

Sprawozdanie PSIR

Projekt

Jakub Sulikowski 318839
Michał Filipczyk 318761
Kacper Samuś 318829

Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych

15 stycznia 2024



**Wydział Elektroniki
i Technik Informacyjnych**

POLITECHNIKA WARSZAWSKA

Spis treści

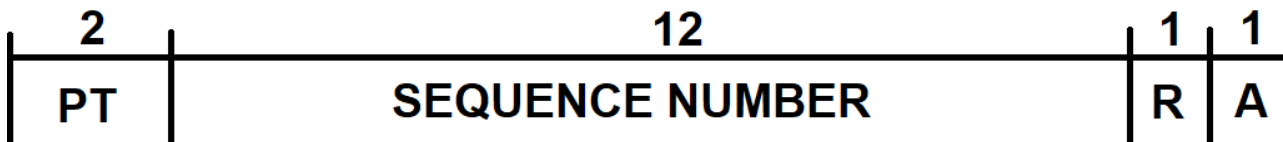
1. Zadanie projektowe	3
2. Specyfikacja ALP	3
2.1. Payload type	3
2.2. Sequence number (Nie zaimplementowano)	3
2.3. RDP_RESULT	3
2.4. Acknowledge	3
3. Opis implementacji serwera	3
3.1. Opis implementacji ALP na serwerze	3
3.2. Opis implementacji API przestrzeni krotek	4
3.3. Opis implementacji przestrzeni krotek na serwerze	4
4. Opis implementacji węzłów	4
4.1. Opis implementacji ALP na serwerze	4
4.2. Opis implementacji API przestrzeni krotek	4
5. Opis funkcjonalności oraz implementacji Aplikacji 1 - (Nie zaimplementowano)	4
6. Opis funkcjonalności oraz implementacji Aplikacji 2 - (Nie zaimplementowano)	4
7. Podsumowanie	4

1. Zadanie projektowe

Celem projektu było stworzenie warstwy programistycznej pośredniej między systemem operacyjnym, a aplikacjami, w postaci przestrzeni krotek. Ponadto, stworzenie protokołu ALP oraz dwóch aplikacji testujących.

2. Specyfikacja ALP

ALP jest to binarny protokół warstwy aplikacji działający na UDP, używany do komunikacji klienta z serwerem operującym przestrzenią krotek.



Rys. 1: Nagłówek pakietu protokołu ALP

Typ	ilość bitów
PAYLOAD TYPE	2
SEQUENCE NUMBER	12
RDP RESULT	1
ACKNOWLEDGE	1

2.1. Payload type

Zawiera informacje o typie przesyłanej wiadomości i wartości jakie przyjmuje odpowiednio:

- ack: potwierdź otrzymanie krotki o zadanych parametrach - wartość: 0b00
- out: umieść krotkę o zadanych parametrach w przestrzeni krotek - wartość: 0b01
- rdp: pobierz daną krotkę z przestrzeni krotek i zwróć RDP_RESULT = 1 jeżeli istnieje w przeciwnym razie zwróć RDP_RESULT = 0 - operacja nie blokująca - wartość: 0b10
- inp: pobierz a następnie wyjmij krotkę z przestrzeni krotek - operacja nie blokująca - wartość: 0b11

2.2. Sequence number (Nie zaimplementowano)

Numer sekwencji służy do identyfikowania czy nie utracono żadnego pakietu oraz czy pakiet należy do tego samego hosta w sesji.

2.3. RDP_RESULT

Jest wynikiem operacji rdp().

2.4. Acknowledge

Ta flaga ma wartość jeden gdy wysyłane jest potwierdzenie otrzymania wiadomości.

3. Opis implementacji serwera

3.1. Opis implementacji ALP na serwerze

Najważniejszą cechą naszej implementacji jest wprowadzenie "rutyn" odbierania i wysyłania. Rutyna wysyłania zawiera wysłanie oryginalnej wiadomości oraz oczekiwanie na potwierdzenie przez określony czas, po którym jeżeli nie otrzyma potwierdzenia następuje ponowne przesłanie oryginalnej wiadomości. Rutyna wysyłania jest

funkcją blokującą. W rutynie odbierania otrzymywana jest wiadomość oraz wysyłane jest potwierdzenie otrzymania wiadomości. Rutyna odbierania jest funkcją nieblokującą. Dodatkowo zaimplementowano mechanizm sprawdzania błędów protokołu.

3.2. Opis implementacji API przestrzeni krotek

API przestrzeni krotek jest swoistym łącznikiem między protokołem aplikacji, a przestrzenią krotek. W celu ułatwienia komunikacji zaimplementowano strukturę `tuple_t` odpowiedzialną za przechowywanie danych krotki - jej nazwę, liczbę pól oraz pola.

3.3. Opis implementacji przestrzeni krotek na serwerze

Przestrzeń krotek zaimplementowano jako `linked.list` co ułatwia operacje na przestrzeni. Na przestrzeni krotek można wykonać 3 operacje:

- Dodać krotkę na końcu listy
- Zwrócić krotkę i usunąć na podstawie indeksu
- Zwrócić krotkę bez usuwania na podstawie indeksu

Mechanizm porównywania danej krotki z danym szablonem jest zrealizowany rekurencyjnie.

4. Opis implementacji węzłów

4.1. Opis implementacji ALP na serwerze

Podobnie jak w implementacji serwera działanie protokołu opiera się na rutynach. Jedyną różnicą między implementacją serwerową, a węzłową jest to, że w węzłach nie zaimplementowano mechanizmu sprawdzania błędów, który figuruje w implementacji protokołu na serwerze.

4.2. Opis implementacji API przestrzeni krotek

W węzłach zaimplementowano odpowiednie funkcje reprezentujące operacje na przestrzeni krotek. Podobnie jak w implementacji na serwerze skorzystano ze struktury `tuple_t`.

5. Opis funkcjonalności oraz implementacji Aplikacji 1 - (Nie zaimplementowano)

6. Opis funkcjonalności oraz implementacji Aplikacji 2 - (Nie zaimplementowano)

7. Podsumowanie

Niestety nie udało się zaimplementować wszystkich wymaganych składników w wyznaczonym czasie.