

1 Streszczenie

Niniejszy dokument zawiera opis specyfikacji protokołu sterującego połączeniami telefonicznymi w sieci IP. Opis jest częścią zadania zaliczeniowego na laboratorium z sieci komputerowych.

2 Opis celów protokołu

Protokół ma umożliwić imitację działania typowej biurowej centrali telefonicznej w sieci IP zgodnie z treścią zadania znajdującą się w /home/inf/PUBLIC/SIK/zadanie/zadanie2007.txt na komputerze students.mimuw.edu.pl.

3 Opis założeń protokołu

1. Protokół obejmuje jedynie obsługę dwustronnego kanału sterującego wymianą informacji, które umożliwiają realizację połączeń głosowych.
2. Sterowanie ma być rozproszone, bez centralnego serwera.
3. Protokół działa w warstwie aplikacji i wykorzystuje gniazda.
4. Do transmisji kanału głosowego protokół używa UDP.
5. Do transmisji kanału sterującego protokół używa TCP.
6. Protokół wykorzystuje model komunikacji P2P.
7. Do synchronizacji 2 abonentów używam wymiany komunikatów.
8. Dodatkowo do synchronizacji wewnątrz grupy protokół używa algorytmu realizującego wzajemne wykluczanie w systemie rozproszonym - "Krażący żeton" (ang. token ring) przykładowy opis jest na http://students.mimuw.edu.pl/SR-ZSI/Wyklady-html/03_synchro/03_synchro.html.
9. Każdy abonent jest identyfikowany przez unikalny numer typu uint32. Numeracja powinna być dana w pliku konfiguracyjnym.
10. Zakładamy, że abonenci w obrębie grupy są połączeni w logiczny pierścień.
11. Żeton jest tworzony przez ten proces z grupy, który ma najniższy numer.

12. Najnowsza wersja treści zadania nakazuje założenie tego, że ten pierścień nie zostanie rozerwany, jak również krążący w nim żeton nie zostanie zgubiony.

4 Opis formatu komunikatów

4.1 Podstawowe założenia

1. Protokół zakłada sieciowy porządek oktetów w liczbach.
2. Protokół używa U2 jako sposobu zapisu liczb całkowitych.
3. Jako format zapisu liczb zmiennoprzecinkowych ustalamy IEEE 754.
4. Niedozwolone jest uzupełnianie formatu danych.
5. Nazwy typów podstawowych są zgodne z tymi znajdującymi się w /home/inf/PUBLIC/SIK/zadanie/opis-protokolow.txt na komputerze students.mimuw.edu.pl.

4.2 Typy złożone

```
wlt {
    uint8    state
    uint32   group_speaker
    uint32   second_speaker
    uint32   second_speaker_wlt
}
```

Struktura ta reprezentuje WLT, jej rozmiar to 13 bajtów. Opis pól:

- state - stan WLT, patrz Opis stanów;
- group_speaker - unikalny numer rozmówcy z grupy abonenta;
- second_speaker - unikalny numer drugiego rozmówcy;
- second_speaker_wlt - numer WLT używany przez drugiego rozmówcę;

```
token_msg {
    uint32    total_length;
    wlt[ wlt_length ] wlt;
}
```

Struktura ta reprezentuje krążący w grupie żeton.
Gdzie $wlts_length = total_length / 13$. Opis pól:

- `total_length` - pole zawierające długość całego pakietu, długość ta obejmuje pola: `total_length`, `wlts`;
- `wlts` - tablica opisująca wszystkie WLT danej grupy, porządek WLT nie może ulec zmianie, ustalany jest jeden raz przez proces tworzący żeton;

```
protocol_msg {  
    uint8      message_type ;  
    uint32     sender ;  
    uint32     sender_wlt_nr ;  
    uint32     receiver_wlt_nr ;  
}
```

Struktura ta reprezentuje komunikat, którym wymieniają się dwaj abonenci w celu obsługi połączenia, jej rozmiar to 13 bajtów. Opis pól:

- `message_type` - pole identyfikujące rodzaj komunikatu;
- `sender` - unikalny numer nadawcy;
- `sender_wlt_nr` - numer WLT nadawcy, przez którą prowadzona jest komunikacja;
- `receiver_wlt_nr` - numer WLT odbiorcy, przez którą prowadzona jest komunikacja;

5 Opis wymienianych komunikatów

Wyróżniamy następujące typy komunikatów:

1. TOKEN_MSG:

- binarnie reprezentowany przez strukturę `token_msg`;
- komunikat krąży w grupie, należy minimalizować czas przetrzymywania żetonu;
- tylko abonent posiadający żeton (ten, który odebrał komunikat, ale jeszcze go nie wysłał) może zmienić stan WLT;
- żeton jest tworzony przez tego abonenta z grupy, który ma najniższy numer;

2. CONNECT_MSG:

- binarnie reprezentowany przez strukturę protocol_msg o polu message_type = 1;
- wysyłanie komunikatu inicjowane jest przez użytkownika (o identyfikatorze sender), który chce nawiązać połączenie przy użyciu WLT (będącej w stanie DISCONNECTED_STATE) o numerze sender_wlt_nr;
- abonent odbierający komunikat, szuka WLT (będącej w stanie DISCONNECTED_STATE) i jeśli ją znajdzie to odsyła komunikat SIGNAL_MSG z uzupełnionym polem sender_wlt_nr;

3. SIGNAL_MSG:

- binarnie reprezentowany przez strukturę protocol_msg o polu message_type = 2;
- wysyłanie komunikatu inicjowane jest przez protokół (o identyfikatorze sender), który odpowiada na próbę nawiązania połączenia przez abonenta o numerze receiver (znalazł WLT w stanie DISCONNECTED_STATE);
- odebranie tego komunikatu oznacza, że udało się połączyć, czekamy tylko na "podniesienie słuchawki";

4. ACCEPT_MSG:

- binarnie reprezentowany przez strukturę protocol_msg o polu message_type = 3;
- wysyłanie komunikatu inicjowane jest przez użytkownika (o identyfikatorze sender), który "podnosi słuchawkę", zgodnie z treścią nie musi to być ten sam abonent co wysłał komunikat SIGNAL_MSG;
- odebranie tego komunikatu oznacza, że rozmówca sender "podniósł słuchawkę";

5. PAUSE_MSG:

- binarnie reprezentowany przez strukturę protocol_msg o polu message_type = 4;
- wysyłanie komunikatu inicjowane jest przez użytkownika (o identyfikatorze sender), który zawiesza połączenie z adresatem komunikatu;
- odebranie tego komunikatu oznacza, że rozmówca sender zawiesił połączenie;

6. RESUME_MSG:

- binarnie reprezentowany przez strukturę protocol_msg o polu message_type = 5;
- wysyłanie komunikatu inicjowane jest przez użytkownika (o identyfikatorze sender), który ódwiesza połączenie", zgodnie z treścią nie musi to być ten sam abonent co wysłał komunikat PAUSE_MSG;
- odebranie tego komunikatu oznacza, że rozmówca sender ódwiślał połączenie";

7. CANCEL_MSG:

- binarnie reprezentowany przez strukturę protocol_msg o polu message_type = 6;
- jest to najbardziej uniwersalny komunikat jest wiele scenariuszy jego użycia
- wysyłanie komunikatu inicjowane jest przez implementacje, jeśli po odebraniu komunikatu CONNECT_MSG nie ma ani jednej WLT w stanie DISCONNECTED_STATE;
- odebranie tego komunikatu w stanie PREPARING_STATE oznacza brak WLT w stanie DISCONNECTED_STATE po drugiej stronie;
- wysyłanie komunikatu inicjowane jest przez użytkownika (o identyfikatorze sender), który ódrzuca połączenie", zgodnie z treścią nie musi to być ten sam abonent co wysłał komunikat SIGNAL_MSG;
- odebranie komunikatu oznacza, w stanie RINGING_STATE, PAUSED_STATE lub PREPARED_STATE oznacze że połączenie zostało odrzucone;
- odebranie tego komunikatu w stanie TALKING_STATE oznacza przerwanie połączenia;
- za każdym razem odebranie tego komunikatu powoduje przejście do stanu DISCONNECTED_STATE

6 Opis stanów

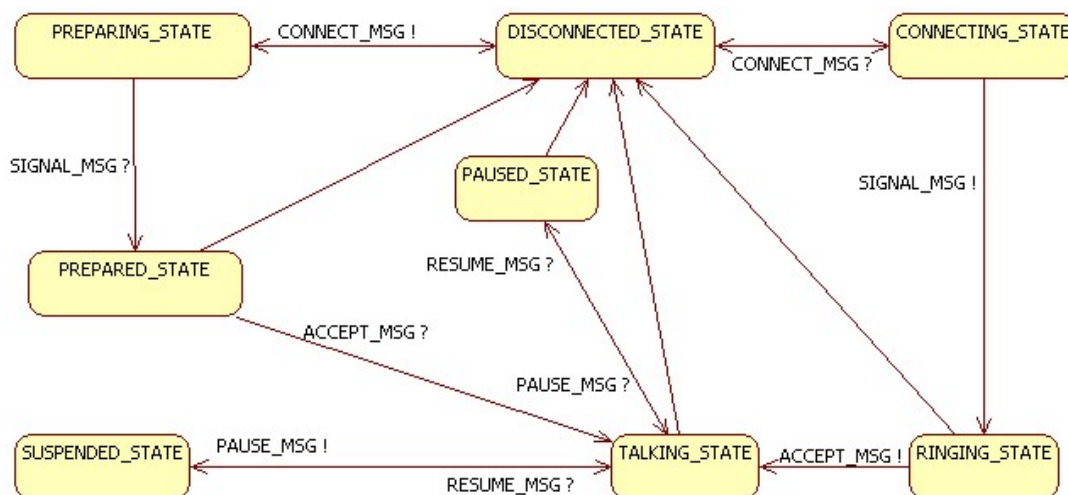
Stan danego abonenta jest produktem stanów jego WLT.

Przypominam, że aby zmienić stan którejs z WLT konieczne jest posiadanie żetonu

Technicznie stan WLT jest możliwą wartością pola state w strukturze wlt. Struktury wszystkich WLT danej grupy są trzymane w żetonie w tablicy wlts.

6.1 Diagram stanów

Poniższy diagram stanów dotyczy stanów pojedynczej WLT.



6.2 Opis

Wyróżniamy następujące rodzaje stanów WLT:

1. DISCONNECTED_STATE:

- reprezentowany przez wartość pola state = 0;
- oznacza, że linia jest wolna i może zostać zajęta;

2. PREPARING_STATE:

- reprezentowany przez wartość pola state = 1;
- oznacza, że linia jest zajęta przez abonenta o numerze group_speaker, który próbuje zestawić połączenie;

3. PREPARED_STATE:

- reprezentowany przez wartość pola state = 2;
- oznacza, że linia jest zajęta przez abonenta o numerze group_speaker, który czeka na komunikat ACCEPT_MSG;

4. CONNECTING_STATE:

- reprezentowany przez wartość pola state = 3;

- oznacza, że linia jest zajęta przez abonenta o numerze `group_speaker`, który zarezerwował ją na połączenie z abonentem o numerze `second_speaker`;
- stan ten jest powinnien być krótkotrwały, gdyż po odebraniu `CONNECT_MSG` można szybko stwierdzić czy należy wysłać `CANCEL_MSG` czy też `SIGNAL_MSG`;

5. `RINGING_STATE`:

- reprezentowany przez wartość pola `state = 4`;
- oznacza, że linia jest zajęta, ale każdy abonent z grupy może odebrać połączenie wysyłając komunikat `ACCEPT_MSG`;

6. `TALKING_STATE`:

- reprezentowany przez wartość pola `state = 5`;
- oznacza, że linia jest zajęta przez abonenta o numerze `group_speaker`, prowadzona jest rozmowa;

7. `PAUSED_STATE`:

- reprezentowany przez wartość pola `state = 6`;
- oznacza, że linia jest zajęta przez abonenta o numerze `group_speaker`;
- połączenie jest zawieszona, `group_speker` czeka na odwieszenie połączenia (komunikat `RESUME_MSG`) lub rezygnację z połączenia (zdarzenie inicjowane przez użytkownika);

8. `SUSPENDED_STATE`:

- reprezentowany przez wartość pola `state = 7`;
- oznacza, że linia jest zajęta, ale każdy abonent z grupy może przejąć połączenie wysyłając komunikat `RESUME_MSG`;