

Serwis cytatów - projekt protokołu

Mateusz Grabowski (291540)

2.05.2011

Spis treści

1	Streszczenie	1
2	Terminologia	2
3	Cele protokołu	2
4	Założenia protokołu	2
5	Format komunikatów	2
5.1	Reprezentacja wartości liczbowych	2
5.2	Reprezentacja znaków	3
5.3	Złożone typy danych	3
5.4	Typy komunikatów	3
6	Opis wymienianych komunikatów	4
7	Opis stanów	4
7.1	Zachowanie serwisu	5
7.1.1	WAIT_REQUEST_STATE	5
7.1.2	SEND_PART_STATE	5
7.2	Zachowanie użytkownika	7
7.2.1	WAIT_FOR_PART_STATE	7
7.2.2	HAS_ALL_PARTS_STATE	8
8	Używane numery	8
8.1	Kody błędów	8

1 Streszczenie

Ninieszy dokument opisuje specyfikację protokołu do obsługi serwisu cytatów - sentencji. Opis protokołu zawiera następujące sekcje:

- cele protokołu
- założenia
- format komunikatów
- stany
- używane numery i kody błędów

2 Terminologia

Serwis - instancja aplikacji korzystająca z protokołu oraz mająca za zadanie udostępnianie losowych cytatów z bazy danych

Użytkownik - aplikacja działająca w celu pozyskania nowego cytatu

Klucz żądania - 64-bitowa nieujemna liczba całkowita wylosowana przez użytkownika i wysłana serwisowi, jednoznacznie identyfikująca dane żądanie

3 Cele protokołu

Główne przeznaczenie protokołu stanowi obsługa serwisu cytatów - sentencji. Każdy z użytkowników może połączyć się z serwisem i na zasadach równych praw ma otrzymać losowo wybrany cytat. Użytkownik może w każdym momencie wysłać żądanie nowego cytatu do serwisu. Po jego otrzymaniu wybierany jest odpowiedni tekst i przesyłany w całości w jednym komunikacie, bądź, jeśli jego rozmiar jest za duży, w wielu częściach.

4 Założenia protokołu

Protokół działa w warstwie aplikacji. Priorytetem jest szybkość przesyłania informacji, dlatego przy transporcie jest wykorzystywany protokół UDP. Nie występuje zatem narzut na nawiązywanie połączenia i śledzenie sesji. Nie ma jednak gwarancji na dostarczenie przesyłki.

Protokół działa w oparciu o model klient - serwer. Obsługiwane są połączenia wielu użytkowników z serwisem, który oczekuje na komunikaty przychodzące na port o numerze **SERVER_PORT**.

Kolejność pakietów może ulec zmianie przed ich dotarciem do celu, mogą one też zostać powielone. Dlatego, gdy konieczne jest podzielenie dłuższego cytatu na mniejsze fragmenty, do każdego komunikatu dodawana jest informacja, który z kolei fragment cytatu się w nim znajduje. Odpowiedni numer jest zapamiętywany w dwubajtowej liczbie, dlatego kawałków nie może być więcej niż 65 536. Powoduje to, że maksymalna długość cytatu nie może przekraczać $65\,536 \cdot \text{MAX_PART_SIZE}$ ¹.

5 Format komunikatów

5.1 Reprezentacja wartości liczbowych

Porządek oktetów w liczbach: sieciowy

System reprezentacji liczb ze znakiem: kod uzupełnień do dwóch (U2)

Format liczb zmiennoprzecinkowych: w protokole nie występują liczby zmiennoprzecinkowe

Dopuszczalne uzupełnienia formatu: komunikat może być uzupełniany zerami

¹zatem w skrajnym przypadku cytat nie może mieć więcej niż 4 292 018 176 bajtowych znaków, co w sumie zajmowałoby prawie 4 GB

5.2 Reprezentacja znaków

Sposób kodowania znaków: UTF-8²

5.3 Złożone typy danych

```
CITE_PART_T {
    uint16          data_length;
    octet [data_length] data;
}
```

`data_length` - pole opisujące długość tablicy `data`

`data` - fragment cytatu, który jest przesyłany użytkownikowi

5.4 Typy komunikatów

```
MSG_T {
    uint8      msg_id;
    uint64     key;
}
```

```
CITE_MSG_T {
    uint8      msg_id;
    uint64     key;
    unit16     index;
    unit16     parts;
    CITE_PART_T text;
}
```

```
ERR_MSG_T {
    uint8      msg_id;
    uint8      errno;
}
```

Znaczenie poszczególnych pól:

`msg_id` - rodzaj komunikatu (pole przyszłościowe)

`key` - klucz identyfikujący żądanie użytkownika (64-bitowa liczba zabezpiecza przed kolizją)

`index` - numer części cytatu (fragmenty numerowane od 0)

`parts` - liczba wszystkich części cytatu (liczba o jeden większa od numeru ostatniej części cytatu)

`text` - część cytatu wysłana w pakiecie

`errno` - kod błędu

²<http://tools.ietf.org/html/rfc3629>

6 Opis wymienianych komunikatów

CITE_REQUEST_MSG(key) : MSG_T

Zapytanie użytkownika o losowy cytat skierowane do serwisu.

REPLY_MSG(key, index, parts, text) : CITE_MSG_T

Część losowego cytatu będącego odpowiedzią na zapytanie. Komunikat mieści się w jednym pakiecie UDP.

SERVER_ERR_MSG(errno) : ERR_MSG_T

Informacja serwisu o błędzie.

msg_id	Komunikat
0	SERVER_ERR_MSG
1	CITE_REQUEST_MSG
2	REPLY_MSG

7 Opis stanów

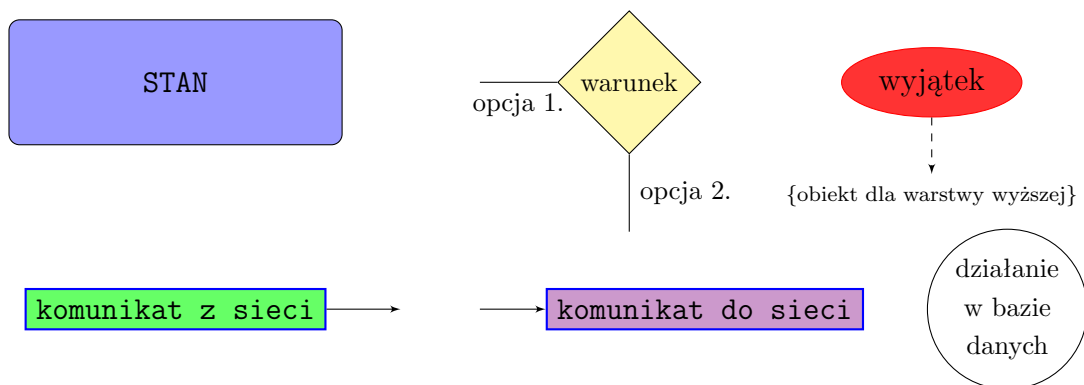
Serwis może znajdować się w jednym z dwóch stanów stanów:

- WAIT_REQUEST_STATE
- SEND_PART_STATE

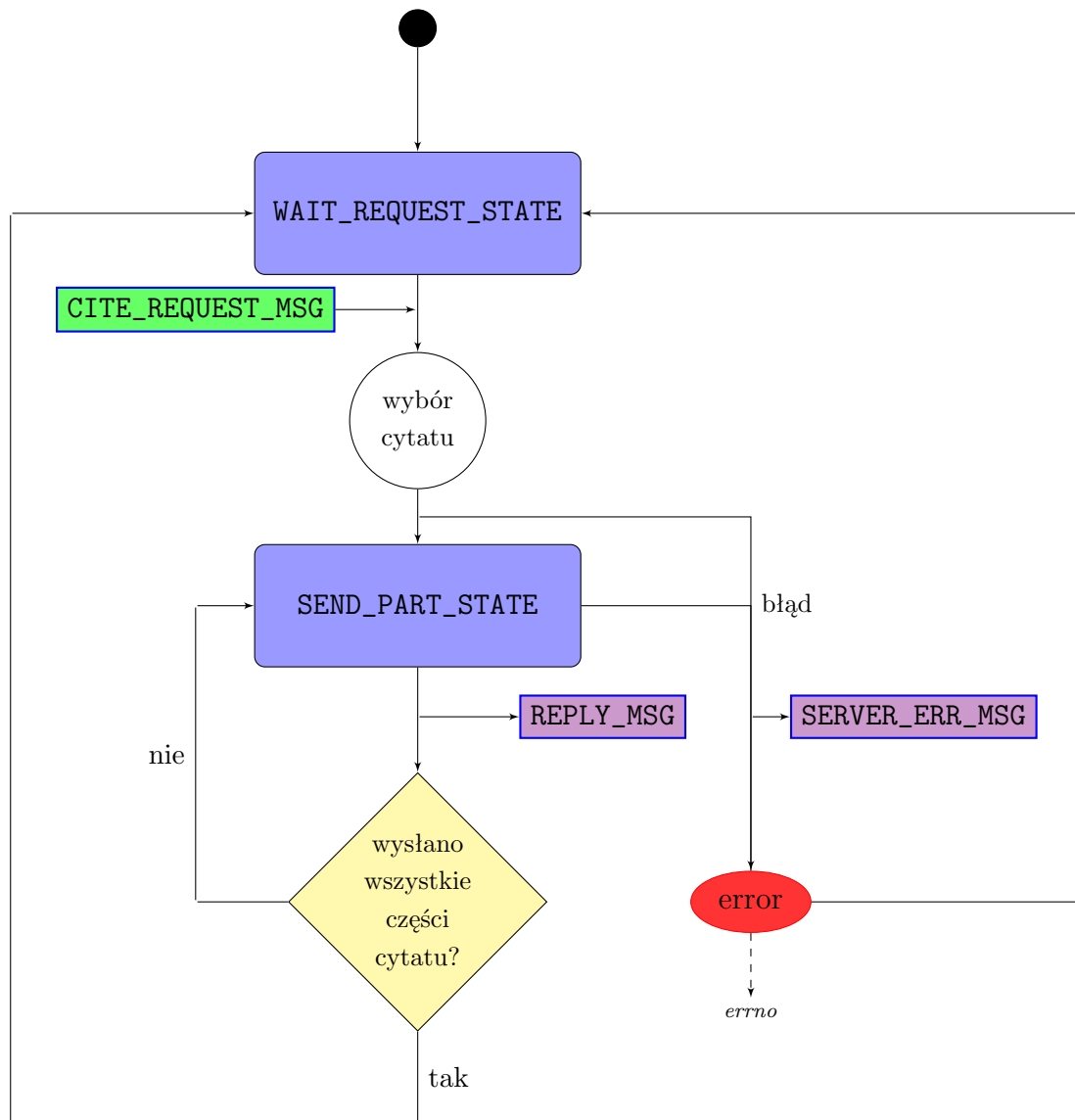
Użytkownik może znajdować się w jednym z dwóch stanów:

- WAIT_FOR_PART_STATE
- HAS_ALL_PARTS_STATE

Oznaczenia używane na diagramach:



7.1 Zachowanie serwisu



7.1.1 WAIT_REQUEST_STATE

Serwis oczekuje w tym stanie na zapytanie o nowy cytat, zatem akceptowany jest w nim tylko komunikat:

- CITE_REQUEST_MSG(key)

Po jego otrzymaniu i pomyślnym wybraniu cytatu serwis przechodzi do stanu SEND_PART_STATE. W przypadku wystąpienia błędu wysyłany jest komunikat SERVER_ERR_MSG z odpowiednio ustawioną wartością `errno`, informowana jest także wartswa wyższa a serwis pozostaje w tym stanie.

7.1.2 SEND_PART_STATE

Serwis znajduje się w tym stanie po otrzymaniu zapytania i wybraniu cytatu z bazy danych. Następuje wysłanie odpowiedniego tekstu w komunikacie

REPLY_MSG(key, index, parts, text), a jeśli jego długość jest większa niż MAX_PART_SIZE, to jest on dzielony na mniejsze fragmenty. Zmienna `key` jest pobierana z komunikatu otrzymanego od użytkownika, a `index` ustalany jest osobno dla każdej części cytatu. Przed wysłaniem pierwszej części obliczana jest ich łączna liczba, następnie zapamiętywana i wstawiana do zmiennej `parts` w każdym komunikacie. Jeżeli cytat nie mieści się w całości w jednym

komunikacie, wówczas dzielony jest na fragmenty, z których każdy, oprócz ostatniego, ma długość `MAX_PART_SIZE`. Następnie są one indeksowane od 0 i kolejno wysyłane. Do reprezentacji fragmentu służy struktura `text`. Tekst zapisywany jest w polu `data`.

Jeżeli fragment nie jest ostatnim, wówczas na końcu tego pola wstawiany jest znak końca bloku transmisji (ETB)³. Ostatni wysłany fragment, z indeksem `parts - 1`, może mieć krótszą długość niż `MAX_PART_SIZE`. Jego tekst jest zakończony znakiem końca transmisji (EOT)⁴. W jej polu `data_length` ustawiana jest długość danego fragmentu bez dodatkowego znaku ETB/EOT.

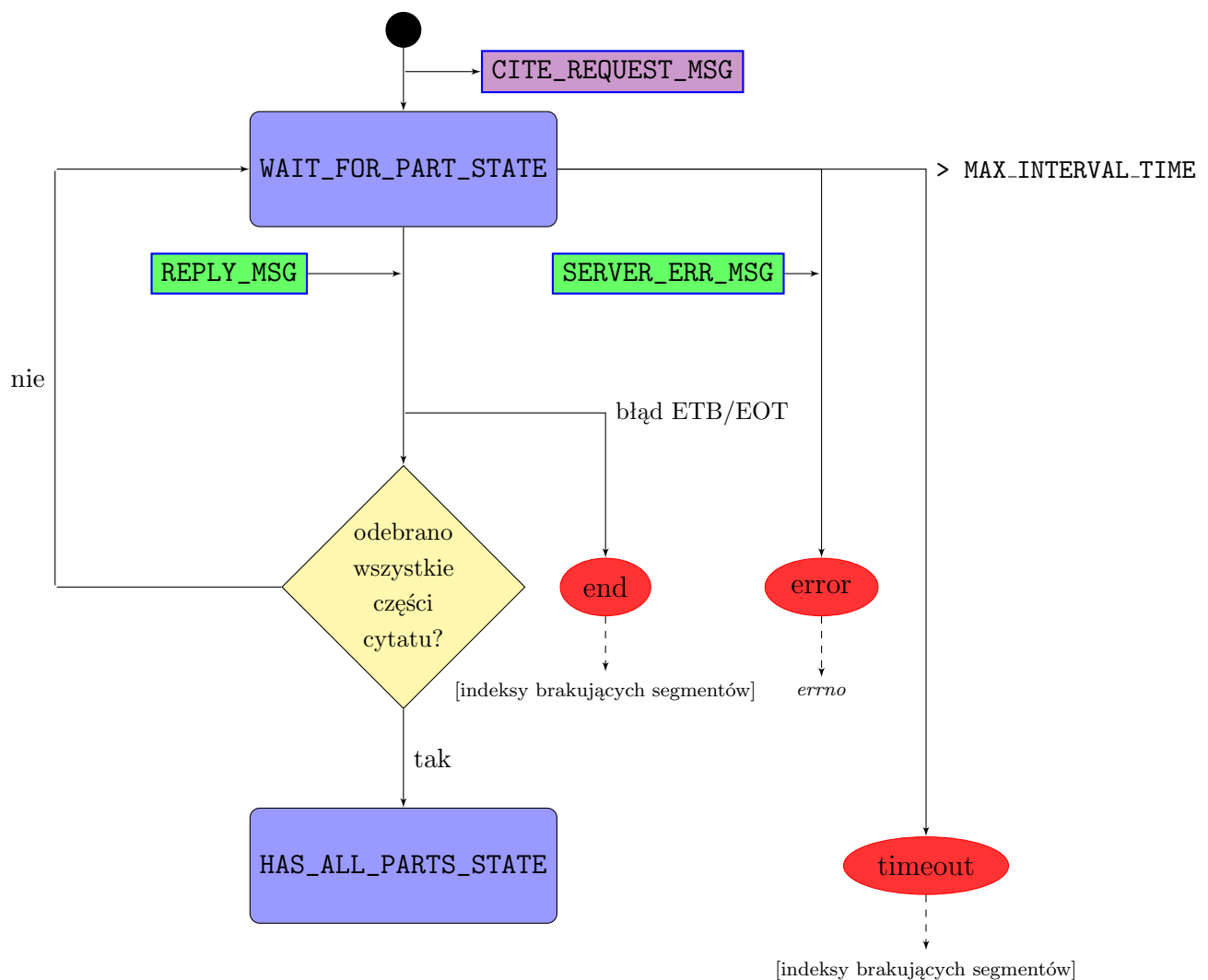
Jeżeli w trakcie działania serwisu wystąpi błąd, to jest on natychmiast zgłaszany klientowi, do którego nie zostały jeszcze wysłane wszystkie części cytatu, poprzez wysłanie komunikatu `SERVER_ERR_MSG` z odpowiednio ustawioną wartością `errno`. Kolejne fragmenty danego cytatu nie są już wysyłane.

Wszystkie komunikaty przychodzące są w tym stanie ignorowane. Po wysłaniu wszystkich fragmentów bądź zgłoszeniu błędu, serwis wraca do stanu `WAIT_REQUEST_STATE`, zachowując ciągłość działania.

³kod w UTF-8: 23

⁴kod w UTF-8: 4

7.2 Zachowanie użytkownika



7.2.1 WAIT_FOR_PART_STATE

Użytkownik przechodzi do tego stanu po wygenerowaniu klucza zapytania i wysłaniu go do serwisu w komunikacie `CITE_REQUEST_MSG`. Znajduje się w nim aż do wystąpienia jednej z akcji:

- otrzymania wszystkich fragmentów cytatu,
- odebrania komunikatu o błędzie `SERVER_ERR_MSG`,
- otrzymania niewłaściwie zakończonego fragmentu bądź
- czas oczekiwania między wysłaniem żądania o cytata a otrzymaniem pierwszego fragmentu albo czas oczekiwania na kolejny kawałek osiągnie wartość `MAX_INTERVAL_TIME`

W przypadku odebrania komunikatu `SERVER_ERR_MSG`, numer `errno` przekazywany jest warstwie wyższej. Jeżeli otrzymany fragment, nie będący ostatnim, nie zawiera na końcu znaku ETB bądź ostatni segment nie kończy się znakiem EOT, wówczas zgłaszany jest błąd warstwie wyższej wraz z indeksami brakujących fragmentów. Podobne zachowanie występuje po przekroczeniu czasu oczekiwania. Jeżeli błąd wystąpi przed otrzymaniem pierwszego poprawnego segmentu, wówczas jest on zgłaszany z `errno` ustawionym na `EBADMSG`.

Jedynymi akceptowanymi komunikatami są:

- `REPLY_MSG(key, index, parts, text)`
- `SERVER_ERR_MSG(errno)`

Pozostałe komunikaty są ignorowane.

7.2.2 HAS_ALL_PARTS_STATE

Użytkownik znajduje się w tym stanie po otrzymaniu wszystkich fragmentów cytatu. Nie są odbierane żadne komunikaty.

8 Używane numery

Stałe używane w protokole:

`SERVER_PORT` = ustalana przy kompilacji liczba całkowita z przedziału $\langle 1024, 65535 \rangle$ (domyślnie = 44444) - numer portu, na którym nasłuchuje serwis

`MAX_PART_SIZE` = ustalana przy kompilacji liczba całkowita z przedziału $\langle 1, 65491 \rangle$ ⁵ (domyślnie = 1024) - maksymalna długość w oktetach, przesyłanej w jednym pakiecie, części cytatu (bez znaku ETB/EOT)

`MAX_INTERVAL_TIME` = ustalana przy kompilacji liczba całkowita dodatnia (domyślnie = 5) - maksymalny czas w sekundach między wysłaniem żądania o cytat a otrzymaniem pierwszego fragmentu oraz, jeśli fragmentów jest więcej, maksymalny czas między otrzymaniem kolejnych kawałków cytatu

8.1 Kody błędów

`EAGAIN` = 11 - tymczasowy brak dostępu (zalecana kolejna próba)

`ENODATA` = 61 - brak cytatów w bazie danych

`EBADMSG` = 74 - brak wiadomości z danymi

`EREMOTEIO` = 121 - brak dostępu do bazy danych

`E_USR1` = 254 - błąd dla potrzeb programistów

`E_USR2` = 255 - błąd dla potrzeb programistów

⁵ograniczenie górne wynika z faktu, że jeden pakiet UDP może zawierać do 65507 oktetów danych, lecz na tekst cytatu można przeznaczyć 65491 z nich