

## SKJ project nr.3

### UDP over TCP tunnel

Jakub Parteka

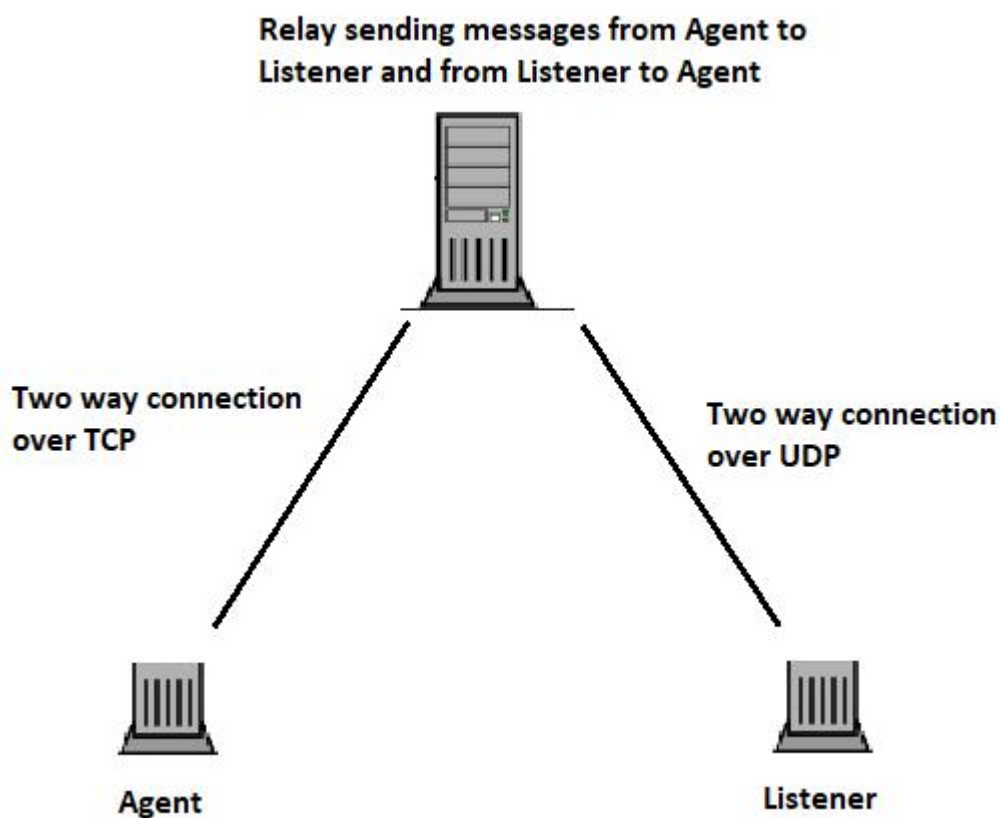
Projekt zawiera 3 znaczące klasy (nie licząc maina, który tylko uruchamia pozostałe procesy). Na początku należy uruchomić klasę Server (jest to klasa, która służy jako przekaźnik), która uruchamia 3 wątki:

1. Pierwszy wątek obsługuje tcp socket (jest na nim założony synchronized, żeby nie doszło do zakleszczenia na socketcie pomiędzy tym wątkiem, a wątkiem wysyłającym poprzez tcp socket), czeka on na połączenie od Agentu. W momencie gdy ktoś się połączy to odbiera od niego wszystkie wysłane wiadomości (do momentu odebrania wiadomości „quit”) i zapisuje je w ArrayLiście 1 (w tej liście, wiadomości czekają aż wątek drugi je „wyjmie” i przekaże je do naszego odbiorcy - Listenera).
2. Wątek drugi działa na tym samym Socketcie co wątek pierwszy (stąd również w tym momencie zostaje na socketcie założony synchronized, żeby żaden inny wątek nam się nie wtrącał) Wątek ten wyciąga wiadomości zawarte w ArrayLiście 2 (lista ta zawiera wiadomości wysłane przez Listenera poprzez UDP, które mają dotrzeć do Agentu) i wysyła je do Agentu nasłuchującego na tcp po czym usuwa je z listy, żeby uniknąć wysłania takiej samej wiadomości dwa razy.
3. Trzeci wątek zajmuje się obsługą gniazda UDP, odbiera on wiadomości od Listenera wysyłającego poprzez UDP, zapisuje je w ArrayLiscie2 (dzięki czemu wątek drugi wie co ma wysłać Agentowi). Również ten wątek zajmuje się wysyłaniem wiadomości poprzez UDP do Listenera, które są zawarte w ArrayLiście 1 (również po wysłaniu usuwa te wiadomości z listy).

Druga klasa „Agent” jest klientem łączącym się z przekaźnikiem za pomocą połączenia TCP. Dzięki klasie Server jest on w stanie komunikować się z Listenerem, wysyłać oraz odbierać od niego wiadomości. Wysyłanie oraz odbiór jest zrealizowane w dwóch osobnych wątkach, które mają w sobie założone więzy synchronized, ponieważ pracują na wspólnym Sockecie.

Trzecia klasa „Listener” jest bardzo podobną zarówno rolą jak i implementacją co „Agent”. Jediną rzeczą różniącą te klasy jest to, że Listener obsługuje UDP Socket, a Agent TCP Socket.

### Schemat zrealizowanego projektu:



Projekt ten umożliwia komunikację Agentu oraz Listenera nawet w sytuacji, w której Listener z różnych powodów (np. blokada firewalle) nie jest w stanie ustanowić połączenia z Agentem poprzez protokół TCP.

Niestety nie udało mi się zrealizować asynchronicznego wysyłania oraz odbioru wiadomości między Listenerem oraz Agentem. Zarówno Agent jak i Listener naprzemiennie wysyłają i odbierają wiadomości. Również nie udało mi się zapewnić funkcjonalności obsługi kilku użytkowników jednocześnie. Klasa Servera jest w stanie obsłużyć na raz tylko dwa połączenia (po jednym na Agentu i Listenera), stąd zapewnia komunikację tylko między nimi.