



БЪЛГАРСКАТА АКАДЕМИЯ НА НАУКИТЕ

ИНСТИТУТ ПО ИНФОРМАЦИОННИ И
КОМУНИКАЦИОННИ ТЕХНОЛОГИИ

Красимир Димитров Тричков

**МОДЕЛИРАНЕ И УПРАВЛЕНИЕ НА ИНФОРМАЦИОННИ
ПОТОЦИ**

ДИСЕРТАЦИЯ

за придобиване на образователната и научна степен «доктор»

по специалност 02.21.10. „Приложение на принципите и методите на
кибернетиката в различни области на науката”

в професионално направление 5.3. „Компютърна и комуникационна
техника”

Научен ръководител:
/проф. дтн. Тодор Стоилов/

София, 2014 г.

Съдържание

Въведение.....	4
Глава 1. Анализ на проблема за намиране на най-кратък път в мрежа	7
1.1 Увод.....	7
1.2 Преглед на проблема за намиране на най-кратък път	7
1.3 Основни алгоритми.....	10
1.3.1 Топологично сортиране	11
1.3.2 Намиране на най-къс път.....	12
1.3.3 Алгоритъм на Дийкстра.....	12
1.3.4 Алгоритъм на Дийкстра при ациклични графи.....	15
1.3.5 Минимално обхващащо дърво.....	18
1.3.6 Алгоритъм на Прим	18
1.3.7 Алгоритъм на Крускал.....	19
1.4 Методи за определяне на всички пътища между зададени два възела от граф	21
1.4.1 Рекурсивен метод чрез изчисляване на поддетерминантата	21
1.4.2 Метод чрез обхождане в ширина/дълбочина	21
1.5 Определяне на сечения посредством метод с булево инвертиране и оптимизиране на резултата	22
1.6 Задачи за вземане на решение.....	22
1.7 Линейно програмиране.....	24
1.7.1 Общ вид на математически модел на задача за линейно програмиране [82].....	24
1.7.2 Форми на задачата за линейно програмиране	25
1.8 Изводи	26
1.9 Цел и задачи.....	26
Глава 2. Определяне на пропускателните способности на канали чрез времеви измервания	28
2.1 Увод	28
2.2 Инструменти за изчисляване на пропускателна способност на база времезакъснението	28
2.3 Изчисляване на пропускателна способност на отделните възли на база времезакъснението	32
2.3.1 Изчисляване на пропускателна способност на възел ИО-Варна.....	37
2.3.2 Изчисляване на пропускателна способност на възел ИО-Перник	40
2.3.3 Изчисляване на пропускателна способност на възел ИО-Кюстендил.....	41
2.3.4 Изчисляване на пропускателна способност на възел ИО-Благоевград	42
2.3.5 Изчисляване на пропускателна способност на възел ИО-Видин	43
2.3.6 Изчисляване на пропускателна способност на възел ИО-Монтана.....	44
2.3.7 Изчисляване на пропускателна способност на възел ИО-Враца.....	45
2.3.8 Изчисляване на пропускателна способност на възел ИО-Плевен.....	46
2.3.9 Изчисляване на пропускателна способност на възел ИО-Ловеч.....	47
2.3.10 Изчисляване на пропускателна способност на възел ИО-Пазарджик	48
2.3.11 Изчисляване на пропускателна способност на възел ИО-Смолян	49
2.3.12 Изчисляване на пропускателна способност на възел ИО-Пловдив	50
2.3.13 Изчисляване на пропускателна способност на възел ИО-Кърджали.....	51
2.3.14 Изчисляване на пропускателна способност на възел ИО-Хасково.....	52
2.3.15 Изчисляване на пропускателна способност на възел ИО-Стара Загора.....	53
2.3.16 Изчисляване на пропускателна способност на възел ИО-Габрово	54
2.3.17 Изчисляване на пропускателна способност на възел ИО-Велико Търново	55
2.3.18 Изчисляване на пропускателна способност на възел ИО-Русе	56

2.3.19 Изчисляване на пропускателна способност на възел ИО-Разград	57
2.3.20 Изчисляване на пропускателна способност на възел ИО-Търговище	58
2.3.21 Изчисляване на пропускателна способност на възел ИО-Сливен.....	59
2.3.22 Изчисляване на пропускателна способност на възел ИО-Ямбол	60
2.3.23 Изчисляване на пропускателна способност на възел ИО-Бургас.....	61
2.3.24 Изчисляване на пропускателна способност на възел ИО-Шумен.....	62
2.3.25 Изчисляване на пропускателна способност на възел ИО-Силистра	63
2.3.26 Изчисляване на пропускателна способност на възел ИО-Добрич	64
2.4 Изводи	66
Глава 3. Проектиране на двусвързана и трисвързана мрежа при отчитане на съществуваща топология тип звезда	67
3.1 Увод	67
3.2 Определяне модела на комуникационна мрежа за получаване на трисвързаност между отделните точки.....	67
3.3 Определяне на времезакъснението между възлите при добавени виртуални канали	71
3.4 Изчисляване на времезакъсненията между възлите на определените виртуални пътища	72
3.5 Дефиниране на задачата за оптимизация за определяне свързаността на мрежата ..	77
3.6 Изводи	85
Глава 4. Анализ на резултатите от проектирането на комуникационната мрежа	86
4.1 Увод	86
4.2 Първи модел.....	86
4.3 Втори модел	88
4.4 Трети модел	90
4.5 Анализ на получените резултати	92
4.6 Изводи	101
Заклучение	102
Приноси на дисертационния труд	103
Бъдещи насоки за работа	104
Публикации, свързани с дисертацията.....	105
Научноизследователски проекти	106
Декларация за оригиналност на резултатите.....	107
Библиография	108
Използвани съкращения	115

Въведение

Цел на дисертационния труд е посредством моделиране и управление на информационни потоци да се направи топологичен синтез и предложи оптимално решение за преминаване на съществуваща комуникационна мрежа от свързаност тип „звезда“ към двусвързана и трисвързана топология. Да се разработи модел за количествена оценка на качеството, непрекъсваемостта и отказоустойчивостта на обслужването с оглед технологично и структурно подобряване на комуникационната мрежа.

Актуалност на проблема

Компютърните комуникационни мрежи функционират в нестационарни условия на възникване на заявки за обслужване: предаване на данни, файлов обмен, дистанционно ползване на изчислителни ресурси, мултимедийни приложения. Нестационарността на натоварването е основна трудност за точността на анализа, моделирането, оптимизацията на компютърно-комуникационните мрежи [1, 2].

Топологичният синтез на избор на комуникационни канали в компютърна мрежа оказва влияние на множество задачи, свързани с експлоатацията и управлението на комуникационната мрежа. Топологичният синтез влияе на задачи като: задача за предаване на съобщение от източник до потребител [3, 4, 5, 6] при зададена топология и пропускателни способности на каналите, и обем трафик между всяка двойка източник - получател [7, 8] е необходимо да се определят информационните потоци в мрежата; определяне на маршрутът на съобщение, за всяка двойка източник - получател, при което се удовлетворяват изисквания за минимален път, времезакъснения, цена и др. Тези оптимизационни задачи имат голяма размерност и голям брой ограничения [9, 10, 11] и техните решения се влияят от топологията на комуникационната мрежа.

В ежедневната си дейност се стремим към определени цели, за постигането на които е необходимо да решаваме взаимно свързани проблеми [12, 13]. Всеки от тези проблеми обикновено има множество решения. Естествено е предпочитанието да изберем от многото възможни решения най-доброто, т.е. оптималното решение [14].

Информационните и комуникационни системи интензивно се използват в производствената и стопанска практика [15, 16, 17]. Изграждането на техническа хетерогенна структура изисква да се определи удачна топология на мрежата. Това ще позволи удачно и оптимално да се изпълняват информационните задачи на потребителите на комуникационната мрежа. Така оптимизирането на структурата на мрежата ще позволи да се оптимизират и информационните потоци в компютърни комуникационни мрежи за да се изпълняват потребителски приложения, които изискват гарантирано качество на обслужване [18, 19, 20]. Това определи обхвата на дисертационния труд, за намиране на удачна топология на мрежа, което ще влияе на управлението на информационни потоци. В дисертационната работа се извършва топологичен синтез и се определя оптимално решение за преминаване на съществуваща

комуникационна мрежа от свързаност тип „звезда“ към двусвързана и трисвързана топология. Разработва се модел за количествена оценка на качеството и отказоустойчивостта на обслужването с оглед технологично и структурно подобряване на комуникационната мрежа [21].

Основната работна задача, която си поставя дисертационния труд е да дефинира и реши оптимизационна задача за определяне на свързаността на мрежа като се използват резултати от измервания на времезакъсненията и пропаданията в комуникационната мрежа и се определи решение за преминаване на съществуваща комуникационна мрежа от свързаност тип „звезда“ към двусвързана и трисвързана топология. С цел подобряване качеството, непрекъсваемостта и отказоустойчивостта на обслужването и с оглед технологично, и структурно подобряване на комуникационната мрежа да се направи количествена оценка на моделите със съответните параметрични състояния.

Целите и задачите на дисертацията са изложени в края на глава първа.

Структура на дисертацията

Дисертационният труд е разделен на увод, четири глави и заключение. Дисертацията съдържа 115 страници, 92 фигури и 18 таблици, 121 цитирани литературни източника и списък на използваните съкращения. По дисертационния труд са публикувани 8 публикации, като 4 от тях са доклади от международни конференции. Изследванията са част от получените резултати на 6 научноизследователски проекта.

В първа глава са анализирани моделите за формализиране на информационни и комуникационни потоци в мрежи чрез графи. Оценени са техните особености, предимства и недостатъци. Разгледани са проблемите за намиране на най-кратък път от един възел към всички останали възли в мрежи, за които цената на дъгата може да варира. Тези задачи имат влияние за определяне на оптимална топология на комуникационна мрежа. Направени са основни изводи на базата на проучените литературни източници и са формулирани целите и задачите на дисертацията.

Във втора глава е определена и анализирана пропускателната способност и времезакъсненията в мрежата при предаване на информационни потоци. Оценявано е пропадането на връзка между двойка възли като характеристика на отказоустойчивостта на съществуваща комуникационна мрежа със свързаност тип „звезда“. Дефиниран е алгоритъм за определяне пропускателните способности на база времезакъснението на достъпа до съответните възли. Определени са средните времеви закъснения между отделните възли при съществуващата топология. Изчислен е капацитетът на информационната мрежа.

В трета глава е синтезирана архитектура на съществуваща мрежа от свързаност тип „звезда“ с определени допълнителни алтернативни комуникационни канали за преход към двусвързана и трисвързана архитектура. Синтезиран е алгоритъм за определяне на теоретичните времезакъснения между възлите на добавените алтернативни

комуникационни канали. Дефинирана е оптимизационна задача за определяне на свързаността на мрежата като са използвани резултатите от измерванията на времезакъсненията и пропаданията в комуникационната мрежа. Оптимизационната задача е решавана за определяне на структура на комуникационната мрежа като двусвързана и трисвързана. Анализирани са получаваните решения.

В четвърта глава са разгледани и анализирани три модела за определяне на оптималния вариант за подобряване на съществуваща комуникационна мрежа. Направена е количествена оценка на получените резултати. На база топологичен синтез е избран най-добрият модел със съответно параметрично състояние за преминаване от съществуваща топология „звезда“ към разпределена трисвързана мрежа между отделните възлови точки с цел постигане на непрекъсваемост на информационните потоци.

В заключението са описани различните дейности, извършени при изпълнението на отделните етапи. Включени са приносите на дисертационния труд, бъдещите насоки на работа, научните публикации по дисертацията, както и научно-изследователските проекти, по които е работено във връзка с изследванията по дисертационния труд.

Глава 1. Анализ на проблема за намиране на най-кратък път в мрежа

1.1 Увод

Към настоящия момент в науката съществува голямо множество от методи и алгоритми за представяне на комуникационни мрежи и системи посредством графи. В тази глава се анализират моделите за формализиране на информационни и комуникационни потоци в мрежи чрез графи, които имат отношение към темата на дисертацията. Оценяват се техните особености, предимства и недостатъци. Разглеждат се проблемите за намиране на най-кратък път от един възел към всички останали възли в мрежи, за които цената на дъгата може да варира. Също така се извършва теоретично изследване и анализиране на аспектите, свързани с моделирането и управлението на информационни потоци в мрежи.

1.2 Преглед на проблема за намиране на най-кратък път

В [22] се засяга и разглежда проблемът за намирането на най-кратък път от един възел към всички останали възли в мрежи, за които цената на дъгата може да варира с времето - всяка дъга има време за преминаване и паркиране със съответна различна стойност на времето, позволена за възлите. Това показва, че този проблем е еквивалентен на класическия проблем за намиране на най-кратък път във времеразширена мрежа.

Намирането на най-кратък път има много полезни приложения. По тази причина, това е било обект на задълбочени изследвания. Дийкстра [23] е представил добре известния алгоритъм, който решава проблема за $O(n^2)$ време. Флойд [24] допуска ограничението, че всички теглови коефициенти трябва да бъдат не отрицателни. Флойд е представил алгоритъм, който решава този по-общ проблем за $O(n^3)$ време. Когато тегловите коефициенти се променят като функция на времето $w_{ij}(t)$, такива мрежи са наречени динамични мрежи. В динамична мрежа, тегловите коефициенти показват динамичен модел.

Въпреки това, тегловите коефициенти не са детерминирани по всяко време или през интервал от време. Тези мрежи представят какво се вижда в реалния свят. Фу [25] и Драйфус [26] показват, че предходни алгоритми, като на Дийкстра [23] могат да бъдат разширени за да предоставят полиномни алгоритми за време за този по-общ тип мрежа. Въпреки това, те предполагат, че функциите на тегловите коефициенти са известни предварително, монотонни са и не се променят. Тези предположения не са подходящи за много реални житейски ситуации. Фу показва, че намирането на най-кратък път в тези мрежи е трудно решим проблем. Този проблем е резултат от нарушаването на принципа на Белман. Принципа на Белман за оптималност посочва, че „всеки под път на най-краткия път трябва да бъде най-краткия път“ [27]. Той е показал, че когато времената за пътуване не променят монотонно принципа на Белман, могат да бъдат нарушени. Фу е разработил няколко вида евристични алгоритми за търсене. Основните свойства и алгоритми са обсъдени и в двете дискретни и непрекъснати условия на времето от Ахуя и други [28], Кай и други [29], Кабини [30] Орда и Ром [31, 32, 33], Палотино и Скутела [34], и Филпот и Меес [27, 35].

Време­зависимите мрежи се използват за моделиране на ситуации в които цената на преминаването към дадена точка (времето за пътуване или закъснение) се променят с времето [36]. Докато общата рамка има много приложения, всекидневните проблеми, свързани с планирането на маршрутите на комуникационните мрежи са най-непреодолими. В зависимост от променящите се натоварвания и задръствания по пътя, едновременно времето за пътуване от възела източник s към крайната точка d и оптималният път между тях могат да се променят с течение на времето. В действителност, условията по пътя често създават "стеснени места", които водят до малко или много едно и също време за достигане до крайната точка, независимо от по-късното тръгване от началната точка и значително редуциране на времето за пътуване. Чрез представяне на времето за пристигане като функция от времето на тръгване, могат да се планират оптималните времена за пътуване, вместо оптимална дължина на пътя, каквато е практиката в действителност. Освен техните естествени приложения в управлението и планирането на комплексни системи, каквито са комуникационните мрежи, време­зависимите най-кратки пътища представляват фундаментален проблем и с последици от параметрите на модела [37].

Проблема се изследва от 1969 г., когато Драйфус [38] наблюдава, че алгоритъма на Дийкстра може да се използва за намиране на време­зависимия най-кратък път, при зададено началното време от възела източник. Драйфус възприема политика за неограничено изчакване в междинните възли, за да осигури, че ако изчакването в даден възел е в резултат на по-ранно време на пристигане (поради намалена претовареност), тогава оптималното време за изчакване се оползотворява във възлите. През 80-те и 90-те години, Орда и Ром [31] подробно изследват политиките за изчакване и типа на функциите на разходите във възлите, които имат нетривиални последици върху зависимите от времето най-кратки пътища. Измежду техните изводи можем да изтъкнем тези, които разглеждат, че ако изчакване не се допуска никъде в мрежата и мрежата не е от вида FIFO (първи влязъл, първи излязъл), то тогава най-краткият път може да не е толкова просто да бъде открит и да наруши оптималните характеристики на подпътищата, за разлика от случая FIFO.

Също така би могло да има пътища, които минават през безкраен брой възли, но въпреки това да имат ограничено закъснение. Така, по този начин, нито алгоритъмът на Дийкстра, нито алгоритъмът на Белман - Форд може да намери най-кратките пътища, зависими от времето и вариантите на тези проблеми са известни като задачи с NP трудност [39]. Във време­зависимите мрежи една точка, която не е от вида FIFO е еквивалентна на FIFO точка, ако изчакването е позволено: може винаги да се изчаква в края на точката, за да се оптимизира времето за пристигане на заглавната част в точката. Проблемът за намирането на най-кратките пътища продължава да е изненадващо комплексен, дори когато точките са FIFO и техните ценови функции са линейни. Проблема, който се решава в дисертационния труд засяга намирането на най-късия път в разпределна мрежа, като за намирането му се използва време­закъснението между отделните точки.

Орда и Ром [31], както и много други са установили проблема, поради значимостта му за планирането. Вместо да се търси най-добрия път при започване в определено време, което лесно се изчислява с алгоритъма на Дийкстра, целта в случая е да се построи

карта за планиране на пътищата която дава, за всяко произволно време на пристигане, оптималното време за тръгване да минимизира времето за пътуване.

Би трябвало да се отбележи, че в тази задача времето е продължителна променлива. Задачата ще има значително по-различен характер в случая, когато става дума за дискретно време и ограничен брой времена за тръгване, които се допускат [40, 41, 42]. В последния случай (с дискретно време) комплексността на функцията на времето за тръгване е ограничена от дискретното време на тръгване, но функцията на времето на пристигане може да бъде нарушена от неестествени прекъсвания. Орда и Ром [31] разширяват парадигмата за намиране на най-краткия път на Белман и Форд [41, 43], за да работят с линейни функции като цени в точките, вместо със скаларни разходи, в случая с променлива, която не е времето. Авторите оставят неразрешена сложността на изчисленията на задачата. По-конкретно, те представят времето на пътуване в техния алгоритъм като функционална комплексност, която „скрива“ актуалните разходи на съставяне и минимизиране на функциите на времето за пристигане. Ако функциите за цените в точките са линейни, тогава съставянето и минимизирането им в междинните възли е отчасти, но броят на прекъсванията в тези функции може да расте. Тази комплексност не се разглежда в анализа на Орда и Ром [31]. Възможно е авторите да предполагат, че комплексността е полиномно ограничена, но няма доказателства за това. В действителност, в няколко последващи статии (теоретични и приложни разработки), авторите представят техните алгоритми, дали размерът на функцията на времето за пристигане не е по-лоша от линейната [44, 45].

Най-систематичното изследване на комплексността на функцията на времето за пристигане е представено от Дийн [46, 47, 48, 49]. След няколко първоначални погрешни твърдения [46, 48, 50], че комплексността е линейна по отношение на броя на прекъсванията на функцията в точките, достига до извода, че комплексността може да бъде полиномна [47, 51, 52]. В системата FIFO с линейни функции на пристигане, функцията за пристигане в крайния възел може да има полиомен брой отделни най-кратки пътища.

Изследването за най-кратките пътища е изключително важно за развитието на комбинаторните алгоритми. Класическите алгоритми, като тези на Дийкстра и Белман - Форд [53, 42], предлагат най-кратки пътища в статична система с фиксирани цени на точката. Но всъщност много приложения използват динамичната система, в която разходите на точката се променят. Промените в топологията на системите, като включването или изключването на отделна точка, може също да бъде моделирано чрез цената.

Времебезависимите мрежи са един модел за моделиране на подобни динамики, но разбира се, че има също така и други възможности за това. Задачата за динамичните най-кратки пътища се занимава с включването и изтриването на отделни точки от графиката, като се вземат предвид да се поддържа структура на данните за ефективно актуализиране на дървото на най-краткия път от единичен източник или поддържащи запитвания за най-краткия път между два възела в отговор на такава промяна [54, 55, 56, 57, 58, 59]. Задачата с вероятностните най-кратки пътища се опитва да реши проблема с несигурността в крайните цени чрез моделирането им като случайни променливи и изчисляване на пътища с минимални очаквани цени [60, 61, 62]. Задачата

за претеглен най-кратък път за региона [63, 64] моделира динамиката на средата в безкрайността: разделена на региони, всеки с различна скорост на пътуване, въпреки че скоростта не е функция на времето.

Задачата с параметрични най-кратки пътища има най-много сходства с време-зависимите пътища [65]. В този случай, разходът във всяка точка e в графиката се променя като линейна функция на параметъра u , напр. $c(e) = a u + b$. Най-краткият път от възела източник s до крайния възел d , зависи от стойността на параметъра. Целта в задачата е да се изчислят най-кратките пътища за всички стойности на параметъра u . Резултатите от Карстенсен [66] показват, че в най-лошия случай най-краткият път от s до d може да се променя $n^{\Omega(\log n)}$, при промяна на параметъра u . По-просто и директно доказателство на тази долна граница е представено от Мълмлей и Шах [67]. Горната граница $n^{O(\log n)}$ е дадена от Гасфийлд [68].

Една по-рестриктивна форма на параметричната функция се използва от Карп и Орлин [69], както и от Янг, Таржан и Орлин [70]. В техния модел крайните разходи се дефинират като $c(e) = c_e - u$.

Тази проста формула, която може да се проследи, води само до полиномен брой от отделни най-кратки пътища. В едно от най-новите си изследвания Ериксон [71] също използва тази формула, за да получи ефикасен алгоритъм за параметричен максимум в планарните графики. Всъщност това е една много по-обща линейна формула, изучавана от Карстенсен и Гасфийлд, която се отнася до времезависимите най-кратки пътища. В две статии, Деене [72, 73], както и други [74, 75] изучават време-зависимия най-кратък път по подобен начин на този, който е описан в статията. Линейните функции на цената представят алгоритъм, който е чувствителен към резултата с времева комплексност, пропорционална на броя на прекъсвания във функцията на времето за пристигане. Те също дават допълнителен алгоритъм (*additive approximation algorithm*) за полиномната комплексност в най-лошия случай. Време-зависимите най-кратки пътища са също изучавани от приложна гледна точка/ перспектива с акцент върху емпиричната проверка и приложения [76, 77, 44, 41].

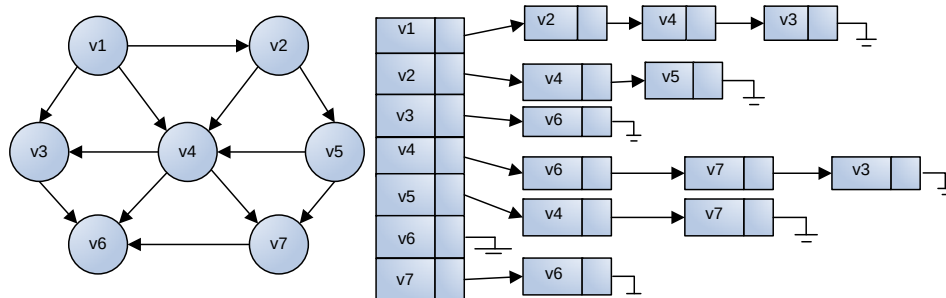
1.3 Основни алгоритми

Един от най-удобните начини за представяне на комуникационните мрежи и системи е чрез графи [78]. Всеки граф е изграден от възли (точки, върхове) и ребра, които ги съединяват и могат да бъдат насочени (еднопосочни) или ненаочени (двупосочни).

Представянето на граф може да се осъществи чрез двумерна матрица на съседство: на всяка дъга (u, v) се съпоставя един елемент от матрицата $A[u][v]$ със стойност 1 или теглото на графа, когато е претеглен. Липсата на дъги се означава с 0 или ∞ . Ако съществува път (u, v) , но не съществува (v, u) може да се запише (-1) за да се посочи ориентацията на дъгите в матрицата. При този начин на работа е необходима $\Theta(|V|^2)$ памет за съхранение на матрицата. Това води до разхищение на памет, особено при граф с малко дъги. Нещата са по-добре структурирани при неориентиран граф, където матрицата е симетрична относно главния диагонал и може да се съхрани като триъгълна: само под главният диагонал. В този случай, разхода на памет е съгласно:

$$\Theta\left(\frac{|V|(|V|-1)}{2}\right) \quad (1)$$

По-ефективно представяне за разреден граф е списъка на съседство. При него за всеки връх се създава списък на съседите му. Необходимата памет тук е $O(|E| + |V|)$. Проверката дали съществува път между два върха е по-сложна в този случай. На фигура 1.1 е показан такъв пример [79].



Фигура 1.1. Проверка за съществуване на път между два върха

За неориентиран граф, списъкът на съседство включва две явявания на всяка дъга, което води от своя страна до удвояване на паметта.

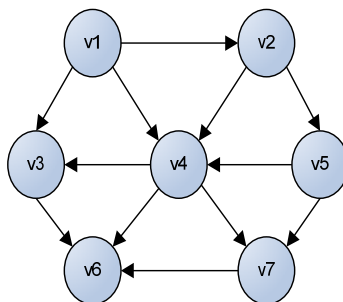
Представянето облекчава задачата за намиране на съседни върхове. Тя се извършва с просто сканиране, т.е. времето е линейно. По същият начин стои и задачата за търсене на наследници. Проблем възниква при търсене на предшественици: няма връзка в списъка и трябва да се сканират всички списъци. В този случай е по-добре да се използват двусвързани списъци.

1.3.1 Топологично сортиране

В ацикличен ориентиран граф може да се номерират върховете така, че за всяко ребро (v, w) , v да предшества w . Това се нарича топологично сортиране. За един граф може да има няколко топологични подредби. Пример за такава подредба може да бъде пътна карта. Топологично сортиране не съществува, ако в графа има цикли, защото за два върха v и w в един цикъл, v предшества w и w предшества v .

Намирането на топологична подредба може да се сведе до построяване на линейен списък Z от върховете на графа, такъв че за $\forall v, w \in V$, ако $\exists (v, w) \in E$, то v предхожда w в Z . Множеството от всички възможни Z е пълно топологично сортиране.

Такъв пример е показан на фигура 1.2 [79].

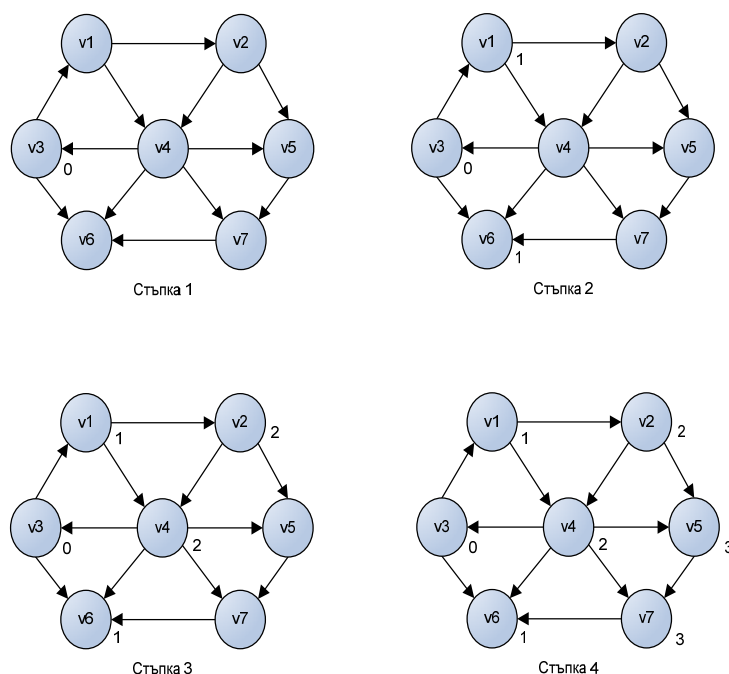


Фигура 1.2. Топологично сортиране

$v1, v2, v5, v4, v3, v7, v6$, както и $v1, v2, v5, v4, v7, v3, v6$ са две топологични подредби. Алгоритъмът първо открива връх без входящи ребра. После го премахва заедно с ребрата от него. Тази операция се повтаря докато не се изчерпи списъка с върховете без входящи ребра.

1.3.2 Намиране на най-къс път

Разглежда се алгоритъм за намиране на най-къс път при безтегловен граф [80, 81]. За зададен безтегловен граф $G=(V,E)$ и стартов връх v , се търси най-късия път от v до всички останали върхове в G . Разглежда се решението за графа, показан на фигура 1.3. Приема се за стартов връх $v3$. Тогава пътят до $v3$ е 0. Търсят се съседните на $v3$. Пътят до тях е 1. Избира се един от тях и отново се търсят съседи. Пътят вече е 2 и така до изчерпване на всички върхове. Тази стратегия на обхождане се нарича обхождане в ширина на графа и се използва при търсене по нива [79].



Фигура 1.3. Намиране на най-къс път

1.3.3 Алгоритъм на Дийкстра

Този алгоритъм е обобщение на предходния и носи името алгоритъм на Дийкстра. Това е типичен пример за greedy algorithm (алгоритъм решаващ проблема по нива, като на всяко ниво се прави това, което изглежда най-доброто). Пример за такава задача е какви монети трябва да се върнат до определена стойност, за да бъде броят им минимален. Проблем на такива алгоритми е, че не винаги работят. Пример за това са връщането на 15 стотинки, което алгоритъма ще даде като $12+3*1$, докато оптималното е $10+5$.

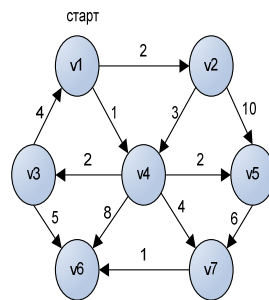
Алгоритъмът на Дийкстра работи на етапи, като за всеки етап избира връх v с минимална дължина d_v , измежду всички, неизползвани до момента и декларира най-късия път от s до v за известен. Следва актуализация на стойностите на d_w според

теглото на реброто (v,w) , т.е. $d_w = d_v + c_{v,w}$. Тази актуализация се прави само ако новото разстояние е по-късо от съществуващото. В противен случай разстоянието не се променя [79].

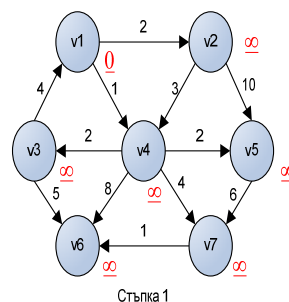
Разглежда се приложението на алгоритъма за графа, показан на фигура 1.4. Стартов връх за конкретният пример е $v1$.

Проследяват се отделните стъпки. Започва се от стартов връх $v1$ и се нанася разстояние 0 (фигура 1.5). Маркира се върха за обходен. Определят се неговите съседи. Те са $v2$ и $v4$ (фигура 1.6), след което мар $v4$ се маркира за обходен. Отново се определят съседите на $v4$ - $v3$, $v5$, $v6$ и $v7$ (фигура 1.7). Сега се избира $v2$. Неговите съседи са $v4$ и $v5$. Тъй като дължината на пътя $10+2=12$ е по-голяма от съществуващата (3), не се правят промени (фигура 1.8).

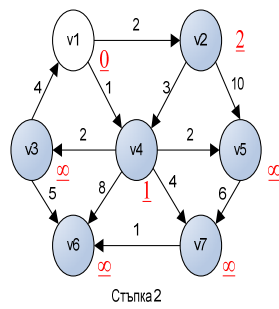
Следващият връх, който ще се обработи е $v5$. Той има само един съсед, който не е обходен - $v7$. Новото разстояние $3+6=9$ е по-голямо от 5 (съществуващото) (фигура 1.9). Следва избор на $v3$ и корекция на разстоянието на $v6$, което става 8 (фигура 1.10). Сега идва ред на $v7$. Отново се променя разстоянието на $v6$, което става 6 (фигура 1.11). Накрая се обхожда $v6$, който няма съседи и не води до промяна на разстоянията (фигура 1.12), след което алгоритъма спира [79].



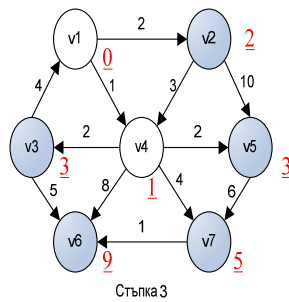
Фигура 1.4. Етапи на алгоритъма на Дийкстра



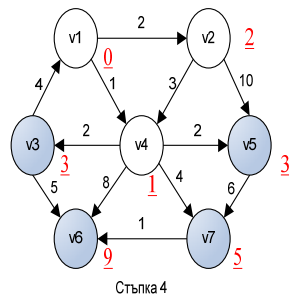
Фигура 1.5. Етапи на алгоритъма на Дийкстра



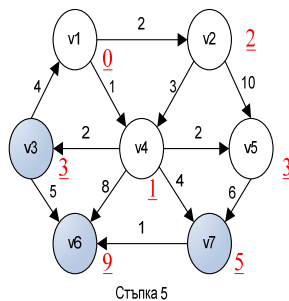
Фигура 1.6. Етапи на алгоритъма на Дийкстра



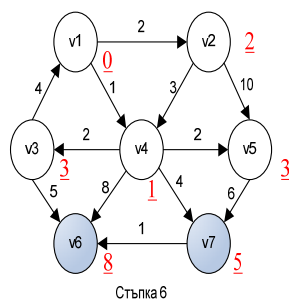
Фигура 1.7. Етапи на алгоритъма на Дийкстра



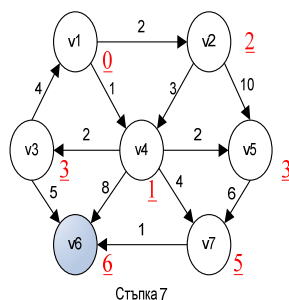
Фигура 1.8. Етапи на алгоритъма на Дийкстра



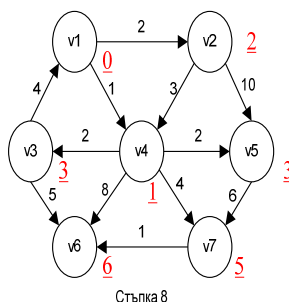
Фигура 1.9. Етапи на алгоритъма на Дийкстра



Фигура 1.10. Етапи на алгоритъма на Дийкстра



Фигура 1.11. Етапи на алгоритъма на Дийкстра



Фигура 1.12. Етапи на алгоритъма на Дийкстра

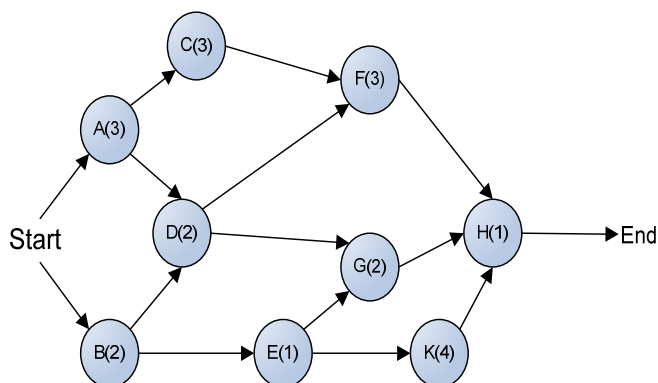
Алгоритъма няма да работи при отрицателни тегла.

1.3.4 Алгоритъм на Дийкстра при ациклични графи

Ако се знае, че графът е ацикличен, може да се подобри алгоритъма на Дийкстра, като се определи еднозначно последователността на избор на връх за обхождане. Това е последователността, съответстваща на топологичната подредба. Алгоритъм, базиран на подреден топологично граф, се обработва за един час. Това се дължи на факта, че пътят до избран според топологичната подредба връх не може вече да се намали, защото към него няма входящи ребра от необходени върхове [79].

Могат да се посочат различни примери от практиката: как да се достигне максимално бързо до дадена точка при движение само в една посока. Най - важен пример е оценката на дейностите и анализа на критичния път. Посредством разглеждане на граф на дейностите, необходими за завършване на софтуерен проект (пример за такъв граф е показан на фигура. 1.13) може да се отговори на следните въпроси [79]:

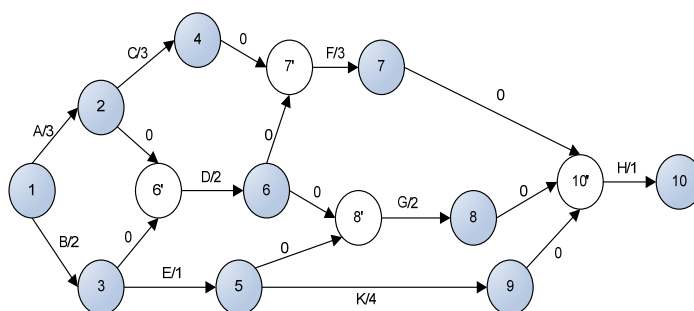
- Кое е най-късото време за завършване на проекта;
- Колко време могат да закъсняват отделните дейности, без това да се отрази на крайния срок.



Фигура 1.13. Теоретичен граф на дейности

В графът на дейностите всеки връх представя дейност, която трябва да се изпълни като е посочено и времето, което заема тя. Ребрата показват реда на изпълнение на дейностите. Графът е ацикличен и освен това допуска работа в паралелизъм за независещи една от друга дейности.

За решаване на поставената задача, трябва да се преобразува графът на дейностите в граф на събитията. При това всяко събитие отговаря на завършване на дейност и всички зависещи от нея дейности. За да се дефинират еднозначно зависимостите между отделните дейности, добавят се фиктивни (празни) възли, когато една дейност зависи от завършването на няколко други. За конкретния пример графът на дейностите е показан на фигура 1.14 [79].



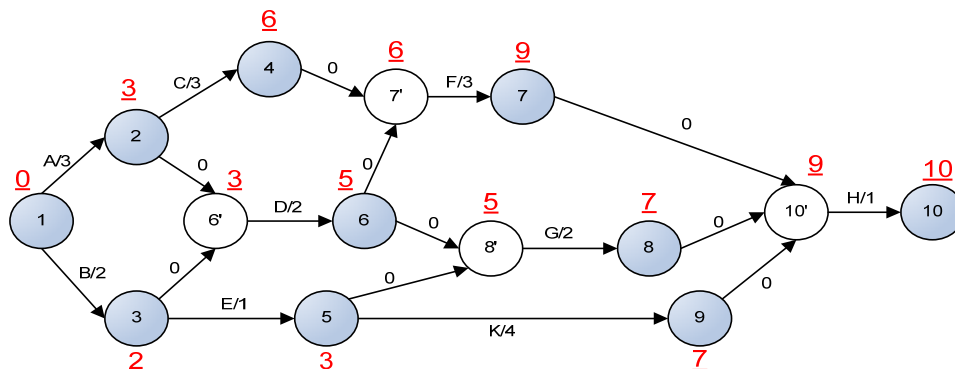
Фигура 1.14. Граф на дейностите

Търси се най-дългият път до края, за определяне ЕС (earliest completion time), най-краткото време за завършване на проект. При това се извършва адаптиране на алгоритъма за намиране на най-къс път. Ако EC_i е най-късото време за изпълнение на върха i , то изчислението става посредством [79]:

$$EC_1 = 0$$

$$EC_w = \max_{(v,w) \in E} (EC_v + c_{v,w}) \quad (2)$$

На фигура 1.15 е показан горния пример, след прилагане на формулите (резултатите за ЕСi са нанесени с подчертан шрифт) [54]:



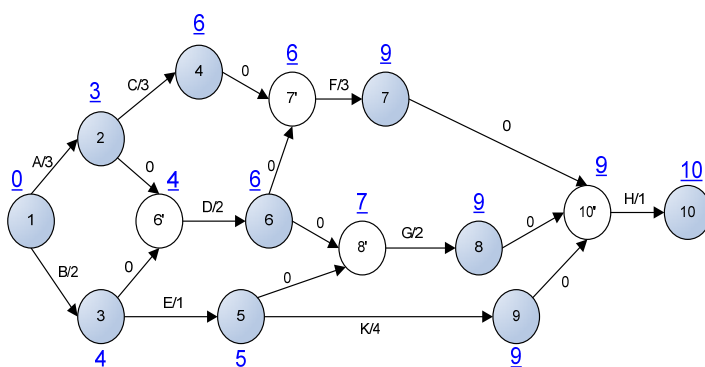
Фигура 1.15. Граф на дейностите - прилагане на резултати за най-кратко време

Може да се определи и най-късното време LC_i (latest completion for node i), до което може да се забави даден етап, без да се отлага крайния срок [79]:

$$LC_n = EC_n \quad (3)$$

$$LC_v = \min_{(v,w) \in E} (LC_w - c_{v,w})$$

На фигура 1.16 са показани резултатите от това изчисление, като резултатите за LC_i са нанесени с подчертан шрифт [79].



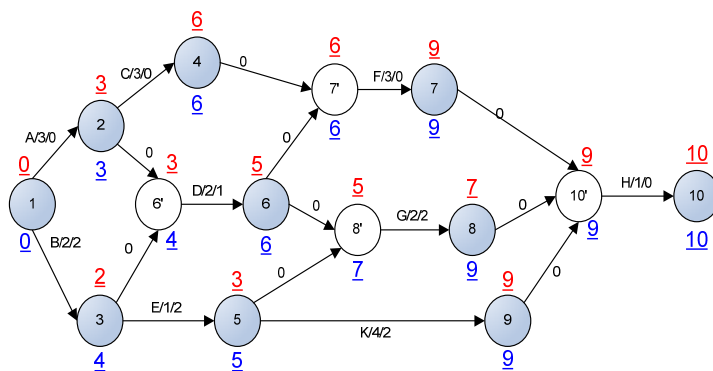
Фигура 1.16. Граф на дейностите - прилагане на резултати за най-късно време

Тези стойности могат да се изчислят за линейно време, за всеки връх, като се използва топологично сортиране за ЕСi и обратно топологично сортиране за LCi.

Може да се изчисли и луфта (slack) във време, с който се разполага. Той показва максималното закъснение за всеки връх, без да се просрочи крайния срок [79]:

$$slack_{(v,w)} = LC_w - EC_w - c_{v,w} \quad (4)$$

На фигура 1.17 е показан пример, като горната подчертана цифра показва EC_w , а долната - LC_w . Означенията по ребрата са: дейност/продължителност/луфт. Както се вижда, някои дейности нямат закъснение. Те са критични. Път, който съдържа такива дейности се нарича критичен [79].

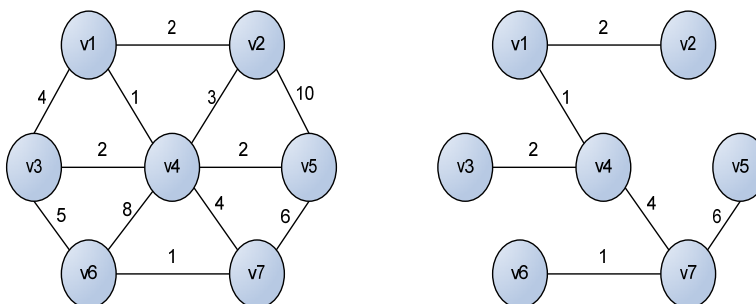


Фигура 1.17. Граф на дейностите за зададеният пример

1.3.5 Минимално обхващащо дърво

Търси се минимално обхващащо дърво в неориентиран граф. Задачата има смисъл и за ориентиран граф, но там алгоритъмът е по-труден.

За даден граф G , минималното обхващащо дърво е дървото, изградено от дъги на графа, които свързват върховете му на минимална цена. Минималното обхващащо дърво съществува само, ако графа е свързан. Ще се работи при това ограничение. Пример за приложение на посочения алгоритъм е окабеляване на мрежа. На фигура 1.18 е показан пример на граф и минималното му обхващащо дърво [79].



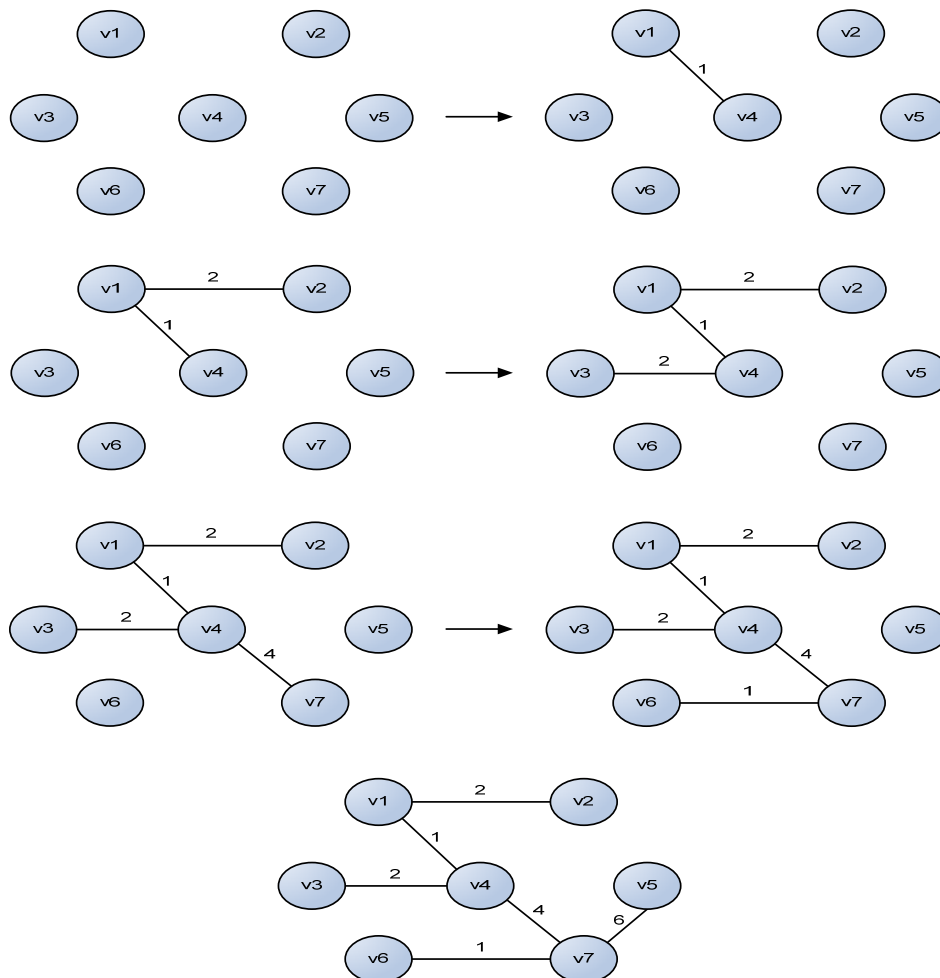
Фигура 1.18. Минимално обхващащо дърво на граф

Трябва да се обърне внимание, че минималното обхващащо дърво има $|V| - 1$ ребра. За всяко минимално обхващащо дърво T , ако ребро (v, w) , което не съществува в T се добави, се създава цикъл. Изтриването на всяко ребро от цикъла, възстановява минималното обхващащо дърво. Цената на минималното обхващащо дърво е толкова по-ниска, колкото по-ниска е цената на изграждащите го ребра, следователно при изграждането на минималното обхващащо дърво трябва да се добавя ребро с минимална стойност. Разглеждаме два алгоритъма, които се различават по избора на ребро с минимална стойност: алгоритъм на Прим и алгоритъм на Крускал.

1.3.6 Алгоритъм на Прим

При този алгоритъм се работи на етапи. На всеки етап се взима един връх v от дървото за корен, добавя се друг w от невключените до момента и се асоциира ребро в дървото. Измежду всички ребра (v, w) , за които v е от дървото, а w не е, избира се това, което има

минимална цена. На фигура 1.19 е показано как работи алгоритъма с начало $v1$. В началния момент $v1$ е дървото и на всеки етап се добавя по един връх и едно ребро [79].

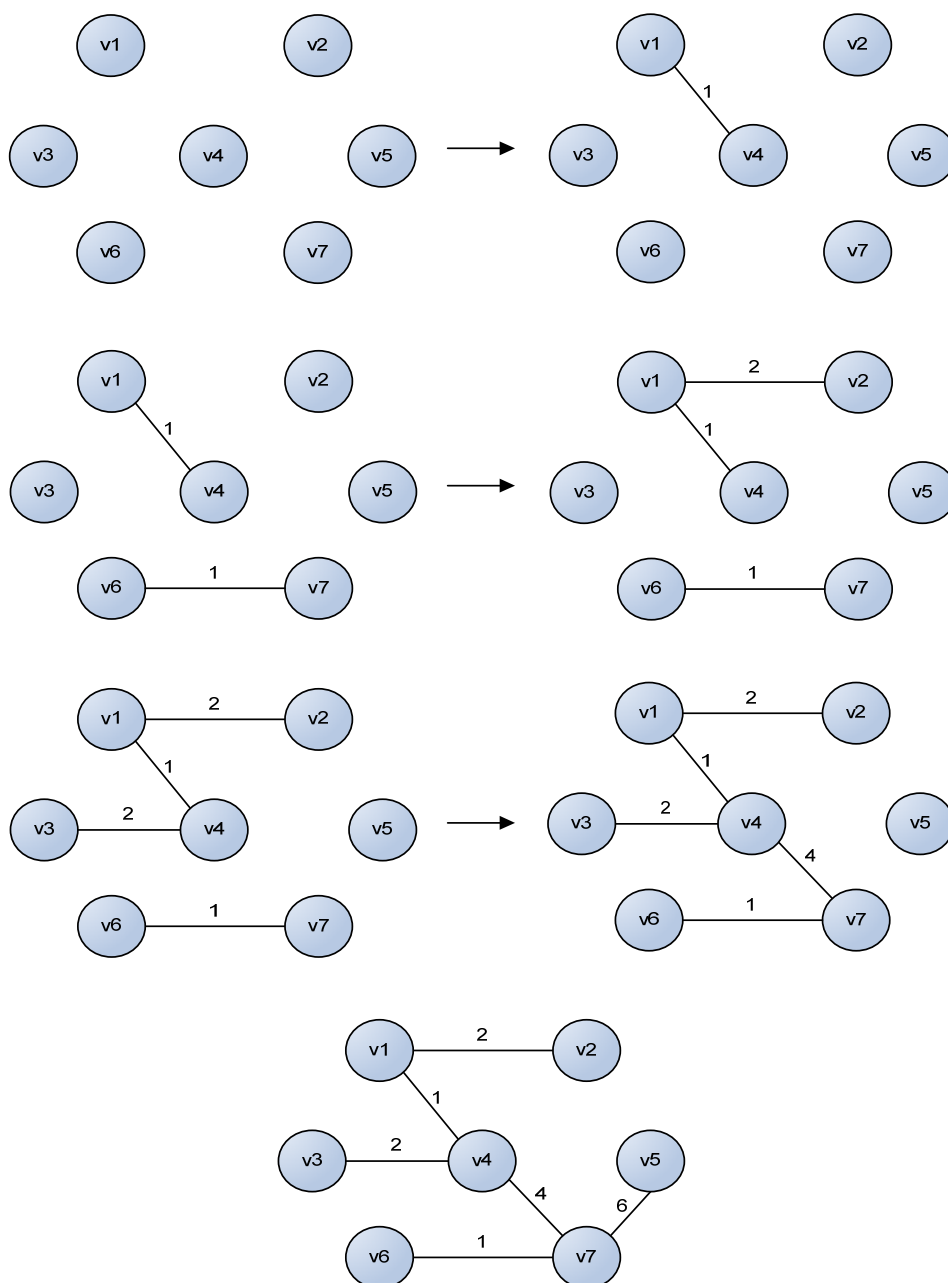


Фигура 1.19. Алгоритъм на Прим

Алгоритъмът е подобен на този на Дийкстра за намиране на най-къс път.

1.3.7 Алгоритъм на Крускал

Стратегията тук е да се избере ребро с най-малко тегло и да се одобри, ако не формира цикъл. За графа от предходния пример действията са показани на фигура 1.20 [79].



Фигура 1.20. Алгоритъм на Крускал

Погледнато формално, алгоритъма на Крускал работи с множество дървета. В началото това са $|V|$ броя дървета, състоящи се от по един връх. Добавяйки ребро, сливат се две дървета в едно. Алгоритъмът приключва когато са включени необходимия брой ребра. При това е налице само едно дърво и то е търсеното. При добавянето на ребро се използва следното правило: ако v и w са два върха в едно дърво, реброто, което ги свързва, не се избира за разглеждане, защото ще формира цикъл. В противен случай реброто се включва и двете дървета (включващи v и w) се сливат. Вижда се, че множествата са инвариантни (върховете в тях стоят постоянно) и към тях могат само да се добавят нови чрез сливане на дървета [79].

1.4 Методи за определяне на всички пътища между зададени два възела от граф

Използвани са два метода - рекурсивен метод чрез изчисляване на поддетерминантата (ресурсоемък метод) и метод чрез обхождане в широчина (оптимизационен метод).

За структурния анализ на комуникационните мрежи (намиране на пътища, сечения и техните характеристики), представени чрез графи е целесъобразно да се използват структурни матрици, както и някои основни булеви операции с тях. Всеки път (сечение) се представя като конюнкция на ребрата, образуващи този път (сечение).

1.4.1 Рекурсивен метод чрез изчисляване на поддетерминантата

Матрицата на свързаност на графа се състои от две матрици, които са в пълно съответствие една спрямо друга и са с размерност $n \times n$, където n е броя на възлите. Първата матрица е структурна и чрез нея се задават наименованията на ребрата между възлите, т.е. описан е начинът на свързване. Втората матрица съдържа тегловните коефициенти на ребрата, напр. числените капацитети на свързващите линии. Рекурсивният метод чрез изчисляване на поддетерминантата изисква от матрицата на свързаността да се намери поддетерминантата, която не включва стълба на началния възел и редицата на крайния възел. Определят се всички комбинации от маршрути, на брой $(n-1)$, и след премахване на циклите се получават всички възможни пътища. Броят им в най-тежкия случай, когато всички възли са взаимосвързани, е експоненциално зависим от броя на възлите. Предимствата на този метод са простия математически модел и възможността за работа без помощта на автоматизирани изчислителни машини. Той принадлежи на групата на ресурсоемките (неоптимизиран) методи.

1.4.2 Метод чрез обхождане в ширина/дълбочина

Този метод решава задачата за намиране на всички пътища между два възела чрез алгоритмичната схема обхождане на граф. Това е процедура, при която систематично, по определени правила "се посещават" всички върхове на графа. Съществуват два вида обхождане, които могат да се използват като алгоритмични схеми.

- Обхождане в ширина - съществено за тази алгоритмична схема е понятието *ниво на обхождане*, което представлява подмножество на множеството от върховете на графа, което в резултат на обхождането в ширина, се разбива на нива. *Дърво* се нарича краен свързан ацикличесен граф имащ поне два върха. Ако даден граф е дърво, то той има един *скелет (покриващо дърво)*, който съвпада с него. Само свързаните графи притежават скелет. В общия случай скелетът на графа не е определен еднозначно. Под обхождане на връх разбираме присъединяването му към дървото, съставено от по-рано обходени върхове. И така, даден е свързан граф, без всякакви ограничения, определен връх се избира за корен на бъдещото покриващо дърво на графа, построяването на което е крайния резултат от алгоритмичната схема обхождане в ширина.
- Обхождане в дълбочина- резултатът е отново изграждането на покриващото дърво, но то е по-високо и по-малко разклонено от полученото чрез обхождане в ширина.

И двата подхода строят коренови дървета, но имат принципни различия. Обхождането в ширина е неудобно поради това, че е необходимо да се помнят всички обходени върхове (или поне върховете от последното обходено ниво), за да може да се построи следващото ниво. За обхождането в дълбочина са достатъчни само върховете, разположени на пътя от началния до текущия връх. От друга страна, ако обхождането се прави с цел да се намери връх със зададени свойства и той се окаже на ниво с по-малък номер, тогава алгоритъмът в ширина ще го намери много по-бързо. Търсенето в дълбочина ще работи непредсказуемо дълго, защото може да се окаже, че връхът е измежду последните обходени (дори и да е на сравнително ниско ниво). При този вид задачи за предпочитане е схемата в ширина. Като компромис между двете схеми могат да се разработят *хибридни обхождания*, съчетаващи добрите страни на едната и другата схема, неутрализиращи, доколкото е възможно недостатъците им. Например, бързо спускане в дълбочина и след това обхождане в ширина на малкото оставащи върхове от "дъното".

1.5 Определяне на сечения посредством метод с булево инвертиране и оптимизиране на резултата

Това е метод за определяне на всички възможни сечения и избиране на най-малкото от тях чрез структурни матрици. От матрицата на свързаност се намират всички възможни пътища и цикли с максимална дължина $n-1$, където n е броят на възлите. Съставя се детерминантата на матрицата след при задраскване на стълба и реда съответно за източника, и за приемника на информация, и ако се изчисли параметрично и булево представлява всички сечения до степен n , където n е броят на точките в мрежата, при това включвайки цикли и повторения. След елиминирането на циклите с просто булево инвертиране на знаците в полученият резултат и минимизиране се получават всички възможни сечения. От тях се избира минималното. Съществено предимство на метода е, че могат да се изброят всички минимални сечения, а недостатък - резултатът само от първата стъпка на изчислителния процес е $(n-1)$.

1.6 Задачи за вземане на решение

Изследването на операции включва в себе си цял ред научни дисциплини, отличаващи се с различни по цели и методи на вземане на решение:

- *Математическото програмиране* изучава такива задачи за вземане на решение, в които най-доброто решение е това, което достига най-голяма стойност (или най-малка) на някой от показателите за ефективност [82]:

$$\max f(\bar{x}) \quad (5)$$

$$\bar{x} \in D$$

$$D = \{\bar{x} \mid g_i(\bar{x}) \leq b_i, \quad i = \overline{1, m}\} \quad (6)$$

където $f(\bar{x})$ – целева функция
 $g_i(\bar{x})$ – функция на ограниченията
 $\bar{x}$ – решение

Задача (5 - 6) се отнася към класа екстремални задачи. Ако областта на допустими решения D съвпада с пространството на реалните числа R , т.е. отсъстват ограниченията (2), то дадената екстремална задача се явява класическа задача за оптимизация [82].

- *Линейно програмиране.* Задача за линейно програмиране е задача на математическото програмиране (5 - 6), в която целевата функция и функцията на ограниченията са линейни. За такива задачи са разработени точни методи за решения.
- *Транспортни задачи* - задачи на линейното програмиране от специален тип, които имат по-ефективни методи за решение.
- *Задачи за назначения* - задачи на разпределение на работи между изпълнители с цел достигане на максимална ефективност.
- *Задачи за нелинейно програмиране* - задачи на математическото програмиране, в които поне една от функциите е нелинейна. В общия случай, тези задачи нямат точни аналитични методи за решение, а основните методи за тяхното решение са само приблизителни.
- *Задачи за изпъкнало програмиране* - задачи за нелинейно програмиране, имащи вдлъбнатата ($\cap$) целева функция и изпъкнала област ($\cup$) от допустими стойности. Това гарантира едноекстремалност на задачите и позволява да се формулира признак за оптималност на решението.
- *Задачи за квадратичното програмиране* - задачи на изпъкналото програмиране, имащи квадратична целева функция с линейни ограничения.
- *Задачи за дискретното програмиране* - задачи на математическото програмиране, имащи дискретна област на допустимите решения (в частност, крайно или бройно множество от решения).
- *Задачи за динамичното програмиране* - това са задачи, в които се прилагат стъпка по стъпка методи за решение.
- *Задачи за стохастичното програмиране* - това са задачи, в които се използват функции на случайни величини.
- *Векторната (многокритериална) оптимизация* изучава задачи за изследване на операции, в които се изисква да се обезпечи най-голямо (най-малко) значение (стойност) на няколко показателя за ефективност в една и съща област от допустими решения.
- *Теория на игрите* разглежда задачи за вземане на решение в конфликтни ситуации.
- *Теория за управление на запаси* изучава задачи за обема доставки и сроковете за съхранение на продукцията.
- *Мрежово планиране и управление* предлага методи за планиране на работи, свързани с мрежовите графици.
- *Теория на разписание или теория на календарното планиране* разглежда методи за планиране на работата във времето.
- *Симулационно моделиране* - моделиране на системи с помощта на електронно изчислителна техника.

- *Моята идея* - на база съществуващите методи и алгоритми, да се предложи решение за оптимизиране пропускателната способност на комуникационните потоци в реална мрежа.

1.7 Линејно програмирање

1.7.1 Общ вид на математически модел на задача за линейно програмиране [82]

В общ вид задачата за линейно програмиране може да се представи по следния начин:

Да се намери набор от управляеми параметри $\bar{x} = (x_1, x_2, \dots, x_n)$, които постигат най-големи (най-малки) стойности на показателя за ефективност

$$F(\bar{x}) = \sum_{j=1}^n c_j x_j \rightarrow \max(\min) \quad (7)$$

при изпълнение на следните ограничения

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + + a_{1n}x_n \leq b_1 \\ \\ a_{p1}x_1 + a_{p2}x_2 + + a_{pn}x_n \leq b_p \end{array} \right\} \quad (8)$$

$$\left. \begin{array}{l} a_{p+11}x_1 + a_{p+12}x_2 + \dots + a_{p+1n}x_n \geq b_{p+1} \\ \dots\dots\dots \\ a_{q1}x_1 + a_{q2}x_2 + \dots + a_{qn}x_n \geq b_q \end{array} \right\} \quad (9)$$

$$\left. \begin{array}{l} a_{q+11}x_1 + a_{q+12}x_2 + \dots + a_{q+1n}x_n = b_{q+1} \\ \dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \end{array} \right\} \quad (10)$$

и на някои променливи се наслагват условия за неотрицателност

$$x_1, x_2, \dots, x_r \geq 0, \quad r \leq n \quad (11)$$

Функция (7) се нарича целева функция или критерий за оптималност или линейна форма.

Векторът на управляемите параметри $\bar{x}$ се нарича решение. Решението се нарича допустимо, ако удовлетворява ограничения (8 - 11). Допустимото решение се нарича план.

Решението $\bar{x}^*$ се нарича оптимално, ако се постига най-голяма стойност на критерия за оптималност $F(\bar{x})$:

$$\bar{x}^* - \text{оптимально решение, ако } F(\bar{x}^*) = \max F(\bar{x}) \quad (12)$$

$$\bar{x}^* - \text{оптимально решение, ако за произвольно } \bar{x} \in D, \quad F(\bar{x}) \leq F(\bar{x}^*) \quad (13)$$

$$\bar{x}^* = \underset{\bar{x} \in D}{\text{Arg max}} F(\bar{x})$$



Задачата за линейното програмиране се нарича разрешима, ако има дори и само едно оптимално решение. При неразрешимите задачи или е празна областта на допустимите решения, или целевата функция не е ограничена.

1.7.2 Форми на задачата за линейно програмиране

В зависимост от вида на ограничението се различават следните форми задачи [82]:

- Канонична
- Симетрична
- Обща

Задача в канонична форма е задача за линейно програмиране, в която всички ограничения (8) - (10) имат равенства ($p = q = 0$) и всички променливи са неотрицателни ($r = n$).

$$F(\bar{x}) = \sum_{j=1}^n c_j x_j \rightarrow \max(\min) \quad (14)$$

$$\left\{ \sum_{j=1}^n a_{ij} x_j = b_i, \quad i = \overline{1, m} \right. \quad (15)$$

$$\left. x_j \geq 0, \quad j = \overline{1, n} \right\} \quad (16)$$

Общият метод за решение на задачи за линейно програмиране е разработен именно за задачи в каноничен вид.

Матричният вид на задачи в канонична форма е:

$$F(\bar{x}) = (\bar{c}^t * \bar{x}) \rightarrow \max \quad (14')$$

$$A \bar{x} = \bar{b} \quad (15')$$

$$\bar{x} \geq \bar{0} \quad (16')$$

където,

$$\bar{c} = \begin{pmatrix} c_1 \\ c_2 \\ \dots \\ c_n \end{pmatrix} - \text{вектор на коефициентите на критерия}$$

$$\bar{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$$

$$A = (a_{ij})_{m \times n} - \text{матрица на условията (технологични коефициенти)}$$

$$A = (\bar{A}_1, \bar{A}_2, \dots, \bar{A}_n)$$

$$\bar{A}_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \dots \\ a_{mj} \end{pmatrix} - \text{вектор на условията}$$

$$\bar{b} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix} \text{ - вектор на ограниченията}$$

$$F(\bar{x}) = (\bar{c}^T * \bar{x}) \rightarrow \max \quad (14'')$$

$$\bar{A}_1 x_1 + \bar{A}_2 x_2 + \dots + \bar{A}_n x_n = \bar{b} \quad (15'')$$

$$x_j \geq 0, \quad j = \overline{1, n} \quad (16'')$$

За нея е разработен метод за решение, който се нарича симплексен.

Задача в симетрична форма е задача за линейно програмиране, в която всички ограничения (8) - (10) имат неравенства ($p = q = m$) и всички променливи са неотрицателни ($r = n$).

$$F(\bar{x}) = \sum_{j=1}^n c_j x_j \rightarrow \max \quad (17)$$

$$\left\{ \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m} \right. \quad (18)$$

$$\left. x_j \geq 0, \quad j = \overline{1, n} \right\} \quad (19)$$

Матричният вид на задачата в симетрична форма е:

$$F(\bar{x}) = (\bar{c}^T * \bar{x}) \rightarrow \max \quad (17')$$

$$\left\{ A\bar{x} \leq \bar{b} \right. \quad (18')$$

$$\left. \bar{x} \geq \bar{0} \right\} \quad (19')$$

Симетричната форма допуска графично решение.

Задача в обща (смесена) форма е задача за линейно програмиране, в която присъстват всички видове ограничения и не всички променливи са неотрицателни.

1.8 Изводи

В настоящата глава са анализирани моделите за формализиране на информационни и комуникационни потоци в мрежи чрез графи и са оценени техните особености, предимства и недостатъци. Разгледани са проблемите за намиране на най-кратък път от един възел към всички останали възли в мрежи, за които цената на дъгата може да варира.

1.9 Цел и задачи

Цел: Посредством моделиране и управление на информационни потоци да се направи топологичен синтез и предложи оптимално решение за преминаване на съществуваща комуникационна мрежа от свързаност тип „звезда“ към двусвързана и трисвързана топология. Да се разработи модел за количествена оценка на качеството, непрекъсваемостта и отказоустойчивостта на обслужването с оглед технологично и структурно подобряване на комуникационната мрежа.

Задачи:

- 1) Да се анализират моделите за формализиране на информационни и комуникационни потоци в мрежи чрез графи и да се оценят техните особености, предимства и недостатъци. Да се разгледат проблемите за намиране на най-кратък път от един възел към всички останали възли в мрежи, за които цената на дъгата може да варира.
- 2) Да се определи и анализира пропускателната способност и времезакъсненията в мрежата при предаване на информационни потоци. Да се оцени пропадането на връзка между двойка възли като характеристика на отказоустойчивостта на съществуваща комуникационна мрежа със свързаност тип „звезда“.
- 3) Да се синтезира алгоритъм за определяне на теоритичните времезакъснения между възлите на добавените алтернативни комуникационни канали. Да се дефинира оптимизационна задача за определяне на свързаността на мрежата като се използват резултатите от измерванията на времезакъсненията и пропаданията в комуникационната мрежа.
- 4) Да се реши оптимизационна задача за определяне на структура на комуникационната мрежа като двусвързана и трисвързана. Да се анализират получаваните решения. Да се направи количествен анализ на получените резултати.

Глава 2. Определяне на пропускателните способности на канали чрез времеви измервания

2.1 Увод

В тази глава се определя и анализира пропускателната способност и времезакъсненията в мрежата при предаване на информационни потоци. Оценява се пропадането на връзка между двойка възли като характеристика на отказоустойчивостта на съществуваща комуникационна мрежа със свързаност тип „звезда“.

2.2 Инструменти за изчисляване на пропускателна способност на база времезакъснението

Пропускателна способност на комуникационен канал е максималното количество информация, което може да премине през този канал за единица време. Измерва се в bit/s. Съгласно теорията, това може да се изрази посредством формула 20:

$$C = \left(\frac{I}{t}\right) \quad (20)$$

където, I - количеството информация; t - време

Максимално достижимата пропускателна способност на даден канал зависи от широчината на честотната лента и отношението сигнал/шум, и се определя от Теорема на Шенон - Хартли, по имената на американските инженери Хартли и Шенон:

Теоремата на Шенон - Хартли след основен преглед на всевъзможните многонивови и многофазови методи за шифриране твърди, че пропускателната способност (капацитета) на канала C , т.е. теоретичната горна граница за скоростта на предаване на данни, които може да се предават със зададена средна мощност на сигнала S през аналогов комуникационен канал, който е под въздействие на адитивен бял гаусов шум с мощност N е равна на (формула 21):

$$C = B \log_2 \left(1 + \frac{S}{N}\right) \quad (21)$$

където,

C - пропускателна способност (капацитет) на канала, бит/секунда, bit/s;

B - честотна лента на пропускане на канала, Hz;

S - пълна мощност на сигнала в лентата на пропускане, W или V^2 ;

N - пълна мощност на шума в лентата на пропускане, W или V^2 ;

$\frac{S}{N}$ - отношение на сигнала към гаусовия шум като отношение на мощности.

Законът носи името на Клод Шанън и Ралф Хартли.

Измерването на времезакъсненията в комуникационна мрежа може да се осъществи посредством три метода - активно измерване, пасивно измерване, хибридно измерване.

При пасивното мрежово измерване информацията се събира посредством пасивно слушане на мрежовия трафик, например като се използват (оптични) линк сплитери или хъбове, които дублират трафика на връзките или чрез наблюдение на буферите в рутерите.

Повечето от съвременните устройства имат някакви вградени пасивни механизми за измерване като RMON, който може да се използва за събиране на различни типове данни от устройствата, като например количество изпратени байтове, загубени пакети и

друга статистика за интерфейса. Тези вградени механизми обикновено осигуряват обобщени данни и по този начин дават малко информация за статута на мрежата или поведението на трафика. Данните, създадени от тези механизми често могат да бъдат получени чрез използване на SNMP протокола. Друг механизъм е IPFIX, който събира данни от IP потока и след това ги изпраща към предварително конфигурирани приемници, като например централен сървър за наблюдения.

Ethereal (в днешни дни наречен Wireshark) и tcpdump са сред най-често използваните инструменти за пасивно измерване.

Основният проблем на пасивните измервания е количеството данни, които се генерират. Ако допуснем, гигабитова връзка с използване на 60% от капацитета си (на IP слоя) и среден размер на пакета 300 байта, то тогава степента на улавяне е около 250000 пакета на секунда. Скоростта на движение е 75 mebibytes (MiBs) за секунда и по този начин, необходимото пространство за съхранение за проследяване за един час е 270000 mebibytes, което се равнява на 270 gibibytes.

Ако има няколко точки за улавяне в мрежата, количеството на прихванатите данни ще създаде проблеми. В зависимост от вида на измерванията, налични са няколко метода за компресия: всички излишни (неподходящи) данни могат да бъдат премахнати от прихванатите пакети, в това число полезния товар и някои от полетата на заглавката на пакета. Нормалните методи за компресия могат да бъдат използвани за премахване на излишъка от пакетите (като пример, може да се използва софтуер за компресиране на информацията с оглед намаляване на необходимото пространство за съхранение). Също така, може да се използва моделиране на трафика, когато не се изисква негов пълен анализ. Моделирането може драстично да намали размера на пространството, необходимо за съхранение, но също така има и някои недостатъци (труден анализ на потока) и не всички методи за събиране на информация дават добри резултати. Ако единствено IP и хедърите от транспортния слой се съхраняват (40 bytes на пакет), изчислението за примера, даден по-горе, би дал скорост за трафика от 10 megabytes за секунда и 36 gigabytes пространство за съхранение, необходими за един час събиране на информация.

Анализът на прихванатата информация също може да създаде проблем. Анализът в реално време е труден, поради голямото количество информация.

Пасивните измервания имат някои предимства в сравнение с активните измервания. Пасивните методи не създават допълнителен трафик, като по този начин не пречат на мрежата и осигуряват точно представяне на мрежовия трафик.

Активните мрежови измервания генерират специални измерителни пакети, които се изпращат по мрежата, като например измерват времето, необходимо на един пакет да достигне до другия край на мрежата (еднопосочно закъснение), наличния капацитет на пътя на мрежата или времето за отговор на дадено приложение. За разлика от пасивните измервания, активните измервания генерират допълнителен трафик в мрежата и по този начин е възможно да наруши нормалния поток на трафика. Ето защо активните измервания трябва да бъдат внимателно планирани преди да бъдат направени и обикновено резервираната честотна лента за измерителните пакети е ограничена до не повече от 5% от общия капацитет на пътя. Такъв е случаят в повечето

SLA-измервания, където измерване се прави постоянно, т.е. тестовия трафик и клиентския трафик споделят една и съща честотна лента.

Някои методи, като например SLOPS, които се използват за измерване на честотната лента на пътя, включват изпращане на измерителни пакети на все по-голям процент и записване на стойността, при която закъсненията започват да се увеличават, т.е. пакетите започват да образуват опашки в даден момент. Тези методи водят обикновено до кратки смущения в потока на нормалния трафик. При пасивните измервания няма допълнителен трафик.

Активните измервания не изискват огромно количество пространство за съхранение и те могат да бъдат използвани за измерване на неща, които е невъзможно да се измерят чрез пасивните измервания. Когато се използва активно измерване, то не остава секретна информация, тъй като използваните данни не включват в себе си подобна информация. Всички активни измерителни пакети са изкуствени, т.е. те са създадени по поръчка и затова съдържат обикновено само случайни битове като полезен товар. Най-известните инструменти за активни измервания са `traceroute` и `ping`, които са вградени в повечето опериращи системи.

Комбинирането на активно и пасивно измерване се нарича хибридно измерване. Пример за хибридно измерване може да бъде сценарий, при който активните измервания са изпратени по мрежата и тяхното развитие се наблюдава посредством пасивни средства по време на измерването. Този тип класифициране позволява на измерителя да проследява пътя на измерването и да записва междинни и от край до край времезакъснения. Това е нещо, което е невъзможно да се направи с активно измерване. Сценария изисква административен достъп до междинните устройства (маршрутизатори).

Активните и пасивни измервания водят до различни типове информация и не винаги резултатите се корелират помежду си. Още по-подробно представяне на мрежата може да се получи чрез комбиниране на резултатите от двата вида активни и пасивни измервания (хибридни измервания).

Пасивните измервания подхождат най-добре към ситуации, в които точките за улавяне могат да бъдат свободно избирани. Например в ситуации, в които цялата мрежа е собственост на една организация или операторът е една организация. Това позволява трафикът да бъде обхванат от всяка една точка по пътя от изпращача до получателя.

В ситуации, в които това не е възможно, се използват активни измервания. Обикновено това са случаите, когато се измерват закъснения в работата на VPN, при преноса чрез ISPs.

Активните измервания могат да бъдат направени посредством мрежови път, по който има части, не контролирани от измерванията.

Когато става въпрос за точност на измерванията, пасивните методи са по-прецизни. Например загубата на пакети може да се измери много точно чрез наблюдения на буфери на рутерите по пътя на мрежата. Наличната честотна лента може да бъде точно измерена чрез наблюдения на използването на връзките между маршрутизаторите. Двата споменати начина за измерване е трудно да бъдат приложени.

За целите на дисертационния труд се използва активно мрежово измерване.

Изчисляването на пропускателната способност на комуникационна мрежа до съответните възли може да се осъществи посредством данни, получени от използването на мрежовите инструменти Ping, Traceroute и Pathchar.

Мрежовият инструмент ping е аббревиатура от „Packet Internet Groper“ [83]. Това е инструмент, който може да се използва за проверка, дали дадено мрежово устройство или компютър в интернет е налично (дали има връзка с него). Той изпраща пакет със заявка към съответната цел и очаква обратно получаване на пакет с отговор. Освен това, измерва времето между изпращането на пакета до целта и получаването на отговор. Това време се нарича време за пътуване (Round-Trip Time или RTT).

Traceroute е мрежов инструмент, който се опитва да проследи пътя на пакета през комуникационната мрежа [84, 85]. Интернет пътя между две точки преминава през много маршрутизатори, компютри и други устройства, взети заедно, което спомага за преминаване на информацията на IP пакетите. Целта е да се получи маршрут за изпратения пакет. Възможно е и използването на някои опции на IP пакета като запис на маршрут и негови варианти. Обикновено те не са строго определени и рядко се изпълняват по успешен начин. В добавка, тези опции могат да бъдат забранени от гледна точка на сигурността. Това са само част от причините за въвеждане в Traceroute като инструмент, който не зависи от никое от тези удобства [86].

Traceroute е IP инструмент, който позволява на потребителя да установи маршрута на пакетите, предназначени за определен хост. Той може да изследва пътя на пакетите, напуснали изпращача, до тяхното пристигане до крайната точка. Traceroute може да показва, през колко маршрутизатора преминават пакетите и колко време отнема пътуването им през тях. Също така може да даде имената на маршрутизаторите и тяхното мрежово присъединяване, както и географско местоположение, ако рутерите имат DNS (Domain System Service).

Pathchar е инструмент, разработен за да се направи извод на характеристиките на връзките през Интернет пътя [87, 88, 89, 90]. От глобална гледна точка на пътя, параметрите връзка - ниво като честотна лента, времезакъснение или закъснения в опашката, могат да бъдат изчислени посредством този инструмент. Когато пътя е избран, характеристиките на връзката по него могат да бъдат заключени посредством измерване на round trip time (RTT) на изпратения пакет от единичен хост.

Съществена разлика между Pathchar и Traceroute е тази, че Pathchar провежда така наречения влаков пакет анализ. Влаковият пакет е последователност от най-малко два пакета, изпратени по пътя. Различните характеристики на пакетите от различни влакове, като например техния размер, води до възможност за получаване на оценки на параметрите връзка - ниво. В този случай, измерването на RTT на влаковия пакет провежда анализ на пътя.

Също както Ping и Traceroute инструментите, Pathchar използва TTL поле в IP пакета. Най-голямото предимство на тези три инструмента, описано по-горе е отсъствието на синхронизация между началния и крайния възел. Ping инструмента например, заменя timestamp на Echo заявката в Echo отговора без да използва каквито и да било други механизми за връзка с хоста източник и хоста дестинация. Те работят независимо, осигурявайки RTT без да използват много усилия [91, 92].

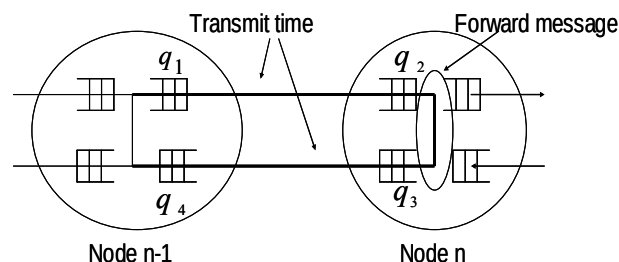
Измерването е определено като закъснение в една посока по време на пътя, при оценка на закъснението връзка - ниво. Ping, Traceroute и Pathchar изчисляват RTT, което е времето, необходимо за отиване и връщане. Закъснението в едната посока се получава чрез разделяне на RTT на две. Това е възможно само теоретично, когато препращащия пакета маршрут е същия като този, който връща пакета. Това е емпиричен подход, но е най-простия за употреба. Също така той е доста надежден при хипотеза, че мрежата е в стабилно състояние и локалния трафик по време на измерването е един и същ. В този случай, средните закъснения в опашката на маршрутизаторите са еднакви и в двете посоки. Тези предположения могат лесно да се прилагат в LAN, където една затворена мрежа може лесно да се управлява. В този аспект, трептенето (разликата) за RTT би могло да бъде малко. Горната граница на това трептене е 10% от общото изчислено RTT [93].

2.3 Изчисляване на пропускателна способност на отделните възли на база времезакъснението

За да се изчисли RTT [91, 94], важно е да се изследва как даден пакет преминава по пътя. На фигура 2.1 е показан типичен пример за изчисляване на RTT между два различни маршрутизатора. Определяйки RTT като n , нека съсредоточим вниманието си върху $(n-1)$ -тия и n -тия възел. Когато пакета достига до $(n-1)$ -тия възел, той изчаква известно време q в опашката за да премина към изходящата връзка. Времето за предаване е времето, което пакета губи във връзката. Това е линейна функция, изразена посредством формула 22, която зависи от времезакъснението и размера на пакета [91].

$$transmit = latency + \frac{size}{bandwidth} \quad (22)$$

Във възел n пакета изчаква за друго закъснение в опашката, q_2 . Маршрутизатора прочита нулево стойност на TTL на пакета и генерира ICMP съобщение. Дори ICMP пакета да изчака определено време в опашката за да бъде предаден, q_3 , то се връща към възел $(n-1)$ с време за предаване (формула 23). Накрая, той изчаква в опашката във възел $(n-1)$, q_4 [91].



Фигура 2.1 Компоненти на RTT са закъсненията в опашките q_i , времето за предаване и препредаване

RTT от $(n-1)$ -тия до n -тия възел и обратно е:

$$rtt = q_1 + (lat + \frac{packet_size}{bw}) + q_2 + forward + q_3 + (lat + \frac{message_size}{bw}) + q_4 \quad (23)$$

Опростявайки формула 23, могат да бъдат направени три хипотези [91]:

- Размерът на ICMP съобщението е достатъчно малък за да се разглежда като

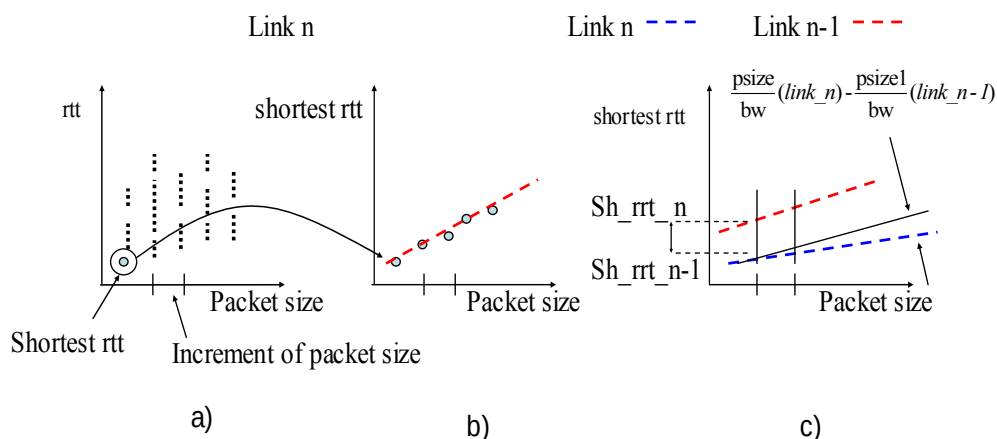
$$\frac{message_size}{bw}$$
незначително
- Обикновено, времето за препредаване в маршрутизатор е достатъчно малко, за да се разглежда като незначително време *forward*.
- За голям брой измервания на даден път, вероятността на събитието в незначително закъснение в опашката q_i не е нула.

Използвайки тези хипотези, формула 23 може да бъде заменена от формула 24 [91].

$$rtt = (lat + \frac{packet_size}{bw}) + lat \quad (24)$$

Pathchar измерва RTT, използвайки формула 24. Това изчисление се повтаря докато не се намери най-кратката възможна стойност. Всъщност, най-кратката стойност представлява RTT с най-голяма вероятност, получена посредством използването на формула 24 и нейните хипотези [95, 96].

Pathchar заключава времезакъснение и честотна лента посредством измерване за набор от размери на пакети за RTT във формула 24. RTT се изчертава спрямо размера на пакета и се избира най-краткия такъв [97, 98]. Фигура 2.2 показва разпръснат участък и относителен избор на RTT. Pathchar се използва за оценка на параметрите на връзката чрез изваждане на вертикални отсечки на съседна връзка. Вертикалната отсечка е общо времезакъснение, следователно, изваждайки n-тата стойност от (n-1)-тия ни дава двойно времезакъснение през връзка n и разделяйки го на две, се получава времезакъснението на връзка n [91].



Фигура 2.2 а) Разпръснат участък от всички измерени RTT. б) Pathchar избира най-краткия RTT от глобалния разпръснат участък. в) Pathchar намира времезакъснението и честотната лента чрез изваждане на пресечните точки и области на съседната връзка.

Pathchar заключава честотната лента посредством изваждане на отсечките на съседните връзки. По тази причина, разликата между наклоните на възли n и (n-1) дава противоположност на честотната лента на тази връзка. Обръщайки тази стойност, получава приблизителна оценка на честотната лента n на връзката.

Pathchar се използва за приблизителна оценка на минималната стойност на RTT за дадения размер на пакета. Увеличаването на размера на пакета е причина за неминуемо

увеличаване на RTT [99, 100, 101]. Получавайки размер на пакета, единствено минималното наблюдавано RTT се разглежда, оценявайки това, че всяко излишно време за предаване на пакета е причина за закъснение в опашката. Разпределянето на закъснението в опашката на възел n може да бъде оценено използвайки Pathchar, защото то е получено чрез натрупване на закъснения в опашката от предишни връзки.

Друг подход за пресмятане на пропускателната способност между два хоста (мрежови устройства) на база времезакъснението е използването на инструмента ping с пакети с различен размер. За целите на дисертационния труд ще се използва този мрежови инструмент [102].

Разглежда се формула за пресмятане на пропускателната способност R между два хоста (мрежови устройства) използвайки ping пакети с различен размер L и RTT, което обозначаваме със Z . Анализираме кои части допринасят за измерената RTT посредством инструмента ping за връзката между два хоста A и B . Предполагаме, че RTT Z за връзка с пропускателна способност R използвайки пакет с размер L е комбинация от закъснението при предаването (transmission delay) $\frac{L}{R}$ и други закъснения V . За това изчисление, пренебрегваме времето за обработка в приемащият хост B . Пакетът, който се връща има същия размер като изпратения пакет [103, 104]. Формулата за RTT е съответно:

$$Z = 2 \left(\frac{L}{R} + V \right) \quad (25)$$

Единствено знаем стойностите на Z и L , но не и на V така, че не можем да изчислим R . Въпреки това, ако измерим RTT Z за два различни размера на пакета L (и приемем, че V остава с един и същ размер за всеки пакет), то може да елиминираме V . Това води до следните две формули 26 и 27:

$$V = \frac{Z_1}{2} - \frac{L_1}{R} \quad (26)$$

$$V = \frac{Z_2}{2} - \frac{L_2}{R} \quad (27)$$

Приемаме, че $L_1 > L_2$ и $Z_1 > Z_2$. Сега елиминираме V и получаваме:

$$R = 2 \frac{L_1 - L_2}{Z_1 - Z_2} \quad (28)$$

Така полученият резултат дефинира модел за определяне на пропускателната способност на комуникационните канали на съществуващата топология тип „звезда“ на база измерени времезакъснения.

С две измервания (с два различни по стойност размера на пакета) можем да изчислим постигнатата пропускателна способност между два хоста [105].

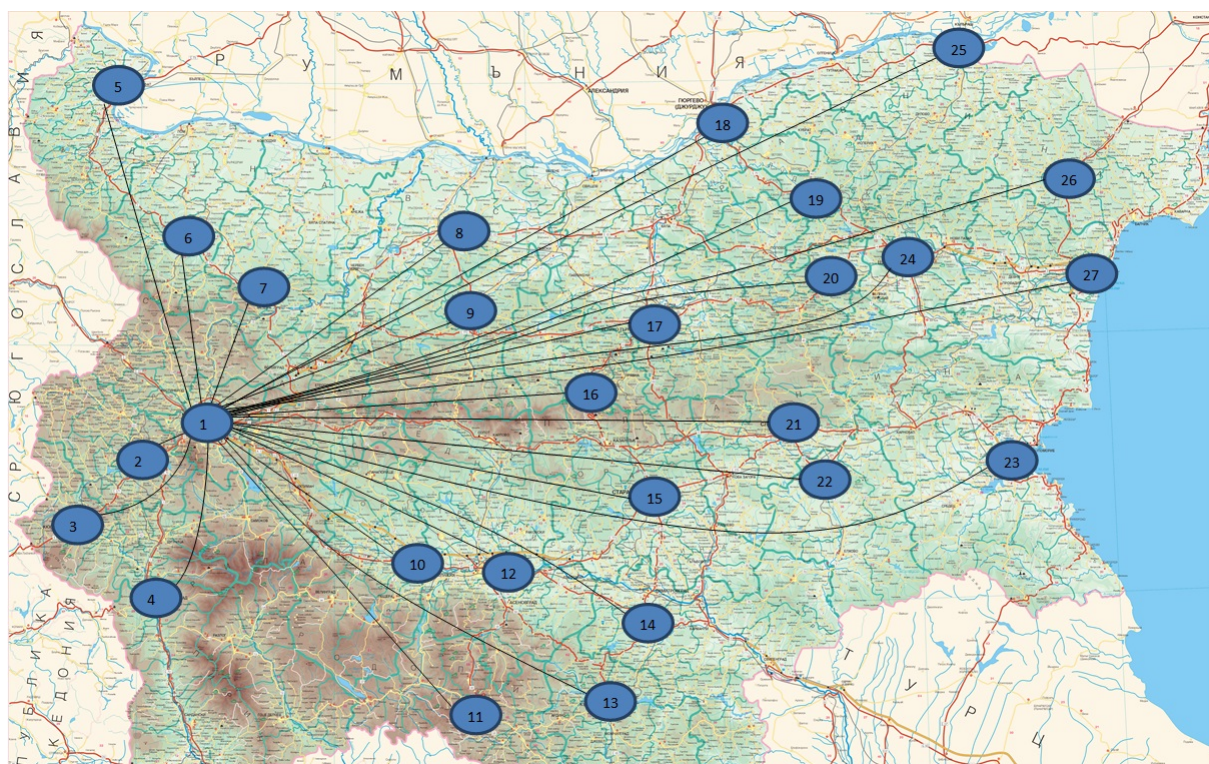
Размера на пакета може да се променя с помощта на -l параметъра в команден ред (параметър на инструмента ping) [106].

Използват се пакети с размери 100 bytes и 1300 bytes. Важна особеност е, че винаги трябва да се използват пакети по-малки от 1400 bytes. Ако крайното устройство (мрежовото устройство) получи 1500 bytes пакет (20 byte IP header + 1480 TCP payload), то игнорира пакета и не се връща отговор на заявката. Това се получава, тъй като изпращаната заявка е по-голяма от IP MTU (Maximum Transmission Unit)- 1476bytes [107, 108].

Разглежданата комуникационна мрежа представлява клонова мрежа с 27 възела (таблица 2.1, фигура 2.3) и централизирано управление - топология „звезда“, като централната точка се намира в ИО-София.

Таблица 2.1

Възел / клон	Град
1	София
2	Перник
3	Кюстендил
4	Благоевград
5	Видин
6	Монтана
7	Враца
8	Плевен
9	Ловеч
10	Пазарджик
11	Смолян
12	Пловдив
13	Кърджали
14	Хасково
15	Стара Загора
16	Габрово
17	Велико Търново
18	Русе
19	Разград
20	Търговище
21	Сливен
22	Ямбол
23	Бургас
24	Шумен
25	Силистра
26	Добрич
27	Варна



Фигура 2.3 Комуникационна клонова мрежа мрежа с 27 възела тип „звезда“

Представеното измерването се осъществява в посока от ИО-София (хост А) към останалите възли.

На фигури 2.4 и 2.5 са показани резултатите за времето RTT между изпращането на пакета до целта и получаването на отговор от отдалеченото мрежово устройство (хост В - ИО-Варна) в команден ред.

```
C:\Users\ktrichkov.UN>ping 10.244.62.2 -l 100

Pinging 10.244.62.2 with 100 bytes of data:
Reply from 10.244.62.2: bytes=100 time=1ms TTL=63
Reply from 10.244.62.2: bytes=100 time=1ms TTL=63
Reply from 10.244.62.2: bytes=100 time=1ms TTL=63
Reply from 10.244.62.2: bytes=100 time=1ms TTL=63

Ping statistics for 10.244.62.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

Фигура 2.4 RTT между изпращането на пакета до целта и получаването на отговор при размер на пакета 100 bytes

```
C:\Users\ktrichkov.UN>ping 10.244.62.2 -l 1300

Pinging 10.244.62.2 with 1300 bytes of data:
Reply from 10.244.62.2: bytes=1300 time=4ms TTL=63
Reply from 10.244.62.2: bytes=1300 time=4ms TTL=63
Reply from 10.244.62.2: bytes=1300 time=5ms TTL=63
Reply from 10.244.62.2: bytes=1300 time=5ms TTL=63

Ping statistics for 10.244.62.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 5ms, Average = 4ms
```

Фигура 2.5 RTT между изпращането на пакета до целта и получаването на отговор при размер на пакета 1300 bytes

На база получените резултати и съгласно формула 28, изчисляваме пропускателната способност на канала между възли (хостове) А и В. За по-голяма точност приемаме, че средното време RTT при размер на пакета 1300 bytes е 4,5 ms.

$$R = 2 \frac{100 - 1300}{1 - 4,5} = 685 \text{ bytes/ms} = 5,48 \text{ M bits/s}$$

За изчисляването пропускателната способност на комуникационната мрежа до съответните възли е използван софтуерния продукт PRTG Network Monitor.

Управлението и наблюдението на едно или няколко мрежови устройства не е сложна задача и за целта можете да се използва софтуера, който се предоставя със самите тях. Тези задачи обаче стават трудни и дори непосилни, когато се работи с мрежи, с голям брой устройства, които трябва да се управляват, наблюдават и конфигурират през цялото денонощие. По тази причина, използването на специализирания софтуер PRTG Network Monitor на германската компания Paessler, спомага за да се направят мрежите по-бързи, по-отзивчиви и по-надеждни. PRTG е достъпно решение, което може лесно и бързо да се надгражда на база дефинирани изисквания и при евентуално бъдещо разширение на използвана мрежа.

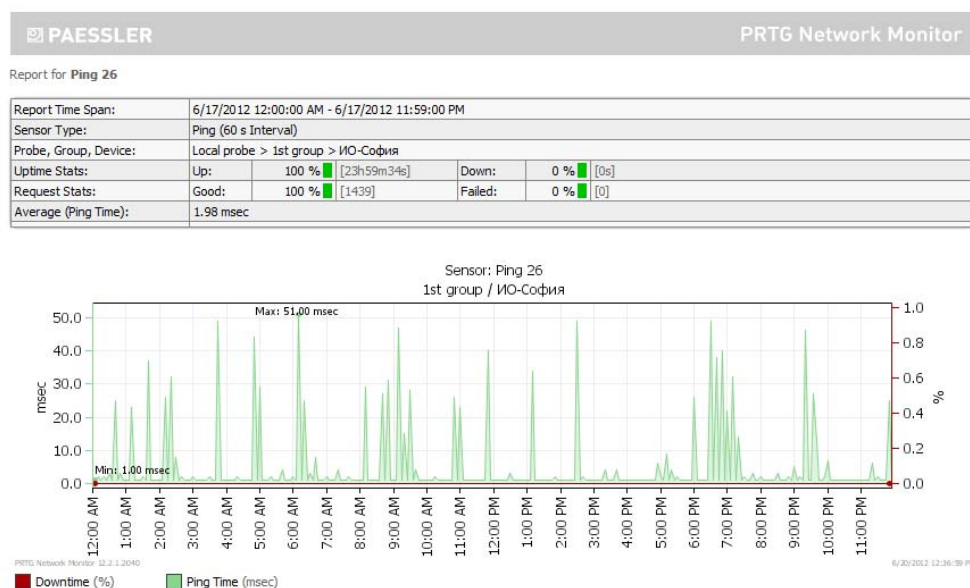
PRTG Network Monitor е мощен софтуерен инструмент, но същевременно се използва лесно и е предназначен за мрежовите специалисти за непрекъснато наблюдение на мрежите - 24 часа в денонощието, 7 дни в седмицата.

Данните се събират с помощта на различни технологии, протоколи и методи, от рода на WMI, SSH, SNMP, NetFlow, jFlow, sFlow, пакетно филтриране и др. Решението предлага над 130 вида софтуерни сензори, предназначени да наблюдават голям брой параметри, като например натоварването на процесора на определени системи, потока данни през техните мрежови интерфейси, свободното дисково пространство на устройства за съхранение на данни, работата на HTTP, SMTP, POP3, FTP сървъри и т. н. Споменатите сензори не са софтуерни компоненти, които се инсталират на наблюдаваните клиентски системи, а работят на централна машина или машини и от тях се наблюдава състоянието на клиентските системи чрез описаните вече протоколи. Лицензът на PRTG Network Monitor зависи от броя сензори - приложението е безплатно при до 10 сензора, като може да се добавят и закупуват допълнителни сензори, в случай на нужда и необходимост. Могат да се създават зависимости между отделните сензори - например за наблюдение на уеб-сървър се използва ping сензор и HTTP сензор и ако ping сензорът отчете грешка, системата веднага поставя съответния HTTP сървър в състояние на изчакване (администраторът ще получи само едно съобщение, че уеб-сървърът не работи, а не две, което намалява объркването в мрежи с голям брой сензори). Възможно е да се настройат и по-сложни сензори, които не само да следят за наличието на дадена услуга, но и да симулират определени действия (например имитация на покупка от онлайн магазин) и да изпълняват специфични скриптове, за да се гарантира, че услугите работят така, като се очаква.

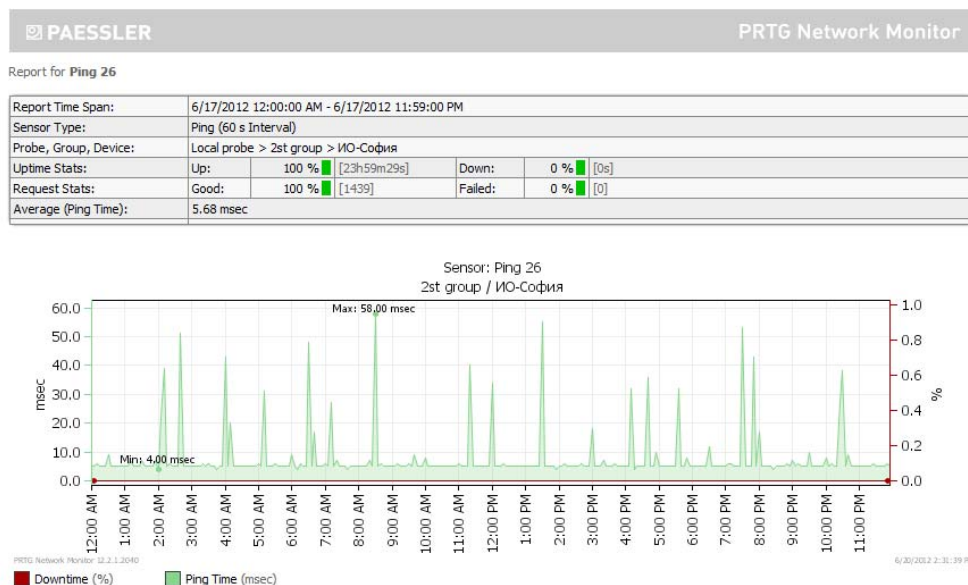
Изпращането на ping заявка се извършва на всеки 60 секунди. На база получените резултати при зададени размери на пакета 100 bytes и 1300 bytes се анализират и описват получените резултати в графичен вид.

2.3.1 Изчисляване на пропускателна способност на възел ИО-Варна

На фигура 2.6 и фигура 2.7 е показана информация за възела при изследване на времезакъснението за период от 24 часа и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.6 Статистиката на резултати при размер на пакета 100 bytes

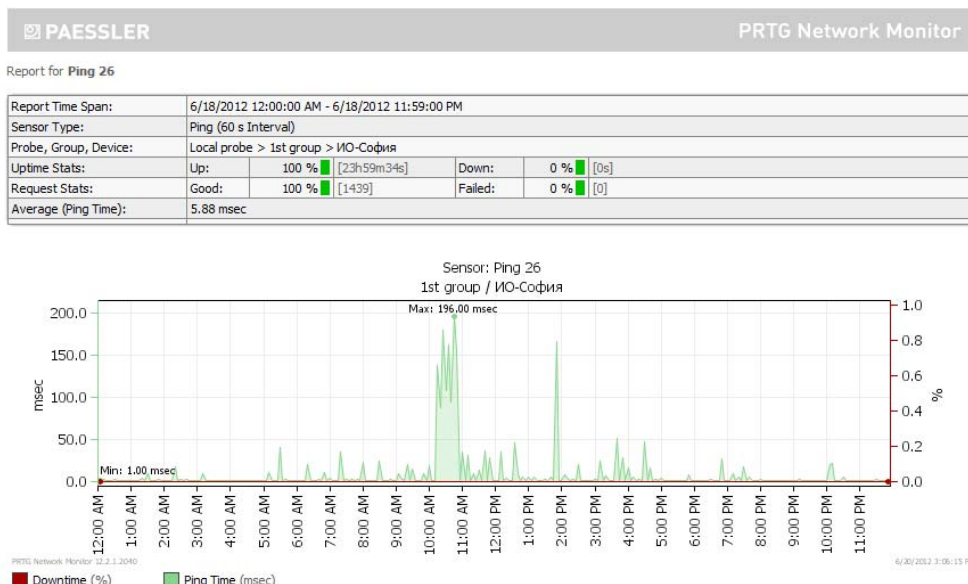


Фигура 2.7 Статистиката на резултати при размер на пакета 1300 bytes

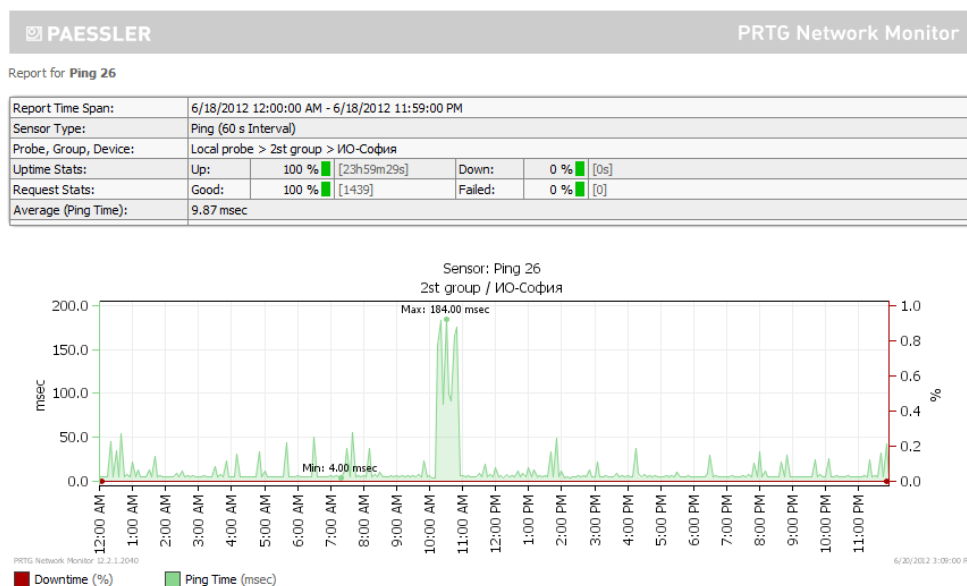
На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

$$R = 2 \frac{100 - 1300}{1,98 - 5,68} = 648 \text{ bytes/ms} = 5,19 \text{ M bits/s}$$

На следващите две фигури 2.8 и 2.9 е показана информация за възела при натовареност на канала в размер на 1 M bytes/s при изследване на времезакъснението за период от 1 час и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.8 Статистиката на резултати при натовареност на канала и размер на пакета 100 bytes



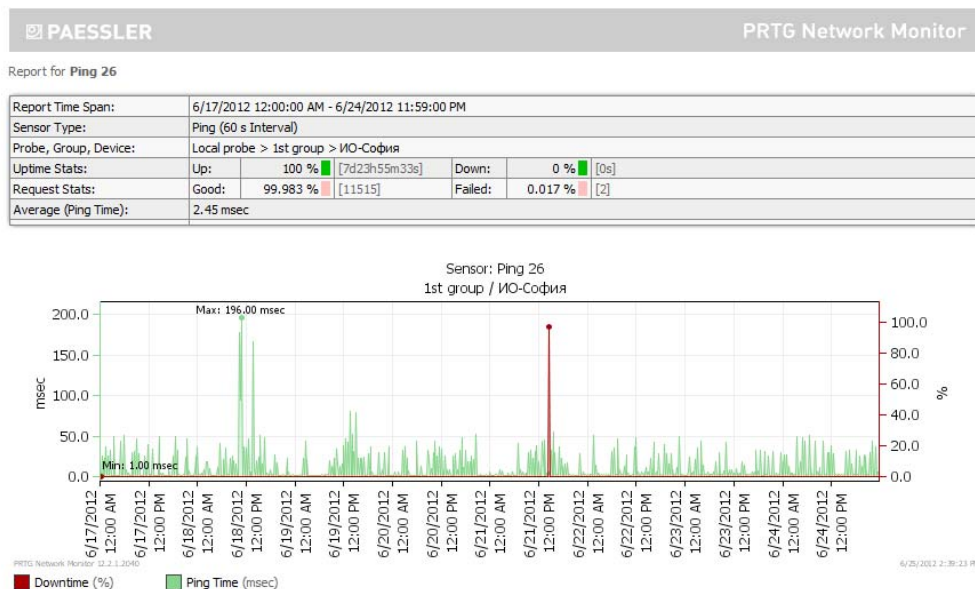
Фигура 2.9 Статистиката на резултати при натовареност на канала и размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност при натовареност на връзката до възела получаваме:

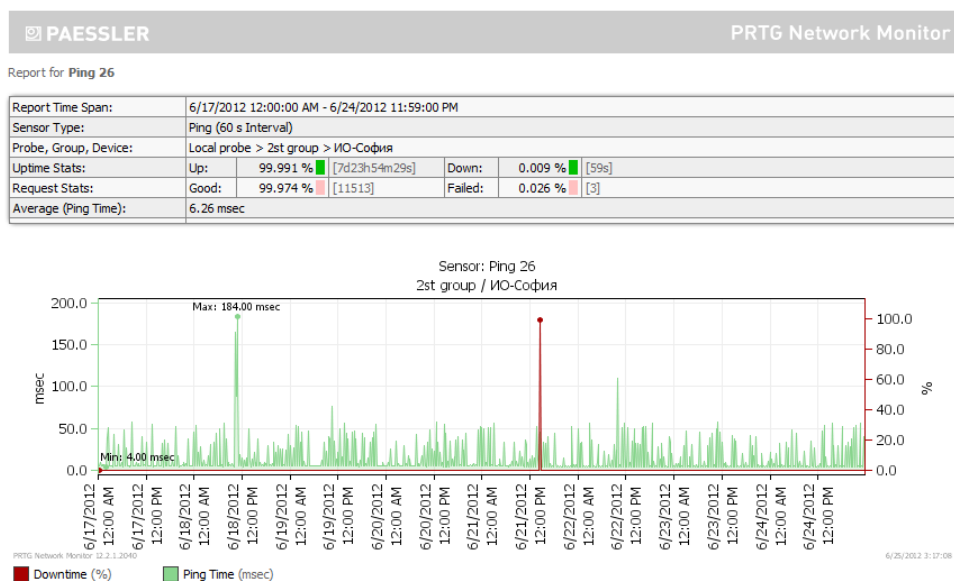
$$R = 2 \frac{100 - 1300}{5,88 - 9,87} = 601 \text{ bytes/ms} = 4,81 \text{ M bits/s}$$

От фигури 2.8 и 2.9 се вижда, че при натоварен канал в рамките на един час, времето за отговор се увеличава, но средната пропускателната способност за периода не спада драстично.

На фигура 2.10 и фигура 2.11 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.10 Статистиката на резултати при размер на пакета 100 bytes



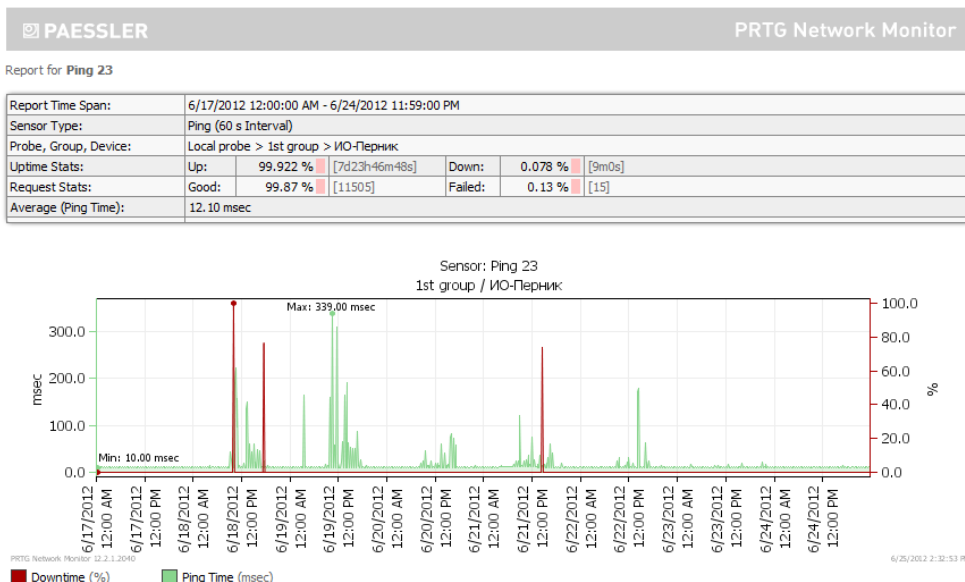
Фигура 2.11 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

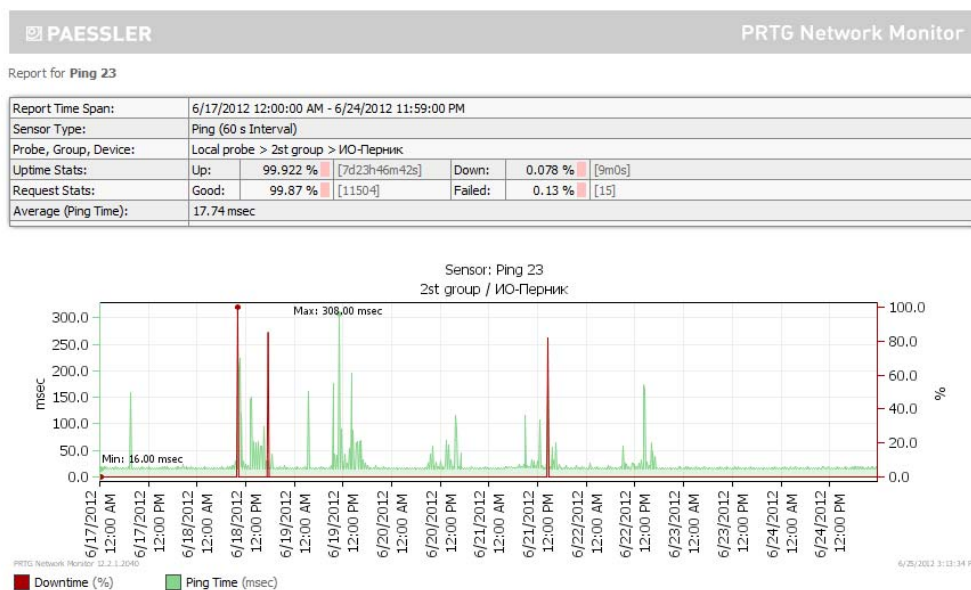
$$R = 2 \frac{100 - 1300}{2,45 - 6,26} = 630 \text{ bytes/ms} = 5,04 \text{ M bits/s}$$

2.3.2 Изчисляване на пропускателна способност на възел ИО-Перник

На фигура 2.12 и фигура 2.13 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.12 Статистиката на резултати при размер на пакета 100 bytes



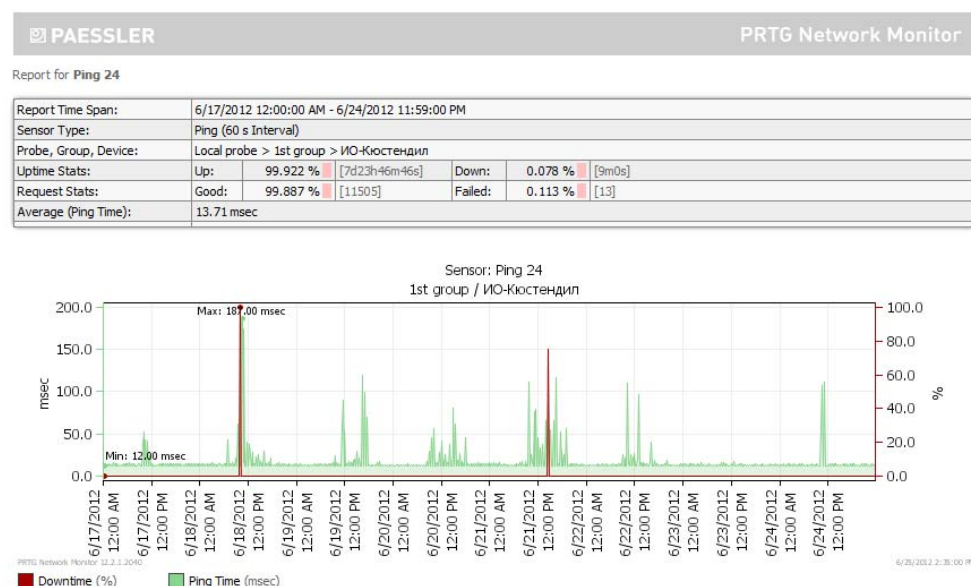
Фигура 2.13 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

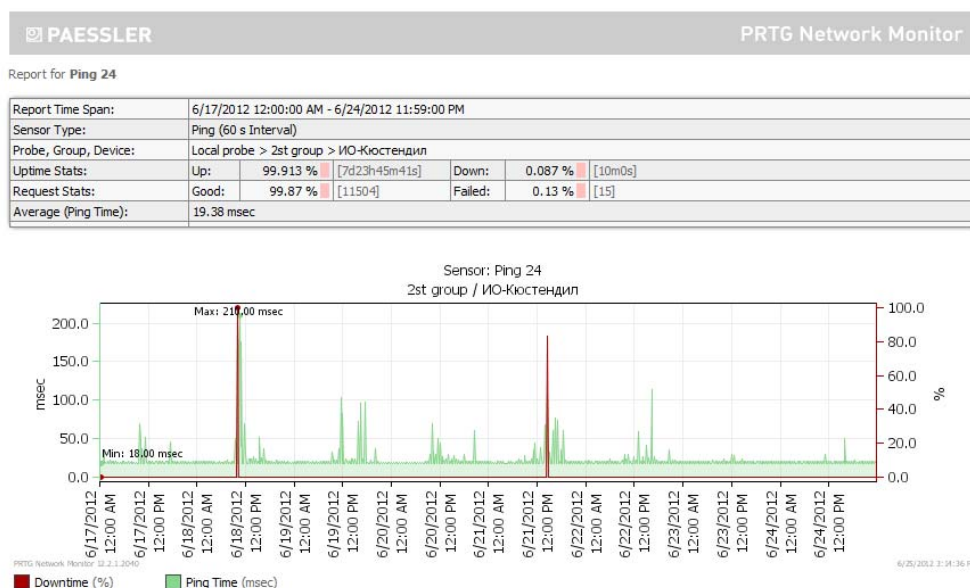
$$R = 2 \frac{100 - 1300}{12,10 - 17,74} = 425,5 \text{ bytes/ms} = 3,4 \text{ M bits/s}$$

2.3.3 Изчисляване на пропускателна способност на възел ИО-Кюстендил

На фигура 2.14 и фигура 2.15 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.14 Статистиката на резултати при размер на пакета 100 bytes



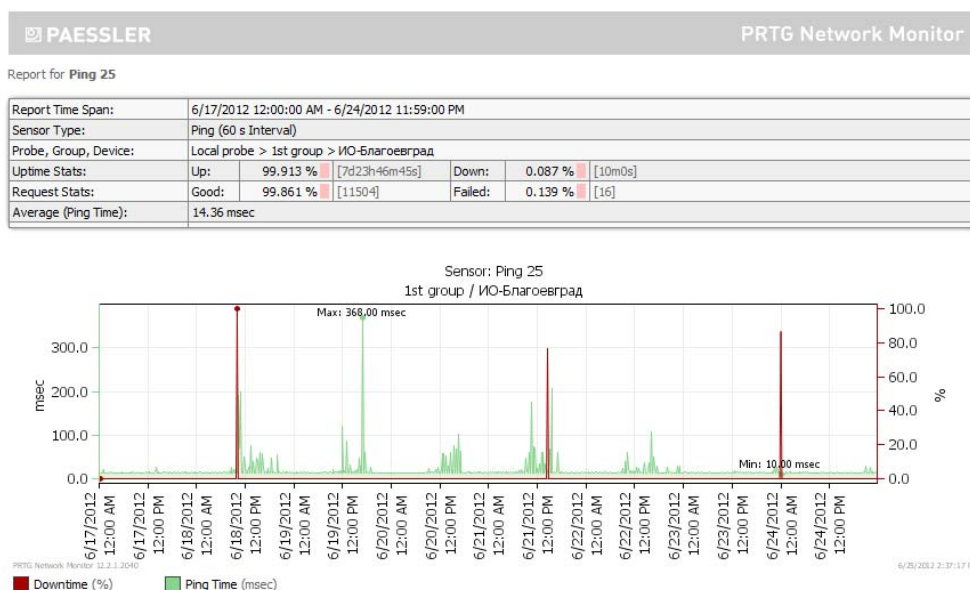
Фигура 2.15 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

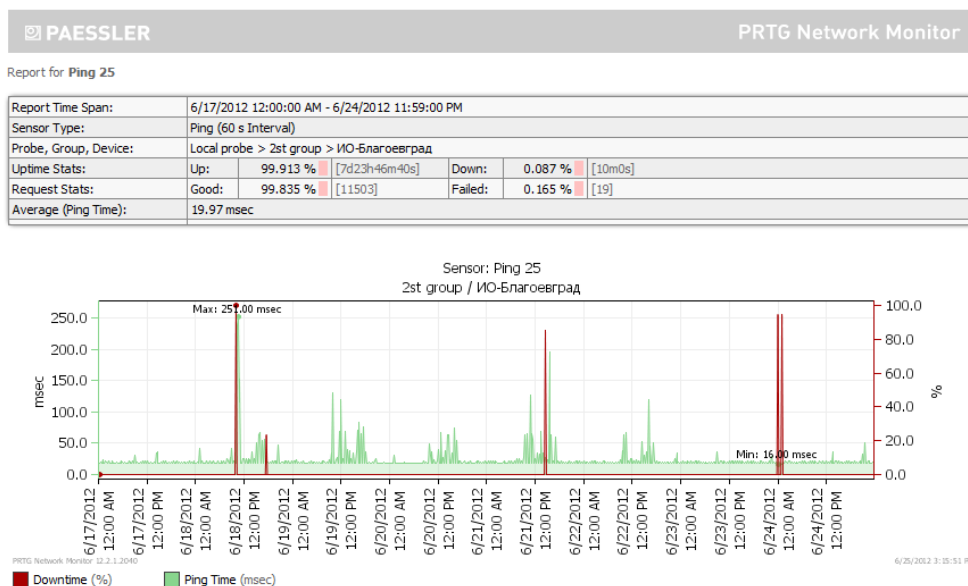
$$R = 2 \frac{100 - 1300}{13,71 - 19,38} = 423 \text{ bytes/ms} = 3,38 \text{ M bits/s}$$

2.3.4 Изчисляване на пропускателна способност на възел ИО-Благоевград

На фигура 2.16 и фигура 2.17 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.16 Статистиката на резултати при размер на пакета 100 bytes



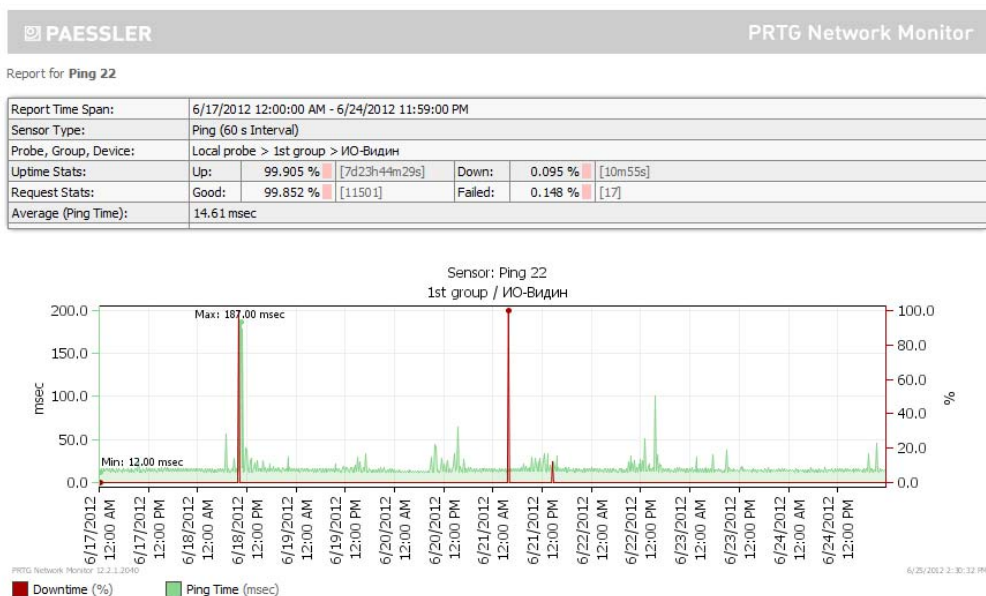
Фигура 2.17 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

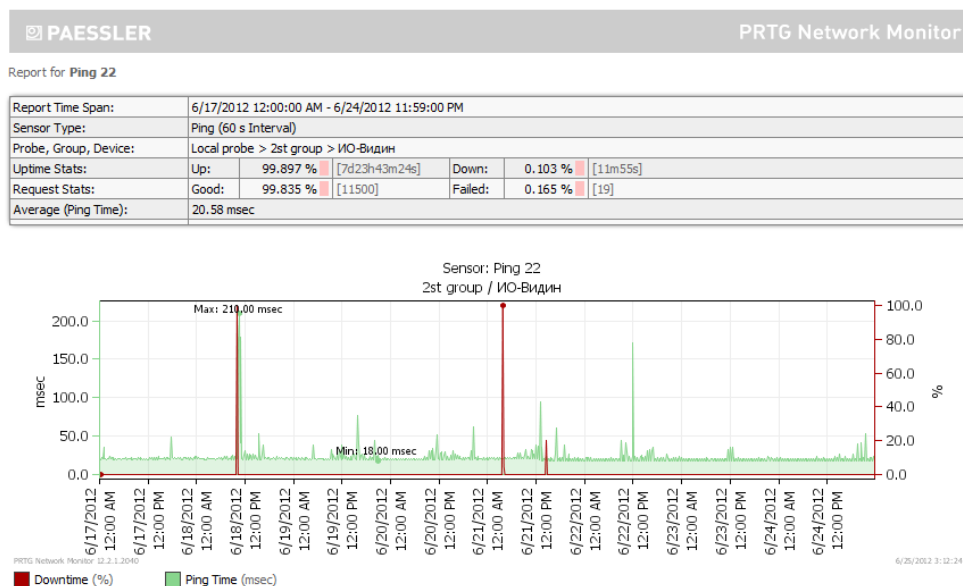
$$R = 2 \frac{100 - 1300}{14,36 - 19,97} = 427,8 \text{ bytes/ms} = 3,42 \text{ M bits/s}$$

2.3.5 Изчисляване на пропускателна способност на възел ИО-Видин

На фигура 2.18 и фигура 2.19 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.18 Статистиката на резултати при размер на пакета 100 bytes



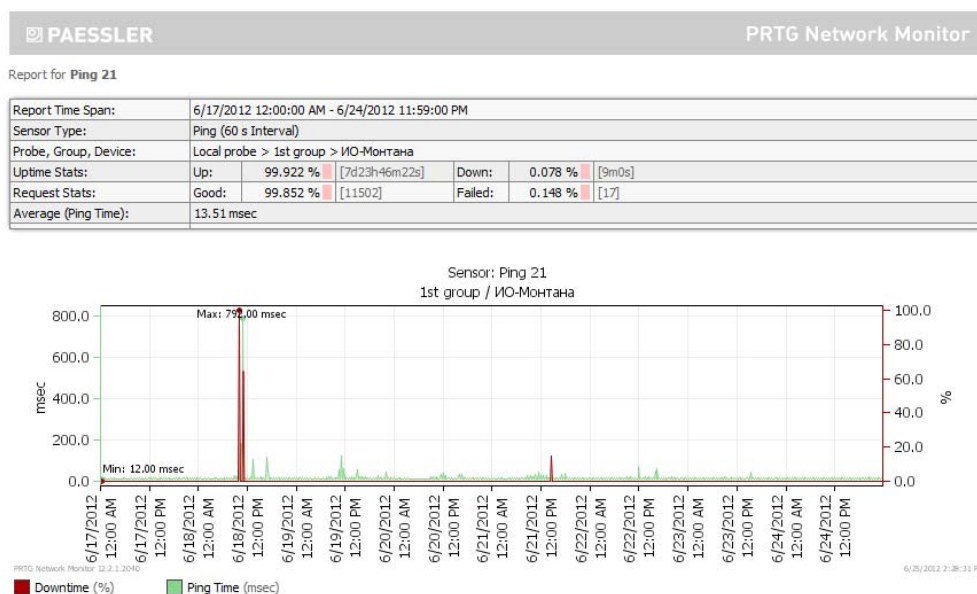
Фигура 2.19 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

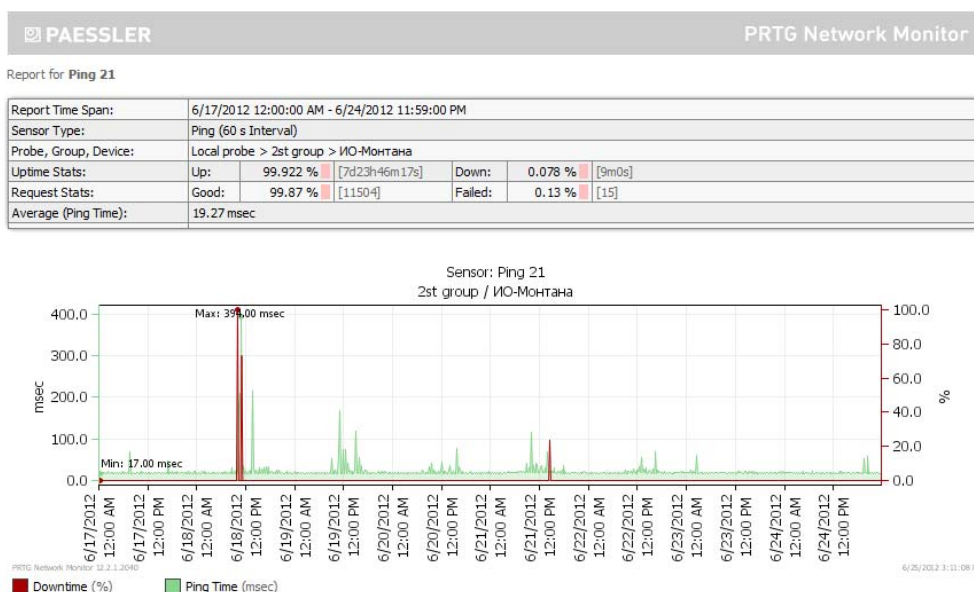
$$R = 2 \frac{100 - 1300}{14,61 - 20,58} = 402 \text{ bytes/ms} = 3,22 \text{ M bits/s}$$

2.3.6 Изчисляване на пропускателна способност на възел ИО-Монтана

На фигура 2.20 и фигура 2.21 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.20 Статистиката на резултати при размер на пакета 100 bytes



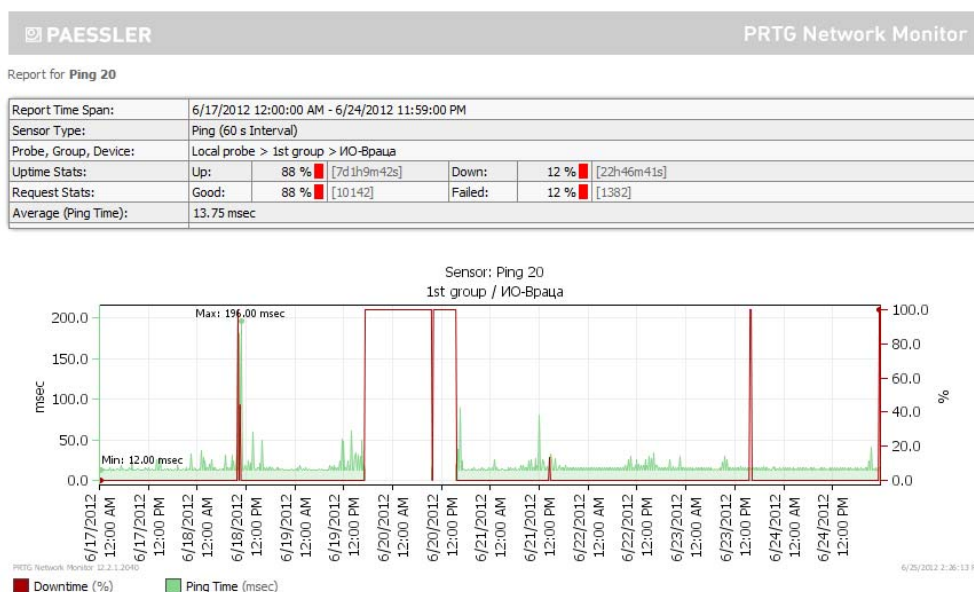
Фигура 2.21 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

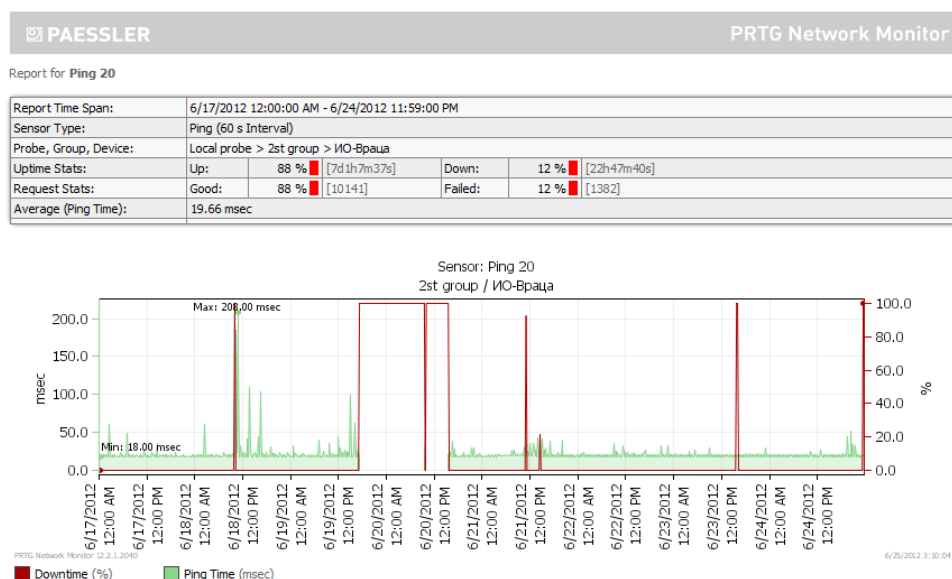
$$R = 2 \frac{100 - 1300}{13,51 - 19,27} = 416,7 \text{ bytes/ms} = 3,33 \text{ M bits/s}$$

2.3.7 Изчисляване на пропускателна способност на възел ИО-Враца

На фигура 2.22 и фигура 2.23 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.22 Статистиката на резултати при размер на пакета 100 bytes



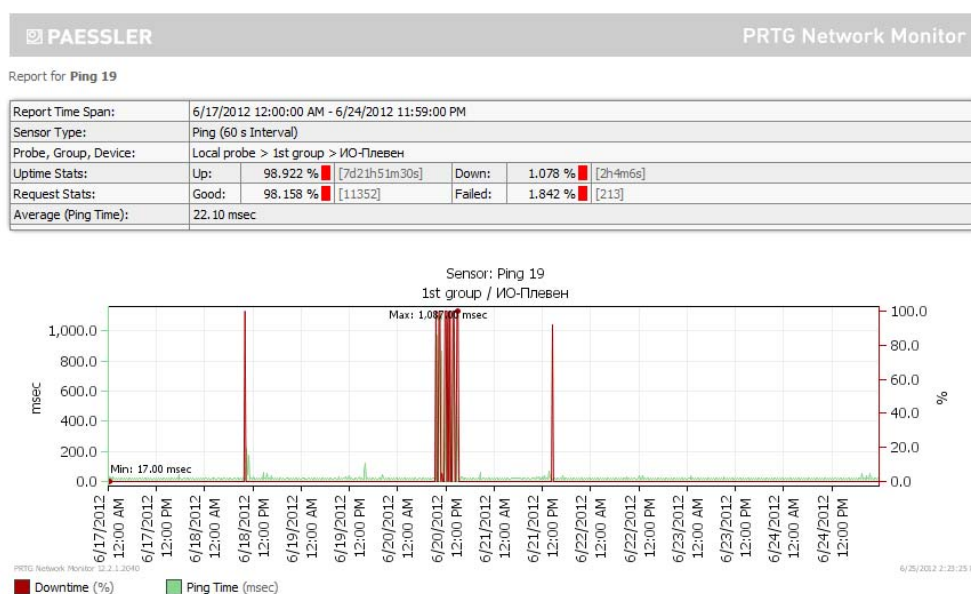
Фигура 2.23 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

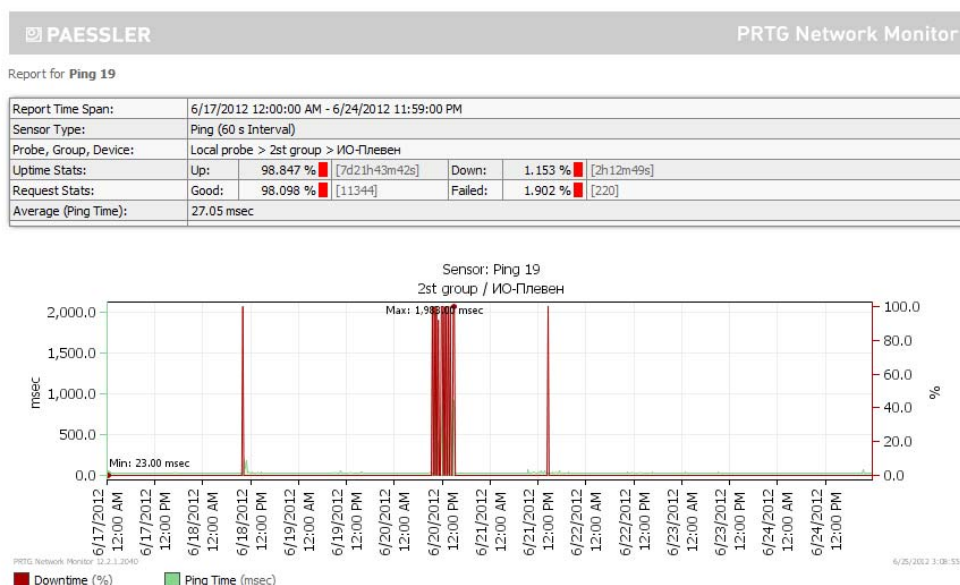
$$R = 2 \frac{100 - 1300}{13,75 - 19,66} = 406,1 \text{ bytes/ms} = 3,25 \text{ M bits/s}$$

2.3.8 Изчисляване на пропускателна способност на възел ИО-Плевен

На фигура 2.24 и фигура 2.25 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.24 Статистиката на резултати при размер на пакета 100 bytes



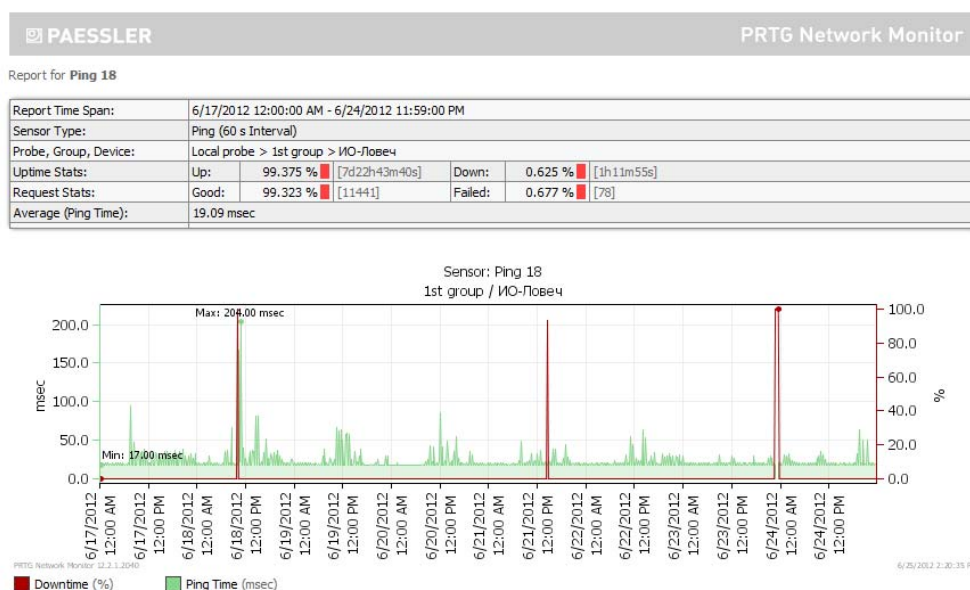
Фигура 2.25 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

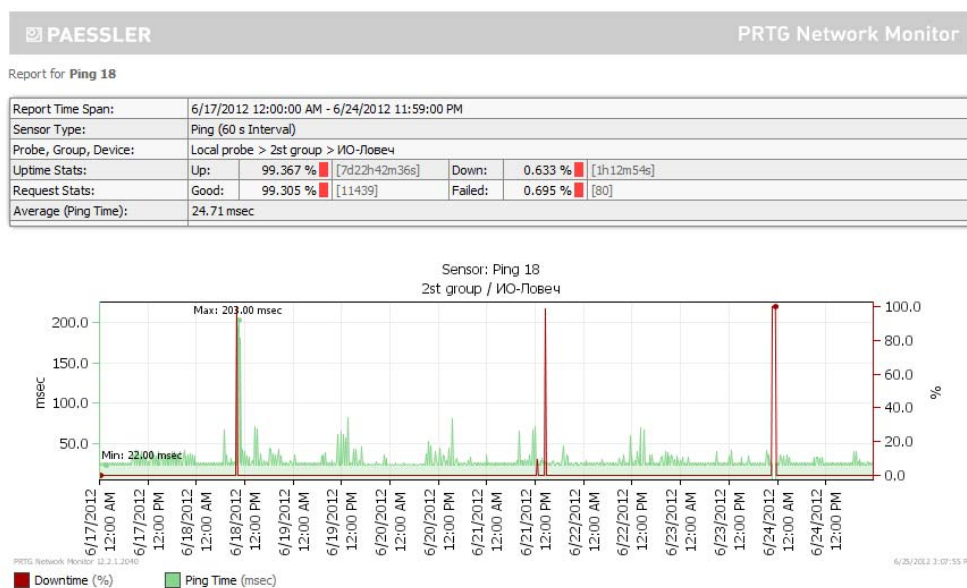
$$R = 2 \frac{100 - 1300}{22,10 - 27,05} = 484,85 \text{ bytes/ms} = 3,88 \text{ M bits/s}$$

2.3.9 Изчисляване на пропускателна способност на възел ИО-Ловеч

На фигура 2.26 и фигура 2.27 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.26 Статистиката на резултати при размер на пакета 100 bytes



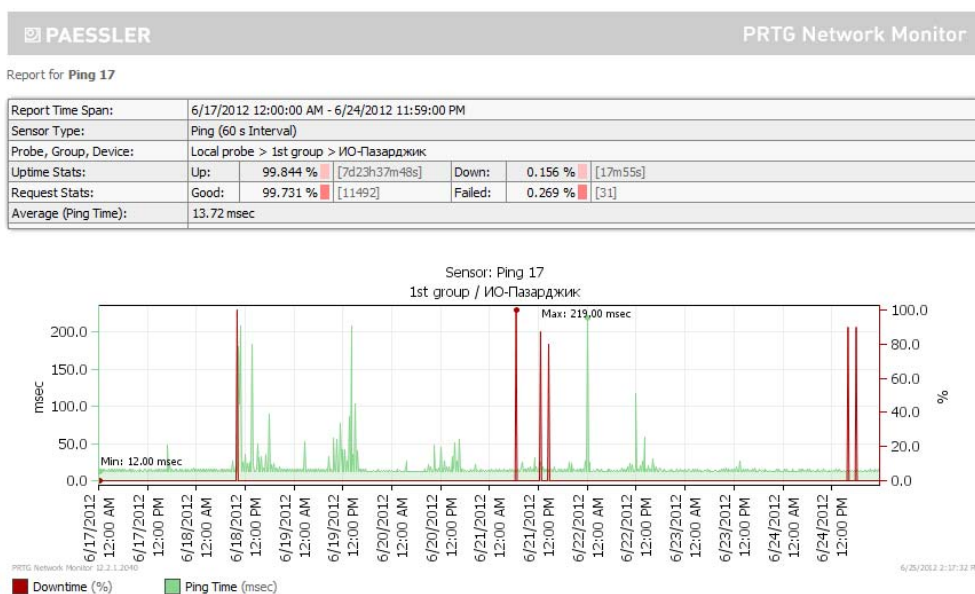
Фигура 2.27 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

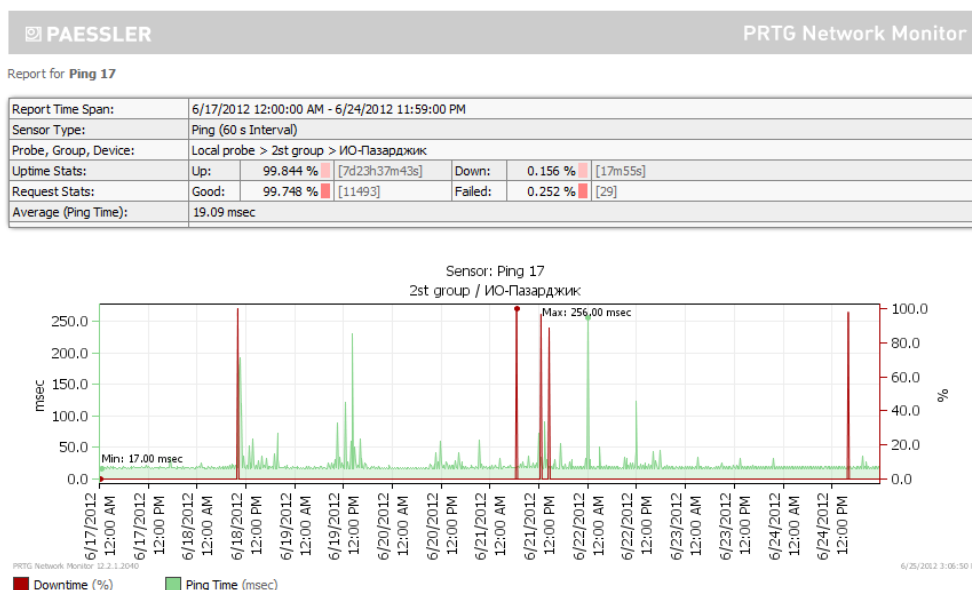
$$R = 2 \frac{100 - 1300}{19,09 - 24,71} = 427,05 \text{ bytes/ms} = 3,42 \text{ M bits/s}$$

2.3.10 Изчисляване на пропускателна способност на възел ИО-Пазарджик

На фигура 2.28 и фигура 2.29 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.28 Статистиката на резултати при размер на пакета 100 bytes



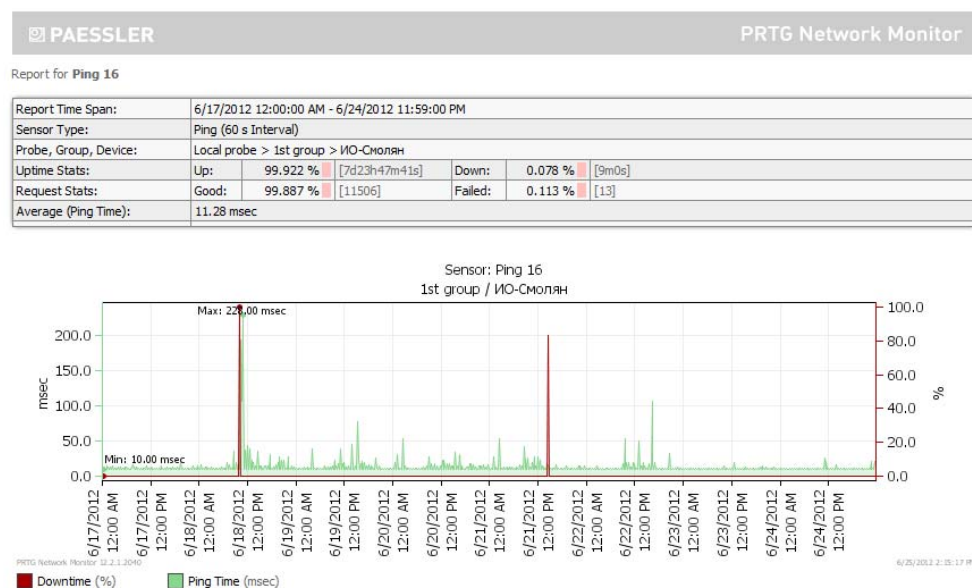
Фигура 2.29 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

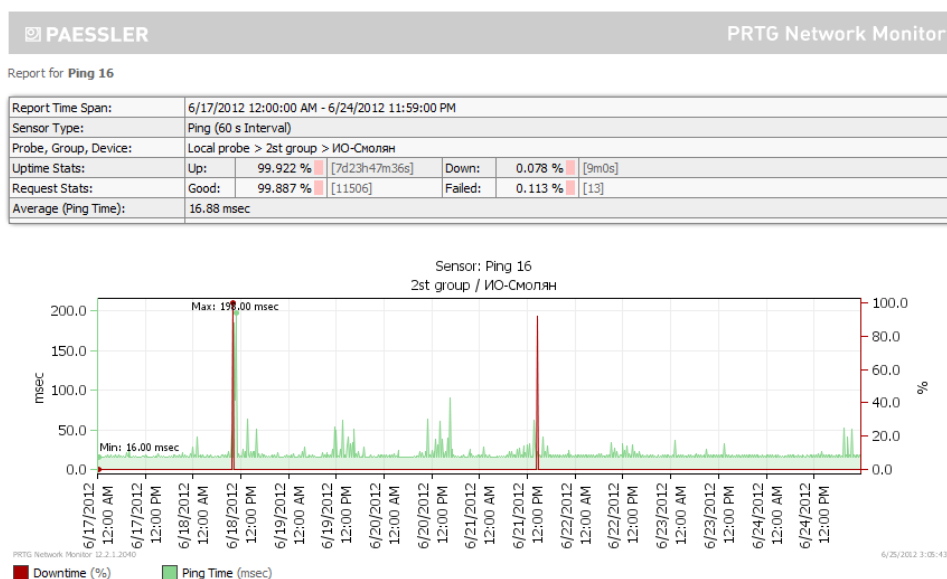
$$R = 2 \frac{100 - 1300}{13,72 - 19,09} = 446,93 \text{ bytes/ms} = 3,58 \text{ M bits/s}$$

2.3.11 Изчисляване на пропускателна способност на възел ИО-Смолян

На фигура 2.30 и фигура 2.31 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.30 Статистиката на резултати при размер на пакета 100 bytes



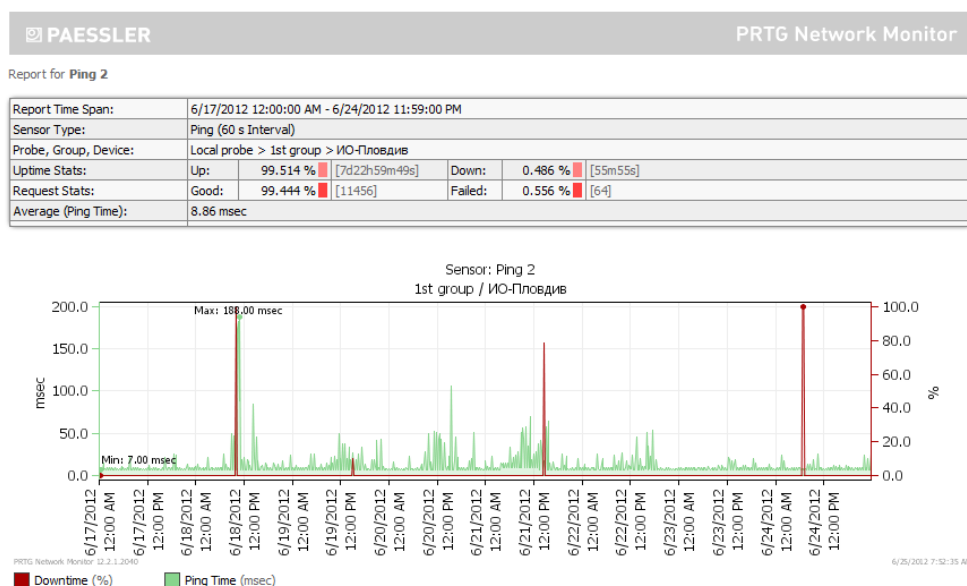
Фигура 2.31 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

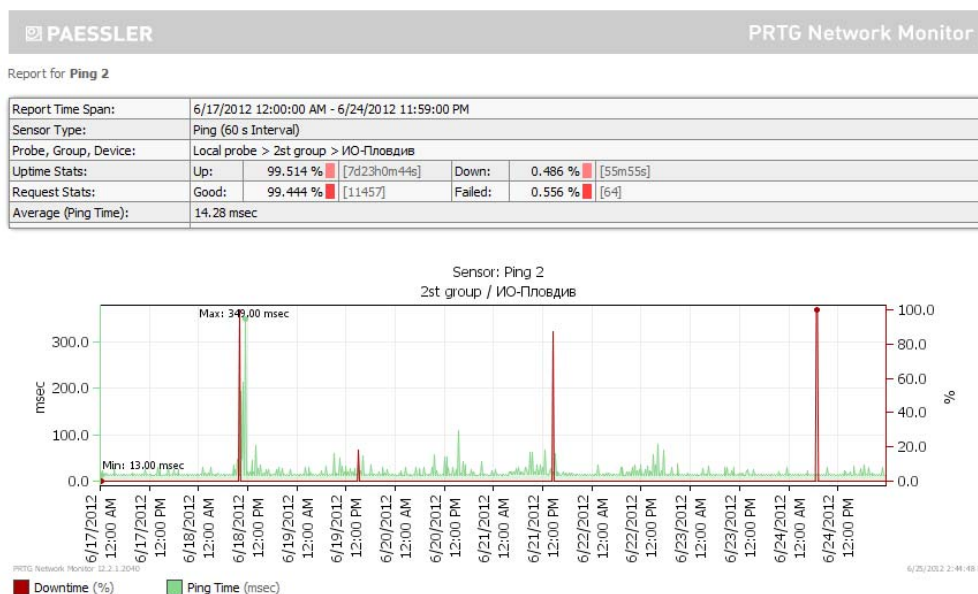
$$R = 2 \frac{100 - 1300}{11,28 - 16,88} = 428,57 \text{ bytes/ms} = 3,43 \text{ M bits/s}$$

2.3.12 Изчисляване на пропускателна способност на възел ИО-Пловдив

На фигура 2.32 и фигура 2.33 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.32 Статистиката на резултати при размер на пакета 100 bytes



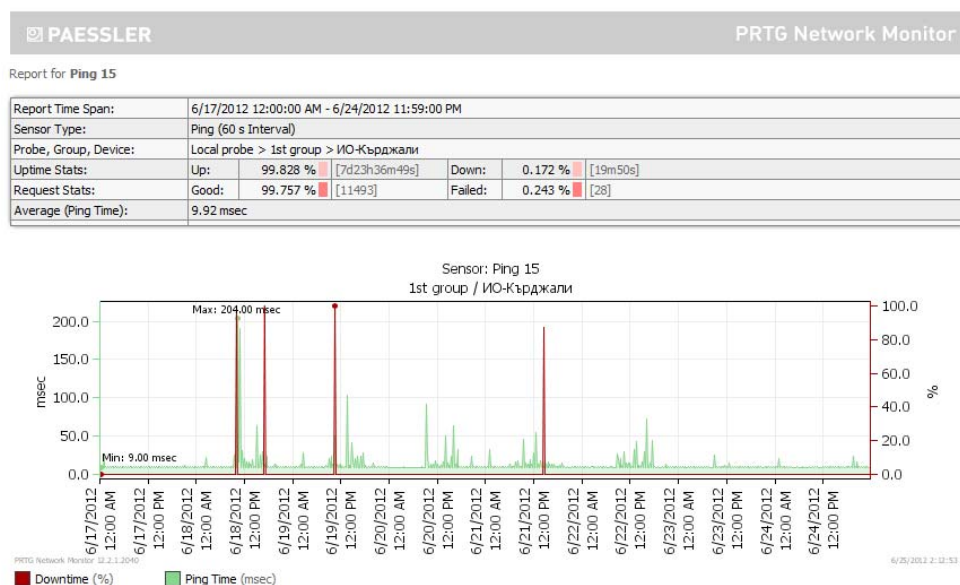
Фигура 2.33 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

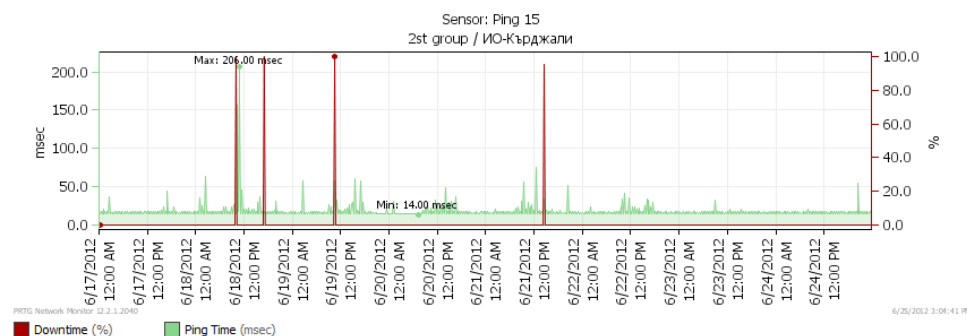
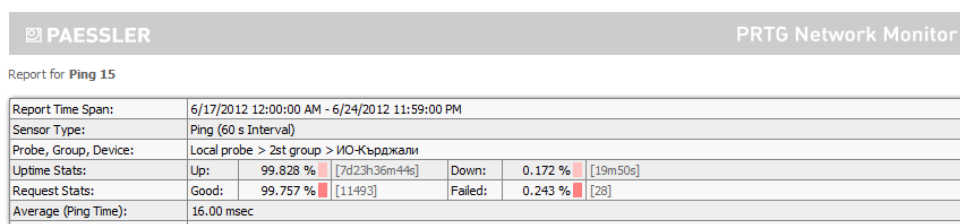
$$R = 2 \frac{100 - 1300}{8,86 - 14,28} = 442,8 \text{ bytes/ms} = 3,54 \text{ M bits/s}$$

2.3.13 Изчисляване на пропускателна способност на възел ИО-Кърджали

На фигура 2.34 и фигура 2.35 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.34 Статистиката на резултати при размер на пакета 100 bytes



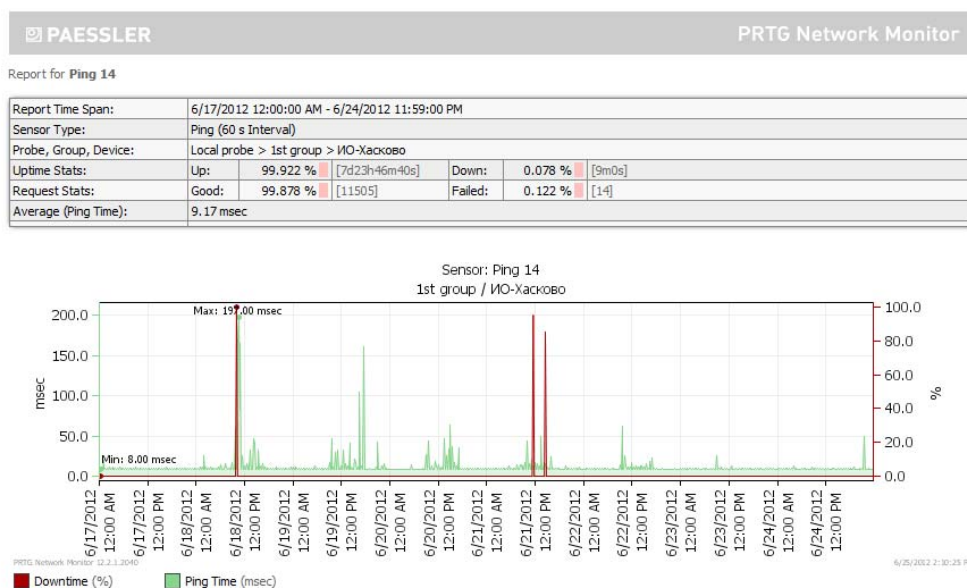
Фигура 2.35 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

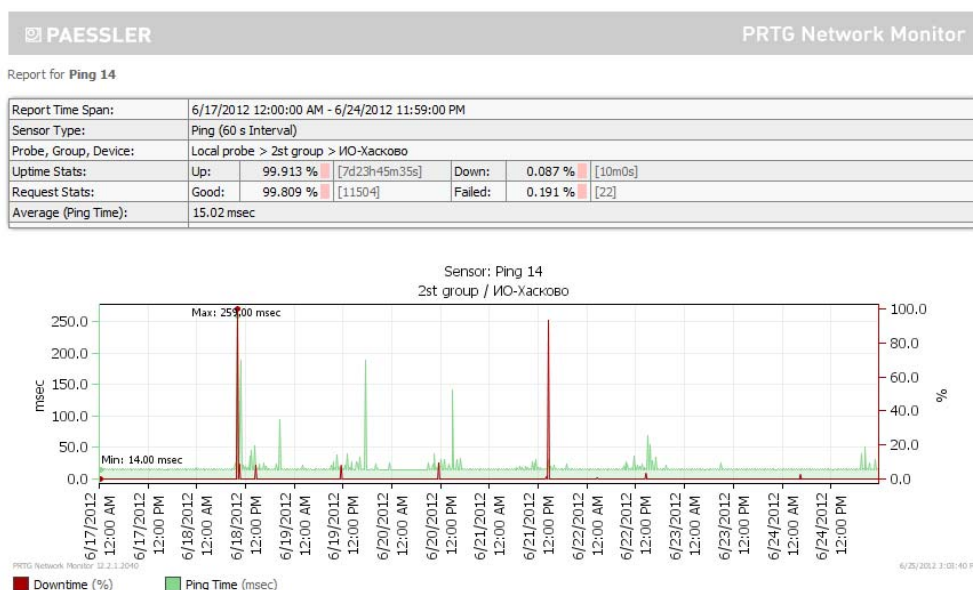
$$R = 2 \frac{100 - 1300}{9,92 - 16,00} = 394,74 \text{ bytes/ms} = 3,16 \text{ M bits/s}$$

2.3.14 Изчисляване на пропускателна способност на възел ИО-Хасково

На фигура 2.36 и фигура 2.37 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.36 Статистиката на резултати при размер на пакета 100 bytes



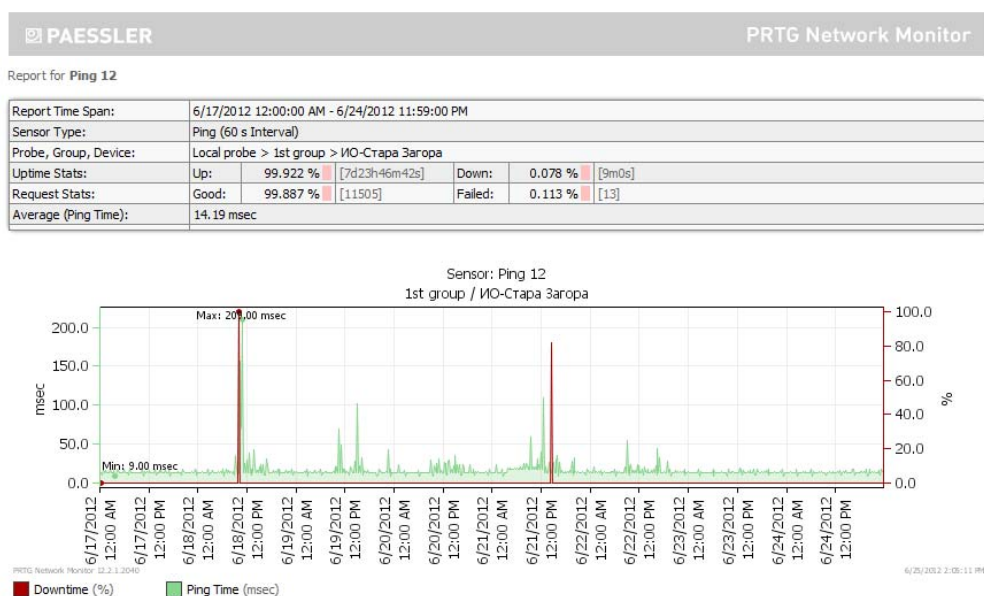
Фигура 2.37 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

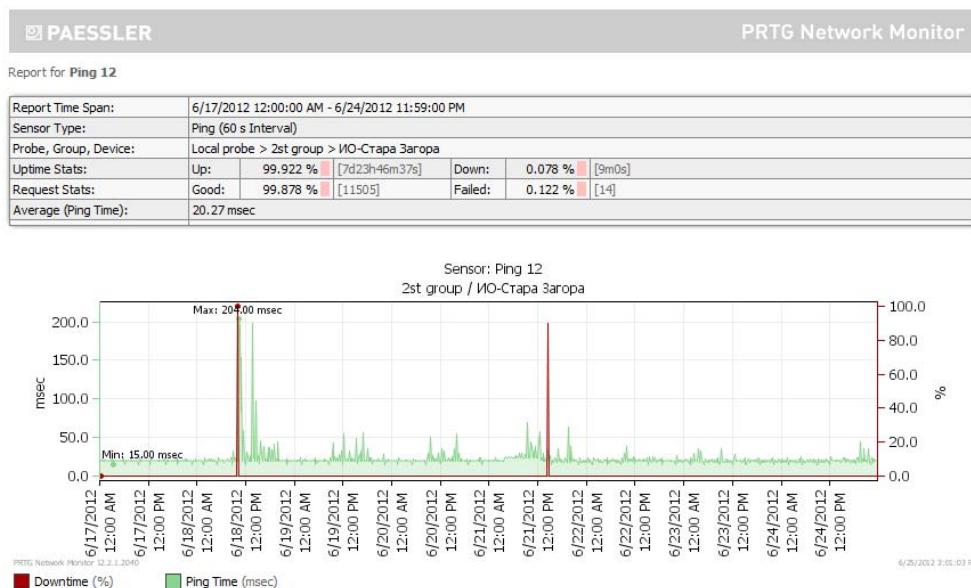
$$R = 2 \frac{100 - 1300}{9,17 - 15,02} = 410,26 \text{ bytes/ms} = 3,28 \text{ M bits/s}$$

2.3.15 Изчисляване на пропускателна способност на възел ИО-Стара Загора

На фигура 2.38 и фигура 2.39 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.38 Статистиката на резултати при размер на пакета 100 bytes



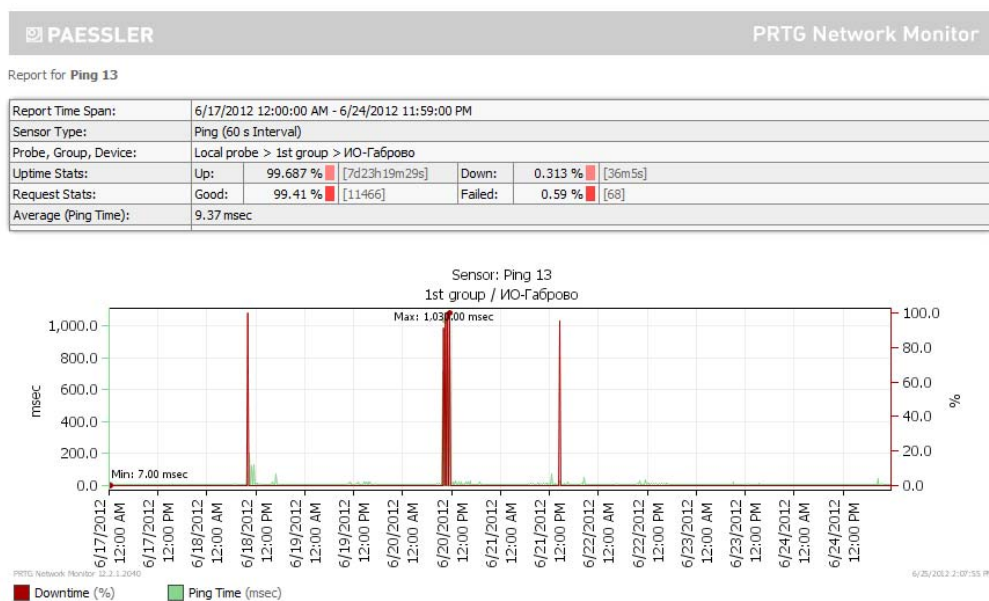
Фигура 2.39 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

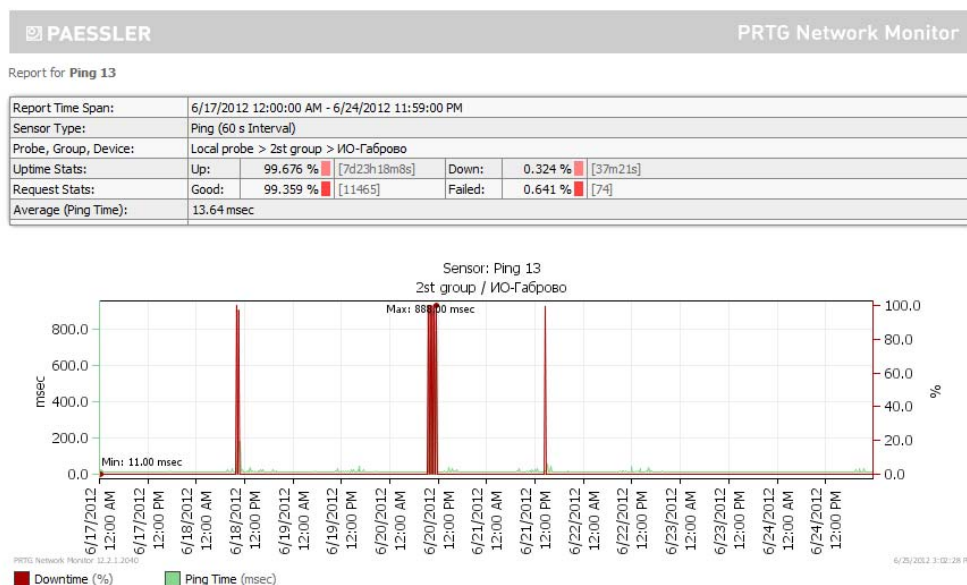
$$R = 2 \frac{100 - 1300}{14,19 - 20,27} = 394,74 \text{ bytes/ms} = 3,16 \text{ M bits/s}$$

2.3.16 Изчисляване на пропускателна способност на възел ИО-Габрово

На фигура 2.40 и фигура 2.41 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.40 Статистиката на резултати при размер на пакета 100 bytes



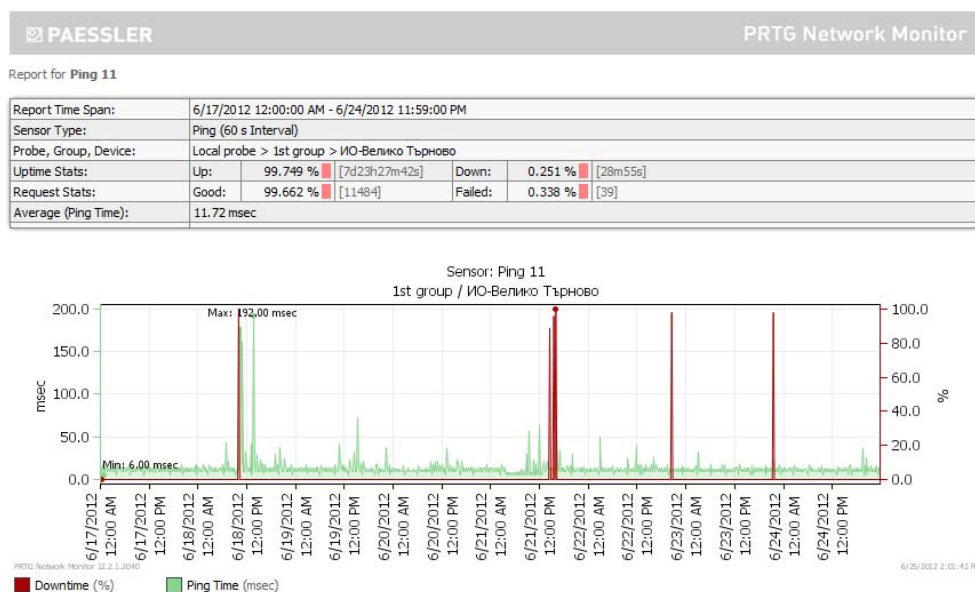
Фигура 2.41 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

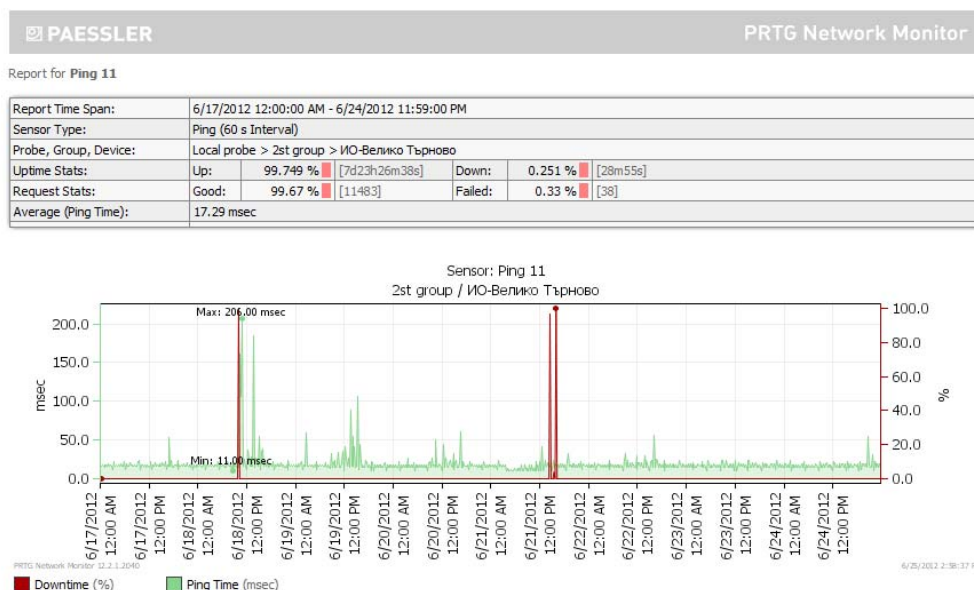
$$R = 2 \frac{100 - 1300}{9,37 - 13,64} = 562,06 \text{ bytes/ms} = 4,97 \text{ M bits/s}$$

2.3.17 Изчисляване на пропускателна способност на възел ИО-Велико Търново

На фигура 2.42 и фигура 2.43 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.42 Статистиката на резултати при размер на пакета 100 bytes



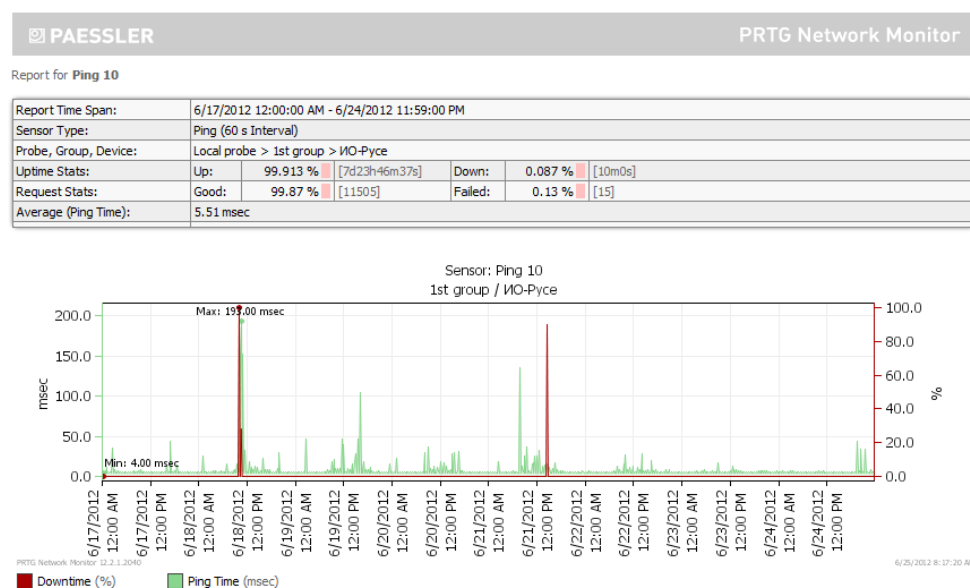
Фигура 2.43 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

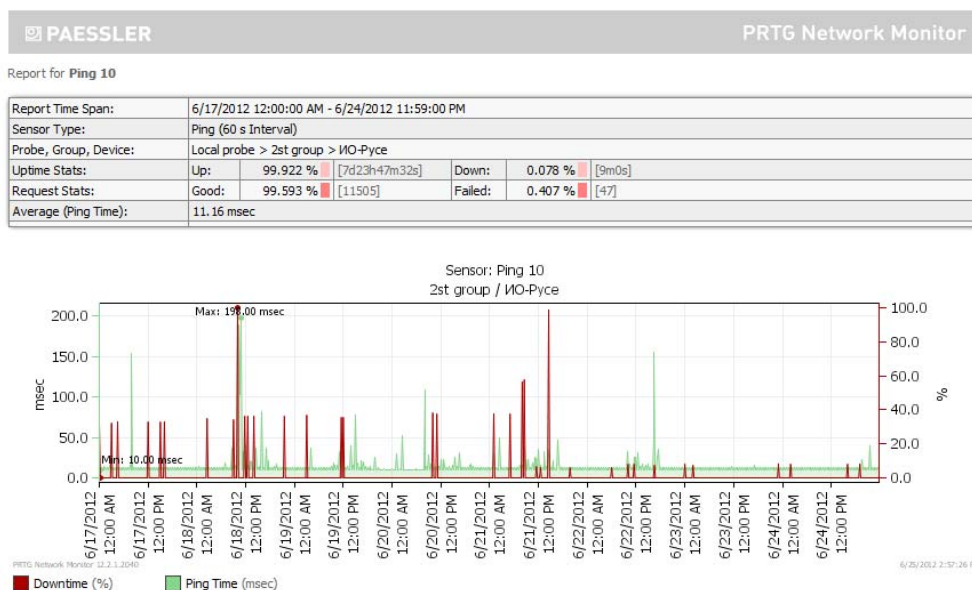
$$R = 2 \frac{100 - 1300}{11,72 - 17,29} = 430,88 \text{ bytes/ms} = 3,45 \text{ M bits/s}$$

2.3.18 Изчисляване на пропускателна способност на възел ИО-Русе

На фигура 2.44 и фигура 2.45 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.44 Статистиката на резултати при размер на пакета 100 bytes



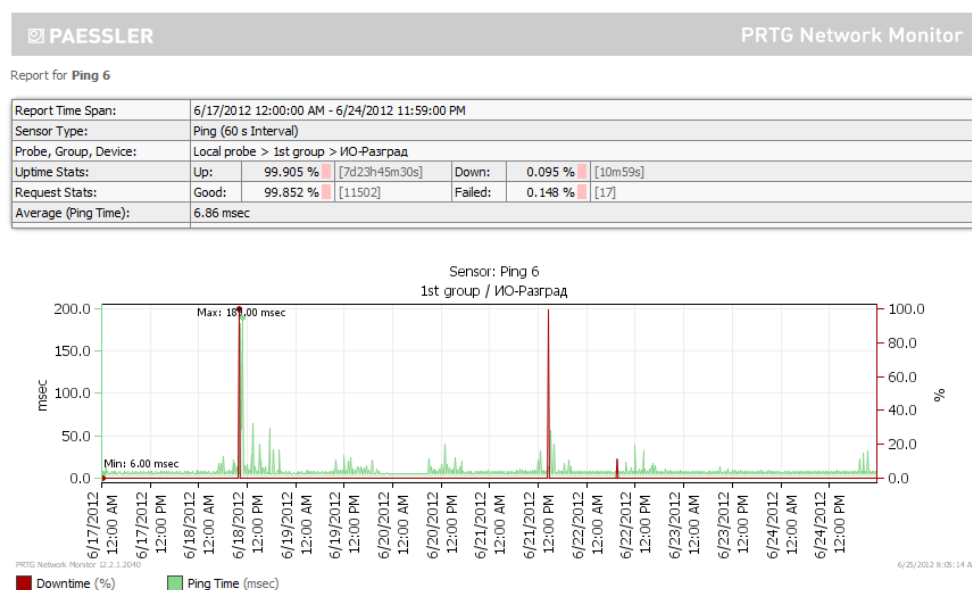
Фигура 2.45 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

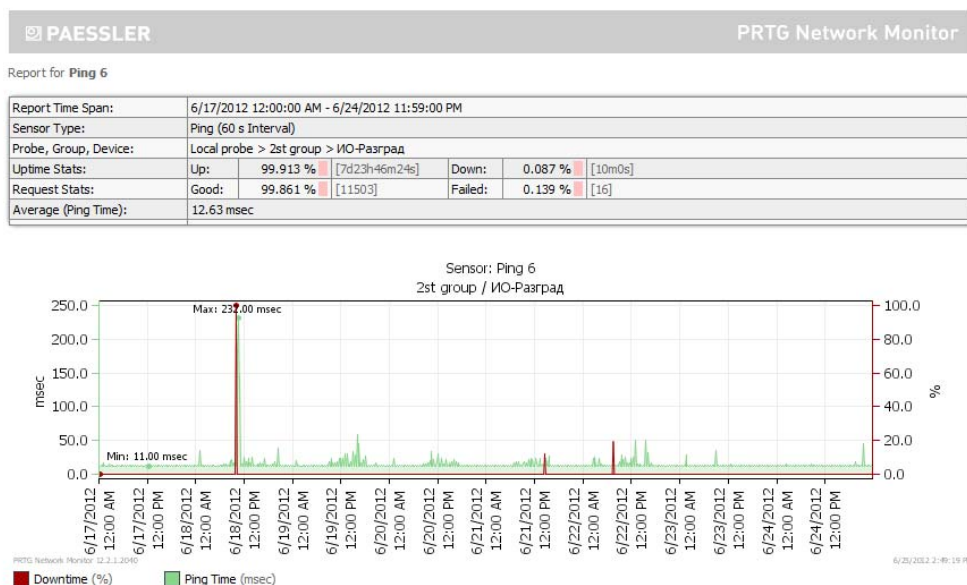
$$R = 2 \frac{100 - 1300}{5,51 - 11,16} = 424,78 \text{ bytes/ms} = 3,4 \text{ M bits/s}$$

2.3.19 Изчисляване на пропускателна способност на възел ИО-Разград

На фигура 2.46 и фигура 2.47 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.46 Статистиката на резултати при размер на пакета 100 bytes



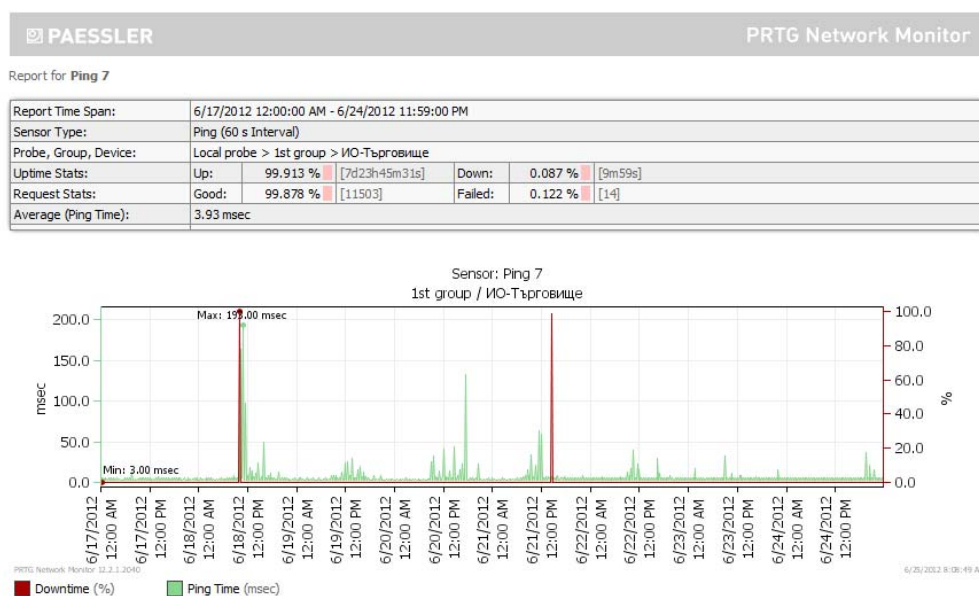
Фигура 2.47 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

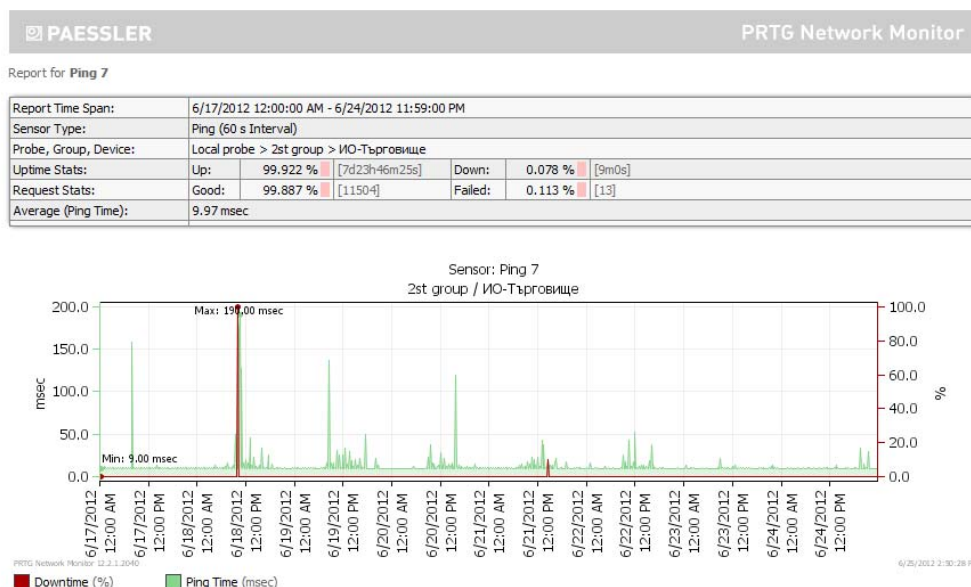
$$R = 2 \frac{100 - 1300}{6,86 - 12,63} = 415,94 \text{ bytes/ms} = 3,33 \text{ M bits/s}$$

2.3.20 Изчисляване на пропускателна способност на възел ИО-Търговище

На фигура 2.48 и фигура 2.49 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.48 Статистиката на резултати при размер на пакета 100 bytes



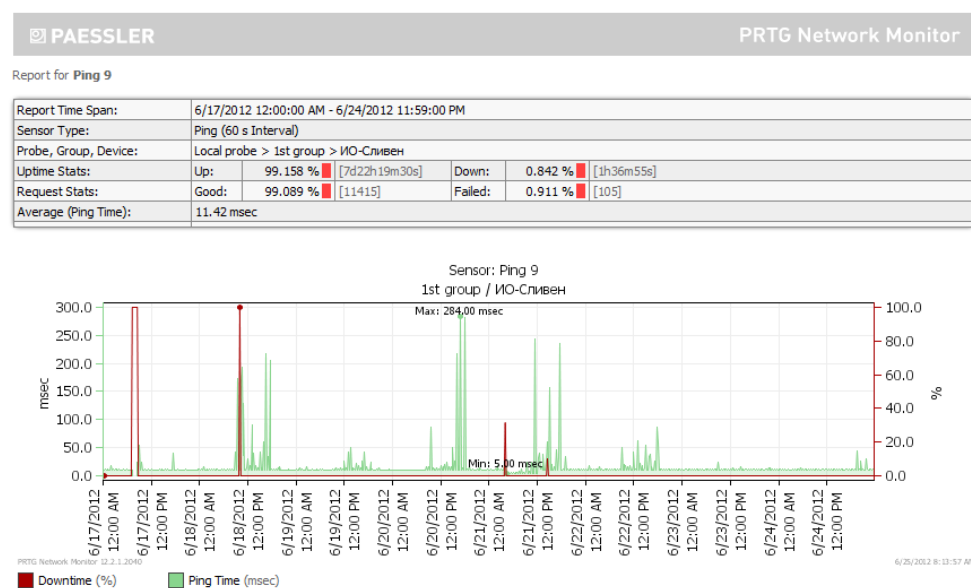
Фигура 2.49 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

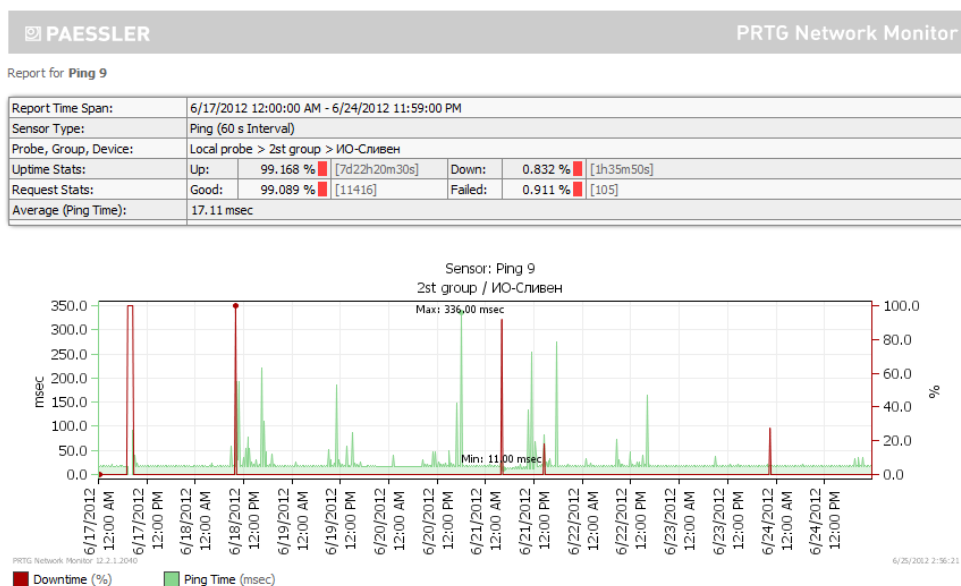
$$R = 2 \frac{100 - 1300}{3,93 - 9,97} = 397,35 \text{ bytes/ms} = 3,18 \text{ M bits/s}$$

2.3.21 Изчисляване на пропускателна способност на възел ИО-Сливен

На фигура 2.50 и фигура 2.51 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.50 Статистиката на резултати при размер на пакета 100 bytes



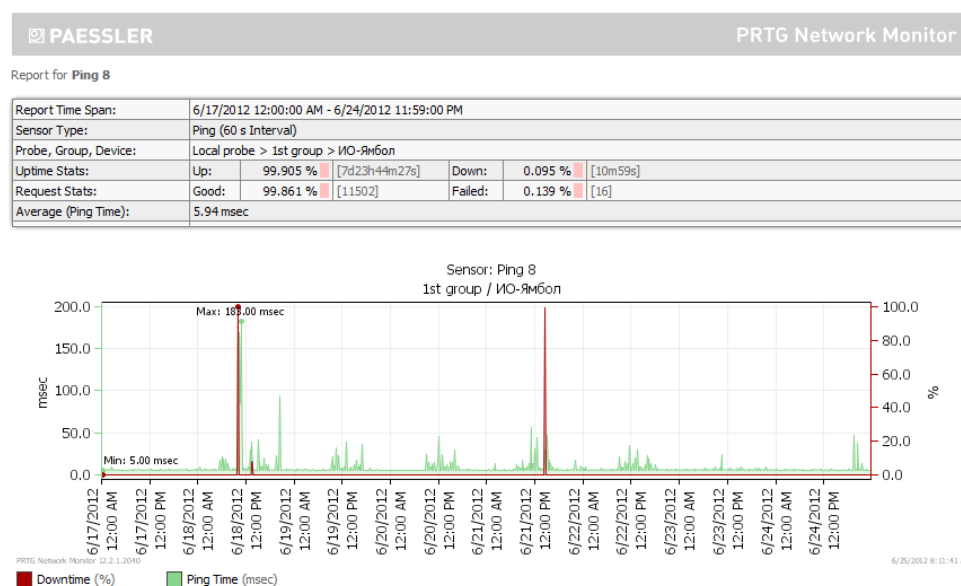
Фигура 2.51 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

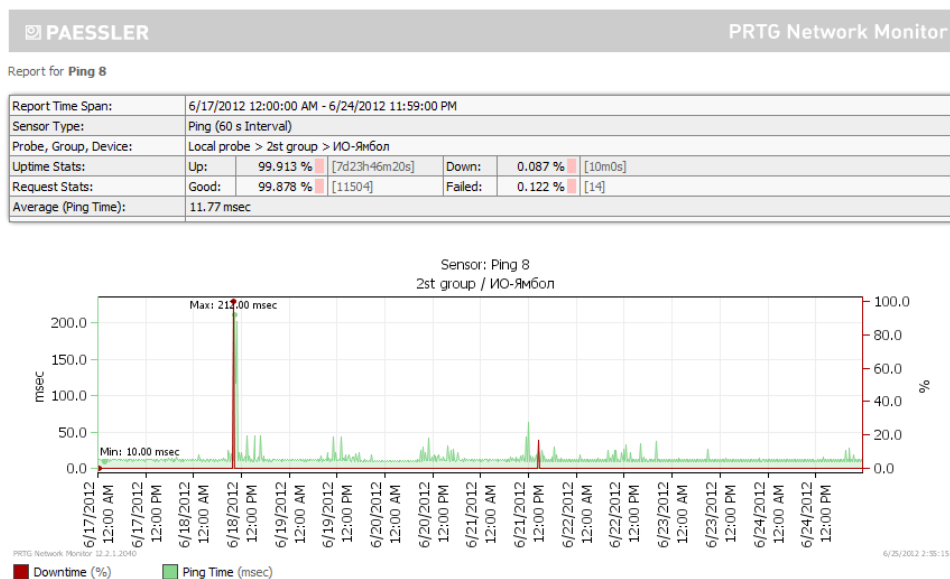
$$R = 2 \frac{100 - 1300}{11,42 - 17,11} = 421,79 \text{ bytes/ms} = 3,37 \text{ M bits/s}$$

2.3.22 Изчисляване на пропускателна способност на възел ИО-Ямбол

На фигура 2.52 и фигура 2.53 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.52 Статистиката на резултати при размер на пакета 100 bytes



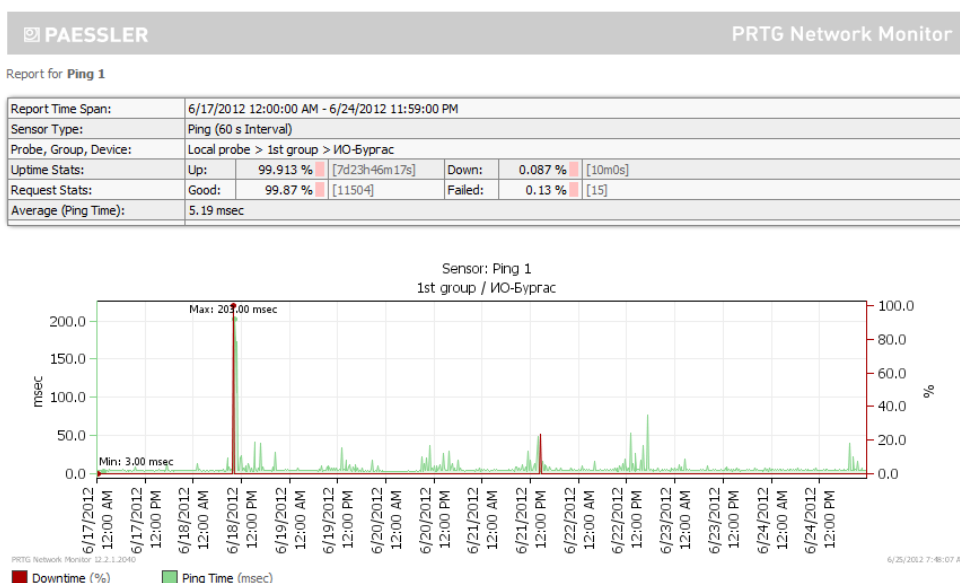
Фигура 2.53 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

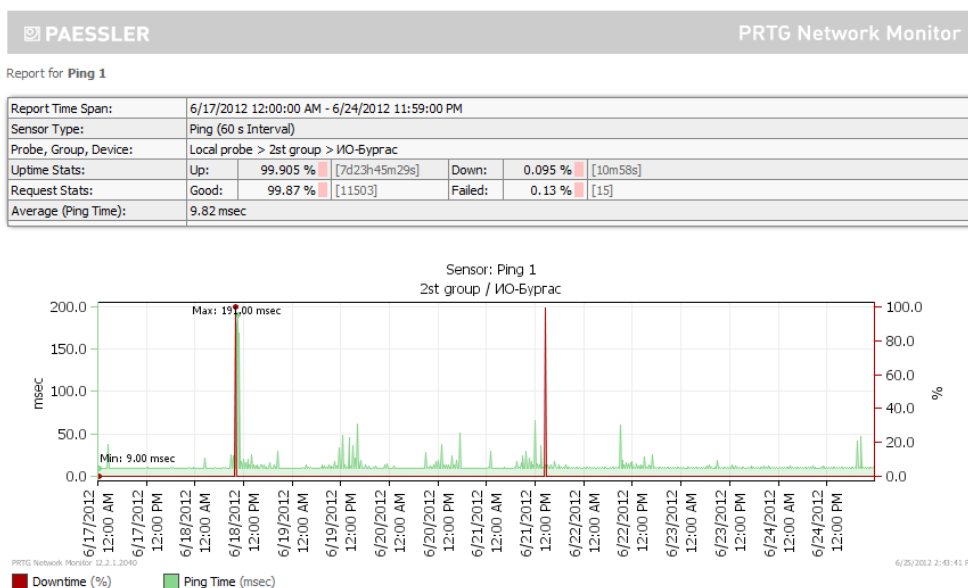
$$R = 2 \frac{100 - 1300}{5,94 - 11,77} = 411,66 \text{ bytes/ms} = 3,29 \text{ M bits/s}$$

2.3.23 Изчисляване на пропускателна способност на възел ИО-Бургас

На фигура 2.54 и фигура 2.55 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.54 Статистиката на резултати при размер на пакета 100 bytes



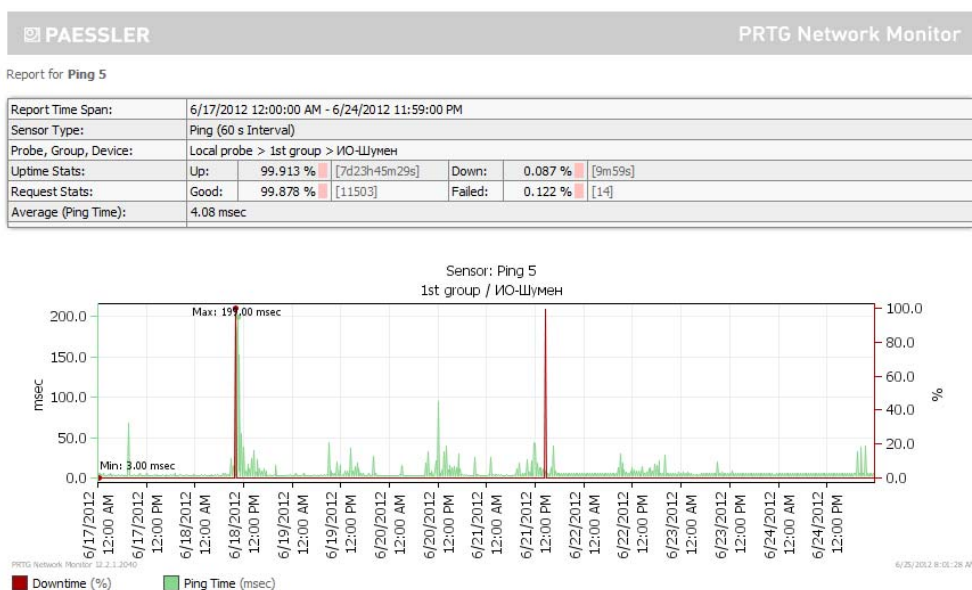
Фигура 2.55 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

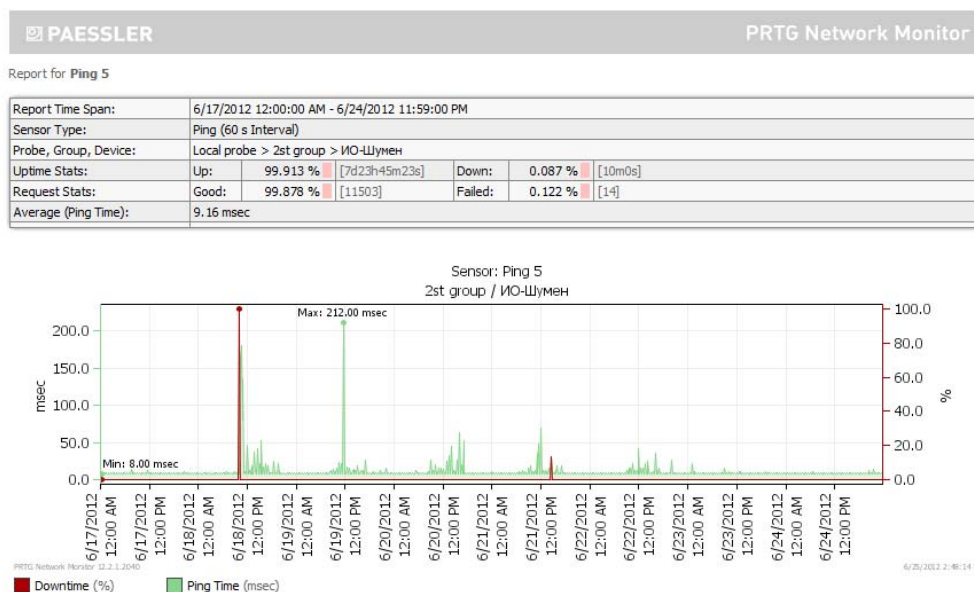
$$R = 2 \frac{100 - 1300}{5,19 - 9,82} = 518,36 \text{ bytes/ms} = 4,15 \text{ M bits/s}$$

2.3.24 Изчисляване на пропускателна способност на възел ИО-Шумен

На фигура 2.56 и фигура 2.57 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.56 Статистиката на резултати при размер на пакета 100 bytes



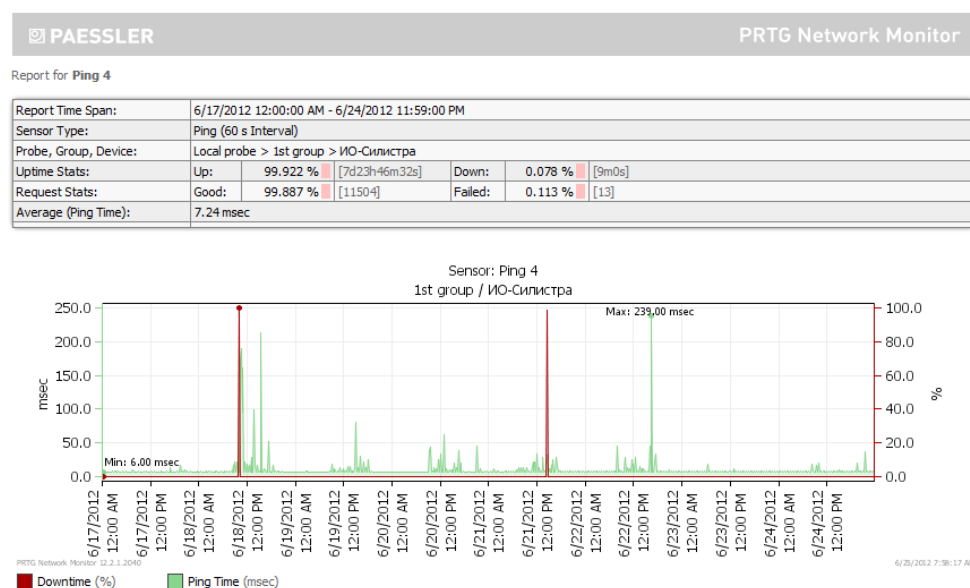
Фигура 2.57 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

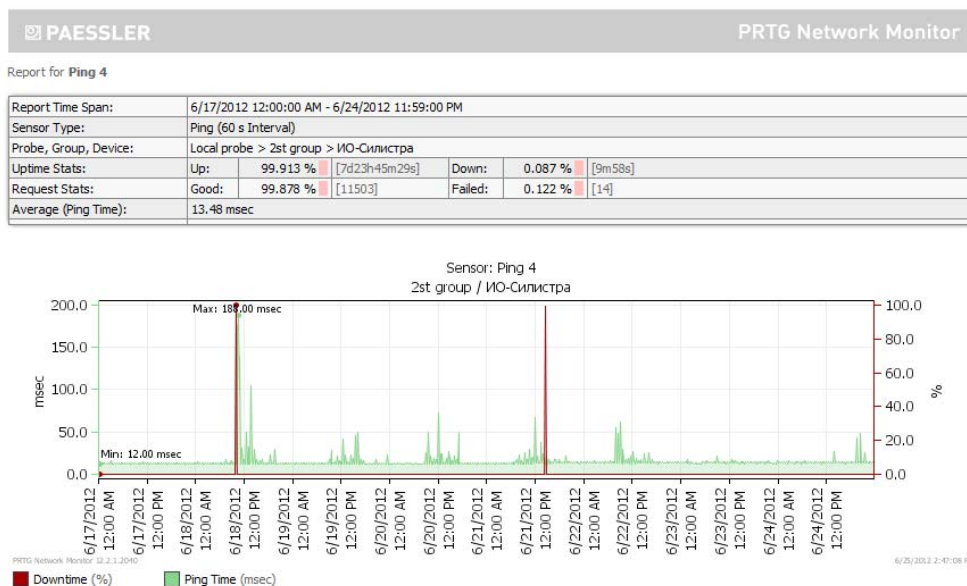
$$R = 2 \frac{100 - 1300}{4,08 - 9,16} = 472,44 \text{ bytes/ms} = 3,78 \text{ M bits/s}$$

2.3.25 Изчисляване на пропускателна способност на възел ИО-Силистра

На фигура 2.58 и фигура 2.59 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.58 Статистиката на резултати при размер на пакета 100 bytes



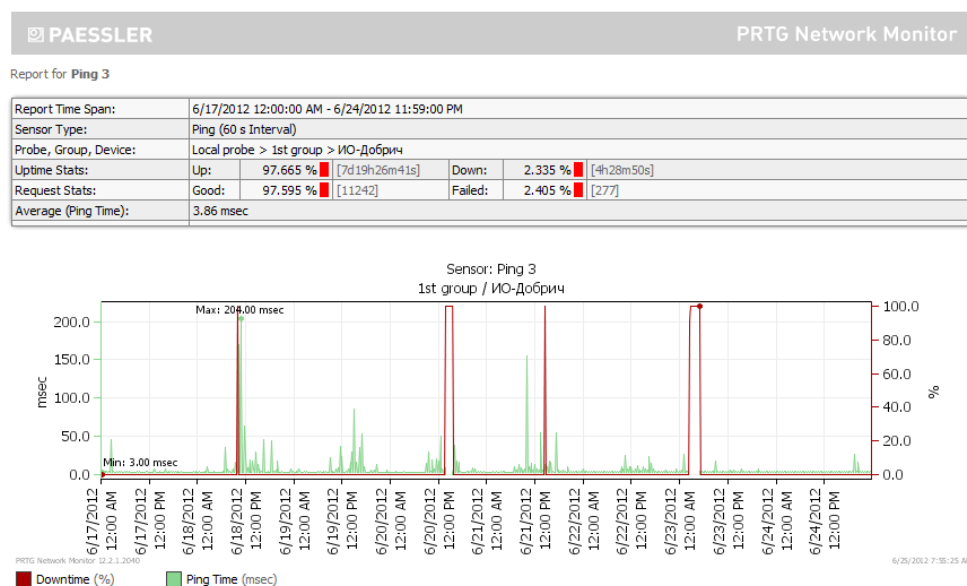
Фигура 2.59 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

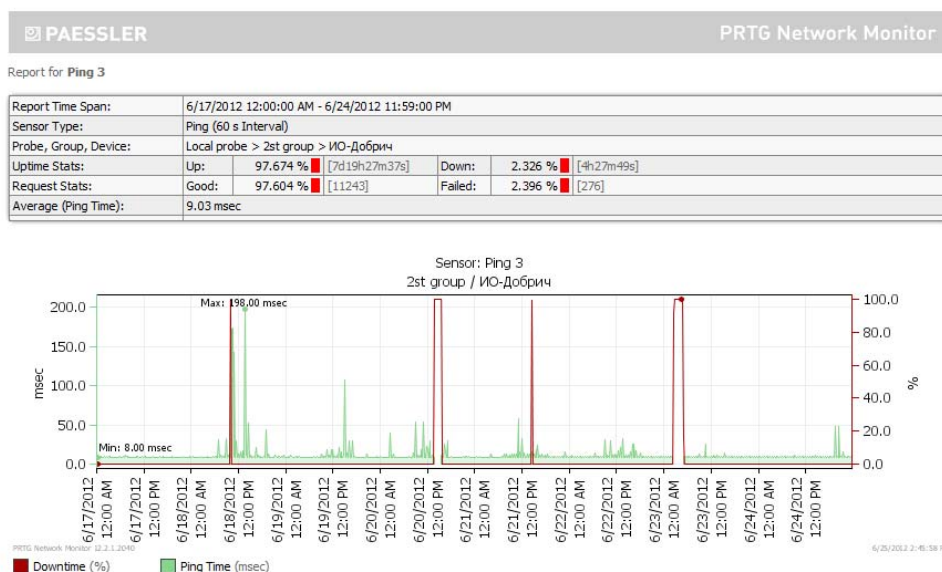
$$R = 2 \frac{100 - 1300}{7,24 - 13,48} = 384,62 \text{ bytes/ms} = 3,08 \text{ M bits/s}$$

2.3.26 Изчисляване на пропускателна способност на възел ИО-Добрич

На фигура 2.60 и фигура 2.61 е показана информация за възела при изследване на времезакъснението за период от една седмица и е изобразена статистиката на получените резултати при размер на пакетите 100 bytes и 1300 bytes.



Фигура 2.60 Статистиката на резултати при размер на пакета 100 bytes



Фигура 2.61 Статистиката на резултати при размер на пакета 1300 bytes

На база получените резултати от изследването, за пропускателната способност на връзката до възела получаваме:

$$R = 2 \frac{100 - 1300}{3,86 - 9,03} = 464,22 \text{ bytes/ms} = 3,71 \text{ M bits/s}$$

В таблица 2.2 са показани обобщени резултати от направените измервания.

Таблица 2.2

Възел / клон	Град	100 bytes		1300 bytes		R - пропускателна способност, M bits/s	Up, %	Down, %
1	София	0	ms	0	ms	0	100	0
2	Перник	12,10	ms	17,74	ms	3,4	99,922	0,078
3	Кюстендил	13,71	ms	19,38	ms	3,38	99,922	0,078
4	Благоевград	14,36	ms	19,97	ms	3,42	99,913	0,087
5	Видин	14,61	ms	20,58	ms	3,22	99,905	0,095
6	Монтана	13,51	ms	19,27	ms	3,33	99,922	0,078
7	Враца	13,75	ms	19,66	ms	3,25	88	12
8	Плевен	22,10	ms	27,05	ms	3,88	98,922	1,078
9	Ловеч	19,09	ms	24,71	ms	3,42	99,375	0,625
10	Пазарджик	13,72	ms	19,09	ms	3,58	99,844	0,156
11	Смолян	11,28	ms	16,88	ms	3,43	99,922	0,078
12	Пловдив	8,86	ms	14,28	ms	3,54	99,514	0,486
13	Кърджали	9,92	ms	16,00	ms	3,16	99,828	0,172
14	Хасково	9,17	ms	15,02	ms	3,28	99,922	0,078
15	Стара Загора	14,19	ms	20,27	ms	3,16	99,922	0,078
16	Габрово	9,37	ms	13,64	ms	4,97	99,687	0,313
17	Велико Търново	11,72	ms	17,29	ms	3,45	99,749	0,251
18	Русе	5,51	ms	11,16	ms	3,4	99,913	0,087
19	Разград	6,86	ms	12,63	ms	3,33	99,905	0,095
20	Търговище	3,93	ms	9,97	ms	3,18	99,913	0,087
21	Сливен	11,42	ms	17,11	ms	3,37	99,158	0,842
22	Ямбол	5,94	ms	11,77	ms	3,29	99,905	0,095
23	Бургас	5,19	ms	9,82	ms	4,15	99,913	0,087
24	Шумен	4,08	ms	9,16	ms	3,78	99,913	0,087
25	Силистра	7,24	ms	13,48	ms	3,08	99,922	0,078
26	Добрич	3,86	ms	9,03	ms	3,71	97,665	2,335
27	Варна	1,98	ms	5,68	ms	5,19	100	0

От получените резултати може да се обобщи, че така съществуващата комуникационна мрежа с топология звезда предизвиква сризове и пропадания в информационната комуникация между отделни възли, което от своя страна води до проблеми с достъпа до клоновата мрежа и нормалната свързаност. Липсата на свързаност между отделните възли на мрежата варира от 10 мин. до 23 часа, което е недопустимо. Изчислените пропускателни способности показват приблизително еднаквите капацитети на мрежата между отделните крайни точки.

2.4 Изводи

В тази глава е определена и анализирана пропускателната способност и времезакъсненията в мрежата при предаване на информационни потоци. Определени са средните времеви закъснения между отделните възли при съществуващата топология. Изчислен е капацитетът на информационната мрежа, като изчислението е извършено между възел ИО-София и останалите възли. Оценено е пропадането на връзката между двойка възли като характеристика на отказоустойчивостта на съществуваща комуникационна мрежа със свързаност тип „звезда“.

Глава 3. Проектиране на двусвързана и трисвързана мрежа при отчитане на съществуваща топология тип звезда

3.1 Увод

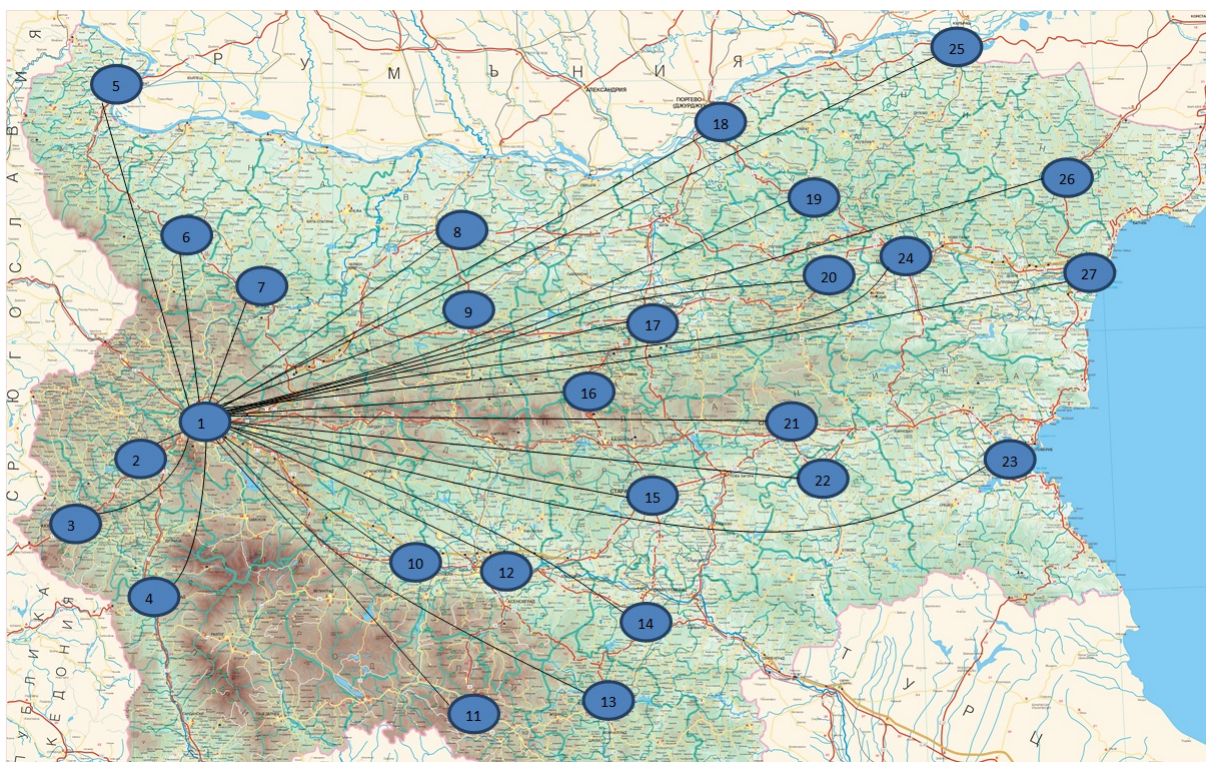
Целта на тази глава е да се синтезира алгоритъм за определяне на теоретичните времезакъснения между възлите на добавените алтернативни комуникационни канали, да се дефинира оптимизационна задача за определяне на свързаността на мрежата като се използват резултатите от измерванията на времезакъсненията и пропаданията в комуникационната мрежа, да се реши оптимизационна задача за определяне на структура на комуникационната мрежа като двусвързана и трисвързана.

3.2 Определяне модела на комуникационна мрежа за получаване на трисвързаност между отделните точки

Времето на реакция в компютърните мрежи се влияе от закъсненията на компютърните системи, от маршрутизиращите алгоритми, от закъснението на сигналите по линиите, от алгоритмите за достъп на заявките до компютърната мрежа, от топологията на мрежата. Ограничаване на времезакъсненията, оптимизиране на съотношенията на различните фактори, предизвикващи закъснения, позволява да се повиши качеството на обслужване [109].

Изграждането на техническа хетерогенна WAN мрежа налага да се моделират и експериментират процедури на управление и оптимизация на трафика. Тези изследвания ще позволяват оптимизирането на информационните потоци в компютърни комуникационни мрежи да се свърже с приложения, които изискват гарантирано качество на обслужване [110, 111].

На фигура 3.1 е представена реална информационна мрежа с точно разположение на отделните точки, чиято свързаност е тип „звезда“. Поради централизираното си управление (целият информационен поток да преминава през една възлова точка), тази свързаност допуска при сригове във връзката между отделните точки да се получат загуби при работа и откъснатост между отделните възли от комуникационната мрежа. Разглежданата мрежа се състои от 27 възела, които представляват клонове в различни градове, предоставящи информационни услуги.



Фигура 3.1 Реална информационна мрежа с 27 възела и свързаност тип „звезда“

При тази топология имаме връзка от всеки отдалечен възел към главния. Основното преимущество на този модел е, че добавянето на нови възли към мрежата е лесно, защото цялото конфигуриране се извършва в централния възел. Неудобствата които възникват са следните:

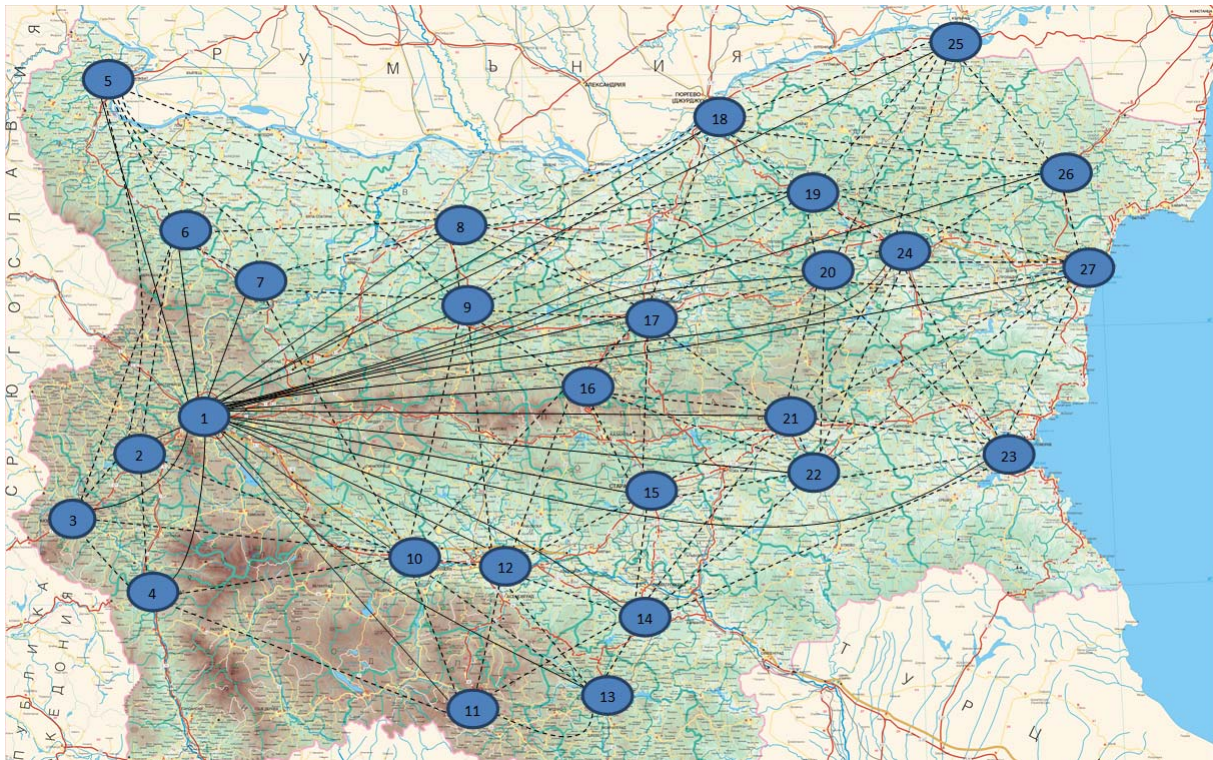
- Има единствена точка на прекъсване. Ако устройството в централният възел откаже, всичките останали клонове спират да комуникират.
- Ако в устройството в центъра се получи временен срыв и изпитва затруднения, това се усеща за целият информационен поток.
- Два клона, които са географски близо, им се налага при комуникация помежду си и да прекарват трафика през главния възел.

На фигура 3.2 е представено решение за подобряване свързаността между отделните клонове като се използва многосвързана топология. На тази база се подобрява обменът на информационни потоци и се елиминира възможността при отпадане на централният комуникационен възел да се достигне до блокиране работата на всички крайни точки. По този начин се получава и взаимозаменяемост на комуникационните канали и отпада възможността за прекъсване на информационните потоци. Представени са добавените виртуални канали. При така дефинираната нова свързаност, теоритично се изчисляват времезакъсненията между отделните директно свързани възли на виртуалната мрежа, като целта е измежду всички възможни алтернативни пътища, заедно със съществуващите такива, да се изберат най-добрите с оглед получаване на двусвързана и трисвързана топология на разглежданата информационна мрежа. Предимствата на представената топология са следните:

- Няма единствена точка на прекъсване. Възлите в мрежата не разчитат на централният клон за осигуряване на свързаност. При възникване на проблем в мрежата, спира да работи и само част от нея, но не и цялата.
- Общата производителност не зависи само от производителността на една система.
- Географски близките възли може да установяват връзка непосредствено помежду си.

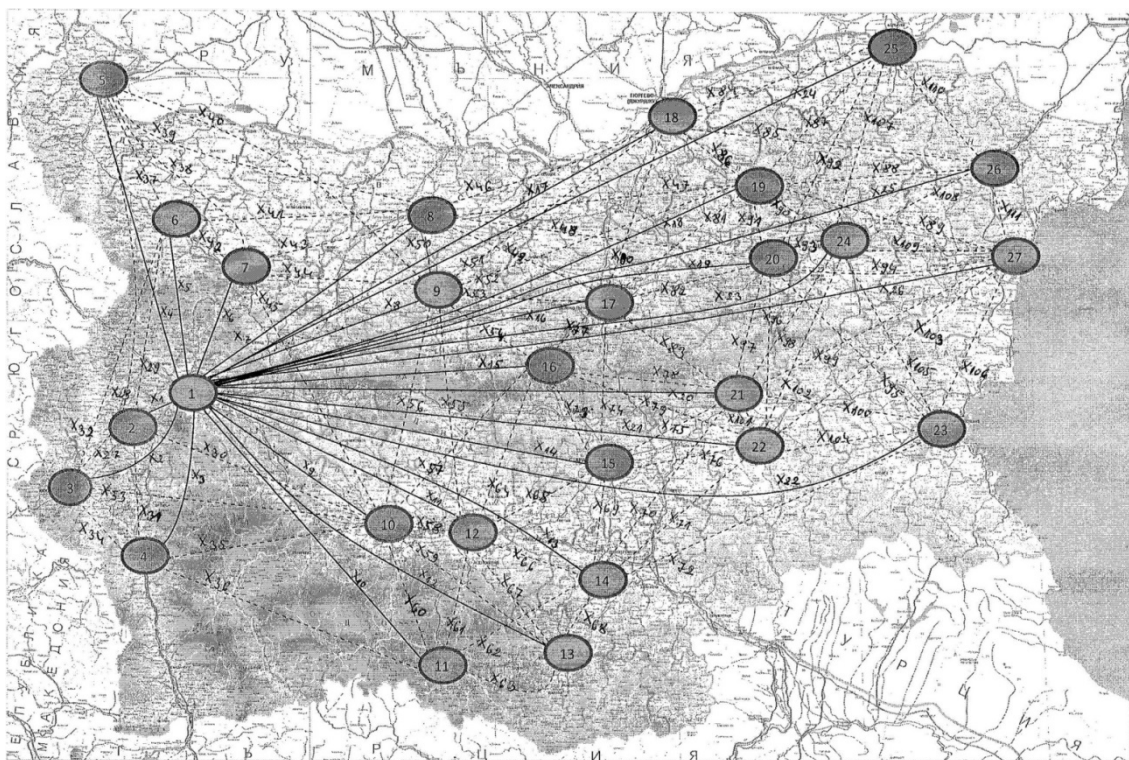
Като недостатък може да се спомене увеличеното обслужване. Когато добавяме нов възел, ние трябва да го конфигурираме за връзки с всички останали, както и да конфигурираме останалите възли за връзка с него.

Допълнителните връзки между възлите са дефинирани на географски принцип, като е определено условие, всяка възлова точка да има минимум четири допълнителни пътя, в които не се включват тези от съществуващата мрежа.



Фигура 3.2 Реална информационна мрежа с 27 възела и добавени виртуални алтернативни канали

На фигура 3.3 е показана дефинираната мрежа, на чиято база се извършват изчисленията за избор на алтернативни пътища, с определени виртуални теглови коефициенти за целите на изследването. В таблица 3.1 е показано съответствието между номер на възел / клон, съвместим към прилежащия град.



Фигура 3.3 Теглови коефициенти на виртуални алтернативни канали

Таблица 3.1

Възел / клон	Град
1	София
2	Перник
3	Кюстендил
4	Благоевград
5	Видин
6	Монтана
7	Враца
8	Плевен
9	Ловеч
10	Пазарджик
11	Смолян
12	Пловдив
13	Кърджали
14	Хасково
15	Стара Загора
16	Габрово
17	Велико Търново
18	Русе
19	Разград
20	Търговище
21	Сливен
22	Ямбол
23	Бургас
24	Шумен
25	Силистра
26	Добрич
27	Варна

3.3 Определяне на времезакъснението между възлите при добавени виртуални канали

Съгласно теорията, приблизителното изчисляване на времезакъснението между клоновете при добавените нови канали се осъществява по следният начин:

$$\tau_{об} = \frac{\text{разстояние между два възела в километри}}{200000 \text{ км/сек.}} \quad (29)$$

където, коефициента 200000 км/сек. се получава от разделянето на скоростта на светлината, която е $c=299792458$ м/сек. към показателя на пречупване на стъклото, който обикновено е около 1,5. Оттук се получава, че скоростта през оптична връзка в идеалният случай е:

$$\frac{c}{1,5} = \frac{299792458}{1,5} \approx 200000 \text{ км/сек.}$$

Полученият резултат в милисекунди дава стойността на времезакъснението на мрежовият канал между два възела. Необходимо е обаче, за повишаване точността, да се добави допълнително закъснение за обработка на пакета в крайните устройства във всеки един възел и тогава се получава [112, 113, 114]:

$$\tau_{i,j} = \tau_{об} + \sum_{i=0}^N \tau_{n,i} \quad (30)$$

където $\tau_{i,j}$ - общо стойност на времезакъснението между два възела, а закъснението за възел i , $\tau_{n,i}$ се определя по:

$$\tau_{n,i} = \tau_{вп} + \tau_{вз} + \tau_{зо} \quad (31)$$

$\tau_{об}$ - времезакъснение между два възела

N - брой възли между изпращащият и получаващият възли

$\tau_{вп}$ - време за предаване на възел i . Това е времето, необходимо на възел i да копира пакета в първия буфер, както и предаването на частите му по комуникационната връзка.

$\tau_{вз}$ - времезакъснение за възел i . Това е времето, необходимо за обработка и проверка на входящият пакет от комуникационното устройство.

$\tau_{зо}$ - закъснение в опашката във възел i . Това закъснение се отнася за времето за изчакване в буфера и зависи от характеристиките на възела.

Тъй като теоретичното измерване се извършва между два директно свързани възела, то закъснението $\tau_{вп}$ отпада и за изчисляването на закъснението се получава:

$$\tau_{n,i} = \tau_{вз} + \tau_{зо} \quad (32)$$

Тогава, за времезакъснението във възел i се получава:

$$\tau_i = \tau_{об} + \tau_{вз} + \tau_{зо} \quad (33)$$

За по-точното измерване на времезакъснението между два възела, добавяме и допълнителен коефициент на закъснение α , който се дължи на затихването в оптичното влакно [115]:

$$\alpha = \alpha_c + \alpha_k \quad (34)$$

α_k - допълнителни кабелни загуби, дължащи се на усуквания, огъвания деформации и други механични въздействия върху оптичното влакно.

α_c - собствени загуби на оптична мощност и се определят от чистотата на стъклото по следната формула.

$$\alpha_c = \alpha_n + \alpha_p \quad (35)$$

α_n - загуби на поглъщане, дължащи се на топлинни изменения в коефициента на пречупване на влакното.

α_p - загуби на разсейване, дължащи се на примеси в материала на световода.

Приемаме следната обща стойност на закъснението за всеки възел:

$$\alpha = \alpha_c + \alpha_k = 0,05 \text{ ms}$$

Тогава, за общото времезакъснение във възел i се получава:

$$\tau_i = \tau_{об} + \tau_{вз} + \tau_{зо} + \alpha \quad (36)$$

Добавяме същото закъснение и за възел j , и за общото закъснение между два възела получаваме:

$$\tau_{i,j} = \tau_{об} + \tau_{вз,i} + \tau_{зо,i} + \alpha_i + \tau_{вз,j} + \tau_{зо,j} + \alpha_j \quad (37)$$

Съгласно [112, 116] и определеното допълнително закъснение α , вземаме следните общи приблизителни стойности за закъсненията за всеки един възел:

$$\tau_{вз} + \tau_{зо} + \alpha = 0,1 + 0,05 + 0,05 = 0,2 \text{ ms}$$

3.4 Изчисляване на времезакъсненията между възлите на определените виртуални пътища

При така дефинираната нова свързаност и съгласно теоретично определения модел се изчисляват времезакъсненията на допълнителните канали между отделните крайни точки за получаване на взаимозаменяемост на каналите. Пресмятането се извършва на база изчисляването на времезакъснението между отделните възли на определената виртуална мрежа.

За произволно избран възел - клон ИО-Варна, изчисляваме времезакъснението спрямо свързаните към него възли посредством дефинираните допълнителни пътища:

За възли 27 и 26

$$\begin{aligned} \tau_{об} &= \frac{54}{200000} = 0,27 \text{ ms} \\ \tau_{27,26} &= 0,27 + 0,2 + 0,2 = 0,67 \text{ ms}. \end{aligned}$$

За възли 27 и 19

$$\begin{aligned} \tau_{об} &= \frac{132}{200000} = 0,66 \text{ ms} \\ \tau_{27,19} &= 0,66 + 0,2 + 0,2 = 1,06 \text{ ms}. \end{aligned}$$

За възли 27 и 24

$$\begin{aligned} \tau_{об} &= \frac{89}{200000} = 0,445 \text{ ms} \\ \tau_{27,24} &= 0,445 + 0,2 + 0,2 = 0,845 \text{ ms}. \end{aligned}$$

За възли 27 и 20

$$\begin{aligned} \tau_{об} &= \frac{124}{200000} = 0,62 \text{ ms} \\ \tau_{27,20} &= 0,62 + 0,2 + 0,2 = 1,02 \text{ ms}. \end{aligned}$$

За възли 27 и 21

$$\begin{aligned} \tau_{об} &= \frac{239}{200000} = 1,195 \text{ ms} \\ \tau_{27,21} &= 1,195 + 0,2 + 0,2 = 1,595 \text{ ms}. \end{aligned}$$

За възли 27 и 22

$$\tau_{06} = \frac{216}{200000} = 1,08 \text{ ms}$$

$$\tau_{27,22} = 1,08 + 0,2 + 0,2 = 1,48 \text{ ms.}$$

За възли 27 и 23

$$\tau_{06} = \frac{130}{200000} = 0,65 \text{ ms}$$

$$\tau_{27,23} = 0,65 + 0,2 + 0,2 = 1,05 \text{ ms.}$$

В таблица 3.2 са дадени разстоянията между крайните точки, в километри, като при определянето им е използвана актуална пътна карта. Дължините са изчислени, като се използват разстоянията по първокласната пътна мрежа.

Със зелен цвят са показани разстоянията между възлите при сега съществуващата мрежа, топология „звезда“. Със син цвят е въведена информацията за дефинираните алтернативни виртуални пътища между крайните точки, съгласно фигура 3.3, а останалите данни представляват допълнителните пътища от възел до възел т.е. ако се вземат всички разстояния, имаме свързаност всеки с всеки.

В таблица 3.3 са дадени всички изчислени времезакъснения между отделните крайни точки, свързани с алтернативните виртуални пътища, съгласно формула 37.

Таблица 3.2

Разстояние в километри между отделните клонове.																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Таблица 3.3

Времезакъснение в милисекунди между отделните клонове.																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Със зелен цвят са показани теоретично изчислените времезакъснения между възлите при сега съществуващата мрежа, топология звезда, съгласно формула 37. Със син цвят е въведена информацията за времезакъсненията на дефинираните алтернативни виртуални пътища между крайните точки, съгласно фигура 3.3, а останалите данни представляват времезакъсненията между останалите възли.

В таблица 3.4 са показани реалните средни времезакъснения, както и най-големите (пикови) времезакъснения на съществуващата мрежа за период от един месец. Изследването е направено посредством програмен продукт PRTG Network Monitor. Изпращането на ping заявка се извършва на всеки 60 секунди. Тази информация ще се използва при изчисляването на оптималните пътища на мрежата.

Таблица 3.4

Времезакъснение в ms между отделните клонове	Средно времезакъснение, ms	Най-голямо времезакъснение, ms
София		
Перник	15	2900
Кюстендил	16	2900
Благоевград	16	2900
Видин	23	2901
Монтана	17	2901
Враца	15	2901
Плевен	21	2901
Ловеч	21	2901
Пазарджик	15	2901
Смолян	17	2902
Пловдив	16	2887
Кърджали	23	2902
Хасково	23	2902
Стара Загора	20	2887
Габрово	19	2886
Велико Търново	18	2887
Русе	19	2887
Разград	6	2887
Търговище	6	2887
Сливен	21	2888
Ямбол	7	2887
Бургас	5	2887
Шумен	5	2887
Силистра	7	2887
Добрич	5	2887
Варна	3	2871

Получените резултати, извършени посредством програмен продукт PRTG Network Monitor и инструмента ping са разделени на две, поради факта, че времезакъснението се измерва на база изпращане и получаване на пакета в двете посоки.

В таблица 3.5 са показани средните измерени времезакъснения на съществуващата мрежа, както и теоретични изчисления такива. Получените резултати показват съществена разлика между тях, което е предпоставка до намалена пропускателна способност при реално измерените, което пък от своя страна предизвиква забавяне и по-голяма възможност за възникване на пропадания при комуникацията между възлите и намалена отказоустойчивост.

Таблица 3.5

Времезакъснение в ms между отделните клонове	Средно времезакъснение, ms	Теоритично времезакъснение, ms
София		
Перник	15	0,555
Кюстендил	16	0,830
Благоевград	16	0,880
Видин	23	1,640
Монтана	17	1,135
Враца	15	0,945
Плевен	21	1,215
Ловеч	21	1,185
Пазарджик	15	0,955
Смолян	17	1,610
Пловдив	16	1,110
Кърджали	23	1,610
Хасково	23	1,530
Стара Загора	20	1,555
Габрово	19	1,415
Велико Търново	18	1,510
Русе	19	1,955
Разград	6	2,060
Търговище	6	1,990
Сливен	21	1,935
Ямбол	7	1,885
Бургас	5	2,315
Шумен	5	2,210
Силистра	7	2,540
Добрич	5	2,805
Варна	3	2,645

3.5 Дефиниране на задачата за оптимизация за определяне свързаността на мрежата

Изследванията в дисертационната работа целят да се дефинира и реши оптимизационна задача за определяне на свързаността на мрежа като се използват резултати от измервания на времезакъсненията и пропаданията в комуникационната мрежа и се предложи решение за преминаване на съществуваща комуникационна мрежа от свързаност тип „звезда“ към двусвързана и трисвързана топология. Като параметри на тази задача са използвани реално измерените и теоретично изчислените времезакъснения между определените възли. Тези измерени и изчислени времезакъснения са използвани като теглови коефициенти в целевата функция на дефинираната и решавана оптимизационна задача.

Оптимизационната задача е дефинирана като линейна оптимизационна задача, която определя максималния поток от информация между всички двойки възли, която може да се пренесе през мрежа, при зададена топология на мрежата. Тази задача е за определяне на максимален поток в мрежа и е дефинирана и решавана в [117, 118].

Задачата за определяне на максимален поток в мрежа се дефинира като:

- зададена е топологията на мрежата
- зададена е матрицата от обеми трафик, който трябва да се предаде между всяка двойка възли в мрежата
- зададени са теглови коефициенти по дъгите на мрежата, които определят „цената“ за прокарване на единица обем трафик по съответната дъга

Решението на задачата показва какви трябва да се пропускателните способности на каналите, така, че да може да се предаде зададената матрица от обеми трафик за предаване между отделните възли.

Задачата се дефинира като задача на линейно програмиране.

$$\min c^T x \text{ като } A \cdot x \geq b$$

където,

c - е матрицата на „цените“ за прокарване на единица трафик по съответна дъга на мрежата

b - е матрицата на обема трафик, който трябва да се предаде между всяка двойка възли в мрежата

x - е решението на задачата, където стойността на x е пропускателната способност на съответната дъга в мрежата

Тази задача е използвана и модифицирана в дисертационната работа за да синтезира топология на двусвързана и трисвързана мрежа. Модификациите се състоят в:

- Използва се начална пълносвързана топология на мрежата. Решенията на задачата определят, кои линии от пълносвързаната топология да отпаднат за да се получи желана двусвързана или трисвързана мрежа.
- Решенията x се определят да принадлежат на множеството 0 или 1. При стойност $x_i = 1$, съответната дъга остава в топологията на мрежата. В случая на $x_i = 0$, дъгата не присъства в топологията. Така изходната задача за намиране на максимален поток се модифицира до линейна задача на целочисленото програмиране.
- Матрицата на необходим обем трафик, b , който трябва да се предаде в мрежата се модифицира да приема стойности 2 или 3 за всяка двойка възли. Така се синтезира топология с двусвързана или трисвързана мрежа.

Общият вид на целевата функция на задачата за синтез на топология представлява [119]:

$$\begin{aligned} \min c(x) \\ x \in A \\ \{g_i(x) \geq b_i, i = 1, n\} \end{aligned} \quad (38)$$

където,

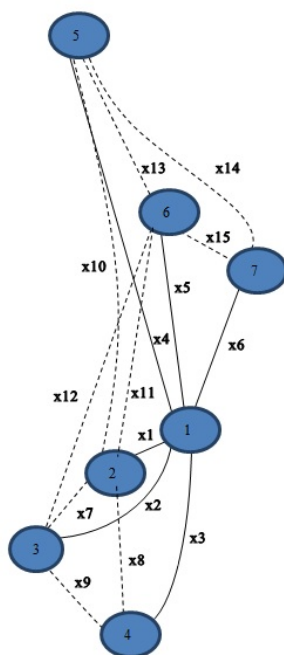
$$\begin{aligned} c(x) & - \text{целева функция} \\ g_i(x) & - \text{функция на ограниченията} \\ x & - \text{решение} \end{aligned}$$

В дисертационната работа матрицата c има инженерно съдържание на закъсненията по отделните линии. Числовите стойности на закъсненията са приети от измерванията, направени в глава 2 и показани в таблица 3.4, както и от теоритично изчислените такива в текущата глава (таблица 3.3).

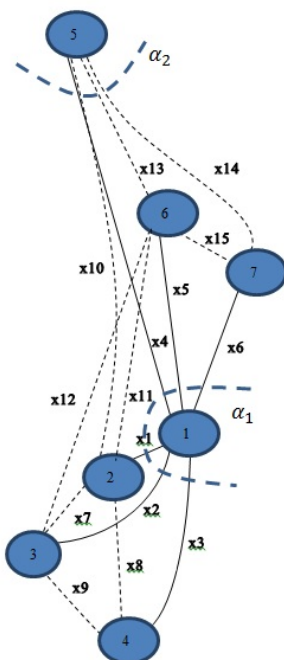
Решението на задачата x е ограничено в диапазона [0 или 1]. Така стойността на променливата x_i определя дали има комуникационен канал ($x_i = 1$) или няма канал ($x_i = 0$).

Техническа особеност е, че при дефинирането на променливите x_i се използва един индекс, вместо прилаганите в текста обяснения за x_{ij} което се тълкува като дъга между възли i и j .

За илюстриране на дефинираната целева функция за поставената задача се използва част от определената комуникационна мрежа, съгласно фигура 3.3. На фигура 3.4 са разгледани седем възела със съответни стойности на връзките между тях, а на фигура 3.5 е показано определяне на сечение за намиране на ограниченията на оптимизационната задача.



Фигура 3.4 Теглови коефициенти между възлите



Фигура 3.5 Сечение за намиране на ограниченията

Ограниченията на задачата се дефинират, съгласно модела на [117, 118] за определяне на максимален поток в мрежи, като ограниченията се дефинират като възможни сечения от дъги, чието премахване отделя всички връзки между разглежданата двойка възли.

Целевата функция за поставената задача аналитично се дефинира като:

$$\min_x \sum_i^{111} c_i x_i \quad (39)$$

При изпълнение на следните ограничения и условия за неотрицателност:

$$\begin{cases} \sum_{i=1}^{111} a_i x_i \geq b_j, j = 2;3 \\ x_i \text{ е бинарно число} \end{cases} \quad (40)$$

a_i - вектор на условията, определящ дали съществува или не връзка между два възела и приема стойности нула или едно.

Решенията x имат стойност или 1 или 0.

Вектора на ограниченията b_j приема две параметрични състояния:

- Параметрично състояние едно, което дефинира ограничение две на целевата функция за намиране на най-добрите два пътя между възлите (синтезиране на двусвързана мрежа).
- Параметрично състояние две, което дефинира ограничение три на целевата функция за намиране на най-добрите три пътя между възлите (синтезиране на трисвързана мрежа).

Обобщения израз за ограниченията на целевата функция приема вида:

$$a_1 x_1 + a_2 x_2 + a_3 x_3 + \dots + a_i x_i \geq b_j \quad (41)$$

При зададена топология в мрежата от 7 възела, съгласно фигура 3.5, се дефинират сеченията, през които минава потока между съответните възли. Тези сечения се дефинират по начин, че при премахване на съответните дъги, възлите ще бъдат откъснати.

За сечение α_1 : при премахване на дъги x_1, x_2, x_3, x_4, x_5 и x_6 възел 1 ще се отдели от останалите. Това сечение определя ограничението:

$$\alpha_1: x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 2$$

Дясната част, съдържащо числото 2 определя, че трябва да се предаде общо 2 единици трафик между възлите. На решението на задачата, това се отразява като се синтезират поне две връзки за всеки от разглежданата двойка възли. При промяна на дясната част на неравенството на три се синтезира трисвързана мрежа за съответната двойка възли.

За сечение α_2 : то съдържа четири дъги и при премахването им x_4, x_{10}, x_{13} и x_{14} , възел 5 ще се отдели от останалите. Това сечение определя аналитичното ограничението да има вида:

$$\alpha_2: x_4 + x_{10} + x_{13} + x_{14} \geq 2$$

За примера, даден на фигура 3.4 и след изпълнение на сеченията за всеки един възел (фигура 3.5) за ограниченията на целевата функция при параметрично състояние едно получаваме следните аналитични изрази:

- Възел 1

$$\begin{aligned}
x_1 + x_2 + x_3 + x_4 + x_5 + x_6 &\geq 2 \\
x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_{10} + x_{11} &\geq 2 \\
x_1 + x_3 + x_4 + x_5 + x_6 + x_7 + x_9 + x_{12} &\geq 2 \\
x_1 + x_2 + x_4 + x_5 + x_6 + x_8 + x_9 &\geq 2 \\
x_1 + x_2 + x_3 + x_5 + x_6 + x_{10} + x_{13} + x_{14} &\geq 2 \\
x_1 + x_2 + x_3 + x_4 + x_6 + x_{11} + x_{12} + x_{13} + x_{15} &\geq 2 \\
x_1 + x_2 + x_3 + x_4 + x_5 + x_{14} + x_{15} &\geq 2
\end{aligned} \tag{42}$$

- Възел 2

$$\begin{aligned}
x_1 + x_7 + x_8 + x_{10} + x_{11} &\geq 2 \\
x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_{10} + x_{11} &\geq 2 \\
x_1 + x_2 + x_8 + x_9 + x_{10} + x_{11} + x_{12} &\geq 2 \\
x_1 + x_3 + x_7 + x_9 + x_{10} + x_{11} &\geq 2 \\
x_1 + x_4 + x_7 + x_8 + x_{11} + x_{13} + x_{14} &\geq 2 \\
x_1 + x_5 + x_7 + x_8 + x_{10} + x_{12} + x_{13} + x_{15} &\geq 2
\end{aligned} \tag{43}$$

- Възел 3

$$\begin{aligned}
x_2 + x_7 + x_9 + x_{12} &\geq 2 \\
x_1 + x_3 + x_4 + x_5 + x_6 + x_7 + x_9 + x_{12} &\geq 2 \\
x_1 + x_2 + x_8 + x_9 + x_{10} + x_{11} + x_{12} &\geq 2 \\
x_2 + x_3 + x_7 + x_8 + x_{12} &\geq 2 \\
x_2 + x_5 + x_7 + x_9 + x_{11} + x_{13} + x_{15} &\geq 2
\end{aligned} \tag{44}$$

- Възел 4

$$\begin{aligned}
x_3 + x_8 + x_9 &\geq 2 \\
x_1 + x_2 + x_4 + x_5 + x_6 + x_8 + x_9 &\geq 2 \\
x_1 + x_3 + x_7 + x_9 + x_{10} + x_{11} &\geq 2 \\
x_2 + x_3 + x_7 + x_8 + x_{12} &\geq 2
\end{aligned} \tag{45}$$

- Възел 5

$$\begin{aligned}
x_4 + x_{10} + x_{13} + x_{14} &\geq 2 \\
x_1 + x_2 + x_3 + x_5 + x_6 + x_{10} + x_{13} + x_{14} &\geq 2 \\
x_1 + x_4 + x_7 + x_8 + x_{11} + x_{13} + x_{14} &\geq 2 \\
x_4 + x_5 + x_{10} + x_{11} + x_{12} + x_{14} + x_{15} &\geq 2 \\
x_4 + x_6 + x_{10} + x_{13} + x_{15} &\geq 2
\end{aligned} \tag{46}$$

- Възел 6

$$\begin{aligned}
x_5 + x_{11} + x_{12} + x_{13} + x_{15} &\geq 2 \\
x_1 + x_2 + x_3 + x_4 + x_6 + x_{11} + x_{12} + x_{13} + x_{15} &\geq 2 \\
x_1 + x_5 + x_7 + x_8 + x_{10} + x_{12} + x_{13} + x_{15} &\geq 2 \\
x_2 + x_5 + x_7 + x_9 + x_{11} + x_{13} + x_{15} &\geq 2 \\
x_4 + x_5 + x_{10} + x_{11} + x_{12} + x_{14} + x_{15} &\geq 2 \\
x_5 + x_6 + x_{11} + x_{12} + x_{13} + x_{14} &\geq 2
\end{aligned} \tag{47}$$

- Възел 7

$$\begin{aligned}
 x_6 + x_{14} + x_{15} &\geq 2 \\
 x_1 + x_2 + x_3 + x_4 + x_5 + x_{14} + x_{15} &\geq 2 \\
 x_4 + x_6 + x_{10} + x_{13} + x_{15} &\geq 2 \\
 x_5 + x_6 + x_{11} + x_{12} + x_{13} + x_{14} &\geq 2
 \end{aligned} \tag{48}$$

За конкретния пример, параметърът i приема стойност 15.

За получаване на двусвързана (трисвързана) мрежа е необходимо да се запишат сеченията на всяка двойка възли от мрежата. Така броят на ограниченията на задачата рязко се увеличава.

Общият аналитичен вид на дефинираната задача за оптимизация приема вида:

$$\min_x c_i^T x_i \begin{cases} A \cdot x_i \geq b \\ x \text{ е бинарно число} \end{cases} \tag{49}$$

където,

c_i – измерени реални и теоретично изчислени времезакъснения между възлите
Решението x определя топологията на синтезируемата комуникационна мрежа.

Ограниченията на задачата отчитат свързаността между отделните възли, съгласно поставените условия за синтезиране на двусвързана и трисвързана мрежа.

Решенията на оптимизационната задача определят нова топология на двусвързана или трисвързана мрежа. Получаваното решение може да се интерпретира и като потенциал за подобряване свързаността на дефинираната комуникационна мрежа [120].

Разбирането за оптималността се състои в определяне на топологията на синтезируемата мрежа при минимална цена, като е изпълнено условието за надеждност.

Надеждността количествено се определя от свързаността на мрежата.

Оптимизационната задача е решена с програмен продукт Матлаб чрез неговата функция `bintprog`. Тази функция има следния вид:

$$\min_x c^T x \begin{cases} A \cdot x_i \geq b \\ x \text{ е бинарно число} \end{cases} \tag{50}$$

където,

c , и b са вектори, A е матрица, и решението x се определя като целочислено бинарно векторно число - това означава, че стойностите на x могат да бъдат 0 или 1.

Задача (50) съгласно формата на Матлаб се представя във вида:

$$x = \text{bintprog}(c, A, b) \tag{51}$$

Този синтаксис решава задачите, дефинирани като бинарно целочислено програмиране, откъдето следва, че за разглеждания случай задача (50) добива вида:

$$\min_x c^T x \text{ като } A \cdot x_i \geq b_j \tag{52}$$

където, x приема стойности 0 или 1.

За примера, даден на фигура 3.4, на база получените аналитични изрази (42 - 48) и след премахване на повтарящите се редове, за матрицата на условията A се получава (таблица 3.6):

Таблица 3.6

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15
1	1	1	1	1	1									
	1	1	1	1	1	1	1		1	1				
1		1	1	1	1	1		1			1			
1	1		1	1	1		1	1						
1	1	1		1	1				1			1	1	
1	1	1	1		1					1	1	1		1
1	1	1	1	1									1	1
1						1	1		1	1				
1	1						1	1	1	1	1			
1		1				1		1	1	1				
1			1			1	1		1	1		1	1	
1				1		1	1		1		1	1		1
	1					1		1			1			
	1	1				1	1				1			
	1			1		1		1		1		1		1
		1					1	1				1		
			1						1			1	1	
			1	1					1	1	1			1
				1	1							1	1	1
				1	1					1	1	1	1	1
					1								1	1

За вектора на ограниченията при използване на зададените две параметрични състояния се получава:

$$b = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ \cdot \\ \cdot \\ \cdot \\ 2 \end{bmatrix} \text{ при синтезиране на двусвързана мрежа}$$

$$b = \begin{bmatrix} 3 \\ 3 \\ 3 \\ 3 \\ \cdot \\ \cdot \\ \cdot \\ 3 \end{bmatrix} \text{ при синтезиране на трисвързана мрежа}$$

За получаване на точни и коректни данни, параметрите на матрицата A се инвертират, като същото действие се извършва и за ограничението b . Следователно, за окончателната функция, използвана в дисертационния труд се получава [121]:

$$\min_x c^T x \text{ като } -A \cdot x_i \geq -b_j \quad (53)$$

където, x приема стойности 0 или 1.

В дисертационната работа е дефинирана задача с 27 възела. Съответствието на номера на променливата към съответната двойка възли е представено в таблица 3.7, а съответствието между номер на възел / клон, съвместим към прилежащия град е дадено в таблица 3.1.

Таблица 3.7

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	A	
1		x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21	x22	x23	x24	x25	x26	26	
2	x1		x27	x31	x28	x29				x30																		5	
3	x2	x27		x34		x32				x33																		3	
4	x3	x31	x34							x35	x36																	2	
5	x4	x28				x37	x38	x39	x40																			4	
6	x5	x29	x32		x37		x42	x41																				2	
7	x6				x38	x42		x43	x44	x45																		3	
8	x7				x39	x41	x43		x50								x49	x46	x47	x48								5	
9	x8				x40		x44	x50		x56		x55				x54	x53	x51	x52									6	
10	x9	x30	x33	x35			x45		x56		x60	x58	x59			x57												4	
11	x10			x36						x60		x61	x63	x62														3	
12	x11								x55	x58	x61		x67	x66	x65	x64												4	
13	x12									x59	x63	x67		x68														1	
14	x13										x62	x66	x68		x69							x70	x71	x72				4	
15	x14											x65		x69		x73	x74					x75	x76					4	
16	x15								x54	x57		x64				x73		x77				x78	x79					3	
17	x16						x49	x53							x74	x77		x80	x81	x82	x83							4	
18	x17						x46	x51									x80		x86						x84	x85		3	
19	x18						x47	x52									x81	x86		x91					x90	x87	x88	x89	5
20	x19						x48										x82		x91		x97	x96	x95	x93	x92		x94		6
21	x20													x70	x75	x78	x83				x97		x101	x100	x98			x99	4
22	x21													x71	x76	x79					x96	x101		x104	x102			x103	3
23	x22													x72							x95	x100	x104		x105			x106	2
24	x23																			x90	x93	x98	x102	x105		x107	x108	x109	3
25	x24																		x84	x87	x92				x107		x110		1
26	x25																		x85	x88					x108	x110		x111	1
27	x26																			x89	x94	x99	x103	x106	x109		x111		111

Матрицата А има правоъгълна форма с над 360 ограничения, определени съгласно уравненията за максимален поток в мрежата. В този брой ограничения има повтарящи се неравенства независимо, че те се отнасят за различни двойки възли. Причината за повторенията са, че съответната компонента от матрица b е едно и също число, 2 или 3. Затова при дефинирането на оптимизационната задача в дисертационния труд еднаквите неравенства са редуцирани до едно неравенство, което се прилага в оптимизационната задача. При премахване на повторенията в задачата за оптимизация са включени 141 неравенства. Съответно матрицата А се свежда до 111 колони и 141 реда. Така дефинираната и решавана задача в дисертационната работа има 111 променливи. Задачата за оптимизация е решавана със 111 неизвестни и 141 ограничения.

Задачата е модифицирана за получаване на решения за двусвързана и трисвързана мрежа.

Получените решения са представени графично за по лесно интерпретиране на резултатите.

Решенията на оптимизационната задача се разглеждат в следващата глава посредством представени и анализирани три модела със съответни параметрични състояния.

3.6 Изводи

В тази глава е синтезиран алгоритъм за определяне на теоретичните времезакъснения между възлите на добавените алтернативни комуникационни канали и е дефинирана оптимизационна задача за определяне на свързаността на мрежата като са използвани резултатите от измерванията на времезакъсненията и пропаданията в комуникационната мрежа. Решена е оптимизационна задача за определяне на структура на комуникационната мрежа като двусвързана и трисвързана. Анализирани са получените решения.

Глава 4. Анализ на резултатите от проектирането на комуникационната мрежа

4.1 Увод

Целта на тази глава е да се направи количествен анализ на получените резултати от оптимизационната задача като се разгледат три модела с по две параметрични състояния и се определи оптималния модел на информационна мрежа за преминаване от топология „звезда“ към двусвързана и трисвързана топология между отделните точки с цел постигане на непрекъсваемост на информационните потоци.

4.2 Първи модел

При този модел, тегловите коефициенти на съществуващата мрежа се взимат съгласно таблица 3.4 от предходната глава. Останалите теглови коефициенти се взимат съгласно таблица 3.3, които представляват теоретично изчислените стойности. Използват се две параметрични състояния за модела за намиране на оптимално топологично решение. Всяко едно състояние, съгласно целевата функция се дефинират както следва:

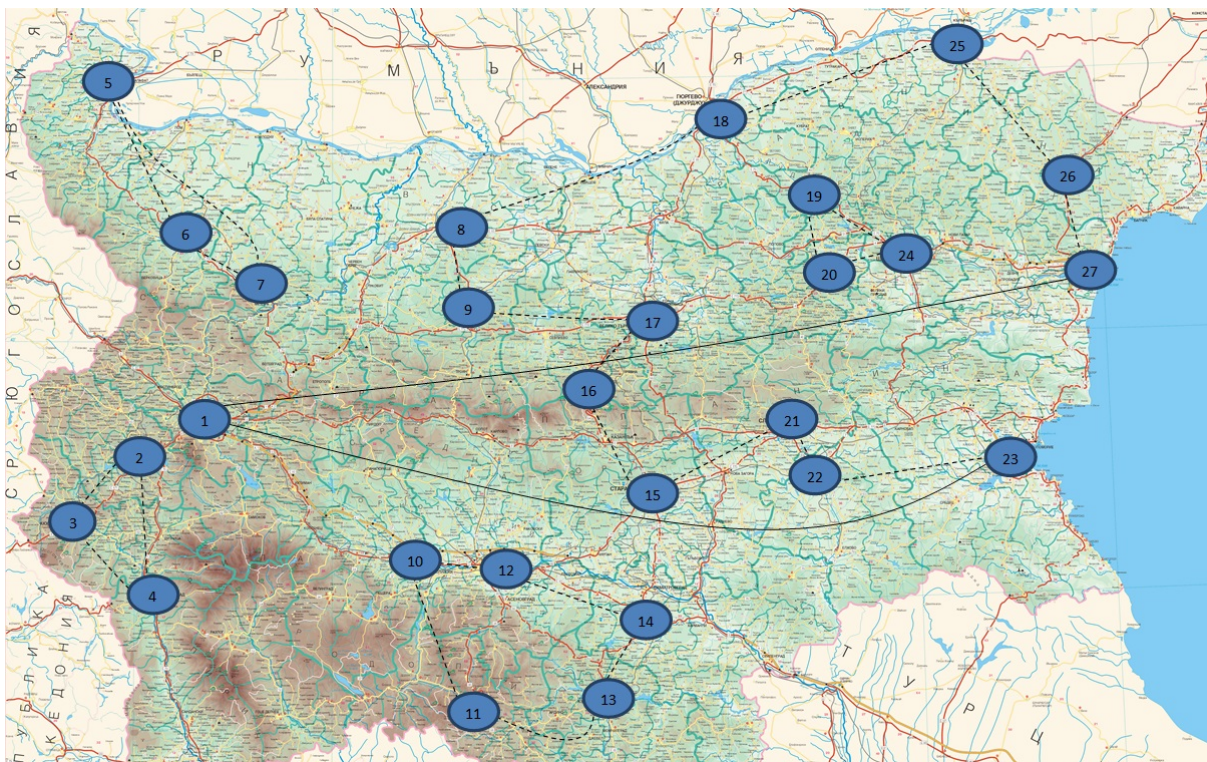
- Параметрично състояние едно за топологичен синтез на двата най-добри пътя.
- Параметрични състояние две за топологичен синтез на трите най-добри пътя.

На фигура 4.1 е показан резултата от параметрично състояние едно за топологичен синтез на двата най-добри пътя, а на фигура 4.2, параметрично състояние две, за топологичен синтез на трите най-добри пътя.

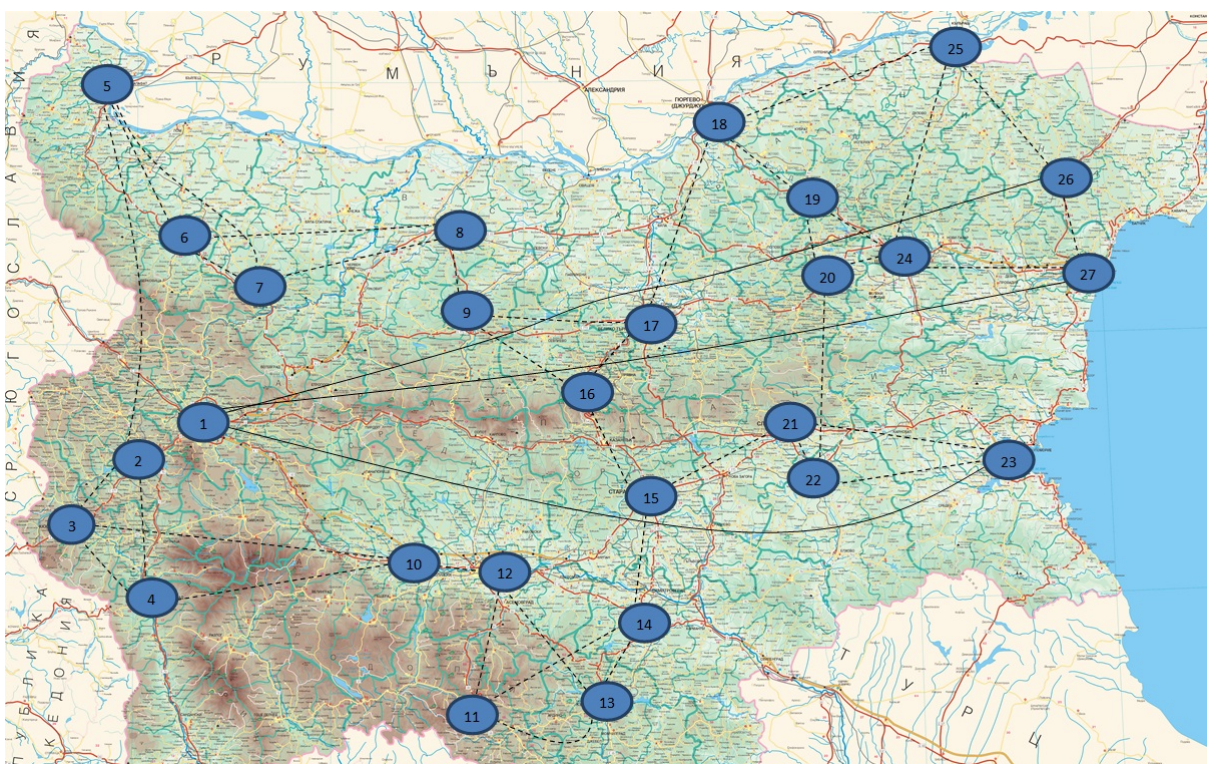
На първата фигура се вижда, че от настоящата мрежа тип „звезда“ остават само две връзки. Също така се забелязва, че се получава откъснатост на няколко групи възли от основната мрежа и поради тази причина, това параметрично състояние е неприложимо за целите на дисертационния труд.

На следващата фигура, от вече съществуващата мрежа остават три връзки. Няма откъснати възли от основната идея на мрежата (както се случва при предишното параметрично състояние), но се получава несиметричност, тъй като възел 1 няма връзка със съседните си точки, а е свързан с крайните такива - 23, 26, 27. Това параметрично състояние е приемливо с оглед цялостното разпределение на алтернативните пътища и тяхната взаимозаменяемост.

На база получените резултати, може да се направи извода, че параметрично състояние две е приемлив вариант в някаква степен, като възможен за подобряване на съществуващата комуникационна мрежа.



Фигура 4.1 Модел с параметрично състояние едно за топологичен синтез на двата най-добри пътя



Фигура 4.2 Модел с параметрично състояние две за топологичен синтез на трите най-добри пътя

4.3 Втори модел

При този модел, тегловите коефициенти на съществуващата мрежа се взимат с минимална стойност, за конкретния случай се използва коефициент нула, което изразява, че вече изградената мрежа има твърда връзка с останалите крайни точки. Останалите теглови коефициенти се взимат съгласно таблица 3.3, които представляват теоретично изчислените стойности. Използват се две параметрични състояния за модела за намиране на оптимално топологично решение. Всяко едно състояние, съгласно целевата функция се дефинират както следва:

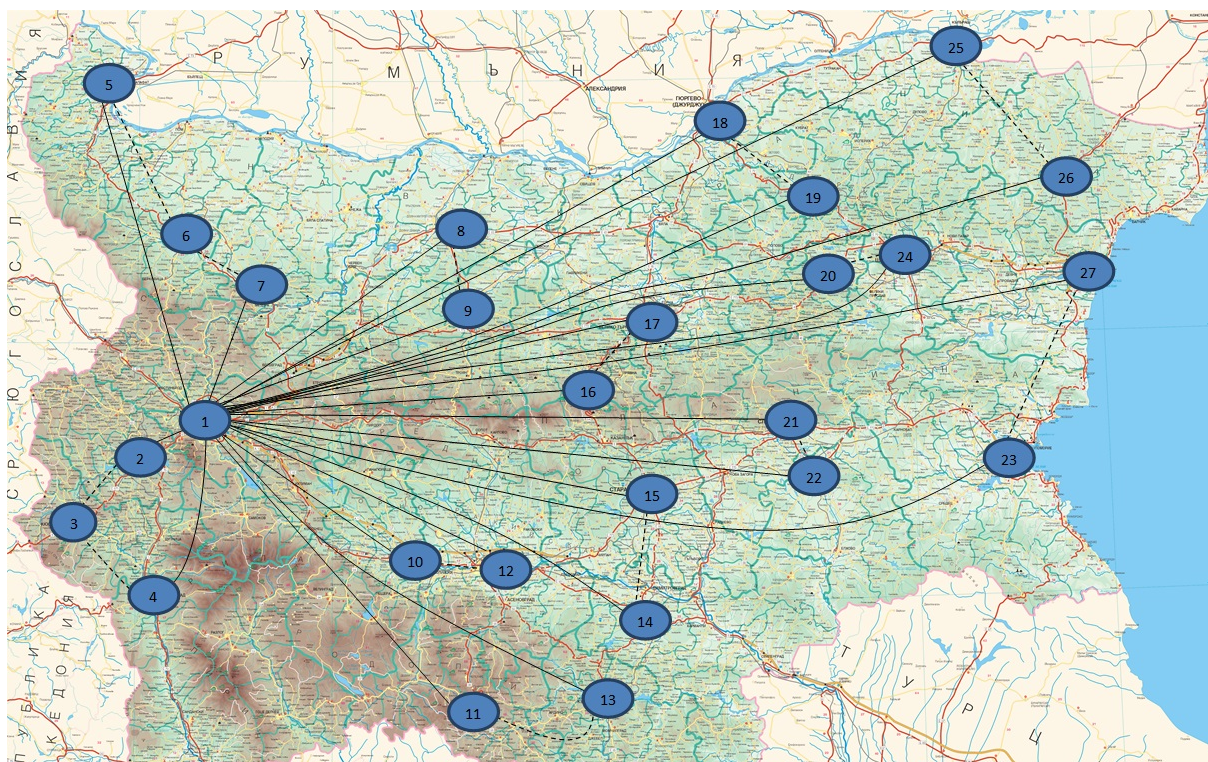
- Параметрично състояние едно за топологичен синтез на двата най-добри пътя.
- Параметрични състояние две за топологичен синтез на трите най-добри пътя.

На фигура 4.3 е показан резултата от параметрично състояние едно за топологичен синтез на двата най-добри пътя, а на фигура 4.4, параметрично състояние две, за топологичен синтез на трите най-добри пътя.

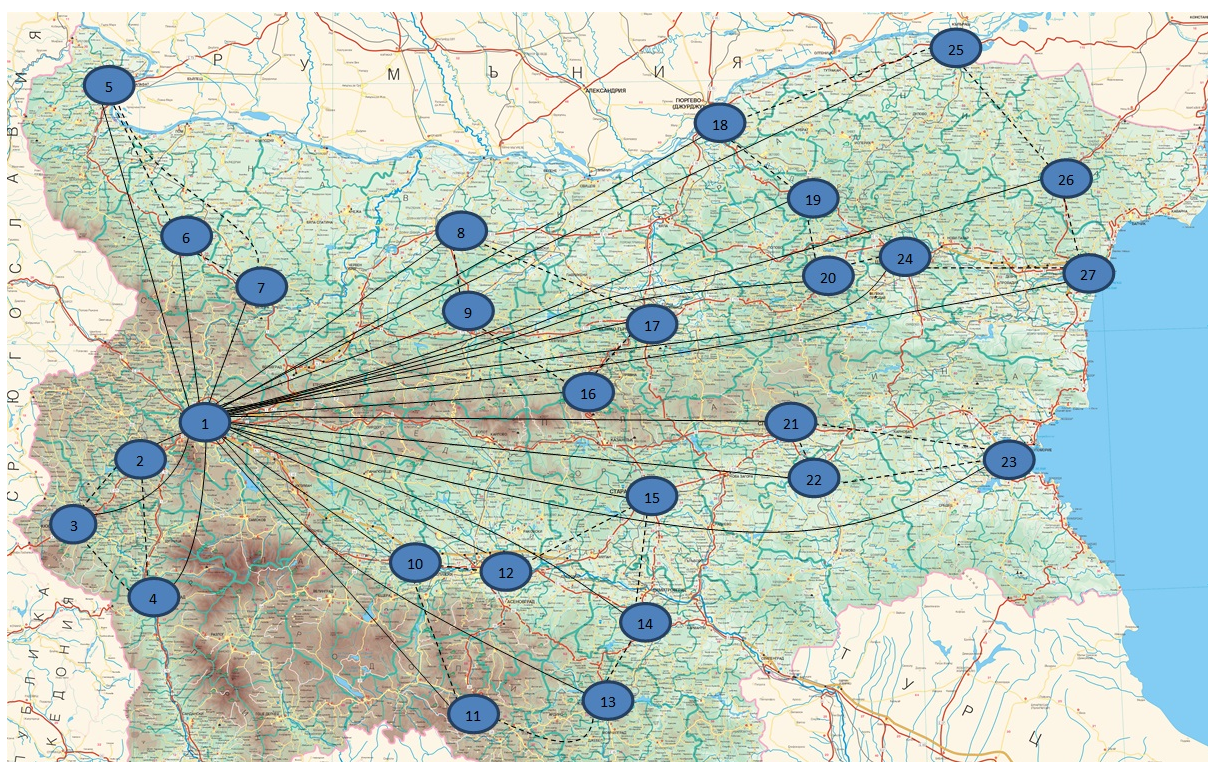
На първата фигура се вижда, че при параметрично състояние едно за този модел, въпреки зададени нулеви стойности на съществуващата мрежа, с цел запазване на всички налични връзки, на база получените резултати се премахват връзките между възли 1 и 3, както и между 1 и 6. Така получената мрежа не е приемлива, тъй като при прекъсване на централния възел, отново ще има срыв във връзките между отделните клонове.

На втората фигура, при параметрично състояние две за модела и зададено топологично решение за минимум три алтернативни пътища, се запазват всички връзки на съществуващата мрежа. Така получената топология отново е неприемлива, тъй като при получаване на прекъсване в централния възел, ще се загуби комуникацията между отделните клонове, което ще породи срыв и загуба на информационни ресурси.

От получените резултати за този модел може да се направи извод, че той не е приложим да целите на дисертационния труд, тъй като не води до подобряване непрекъсваемостта на обслужването на информационните ресурси, както и не подобрява структурата на комуникационната мрежа с цел определяне на взаимнозаменяеми алтернативни маршрути.



Фигура 4.3 Модел с параметрично състояние едно за топологичен синтез на двата най-добри пътя



Фигура 4.4 Модел с параметрично състояние две за топологичен синтез на трите най-добри пътя

4.4 Трети модел

При този модел, тегловите коефициенти на съществуващата мрежа се взимат съгласно таблица 3.3 от предходната глава, където са показани теоретично изчислените времезакъснения на всички определени възли. Използват се две параметрични състояния за модела за намиране на оптимално топологично решение. Всяко едно състояние, съгласно целевата функция се дефинират както следва:

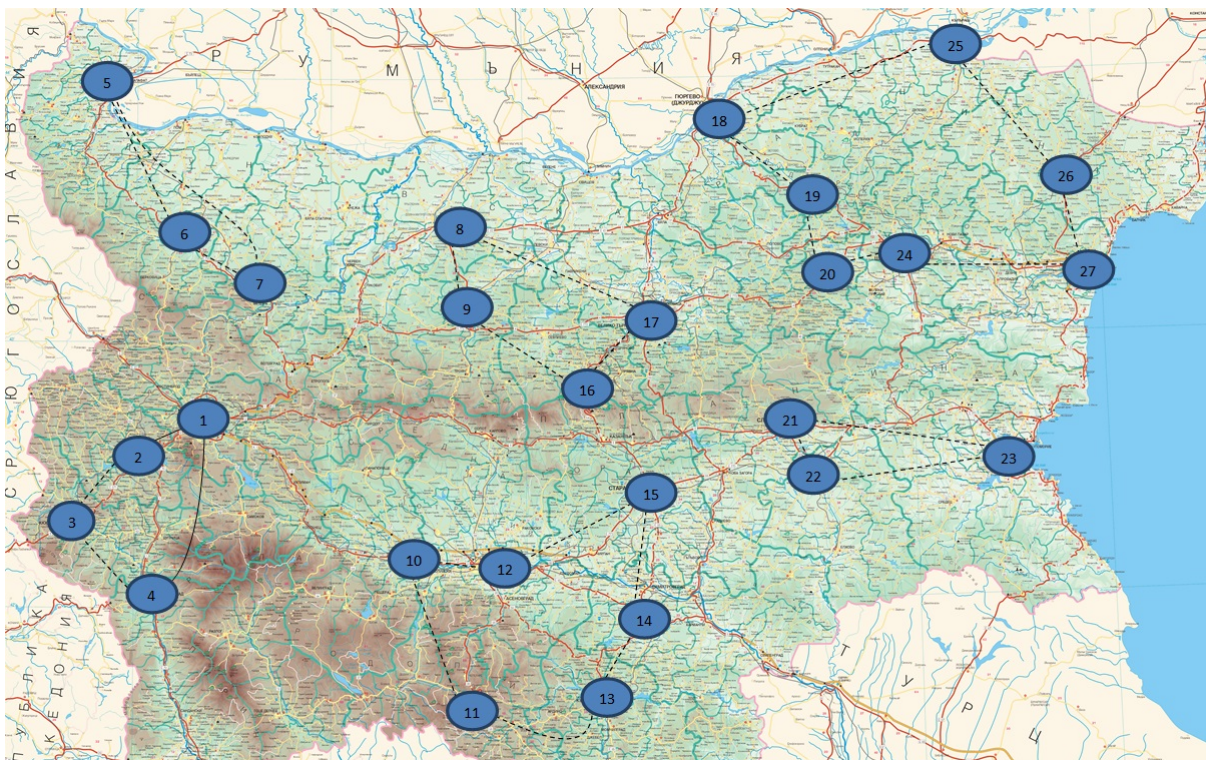
- Параметрично състояние едно за топологичен синтез на двата най-добри пътя.
- Параметрични състояние две за топологичен синтез на трите най-добри пътя.

На фигура 4.5 е показан резултата от параметрично състояние едно за топологичен синтез на двата най-добри пътя, а на фигура 4.6, параметрично състояние две, за топологичен синтез на трите най-добри пътя.

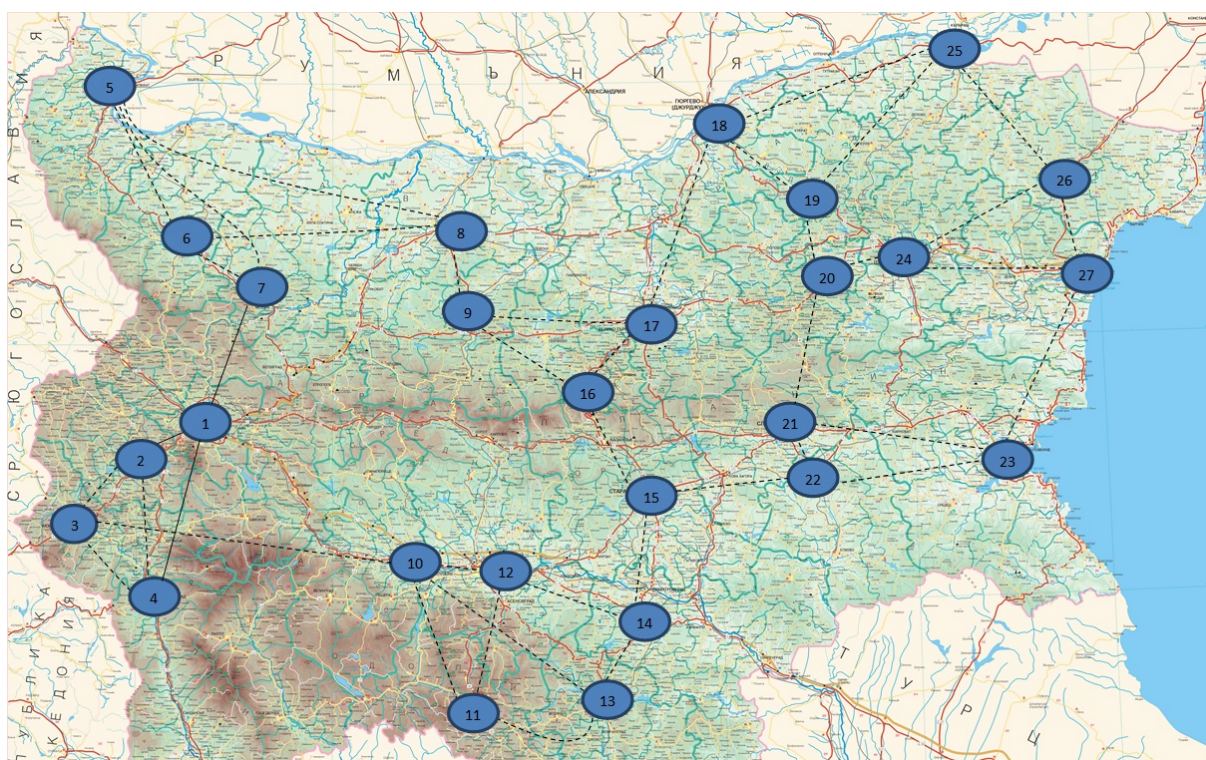
На първата фигура се вижда, че при параметрично състояние едно за този модел, се получава разпокъсаност на мрежата, като полученият резултат води до шест, напълно самостоятелно обособени групи възли. Поради тази причина, получената топология на мрежа не води до подобряване на сегашната комуникационна структура и не може да се използва като алтернатива за осигуряване непрекъснатост на информационните потоци и услуги.

На втората фигура, при параметрично състояние две за модела, се постига достатъчно добър резултат, като няма откъснати възли, както се получава при предишното състояние. Възникват няколко критични възлови точки (между 1 и 7, 8 и 9, 3 и 10, 14 и 15 възли), но чрез подходящо планиране е възможно премахването им, посредством добавяне на допълнителни резервни канали за връзка. Това параметрично състояние е приемливо с оглед цялостното разпределение на алтернативните пътища и тяхната взаимозаменяемост.

На база получените резултати, параметрично състояние две за модела се явява подходящ за подобряване на съществуващата комуникационна мрежа.



Фигура 4.5 Модел с параметрично състояние едно за топологичен синтез на двата най-добри пътя



Фигура 4.6 Модел с параметрично състояние две за топологичен синтез на трите най-добри пътя

4.5 Анализ на получените резултати

Сега съществуващата комуникационна мрежа тип „звезда“ води до прекъсване, пропадане и нестабилност за поддържане на постоянна връзка между отделните клонове и тяхната връзка един с друг.

След разработване на трите възможни модела със съответните им параметрични състояния, преминаваме към графично им изобразяване. За тази цел представяме 27-те възли в квадратна матрица (27 x 27), като всеки ред и колона отговаря на един възел (от 1 до 27). Максималният брой на възможните комбинации може да се изчисли по следната формула:

$$N * \frac{N-1}{2} \quad (54)$$

където, N е броят на възлите.

За конкретната квадратна таблица при N = 27 възела, максималният брой на възможните комбинации между всички възли е 351. След изключване на елементите по главния диагонал, получаваме две симетрични триъгълни матрици, които отразяват търсените комбинации между възлите. От общия възможен брой комбинации (351) са избрани само 111, означени по-горе с елементите от x1 до x111 (таблица 4.1). В таблицата са посочени тези 111 комбинации, както и техните симетрични стойности спрямо главния диагонал.

Таблица 4.1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	A	
1		x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21	x22	x23	x24	x25	x26	26	
2	x1		x27	x31	x28	x29				x30																		5	
3	x2	x27		x34		x32				x33																		3	
4	x3	x31	x34							x35	x36																	2	
5	x4	x28				x37	x38	x39	x40																			4	
6	x5	x29	x32			x37		x42	x41																			2	
7	x6					x38	x42		x43	x44	x45																	3	
8	x7					x39	x41	x43		x50								x49	x46	x47	x48							5	
9	x8					x40		x44	x50		x56		x55					x54	x53	x51	x52							6	
10	x9	x30	x33	x35			x45		x56		x60	x58	x59					x57										4	
11	x10			x36						x60		x61	x63	x62														3	
12	x11								x55	x58	x61		x67	x66	x65	x64												4	
13	x12									x59	x63	x67		x68														1	
14	x13										x62	x66	x68		x69							x70	x71	x72				4	
15	x14											x65		x69		x73	x74						x75	x76				4	
16	x15								x54	x57		x64				x73		x77					x78	x79				3	
17	x16							x49	x53						x74	x77		x80	x81	x82	x83							4	
18	x17							x46	x51								x80		x86						x84	x85		3	
19	x18							x47	x52								x81	x86		x91					x90	x87	x88	x89	5
20	x19							x48									x82			x91		x97	x96	x95	x93	x92		x94	6
21	x20												x70	x75	x78	x83				x97		x101	x100	x98			x99	4	
22	x21												x71	x76	x79					x96	x101		x104	x102			x103	3	
23	x22												x72							x95	x100	x104		x105			x106	2	
24	x23																		x90	x93	x98	x102	x105		x107	x108	x109	3	
25	x24																		x84	x87	x92				x107		x110		1
26	x25																		x85	x88					x108	x110		x111	1
27	x26																			x89	x94	x99	x103	x106	x109		x111		111

В таблица 4.2 са показани критериите, които се използват при оценка на трите модела

Таблица 4.2

А	Сума на връзките на даден възел с другите възли
Б	Сума на времезакъсненията на връзките на даден възел с останалите
В	Зависимост на системата от един възел
Г	Брой групи възли

Конкретните стойности на времезакъсненията на елементите от x1 до x111 за всеки от получените и описани по-горе три модела, всеки с по две параметрични състояния, е представен в отделна квадратна таблица (27 x 27), както и симетричните им стойности спрямо главния диагонал. Следващата стъпка за графичното изображение на отделните модели е свързване на получените стойности. За по-ясно представяне, свързването на възлите е илюстрирано чрез оцветяване на стойностите на съответните времезакъснения с различни цветове, вместо с линии. Използваната методика е последователно свързване на всички възли в дадената колона или ред на таблицата. При прекъсване на съответните връзки между възлите (откъснатост между група възли), се налага промяна на цветовете. След свързване на всички възли по цялата таблица ясно се виждат обособените гнезда и прекъсване на връзките между възлите. Избрани са първите два критерия А и Б с цел прилагане на количествен анализ на трите модела. Критерий А е сума на броя на връзките на даден възел, като не се включват съответните симетрични стойности, а само стойностите на връзките, надясно от главния диагонал. Критерий Б включва сума на времезакъсненията по връзките на даден възел, представен на съответния ред с останалите възли. За получаване на сумата на времезакъсненията по критерий Б също не са включени съответните симетрични стойности, а само времезакъсненията по връзките, надясно от главния диагонал. В две последователни колони след всяка таблица са сумирани броя на връзките на дадения възел, представен на съответния ред, както и времезакъсненията (Сума А и сума Б). Двете суми А и Б не включват съответните симетрични стойности, а само стойностите на връзките, надясно от главния диагонал.

Към така определените два критерия добавяме допълнително критерий В (зависимост на системата от един възел) и критерий Г (брой групи възли).

В таблици 4.3 и 4.4 са показани получените резултати от двете параметрични състояния на първи модел. В първата таблица се получава обособяване на 5 групи на свързаност между възли, оцветени в пет различни цвята. Във втората таблица всички възли са свързани помежду си в една група, оцветена в един цвят.

При този модел, използвайки реално измерените времезакъснения за теглови коефициенти, за параметрично състояние едно, се получава откъснатост на отделни групи възли и по тази причина не може да бъде използвано за подобряване свързаността на мрежата. За параметрично състояние две на този модел се получава приемлив резултат, но недостатъкът е получаването на несиметричност, тъй като възел едно няма връзка със своите съседни точки, базирани на географски принцип. При това състояние се наблюдават и критични точки на достъп.

Таблица 4.3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	А	Б
1																							5,000				3,000	2	8,000
2			0,690	0,810																								2	1,500
3		0,690		0,745																								1	0,745
4		0,810	0,745																									0	0,000
5						0,920	1,115																					2	2,035
6					0,920		0,595																					1	0,595
7					1,115	0,595																						0	0,000
8								0,575										1,150										2	1,725
9								0,575									0,865											1	0,865
10											1,080	0,590																2	1,670
11											1,080		1,165															1	1,165
12											0,590			0,790														1	0,790
13												1,165		0,635														1	0,635
14												0,790	0,635															0	0,000
15															0,805						0,755							2	1,560
16															0,805	0,620												1	0,620
17								0,865							0,620													0	0,000
18								1,150																		1,005		1	1,005
19																			0,590					0,655				2	1,245
20																		0,590						0,615				1	0,615
21															0,755							0,545						1	0,545
22																						0,545	0,860					1	0,860
23	5,000																						0,860					0	0,000
24																			0,655	0,615								0	0,000
25																	1,005									0,850		1	0,850
26																									0,850	0,670		1	0,670
27	3,000																									0,670		0	0
																												27	27,695

Таблица 4.4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	А	Б
1																							5,000			5,000	3,000	3	13,000
2			0,690	0,810	1,835																							3	3,335
3		0,690		0,745						1,370																		2	2,115
4		0,810	0,745							1,420																		1	1,420
5		1,835				0,920	1,115																					2	2,035
6					0,920		0,595	1,035																				2	1,630
7					1,115	0,595		0,940																				1	0,940
8						1,035	0,940		0,575																			1	0,575
9								0,575								0,770	0,865											2	1,635
10			1,370	1,420								0,590																1	0,590
11												0,890	1,165	1,265														3	3,320
12									0,590	0,890		1,040																1	1,040
13											1,165	1,040		0,635														1	0,635
14											1,265		0,635		0,710													1	0,710
15														0,710		0,805					0,755							2	1,560
16								0,770							0,805		0,620											1	0,620
17								0,865								0,620		0,945										1	0,945
18																	0,945		0,730						1,005			2	1,735
19																		0,730		0,590				0,655				2	1,245
20																			0,590			1,025		0,615				2	1,640
21															0,755							0,545	0,975					2	1,520
22																				1,025	0,545		0,860					1	0,860
23	5,000																				0,975	0,860						0	0,000
24																			0,655	0,615					0,975		0,845	2	1,820
25																	1,005							0,975		0,850		1	0,850
26	5,000																								0,850		0,670	1	0,670
27	3,000																							0,845		0,670		0	0
																												41	46,445

В таблици 4.5 и 4.6 са показани получените резултати от двете параметрични състояния на втори модел. И в двете таблици, всички възли са свързани помежду си в една група, оцветена в един цвят. Налице е пълна зависимост от възел 1, който е свързан с всички останали.

Освен свързаността между възлите, в таблиците са включени стойностите по двата избрани критерия А и Б. Както бе споменато по-горе, на всеки ред в колонка А са посочени броя на връзките на всеки възел с другите възли (без техните симетрични стойности). На всеки ред в колонка Б е посочена съответната сума от времезакъсненията на даден възел. Впоследствие резултатите от тези критерии ще бъдат използвани за сравнение на различните варианти и оценка.

Например, ако разгледаме първи ред, възел 1 е свързан с останалите 24 възела, без възел 3 и възел 6. Сумата на времезакъсненията е 0. На втори ред, възел 2 е свързан с възел 3 със стойност на времезакъснението 0,69. Тези резултати са отразени в последните две колонки (А, Б) на втори ред - съответно 1 и 0,69. Последователно разглеждаме трети ред и свързаността на възел 3 - съответно с възел 2 и 4. Но както посочихме, симетричните стойности се изключват от сумарните резултати в таблицата. По тази причина на трети ред, в колонка А е изписано само 1 (свързаността с възел 4) и съответното времезакъснение - 0,745. За всеки следващ ред по един и същи начин се получават стойностите за всеки възел, а на последния ред - техните сумарни стойности. Съгласно получените резултати за двете параметрични състояния от втори модел, може да се заключи, че той не е подходящ за използване, поради запазване на настоящата структура и несиметричност на добавените нови връзки. Това би довело отново до липса на сигурна и надеждна връзка между клоновата мрежа в разглежданата структура.

В таблици 4.7 и 4.8 са показани получените резултати от двете параметрични състояния на трети модел. В първата таблица са обособени 6 групи на свързаност между възлите, оцветени в шест различни цвята. Във втората таблица, всички възли са свързани помежду си в една група, оцветена в един цвят.

Разглеждайки последния модел, параметрично състояние едно води до постоянна откъснатост между отделни групи възли и тази топология не е приложима, както останалите досега споменати състояния. При параметрично състояние две за този модел се постига най-оптимално топологично разположение на получените алтернативни пътища, въпреки някои критични точки.

Таблица 4.5

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	А	Б
1		0,000		0,000	0,000		0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	24	0,000
2	0,000		0,690																									1	0,690
3		0,690		0,745																								1	0,745
4	0,000		0,745																									0	0,000
5	0,000					0,920																						1	0,920
6				0,920		0,595																						1	0,595
7	0,000				0,595																							0	0,000
8	0,000							0,575																				1	0,575
9	0,000						0,575																					0	0,000
10	0,000									0,590																		1	0,590
11	0,000										1,165																	1	1,165
12	0,000							0,590																				0	0,000
13	0,000								1,165																			0	0,000
14	0,000												0,710															1	0,710
15	0,000												0,710															0	0,000
16	0,000													0,620														1	0,620
17	0,000													0,620														0	0,000
18	0,000														0,730													1	0,730
19	0,000														0,730													0	0,000
20	0,000																						0,615					1	0,615
21	0,000																				0,545							1	0,545
22	0,000																			0,545								0	0,000
23	0,000																										1,050	1	1,050
24	0,000																			0,615								0	0,000
25	0,000																									0,850		1	0,850
26	0,000																								0,850			0	0,000
27	0,000																					1,050						0	0
																												38	10,400

Таблица 4.6

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	А	Б
1		0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	26	0,000
2	0,000		0,690	0,810																								2	1,500
3	0,000	0,690		0,745																								1	0,745
4	0,000	0,810	0,745																									0	0,000
5	0,000					0,920	1,115																					2	2,035
6	0,000				0,920		0,595																					1	0,595
7	0,000				1,115	0,595																						0	0,000
8	0,000							0,575									1,075											2	1,650
9	0,000							0,575								0,770												1	0,770
10	0,000									1,080	0,590																	2	1,670
11	0,000									1,080		1,165																1	1,165
12	0,000									0,590				0,910														1	0,910
13	0,000										1,165		0,635															1	0,635
14	0,000											0,635	0,710															1	0,710
15	0,000										0,910	0,710																0	0,000
16	0,000							0,770								0,620												1	0,620
17	0,000						1,075									0,620												0	0,000
18	0,000																	0,730						1,005				2	1,735
19	0,000																	0,730	0,590									1	0,590
20	0,000																		0,590					0,615				1	0,615
21	0,000																				0,545	0,975						2	1,520
22	0,000																				0,545		0,860					1	0,860
23	0,000																				0,975	0,860						0	0,000
24	0,000																			0,615						0,845		1	0,845
25	0,000																	1,005							0,850			1	0,850
26	0,000																							0,850		0,670		1	0,670
27	0,000																						0,845		0,670			0	0
																												52	20,690

Таблица 4.7

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	А	Б
1		0,555		0,880																								2	1,435
2	0,555		0,690																									1	0,690
3		0,690		0,745																								1	0,745
4	0,880		0,745																									0	0,000
5						0,920	1,115																					2	2,035
6					0,920		0,595																					1	0,595
7					1,115	0,595																						0	0,000
8								0,575									1,075											2	1,650
9							0,575										0,770											1	0,770
10										1,080	0,590																	2	1,670
11										1,080		1,165																1	1,165
12										0,590				0,910														1	0,910
13											1,165			0,635														1	0,635
14												0,635		0,710														1	0,710
15											0,910		0,710															0	0,000
16								0,770									0,620											1	0,620
17							1,075									0,620												0	0,000
18																		0,730							1,005			2	1,735
19																		0,730		0,590								1	0,590
20																			0,590					0,615				1	0,615
21																					0,545	0,975						2	1,520
22																					0,545		0,860					1	0,860
23																					0,975	0,860						0	0,000
24																				0,615						0,845		1	0,845
25																			1,005							0,850		1	0,850
26																									0,850		0,670	1	0,670
27																								0,845		0,670		0	0
																												27	21,315

Таблица 4.8

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	А	Б
1		0,555		0,880			0,945																					3	2,380
2	0,555		0,690	0,810																								2	1,500
3		0,690		0,745						0,865																		2	1,610
4	0,880	0,810	0,745																									0	0,000
5						0,920	1,115	1,155																				3	3,190
6					0,920		0,595	1,035																				2	1,630
7	0,945				1,115	0,595																						0	0,000
8					1,155	1,035			0,575																			1	0,575
9								0,575								0,770	0,865											2	1,635
10			0,865								1,080	0,590	1,230															3	2,900
11										1,080		0,890	1,165															2	2,055
12										0,590	0,890			0,790														1	0,790
13										1,230	1,165			0,635														1	0,635
14												0,790	0,635		0,710													1	0,710
15														0,710		0,805						0,840						2	1,645
16								0,770							0,805		0,620											1	0,620
17								0,865								0,620		0,945										1	0,945
18																	0,945		0,730						1,005			2	1,735
19																		0,730		0,590					1,055			2	1,645
20																				0,590		1,060			0,615			2	1,675
21																					1,060		0,545	0,975				2	1,520
22															0,840							0,545		0,860				1	0,860
23																						0,975	0,860				1,050	1	1,050
24																					0,615					1,035	0,845	2	1,880
25																		1,005	1,055							0,850		1	0,850
26																								1,035	0,850		0,670	1	0,670
27																							1,050	0,845		0,670		0	0
																												41	34,705

Количественият анализ на трите модела със съответните параметрични състояния е представен в таблица 4.9 и представлява сравнение между стойностите на получените суми по двата критерия А и Б.

Таблица 4.9

Таблица	Модел	Параметрично състояние	А	Б	В	Г
4.3	Първи модел	Параметрично състояние едно	27	27,695	ниска	5
4.4	Първи модел	Параметрично състояние две	41	46,445	ниска	1
4.5	Втори модел	Параметрично състояние едно	38	10,400	висока	1
4.6	Втори модел	Параметрично състояние две	52	20,690	висока	1
4.7	Трети модел	Параметрично състояние едно	27	21,315	ниска	6
4.8	Трети модел	Параметрично състояние две	41	34,705	ниска	1

Съгласно получените резултати, таблици 4.5 и 4.6, които изобразяват втори модел с параметрични състояния едно и две, имат най-ниски стойности по критерий Б (сума на времезакъсненията) и липсват прекъсвания между възлите, но са силно зависими от един възел (възел 1).

Таблицы 4.3 (първи модел с параметрично състояние едно) и 4.7 (трети модел с параметрично състояние едно) са следващите две стойности по критерий Б (сума на времезакъсненията), но при тях са налице прекъсвания между възлите (съответно 5 и 6 обособени независими групи).

Таблицы 4.4 (първи модел с параметрично състояние две) и 4.8 (трети модел с параметрично състояние две) имат най-високи стойности по критерий Б, но липсват прекъсвания между възлите и зависимост от един възел. Следователно, най-добрият вариант за избор на разпределена мрежа се явява трети модел с параметрично състояние две, тъй като е с по-ниска стойност на сумата от времезакъсненията.

На база разгледаните три модела за подобряване непрекъсваемостта на комуникационното и информационно обслужване, както и качеството на доставяните услуги, може да се направи извода, че последния разгледан модел (трети модел с параметрично състояние две) е оптимален за получаване на приемливо ниво на алтернативни пътища, което ще доведе до непрекъсваемост и сигурност на информационния ресурс.

Приносът на дисертационния труд се състои в подобряване на комуникационната структура на разглежданата мрежа посредством направен топологичен синтез, което да доведе до осезателно намаляване на пропаданията и загуба на връзка между отделните крайни точки. Това води до оптимална свързаност и намаляване загубите на информационни ресурси вследствие подобряване отказоустойчивостта на предложената комуникационна мрежа.

4.6 Изводи

В настоящата глава са разгледани и анализирани три модела за определяне на оптималния вариант за подобряване на съществуващата комуникационна мрежа. Направен е количествен анализ на получените резултати с цел постигане на непрекъсваемост на информационните потоци. На база топологичен синтез е избран най-добрият модел със съответно параметрично състояние за преминаване от съществуващата топология „звезда“ към разпределена трисвързана мрежа между отделните възлови точки.

Заклучение

За нуждите на дисертацията са анализирани 121 литературни източника, свързани с основните положения, методи, модели и алгоритми за формализиране на информационни и комуникационни потоци в мрежи чрез графи, необходими за преминаване на информационната мрежа от свързаност тип „звезда“ към двусвързана и трисвързана топология. Оценени са техните особености, предимства и недостатъци. Разгледани са проблемите за намиране на най-кратък път от един възел към всички останали възли в мрежи, за които цената на дъгата може да варира. Тези задачи имат влияние за определяне на оптимална топология на комуникационна мрежа. Направени са основни изводи на базата на проучените литературни източници.

Определени са чрез измервания и са анализирани пропускателната способност и времезакъсненията в мрежата при предаване на информационни потоци, като е оценено пропадането на връзка между двойка възли като характеристика на отказоустойчивостта на съществуваща комуникационна мрежа със свързаност тип „звезда“.

На базата на специализирана литература по този въпрос е синтезиран алгоритъм за определяне на теоретичните времезакъснения между възлите на добавените алтернативни комуникационни канали и е дефинирана оптимизационна задача за определяне на свързаността на мрежата като са използвани резултатите от измерванията на времезакъсненията и пропаданията в комуникационната мрежа.

Решена е оптимизационна задача за определяне на структура на комуникационната мрежа като двусвързана и трисвързана. Анализирани са получаваните решения и е направен количествен анализ на получените резултати с цел постигане на непрекъсваемост на информационните потоци.

Посредством получените резултати от оптимизационната задача са разгледани и анализирани три модела за определяне на оптималния вариант за подобряване на съществуваща комуникационна мрежа. Направена е количествена оценка на получените резултати. На база топологичен синтез е избран най-добрият модел със съответно параметрично състояние за преминаване от съществуваща топология „звезда“ към разпределена трисвързана мрежа между отделните възлови точки с цел постигане на непрекъсваемост на информационните потоци.

Данните и измерванията в дисертацията са направени на реално съществуващата комуникационна мрежа, която се използва в практиката на фирма Информационно обслужване АД.

Приноси на дисертационния труд

Основните резултати имат научно приложен и приложен характер.

1. Анализирани са моделите за формализиране на информационни и комуникационни потоци в мрежи чрез графи и влиянието на топологията на мрежата. Оценени са техните особености, предимства и недостатъци. Разгледани са проблемите за намиране на най-кратък път от един възел към всички останали възли в мрежи, за които цената на дъгата може да варира.
2. Определени са чрез измервания и са анализирани пропускателната способност и времезакъсненията в мрежата при предаване на информационни потоци. Оценявано е пропадането на връзка между двойка възли като характеристика на отказоустойчивостта на съществуваща комуникационна мрежа със свързаност тип „звезда“.
3. Синтезиран е алгоритъм за определяне на теоритичните времезакъснения между възлите на добавените алтернативни комуникационни канали. Измерените и теоритично изчислени времезакъснения са използвани при дефинирането на задача за оптимален синтез на топология на мрежата.
4. Дефинирана е и е решавана оптимизационна задача за определяне на структура на комуникационната мрежа с 27 възела като двусвързана и трисвързана. Анализирани са получаваните решения. Направен е количествен анализ на получените резултати с цел постигане на непрекъсваемост на информационните потоци.

Данните и измерванията в дисертацията са направени на реално съществуващата комуникационна мрежа, която се използва в практиката на фирма Информационно обслужване АД.

Бъдещи насоки за работа

На основание получения резултат от определената оптимална комуникационна мрежа, бъдещата работа ще продължи в няколко насоки:

- Следене състоянието и надеждността на мрежата за възникване на критични точки.
- Повишаване качеството на обслужване с оглед недопускане пропадания на връзката между възлите.
- Повишаване на продуктивността на мрежата с цел намаляване времезакъсненията.
- Повишаване на пропускателната способност на комуникационните канали.
- Анализирание на необходимостта от добавяне или промяна на определените алтернативни пътища на мрежата.
- Подобряване точността на алгоритъма за определяне на времезакъсненията при въвеждане на взаимнозаменяеми комуникационни връзки.

Публикации, свързани с дисертацията

1. Krasimir Trichkov, Elisaveta Trichkova, Elena Ivanova. Application Aspects of Web Services. E-learning Conference'07, Istanbul, Turkey, 27-28 August, 2007, ISBN: 978-975-6437-75-9, p. 167-171.
2. Krasimir Trichkov, Elisaveta Trichkova, Elena Ivanova. Integrated Information System Based on Web Services. International Conference on Computer Systems and Technologies - CompSysTech'2007, Rousse, Bulgaria, 14-15 June, 2007, ISBN: 978-954-9641-50-9, IIIA.16-1 – IIIA.16.5.
3. Trichkov Kr. Distributed Searching in Global Network. Preprints of the International IFAC Workshop DECOM-TT 2004, Automatic Systems for Building the Infrastructure in Developing Countries, Regional and Global Aspects, October 3-5, 2004, Bansko, Bulgaria, p.235-238, ISBN 954-9641-41-4
4. Trichkov Kr. Standardization Approach for Information Retrieval in WAN. Proceedings of the International Conference of Computer Systems and Technologies (e-learning), CompSysTech, 2004, Rousse, Bulgaria, 17-18 June, p.IIB.6-1-IIB.6-6
5. Trichkova E., K.Trichkov, Network management systems and security management in web services. Международна научна конференция "Информатика в научното познание 2008", 26-28 юни 2008 г., Варна, България, ISSN 1313-4345, p.194-204.
6. Trichkov K., Elisaveta Trichkova, Modeling and Execution on Web Services in the Web Information Systems, National Scientific Conference "Mathematics, Informatics and Computer Sciences"- St. Cyril and St. Methodius University of Veliko Tarnovo, Veliko Tarnovo, Bulgaria, 12-13 May, 2006, 184-189.
7. Елена Иванова, Тодор Стоилов, Красимир Тричков, Елисавета Тричкова. Защита на Web услуги в глобални мрежи. Конференция Автоматика и Информатика'07, 3-6 Октомври 2007, ISSN 1313-1850, стр. VI-35 – VI-38.
8. Елена Иванова, Тодор Стоилов, Красимир Тричков, Елисавета Тричкова, Съвременни информационни решения на Web услуги в глобални мрежи в интерес на сигурността. Научна поддръжка на трансформацията на сектора за сигурност, Академично издателство „ проф. Марин Дринов” - БАН, 2006, София, ст. 172-183. ISBN-10: 954-322-175-8, ISBN-13: 978-954-322-175-2, ISBN-10: 954-91827-2-X, ISBN-13: 978-954-91872-2-9.

Научноизследователски проекти

Изследванията в дисертационния труд са част от получените резултати на следните научноизследователски проекти:

1. Международен проект по 6-та Рамкова програма на Европейската комисия, Информационно общество. Project EC FP6, program IST No. 027178 "Virtual Internet Service Provider VISP"
2. Договор за отпускане на финансова подкрепа за проект за мобилност по секторна програма „Леонардо да Винчи“. Програма „Учене през целия живот“. 2008-1-BG1-LEO03-00374. Тема „Обмяна на опит за иновационни приложения на решения за електронно обучение“.
3. Международен проект по програма „Обучение през целия живот“ - ERASMUS. Договор с Европейската комисия: 142399-LLP-1-2008-1-BG-ERASMUS-ENW на тема: „ETN Teaching, Research, Innovation in Computing Education“.
4. The thematic Network Teaching, Research and Innovation in Computing Education - TRICE is established in accordance with the ERASMUS Programme of the EUROPEAN COMMISSION.
5. Европейски Социален Фонд 2007-2013 Г. Програма "Развитие На Човешките Ресурси" . Подкрепа на творческото развитие на докторанти, пост-докторанти и млади учени в областта на компютърните науки, BG 051PO001-3.3.04/13.
6. Проект 4ИФ-02-87/7.3.2008 „Интегриране на GPS функционалности в информационна система за оптимизиране на междуведомствени управленски дейности с приложение за координиране на специализирани служби“ към Изпълнителна агенция за насърчаване на малките и средни предприятия - Министерство на промишлеността.

Декларация за оригиналност на резултатите

Декларирам, че настоящата дисертация съдържа оригинални резултати, получени при проведени от мен научни изследвания, с подкрепата и съдействието на научния ми ръководител. Резултатите, които са получени, описани и/или публикувани от други учени, са надлежно и подробно цитирани в библиографията.

Настоящата дисертация не е прилагана за придобиване на научна степен в друго висше училище, университет или научен институт.

К. Тричков:

Библиография

1. Стоилов Т. Експериментална WAN и LAN мрежа за моделиране и оптимизация на трафик. Сборник доклади, Национална конференция с международно участие "Компютърни системи и технологии КомпСисТех'2000", 22-23 юни 2000, София. стр. IV-4.1-IV-4.5.
2. Deo N., Pang C.Y., "Shortest Path Algorithms: Taxonomy and Annotation," Networks 14 pp., 275-323 (1984).
3. Carlos J., M.Papageorgiou, C.Schaffner. Integrated Optimal routing and congestion control in packet-switched network. IFAC Symposium Large Scale Systems: Theory and Applications, Aug.23-25, 1992, Beijing China, Vol.1, p.59-64.
4. Стоилов Т. Оптимизация на достъпа до общ комуникационен канал. Национална конференция по Автоматика и информатика'98. Сборник доклади. *Симпозиум "Проблеми на информационните технологии"*, Том 1, София, 20-23 октомври 1998, стр. 9-12.
5. Наков О., Р. Трифонов, Г. Попов, П. Вачков, С. Манолов, Р. Йошинов. Мрежова и информационна сигурност. София, Изд. "Авангард Прима" – 2013, 367 стр. ISBN 978-619-160-183-7.
6. Castro J., N.Nabona. Nonlinear multicommodity network flows through primal partitioning and comparison with alternative methods. IFIP Conference on Systems Modeling and Optimization, Compiègne, France 5-9 July 1993, p.875-887.
7. Kirov G., Soft computing models for network routing, International Conference "Automatics and Informatics'2002", Sofia, Bulgaria, November 5-7.
8. Chen Z., T. Berger. Performance analysis of random routing algorithm for n-D connected networks. , TENCON'93 , Beijing, China, october 19-21 1993, p.233-236.
9. Huitema Chr. Routing in the Internet. Prentice Hall, New Jersey, 1995, 319 p.
10. Pitsillides A., J.F.Lambert, R.E.Warfield. A hierarchical control approach for broadband ISDN Communication networks. 13 IFAC Congress 1993, Sidney, Australia, vol.7, p.223-226.
11. Woo, S. H., Yang S.B., An improved network clustering method for I/O-efficient query processing. In ACM-GIS, pages 62–68, 2000.
12. Shekhar S., Liu D.R., CCAM: A connectivity-clustered access method for networks and network computations. IEEE Trans. Knowl. Data Eng., 9(1):102–119, 1997.
13. Jiang B., I/O-efficiency of shortest path algorithms: An analysis. In ICDE, pages 12–19, 1992.
14. Tipper D., M. Sundareshan. Numerical Methods for Modeling Computer Networks under Nonstationary Conditions. IEEE Journal on selected area in communications, vol.8, N9, 1990, p.1682-1695.
15. Jing N., Huang Y.W., Rundensteiner E. A., Hierarchical optimization of optimal path finding for transportation applications. In CIKM, pages 261–268, 1996.
16. Kimura T. Approximations for the waiting time in the GI/G/S queue. Journal of the O.R.Society of Japan, 34, 1992, 2(June), p.173-186.
17. G. Spasov, "Future Internet Technology", Journal Automatics & Informatics, Year XLIII, № 3/2009, pp. 11-15, ISSN 0861-7562.

18. Стоилов Т. Проблеми при експериментирането в компютърни WAN и LAN мрежи. Международна Конференция "Автоматика и информатика '2000", 24-26 октомври 2000, Сборник доклади, т.1, Симпозиум Компютърни и комуникационни системи в приложения с повишена отговорност. Симпозиум Информационни технологии. София, стр.5-7.
19. Наков О., Р. Трифонов, Г. Попов, П. Вачков, С. Манолов, Р. Йошинов. Мрежова и информационна сигурност. София, Изд. "Авангард Прима" – 2013, 367 стр. ISBN 978-619-160-183-7.
20. Ling S.T., Furuno K., Tezuka Y., Optimal Path in Networks with Time-Varying Traverse Time and Expenses Branches, Technological Reports of Osaka University, Japan (1972).
21. Dochev, D., Yoshinov, R. Pavlov. (2000) An Open Distributed Computer Environment for Educational Telematics, in Proceedings of the EDEN Fourth Open Classroom Conference, Barcelona, Spain, November 19-21, pp.247-252.
22. S. Abbasi, S. Ebrahimnejad, Finding the Shortest Path in Dynamic Network using Labeling Algorithm, International Journal of Business and Social Science, Vol. 2 No. 20; November 2011, 239 - 243.
23. E.W. Dijkstra, A note on two papers in connection with graphs, Numeriske Mathematics 1 (1959) 269–271.
24. R.W. Floyd, Algorithm 97: Shortest paths, Communications of the ACM 5 (1962) 345.
25. L. Fu, Real-time vehicle routing and scheduling in dynamic and stochastic networks, Ph.D. Thesis at the University of Alberta, 1996.
26. E.S. Dreyfus, An appraisal of some shortest path algorithms, Operations Research 17 (1969) 395–412.
27. Philpott, A.B., Mees, A.I.: Continuous-time shortest path problems with stopping and starting costs. Appl. Math. Lett. 5, 63–66 (1992).
28. Ahuja, R.K., Orlin, J.B., Pallottino, S., Scutella, M.G.: Dynamic Shortest Paths Minimizing Travel Times and Costs. Networks 41, 197–205 (2003).
29. X. Cai, T. Kloks, C.K. Wong, Time-varying shortest path problems with constraints, Networks 29 (3) (1997) 141–149.
30. Chabini, L. Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time. Journal of the Transportation Research Board, 1645:170–175, 1998.
31. A. Orda, R. Rom, Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length, Journal of ACM 37 (3) (1990) 607–625.
32. Orda, A., Rom, R.: Minimum weight paths in time-dependent networks. Networks 21, 295–320 (1991).
33. Orda A., Rom R., Distributed Shortest-Path Protocols for Time-Dependent Networks, pp. 439-445 in Proceedings of ICC 88, Tel Aviv, Israel (November 1988).
34. Pallottino, S., Scutella, M.G.: Shortest path algorithms in transportation models: Classical and innovative aspects. In: Marcotte, P., Nguyen, S. (eds.) Equilibrium and advanced transportation modelling, pp. 245–281. Kluwer, Norwell (1998).

35. Philpot, A.B., Mees, A.I.: A finite-time algorithm for shortest path problems with time-varying costs. *Appl. Math. Lett.* 6, 91–94 (1993).
36. Chon H. D., Agrawal D., Abbadi A. E., Fates: Finding a time dependent shortest path. In *Mobile Data Management*, pages 165–180, 2003.
37. Foschini L., Hershberger J., Suri S., On the Complexity of Time-Dependent Shortest Paths, A preliminary version of this paper was presented at the 22nd ACM-SIAM Symposium on Discrete Algorithms (SODA), 2011.
38. S. E. Dreyfus. An appraisal of some shortest-path algorithms. *Operations, Research*, 17(3):395–412, 1969.
39. H.D. Sherali, K. Ozbay, and S. Subramanian. The time-dependent shortest pair of disjoint paths problem: Complexity, models, and algorithms. *Networks*, 31(4): 259 - 272, 1998.
40. X. Cai, T. Kloks, and C. K.Wong. Time-varying shortest path problems with constraints. *Networks*, 29(3):141–150, 1997.
41. K. Nachtigall. Time depending shortest-path problems with applications to railway networks. *European Journal of Operational Research*, 83(1):154–166, 1995.
42. L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, New Jersey, 1962.
43. Ford Jr L.R., Fulkerson D.R., Constructing Maximal Dynamic Flows from Static Flows, *Operations Research* 6 pp. 419-433 (1958).
44. B. Ding, J. X. Yu, and L. Qin. Finding time-dependent shortest paths over large graphs. In *Proc. of the 11th International Conference on Extending Database Technology*, pages 205–216, 2008.
45. E. Kanoulas, Y. Du, T. Xia, and D. Zhang. Finding fastest paths on a road network with speed patterns. In *Proc. of the 22nd International Conference on Data Engineering*, page 10, 2006.
46. B. C. Dean. Continuous-time dynamic shortest path algorithms. Master’s thesis, Massachusetts Institute of Technology, 1999.
47. B. C. Dean. Shortest paths in FIFO time-dependent networks: Theory and algorithms. Technical report, 2004.
48. Shekhar S., Fetterer A., Goyal B., Materialization trade-offs in hierarchical shortest path algorithms. In *SSD*, pages 94–111, 1997.
49. Jing N., Huang Y.W., Rundensteiner E. A., Hierarchical encoded path views for path query processing: An optimal model and its performance evaluation. *IEEE Trans. Knowl. Data Eng.*, 10(3):409–432, 1998.
50. Narváez P., Siu K. Y., Tzeng H.Y., New dynamic SPT algorithm based on a ball-and-string model. *IEEE/ACM Trans. Netw.*, 9(6):706–718, 2001.
51. Narváez P., Siu K.Y., Tzeng H.Y., New dynamic algorithms for shortest path tree computation. *IEEE/ACM Trans. Netw.*, 8(6):734–746, 2000.
52. Huang Y. W., Jing N., Rundensteiner E. A., Effective graph clustering for path queries in digital map databases. In *CIKM*, pages 215–222, 1996.
53. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.

54. Пенчев С., Т. Стоилов. Овременени мрежи на Петри (ОМП) - йерархични модели, класификация и области на приложение в комуникационните процеси. Национална конференция по Автоматика и Информатика '97, Сборник доклади, Програмни средства и системи, компютърни системи и мрежи, приложни задачи на информатиката. Том 6, 1997, стр.65-68.
55. V. King. Fully dynamic algorithms for maintaining all-pairs shortest paths and transitive closure in digraphs. In Proc. of the 40th Annual Symposium on Foundations of Computer Science, pages 81–89, 1999.
56. L. Roditty and U. Zwick. On dynamic shortest paths problems. In Proc. of 12th Annual European Symposium on Algorithms, volume 3221, pages 580–591, 2004.
57. M. Thorup. Worst-case update times for fully-dynamic all-pairs shortest paths. In Proc. of the 37th Annual ACM Symposium on Theory of Computing, pages 112–119, 2005.
58. Thorup M., Undirected single-source shortest paths with positive integer weights in linear time. J. ACM, 46(3):362–394, 1999.
59. Demetrescu C., Emiliozzi S., Italiano G. F., Experimental analysis of dynamic all pairs shortest path algorithms. In SODA, pages 369–378, 2004.
60. D. P. Bertsekas and J. N. Tsitsiklis. An analysis of stochastic shortest path problems. Mathematics of Operations Research, 16(3):580–595, 1991.
61. E. Nikolova, M. Brand, and D. R. Karger. Optimal route planning under uncertainty. In Proc. of International Conference on Automated Planning and Scheduling, 2006.
62. E. Nikolova, J. A. Kelner, M. Brand, and M. Mitzenmacher. Stochastic shortest paths via quasi-convex maximization. In Proc. of 14th European Symposium on Algorithms, pages 552–563, 2006.
63. J.S.B. Mitchell and C.H. Papadimitriou. The weighted region problem: Finding shortest paths through a weighted planar subdivision. Journal of the ACM, 38(1):18–73, 1991.
64. Jung S., Pramanik S., Hiti graph model of topographical roadmaps in navigation systems. In ICDE, pages 76–84, 1996.
65. Sung K., G. Bell M., Seong M., Park S., Shortest paths in a network with time-dependent flow speeds. European Journal of Operational Research, 121(12):32–39, 2000.
66. P. J. Carstensen. Parametric cost shortest path problems. Unpublished Bellcore memo, 1984.
67. K. Mulmuley and P. Shah. A lower bound for the shortest path problem. In Proc. of the 15th Annual IEEE Conference on Computational Complexity, pages 14–21, 2000.
68. D. M. Gusfield. Sensitivity analysis for combinatorial optimization. PhD thesis, University of California, Berkeley, 1980.
69. R.M. Karp and J.B. Orlin. Parametric shortest path algorithms with an application to cyclic staffing. Discrete Applied Mathematics, 3(1):37–45, 1981.
70. N. Young, R. Tarjan, and J. Orlin. Faster parametric shortest path and minimum balance algorithms. Networks, 21(2):205–221, 2006.

71. J. Erickson. Maximum flows and parametric shortest paths in planar graphs. In Proc. of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms, pages 794–804, 2010.
72. F. Dehne, M. T. Omran, and J.-R. Sack. Shortest paths in timedependent FIFO networks using edge load forecasts. In Proc. of 2nd International Workshop on Computational Transportation Science, pages 1–6, 2009.
73. F. Dehne, M.T. Omran, and J.R. Sack. Shortest paths in time-dependent FIFO networks. *Algorithmica*, pages 1–20, 2010.
74. Handley S., Langley P., Rauscher F. A., Learning to predict the duration of an automobile trip. In KDD, pages 219–223, 1998.
75. Frigioni D., Marchetti-Spaccamela A., Nanni U., Fully dynamic output bounded single source shortest path problem (extended abstract). In SODA, pages 212–221, 1996.
76. D. Delling and D. Wagner. Time-dependent route planning. In Robust and Online Large-Scale Optimization, volume 5868, pages 207–230. 2009.
77. G. Nannicini, D. Delling, L. Liberti, and D. Schultes. Bidirectional A^* search for time-dependent fast paths. In Workshop on Experimental Algorithms, volume 5038 of Lecture Notes in Computer Science, pages 334–346, 2008.
78. Even S., Graph Algorithms, Computer Science Press, New York (1979).
79. <http://tu-materiali.com/download.php?id=1061>, Гоцева Д., Лекции по САА
80. Frigioni D., Ioffreda M., Nanni U., Pasquale G., Experimental analysis of dynamic algorithms for the single-source shortest-path problem. *ACM Journal of Experimental Algorithms*, 3:5, 1998.
81. Demetrescu C., Italiano G. F., A new approach to dynamic all pairs shortest paths. *J. ACM*, 51(6):968–992, 2004.
82. Громницкий В.С. Методы оптимизации, Курс лекций, Государственное образовательное учреждение высшего профессионального образования «Нижегородский государственный университет им. Н.И. Лобачевского», Экономический факультет, Кафедра экономической информатики, УДК 65.012.122 (075.8) ББК Увбя73 Г 87
83. The Ping Page Researched and Produced by PING 127.0.0.1 Computer Services New York City, NY, USA, <http://www.ping127001.com/pingpage.htm>, Update May, 2004.
84. *Traceroute* <http://www.freessoft.org/CIE/Topics/54.html> .
85. Examine your network with Ping and Traceroute, <http://hotwired.lycos.com/webmonkey/97/42/index3a.html?tw=backend> .
86. Lakov D., G. Kirov, Routing of computer nets via fuzzy logic, Youth Science Session, Sofia, Bulgaria (in Bulg.), July 1-2, 1999, pp. 100-105.
87. Allen B. Downey, *Using Pathchar to estimate Internet link characteristics*, Colby College, www.colby.edu/downey. Applications, Technologies, Architectures, and Protocols for Computer Communication, p. 241 – 250, 1999.
88. Ravi S.Prasad, Costantinos Dovrolis, Bruce A. Mah, *The effect of Layer-2 switches on Pathchar-like tools*, Networking and Telecommunications Group, College of computing, Georgia Tech. Internet Measurement Workshop, Session 10 active measurement, p. 321 – 322, 2002 .

89. Van Jacobson, *Pathchar- A tool to infer characteristics of Internet paths*, Network Research Group, Lawrence Berkeley National Laboratory. ACM SIGCOMM'99 is the annual conference of the Special Interest Group on Data Communication, September, 1999.
90. Bruce A. Mah, *Pchar: Child of Pathchar*, Security and Networking Research Department, Sandia National Laboratories, California, 21 July, 1999.
91. <http://etd.adm.unipi.it/theses/available/etd-06242004-163624/unrestricted/4MeasuringNetworkDelay.doc>
92. Daduna H., *Queueing Networks with Discrete Time Scale*. Springer, 2001.
93. Gallager R., *Discrete Stochastic Processes*. Kluwer Academic Publishers, Boston, 1996.
94. Bertsekas D. P., Gallager R., *Data Networks*. New Jersey: Prentice - Hall, Inc., 1992.
95. Jafari R., Sohrawy K., Combined $m/g/1-g/m/1$ type structured chains: A simple algorithmic solution and applications. IEEE Proceedings of INFOCOM, Vol. 2, April 2001.
96. He J., Sohrawy K., A new analysis framework for discrete time queueing systems with general stochastic sources. IEEE Proceedings of INFOCOM, Vol. 2, April 2001.
97. Roberts J.W., Virtamo J. T., The superposition of periodic cell arrival streams in an atm multiplexer. IEEE Transactions on Communications, Feb. 1991.
98. Morrison J. A., Two discrete-time queues in tandem. IEEE Transactions on Communications, Vol. COM-27, No. 3, March 1979.
99. Walley S. K., Viterbi A. M., A tandem of discrete-time queues with arrivals and departures at each stage. Queueing Systems, Vol. 23, pp. 157-176, 1996.
100. Wang Q., Frost V., Estimation of cell loss probabilities for tandem atm queues. IEEE Proceedings of the International Conference on Communications, 1994.
101. Cruz R. L., A calculus for network delay. i. network elements in isolation. IEEE Transactions on Information Theory, vol. 37:1, pp. 114-131, Jan. 1991.
102. Cruz R. L., A calculus for network delay. ii. network analysis. IEEE Transactions on Information Theory, vol. 37:1, pp. 132-141, Jan. 1991.
103. Parekh A. K., Gallager R., A generalized processor sharing approach to flow control in integrated services networks: The single-node case. IEEE/ACM Transactions on Networking, Vol. 1, pp. 344-357, 1993.
104. Neely M. J., Rohrs C. E., Equivalent models and analysis for multistage tree networks of deterministic service time queues. Proceedings of the 38th Annual Allerton Conference on Communication, Control, and Computing, Oct. 2001.
105. N. Kakanakov, E. Kostadinova, G. Spasov, "Using SNMP for Remote Measurement and Automation", Proc. Conf. Electronics ET'07, 20-22 Sept 2007, Book 2, pp. 85-90, ISBN: 1313-1842.
106. Modiano E., Wielselthier J. E., Ephremides A., A simple analysis of average queueing delay in tree networks. IEEE Transactions on Information Theory, Vol. IT-42, no. 2, pp. 660-664, March 1996.
107. Бонева Т., Т. Стоилов. Наблюдение на IP трафика в компютърни мрежи. Национална конференция по Автоматика и информатика'98. Сборник доклади.

Симпозиум "Проблеми на информационните технологии", Том 1, София, 20-23 октомври 1998, стр. 13-16.

108. Morrison J. A., A combinatorial lemma and its applications to concentrating trees of discrete-time queues. Bell Sys. Tech. J., May-June 1978.
109. N. Kakanakov, M. Shopov, G. Spasov, "Time-Delay Simulation Analysis of Local Controller Networks", ACM International Conference Proceeding Series; Vol. 374, Proceeding of CompSysTech'08, Gabrovo, June 2008, pp. III-A.12-1-6, ISBN:978-954-9641-52-3.
110. Aiello W., Ostrovsky R., Kushilevitz E., Ros' en A., Dynamic routing on networks with fixed-size buffers. Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, Jan. 2003.
111. <http://www.mit.edu/~modiano/papers/J2.pdf>
112. http://researchwebshelf.com/uploads/137_P15.pdf
113. Cooke K.L., Halsey E., The Shortest Route Through a Network with Time Dependent Internodal Transit Times, Journal Math. Analysis Applications 14 pp. 493-498 (1966).
114. http://web.mit.edu/dimitrib/www/Queueing_Data_Nets.pdf
115. <https://tcom-sf.org/forums/download.php?id=228>
116. Halpern J., The Shortest Route with Time Dependent Length of Edges and Limited Delay Possibilities in Nodes, Zeitschrift fuer Operations Research 21 pp. 117-124 (1977).
117. Л. Р. Форд, Д. Р. Фалкерсон, Потоки с сетях, Издательство „Мир“, Москва, 1966.
118. Л. С. Лэсдон, Оптимизация больших систем, Издательство „Наука“, Москва, 1975.
119. Стоилов Т. Неитеративна координация в йерархични системи. Академично издателство на БАН, "Проф.М.Дринов", София, 1998, стр.272
120. Д. Младенов, К. Стоилова, Т.Стоилов. Теория и практика на йерархичните системи. Издателство на Българската академия на науките, София, 1989.
121. Stoilova K., T. Stoilov, R. Yoshinov. Predictive Coordination in Two Level Hierarchical Systems. WSEAS Transaction of Mathematics. Issue 3, Volume 3, July 2004. p.631-636.

Използвани съкращения

WAN - Wide Area Network
FIFO - First In First Out
NP - Nondeterministic polynomial time
EC - Earliest completion time
LC - Latest completion time
Ping - Packet internet groper
RTT - Round-Trip Time
IP - Internet Protocol
DNS - Domain System Service
TTL - Time To Live
LAN - Local Area Network
ICMP - Internet Control Message Protocol
MTU - Maximum Transmission Unit
PRTG - Paessler Router Traffic Grapher
WMI - Windows Management Instrumentation
SSH - Secure SHell
SNMP - Simple Network Management Protocol
jFlow - Juniper Flow
HTTP - Hypertext Transfer Protocol
SMTP - Simple Mail Transfer Protocol
POP3 - Post Office Protocol, version 3
FTP - File Transfer Protocol
RMON - Remote Network MONitoring
IPFIX - Internet Protocol Flow Information Export
SLA - Service-Level Agreement
SLOPS - Self-Loading Periodic Streams
VPN - Virtual Private Network
ISP - Internet Service Provider