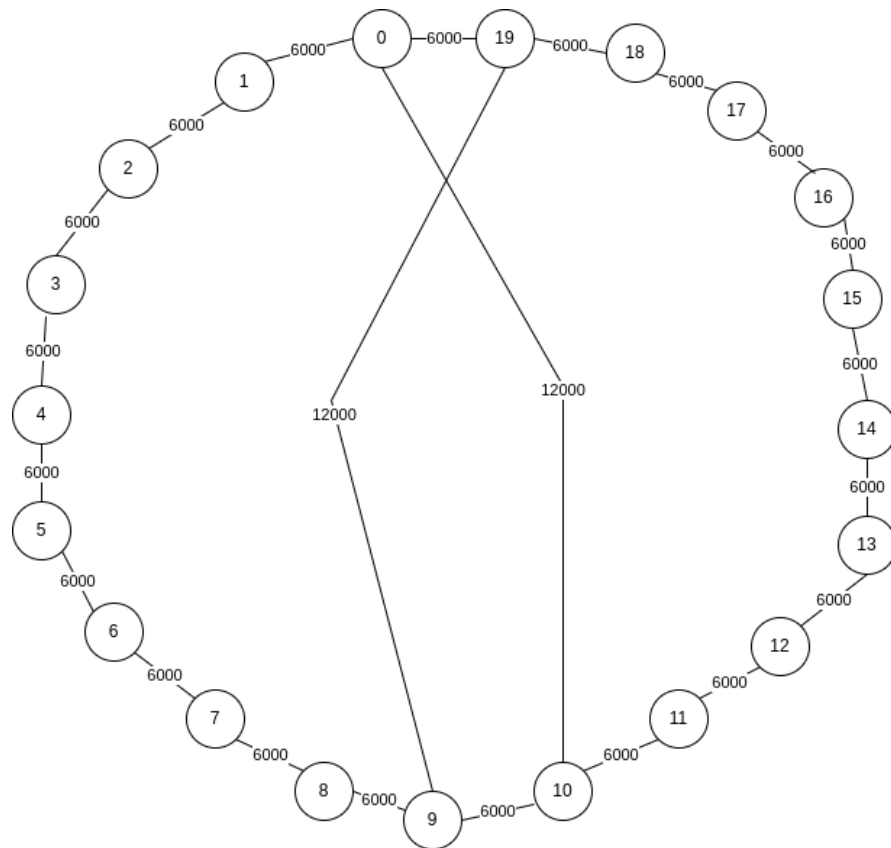
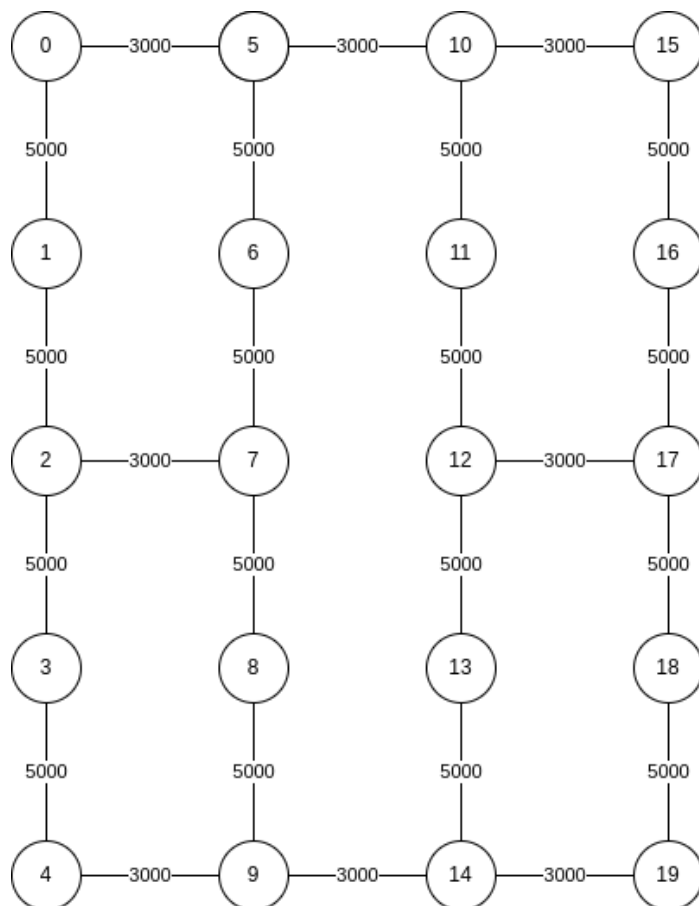


1. Użyte modele sieci.

Graf 1:



Graf 2:



Średni rozmiar pakietu m przyjęty został jako 10bitów.

Macierz nateżeń N jest generowana losowo tak, że N_{ij} gdzie $i=j$ równa się 0. Natomiast każdy pozostały element jest liczbą od 1 do 5. Daje to średnio 1140 pakietów przepływających przez całą sieć (11400 bitów).

Funkcja przepływu a jest generowana przez program tak, aby spełnić macierz N oraz nie przekroczyć przepustowości na każdej krawędzi.

2. Opis programu.

Program na wejściu oferuje opcje:

- z** – wybór podpunktu z listy zadań (1, 2, 3, 4)
- t** – ustawienie maksymalnego dopuszczalnego opóźnienia. Domyślnie 0.01.
- p** – ustawienie prawdopodobieństwa nieuszkodzenia krawędzi. Domyślnie 0.97.
- m** – ustawienie średniego rozmiaru pakietu. Domyślnie 10bitów.
- g** – wybór grafu 1 lub 2

Zadaniem programu jest wyznaczenie prawdopodobieństwa, że nierozspójniona sieć zachowa opóźnienie niższe niż ustalone wcześniej T_{max} , czyli niezawodność sieci. Najważniejszą rolę odgrywa w tym funkcja ***generate_a***, która generuje funkcje przepływu oraz funkcja ***test_reliability***, która wielokrotnie wywołuje *generate_a* i zlicza otrzymane wyniki w ten sposób aproksymując prawdopodobieństwo danego scenariusza.

Działanie *generate_a*:

1. Odłączanie krawędzi w grafie z ustalonym wcześniej prawdopodobieństwem.
2. Pakiet networkx sprawdza czy otrzymany graf jest spójny. Jeżeli nie punkt 1 jest powtarzany, a każde rozspójnienie jest zliczane do końcowych statystyk.
3. Wysyłanie pojedynczo pakietów z wierzchołka I do wierzchołka j najkrótszą ścieżką (dijkstra path), aż przesłane zostaną wszystkie pakiety określone w *macierzy N* oraz na żadnej krawędzi nie zostanie przekroczony limit przepustowości.
4. Jeżeli z powodzeniem przesłano wszystkie pakiety obliczone jest średnie opóźnienie w sieci.

Możliwe wyniki:

- **rozspójnienie grafu** – nie wpływa na % niezawodności ale również wiele nam mówi o tym czy topologia sieci jest dobra.
- **przekroczenie przepustowości** – obniża niezawodność. Występuje gdy nie było możliwe znalezienie ścieżki, z jednego wierzchołka do drugiego tak aby na każdej krawędzi przepływ był mniejszy niż przepustowość.
- **przekroczone maksymalne opóźnienie** – obniża niezawodność. Obliczone średnie opóźnienie okazało się większe niż ustalone maksymalne.
- **przesłano wszystkie pakiety oraz $T < T_{max}$**

3. Przykładowe zbadanie niezawodności.

Graf 1:

```
/home/sebastian/PycharmProjects/TSL2/venv/bin/python /home/sebastian/PycharmProjects/TSL2/main.py -z 1 -g 1 -p 0.95 -t 0.1
Niezawodność 95.8% (1.7% przekroczone opóźnienie, 2.5% przekroczone przepustowość), Częstotliwość rozspójnień - 14.24%
```

Dla ustalonego prawdopodobieństwa nieuszkodzenia sieci 0.95 oraz $T_{\max} = 0.1$ **niezawodność sieci wyniosła 95.8%**. Natomiast, aż 14.24% grafów zostało rozspójnionych. Co w rzeczywistości daje tylko **82,15%** przypadków, w których $T < T_{\max}$ oraz $a < c$.

Graf 2:

```
/home/sebastian/PycharmProjects/TSL2/venv/bin/python /home/sebastian/PycharmProjects/TSL2/main.py -z 1 -g 2 -p 0.95 -t 0.1
Niezawodność 85.5% (0.1% przekroczone opóźnienie, 14.4% przekroczone przepustowość), Częstotliwość rozspójnień - 4.49%
```

Dla tych samych parametrów **niezawodność sieci numer 2 wyniosła 85.5%**. Z otrzymanego wyniku można odczytać, że problemem nie jest duże opóźnienie tylko za mała przepustowość. Wystąpiło dużo mniej rozspójnień niż w przypadku *Grafu 1* co świadczy o tym, że sieć numer 2 jest lepiej skonstruowana. $T < T_{\max}$ zostało rzeczywiście spełnione w **81,66%** przypadków.

4. Zwiększanie wartości w macierzy natężeń.

Graf 1:

```
/home/sebastian/PycharmProjects/TSL2/venv/bin/python /home/sebastian/PycharmProjects/TSL2/main.py -z 2 -g 1 -p 0.9 -t 0.05
Podaj stałą o którą zwiększane będą wartości N: 1
Podaj max ilość powtórzeń: 5
Początkowa niezawodność 89.5% (9.95% przekroczone opóźnienie, 0.55% przekroczone przepustowość), Częstotliwość rozspójnień - 41.02%
N zwiększone o 1 - Niezawodność 28.35% (9.4% przekroczone opóźnienie, 62.25% przekroczone przepustowość), Częstotliwość rozspójnień - 41.38%
N zwiększone o 2 - Niezawodność 16.65% (7.75% przekroczone opóźnienie, 75.6% przekroczone przepustowość), Częstotliwość rozspójnień - 41.31%
N zwiększone o 3 - Niezawodność 0.0% (0.0% przekroczone opóźnienie, 100.0% przekroczone przepustowość), Częstotliwość rozspójnień - 42.33%
```

Przyjęte parametry to $p=0.9$ oraz $T_{\max}=0.05$. Elementy macierzy natężeń były zwiększane każdy o 1 co daje 380 nowych pakietów w całej sieci (3800 bitów). Jak widać niezawodność drastycznie spada do 0%, ponieważ z tak dużym prawdopodobieństwem uszkodzenia krawędzi oraz duża ilość nowych pakietów **przekraczana zostaje maksymalna przepustowość, a także rośnie średnie opóźnienie**. Dodawanie pakietów do macierzy N, w **żaden sposób nie wpływa na częstotliwość rozspójnień grafu**. Jest to zależne wyłącznie od konstrukcji sieci oraz prawdopodobieństwa uszkodzenia krawędzi. Mała różnica między otrzymanymi wynikami (41 - 42 %) to tylko błąd estymacji.

Graf 2:

```
/home/sebastian/PycharmProjects/TSL2/venv/bin/python /home/sebastian/PycharmProjects/TSL2/main.py -z 2 -g 2 -p 0.95 -t 0.1
Podaj stałą o którą zwiększane będą wartości N: 1
Podaj max ilość powtórzeń: 4
Początkowa niezawodność 86.15% (0.1% przekroczone opóźnienie, 13.75% przekroczone przepustowość), Częstotliwość rozspójnień - 5.53%
N zwiększone o 1 - Niezawodność 72.55% (10.6% przekroczone opóźnienie, 16.85% przekroczone przepustowość), Częstotliwość rozspójnień - 4.03%
N zwiększone o 2 - Niezawodność 0.0% (0.0% przekroczone opóźnienie, 100.0% przekroczone przepustowość), Częstotliwość rozspójnień - 4.67%
```

Eksperyment przeprowadzony na drugim grafie pokazuje nam to samo, czyli że **rośnie zarówno opóźnienie jak i częściej jest przekraczana przepustowość**. Ostatecznie pakietów jest za dużo, żeby je wszystkie przesłać przez co niezawodność spada do 0%.

Podsumowanie:

Zwiększanie ilości pakietów przepływających przez sieć sprawia, że rośnie:

- średnie opóźnienie
 - przepływ (co prowadzi do przekroczenia dozwolonej przepustowości)
- Nie ma to żadnego wpływu na rozspójnianie się grafu.

5. Zwiększanie przepustowości

Graf 1:

```
/home/sebastian/PycharmProjects/TSL2/venv/bin/python /home/sebastian/PycharmProjects/TSL2/main.py -z 3 -g 1 -p 0.9 -t 0.05
Podaj stałą o którą zwiększana będzie przepustowosc: 200
Podaj ilość powtórzeń: 10
Początkowa niezawodność 73.05% (20.05% przekroczone opóźnienie, 6.9% przekroczone przepustowosc), Czesotliwość rozspójnień - 40.3%
Przepustowosc zwiększona o 200 bitów - Niezawodność 88.25% (10.8% przekroczone opóźnienie, 0.95% przekroczone przepustowosc), Czesotliwość rozspójnień - 41.67%
Przepustowosc zwiększona o 400 bitów - Niezawodność 94.6% (5.4% przekroczone opóźnienie, 0.0% przekroczone przepustowosc), Czesotliwość rozspójnień - 43.21%
Przepustowosc zwiększona o 600 bitów - Niezawodność 98.0% (2.0% przekroczone opóźnienie, 0.0% przekroczone przepustowosc), Czesotliwość rozspójnień - 40.86%
Przepustowosc zwiększona o 800 bitów - Niezawodność 98.8% (1.2% przekroczone opóźnienie, 0.0% przekroczone przepustowosc), Czesotliwość rozspójnień - 41.19%
Przepustowosc zwiększona o 1000 bitów - Niezawodność 99.8% (0.2% przekroczone opóźnienie, 0.0% przekroczone przepustowosc), Czesotliwość rozspójnień - 42.13%
Przepustowosc zwiększona o 1200 bitów - Niezawodność 100.0% (0.0% przekroczone opóźnienie, 0.0% przekroczone przepustowosc), Czesotliwość rozspójnień - 42.06%
```

Dla dobranych parametrów $p = 0.9$ oraz $T_{\max} = 0.05$ początkowa niezawodność wyniosła 73.85%. Następnie w każdym kroku przepustowość każdej z krawędzi była zwiększana o 200 bitów. Bez zaskoczenia **rozwiązało to całkowicie problem przekraczania przepustowości**, ale także **opóźnienie zaczęło maleć** dzięki czemu przy dodaniu 1200 bitów przepustowości do każdej krawędzi **niezawodność wzrosła do 100%**. Częstotliwość rozspójnień znowu pozostała niezmienna.

Graf 2:

```
/home/sebastian/PycharmProjects/TSL2/venv/bin/python /home/sebastian/PycharmProjects/TSL2/main.py -z 3 -g 2 -p 0.9 -t 0.015
Podaj stałą o którą zwiększana będzie przepustowosc: 200
Podaj ilość powtórzeń: 10
Początkowa niezawodność 38.05% (34.85% przekroczone opóźnienie, 27.1% przekroczone przepustowosc), Czesotliwość rozspójnień - 19.39%
Przepustowosc zwiększona o 200 bitów - Niezawodność 51.2% (20.35% przekroczone opóźnienie, 28.45% przekroczone przepustowosc), Czesotliwość rozspójnień - 18.6%
Przepustowosc zwiększona o 400 bitów - Niezawodność 59.15% (14.85% przekroczone opóźnienie, 26.0% przekroczone przepustowosc), Czesotliwość rozspójnień - 20.76%
Przepustowosc zwiększona o 600 bitów - Niezawodność 63.95% (9.7% przekroczone opóźnienie, 26.35% przekroczone przepustowosc), Czesotliwość rozspójnień - 20.79%
Przepustowosc zwiększona o 800 bitów - Niezawodność 66.65% (10.5% przekroczone opóźnienie, 22.85% przekroczone przepustowosc), Czesotliwość rozspójnień - 21.54%
Przepustowosc zwiększona o 1000 bitów - Niezawodność 67.25% (28.9% przekroczone opóźnienie, 3.85% przekroczone przepustowosc), Czesotliwość rozspójnień - 20.29%
Przepustowosc zwiększona o 1200 bitów - Niezawodność 69.3% (29.65% przekroczone opóźnienie, 1.05% przekroczone przepustowosc), Czesotliwość rozspójnień - 19.87%
Przepustowosc zwiększona o 1400 bitów - Niezawodność 70.65% (29.35% przekroczone opóźnienie, -0.0% przekroczone przepustowosc), Czesotliwość rozspójnień - 19.29%
Przepustowosc zwiększona o 1600 bitów - Niezawodność 70.1% (29.9% przekroczone opóźnienie, 0.0% przekroczone przepustowosc), Czesotliwość rozspójnień - 19.87%
Przepustowosc zwiększona o 1800 bitów - Niezawodność 72.2% (27.8% przekroczone opóźnienie, -0.0% przekroczone przepustowosc), Czesotliwość rozspójnień - 20.51%
Przepustowosc zwiększona o 2000 bitów - Niezawodność 75.05% (24.95% przekroczone opóźnienie, 0.0% przekroczone przepustowosc), Czesotliwość rozspójnień - 19.16%

/home/sebastian/PycharmProjects/TSL2/venv/bin/python /home/sebastian/PycharmProjects/TSL2/main.py -z 3 -g 2 -p 0.9 -t 0.015
Podaj stałą o którą zwiększana będzie przepustowosc: 1000
Podaj ilość powtórzeń: 5
Początkowa niezawodność 28.05% (43.55% przekroczone opóźnienie, 28.4% przekroczone przepustowosc), Czesotliwość rozspójnień - 18.53%
Przepustowosc zwiększona o 1000 bitów - Niezawodność 69.35% (7.55% przekroczone opóźnienie, 23.1% przekroczone przepustowosc), Czesotliwość rozspójnień - 20.06%
Przepustowosc zwiększona o 2000 bitów - Niezawodność 70.0% (30.0% przekroczone opóźnienie, 0.0% przekroczone przepustowosc), Czesotliwość rozspójnień - 20.19%
Przepustowosc zwiększona o 3000 bitów - Niezawodność 88.95% (11.05% przekroczone opóźnienie, -0.0% przekroczone przepustowosc), Czesotliwość rozspójnień - 19.9%
Przepustowosc zwiększona o 4000 bitów - Niezawodność 98.8% (1.2% przekroczone opóźnienie, 0.0% przekroczone przepustowosc), Czesotliwość rozspójnień - 19.48%
Przepustowosc zwiększona o 5000 bitów - Niezawodność 100.0% (0.0% przekroczone opóźnienie, 0.0% przekroczone przepustowosc), Czesotliwość rozspójnień - 19.71%
```

Wyniki dla grafu numer 2 potwierdzają wnioski wyciągnięte z eksperymentu na pierwszej sieci. Niezawodność po odpowiednim zwiększeniu przepustowości wzrosła do 100%

Podsumowanie:

Zwiększanie przepustowości zmniejsza:

- średnie opóźnienie
- częstotliwość przekraczania przepustowości.

Nie ma żadnego wpływu na rozspójnianie się grafu.

6. Dodawanie nowych krawędzi.

Graf 1:

```
/home/sebastian/PycharmProjects/TSL2/venv/bin/python /home/sebastian/PycharmProjects/TSL2/main.py -z 4 -g 1 -p 0.9 -t 0.03
Podaj ilość powtórzeń: 10
Początkowa niezawodność 58.55% (25.6% przekroczone opóźnienie, 15.85% przekroczone przepustowosc), Czesotliwość rozspójnień - 42.16%
- Niezawodność 71.85% (20.0% przekroczone opóźnienie, 8.15% przekroczone przepustowosc), Czesotliwość rozspójnień - 37.15%
- Niezawodność 81.25% (5.8% przekroczone opóźnienie, 12.95% przekroczone przepustowosc), Czesotliwość rozspójnień - 24.5%
- Niezawodność 81.85% (7.0% przekroczone opóźnienie, 11.15% przekroczone przepustowosc), Czesotliwość rozspójnień - 22.36%
- Niezawodność 86.5% (6.0% przekroczone opóźnienie, 7.5% przekroczone przepustowosc), Czesotliwość rozspójnień - 17.36%
- Niezawodność 91.35% (5.4% przekroczone opóźnienie, 3.25% przekroczone przepustowosc), Czesotliwość rozspójnień - 15.5%
- Niezawodność 92.15% (4.05% przekroczone opóźnienie, 3.8% przekroczone przepustowosc), Czesotliwość rozspójnień - 11.54%
- Niezawodność 94.75% (2.7% przekroczone opóźnienie, 2.55% przekroczone przepustowosc), Czesotliwość rozspójnień - 11.62%
- Niezawodność 97.35% (1.3% przekroczone opóźnienie, 1.35% przekroczone przepustowosc), Czesotliwość rozspójnień - 8.17%
- Niezawodność 97.95% (1.3% przekroczone opóźnienie, 0.75% przekroczone przepustowosc), Czesotliwość rozspójnień - 7.02%
- Niezawodność 99.35% (0.55% przekroczone opóźnienie, 0.1% przekroczone przepustowosc), Czesotliwość rozspójnień - 8.0%
```

Jak widać dodawanie nowych krawędzi sprawiło, że **zaczęło maleć opóźnienie, problem z przekraczaniem dozwolonej przepustowości również został rozwiązany** oraz **częstotliwość rozpójdania się grafu zmalała**.

Przy dodaniu odpowiedniej ilości nowych krawędzi otrzymujemy sieć ze 100% niezawodnością oraz 0% częstotliwością rozpójdnień.

Graf 2:

```
/home/sebastian/PycharmProjects/TSL2/venv/bin/python /home/sebastian/PycharmProjects/TSL2/main.py -z 4 -g 2 -p 0.9 -t 0.015
Podaj ilość powtórzeń: 15
Początkowa niezawodność 41.0% (30.25% przekroczone opóźnienie, 28.75% przekroczone przepustowosc), Czestotliwość rozpójdnień - 20.73%
- Niezawodność 49.15% (24.6% przekroczone opóźnienie, 26.25% przekroczone przepustowosc), Czestotliwość rozpójdnień - 16.42%
- Niezawodność 56.9% (18.3% przekroczone opóźnienie, 24.8% przekroczone przepustowosc), Czestotliwość rozpójdnień - 15.47%
- Niezawodność 60.95% (14.65% przekroczone opóźnienie, 24.4% przekroczone przepustowosc), Czestotliwość rozpójdnień - 14.68%
- Niezawodność 72.7% (16.1% przekroczone opóźnienie, 11.2% przekroczone przepustowosc), Czestotliwość rozpójdnień - 13.76%
- Niezawodność 81.0% (13.25% przekroczone opóźnienie, 5.75% przekroczone przepustowosc), Czestotliwość rozpójdnień - 11.43%
- Niezawodność 91.55% (5.4% przekroczone opóźnienie, 3.05% przekroczone przepustowosc), Czestotliwość rozpójdnień - 10.35%
- Niezawodność 93.95% (2.45% przekroczone opóźnienie, 3.6% przekroczone przepustowosc), Czestotliwość rozpójdnień - 9.3%
- Niezawodność 96.9% (2.5% przekroczone opóźnienie, 0.6% przekroczone przepustowosc), Czestotliwość rozpójdnień - 8.88%
- Niezawodność 98.55% (0.9% przekroczone opóźnienie, 0.55% przekroczone przepustowosc), Czestotliwość rozpójdnień - 7.88%
- Niezawodność 98.95% (0.85% przekroczone opóźnienie, 0.2% przekroczone przepustowosc), Czestotliwość rozpójdnień - 7.49%
- Niezawodność 99.45% (0.45% przekroczone opóźnienie, 0.1% przekroczone przepustowosc), Czestotliwość rozpójdnień - 6.59%
- Niezawodność 99.8% (0.15% przekroczone opóźnienie, 0.05% przekroczone przepustowosc), Czestotliwość rozpójdnień - 5.62%
- Niezawodność 99.95% (0.05% przekroczone opóźnienie, -0.0% przekroczone przepustowosc), Czestotliwość rozpójdnień - 6.89%
- Niezawodność 99.9% (0.05% przekroczone opóźnienie, 0.05% przekroczone przepustowosc), Czestotliwość rozpójdnień - 6.72%
- Niezawodność 100.0% (0.0% przekroczone opóźnienie, 0.0% przekroczone przepustowosc), Czestotliwość rozpójdnień - 6.24%
```

Ten sam efekt otrzymujemy dla grafu numer 2 jednak częstotliwość rozpójdnień spada wolniej niż w pierwszym przypadku.

Podsumowanie:

Dodawanie nowych krawędzi pozytywnie wpływa na wszystkie mierzone statystyki:

- maleje średnie opóźnienie
- niwelowany jest problem przekraczania przepustowości
- rzadziej dochodzi do rozpójdnienia sieci

Wnioski:

Bardzo ważne jest dobre skonstruowanie grafu, ponieważ mimo możliwości otrzymania wysokiej niezawodności będzie towarzyszył nam problem rozpójdniania się sieci przez co w rzeczywistości nie będzie ona tak niezawodna jak wyliczono. Drugim ważnym czynnikiem jest przepustowość krawędzi, ponieważ zbyt niska doprowadzi do braku możliwości przesłania wszystkich pakietów. Samo zwiększanie jej pomoże otrzymać 100% niezawodności dla nierozpójdniionych sieci. Lepšie jest jednak dodawanie nowych krawędzi dopóki nie otrzymamy prawie 100% gwarancji, że sieć zawsze będzie spójna jeżeli chcemy otrzymać prawdziwie niezawodną sieć.