

# Sprawozdanie z laboratorium kursu „Technologie sieciowe”

## Lista 2 – „Model”

Bartosz Chodorowski <chomzee@ethernet.pl>  
Numer indeksu: 166142

Wrocław, 30 marca 2009



# 1 Cele

Celem zajęć laboratoryjnych było stworzenie oprogramowania, które umożliwi przeprowadzenie symulacji działania sieci komputerowych i oceny ich niezawodności.

## 2 Realizacja

Napisano dwa nieduże programy w języku C++. Skorzystano z biblioteki Boost, która udostępnia podstawowe algorytmy operujące na grafach.

### 2.1 Zasada działania

Pierwszy program został nazwany „Network Analyzer 1”, składa się z jednego pliku źródłowego – `na1.cpp`. Służy do szacowania niezawodności sieci na podstawie następującego modelu. Sieć reprezentowana jest przez spójny graf nieskierowany. Każdy wierzchołek symbolizuje maszynę działającą w sieci, krawędź natomiast połączenie pomiędzy hostami. Ponadto każdej krawędzi przypisana jest waga – liczba zmiennoprzecinkowa z przedziału  $[0, 1]$  oznaczająca niezawodność danego łącza.

Program wykonuje określoną liczbę symulacji. Każda z nich kończy się sukcesem (sieć przetrwała) lub porażką. Ilość porażek jest zliczana, po czym obliczana jest średnia niezawodność sieci według wzoru (1).

$$R_{avg} = \frac{n - f}{n} \cdot 100\% \quad (1)$$

Gdzie:

$R_{avg}$	–	średnia niezawodność sieci
$n$	–	liczba eksperymentów
$f$	–	liczba porażek

Pojedynczy eksperyment (symulację) można przedstawić za pomocą pseudokodu umieszczonego w listingu 1. Dla każdej krawędzi  $e$  zadanego grafu  $G$  losowana jest liczba  $t$  z przedziału  $[0, 1]$ . Jeśli  $t$  przekracza niezawodność krawędzi  $e$ , usuwana jest ona z grafu. Jeśli po rozpatrzeniu wszystkich krawędzi graf nadal jest spójny, zwracamy sukces, w przeciwnym wypadku porażkę.

Listing 1: Pseudokod pojedynczej symulacji w programie `na1`

```
1 funkcja symulacja(G)
2   dla każdej krawędzi e grafu G
3       wylosuj t z przedziału [0,1]
4       jeśli t > niezawodność(e)
5           usuń e z grafu G
6   jeśli graf G jest spójny
7       zwróć sukces
8   w przeciwnym razie
9       zwróć porażkę
```

„Network Analyzer 2” również składa się z jednego pliku – `na2.cpp`. Rozpatruje on nieco bardziej skomplikowany model sieci. Każdej krawędzi przyporządkowana jest, oprócz niezawodności, przepustowość (w jednostkach danych na sekundę). Program wczytuje również macierz natężeń  $N$ . Każdy element  $N_{i,j}$  tej macierzy wyraża liczbę jednostek danych, którą węzeł  $i$  chce wysłać do węzła  $j$  w każdej sekundzie. Na podstawie macierzy  $N$ , program oblicza dla każdej krawędzi przepływ, czyli faktyczną liczbę jednostek informacji, które przepływają przez tę krawędź w ciągu sekundy. Program liczy też średnie opóźnienie sieci dane wzorem (2).

$$d = \frac{1}{\sum_{i,j} N_{i,j}} \cdot \sum_{e \in E} \frac{a(e)}{c(e) - a(e)} \quad (2)$$

Gdzie:

- $d$  – średnie opóźnienie sieci
- $N$  – macierz natężeń
- $E$  – zbiór krawędzi rozpatrywanego graf
- $c(e)$  – przepustowość krawędzi  $e$
- $a(e)$  – przepływ na krawędzi  $e$

Pojedyncza symulacja zawodzi, gdy następuje jedna z poniższych sytuacji:

- sieć rozpódnia się ze względu na zawodność łącz,
- istnieje krawędź, dla której obliczony przepływ jest nie mniejszy niż jej przepustowość,
- obliczone średnie opóźnienie sieci jest większe niż założona z góry maksymalna wartość opóźnienia  $MaxT$ .

Pseudokod działania symulacji w tym modelu sieci przedstawia listing 2. Podobnie jak w przypadku programu `na1`, program `na2` wykonuje określoną ilość eksperymentów. Zlicza niepowodzenia (dzieląc je na niepowodzenia wynikające z rozpódnienia, przeciążenia lub przekroczenia czasu) oraz oblicza średnią niezawodność sieci daną wzorem (1).

Listing 2: Pseudokod pojedynczej symulacji w programie `na2`

```

1 funkcja symulacja(G, N, MaxT)
2   dla każdej krawędzi e grafu G
3     wylosuj t z przedziału [0,1]
4     jeśli t > niezawodność(e)
5       usuń e z grafu G
6   jeśli graf G nie jest spójny
7     zwróć porażkę
8
9   oblicz przepływ na każdej krawędzi grafu G na podstawie macierzy N
10  dla każdej krawędzi e grafu G
11    jeśli przepływ(e) >= przepustowość(e)
12      zwróć porażkę
13
14  oblicz średnie opóźnienie sieci d
15  jeśli d > MaxT
16    zwróć porażkę
17
18  zwróć sukces

```

W linii 9 listingu 2 obliczany jest przepływ na każdej krawędzi grafu  $G$ . Jeśli macierz natężeń wymusza transmisję danych między wierzchołkami nie połączonymi bezpośrednio, dane routowane są najkrótszą ścieżką łączącą zadane wierzchołki (bez względu na przepustowości poszczególnych łącz).

## 2.2 Przykładowe uruchomienia

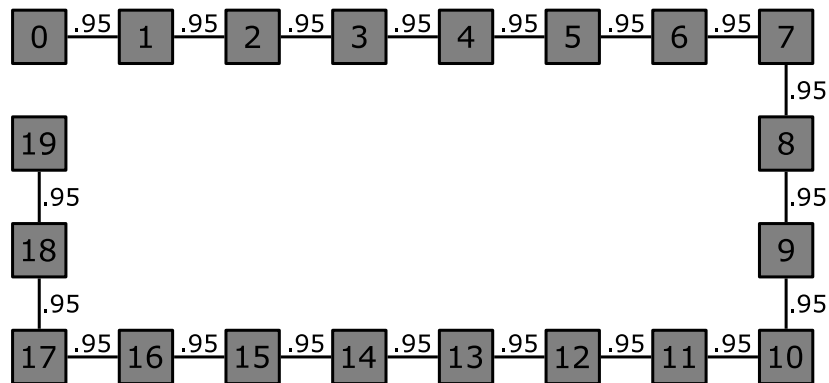
Kompilacji programów dokonuje się poleceniem `make`.

Listing 3: Kompilacja programów

```
1 $ make
2 g++ -O2 -o na1 na1.cpp
3 g++ -O2 -o na2 na2.cpp
4 $
```

Sieć przedstawioną na rysunku 1 opisuje plik `input/network1a` zawarty w listingu 4.

Rysunek 1: Sieć opisana w pliku `input/network1a`



Listing 4: Zawartość pliku `input/network1a`

```
1 100000
2 20
3 19
4 0      1      0.95
5 1      2      0.95
6 2      3      0.95
7 3      4      0.95
8 4      5      0.95
9 5      6      0.95
10 6     7      0.95
11 7     8      0.95
12 8     9      0.95
13 9     10     0.95
14 10    11     0.95
15 11    12     0.95
16 12    13     0.95
17 13    14     0.95
18 14    15     0.95
19 15    16     0.95
20 16    17     0.95
21 17    18     0.95
22 18    19     0.95
```

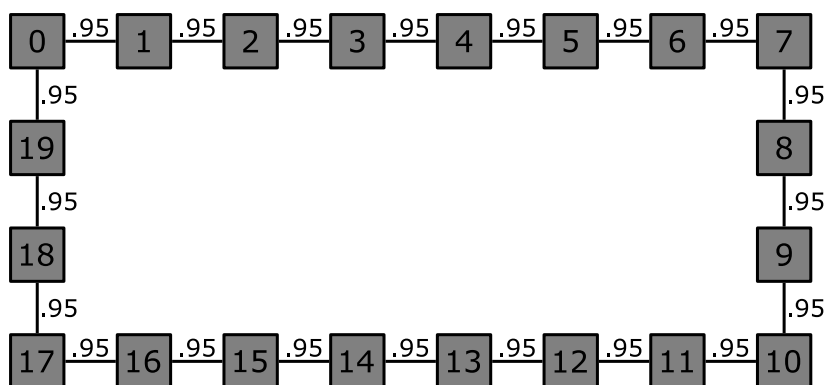
Listing 4 prezentuje format danych, jakie przyjmuje program `na1`. W pierwszej linii definiujemy liczbę eksperymentów, w drugiej liczbę wierzchołków, w trzeciej liczbę krawędzi. Następnie każda kolejna linia definiuje jedną krawędź. Dwie pierwsze liczby to numery wierzchołków, które krawędź łączy (numerując od zera). Trzecią wartością jest liczba zmiennoprzecinkowa z przedziału  $[0, 1]$  określająca niezawodność łącza.

Listing 5: Wynik działania programu `na1` dla danych z pliku `input/network1a`

```
1 $ ./na1 < input/network1a
2 Simulations: 100000
3 Failures: 62117
4 Average reliability: 37.883%
5 $
```

Przykładowy wynik działania programu `na1` prezentuje listing 5. Wypisuje on na wyjście łączną ilość przeprowadzonych symulacji, ilość nieudanych eksperymentów oraz średnią niezawodność sieci wyrażoną w procentach.

Rysunek 2: Sieć opisana w pliku `input/network1b`



Listing 6: Zawartość pliku `input/network1b`

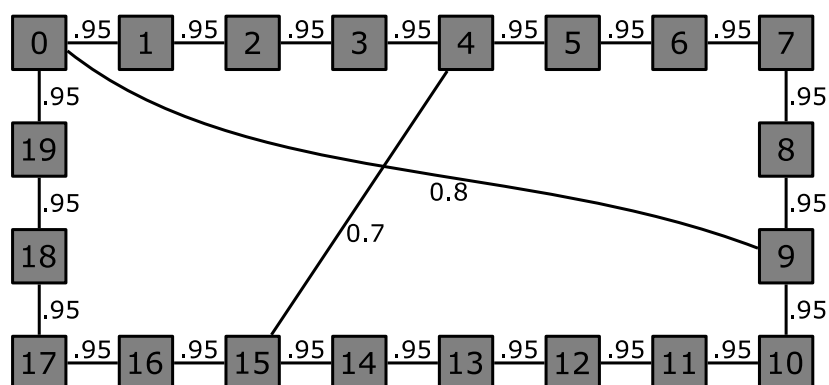
```
1 100000
2 20
3 20
4 0      1      0.95
5 1      2      0.95
6 2      3      0.95
7 3      4      0.95
8 4      5      0.95
9 5      6      0.95
10 6     7      0.95
11 7     8      0.95
12 8     9      0.95
13 9     10     0.95
14 10    11     0.95
15 11    12     0.95
16 12    13     0.95
17 13    14     0.95
18 14    15     0.95
19 15    16     0.95
20 16    17     0.95
21 17    18     0.95
22 18    19     0.95
23 19    0      0.95
```

Listing 7: Wynik działania programu na1 dla danych z pliku input/network1b

```
1 $ ./na1 < input/network1b
2 Simulations: 100000
3 Failures: 26515
4 Average reliability: 73.485%
5 $
```

Sieć przedstawioną na rysunku 2 opisuje plik input/network1b zawarty w listingu 6. Wynik działania programu na1 dla tego pliku zawiera listing 7. Dodanie jednej krawędzi znacząco (dwukrotnie) zwiększyło niezawodność sieci.

Rysunek 3: Sieć opisana w pliku input/network1c



Listing 8: Zawartość pliku input/network1c

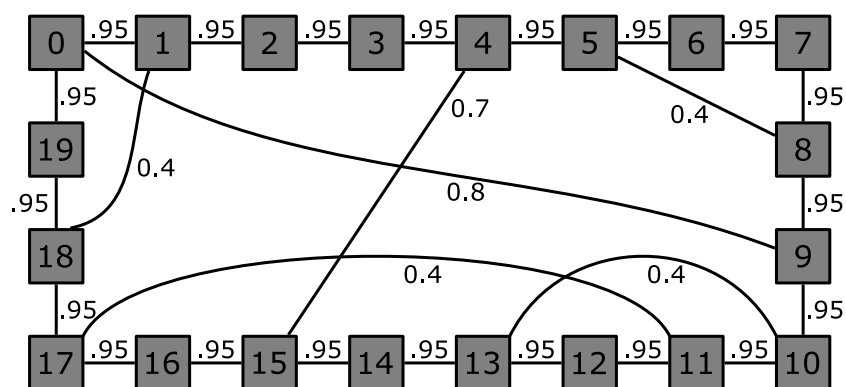
```
1 100000
2 20
3 22
4 0      1      0.95
5 1      2      0.95
6 2      3      0.95
7 3      4      0.95
8 4      5      0.95
9 5      6      0.95
10 6     7      0.95
11 7     8      0.95
12 8     9      0.95
13 9     10     0.95
14 10    11     0.95
15 11    12     0.95
16 12    13     0.95
17 13    14     0.95
18 14    15     0.95
19 15    16     0.95
20 16    17     0.95
21 17    18     0.95
22 18    19     0.95
23 19    0      0.95
24 0      9      0.8
25 4      15     0.7
```

Listing 9: Wynik działania programu na1 dla danych z pliku input/network1c

```
1 $ ./na1 < input/network1c
2 Simulations: 100000
3 Failures: 12850
4 Average reliability: 87.15%
5 $
```

Sieć przedstawioną na rysunku 3 opisuje plik input/network1c zawarty w listingu 8. Wynik działania programu na1 dla tego pliku zawiera listing 9. Dodanie kolejnych dwóch, bardziej zawodnych krawędzi wpłynęło korzystnie na niezawodność sieci.

Rysunek 4: Sieć opisana w pliku input/network1d



Listing 10: Zawartość pliku input/network1d

```
1 100000
2 20
3 26
4 0      1      0.95
5 1      2      0.95
6 2      3      0.95
7 3      4      0.95
8 4      5      0.95
9 5      6      0.95
10 6     7      0.95
11 7     8      0.95
12 8     9      0.95
13 9     10     0.95
14 10    11     0.95
15 11    12     0.95
16 12    13     0.95
17 13    14     0.95
18 14    15     0.95
19 15    16     0.95
20 16    17     0.95
21 17    18     0.95
22 18    19     0.95
23 19    0      0.95
24 0      9      0.8
25 4      15     0.7
26 1      18     0.4
27 5      8      0.4
28 10     13     0.4
29 11     17     0.4
```

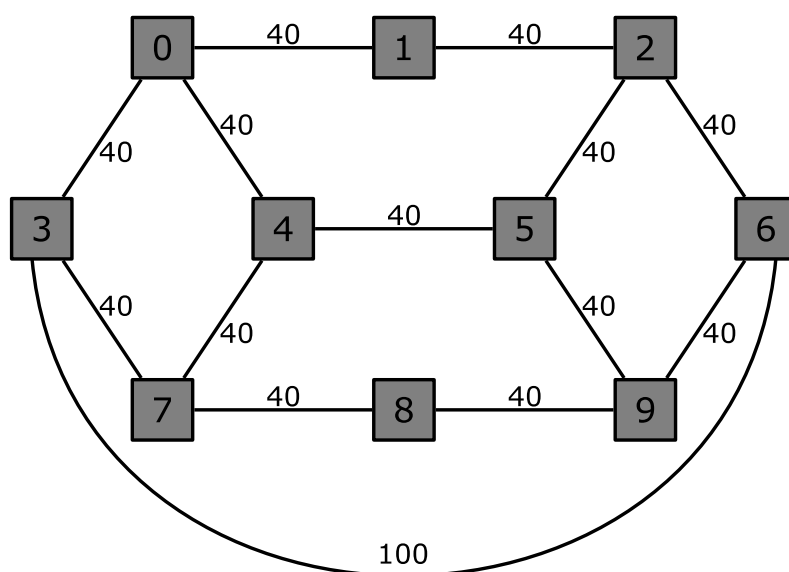


Listing 11: Wynik działania programu na1 dla danych z pliku input/network1d

```
1 $ ./na1 < input/network1d
2 Simulations: 100000
3 Failures: 8748
4 Average reliability: 91.252%
5 $
```

Sieć przedstawioną na rysunku 4 opisuje plik input/network1d zawarty w listingu 10. Wynik działania programu na1 dla tego pliku zawiera listing 11. Dodanie kolejnych czterech, jeszcze bardziej zawodnych krawędzi spowodowało uzyskanie średniej niezawodności na poziomie powyżej 90%.

Rysunek 5: Sieć opisana w pliku input/network2. Wagi oznaczają przepustowości łączy



Listing 12: Zawartość pliku input/network2

```
1 100000
2 0.140078
3 10
4 14
5 0      1      0.95    40
6 1      2      0.95    40
7 0      3      0.95    40
8 0      4      0.95    40
9 2      5      0.95    40
10 2     6      0.95    40
11 4     5      0.95    40
12 4     7      0.95    40
13 5     9      0.95    40
14 3     7      0.95    40
15 7     8      0.95    40
16 8     9      0.95    40
17 9     6      0.95    40
18 3     6      0.95    100
19 1 1 1 1 3 1 1 1 1 1
20 1 0 1 1 1 0 1 1 1 1
21 1 1 0 1 1 1 1 1 1 1
22 1 1 1 0 1 1 6 1 1 1
23 1 1 1 1 0 1 1 1 1 1
```

```

24 1 4 1 1 1 0 1 9 1 1
25 1 1 1 0 1 1 0 1 1 6
26 1 1 1 1 1 1 1 0 1 1
27 1 0 1 1 1 1 1 1 0 1
28 1 1 1 9 1 1 1 4 1 0

```

Listing 13: Wynik działania programu `na2` dla danych z pliku `input/network2`

```

1 $ ./na2 < input/network2
2 Simulations: 100000
3 Failures: 9893
4     Broken: 631
5     Overloaded: 3554
6     Timeout: 5708
7 Average reliability: 90.107%
8 $

```

Sieć przedstawiona na rysunku 5 jest opisana przez plik `input/network2` zawarty w listingu 12. Plik ten definiuje również maksymalne opóźnienie sieci (0,140078 sekundy) oraz macierz natężeń. Wynik działania programu `na2` dla tego pliku zawiera listing 13. Wykonano 100000 symulacji, podczas których sieć zawiodła 9893 razy: 631 razy rozspójniła się (broken), 3554 razy przeciążone zostało co najmniej jedno łącze (overloaded), 5708 razy sieć nie spełniła wymagań dotyczących maksymalnego średniego opóźnienia (timeout). Niezawodność sieci wyniosła 90,107%.

### 3 Wnioski

Programy, które zostały napisane na potrzeby zajęć laboratoryjnych, okazały się użytecznym narzędziem pozwalającym badać teoretyczną niezawodność sieci komputerowych. Dzięki sparametryzowaniu wszystkich wielkości osiągnięto elastyczność – można badać dowolnie zbudowane sieci nie ingerując w kody źródłowe programów.

