**Raport z przedmiotu „Metody formalne”**

Sieci Petriego – model TCP

**Autorzy:**

Maciej Bruno-Kamiński

Mateusz Juraszek

Mateusz Maj

Wstęp

Projekt został podzielony na dwie części. Pierwszą z nich było stworzenie aplikacji, w której możliwa jest budowa i analiza sieci Petriego. Wśród dostępnych opcji analizy można sprawdzić własności wybranej sieci lub przeprowadzić symulację jej działania. W ramach drugiego zadania należało stworzyć model działania protokołu TCP w oparciu o zbudowany program.

Niniejszy raport zawiera szczegółowe informacje związane z drugim tematem projektu. Znajduje się w nim opis działania wspomnianego protokołu oraz model zbudowany w ramach ćwiczenia. W ostatnim punkcie przeprowadzono analizę stworzonej sieci oraz określono właściwości, które są dla niej spełnione.

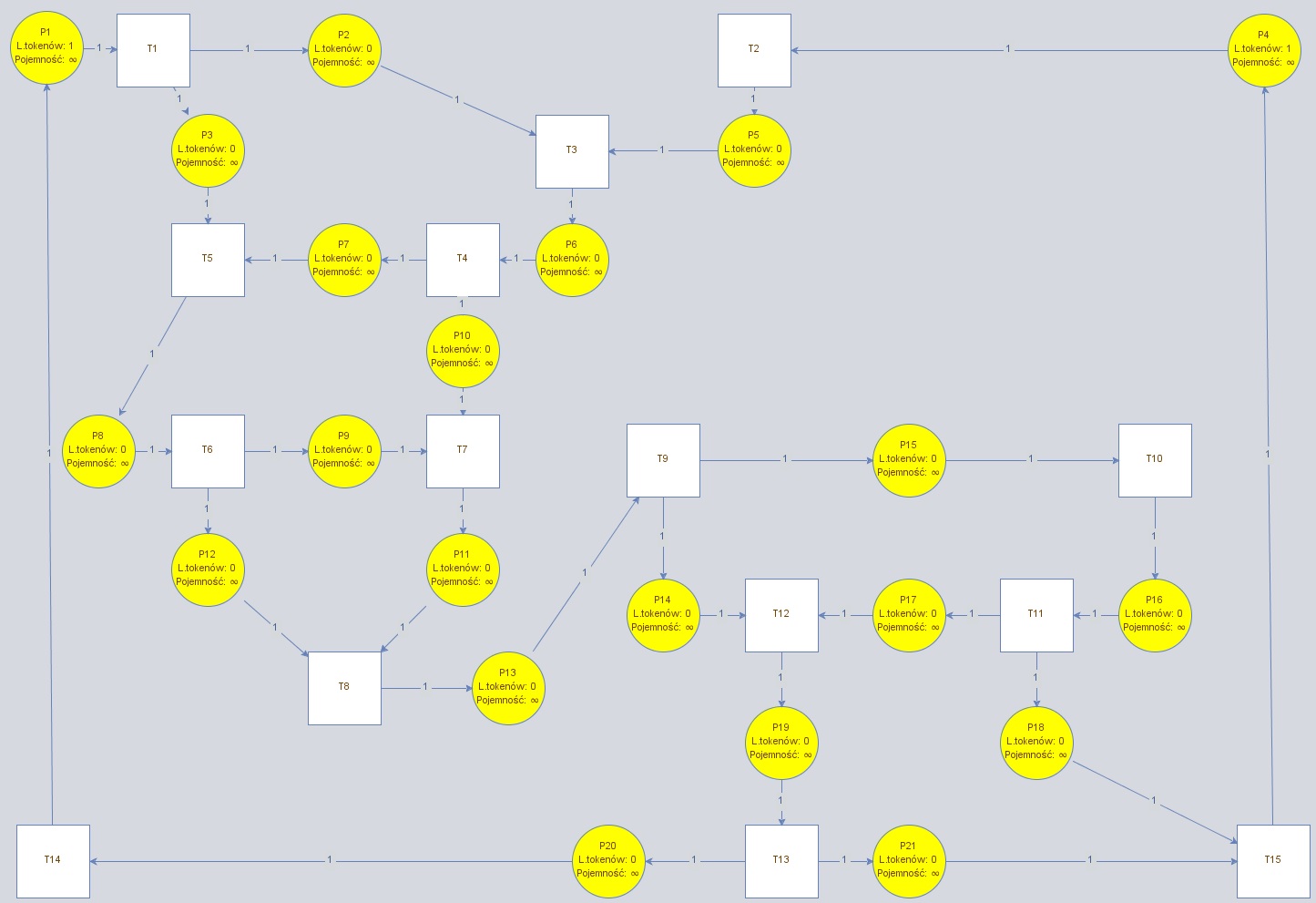
Transmission Control Protocol (TCP) jest to połączeniowy oraz niezawodny strumieniowy protokół komunikacyjny, wykorzystywany do przesyłania danych między procesami uruchomionymi na różnych maszynach. Operujący w warstwie transportowej modelu OSI, protokół TCP działa w trybie klient-serwer. Klient inicjuje połączenie do serwera. Serwer, z kolei, oczekuje na nawiązanie połączenia poprzez nasłuchiwanie na określonym porcie. W przeciwieństwie do UDP, TCP gwarantuje wyższym warstwom dostarczenie wszystkich pakietów w całości, z zachowaniem kolejności i bez duplikatów.

W celu nawiązania połączenia wykorzystywana jest procedura *three-way handshake*. Polega ona na wysłaniu trzech oddzielnych segmentów do drugiej strony, w określonej kolejności. Po każdej operacji wysłania bądź odebrania dana maszyna przechodzi w kolejny ze stanów TCP. Klient rozpoczyna transmisję od wysłania wiadomości SYN do serwera. W odpowiedzi otrzymuje segmenty SYN i ACK oraz przechodzi w stan ESTABLISHED, oznaczający gotowość do nawiązania połączenia. Jednocześnie wysyła do serwera ostatnią wiadomość ACK. Po wszystkim transmisja właściwych danych może się rozpocząć.

Zakończenie połączenia również jest realizowane w trybie *three-way handshake* i może zostać zainicjowane przez dowolną stronę. Ta strona wysyła wtedy pakiet z flagą FIN (*finished*), a w odpowiedzi otrzymuje segmenty ACK i FIN. Ostatnim etapem jest poinformowanie drugiej strony pakietem ACK oraz przeczekanie odpowiedniej ilości czasu (maksimum 4 minuty) w celu upewnienia się, że otrzymała ona potwierdzenie. W naszym modelu klient odpowiada za zakończenie połączenia.

Model protokołu TCP za pomocą sieci Petriego

Na rysunku 2.1 został zaprezentowany model działania protokołu TCP za pomocą sieci Petriego. Obejmuje on nawiązywanie oraz kończenie połączenia. Tabele 2.1 oraz 2.2 zawierają opis dostępnych miejsc oraz przejść.

****

**Rysunek 2.**Błąd! W dokumencie nie ma tekstu o podanym stylu.**.1** Model TCP

|  |  |
| --- | --- |
| **Nazwa miejsca** | **Opis** |
| P1 | (Klient) stan CLOSED |
| P2 | Wiadomość SYN w drodze do serwera |
| P3 | (Klient) stan SYN-SENT (wysłano SYN; oczekiwanie na SYN+ACK) |
| P4 | (Serwer) stan CLOSED |
| P5 | (Serwer) stan nasłuchiwania LISTEN |
| P6 | (Serwer) stan SYN-RECEIVED (otrzymano SYN) |
| P7 | Wiadomość SYN+ACK w drodze do klienta |
| P8 | (Klient) stan przyjęcia SYN+ACK |
| P9 | Wiadomość ACK w drodze do serwera |
| P10 | (Serwer) oczekiwanie na wiadomość ACK |
| P11 | (Serwer) gotowość do nawiązania połączenia |
| P12 | (Klient) gotowość do nawiązania połączenia |
| P13 | (Klient/Serwer) stan ESTABLISHED |
| P14 | (Klient) stan FIN-WAIT-1 (wysłano FIN) |
| P15 | Wiadomość FIN w drodze do serwera |
| P16 | (Serwer) stan CLOSE-WAIT |
| P17 | Wiadomość FIN+ACK w drodze do klienta |
| P18 | (Serwer) stan LAST-ACK (oczekiwanie na ACK) |
| P19 | (Klient) stan FIN-WAIT-2 |
| P20 | (Klient) stan TIME-WAIT |
| P21 | Wiadomość ACK w drodze do serwera |

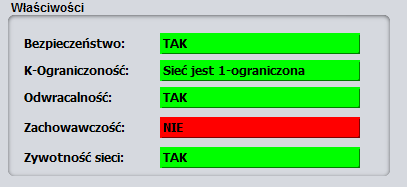
**Tabela 2.1** Opis miejsc w sieci

|  |  |
| --- | --- |
| **Nazwa przejścia** | **Opis** |
| T1 | (Klient) wysłanie wiadomości SYN |
| T2 | (Serwer) uruchomienie serwera |
| T3 | (Serwer) odebranie wiadomości SYN |
| T4 | (Serwer) wysłanie wiadomości SYN+ACK |
| T5 | (Klient) odebranie wiadomości SYN+ACK |
| T6 | (Klient) wysłanie wiadomości ACK |
| T7 | (Serwer) odebranie wiadomości ACK |
| T8 | (Klient/Serwer) nawiązywanie połączenia |
| T9 | (Klient) wysłanie wiadomości FIN |
| T10 | (Serwer) odebranie wiadomości FIN |
| T11 | (Serwer) wysłanie wiadomości FIN+ACK |
| T12 | (Klient) odebranie wiadomości FIN+ACK |
| T13 | (Klient) wysłanie wiadomości ACK |
| T14 | (Klient) zamykanie połączenia |
| T15 | (Serwer) zamykanie połączenia |

**Tabela 2.2** Opis przejść w sieci

Właściwości badanej sieci

W poniższym rozdziale zbadano kilka własności zbudowanej przez nas sieci. Wśród nich znajdują się k-ograniczoność, bezpieczeństwo, odwracalność, zachowawczość oraz żywotność sieci (w tym miejsc i przejść). Zbiorcze podsumowanie zostało zaprezentowane na rysunku 3.1.



**Rysunek 3.1** Właściwości sieci w raporcie programu

## **K-ograniczoność miejsc**

O k-ograniczoności miejsc mówimy wtedy, kiedy liczba znaczników w żadnym miejscu nie przekracza liczby *k*. W zbudowanym przez nas modelu przepływ danych jest jednoznaczny i w stanach może być co najwyżej jeden znacznik, zatem sieć jest 1-ograniczona.

## **Bezpieczeństwo**

Bezpieczeństwo sieci wiąże się z definicją k-ograniczoności. Sieć uznaje się za bezpieczną jeśli w każdym miejscu może znajdować się co najwyżej jeden znacznik (). Ponieważ w punkcie 4.1 wykazano, że badana sieć jest 1-ograniczona, można stwierdzić, że jest ona również bezpieczna.

## **Odwracalność**

Odwracalność sieci definiuje się jako możliwość powrotu do stanu początkowego z każdego miejsca w modelu, innymi słowy: czy model nie posiada żadnych miejsc terminalnych. Ponieważ zaprezentowany model TCP działa w pętli przechodząc od stanu CLOSED przez procesy nawiązywania oraz zamykania połączenia, kończąc ponownie w stanie CLOSED, właściwość ta jest spełniona.

## **Zachowawczość**

Zachowawczość oznacza, że w każdym stanie sieci suma znaczników będzie niezmienna. Nasza sieć modelująca TCP nie spełnia tej własności. W wielu stanach jeden znacznik zostaje rozdzielony na dwa, symbolizując w jednym momencie gotowość strony wysyłającej do podjęcia następnych kroków (np. oczekiwania na odpowiedź) oraz właściwą wiadomość będącą w drodze do drugiej maszyny.

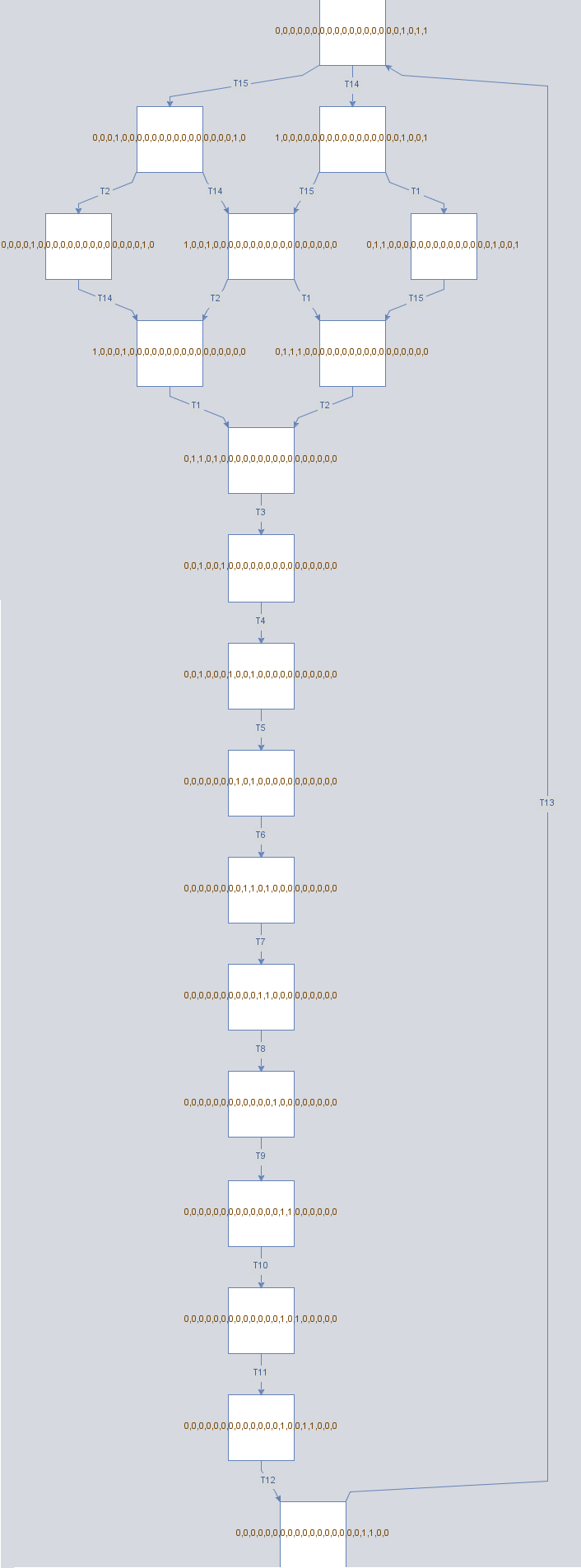
## **Żywotność sieci**

## **Żywotność miejsc i przejść**



**Rysunek 3.2** Żywotność poszczególnych miejsc i przejść

Graf osiągalności/pokrycia



**Rysunek 4.1** Graf pokrycia/osiągalności dla modelu TCP