

## Technologie Sieciowe, sprawozdanie nr 2

### 1. Opis programu

Program symuluje zachowanie i wydajność sieci internetowej. Modelem sieci jest graf nieskierowany, gdzie wierzchołkami są serwery, a krawędziami połączenia między nimi.

Każda krawędź ma 3 parametry:

- niezawodność - prawdopodobieństwo, że będzie dostępna w pewnym czasie  $t$
- przepustowość - maksymalna liczba b/s, którą obsługuje połączenie
- przepływ - obliczana w trakcie działania programu faktyczna liczba pakietów w danym połączeniu (obliczany na podstawie macierzy przepływu  $N$ )

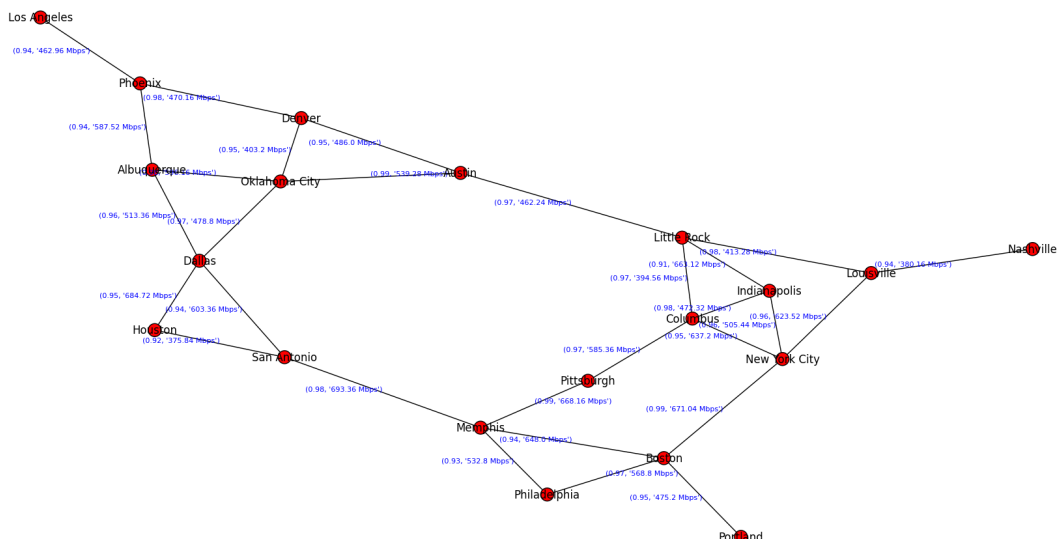
Program napisałem w języku python z wykorzystaniem biblioteki `igraph`.

### 2. Działanie programu

Program symuluje niezależne powtórzenia eksperymentu, a następnie zlicza stosunek sukcesów do wszystkich powtórzeń. Pojedynczy eksperyment polega na:

1. symulacji utraty połączeń n podstawie niezawodności (jeśli graf przestanie być spójny to eksperyment jest nieudany)
2. wyliczeniu przepływu na podstawie najkrótszych ścieżek dla każdej pary wierzchołków i porównanie go z przepustowością (jeśli dla jakiejś krawędzi przepływ przekracza przepustowość to eksperyment jest nieudany)
3. wyliczeniu opóźnienia według zadanego wzoru (jeśli jest większe od ustalonego max to eksperyment jest nieudany)

Jeżeli wszystkie powyższe punkty spełnione, dopiero wtedy eksperyment uznaje się za sukces.



### 3. Komentarz

Ze względu na brak narzuconych danych, postarałem się zasymulować sieć “rzeczywistą” a nie “doskonałą”.

#### 4. Wnioski

```
Niezawodność sieci: 77.40%
Niezawodność sieci dla wartości w N pomnożonych przez 1.1: 73.00%
Niezawodność sieci dla wartości w N pomnożonych przez 1.2: 60.80%
Niezawodność sieci dla wartości w N pomnożonych przez 1.3: 4.10%
Niezawodność sieci dla wartości w N pomnożonych przez 1.4: 0.10%
Niezawodność sieci dla wartości w N pomnożonych przez 1.5: 0.10%
Niezawodność sieci dla przepustowości przez 1.2: 82.80%
Niezawodność sieci dla przepustowości przez 1.4: 81.40%
Niezawodność sieci dla przepustowości przez 1.6: 83.40%
Niezawodność sieci dla przepustowości przez 1.8: 83.10%
Niezawodność sieci dla przepustowości przez 2.0: 82.00%
Niezawodność sieci dla 2 dodanych krawędzi: 94.10%
Niezawodność sieci dla 4 dodanych krawędzi: 95.30%
Niezawodność sieci dla 6 dodanych krawędzi: 94.40%
Niezawodność sieci dla 8 dodanych krawędzi: 93.90%
Niezawodność sieci dla 10 dodanych krawędzi: 95.10%
```

Ze względu na plan eksperymentu, niezawodność ma kolejne ograniczenia:

- niezawodność połączeń
- przepustowość połączeń
- opóźnienie

Ostatecznie górnym ograniczeniem jest najmniejsze z nich. Na przykład, zwiększanie przepustowości nic nie da, jeśli niezawodność połączeń jest niska i sieć nie zachowuje spójności. Tak samo, w 100% niezawodne połączenia nic nie działają, jeśli maksymalne opóźnienie jest zbyt wyśrubowane.

Na przykładzie powyższego wywołania widać, że:

- zwiększanie wartości N sprowadza niezawodność sieci do 0 (obniża ograniczenie górne)
- zwiększenie przepustowości (jako operacja odwrotna) zwiększa niezawodność sieci, ale do maksymalnie ok 83%. Oznacza to, że przepustowość przestaje być ograniczeniem górnym i napotykamy inny sufit (niezawodność połączeń lub opóźnienie)
- dodawanie połączeń do sieci oczywiście zwiększa niezawodność sieci, ale do poziomu 95% (ta sama obserwacja co wyżej). Dużo jednak zależy od przydatności nowych połączeń (czy na przykład mogą zastępować inne zawodne połączenia)

Warto zauważyć, że te wyniki mogą nie dawać pełnego obrazu sieci. Jedno wąskie gardło może sprowadzić niezawodność sieci do 0%. Prawdziwa sieć mogłaby też pracować optymalniej (ta implementacja szuka jednej najkrótszej ścieżki, nie zarządza zasobami na globalną skalę).