

## Lista 4.

**termin wykonania: 2021-06-13**

### Zadanie 1.

Rozbudowujemy zadanie z listy 3:

- Wierzchołki grafu, wykonujące protokół routingu, nazywamy *routerami*.
- Dodatkowo dodajemy nowy rodzaj elementów nazwanych *hostami*.
- Każdy host może być podłączony do jednego routera.
- Hosty podłączone do jednego routera są indeksowane kolejnymi liczbami od zera.
- Host o indeksie  $h$  podłączony do routera  $r$  ma w sieci adres, który jest parą numerów  $(r, h)$ .
- Host może generować pakiety adresowane do innego hosta. Nazwijmy je *pakietami standardowymi*, w odróżnieniu od *pakietów z ofertami* przesyłanymi między routerami na potrzeby *protokołu routingu*. Taki *pakiet standardowy* zawiera:
  - adres nadawcy:  $(r_s, h_s)$ ,
  - adres odbiorcy:  $(r_d, h_d)$  oraz
  - listę odwiedzonych routerów.
- W każdym routerze  $r$ , oprócz wątków  $Sender_r$  i  $Receiver_r$ , uruchomiony jest dodatkowy wątek  $Forwarder_r$ , który zajmuje się przekazywaniem *pakietów standardowych*. Działa on powtarzając następujące czynności:
  - Oczekuje na *pakiet standardowy* od
    - hosta podłączonego do  $r$  (nadawcy pakietu) lub
    - wątku  $Forwarder_j$ , sąsiedniego routera  $j, j \in N(r)$ .
  - Gdy otrzyma taki pakiet  $p$  o adresie odbiorcy  $(r_d, h_d)$ , to:
    - Dopisuje swój indeks  $r$  do listy odwiedzonych routerów pakietu  $p$ .
    - Jeśli  $r_d = r$ , to przesyła pakiet do (bezpośrednio podłączonego) hosta  $(r_d, h_d)$  -- odbiorcy pakietu.
    - Jeśli  $r_d \neq r$ , to przesyła pakiet do wątku  $Forwarder_n$  sąsiedniego routera  $n$ , takiego, że  $R_r[r_d].nexthop = n$ , gdzie  $R_r$  jest tablicą routingu routera  $r$ .
- Każdy host  $(r, h)$  działa następująco:
  - Losuje sobie jakiś inny istniejący host  $(r', h')$  i nadaje pakiet o adresie odbiorcy:  $(r', h')$ , przysyłając go do  $Forwarder_r$  swojego routera  $r$ .
  - Następnie przechodzi do trybu, w którym powtarza następujące czynności:
    - Oczekuje na nadejście jakiegoś pakietu od innego hosta.
    - Gdy nadejdzie taki pakiet  $p$  od nadawcy  $(r_s, h_s)$ , to:
      - Drukuje pakiet  $p$  (adresy nadawcy, odbiorcy i listę odwiedzonych routerów).
      - Po losowym opóźnieniu, nadaje pakiet o adresie odbiorcy  $(r, h_s)$ .

Po wygenerowaniu grafu połączeń, przed uruchomieniem działania systemu, wydrukować graf połączeń między routerami, zaznaczając przy każdym routerze liczbę podłączonych do niego hostów.

W czasie działania systemu, protokół routingu modyfikujący tablice routingu wykonuje się współbieżnie z przesyłaniem pakietów standardowych.

Dobrać wyświetlanie informacji oraz opóźnienia w działaniach protokołu routingu i hostów, tak aby było widoczne, że kolejne pakiety standardowe przesyłane między daną parą hostów podróżują coraz krótszymi trasami.

Komunikaty dotyczące protokołu routingu ograniczyć tylko do drukowania tablicy routingu, gdy zostanie ona zmodyfikowana przez *pakiet z ofertami*.

POPRAWKA: Słusznie zwrócono mi uwagę, że między wątkami *Forwarder* może powstać deadlock. W związku z tym, należy rozbić ten wątek na dwa wątki:

- jeden odbiera pakiety standardowe i umieszcza je w kolejce,
- drugi pobiera pakiety z kolejki i usiłuje je przekazać dalej.

Można założyć, że kolejka pakietów jest nieograniczona.

Punktacja:

- implementacja w Go: 3 p.
- implementacja w Adzie: 2.5 p.