⊰ Remote Queues Protocol ⊱

opis protokołu

Agata Hejmej
236032

15.04.2008

1. Streszczenie

Dokument zawiera specyfikację protokołu zdalnych kolejek RQP. Opisany protokół jest protokołem warstwy aplikacji i ma umożliwiać zapis do zdalnych kolejek komunikatów IPC.

Dokument obejmuje:

- cele i założenia protokołu
- format komunikatów i ich opis
- opis stanów
- podsumowanie używanych numerów.

2. Cele protokołu

Protokół RQP ma na celu rozszerzenie możliwości uniksowych kolejek komunikatów IPC. Będzie on umożliwiał zapis danych przez sieć do kolejek na systemach zdalnych.

Host nadawczy ma mieć możliwość wysłania porcji danych do kolejki o wybranym numerze, znajdującej się na jednej ze zdalnych maszyn.

Dla danego numeru kolejki nadawca najpierw utożsamia ten numer z jedną z maszyn, które taką kolejkę udostępniają, a następnie wysyła komunikaty o danym numerze tylko do tej maszyny.

3. Założenia

3.1. Podstawowe pojęcia

Nadawca - użytkownik instancji protokołu, który chce przesyłać informacje. Nadawca korzysta z interfejsu programistycznego do wysyłania danych.

Odbiorca - użytkownik odczytujący informacje z docelowej kolejki uniksowej, korzystający ze standardowego interfejsu IPC.

Agenci kanałowi - procesy, które w imieniu nadawcy i odbiorcy zajmują się odpowiednio przesyłaniem i odbieraniem danych. Dalej nazywani MCA nadającym i MCA odbierającym.
Na jednej maszynie może działać tylko jeden MCA nadający i/lub jeden MCA odbierający.

Połączenie - połączenie między nadawcą i odbiorcą. Połączenie składa się z:

* **kolejki transmisyjnej** - bufora po stronie nadawcy, implementowanego przez MCA nadającego,
* **kanału transmisyjnego** - implementowanego przez MCA nadającego i MCA odbierającego,
* **kolejki zdalnej** - kolejki uniksowej pracującej na maszynie odbiorcy.

Zestawianie - proces budowania kanału transmisyjnego między dwoma agentami.

Komunikat - porcja danych (wzbogaconą w trakcie transmisji o nagłówek protokołu) przesyłana od nadawcy do odbiorcy. Definiujemy dwa rodzaje komunikatów – „pewne" oraz „niepewne". Typ komunikatu jest ustalany w momencie wkładania go do kolejki transmisyjnej.

***komunikat pewny** - komunikat z danymi, który nie powinien zostać usuwany z kolejki transmisyjnej do chwili, gdy nie mamy pewności, że trafił bezpiecznie do kolejki docelowej lub został odrzucony.
***komunikat niepewny** – komunikat z danymi nie wymagający potwierdzania.

DEAD.LETTER.Q - specjalna kolejka, zdefiniowana dla każdego agenta, do której trafiają uszkodzone komunikaty. W przypadku jej braku komunikaty „niepewne" mają być usuwane, a komunikaty „pewne" mają pozostać w kolejce transmisyjnej, przy jednoczesnym zaprzestaniu transmisji i przerwaniu działania przez MCA nadającego.

3.2. Założenia techniczne

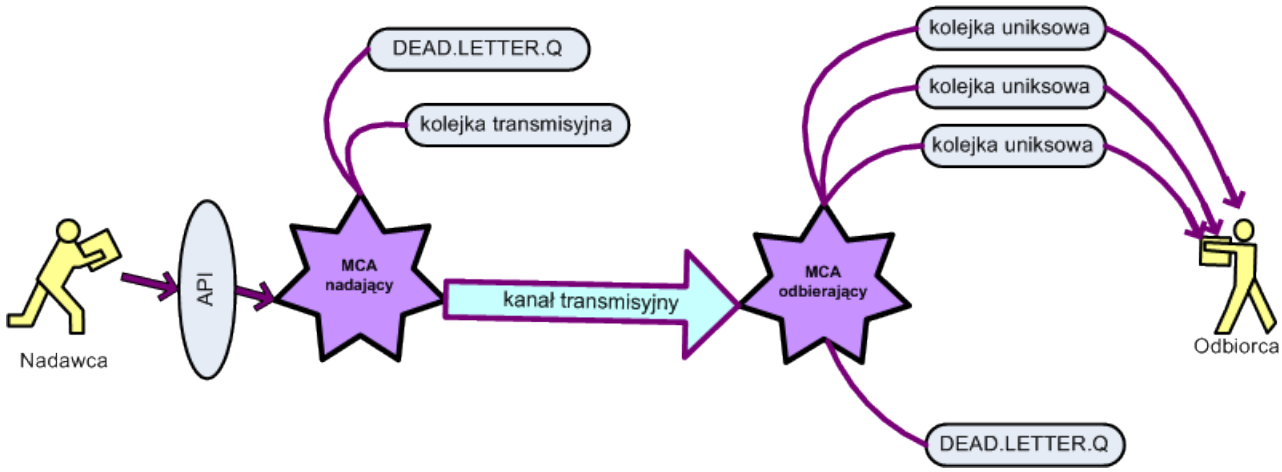
RQP to protokół warstwy aplikacji. W warstwie transportu ma on korzystać z protokołu TCP.

RQP realizuje architekturę klient-serwer, w której każdy nadawca jest klientem, a każdy odbiorca serwerem.

MCA odbierający nasłuchują na porcie 5454.

TIMEOUT, po którym MCA nadający przestaje czekać na odpowiedni komunikat od MCA odbierającego ustawiamy na 15 sekund.

3.3. Ogólny model RQP



4. Format komunikatów

Liczby w komunikatach będą przesyłane w sieciowym porządku oktetów. Uzupełnienia formatu liczb są niedopuszczalne.

Występujące komunikaty:

CZY_JEST_KOLEJKA_MSG {
uint8 typ_wiadomosci;
uint64 klucz_kolejki;
uint8 odpowiedz;
}

gdzie:

- odpowiedz może przyjmować wartości : PUSTE=0, JEST=1, JUZ_NIE_MA=2.

POTWIERDZENIE_MSG {
uint8 typ_wiadomosci;
uint64 nr_wiadomosci;
}

gdzie:

- nr_wiadomosci jest unikatowym identyfikatorem, np. kolejną liczbą naturalną dla każdej wiadomości wysyłanej przez danego MCA nadającego.

KOMUNIKAT_MSG {
uint8 typ_wiadomosci;
uint8 dl_wiadomosci;
uint64 nr_wiadomosci;
uint64 klucz_kolejki;
uint8 czy_pewny;
octet [dlugosc] wiadomosc;
}

gdzie:

- dlugosc = dl_wiadomosci – 19
  (dl_wiadomosci to ilość oktetów zajmowanych przez pola:
  typ_wiadomosci, dl_wiadomosci, nr_wiadomosci, klucz_kolejki, czy_pewny, wiadomosc,
  a 19 to ilosc oktetów zajmowanych przez wszystkie te pola oprócz pola wiadomosc)

- czy_pewny może przyjmować wartości: PEWNY=1, NIEPEWNY=0.

5. Opis komunikatów

5.1. CZY_JEST_KOLEJKA_MSG

1. Komunikat ten jest wysyłany przez MCA nadającego do wszystkich dostępnych MCA odbierających, w przypadku gdy MCA nadający nie ma ustalonej maszyny dla danego klucza kolejki (pole odpowiedz PUSTE). Jeżeli w trakcie TIMEOUT nikt nie odpowie na ten komunikat, wiadomość zostaje uznana za niepoprawną (nie ma takiej kolejki) i zostaje przeniesiona do DEAD.LETTER.Q.

2. Komunikat ten jest odsyłany przez MCA odbierającego z uzupełnionym polem odpowiedz na JEST, w przypadku, gdy posiada on dostęp do kolejki o danym kluczu.

3. Komunikat ten także wysyła MCA odbierający w przypadku gdy dostał wiadomość (KOMUNIKAT_MSG), o typie NIEPEWNY przeznaczoną do kolejki, której nie posiada (uzupełnia pole odpowiedz na JUZ_NIE_MA). Ma to na celu wykrycie, czy dany MCA odbierający posiada jeszcze kolejkę o danym kluczu. (W przypadku gdy komunikat ma typ PEWNY, do wykrycia takiej sytuacji wystarczy żądanie komunikatu POTWIERDZENIE_MSG).

Pola:

typ_wiadomosci – CZY_JEST_KOLEJKA_MSG = 1,
klucz_kolejki – numer kolejki uniksowej, do której mają trafiać wiadomości,
odpowiedz – pole mówiące, czy dany MCA odbierający ma dostęp do kolejki o danym kluczu (ma znaczenie w przypadkach 2 i 3).

5. 2. POTWIERDZENIE_MSG

Komunikat wysyłany przez MCA odbierającego, kiedy dotrze do niego poprawna wiadomość (KOMUNIKAT_MSG) z polem czy_pewny ustawionym na PEWNY.

Pola:

typ_wiadomosci – POTWIERDZENIE_MSG = 2,
nr_wiadomosci – unikatowy identyfikator wiadomości, której otrzymanie potwierdzamy.

5.3. KOMUNIKAT_MSG

Komunikat wysyłany przez MCA nadającego do ustalonego MCA odbierającego.

W przypadku, gdy nie ma ustalonego MCA odbierającego dla danego klucza kolejki, MCA nadający wysyła komunikat – zapytanie CZY_JEST_KOLEJKA_MSG do wszystkich dostępnych MCA odbierających i wstrzymuje komunikat w kolejce transmisyjnej do czasu ustalenia MCA odbierającego. Podobnie wstrzymuje kolejne komunikaty o danym kluczu (dopóki trwa ustalanie), już bez ponawiania zapytania.

W przypadku, gdy wysyłana wiadomość ma pole czy_pewny ustawione na PEWNY, wysłany komunikat jest zatrzymywany w kolejce transmisyjnej do czasu, aż MCA odbierający przyśle POTWIERDZENIE_MSG. (Jeśli NIEPEWNY nie czekamy na potwierdzenie i usuwamy komunikat z kolejki transmisyjnej).

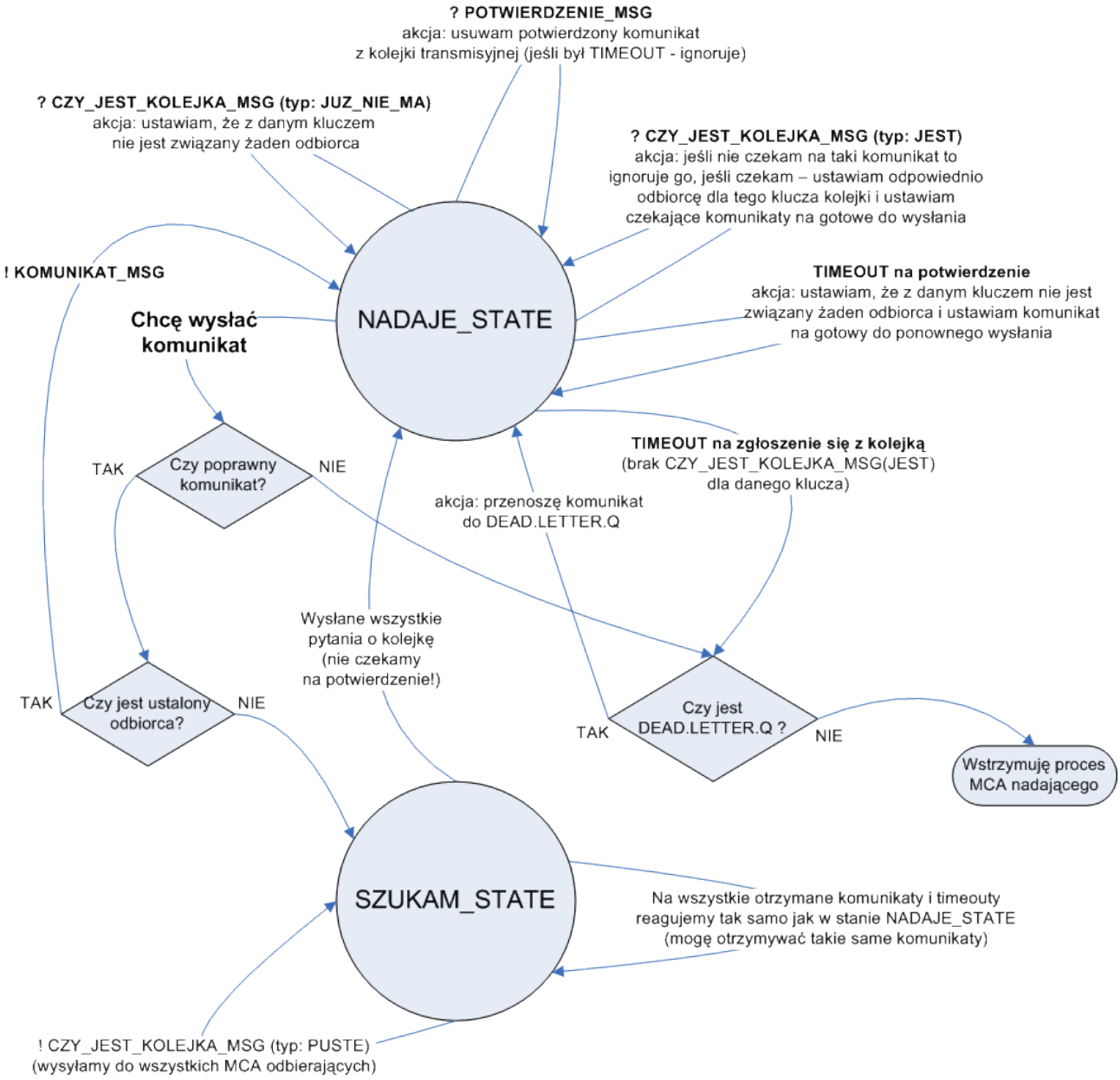
Jeżeli potwierdzenie otrzymania wiadomości nie przyjdzie zanim nastąpi TIMEOUT, wiadomość jest wstrzymywana w kolejce transmisyjnej, a MCA nadający uznaje, że kolejka, do której miała trafić wiadomość nie jest już związana z żadnym MCA odbierającym.

Pola:

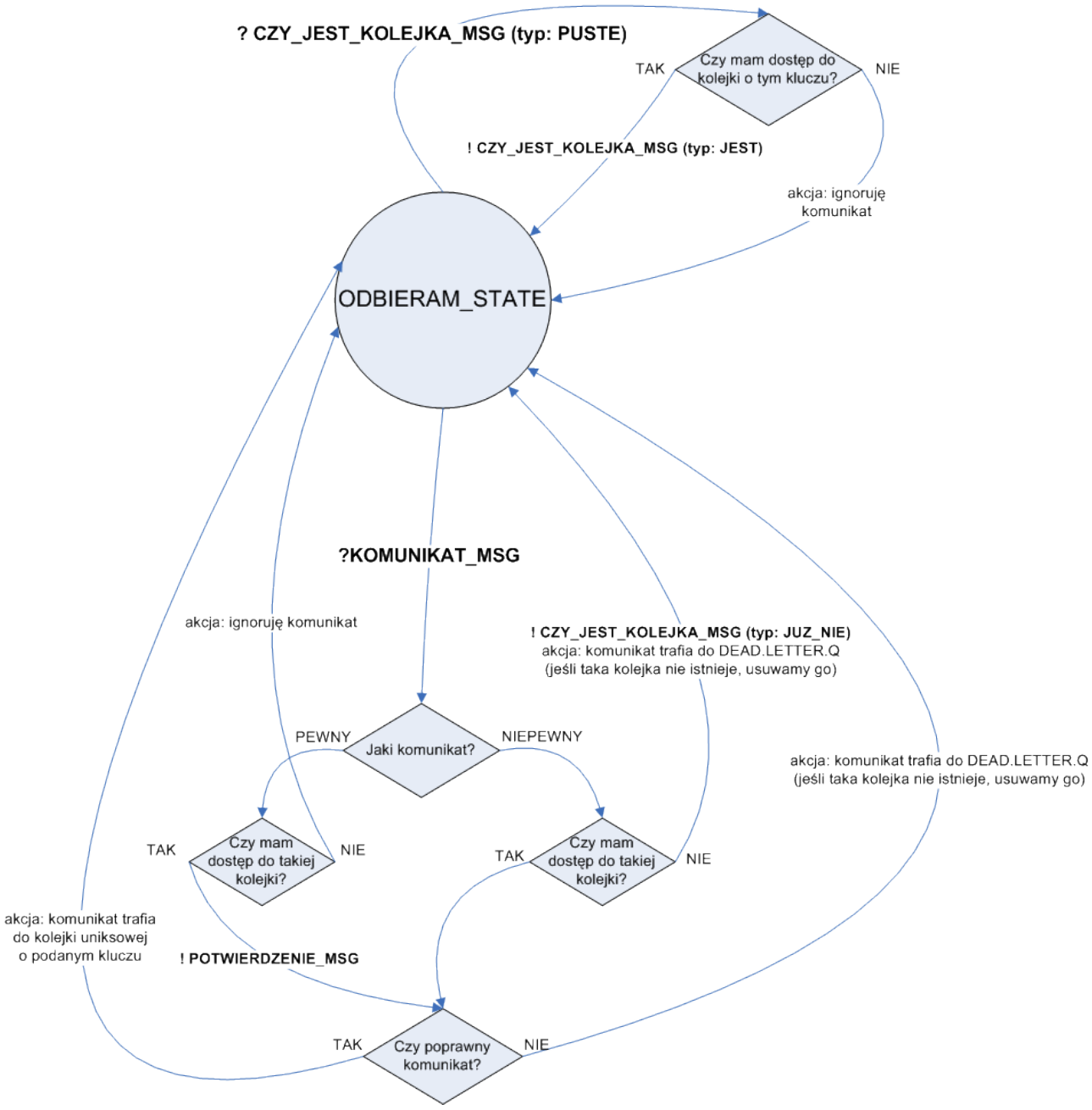
typ_wiadomosci – KOMUNIKAT_MSG = 3,
dl_wiadomosci – długość całego komunikatu,
nr_wiadomosci – unikatowy identyfikator komunikatu,
klucz_kolejki – klucz kolejki, do której ma trafić wiadomość,
czy_pewny – informacja o typie wiadomości ,
wiadomosc – wiadomość, którą nadawca chce przesłać do , zdalnej kolejki uniksowej o zadanym kluczu.

6. Opis stanów

MCA nadający



MCA odbierający



7. Podsumowanie używanych numerów

CZY_JEST_KOLEJKA_MSG = 1
POTWIERDZENIE_MSG = 2
KOMUNIKAT_MSG = 3
PUSTE = 0
JEST = 1
JUZ_NIE_MA = 2
NIEPEWNY = 0
PEWNY = 1