

Technologie sieciowe sprawozdanie 2

Wojciech Wróblewski

Przykłady topologii grafów

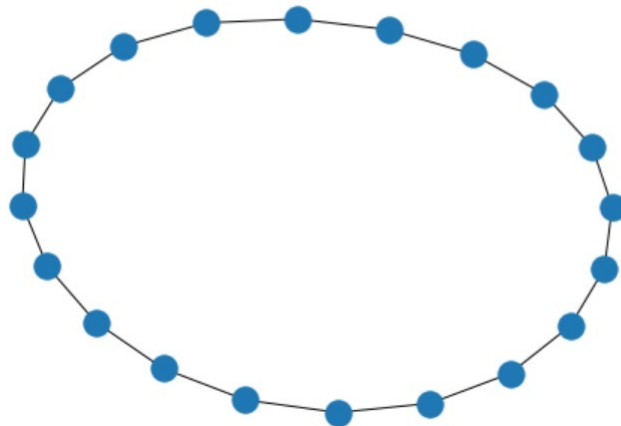
Rozważamy model sieci $S = \langle G, H \rangle$. Zaproponuj topologię grafu G ale tak aby żaden wierzchołek nie był izolowany oraz aby: $|V| = 20, |E| < 30$. W dalszej części sprawozdanie reprezentujemy funkcję krawędzi ze zbioru H .

- funkcję przepustowości c (rozumianą jako maksymalną liczbę bitów, którą można wprowadzić do kanału komunikacyjnego w ciągu sekundy)
- funkcję przepływu a (rozumianą jako faktyczną liczbę pakietów, które wprowadza się do kanału komunikacyjnego w ciągu sekundy).

Topologie grafu można wybrać na wiele sposobów. Jako pierwszy przypadek proponuję topologię pierścienia, czyli klasyczny graf cykliczny C_{20} . Graf można bardzo łatwo przekształcić np losowo dodając mu kolejne krawędzie.

Drugim przykładem będzie graf w dalszej części nazywany A , reprezentujący topolo-

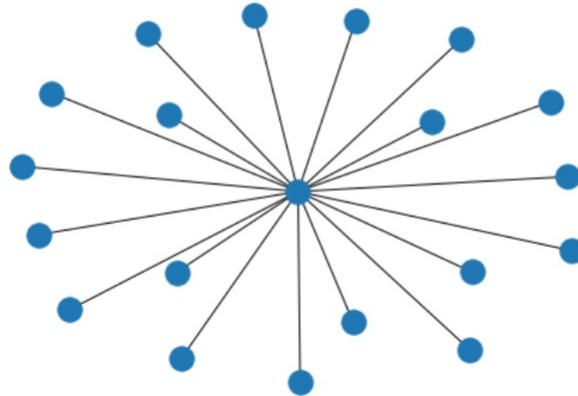
liczba krawędzi: 20
liczba wierzchołków: 20



gię gwiazdy. W topologii gwiazdy elementy końcowe podłączone są do jednego punktu centralnego.

graf A

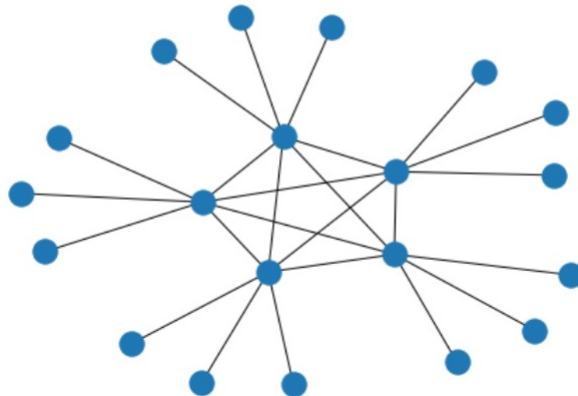
liczba krawędzi: 19
liczba wierzchołków: 20



Ostatnie dwa modele to modyfikacje poprzednich. Pierwszy graf B jest grafem, który w centrum zawiera podgraf pełny K_5 .

graf B

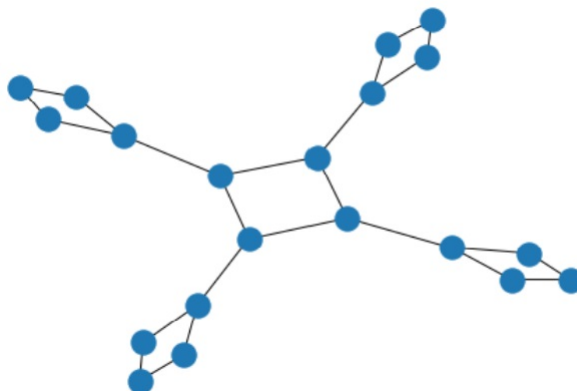
liczba krawędzi: 25
liczba wierzchołków: 20



Ostatni model w sprawozdaniu występuje pod nazwą D.

graf D

liczba krawędzi: 24
liczba wierzchołków: 20



Cel doświadczenia

Doświadczenie polega na stwierdzeniu jaki wpływ na niezawodność sieci ma dobór jej topologii, jej modyfikacji oraz parametrów charakterystycznych jak przepustowość, średnia wielkość pakietów, liczba pakietów. Wcześniejsze wizualizacje grafów oraz ich modele testowe zostały zaimplementowane w języku Python z wykorzystaniem biblioteki networkx. Zamysł implementacyjny funkcji będzie przedstawiany w pseudokodzie przypominającym języki C, Pascal.

Miara niezawodności

Przyjmijmy miarę niezawodności jako prawdopodobieństwo tego, że w dowolnym przedziale czasowym, nierozspójniona sieć zachowuje $T < T_{max}$, gdzie:

$$T = \frac{1}{G} \cdot \sum_{e \in E} \frac{a(e)}{\frac{c(e)}{m} - a(e)}, \text{ jest średnim opóźnieniem pakietu w sieci,}$$

G jest sumą wszystkich elementów macierzy natężeń N,

m jest średnią wielkością pakietu w bitach.

Model topologiczny grafu, p (prawdopodobieństwo nierozspójnienia krawędzi), macierz natężeń N oraz T_{max} traktujemy jako parametry przy wyznaczaniu niezawodności.

Macierz natężeń N

$N=[n(i,j)]$ 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 $v(j)$. Generujemy funkcję, która w zaplanowany sposób generuje nam macierz kwadratową $N_{|V|\times|V|}$.

```
procedure matrix_gen(SIZE=|V|)
begin
  for i:=0 to SIZE-1:
    for j:=0 to SIZE-1:
      if i==j:
        matrix[i][j]:=0
      else
        matrix[i][j]:=rand(specific_range)
    end
  end
end
```

Dobierzmy topologie grafów, które będziemy testować. Dane będą testowane dla grafu C_{20} , grafu D oraz kontrastowo grafu Petersena, który akurat nie spełnia zadanych warunków. Tworząc graf przypiszmy jego krawędziom właściwości :

- p - prawdopodobieństwo nierozspójnienia
- a - przepływ
- c - przepustowość

Przepływ będziemy wypełniać na podstawie macierzy natężeń dobierając najkrótszą drogę między wierzchołkami. Parametr p ustawimy jako jednakowy dla każdej krawędzi grafu i w ramach testów sprawdzimy zachowanie funkcji niezawodności, przy jego stałej wartości jak i takiej losowanej z zadanego zakresu. Funkcja opóźnienia:

```
procedure delay(graph, matrix)
begin
  set_a_property(graph,matrix)
  G := sum_of_elements(N)
  return count_delay_sum_for_edges()
end
```

Ostateczna funkcja obliczająca niezawodność grafu.

```
procedure graph_reliability(graph, matrix, Tmax, p, repeat_counter)

begin
  counter := 0

  for edges in graph->edges
    edge -> p := p
  for i:=0 to repeat_counter -1
    g := graph
```

```

for edge in graph->edges
    if random((0,1)) > edge->p do
        g.remove(edge)
    if is_connected(g) == False do
        continue

for edge in graph->edges
    if edge->a > edge->c do
        break
del := delay(g,matrix)

if del > = && del < Tmax do
    counter += 1
return counter / repeat_counter
end

```

```

print("Niezwadność grafu A ",reliability(star, matrix, 0.05, 0.98,1000))
print("Niezwadność grafu B ",reliability(complete, matrix, 0.05, 0.98,1000))
print("Niezwadność grafu C_20 ",reliability(cykliczny, matrix, 0.05, 0.98,1000))
print("Niezwadność grafu D ",reliability(quadratic, matrix, 0.05, 0.98,1000))
print("Niezwadność grafu Petersena ",reliability(petersen, matrix2, 0.05, 0.98,1000))

```

```

Niezwadność grafu A 0.2038
Niezwadność grafu B 0.2798
Niezwadność grafu C_20 0.2254
Niezwadność grafu D 0.5042
Niezwadność grafu Petersena 0.976

```

```

print("Niezwadność grafu A ",reliability(star, matrix, 0.05, random.uniform(0.87,0.99),1000))
print("Niezwadność grafu B ",reliability(complete, matrix, 0.05, random.uniform(0.87,0.99),1000))
print("Niezwadność grafu C_20 ",reliability(cykliczny, matrix, 0.05,random.uniform(0.87,0.99),1000))
print("Niezwadność grafu D ",reliability(quadratic, matrix, 0.05, random.uniform(0.87,0.99),1000))
print("Niezwadność grafu Petersena ",reliability(petersen, matrix2, 0.05, random.uniform(0.87,0.99),1000))

```

```

Niezwadność grafu A 0.0902
Niezwadność grafu B 0.1252
Niezwadność grafu C_20 0.0302
Niezwadność grafu D 0.2283
Niezwadność grafu Petersena 0.4442

```

Po sprawdzeniu wartości pomiarów dla przypadku gdy porównujemy miarę niezawodności dla $p=\text{const}$ (w tym wypadku wysokiego) i p losowego z zakresu $[0.88,0.99]$ widzimy, że dla pojedynczego pomiaru właściwość ta jest mocno różnicująca.

Pomiary

Zbadajmy jak, przy dobranej strukturze topologicznej, dobranych przepustowościach i manipulacji polegającej na zwiększaniu wartości w macierzy natężeń, zmieni się niezawodność.(Rys. 1)

Pomiary obejmują grafy : cykliczny C_{20} , D oraz petersena. Ten ostatni nie spełnia warunków zadanych dla topologii grafu jednak jest bardzo ciekawym i kontrastującym przykładem. Obserwujemy, że wraz ze wzrostem liczby pakietów w macierzy

C_20 graph					
step	p	c	t_max	reliability	
	0	0.98	1024	0.05	0.398
	2	0.98	1024	0.05	0.406
	4	0.98	1024	0.05	0.433
	6	0.98	1024	0.05	0.243
	8	0.98	1024	0.05	0.206
	10	0.98	1024	0.05	0.131
	12	0.98	1024	0.05	0.09
	14	0.98	1024	0.05	0.143
	16	0.98	1024	0.05	0
	18	0.98	1024	0.05	0.05
20	0.98	1024	0.05	0	
D graph					
0	0.98	1024	0.05	0.472	
5	0.98	1024	0.05	0.475	
10	0.98	1024	0.05	0.499	
15	0.98	1024	0.05	0.422	
20	0.98	1024	0.05	0.411	
25	0.98	1024	0.05	0.222	
30	0.98	1024	0.05	0.235	
35	0.98	1024	0.05	0.2	
40	0.98	1024	0.05	0.23	
45	0.98	1024	0.05	0.149	
50	0.98	1024	0.05	0.05	
Peterson graph					
0	0.98	1024	0.05	0.982	
10	0.98	1024	0.05	0.978	
20	0.98	1024	0.05	0.965	
30	0.98	1024	0.05	0.977	
40	0.98	1024	0.05	0.964	
50	0.98	1024	0.05	0.962	
60	0.98	1024	0.05	0.924	
70	0.98	1024	0.05	0.928	
80	0.98	1024	0.05	0.74	
90	0.98	1024	0.05	0.778	
100	0.98	1024	0.05	0.25	

Rysunek 1: operacja - zwiększanie natężeń

natężeń i przy stałej przepustowości niezawodność modelu spada. W danych w tabeli wartość step odpowiada jednostce stopniowego powiększania danych w macierzy natężeń. $\forall x \in N : x+ = step$

Teraz sprawdzimy jak zachowa się niezawodność przy zmianie przepustowości i zachowaniu innych parametrów jako stałych. (Rys. 2) W tym przypadku obserwacje rozszerzymy o ponowne wykonanie doświadczenia ze zmienną wartością p (prawdopodobieństwa nierozspójnienia). $\forall edge \in E : c(edge) + = step$ Obserwujemy, że przy wzroście przepustowości i zachowaniem zadanych stałych parametrów, niezawodność sieci znacząco rośnie. W przypadku gdy p jest stałe dla każdej krawędzi grafu, ale co próbę losowane z zakresu [0.88,0.99], trend nie ulega zmianie.

Zobaczmy jak zmieni się niezawodność przy manipulacjach krawędziami.

(Rys. 3) Do testowanych topologii będziemy losowo dodawać krawędzie, które w grafie początkowym nie istniały. Kolumna edges zawiera nową zmodyfikowaną liczbę krawędzi po operacji losowego dodania. Tabela wyników została oparta na losowanej wartości p z zakresu [0.88,0.99], jednak mimo losowości wyniki jednoznacznie pokazują wzrost niezawodności modelu po zastosowaniu operacji dodawania nowych krawędzi.

C_20									
step	p	c	t_max	reliability	step	p_random	c	t_max	reliability
10	0.98	512	0.05	0	10	0.877346386	512	0.05	0.022
60	0.98	512	0.05	0.183	60	0.986252429	512	0.05	0.05
110	0.98	512	0.05	0.236	110	0.88762066	512	0.05	0.19
160	0.98	512	0.05	0.381	160	0.929415338	512	0.05	0.212
210	0.98	512	0.05	0.309	210	0.930953256	512	0.05	0.337
260	0.98	512	0.05	0.434	260	0.98167144	512	0.05	0.428
310	0.98	512	0.05	0.348	310	0.88942628	512	0.05	0.379
360	0.98	512	0.05	0.358	360	0.982875607	512	0.05	0.311
410	0.98	512	0.05	0.389	410	0.976642002	512	0.05	0.362
460	0.98	512	0.05	0.422	460	0.914183721	512	0.05	0.371
510	0.98	512	0.05	0.463	510	0.965322794	512	0.05	0.326
560	0.98	512	0.05	0.437	560	0.898291195	512	0.05	0.346
petersen									
10	0.98	128	0.05	0.828	10	0.964584466	128	0.05	0.872
60	0.98	128	0.05	0.917	60	0.889148942	128	0.05	0.959
110	0.98	128	0.05	0.963	110	0.883592333	128	0.05	0.97
160	0.98	128	0.05	0.983	160	0.955524258	128	0.05	0.98
210	0.98	128	0.05	0.983	210	0.923357264	128	0.05	0.977
260	0.98	128	0.05	0.967	260	0.901778823	128	0.05	0.976
310	0.98	128	0.05	0.968	310	0.95375139	128	0.05	0.971
360	0.98	128	0.05	0.977	360	0.921856816	128	0.05	0.97
410	0.98	128	0.05	0.991	410	0.949735544	128	0.05	0.97
460	0.98	128	0.05	0.991	460	0.942399268	128	0.05	0.974
510	0.98	128	0.05	0.987	510	0.951226018	128	0.05	0.976
560	0.98	128	0.05	0.966	560	0.931944202	128	0.05	0.972
D graph									
10	0.98	512	0.05	0.332	10	0.916711685	512	0.05	0.35
60	0.98	512	0.05	0.432	60	0.948371203	512	0.05	0.312
110	0.98	512	0.05	0.369	110	0.949463458	512	0.05	0.358
160	0.98	512	0.05	0.492	160	0.965573843	512	0.05	0.321
210	0.98	512	0.05	0.436	210	0.928021557	512	0.05	0.508
260	0.98	512	0.05	0.398	260	0.982283263	512	0.05	0.52
310	0.98	512	0.05	0.459	310	0.889702988	512	0.05	0.458
360	0.98	512	0.05	0.506	360	0.985531166	512	0.05	0.515
410	0.98	512	0.05	0.424	410	0.888280624	512	0.05	0.454
460	0.98	512	0.05	0.499	460	0.911165361	512	0.05	0.489
510	0.98	512	0.05	0.505	510	0.877594761	512	0.05	0.497
560	0.98	512	0.05	0.532	560	0.985015576	512	0.05	0.512
610	0.98	512	0.05	0.498	610	0.941788796	512	0.05	0.5

Rysunek 2: operacja - zwiększanie przepustowości

W poprzednich statystykach parametr m , we wzorze na opóźnienie został przyjęty jako stały. Sprawdźmy zatem jaki wpływ na ostateczną niezawodność ma średnia wielkość pakietu.(Rys. 4)

Policzyliśmy odchylenie standardowe oraz wartość średnią miary niezawodności, a dane zostały obliczane dla prawdopodobieństwa nierozspójnienia p z zakresu $[0.88, 0.99]$. Obserwujemy, że wraz ze wzrostem średniej wielkości pakietu odchylenie standardowe wyników oraz sama miara niezawodności spada.

Wnioski

Istnieje wiele parametrów, które decydują o funkcjonowaniu sieci. Istotnym punktem jest topologia sieci, która powinna zapewnić jej spójność, jak i poszczególne parametry i funkcje przypisywane strukturze. Dobrze zaprojektowana sieć komputerowa jest przemyślana pod względem potrzeb jakie musi spełniać. Pokazaliśmy, że prawdopodobieństwo nierozspójnienia krawędzi grafu sieci jak i przepustowość oraz sama średnia

C_20 graph					
edges	p_rand	c	t_max	reliability	
20	0.963036861343565	512	0.05	0	
21	0.919079271504508	512	0.05	0	
22	0.904537665577662	512	0.05	0.029	
23	0.955986036174267	512	0.05	0.135	
24	0.932970619984194	512	0.05	0.341	
24	0.928303562238867	512	0.05	0.289	
25	0.929044761694399	512	0.05	0.467	
26	0.901930258638985	512	0.05	0.504	
26	0.967808220587442	512	0.05	0.524	
26	0.972952155414502	512	0.05	0.507	
27	0.873288819710609	512	0.05	0.69	
28	0.87893877846304	512	0.05	0.648	
29	0.972828055988724	512	0.05	0.712	
30	0.926885323271522	512	0.05	0.653	
31	0.890178397919958	512	0.05	0.72	
32	0.884251189369739	512	0.05	0.79	
33	0.971820237546081	512	0.05	0.814	
D graph					
24	0.918057815770656	512	0.05	0	
25	0.892433580889337	512	0.05	0.029	
26	0.965303243629455	512	0.05	0.021	
27	0.912042757981746	512	0.05	0.071	
28	0.895323972702917	512	0.05	0.517	
29	0.966769960892319	512	0.05	0.52	
30	0.939261341915272	512	0.05	0.6	
31	0.957664117994218	512	0.05	0.678	
32	0.905164821947926	512	0.05	0.809	
33	0.935441688829983	512	0.05	0.827	
34	0.946793447368955	512	0.05	0.835	
35	0.941855952804562	512	0.05	0.852	

Rysunek 3: operacja - dodawanie krawędzi

wielkość pakietów wpływa w istotny sposób na jej niezawodność. Kontrastowy graf Petersena jako graf regularny charakteryzował się bardzo dobrymi wynikami w statystykach niezawodności. Dlatego dobrym pomysłem przy budowaniu sieci jest korzystanie z modeli grafów k-spójnych, regularnych lub pełnych. Gorzej z grafami, które posiadają mniej krawędzi sąsiadujących w takich przypadkach zaobserwowaliśmy spadek niezawodności. Jednak dobrym usprawnieniem danej architektury jest modyfikacja sieci poprzez dodanie nowych krawędzi. Przy projektowaniu sieci należy tak dobrać parametry, żeby przy wymaganiach jakie oczekujemy (np. maksymalne opóźnienie), sieć działała niezawodnie oraz stabilnie.

graph	m	p_rand	t_max	reliability
cykliczny graph	6	0.97798639489693	0.05	0.005
cykliczny graph	6	0.9475365979266	0.05	0.001
cykliczny graph	6	0.95585312714777	0.05	0.002
cykliczny graph	6	0.91503418618315	0.05	0.001
cykliczny graph	6	0.97246181444269	0.05	0.001
cykliczny graph	6	0.95684428019759	0.05	0.001
cykliczny graph	6	0.91787305196247	0.05	0.003
cykliczny graph	6	0.92680576208086	0.05	0.032
cykliczny graph	6	0.9386078061645	0.05	0.002
cykliczny graph	6	0.87051243918411	0.05	0.007
cykliczny graph	6	0.97594891330878	0.05	0.002
cykliczny graph	6	0.9713536094419	0.05	0.001
			STDEV	0.0087577
			AVG	0.0048333
cykliczny graph	4	0.98121556463484	0.05	0.013
cykliczny graph	4	0.94547105714131	0.05	0.041
cykliczny graph	4	0.93378419509119	0.05	0.024
cykliczny graph	4	0.90697158153113	0.05	0.005
cykliczny graph	4	0.88048437638645	0.05	0.013
cykliczny graph	4	0.93450640985703	0.05	0.008
cykliczny graph	4	0.93126534161766	0.05	0.016
cykliczny graph	4	0.89648080895961	0.05	0.023
cykliczny graph	4	0.90217631741499	0.05	0.01
cykliczny graph	4	0.9440981655602	0.05	0.015
cykliczny graph	4	0.97732573095352	0.05	0.023
cykliczny graph	4	0.89508555453476	0.05	0.003
			STDEV	0.0104258
			AVG	0.0161667
cykliczny graph	2	0.93800310244012	0.05	0.057
cykliczny graph	2	0.924110159996	0.05	0.014
cykliczny graph	2	0.97336817412998	0.05	0.443
cykliczny graph	2	0.91974950786241	0.05	0.009
cykliczny graph	2	0.94674320821397	0.05	0.008
cykliczny graph	2	0.92489788267099	0.05	0.069
cykliczny graph	2	0.97560286930686	0.05	0.175
cykliczny graph	2	0.94814550755415	0.05	0.011
cykliczny graph	2	0.91801446876201	0.05	0.351
cykliczny graph	2	0.95438309467589	0.05	0.071
cykliczny graph	2	0.89614198893478	0.05	0.005
cykliczny graph	2	0.93190726188086	0.05	0.009
			STDEV	0.1474799
			AVG	0.1018333
cykliczny graph	1	0.96759642170144	0.05	0.034
cykliczny graph	1	0.93092545975399	0.05	0.033
cykliczny graph	1	0.9873822284488	0.05	0.02
cykliczny graph	1	0.92641308880522	0.05	0.05
cykliczny graph	1	0.98068544907955	0.05	0.011
cykliczny graph	1	0.89467071269787	0.05	0.025
cykliczny graph	1	0.93132618176631	0.05	0.363
cykliczny graph	1	0.93078232745858	0.05	0.032
cykliczny graph	1	0.98782722855367	0.05	0.049
cykliczny graph	1	0.90698967284006	0.05	0.087
cykliczny graph	1	0.93965874871913	0.05	0.228
cykliczny graph	1	0.97293431844451	0.05	0.164
cykliczny graph	1	0.87911842068336	0.05	0.022
			STDEV	0.1048292
			AVG	0.086

Rysunek 4: operacja - zwiększanie średniej wielkości pakietu