

$$y(x_1, x_2, ..., x_m) = \bigvee_{i=1}^{n} K_i = \bigvee_{i=1}^{S} O_i \tag{4.24}$$

წყარო: (ცირამუა, 2014).

4.4. ორთოგონალიზაციის მეთოდით სისტემის უმტყუნო მუშაობის ალბათობის რაოდენობრივი შეფასება

ორთოგონალიზაციის ალგორითმის გამოყენებით განვსაზღვროთ N 29 ილუსტრაციაზე მოცემული სისტემის უმტყუნო მუშაობის ალბათობა.

როგორც წინა თავში განვიხილეთ, ფუნქციონირების უმოკლესი გზების დახმარებით მატრიცული ფორმით აღნიშნული სისტემის ქმედითუნარიანობის პირობას აქვს შემდეგი სახე:

$$y(x_1, ..., x_{21}) = \begin{vmatrix} x_1x_2x_3x_4x_5 \\ x_1x_2x_3x_6x_7 \\ x_1x_2x_3x_8x_9 \\ x_1x_2x_3x_8x_9x_{10}x_{11} \\ x_1x_{12}x_{13}x_{14}x_{15} \\ x_1x_{12}x_{13}x_{14}x_{15}x_{16}x_{17} \\ x_1x_{12}x_{13}x_{18}x_{19} \\ x_1x_{12}x_{13}x_{18}x_{19}x_{20}x_{21} \end{vmatrix} = x_1 \begin{vmatrix} x_2x_3x_4x_5 \\ x_2x_3x_6x_7 \\ x_2x_3x_8x_9 \\ x_2x_3x_8x_9x_{10}x_{11} \\ x_{12}x_{13}x_{14}x_{15} \\ x_{12}x_{13}x_{14}x_{15}x_{16}x_{17} \\ x_{12}x_{13}x_{18}x_{19} \\ x_{12}x_{13}x_{18}x_{19}x_{20}x_{21} \end{vmatrix} \tag{4.25}$$

სისტემის ქმედითუნარიანობის მიღებული (4.25)ფუნქციის ცალკეული წევრი გადავნომროთ შემდეგი სახით.

$$\begin{aligned} &K_1 = x_2x_3x_4x_5 && K_2 = x_2x_3x_6x_7 && K_3 = x_2x_3x_8x_9 \\ &K_4 = x_2x_3x_8x_9x_{10}x_{11} && K_5 = x_{12}x_{13}x_{14}x_{15} && K_6 = x_{12}x_{13}x_{14}x_{15}x_{16}x_{17} \\ &K_7 = x_{12}x_{13}x_{18}x_{19} && K_8 = x_{12}x_{13}x_{18}x_{19}x_{20}x_{21} \end{aligned} \tag{4.26}$$

(4.26) აღნიშვნების გათვალისწინებით სისტემის ქმედითუნარიანობის ფუნქცია (4.25) შეიძლება ჩავწეროთ შემდეგი სახით:

$$y(x_1,\ldots,x_{21}) = x_1 \bigvee_{i=1}^{8} K_i = x_1 \left| \begin{array}{l} K_1 \\ K_1' K_2 \\ K_1' K_2' K_3 \\ K_1' K_2' K_3' K_4 \\ K_1' K_2' K_3' K_4' K_5 \\ K_1' K_2' K_3' K_4' K_5' K_6 \\ K_1' K_2' K_3' K_4' K_5' K_6' K_7 \\ K_1' K_2' K_3' K_4' K_5' K_6' K_7' K_8 \end{array} \right| \qquad (4.27)$$

განვსაზღვროთ (4.27)-ში შემავალი კონიუქციები

$$K_1' K_2 = \left| \begin{array}{l} x_2 x_3 x_4' x_6 x_7 \\ x_2 x_3 x_4 x_5' x_6 x_7 \end{array} \right| \qquad K_1' K_2' K_3 = \left| \begin{array}{l} x_2 x_3 x_4' x_6' x_8 x_9 \\ x_2 x_3 x_4 x_5' x_6' x_8 x_9 \\ x_2 x_3 x_4' x_6 x_7' x_8 x_9 \\ x_2 x_3 x_4 x_5' x_6 x_7' x_8 x_9 \end{array} \right| \qquad K_1' K_2' K_3' K_4 = 0$$

$$K_1' K_2' K_3' K_4' K_5 = \left| \begin{array}{l} x_2' x_{12} x_{13} x_{14} x_{15} \\ x_2 x_3' x_{12} x_{13} x_{14} x_{15} \\ x_2 x_3 x_4' x_6' x_8' x_{12} x_{13} x_{14} x_{15} \\ x_2 x_3 x_4 x_5' x_6' x_8' x_{12} x_{13} x_{14} x_{15} \\ x_2 x_3 x_4' x_6 x_7' x_8' x_{12} x_{13} x_{14} x_{15} \\ x_2 x_3 x_4 x_5' x_6 x_7' x_8' x_{12} x_{13} x_{14} x_{15} \\ x_2 x_3 x_4' x_6' x_8 x_9' x_{12} x_{13} x_{14} x_{15} \\ x_2 x_3 x_4 x_5' x_6' x_8 x_9' x_{12} x_{13} x_{14} x_{15} \\ x_2 x_3 x_4' x_6 x_7' x_8 x_9' x_{12} x_{13} x_{14} x_{15} \\ x_2 x_3 x_4 x_5' x_6 x_7' x_8 x_9' x_{12} x_{13} x_{14} x_{15} \end{array} \right| \qquad K_1' K_2' K_3' K_4' K_5' K_6 = 0$$

$$
K_1'K_2'K_3'K_4'K_5'K_6'K_7 = \left|\begin{array}{l}
x_2'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3x_4'x_6'x_8'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3x_4x_5'x_6'x_8'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3x_4'x_6x_7'x_8'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3x_4x_5'x_6x_7'x_8'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3x_4'x_6'x_8x_9'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3x_4x_5'x_6'x_8x_9'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3x_4'x_6x_7'x_8x_9'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3x_4x_5'x_6x_7'x_8x_9'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3x_4'x_6'x_8'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3x_4x_5'x_6'x_8'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3x_4'x_6x_7'x_8'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3x_4x_5'x_6x_7'x_8'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3x_4'x_6'x_8x_9'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3x_4x_5'x_6'x_8x_9'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3x_4'x_6x_7'x_8x_9'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3x_4x_5'x_6x_7'x_8x_9'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19}
\end{array}\right| \qquad K_1'K_2'K_3'K_4'K_5'K_6'K_7'K_8 = 0 \qquad (4.28)
$$

მიღებული კონუნქციების (4.28) ჩასმით ორთოგონალური დიზუნქციებით გამოსახულ სისტემის ქმედითუნარიანობის ფუნქციაში (4.27) საბოლოოდ მივიღებთ ორთოგონალურ დიზუნქციურ ნორმალურ ფორმას:

$$
y(x_1,\ldots,x_{21}) = x_1 \left| \begin{array}{l}
x_2x_3x_4x_5 \\
x_2x_3x_4'x_6x_7 \\
x_2x_3x_4x_5'x_6x_7 \\
x_2x_3x_4'x_6'x_8x_9 \\
x_2x_3x_4x_5'x_6'x_8x_9 \\
x_2x_3x_4'x_6x_7'x_8x_9 \\
x_2x_3x_4x_5'x_6x_7'x_8x_9 \\
x_2'x_{12}x_{13}x_{14}x_{15} \\
x_2x_3'x_{12}x_{13}x_{14}x_{15} \\
x_2x_3x_4'x_6'x_8'x_{12}x_{13}x_{14}x_{15} \\
x_2x_3x_4x_5'x_6'x_8'x_{12}x_{13}x_{14}x_{15} \\
x_2x_3x_4'x_6x_7'x_8'x_{12}x_{13}x_{14}x_{15} \\
x_2x_3x_4x_5'x_6x_7'x_8'x_{12}x_{13}x_{14}x_{15} \\
x_2x_3x_4'x_6'x_8x_9'x_{12}x_{13}x_{14}x_{15} \\
x_2x_3x_4x_5'x_6'x_8x_9'x_{12}x_{13}x_{14}x_{15} \\
x_2x_3x_4'x_6x_7'x_8x_9'x_{12}x_{13}x_{14}x_{15} \\
x_2x_3x_4x_5'x_6x_7'x_8x_9'x_{12}x_{13}x_{14}x_{15} \\
x_2'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3x_4'x_6'x_8'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3x_4x_5'x_6'x_8'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3x_4'x_6x_7'x_8'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3x_4x_5'x_6x_7'x_8'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3x_4'x_6'x_8x_9'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3x_4x_5'x_6'x_8x_9'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3x_4'x_6x_7'x_8x_9'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2x_3x_4x_5'x_6x_7'x_8x_9'x_{12}x_{13}x_{14}'x_{18}x_{19} \\
x_2'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3x_4'x_6'x_8'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3x_4x_5'x_6'x_8'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3x_4'x_6x_7'x_8'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3x_4x_5'x_6x_7'x_8'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3x_4'x_6'x_8x_9'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3x_4x_5'x_6'x_8x_9'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3x_4'x_6x_7'x_8x_9'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19} \\
x_2x_3x_4x_5'x_6x_7'x_8x_9'x_{12}x_{13}x_{14}x_{15}'x_{18}x_{19}
\end{array} \right| \qquad (4.29)
$$

როგორც ჩანს (4.29) გამოსახულების ყველა წევრი წვილ-წყვილად ორთოგონალურია. ორთოგონალური დიზუნქციის ნორმალური ფორმით წარმოდგენილი ნებისმიერი ლოგიკურ-ალბათური ფუნქციის ჭეშმარიტობის ალბათობა ტოლია ამ ფუნქციის ყველა ორთოგონალური წევრის ჭეშმარიტობის ალბათობების ჯამის.

(4.29) ფორმულის თანახმად, განხილული სისტემისათვის შეგვიძლია გამოვთვალოთ უმტყუნო მუშაობის ალბათობა.

$$
\begin{aligned}
P\{y=1\} = {} & R_1(R_2R_3R_4R_5 + R_2R_3Q_4R_6R_7 + R_2R_3R_4Q_5R_6R_7 + R_2R_3Q_4Q_6R_8R_9 + \\
& + R_2R_3R_4Q_5Q_6R_8R_9 + R_2R_3Q_4R_6Q_7R_8R_9 + R_2R_3R_4Q_5R_6Q_7R_8R_9 + Q_2R_{12}R_{13}R_{14}R_{15} + \\
& + R_2Q_3R_{12}R_{13}R_{14}R_{15} + R_2R_3Q_4Q_6Q_8R_{12}R_{13}R_{14}R_{15} + R_2R_3R_4Q_5Q_6Q_8R_{12}R_{13}R_{14}R_{15} + \\
& + R_2R_3Q_4R_6Q_7Q_8R_{12}R_{13}R_{14}R_{15} + R_2R_3R_4Q_5R_6Q_7Q_8R_{12}R_{13}R_{14}R_{15} + \\
& + R_2R_3Q_4Q_6R_8Q_9R_{12}R_{13}R_{14}R_{15} + R_2R_3R_4Q_5Q_6R_8Q_9R_{12}R_{13}R_{14}R_{15} + \\
& + R_2R_3Q_4R_6Q_7R_8Q_9R_{12}R_{13}R_{14}R_{15} + R_2R_3R_4Q_5R_6Q_7R_8Q_9R_{12}R_{13}R_{14}R_{15} + \\
& + Q_2R_{12}R_{13}Q_{14}R_{18}R_{19} + R_2Q_3R_{12}R_{13}Q_{14}R_{18}R_{19} + R_2R_3Q_4Q_6Q_8R_{12}R_{13}Q_{14}R_{18}R_{19} + \\
& + R_2R_3R_4Q_5Q_6Q_8R_{12}R_{13}Q_{14}R_{18}R_{19} + R_2R_3Q_4R_6Q_7Q_8R_{12}R_{13}Q_{14}R_{18}R_{19} + \\
& + R_2R_3R_4Q_5R_6Q_7Q_8R_{12}R_{13}Q_{14}R_{18}R_{19} + R_2R_3Q_4Q_6R_8Q_9R_{12}R_{13}Q_{14}R_{18}R_{19} + \\
& + R_2R_3R_4Q_5Q_6R_8Q_9R_{12}R_{13}Q_{14}R_{18}R_{19} + R_2R_3Q_4R_6Q_7R_8Q_9R_{12}R_{13}Q_{14}R_{18}R_{19} + \\
& + R_2R_3R_4Q_5R_6Q_7R_8Q_9R_{12}R_{13}Q_{14}R_{18}R_{19} + Q_2R_{12}R_{13}R_{14}Q_{15}R_{18}R_{19} + \\
& + R_2Q_3R_{12}R_{13}R_{14}Q_{15}R_{18}R_{19} + R_2R_3Q_4Q_6Q_8R_{12}R_{13}R_{14}Q_{15}R_{18}R_{19} + \\
& + R_2R_3R_4Q_5Q_6Q_8R_{12}R_{13}R_{14}Q_{15}R_{18}R_{19} + R_2R_3Q_4R_6Q_7Q_8R_{12}R_{13}R_{14}Q_{15}R_{18}R_{19} + \\
& + R_2R_3R_4Q_5R_6Q_7Q_8R_{12}R_{13}R_{14}Q_{15}R_{18}R_{19} + R_2R_3Q_4Q_6R_8Q_9R_{12}R_{13}R_{14}Q_{15}R_{18}R_{19} + \\
& + R_2R_3R_4Q_5Q_6R_8Q_9R_{12}R_{13}R_{14}Q_{15}R_{18}R_{19} + R_2R_3Q_4R_6Q_7R_8Q_9R_{12}R_{13}R_{14}Q_{15}R_{18}R_{19} + \\
& + R_2R_3R_4Q_5R_6Q_7R_8Q_9R_{12}R_{13}R_{14}Q_{15}R_{18}R_{19})
\end{aligned}
$$

ყველა ელემენტის ერთნაირი საიმედოობის შემთხვევაში სისტემის უმტყუნო მუშაობის ალბათობა (4.29) ტოლი იქნება

$$
\begin{aligned}
P\{y=1\} = R_c = {} & R(R^4 + R^4Q + R^5Q + R^4Q^2 + R^5Q^2 + R^5Q^2 + R^6Q^2 + R^4Q + \\
& + R^5Q + R^6Q^3 + R^7Q^3 + R^7Q^3 + R^8Q^3 + R^7Q^3 + R^8Q^3 + R^8Q^3 + R^9Q^3 + \\
& + R^4Q^2 + R^5Q^2 + R^6Q^4 + R^7Q^4 + R^7Q^4 + R^8Q^4 + R^7Q^4 + R^8Q^4 + R^9Q^4 + \\
& + R^5Q^2 + R^6Q^2 + R^7Q^4 + R^8Q^4 + R^8Q^4 + R^9Q^4 + R^9Q^4 + R^8Q^4 + R^9Q^4 + R^{10}Q^4) =
\end{aligned}
$$

$$= R(R^4 + 2R^4Q + 2R^5Q + 2R^4Q^2 + 4R^5Q^2 + 2R^6Q^2 + R^6Q^3 + 3R^7Q^3 + 3R^8Q^3 +$$
$$+ R^9Q^3 + R^6Q^4 + 4R^7Q^4 + 6R^8Q^4 + 4R^9Q^4 + R^{10}Q^4) = R(R^4 + 2R^4(1-R) +$$
$$+ 2R^5(1-R) + 2R^4(1-R)^2 + 4R^5(1-R)^2 + 2R^6(1-R)^2 + R^6(1-R)^3 +$$
$$+ 3R^7(1-R)^3 + 3R^8(1-R)^3 + R^9(1-R)^3 + R^6(1-R)^4 + 4R^7(1-R)^4 +$$
$$+ 6R^8(1-R)^4 + 4R^9(1-R)^4 + R^{10}(1-R)^4)$$

$$P\{y=1\} = R_c = 5R^4 - 4R^6 - 5R^8 + 9R^{10} - 5R^{12} + R^{14}$$

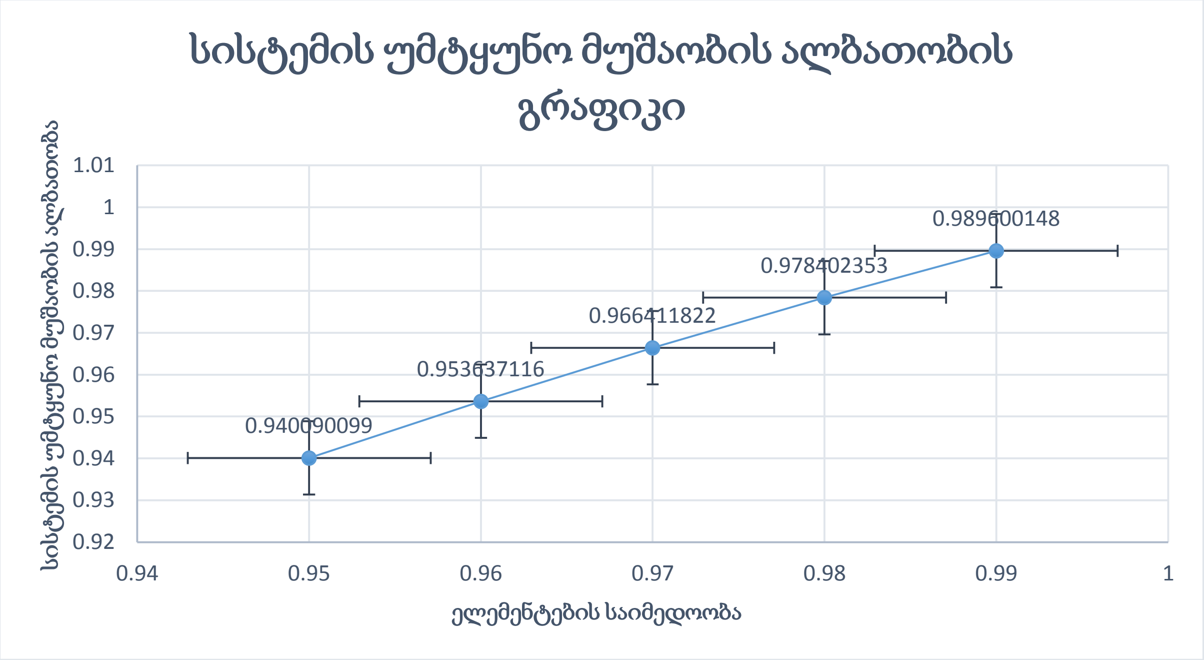
ცხრილ N 5. სისტემის უმტყუნო მუშაობის ალბათობები მნიშვნელობები, სისტემაში შემავალი ელემენტების საიმედოობის სხვადასხვა მნიშვნელობებისათვის.

	$R = 0.95$	$R = 0.96$	$R = 0.97$	$R = 0.98$	$R = 0.99$
$P\{Y = 1\}$	0.940090098752321	0.9536371160895537	0.966411822204292	0.9784023534554174	0.9896001484287327

წყარო: (კვლევითი მეთოდები)

ცხრილ N 5-ის საფუძველზე აგებულია სისტემის უმტყუნო მუშაობის ალბათობის გრაფიკი, რომელიც მოცემულია გრაფიკი N 1.

გრაფიკი N 1 - სისტემის უმტყუნო მუშაობის ალბათობის გრაფიკი



წყარო: (კვლევის მასალები)

როგორც გრაფიკიდან ჩანს, სისტემის ქმედითუნარიანობის მაჩვენებელი დამოკიდებულია როგორც ცალკეული ელემენტების უმტყუნო მუშაობის ალბათობაზე, ასევე სისტემის სტრუქტურაზე: რაც უფრო მაღალია ელემენტების უმტყუნო მუშაობის ალბათობები, მით უფრო მაღალია სისტემის ქმედითუნარიანობა.

ორთოგონალიზაციის ალგორითმი საკმაოდ შრომატევადია ხელით გამოთვლების დროს, მაგრამ თუ კომპიუტერის საშუალებით ვაწარმოებთ უმტყუნო მუშაობის ალბათობის რაოდენობრივ შეფასებას, მაშინ ორთოგონალიზაციის მეთოდი მნიშვნელოვნად მარტივი და ეფექტურია დიდი განზომილების ამოცანების ამოხსნისათვის.

განხილული მეთოდი საშუალებას გვაძლევს მოვახდინოთ სხვადასხვა სტრუქტურების შეფასება და ანალიზი, რაც საბოლოოდ სისტემის გაუმჯობესების საშუალებას იძლევა.

4.5. ცალკეული ელემენტის როლი სისტემის საიმედოობაში

როგორც სადისერტაციო ნაშრომის წინა თავში ავღნიშნეთ კომპიუტერული ქსელის დაპროექტებისას ერთერთ მთავარ ამოცანას წარმოადგენს სისტემის რაციონალური სტრუქტურის შექმნა. ამისთვის აუცილებელია ისეთი მათემატიკური აპარატის დამუშავება, რომელიც სისტემის საიმედოობაში მისი შემადგენელი ელემენტების როლის რაოდენობრივი შეფასების საშუალებას მოგვცემს, რის საფუძველზეც შესაძლებელი იქნება განისაზღვროს სისტემის "სუსტი ადგილები (კვანძები)", შეირჩეს ელემენტებს შორის ფუნქციების ოპტიმალური განაწილება და სისტემების უმტყუნოებაზე რაციონალური ზემოქმედება.

ლოგიკურ-ალბათური მეთოდის გამოყენება, სისტემის უმტყუნო მუშაობის ალბათობის შეფასებისა და ანალიზის გარდა, საშუალებას იძლევა განისაზღვროს კონკრეტული ელემენტის როლი სისტემის ქმედითუნარიანობის უზრუნველყოფაში.

ამ მიზნით განვიხილავთ სისტემის შემადგენელი ელემენტების სხვადასხვა მაჩვენებლების, კერძოდ "წონის", "მნიშვნელობისა" და "წვლილის" შეფასების ამოცანას ლოგიკურ-ალბათური მეთოდის გამოყენებით, რომლებიც სასარგებლოა სისტემის ანალიზისათვის როგორც პროექტირებისას, ასევე მისი ექსპლუატაციის დროს სისტემის დათვალიერების, დიაგნოსტირებისა და შეკეთების თანმიმდევრობის თვალსაზრისით.

სისტემის ცალკეული ელემნტების როლის განსაზღვრა ასევე მნიშვნელოვანია ისეთი ამოცანების გადასაჭრელად, როგორიცაა გაუმართაობის მოძებნა, ოპტიმალური რეზერვირების სტრატეგიის შემუშავება, საიმედოობის მაჩვენებლებისადმი წაყენებული მოთხოვნის განაწილება და ა.შ.

ისეთი ამოცანის გადასაწყვეტად, როგორიცაა მთლიანი სისტემისადმი წაყენებული საიმედოობის მიღწევად ცალკეული ელმენტების საიმედოობის უზრუნველყოფა, აუცილებელია იმის ცოდნა თუ რომელი ელემენტი უფრო მნიშვნელოვანია და რომელი ნაკლებმნიშვნელოვანი. ე.ი. უნდა ვიცოდეთ ამა თუ იმ ელემენტის მწყობრიდან გამოსვლა რა სახის ზემოქმედებას იწვევს სისტემის საერთო საიმედოობაზე.

4.5.1. სისტემაში ელემენტის ”წონის” რაოდენობრივი შეფასება

სისტემის შემადგენელი ელემენტების საიმედოობის რიცხობრივი მახასიათებლების შესახებ ინფორმაციის არცოდნის შემთხვევაში, მთლიანი სისტემის საიმედოობაზე ცალკეული ელემენტთა ზემოქმედების განსაზღვრა შესაძლებელია ისეთი მცნებით როგორიცაა ელემენტის "წონა". ნაშრომში მოყვანილია სისტემაში ელემენტის "წონის" აღწერის ორი მეთოდი ფორმალურ-ტოპოლოგიური და ფუნქციონალურ-ტოპოლოგიური, რომლებიც მუშაობენ ელემენტების მტყუნების ალბათობის საწყისი მნიშვნელობების არარსებობის პირობებში. პირველი მეთოდი (ფორმალურ-ტოპოლოგიური) ეფუძვნება სისტემის გრაფის გამოკვლევას. უშუალო გზების მატრიცა წარმოადგენს ელემენტის სტრუქტურული მნიშვნელობის – მისი რანგის (R_i) სიდიდის განსაზღვრის საფძველს. სტრუქტურის მიხედვით რანგის მნიშვნელობა ასახავს მოცემული ელემენტის სხვა ელემენტებთან კავშირების რიცხვს.

$$R_i = \frac{A_i + A_i^2}{\sum_{i=1}^{n}\left(A_i + A_i^2\right)} \qquad (4.30)$$

სადაც A_i, A_i^2 - უშუალო გზების მატრიცის იმ სტრიქონების ჯამია, რომლებიც მთავარ დიაგონალზე შეიცავენ ერთიანს.

მეორე მეთოდი (ფუნქციონალურ-ტოპოლოგიური) დაფუძნებულია სისტემის სტრუქტურული სქემის ანალიზისა და მისი მუშაობის პრინციპების აღწერაზე. სისტემის ფუნქციონირების შესწავლის საფუძველზე შესაძლებელია გრაფის აგება, რომელიც აღწერს სისტემაში ელემენტების დომინირების შესაძლებლობებს და დომინირების მატრიცას. ელემენტის ფუნქციონალურ-სტრუქტურული რანგის მნიშვნელობა შესაძლებელია გამოვთვალოთ (4.30) გამოსახულების ანალოგიური ფორმულით:

$$R_i = \frac{B_i + B_i^2}{\sum_{i=1}^{n}\left(B_i + B_i^2\right)} \qquad (4.31)$$

სადაც B_i, B_i^2 - მატრიცის მომიჯნავე სტრიქონების ჯამი. მიუხედავად ამისა ამ მეთოდებს გააჩნიათ ნაკლოვანებები. მაგალითად, უცნობია როგორ შევაფასოთ ელემენტები, რომელსაც გააჩნია ნულოვანი რანგი (პირველი მეთოდის შემთხვევაში), არ არსებობს ერთიანი აზრი ელემენტების ურთიერთმოქმედების ხარისხზე (მეორე მეთოდში).

განსაზღვრება 4.1. n ელემენტისაგან შემდგარი სისტემის ”წონა” არის სისტემის ყველა შესაძლო 2^n მდგომარეობებიდან ქმედითუანრიანი მდგომარეობების წილი.

განსაზღვრება 4.2. სისტემის ”წონა” x_i არგუმენტის მიხედვით არის იმ მდგომარეობათა რაოდენობა, რომლებზეც ლოგიკურ-ალგებრული ფუნქცია $\Delta_{x_i} y(x_1,..x_n)$ ღებულობს ერთის ტოლ მნიშვნელობას.

$$G\{\Delta_{x_i} y(x_1,..x_n)\} \qquad (4.32)$$

x_i ელემენტის ”წონა” (4.32) ახასიათებს ამ ელემენტის როლს სისტემის სტრუქტურულ საიმედოობაში. სხვადასხვა რაოდენობის ელემენტებისაგან შემდგარი სისტემებისათვის ერთიდაიგივე ”წონა” ობიექტურად არ ახასიათებს (აფასებს) მოცემულ ელემენტს. ამიტომ მიზანშეწონილია ელემენტის ”წონა” განისაზღვროს 0-დან 1-მდე დიაპაზონით.

n ელემენტისაგან შემდგარ სისტემაში x_i ელემენტის ”წონა” არის x_i არგუმენტის მიხედვით სისტემის ”წონის” ფარდობა სისტემის შესაძლო ყველა მდგომარეობათა რაოდენეობასთან.

$$g_{x_i} = \frac{G\{\Delta_{x_i} y(x_1,..x_n)\}}{2^n} \qquad (4.33)$$

თუ სისტემის ქმედითუნარიანობის ფუნქცია წარმოდგენილია ორთოგონალური დიზიუნქციის ნორმალური ფორმით, მაშინ ლოგიკური ფუნქციის "წონის" განსაზღვრა შეიძლება შემდეგი სახით:

$$g_{y(x_1,\dots,x_n)} = \frac{\sum_{f=1}^{k} 2^{n-r_f}}{2^n} = \sum_{f=1}^{k} 2^{-r_f} \qquad (4.34)$$

სადაც k - ლოგიკურ ფუნქციაში ორთოგონალური კონიუნქციების რიცხვია; n - ფუნქციის არგუმენტების რიცხვი; r_f ელემენტარული ორთოგონალური კონიუნქციის რანგი.

მონოტონური ლოგიკურ-ალგებრული ფუნქციისათვის

$$\Delta_{x_i} y(x_1,..x_n) = y_1^{(i)}(x_1,\dots,x_n) y_0'^{(i)}(x_1,\dots,x_n) \qquad (4.35)$$

$$\Delta_{x_i} y(x_1,..x_n) = y_1^{(i)}(x_{n-1}) \setminus y_0^{(i)}(x_{n-1}) \qquad (4.36)$$

სადაც ფუნქციების $y_1^{(i)}$ და $y_{01}^{(i)}$ ორთოგონალური დიზიუნქციის ნორმალურ ფორმაში ჩაწერით მივიღებთ

$$G\{\Delta_{x_i} y(X_n)\} = G\{y_1^{(i)}(x_{n-1})\} - G\{y_0^{(i)}(x_{n-1})\} =$$
$$= \sum_{f=1}^{k} 2^{n-(r_f-1)} - \sum_{j=1}^{l} 2^{n-(r_j-1)} \qquad (4.37)$$

სადაც k , r_f - იმ ორთოგონალური კონიუნქციების რიცხვი და რანგია, რომლებიც შეიცავენ x_i არგუმენტს; l , r_j ,- იმ ორთოგონალური კონიუნქციების რიცხვი და რანგია, რომლებიც შეიცავენ x_i არგუმენტის უარყოფას ე.ი. x_i' არგუმენტს.

თუ (4.37) გამოსახულებას გავყოფთ 2^n -ზე, მივიღებთ სისტემაში x_i ელემენტის ”წონი” გამოსათვლელ ფორმულას.

$$g_{x_i} = \sum_{f=1}^{k} 2^{-(r_f-1)} - \sum_{j=1}^{l} 2^{-(r_j-1)} \qquad (4.38)$$

წყარო: (ცირამუა, 2014).

ელემენტის "წონა" ახასიათებს სისტემის ისეთი კრიტიკული მდგომარეობების ფარდობით რაოდენობას, რომელშიც ერთი სტრუქტურული ელემენტის მტყუნებას მივყავართ სისტემის მტყუნებამდე (და პირიქით, მტყუნებული ელმენტის აღდგენას მივყავართ სისტემის აღდგენამდე) სისტემის ყველა შესაძლო მდგომარეობებს შორის როცა $x_i = 1$.

სისტემაში ელემენტის "წონის" შეფასების ალგორითმი(4.38) გამოსახულებაზე დაყრდნობით შეიძლება ჩამოვაყალიბოთ შემდეგნაირად:

1. სისტემის ქმედითუანრიანობის ფუნქციის ორთოგონალური დიზიუნქციის ნორმალური ფორმის განსაზღვრა;

2. იმ კონიუნქციების ამოღება, რომლებიც არ შეიცავენ x_i ან x_i' არგუმენტს;

3. x_i ელემენტების შემცველი კონიუნქციების ჯამური წონის გამოთვლა;

4. x_i' ელემენტების შემცველი კონიუნქციების ჯამური წონის გამოთვლა;

5. სისტემის ”წონის” განსაზღვრა x_i არგუმენტის მიხედვით, რომელიც მე-3 და მე-4 ეტაპებზე მიღებული შედეგების სხვაობის ტოლია;

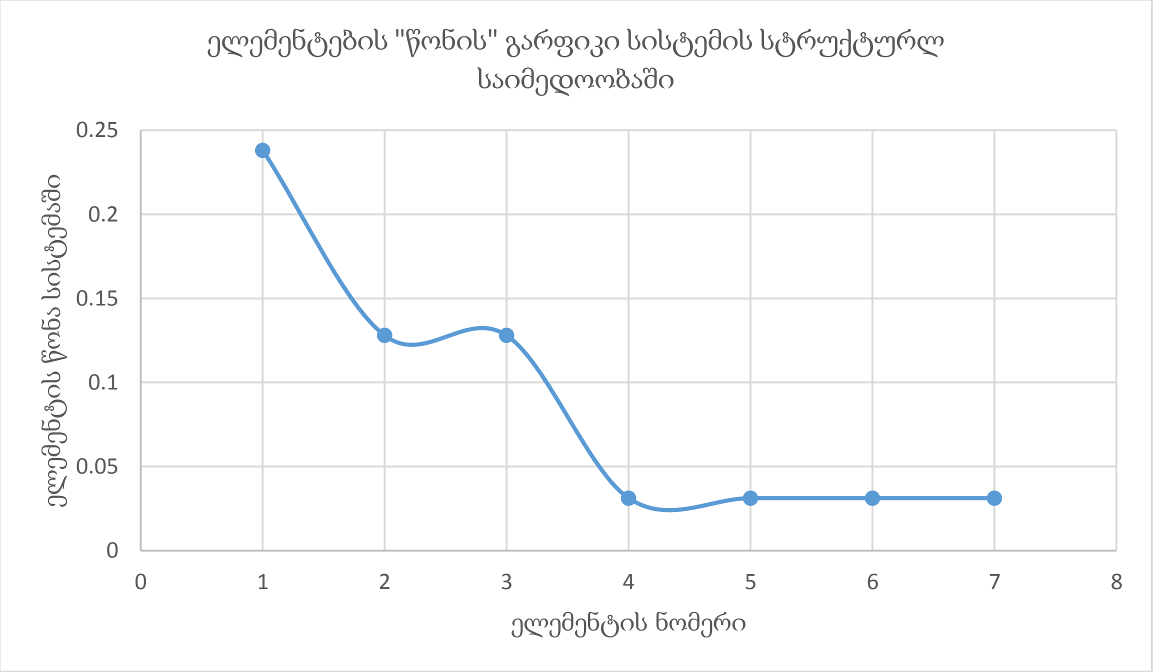
6. x_i ელემენტის ”წონის” განსაზღვრა ხდება მე-5 ეტაპზე სისტემის `წონის˜ რაოდენობრივი მაჩვენებლის გაყოფით 2^n -ზე, სადაც n - სისტემაში შემავალი ელემენტების რიცხვია. აღნიშნული ალგორითმის საშუა ლებით გამოვთვალოთ ცალ-კეული ელემენტის "წონა" ილუსტრაცია N 29-ზე მოცემული სისტემის სტრუქტურაში. აღნიშნული სისტემის ქმედითუნარიანობის პირობა ორთოგონალური დიზიუნქციის ფორმით მოცემულია (4.4) ფორმულით, რომლის თანახმად შეგვიძლია გამოვთვალოთ სისტემის თითეული ელემენტის „წონა“ სტურქტურულ საიმედოობაში

ცხრილი N 6. სისტემის ცალკეული ელემენტების „წონა“

ელემენტები	x_1	x_2	x_3	x_4	x_5	x_6	x_7
წონა	0.238	0.128	0.128	0.0313	0.0313	0.0313	0.0313

წყარო: (კვლევის მასალები)

გრაფიკი. N 2. ელემენტების "წონის" გარფიკი სისტემის სტრუქტურლ საიმედოობაში



წყარო: (კვლევის მასალები)

4.5.2. სისტემაში ელემენტის "მნიშვნელობის" შეფასება

ელემენტის "წონა" ახასიათებს სისტემაში მის სტრუქტურულ მდებარეობას (მდგომარეობას) და არ არის დამოკიდებული საიმედოობის მაჩვენებლებზე. ამის საპირისპიროდ ელემენტის "მნიშვნელობა" განისაზღვრება არა მარტო სისტემაში მისი სტრუქტურული მდებარეობით, არამედ ყველა დანარჩენი ელემენტების საიმედოობით, გარდა თავისივე (x_i ელემენტის) საიმედოობისა. სისტემის ელემენტების სიმედოობის შესახებ ინფორმაციის არსებობის შემთხვევაში შესაძლებელია მათი მთლიანი სისტემის საიმედოობაზე ზემოქმედების შეფასება ელემენტის "მნიშვნელობის" მიხედვით.

$y(x_1,...,x_n)$ სისტემაში x_i ელემენტის "მნიშვნელობა" არის სისტემის უმტყუნო მუშაობის ალბათობის R_c კერძო წარმოებული ელემენტის უმტყუნო მუშაობის ალბათობით R_i

$$\xi_{x_i} = \frac{\partial P\{y(x_1,...,x_n)=1\}}{\partial P\{x_i=1\}} = \frac{\partial R_c}{\partial R_i} \tag{4.39}$$

x_i ელემენტის "მნიშვნელობა" რიცხობრივად ტოლია

$$\xi_{x_i} = P\{\Delta_{x_i} y(x_1,...,x_n)=1\} \tag{4.40}$$

მონოტონური სტრუქტურის მქონე სისტემებისათვის

$$\xi_{x_i} = \frac{\partial R_c}{\partial R_i} = R_{c_1}^{(i)} - R_{c_0}^{(i)} \tag{4.41}$$

$$R_{c_1}^{(i)} = P\{y_1^{(i)}(x_1,...,x_n)=1\} \tag{4.42}$$

$$R_{c_0}^{(i)} = P\{y_0^{(i)}(x_1,...,x_n)=1\} \tag{4.43}$$

სადაც $R_{c_1}^{(i)}$ და $R_{c_0}^{(i)}$ სისტემის უმტყუნო მუშაობის ალბათობაა x_i ელემნტის აბსოლუტური საიმედოობისა და მტყუნების დროს.

(4.40) - (4.43) გამოსხულებების საფუძველზე შეიძლება ჩამოვაყალიბოთ შემდეგის ხუთი შედეგი:

$$\frac{\partial R_c}{\partial Q_i} = R_{c_0}^{(i)} - R_{c_1}^{(i)} \tag{4.44}$$

$$\frac{\partial R_c}{\partial R_i} = -\frac{\partial R_c}{\partial Q_i} \tag{4.45}$$

$$R_c = R_{c_0}^{(i)} + R_i \frac{\partial R_c}{\partial R_i} = R_{c_1}^{(i)} + Q_i \frac{\partial R_c}{\partial Q_i} \qquad (4.46)$$

$$\frac{\partial R_c}{\partial R_i} = \frac{R_c - R_{c_0}^{(i)}}{R_i} = \frac{R_{c_1}^{(i)} - R_c}{Q_i} \qquad (4.47)$$

$$\frac{\partial R_c}{\partial Q_i} = \frac{R_c - R_{c_1}^{(i)}}{Q_i} = \frac{R_{c_0}^{(i)} - R_c}{R_i} \qquad (4.48)$$

ელემენტის ”წონა” არის მისი ”მნიშვნელობის” კერძო შემთხვევა ყველა ელემენტის ერთნაირი საიმედოობისა და 0,5-ის ტოლი უმტყუნო მუშაობის ალბათობის დროს. (4.39) გამოსახულების სახით მოცემული x_i ელემენტის "მნიშვნელობა", რომელიც ახასიათებს სისტემის საიმედოობის ცვლილების სისწრაფეს, შესაძლებელია ინტერპრეტირებული იქნას, როგორც სისტემის უმტყუნო მუშაობის ალბათობის პირობა x_i ელემნტის ქმედითუნარიანობის დროს. გამოვთვალოთ ცალკეული ელემენტის "მნიშვნელობა" ილუსტრაცია N 29-ზე მოცემული სისტემის სტრუქტურაში. (დავუშვათ, ყველა ელემენტი ერთნაირი საიმედოობისაა). მაშინ

$$\frac{\partial R_c}{\partial R_2} = 3R^4\text{-}3R^6\text{-}5R^8\text{+}9R^{10}\text{-}5R^{12}\text{+}R^{14} \qquad (4.49)$$

$$\frac{\partial R_c}{\partial R_4} = R^4 - 2R^6 - R^8 - R^9 + 9R^{10} - 4R^{11} - 6R^{12} + 5R^{13} - R^{14}$$

$$\frac{\partial R_c}{\partial R_8} = R^4\text{-}2R^6\text{-}R^8\text{+}5R^{10}\text{-}4R^{12}\text{+}R^{14}$$

$$\frac{\partial R_c}{\partial R_{12}} = 2R^4\text{-}R^6 - R^7\text{-}2R^8 - 6R^9\text{+}13R^{10}\text{-}R^{11}\text{-}5R^{12}\text{+}R^{14}$$

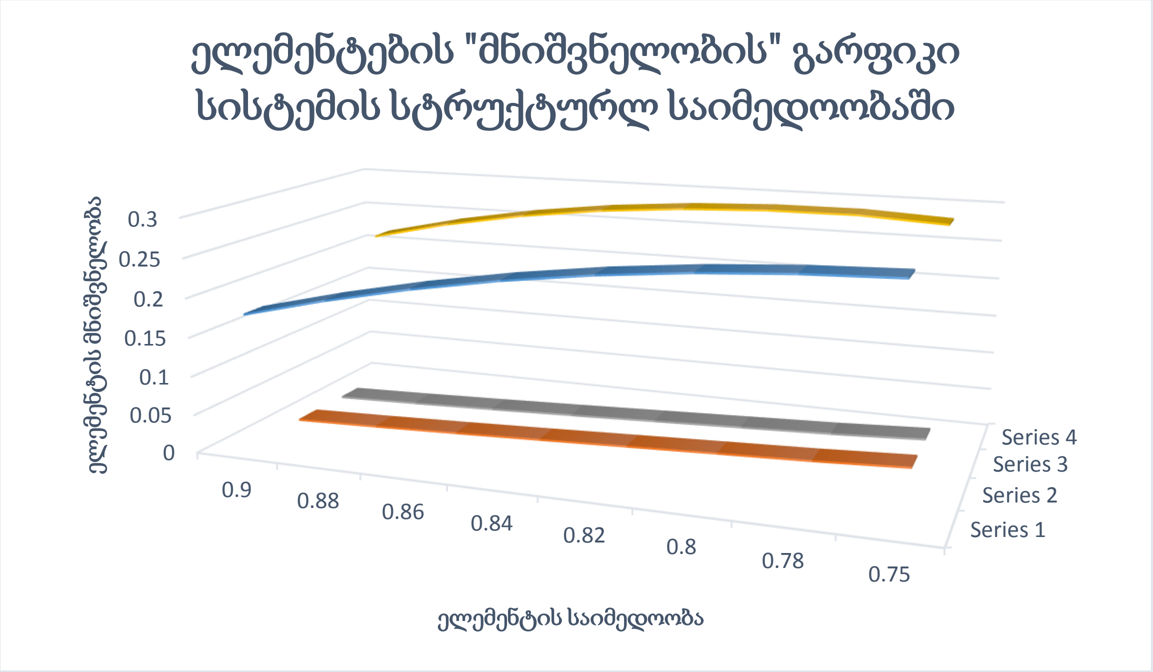
გამომთველის შედეგები მოცემულია ცხრილი N 6-ში რის საფუძველზეც აგებულია გრაფიკი N 3, სადაც თვალნათლივ ჩანს ელემენტის როლი სისტემის საიმედოობაში.

ცხრილი N 6. სისტემის ცალკეული ელემენტების „მნიშვნელობა“.

R_i	ξ_{x_2}	ξ_{x_4}	ξ_{x_8}	$\xi_{x_{12}}$
0.9	0.176367	0.005782	0.005193	0.215291
0.88	0.20287	0.00887	0.008089	0.239666
0.86	0.225628	0.012464	0.011519	0.258525
0.84	0.244478	0.01641	0.015343	0.27222
0.82	0.25935	0.020536	0.019398	0.281122
0.8	0.270258	0.024673	0.023513	0.285625
0.78	0.277296	0.028658	0.027524	0.286142
0.75	0.280976	0.034043	0.033016	0.280378

წყარო: (კვლევის მასალები)

გრაფიკი N 3 ელემენტების "მნიშვნელობის" გარფიკი სისტემის სტრუქტურლ საიმედოობაში



წყარო: (კვლევის მასალები)

4.6.3 სისტემაში ელემენტის "წვლილის" შეფასება

$y(x_1,...,x_n)$ სისტემის საიმედოობაში x_i ელემენტის "წვლილი" არის R_i ელემენტის უმტყუნო მუშაობის ალბათობის ნამრავლი მის "მნიშვნელობაზე"

$$B_{x_i} = R_i \frac{\partial R_c}{\partial R_i} \tag{4.50}$$

(4.49) გამოსახულების (4.50) ფორმულაში ჩასმით მივიღებთ

$$B_{x_i} = R_i \frac{R_c - R_{c_0}^{(i)}}{R_i} = R_c - R_{c_0}^{(i)} \tag{4.51}$$

ელემენტის "წვლილი" B_{x_i} ახასიათებს x_i ელემენტის მტყუნებული მდგომარეობიდან ქმედითუნარიან მდგომარეობაში (იგივე უმტყუნო მუშაობის ალბათობით R_i) აღდგენის შემდეგ სისტემის საიმედოობის ზრდას. როგორც (4.50) გამოსახულებიდან ჩანს ელემენტის "წვლილი" ნაკლებია ან ტოლოა მის "მნიშვნელობაზე" $B_{x_i} \le \xi_{x_i}$

სისტემის საიმედოობაში ელემენტების როლის შეფასების აღწერილ სამივე კრიტერიუმს "წონას", "მნიშვნელობასა" და "წვლილს" სახვადასხვა ინფორმაციული მნიშვნელობა გააჩნია. ”წონა” - g_{x_i} ახასიათებს, სისტემაში x_i ელემენტის მხოლოდ სტრუქტურულ მდებარეობას.

"მნიშვნელობა" - ξ_{x_i} განისაზღვრება არა მხოლოდ სისტემაში x_i ელემენტის სტრუქტურული მდებარეობით, არამედ დამოკიდებულია ყველა დანარჩენი ელემენტების უმტყუნო მუშაობის ალბათობაზე (გარდა თავისივე x_i ელემენტის საიმედოობისა).

"წვლილი" - B_{x_i} ახასიათებს, როგორც x_i ელემენტის მდებარეობას სისტემის სტრუქტურაში, ასევე მისი ფუნქციონირების პირობას და სისტემის შემადგენელი ყველა m ელემანტის უმტყუნო მუშაობია ალბათობასთან კავშირს.

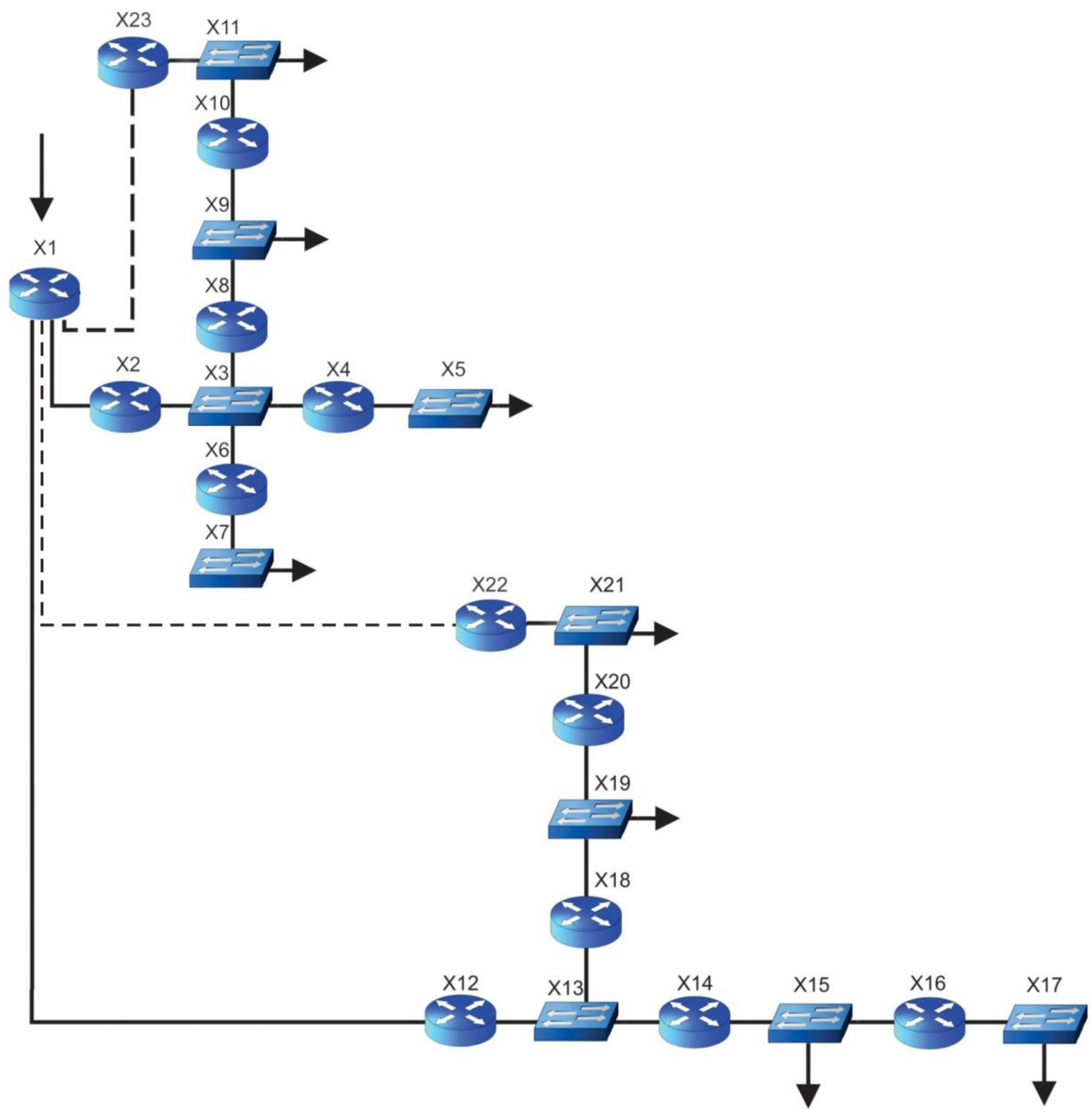
4.6. საბაზო სტრუქტურისა და რეზერვირებული სისტემის უმტყუნო მუშაობის ალბათობის შედარებითი ანალიზი

უნდა აღინიშნოს, რომ საინფორმაციო-საკომუნიკაციო სისტემის პროექტირებისას ვერ ხერხდება სისტემის სტრუქტურის სრულყოფილების თვალსაზრისით ერთმნიშვნელოვანი პასუხის მოძებნა, ყოველთვის არსებობს სისტემის აგების რამოდენიმე ალტერნატიული ვარიანტი, ამასთანავე ყოველი მათგანი უშვებს მოდიფიცირების შესაძლებლობას. ამრიგად, ყველაზე მეტად რაციონალური სტრუქტურის ამორჩევა ფაქტიურად წარმოებს შერჩევისა და სხვადასხვა ვარიანტების მიმდევრობითი შედარების გზით. სისტემის სტრუქტურის სრულყოფილების თვალსაზრისით, უნდა შეფასდეს სისტემის სტრუქტურის შესაძლო გადაწყობათა ვარიანტები და მათ შორის შეირჩეს ყველაზე რაციონალური სტრუქტურის მქონე სისტემა. როგორც ილუსტრაცია N 29-დან ჩანს მე-13, მე-19 და 21-ე კვანძების ფუნქციონირება მთლიანად დამოკიდებულია მე-12 კავშირის არხის მდგომარეობაზე და მისი დაზიანება იწვევს კავშირის გაწყვეტას აღნიშნულ კვანძებთან. ამ პრობლემის გადაჭრა შესაძლებელია სარეზერვო არხის (x_{22}) დამატებით ცენტრალურ (პირველ) და 21-ე კავძს შორის, მაშინ მე-12 კავშირის არხის დაზიანება არ გამოიწვევს დანარჩენი (მე-13, მე-19 და 21-ე) კვანძების ფუნქციონირების დარღვევას, ვინაიდან ინფორმაციის გადაცემა შესაძლებელი გახდება სარეზერვო არხის საშუალებით, რომლის ჩართვაც საჭიროების შემთხვევაში მოხდება ავტომატურად.

ილუსტრაცია N 30-ზე მოცემულია სისტემას სტრუქტურული სქემა, სადაც გამოყენებულია ორი სარეზერვო არხი x_{22} და x_{23}

ლოგიკურ-ალბათური მეთოდის გამოყენებით მოვახდინოთ სისტემის საიმედოობის შედარებითი ანალიზი და შეფასება რეზერვირების არხების გამოყენების შემთხვევაში.

ილუსტრაცია N 30. სტრუქტურული სქემა სარეზერვო x_{22} და x_{23} არხებით.



წყარო: (კვლევის მასალები)

როგორც წინა პარაგრაფებში იყო აღწერილი, სისტემის უმტყუნო მუშაობის ალბათობის რაოდენობრივი შეფასებისათვის, თავდაპირველად ლოგიკურ-ალბათული მოდელის გამოყენებით უნდა აღიწეროს სისტემის ქმედითუნარიანობის პირობა, შემდეგ განვსაზღვროთ მიღებული ფუნქციის ორთოგონალური დიზიუნქციის ნორმალური

ფორმა, რის შემდეგაც შეგვიძლია სისტემის უმტყუნო მუშაობის ალბათობის გამოთვლა, რომელიც ამ ფუნქციის ყველა ორთოგონალური წევრების ჭეშმარიტობის ალბათობების ჯამის ტოლია.

ილუსტრაცია N 30-ზე წარმოდგემნილი სისტემების ქმედითუნარიანობის პირობას, ფუნქციონირების უმოკლესი გზების საშუალებით, მატრიცული ფორმით აქვს შემდეგი სახე:

$$y = (x_1, \dots, x_{21}) = x_1 \left| \begin{array}{c} x_2x_3x_4x_5 \\ x_2x_3x_4x_7 \\ x_2x_3x_8x_9 \\ x_2x_3x_8x_9x_{10}x_{11} \\ x_{11}x_{23} \\ x_9x_{10}x_{11}x_{23} \\ x_3x_4x_5x_8x_9x_{10}x_{11}x_{23} \\ x_3x_6x_7x_8x_9x_{10}x_{11}x_{23} \\ x_{12}x_{13}x_{14}x_{15} \\ x_{12}x_{13}x_{14}x_{15}x_{16}x_{17} \\ x_{12}x_{13}x_{18}x_{19} \\ x_{12}x_{13}x_{18}x_{19}x_{20}x_{21} \\ x_{21}x_{22} \\ x_{19}x_{20}x_{21}x_{22} \\ x_{13}x_{14}x_{15}x_{18}x_{19}x_{20}x_{21}x_{22} \\ x_{13}x_{14}x_{15}x_{16}x_{17}x_{18}x_{19}x_{20}x_{21}x_{22} \end{array} \right| \qquad (4.52)$$

გარდავქმნათ სისტემების ქმედითუნარიანობის პირობის აღმწერი (4.52) გამოსახულება ორთოგონალურ დიზუნქციურ ნორმალურ ფორმად, ნაშრომში აღწერილი მეთოდის გამოყენებით.

ორთოგონალური დიზუნქციის ნორმალური ფორმით წარმოდგენილი ნებისმიერი ლოგიკურ-ალბათური ფუნქციის ჭეშმარიტობის ალბათობა ტოლია ამ ფუნქციის ყველა ორთოგონალური წევრის ჭეშმარიტობის ალბათობების ჯამის.

საბოლოოდ ყველა ელემენტის ერთნაირი საიმედოობის შემთხვევაში განხილული სისტემების უმტყუნო მუშაობის ალბათობა ტოლი იქნება:

$$P\{y=1\}=R_c=5R^5-6R^7+2R^8+4R^9-5R^{10}-5R^{11}+4R^{12}+4R^{13}-R^{14}-R^{15}$$

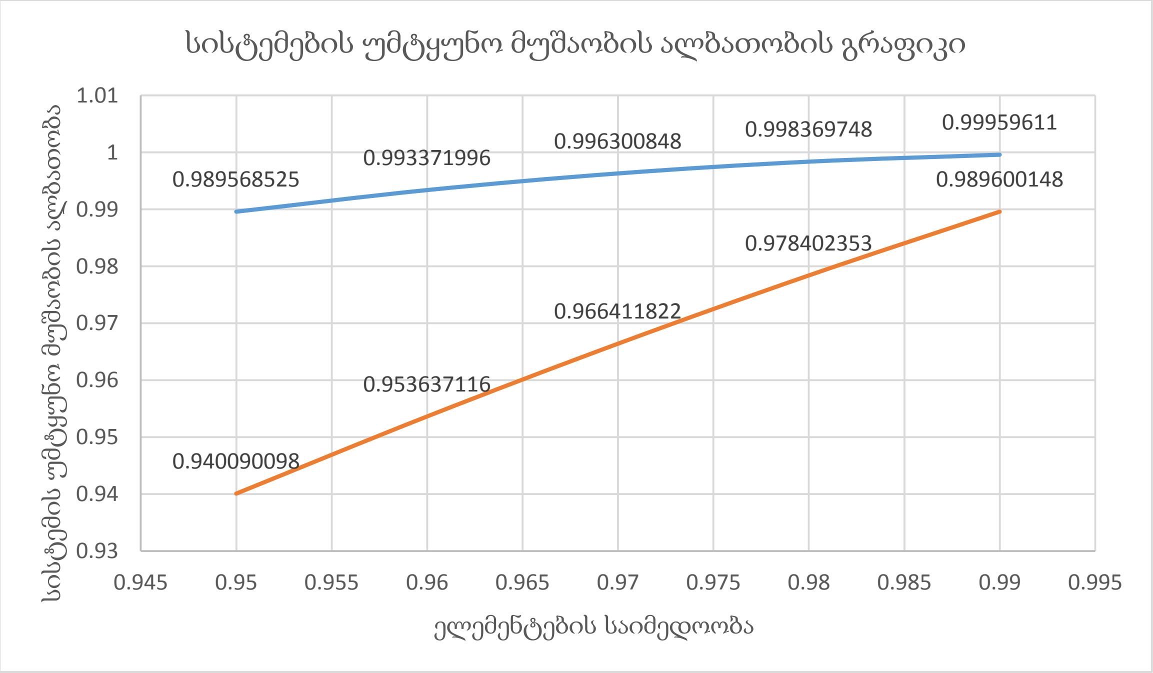
ცხრილ N 7-ში მოცემულია სისტემის უმტყუნო მუშაობის (საიმედოობის) ალბათობის რიცხვითი მნიშვნელობები, სისტემაში შემავალი ელემენტების საიმედოობის სხვადასხვა მნიშვნელობებისათვის.

ცხრილი N 7-სისტემის უმტყუნო მუშაობის (საიმედოობის) ალბათობის რიცხვითი მნიშვნელობებიის საფუძველზე აგებულია უმტყუნო მუშაობის ალბათობის გრაფიკი სისტემის ორივე ვარიანტისათვის.

სისტემები	$P=0,95$	$P=0,96$	$P=0,97$	$P=0,98$	$P=0,99$
რეზერვირებით	0.989568525	0.993371996	0.996300848	0.998369748	0.99959611
საბაზო	0.940090098	0.953637116	0.966411822	0.978402353	0.989600148

წყარო: (კვლევის მასალები)

გრაფიკი N 2. სისტემების უმტყუნო მუშაობის ალბათობის გრაფიკი 1-საბაზო ვარიანტი, 2 - რეზერვირებით.



წყარო: (კვლევის მასალები)

როგორც გრაფიკიდან ჩანს და როგორც მოსალოდნელი იყო, სისტემაში სარეზერვო კავშირის არხების დამატებამ უზრუნველყო სისტემის საიმედოობის მნიშვნელოვანი ამაღლება.

მიღებული შედეგები

სადისერტაციო ნაშრომში ჩატარებული კვლევის შედეგად მიღებულია შემდეგი თეორიული და პრაქტიკული შედეგები:

1. დასაბუთებულია, რომ კომპიუტერული ქსელი მიეკუთვნება რთული სტრუქტურის მქონე სისტემებს, რომელთა დაპროექტების დროს აუცილებელია განხორციელდეს სისტემის მათემატიკური, კერძოდ კი ლოგიკურ-ალბათური მოდელირება
2. განხილულია საქართველოს უნივერსიტეტის კომპიუტერული ქსელის დროითი და სივრცული განვითარების დინამიკა
3. დამუშავებულია კომპიუტერული ქსელის სტრუქტურული საიმედოობის ლოგიკურ-ალბათური მოდელები
4. შესრულებულია საქართველოს უნივერსიტეტის კომპიუტერული ქსელის ფრაგმენტის საიმედოობის რაოდენობრივი შეფასება ლოგიკურ-ალბათური მოდელების გამოყენებით და ჩატარებულია მიღებული შედეგების ანალიზი
5. დამუშავებულია კომპიუტერული ქსელის თითოეული კვანძის „წონის“, „მნიშვნელობისა“ და „წვლილის“ რაოდენობრივი შეფასების მეთოდოლოგია ლოგიკურ-ალბათური მეთოდების ბაზაზე
6. ჩატარებულია საქართველოს უნივერსიტეტის ქსელის ფრაგმენტის მაგალითზე თითოეული კვანძის „წონის“, „მნიშვნელობისა“ და „წვლილის“ რაოდენობრივი შეფასება და შესაბამისი ანალიზი.
7. დასაბუთებულია, რომ მხოლოდ მათემატიკური მოდელირების (ლოგიკურ-ალბათური მოდელირების) საფუძველზეა შესაძლებელი მაღალი ეფექტიანობის (სტრუქტურული საიმედოობის) და ოპტიმალური ქსელის დაპროექტება.

გამოყენებული ლიტერატურა

ცირამუა, ს. (2014). *რთული სტრუქტურის სისტემების ლოგიკურ-ალბათური მოდელირება.* თბილისი: ელექტრონული სალექციო კურსი

Закер, К. (1997). Компьютерные сети

Олифер, В.Г., & Олифер, В.Г. (1999). Компьютерные сети

Питер, М. Таненбаум Э. (2003). *Компьютерные сети.*

Рабинин,А. (2000). *Надежность и Безопасность Структурно-сложных систем. Политехника, Санкт-Петербург .*

Рабинин, А. (1993). *Концепция логико-вероятностной теории безопасности Приборы и системы управления.*

Рабинин, А. & Черкесов, Н. (1981). *Логико-вероятностные методы исследования надежности структурно-сложных систем.*

Соловьев, Е. & Каминский, Ю. (1999). *Надежность и безопасность АС.*

Таненбауь, Э. (2001). Современные операционные системы

Introduction to Protocols. (უთარიღო). მოძიებულია 2015 წლის 2 თებერვალი, http://pluto.ksi.edu/~cyh/cis370/ebook/ch06b.htm

About Cisco IOS Software Documentation. (უთარიღო). მოძებნილია 2014 წლის 20 ნოემბერი, http://www.cisco.com/c/en/us/td/docs/ios/12_2/ip/configuration/guide/fipr_c/1cfabout.html

Anderson, J. (2008). ecurity Engineering: A Guide to Building Dependable Distributed Systems Hardcover.

B. & Foss, A, & Rios, R. (2014). IP Routing on Cisco IOS, IOS XE, and IOS XR: An Essential Guide to Understanding and Implementing IP Routing Protocols. Paperback Edgeworth.

Bonaventure, O. (2014). Computer Networking : Principles, Protocols and Practice

Boudec, J & Thiran, P. (2011). Network Calculus: A Theory of Deterministic Queuing Systems for the Internet

Cisco Systems (2010). Internetworking Technology Handbook. Cisco Press, 4 edition.

Cole, E. (2009). Network Security Bible Paperback.

Configuration Professional: Site-to-Site IPsec VPN Between Two IOS Routers Configuration Example (ნოემბერი 30, 2011). მოძებნილია 2013 წლის 10 ოქტომბერს.

http://www.cisco.com/c/en/us/support/docs/cloud-systems-management/configuration-professional/113337-ccp-vpn-routerA-routerB-config-00.html

Configuring a Router IPsec Tunnel Private-to-Private Network with NAT and a Static (ივნისი 5, 2006). მოძებნილია 2012 წლის 10 იანვარს.

http://www.cisco.com/c/en/us/support/docs/security-vpn/ipsec-negotiation-ike-protocols/14144-static.html

Coulte, C. (1997). Understanding Networking Technologies. Atrium Technical Inc

Das, N. (2012). Optical Communication.

Davidoff , S. & Ham, J. (2012). Network Forensics: Tracking Hackers through Cyberspace Hardcover.

Dean, T. (2012) . Network+ Guide to Networks.

DHCP with the WLC (სექტემბერი 10, 2009 წელი). მოძებნილია 2014 წლის 10 ივნისი,

Dutton, H & Lenhard, P. (1995). High-Speed Networking Technology: An Introductory Survey . Prentice Hall; 3rd edition

Fordham, S. (2015). VPNs and NAT for Cisco Networks: A CCIE v5 guide to Tunnels, DMVPN, VPNs and NAT. Cisco CCIE Routing and Switching v5.0

Gallager, R. (2008). Principles of Digital Communication. Cambridge University Press

Gibson, D. (2104). CompTIA Security+.

Greer, K. (2010). Thinking Networks - the Large and Small of it: Autonomic and Reasoning Processes for Information Networks.

Grigorik, I. (2013). High Performance Browser Networking: What Every Web Developer Should Know about Networking and Web Performance

Halabi, S. (2000). Internet Routing Architectures

Hamilton Ortiz, J. (2012). Telecommunications Networks

Hester, M. (2013) . Microsoft Windows Server 2012 Administration Instant Reference.

http://www.cisco.com/c/en/us/support/docs/wireless/4400-series-wireless-lan-controllers/110865-dhcp-wlc.html

Hundley, K. 14,(2009). Alcatel-Lucent Scalable IP Networks

Hunt, C. (2002). TCP/IP Network Administration.

Kaufman, C. & Perlman, R. & Speciner, M. (2002). Network Security: Private Communication in a Public World.

Kaufmann, M. (2007).Network Routing: Algorithms, Protocols, and Architectures

Kenneth, C., Mansfield, Jr, & Antonakos, J. L. (2010). *Computer Networking for LANS to WANS: Hardware, Software and Security.* United States of America

Kim, P. (2014). The Hacker Playbook: Practical Guide To Penetration Testing

Knapp, E. & Langill, J. (2014). Industrial Network Security, Second Edition: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems.

Lapidoth, A. (2009). A Foundation in Digital Communication. Cambridge University Press.

Macfarlane, J. (2006) Network Routing Basics: Understanding IP Routing in Cisco Systems Hardcover

Mackin, J. & Northrup, T. (2011). Self-Paced Training Kit (Exam 70-642) Configuring Windows Server 2008 Network Infrastructure (MCTS) (2nd Edition). Microsoft Press.

Marsic, I. (2013). Computer Networks: Performance and Quality of Service. Rutgers University

McNab, C. (2007). Network Security Assessment: Know Your Network.

Narottam Das, N. (2014). Advances in Optical Communication.

Odom, W. (2013). CCNA Routing and Switching 200-120 Network Simulator.

Peig, O. (2010). Open Networks. Free Technology Academy.

Sanders, C. (2011). Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems Paperback.

Sanders, C. (2013). Applied Network Security Monitoring: Collection, Detection, and Analysis.

Stallings, W. (2013). Cryptography and Network Security: Principles and Practice.

Stallings, W. (2013). Network Security Essentials Applications and Standards.

Stanek, W. (2013). Windows Server 2012 Inside Out.

Stewart, M. (2010). Network Security, Firewalls, And Vpns.

Traynor, P. & McDaniel, P. & La Porta, P. (2008). Security for Telecommunications Networks.

Tse, D & Viswanath, P. (2005). Fundamentals of Wireless Communication. Cambridge University Press

White, R. & Tantsura, J. (2015). Navigating Network Complexity: Next-generation routing with SDN, service virtualization, and service chaining Paperback