

Sprawozdanie z listy 2 na laboratorium

Technologie Sieciowe

Wojciech Strzelecki (254618)

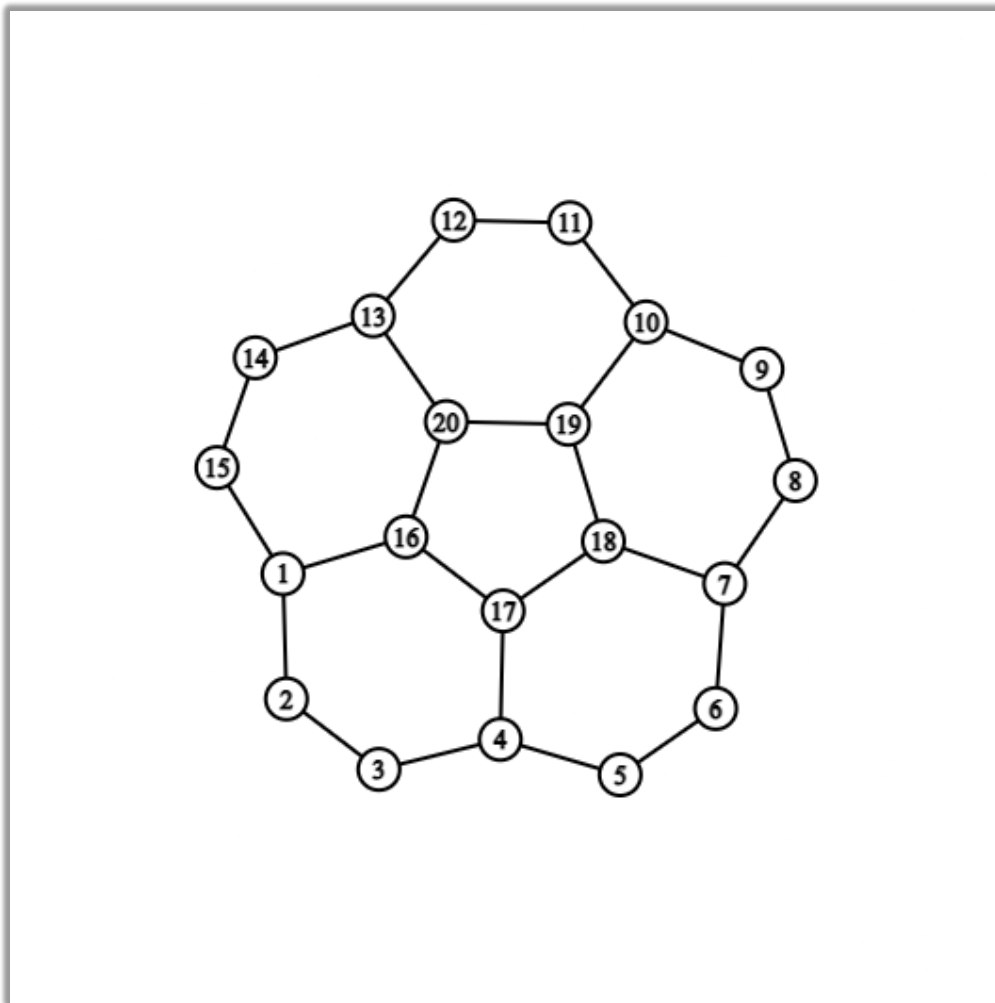
1. Wstęp

1.1 Polecenie

2. Rozważmy model sieci $S = \langle G, H \rangle$. Przez $N = [n(i, j)]$ będziemy oznaczać macierz natężeń strumienia pakietów, gdzie element $n(i, j)$ jest liczbą pakietów przesyłanych (wprowadzanych do sieci) w ciągu sekundy od źródła $v(i)$ do ujścia $v(j)$.
- Zaproponuj topologię grafu G ale tak aby żaden wierzchołek nie był izolowany oraz aby: $|V|=20$, $|E|<30$. Zaproponuj N oraz następujące funkcje krawędzi ze zbioru H : funkcję przepustowości 'c' (rozumianą jako maksymalną liczbę bitów, którą można wprowadzić do kanału komunikacyjnego w ciągu sekundy), oraz funkcję przepływu 'a' (rozumianą jako faktyczną liczbę pakietów, które wprowadza się do kanału komunikacyjnego w ciągu sekundy). Pamiętaj aby funkcja przepływu realizowała macierz N oraz aby dla każdego kanału 'e' zachodziło: $c(e) > a(e)$.
 - Niech miarą niezawodności sieci jest prawdopodobieństwo tego, że w dowolnym przedziale czasowym, nierozspójniona sieć zachowuje $T < T_{\max}$, gdzie: $T = 1/G * \sum_e (a(e)/(c(e)/m - a(e)))$, jest średnim opóźnieniem pakietu w sieci, \sum_e oznacza sumowanie po wszystkich krawędziach 'e' ze zbioru E , 'G' jest sumą wszystkich elementów macierzy natężeń, a 'm' jest średnią wielkością pakietu w bitach. Napisz program szacujący niezawodność takiej sieci przyjmując, że prawdopodobieństwo nieuszkodzenia każdej krawędzi w dowolnym interwale jest równe 'p'. Uwaga: 'N', 'p', 'T_max' oraz topologia wyjściowa sieci są parametrami.
 - Przy ustalonej strukturze topologicznej sieci i dobranych przepustowościach stopniowo zwiększaj wartości w macierzy natężeń. Jak będzie zmieniać się niezawodność zdefiniowana tak jak punkcie poprzednim ($\Pr[T < T_{\max}]$).
 - Przy ustalonej macierzy natężeń i strukturze topologicznej stopniowo zwiększaj przepustowości. Jak będzie zmieniać się niezawodność zdefiniowana tak jak punkcie poprzednim ($\Pr[T < T_{\max}]$).
 - Przy ustalonej macierzy natężeń i pewnej początkowej strukturze topologicznej, stopniowo zmieniaj topologię poprzez dodawanie nowych krawędzi o przepustowościach będących wartościami średnimi dla sieci początkowej. Jak będzie zmieniać się niezawodność zdefiniowana tak jak punkcie poprzednim ($\Pr[T < T_{\max}]$).

Napisz sprawozdanie zawierające opis zrealizowanych programów, komentarz do przeprowadzonych badań oraz wnioski.

1.2 Proponowana topologia sieci



Rysunek 1- wygenerowany za pomocą strony csacademy.com

Proponowany graf spełnia warunku zadania. W grafie mamy 20 wierzchołków oraz 25 krawędzi, co więcej żaden wierzchołek nie jest izolowany. Każdy wierzchołek w grafie jest co najwyżej 3 stopnia. Początkowo chciałem się zdecydować na symetryczne połączenie 2 grafów kołowych o 10 wierzchołkach, jednak nie spełniał on warunku $|E| < 30$, więc zmieniłem koncepcję o połączenie 2 grafów kołowych jednego o 15 wierzchołkach, a drugiego o 5 i połączenie obu grafów. Finalnie na 2 wymiarowej płaszczyźnie moja propozycja przedstawia się jak połączenie 5 sześciokątów z użyciem wspólnych krawędzi.

1.3 Początkowe warunki zadania

Chciałem uniknąć symetryczności macierzy N więc zdecydowałem się wypełniać ją losowo. Macierz A , odpowiedzialną za przedstawienie faktycznej liczby pakietów wprowadzanych do danego kanału w ciągu sekundy, stworzyłem korzystając z macierzy N oraz algorytmu Dijkstry do znajdowania najkrótszych ścieżek w grafie. Funkcję przepustowości C określiłem dla każdej krawędzi jako sumę poszczególnych wartości w macierzy A oraz przemnożenie jej przez rozmiar pakietu (przyjętą jako 1024 bity). Dodatkowo dla każdej krawędzi dodałem losową rezerwę z przedziału (1000, 2000) pakietów. Początkowo za T_Max ustawiłem 0.05 sekundy. Domyślnie, dla każdej krawędzi ustawiłem prawdopodobieństwo nieuszkodzenia na 0.95.

```

102 private static void generateN() {
103     for (int i = 0; i < 20; i++) {
104         for (int j = 0; j < 20; j++) {
105             if (i == j) {
106                 N[i][j] = 0;
107             }
108             else {
109                 N[i][j] = generator.nextInt(100);
110             }
111         }
112     }
113 }

```

Rysunek 2 - procedura generująca natężenia

```

243 private static void generateA(SimpleWeightedGraph<Integer, DefaultWeightedEdge> net) {
244     restartA();
245     GraphPath<Integer, DefaultWeightedEdge> path;
246     ArrayList<Integer> nodes;
247     int val;
248     for (int i = 1; i <= 20; i++) {
249         for (int j = 1; j <= 20; j++) {
250             if (i == j) continue;
251             path = DijkstraShortestPath.findPathBetween(net, i, j);
252             nodes = (ArrayList<Integer>) path.getVertexList();
253             val = N[i - 1][j - 1];
254             for (int n = 0; n < nodes.size() - 1; n++) {
255                 A[nodes.get(n) - 1][nodes.get(n + 1) - 1] += val;
256             }
257         }
258     }
259 }
260 }

```

Rysunek 3 - metoda generująca A

```

162 private static void generateC() {
163     for (int i = 0; i < 20; i++) {
164         for (int j = i; j < 20; j++) {
165             if (i == j) {
166                 C[i][j] = 0;
167             } else {
168                 Random rand = new Random();
169                 C[i][j] = (A[i][j] + A[j][i] + 1000 + rand.nextInt(1000)) * packetSize;
170                 C[j][i] = C[i][j];
171             }
172         }
173     }
174 }
175 }

```

Rysunek 4 - tworzenie funkcji przepustowości

1.4 Testy niezawodności

Testy przeprowadzałem w następujący sposób. Początkowo dla każdej krawędzi sprawdzam czy nie została uszkodzona. Jeśli któraś została uszkodzona to usuwam ją z grafu. Następnie sprawdzam czy graf pozostaje spójny. Jeżeli tak, to obliczam opóźnienie pakietu. Dopiero jeśli opóźnienie t wyliczane ze wzoru $1/G \sum_e (A(e)/(C(e)/m - A(e)))$ spełnia warunek $t < T_{Max}$ próbę zaliczam do udanych. Dla mojej topologii i przeprowadzonych testach w liczbie 10 000 średnia niezawodność wyszła mi około 85%.

```
Matrix N :
000000 000044 000085 000001 000080 000055 000090 000079 000015 000049 000079 000000 000088 000062 000087 000015 000000 000046 000016 000091
000079 000000 000098 000029 000050 000051 000078 000085 000070 000033 000001 000030 000038 000034 000012 000074 000050 000002 000094 000026
000031 000098 000000 000026 000053 000085 000008 000083 000088 000099 000065 000080 000072 000085 000071 000075 000068 000058 000065 000077
000065 000046 000041 000000 000037 000004 000007 000037 000006 000007 000039 000029 000011 000084 000068 000036 000028 000043 000016 000057
000023 000003 000086 000027 000000 000098 000069 000082 000094 000081 000082 000034 000057 000005 000022 000017 000049 000034 000045 000048
000034 000038 000078 000089 000089 000000 000018 000007 000007 000014 000022 000029 000097 000005 000046 000021 000066 000029 000008 000055
000070 000089 000018 000027 000069 000015 000000 000061 000002 000028 000013 000056 000095 000001 000086 000026 000072 000019 000047 000031
000015 000071 000072 000002 000099 000043 000063 000000 000000 000096 000089 000088 000073 000090 000077 000026 000008 000011 000095 000094
000073 000019 000043 000050 000076 000030 000095 000049 000000 000037 000085 000047 000021 000010 000097 000076 000026 000004 000085 000099
000027 000002 000067 000003 000071 000038 000042 000094 000011 000000 000018 000087 000036 000043 000063 000047 000033 000058 000060 000009
000097 000054 000085 000053 000027 000025 000003 000016 000067 000063 000000 000029 000035 000097 000093 000062 000015 000070 000096 000058
000055 000090 000047 000020 000071 000077 000093 000036 000060 000016 000095 000000 000021 000016 000067 000098 000030 000016 000044 000018
000015 000053 000099 000024 000081 000064 000021 000094 000066 000006 000072 000088 000000 000038 000057 000041 000042 000041 000029 000081
000025 000067 000095 000010 000060 000004 000048 000090 000065 000021 000077 000046 000020 000000 000084 000067 000064 000068 000059 000082
000071 000000 000048 000036 000020 000014 000057 000080 000059 000032 000067 000020 000016 000091 000000 000098 000052 000040 000045 000011
000056 000004 000014 000054 000026 000008 000048 000083 000014 000022 000004 000059 000031 000070 000001 000000 000045 000068 000019 000044
000092 000059 000089 000081 000069 000059 000014 000077 000067 000078 000078 000003 000033 000075 000010 000000 000084 000098 000076
000056 000065 000099 000078 000068 000035 000001 000027 000052 000074 000070 000079 000075 000015 000012 000050 000050 000000 000043 000046
000081 000007 000021 000032 000075 000048 000045 000046 000051 000097 000020 000052 000044 000058 000040 000001 000091 000083 000000 000076
000062 000062 000093 000025 000071 000098 000006 000036 000041 000004 000034 000025 000063 000018 000098 000009 000086 000073 000061 000000
```

Rysunek 5 - wyjściowa macierz N

```
Matrix C :
000000 5860352 1412096 1213440 1935360 1641472 1787904 1961984 1385472 1739776 1183744 1070080 2027520 1529856 5497856 9849856 1658880 1235968 1402880 1111040
5860352 000000 4486144 1867776 1363968 1855488 1525760 1316864 1241088 1028096 1177600 1958912 1289216 1041408 1113088 1713152 1031168 1590272 1782784 1449984
1412096 4486144 000000 5295104 1879040 1340416 1380352 1777664 1739776 1266688 1400832 1569792 1112064 1264640 1632256 1812480 1441792 1157120 1719296 1172480
1213440 1867776 5295104 000000 5719040 1782784 1241088 1800192 1136640 1131520 1735680 1157120 1727488 1815552 1436672 1103872 8396800 1042432 1454080 1413120
1935360 1363968 1879040 5719040 000000 5072896 1635328 1641472 1709056 1462272 1024000 1802240 1516544 1606656 1429504 1510400 2034688 1163264 1812480 1937408
1641472 1855488 1340416 1782784 5072896 000000 4779008 1455104 1350656 1666048 1117184 1854464 1477632 1152000 1411072 1585152 1173504 1549312 1746944 1107968
1787904 1525760 1380352 1241088 1635328 4779008 000000 5960704 1519616 1509376 1216512 1741824 1944576 1229824 1911808 1751040 1733632 9240576 1276928 1750016
1961984 1316864 1777664 1800192 1641472 1455104 5960704 000000 4708352 1552384 1533952 1496064 1168384 1959936 1195008 2034688 1063936 1960960 1846272 1302528
1385472 1241088 1739776 1136640 1709056 1350656 1519616 4708352 000000 5821440 1174528 1871872 1187840 1031168 1753088 1661952 1614848 1937408 1646592 1201152
1739776 1028096 1266688 1131520 1462272 1666048 1509376 1552384 5821440 000000 5248000 1455184 1408000 1067008 2029568 1817600 1923072 1890304 8426496 1538048
1183744 1177600 1400832 1735680 1024000 1117184 1216512 1533952 1174528 5248000 000000 4545536 1316864 1649664 1440768 1936384 1750016 1647616 1671168 1368064
1070080 1958912 1569792 1157120 1802240 1854464 1741824 1496064 1871872 1455104 4545536 000000 4999168 1982464 1105920 1516544 1536000 1667072 1280000 1155072
2027520 1289216 1112064 1727488 1516544 1477632 1944576 1168384 1187840 1408000 1316864 4999168 000000 5599232 1861632 1961984 1925120 1654784 1125376 8943616
1529856 1041408 1264640 1815552 1606656 1152000 1229824 1959936 1031168 1067008 1649664 1982464 5599232 000000 5220352 1746944 1786880 1666048 1328128 1064960
5497856 1113088 1632256 1436672 1429504 1411072 1911808 1195008 1753088 2029568 1440768 1105920 1861632 5220352 000000 1388544 1085440 2006016 1976320 1112064
9849856 1713152 1812480 1103872 1510400 1585152 1751040 2034688 1661952 1817600 1936384 1516544 1961984 1746944 1388544 000000 7869440 1968128 1511424 8549376
1658880 1031168 1441792 8396800 2034688 1173504 1733632 1063936 1614848 1923072 1750016 1536000 1925120 1786880 1085440 7869440 000000 9459712 1270784 1657856
1235968 1590272 1157120 1042432 1163264 1549312 9240576 1960960 1937408 1890304 1647616 1667072 1654784 1666048 2006016 1968128 9459712 000000 8898560 1557504
1402880 1782784 1719296 1454080 1812480 1746944 1276928 1846272 1646592 8426496 1671168 1280000 1125376 1328128 1976320 1511424 1270784 8898560 000000 7861248
1111040 1449984 1172480 1413120 1937408 1107968 1750016 1302528 1201152 1538048 1368064 1155072 8943616 1064960 1112064 8549376 1657856 1557504 7861248 000000
```

Rysunek 6 - funkcja przepustowości

```
MyGraph average delay: 9.872678780430961E-4 8.384866088220015/8493
MyGraph reliability: 0.8493 8493.0/10000
```

Średnie opóźnienie pakietu liczyłem dla prób w których graf przeszedł test spójności.

2. Badania

2.1. Stopniowe zwiększanie wartości w macierzy natężeń

Zadanie polegało na sprawdzaniu niezawodności sieci przy zwiększaniu wartości macierzy natężeń. Innymi słowy, jak zachowywałaby się sieć jeśli poszczególne wierzchołki chciałyby nadawać więcej pakietów do siebie. Przeprowadziłem 5 prób i w każdej zwiększałem każdą z wartości w macierzy o $nDiff = 5$. Poniżej wyniki symulacji, które zgodne są z oczekiwaniami. Sieć będzie chodziła gorzej przy wzmożonej aktywności wierzchołków.

```
MyGraph v.0 average delay: 0.0010050517051204571    0.8955010692623272/891
MyGraph v.0 reliability: 0.891    891.0/1000
```

```
MyGraph v.0 average delay: 0.0013944218079918267    1.189441802217028/853
MyGraph v.0 reliability: 0.853    853.0/1000
```

```
MyGraph v.0 average delay: 0.0017670558357962968    1.4083435011296486/797
MyGraph v.0 reliability: 0.797    797.0/1000
```

```
MyGraph v.0 average delay: 0.0018021363144457916    1.2867253285142952/714
MyGraph v.0 reliability: 0.714    714.0/1000
```

```
MyGraph v.0 average delay: 0.002132129955087331    1.2046534246243419/565
MyGraph v.0 reliability: 0.565    565.0/1000
```

```
MyGraph v.0 average delay: 0.0020742218739211856    0.9499936182559031/458
MyGraph v.0 reliability: 0.458    458.0/1000
```

2.2 Stopniowe zwiększanie przepustowości węzłów

Zadanie polegało na sprawdzaniu niezawodności przy zwiększaniu funkcji przepustowości. Na potrzeby tego eksperymentu zmieniłem wyjściową funkcję przepustowości na $c(e) = a(e) + 100 + \text{rand}(100)$, a po każdej z 5 prób zwiększałem przepustowość każdej krawędzi o 100 pakietów. Widać tutaj ewidentnie jak duże znaczenie ma przepustowość krawędzi.

```
MyGraph v.0 average delay: 0.0019769143420429363    1.4332628979811288/725
MyGraph v.0 reliability: 0.725    725.0/1000
OVERLOAD - 242
COHESIVE - 33
```

```
MyGraph v.0 average delay: 0.0015902086246771488    1.16562292188835/733
MyGraph v.0 reliability: 0.733    733.0/1000
OVERLOAD - 237
COHESIVE - 29
```

```
MyGraph v.0 average delay: 0.0016158345124263212    1.2167233878570198/753
MyGraph v.0 reliability: 0.753    753.0/1000
OVERLOAD - 206
COHESIVE - 41
```

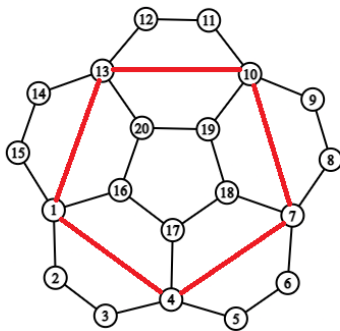
```
MyGraph v.0 average delay: 0.0016143368484858342    1.3398995842432424/830
MyGraph v.0 reliability: 0.83    830.0/1000
OVERLOAD - 131
COHESIVE - 39
```

```
MyGraph v.0 average delay: 0.0014337208120227373    1.2215301318433722/852
MyGraph v.0 reliability: 0.852    852.0/1000
OVERLOAD - 106
COHESIVE - 41
```

2.3 Stopniowe zwiększanie liczby krawędzi

W tym podpunkcie zaproponowałem 2 nowe topologie sieci bazujące na moim wyjściowym grafie. W pierwszej zaproponowałem 5 nowych krawędzi łączących 2 krańce zewnętrznych sześciokątów. Planowo miała to być pewnego rodzaju „autostrada” dla pakietów. W drugiej propozycji dodałem 10 nowych krawędzi łączącą w zewnętrznych sześciokątach wierzchołki o stopniu 2 z wierzchołkami wewnętrznego pięciokąta.

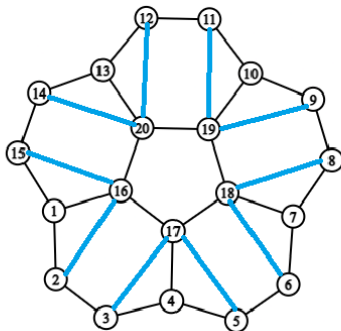
1) Pierwsza propozycja



```
MyGraph v.0 average delay: 0.0017469044244952609    1.0900683608850428/624
MyGraph v.0 reliability: 0.624    624.0/1000
```

```
MyGraph v.5 average delay: 0.001087423890439875    0.9036492529555361/831
MyGraph v.5 reliability: 0.831    831.0/1000
```

2) Druga propozycja



```
MyGraph v.10 average delay: 0.0010527080111321615 0.9284884658185664/882
MyGraph v.10 reliability: 0.882 882.0/1000
```

Testy wykonałem dla przepustowości takiej jak w poprzednim podpunkcie, a dla nowych krawędzi, zgodnie z poleceniem, ustawiłem średnie przepustowości. Z powyższych testów wynika, że zwiększanie krawędzi ma pozytywny wpływ na niezawodność. Pytanie które musimy sobie zadać, to czy taka poprawa niezawodności wynikająca z dodatkowych krawędzi jest opłacalna finansowo?

2.4 Własny test – sprawdzenie wpływu maksymalnego czasu oraz niezawodności krawędzi

Interesowało mnie co ma większy wpływ na powodzenie eksperymentu, czy jest to ograniczenie czasowe czy jednak przepustowości, dlatego sprawdziłem moją topologię dla różnych przedziałów czasowych. Oto efekt.

```
T_MAX = 0.05

MyGraph v.0 average delay: 0.0034597139602845104 2.411420630318304/697
MyGraph v.0 reliability: 0.697 697.0/1000
OVERLOAD - 259


T_MAX = 0.01

MyGraph v.0 average delay: 0.002025811577734543 1.2560031781954166/620
MyGraph v.0 reliability: 0.62 620.0/1000
OVERLOAD - 284


T_MAX = 0.005

MyGraph v.0 average delay: 0.0015553513816259671 0.8865502875268012/570
MyGraph v.0 reliability: 0.57 570.0/1000
OVERLOAD - 271
```

Jak widzimy większość nieudanych prób wynikało z przeładowania (overload) ścieżki.

Kolejną sprawą jest wpływ niezawodności krawędzi. Poniżej wyniki testów dla różnych niezawodności.

```
MyGraph v.0 average delay: 0.0020601726370632106 1.4009173932029833/680
MyGraph v.0 reliability: 0.68 680.0/1000
OVERLOAD - 277
COHESIVE - 43
```

```
MyGraph v.0 average delay: 0.0022068478679976218    1.3086607857225898/593
MyGraph v.0 reliability: 0.593    593.0/1000
OVERLOAD - 338
COHESIVE - 69
```

Rysunek 8 niezawodność = 0.92

```
MyGraph v.0 average delay: 0.0023136379105202554    1.1545053173496074/499
MyGraph v.0 reliability: 0.499    499.0/1000
OVERLOAD - 405
COHESIVE - 96
```

Rysunek 9 niezawodność = 0.9

```
MyGraph v.0 average delay: 0.0030198310932887523    1.0297624028114645/341
MyGraph v.0 reliability: 0.341    341.0/1000
OVERLOAD - 471
COHESIVE - 187
```

Rysunek 10 niezawodność = 0.85

3. Podsumowanie

Po przekonanych przeze mnie testach z pewnością mogę stwierdzić, że przy okazji planowania sieci należy uwzględnić wiele kwestii. W projekcie powinniśmy zwrócić uwagę na nasze potrzeby i możliwości. Z pewnością mogę stwierdzić, że ważnymi aspektami będą przepustowość i niezawodność krawędzi, jak również odpowiednio dobrana topologia z wierzchołkami o jak największym stopniu, aby zmniejszyć ryzyko izolacji w przypadku uszkodzenia kanału komunikacyjnego.