Warszawska Wyższa Szkoła Informatyki

Opracowanie protokołu komunikacyjnego na potrzeby wymiany informacji w organizacji

Robert Hryniewicz

Promotor:
dr inż. Krzysztof Różanowski

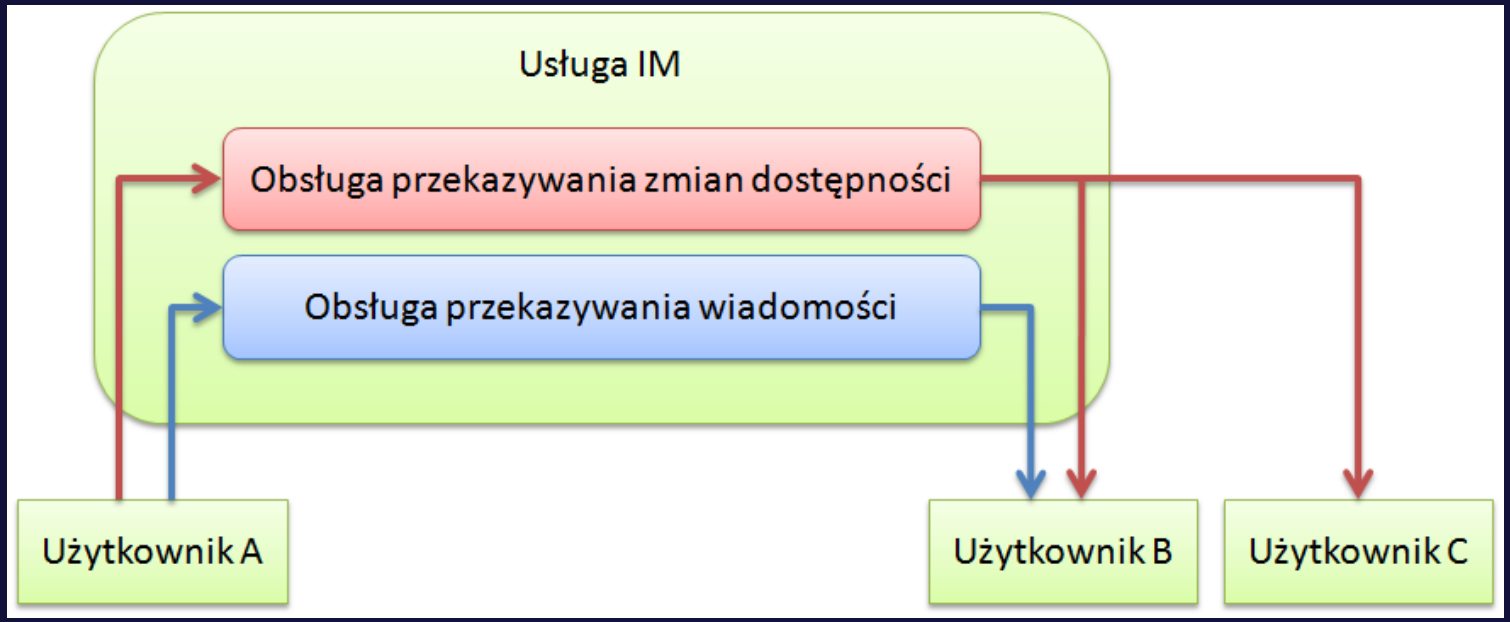
Cele pracy

- Opracowanie protokołu komunikacyjnego służącego do szybkiej wymiany informacji usprawniającego pracę grupową w dowolnej organizacji;
- Implementacja opracowanego protokołu w postaci bibliotek programistycznych oraz korzystających z nich aplikacji dla serwera i klienta;
- Analiza działania protokołu i zbadanie jego zachowania w różnych warunkach w rzeczywistym środowisku.

Zakres pracy

- Wprowadzenie do zagadnienia komunikacji w czasie rzeczywistym i przedstawienie aspektów związanych z wykorzystywaniem jej w biznesie;
- Przegląd wybranych protokołów i przeanalizowanie wymagań funkcjonalnych i pozafunkcjonalnych dla opracowywanego protokołu;
- Przedstawienie i uzasadnienie wyboru konkretnych rozwiązań oraz stworzenie specyfikacji protokołu;
- Zaprojektowanie algorytmów przetwarzania struktur danych protokołu;
- Implementacja protokołu oraz opis sposobu podziału na komponenty i zastosowanych rozwiązań programistycznych;
- Opracowanie przypadków testowych, wykonanie testów i przeanalizowanie działania protokołu na podstawie otrzymanych wyników.

Komunikacja natychmiastowa (IM)



- Usługa komunikacji natychmiastowej zapewnia przekazywanie krótkich wiadomości tekstowych pomiędzy użytkownikami;
- Usługa ta przekazuje również powiadomienia o zmianach dostępności użytkowników;
- Możliwe jest także powiadamianie użytkowników o stanie każdej wysłanej wiadomości.

Komunikacja natychmiastowa w biznesie

Dostarcza wiele korzyści usprawniając wymianę informacji, ale jednocześnie niesie ze sobą zagrożenia:

- Uzyskanie dostępu do poufnych informacji przez nieuprawnione osoby (podsłuch komunikacji);
- Rozsyłanie spamu i złośliwego oprogramowania;
- Stosowanie inżynierii społecznej do wyłudzania informacji (phishing);
- Nękanie użytkowników;
- Ujawnienie tajemnicy przedsiębiorstwa spowodowane nieodpowiedzialnym wykorzystywaniem.

Funkcje protokołu IPK

- Przesyłanie wiadomości i powiadomień o zmianach stanu ich dostarczenia do użytkowników;
- Przesyłanie wiadomości do grup użytkowników;
- Przesyłanie powiadomień o zmianach stanu dostępności użytkowników;
- Przesyłanie listy użytkowników i grup (listy kontaktów);
- Zapewnianie użytkownikom dostępu do historii wysłanych wiadomości z dowolnego miejsca (archiwizacja);
- Uwierzytelnianie użytkowników;
- Centralna administracja kontami użytkowników i grupami;
- Mechanizmy zapewniające bezpieczeństwo przesyłanych informacji;
- Kompresja przesyłanych wiadomości i list.

Zastosowane rozwiązania

- Protokół IPK korzysta z protokołu UDP jako protokołu transportowego;
- Klucze szyfrujące ustalane są podczas zestawiania połączenia i uwierzytelniania użytkownika przez protokół SRP;
- Komunikaty są szyfrowane przy użyciu AES;
- Kontrola integralności komunikatów zapewniana jest przez HMAC-SHA256;
- Kompresja algorytmem DEFLATE;
- Protokół podzielony jest na dwie podwarstwy: niższą „*warstwę zabezpieczeń”* i korzystającą z niej wyższą „*warstwę wymiany komunikatów”* (PDU nazywane jest „*komunikatem*”);
- Zawsze jeden datagram UDP przenosi jeden komunikat warstwy zabezpieczeń;
- Po odebraniu komunikatu do nadawcy wysyłana jest odpowiedź będąca potwierdzeniem odbioru.

Specyfikacja protokołu

Warstwa zabezpieczeń odpowiada za:

- Ustanowienie połączenia między klientem i serwerem;
- Uwierzytelnienie użytkownika (login i hasło);
- Bezpieczne ustalenie zestawu tajnych kluczy szyfrujących i szyfrowanie komunikatów wyższej warstwy;
- Kontrolę integralności komunikatów i powiadamianie o błędach.

Zdefiniowane typy komunikatów:

- DANE_KOMUNIKATU;
- LOGOWANIE_KLIENT;
- LOGOWANIE_SERWER;
- ZMIANA_KLUCZY (SK)/(KS);
- BŁĄD_KOMUNIKATU.

Specyfikacja protokołu

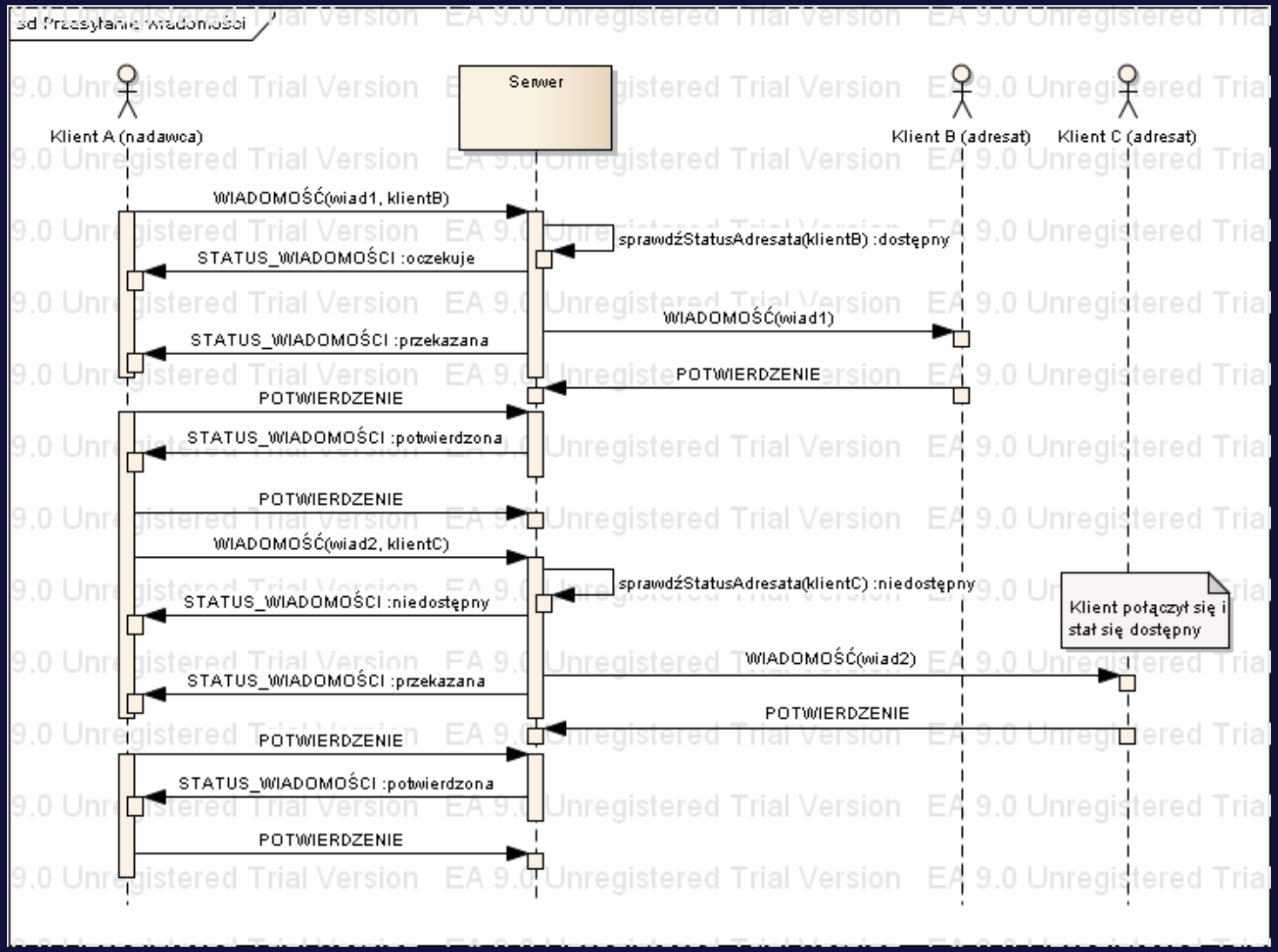
Warstwa wymiany komunikatów odpowiada za:

- Realizację wszystkich funkcji użytkowych związanych z przesyłaniem wiadomości, powiadomień i list pomiędzy klientem i serwerem;
- Dzieleniem list na fragmenty i przesyłaniem ich w oddzielnych komunikatach, gdy są zbyt długie;
- Podtrzymywaniem połączenia w trakcie bezczynności użytkownika (keepalive).

Zdefiniowane typy komunikatów:

- POTWIERDZENIE, WIADOMOŚĆ, DOSTĘPNOŚĆ;
- STATUS_WIADOMOŚCI, LISTA, PUSTA_OPERACJA;
- i inne…

Przesyłanie wiadomości



Komponenty systemu

- Język C#, platforma Microsoft .NET Framework 4.0;
- Baza danych Microsoft SQL Server 2008 Express;

IPKKomunikaty.dll, IPKKlient.dll, IPKSerwer.dll

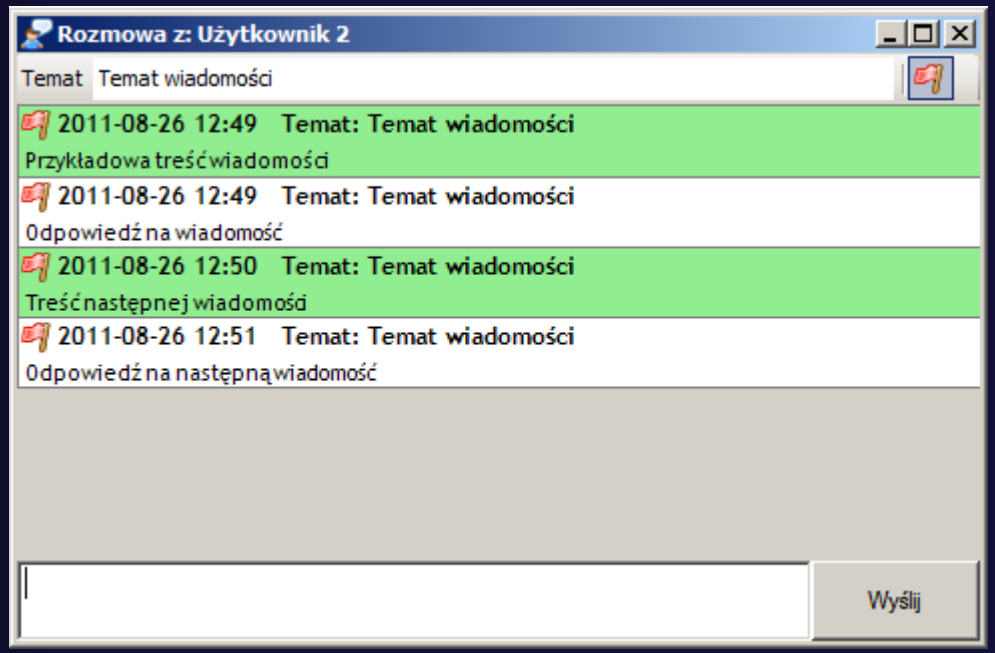
- biblioteki programistyczne, w których zaimplementowano funkcje protokołu;

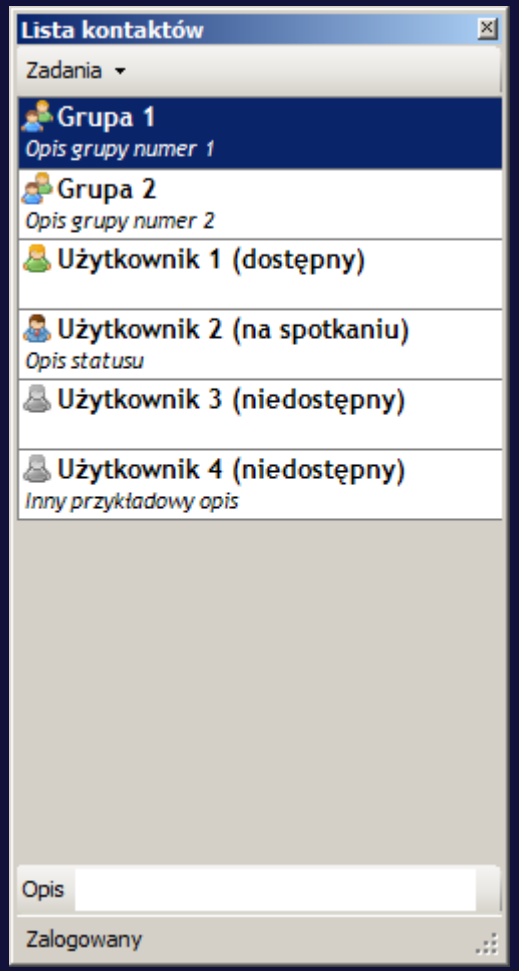
Komunikator.exe, UslugaSerweraIPK.exe, InstalatorUslugiSerweraIPK.msi, KonsolaAdministracyjna.exe

- aplikacja kliencka, usługa i jej instalator dla serwera oraz aplikacja do zarządzania użytkownikami i grupami;

- Dodatkowe zewnętrzne biblioteki (SRP, JSON, pula wątków).

Aplikacja klienta





Analiza działania protokołu

- Opracowanych zostało łącznie osiem przypadków testowych w celu sprawdzenia poprawności realizacji funkcji protokołu oraz jego zachowania w różnych sytuacjach;
- Ruch sieciowy przechwytywany był za pomocą programu Wireshark;
- Następnie odczytane zostały i zapisane w tabelach wartości poszczególnych pól każdego przesłanego przez sieć komunikatu podczas wszystkich testów.

Analiza działania protokołu

1. CEL: Zbadanie poprawności procedury zestawiania połączenia i szyfrowania danych.

   REZULTAT: Po nawiązaniu połączenia dalsze komunikaty wyższej warstwy są szyfrowane i nie jest możliwe odczytanie przesyłanych treści.

2. CEL: Zbadanie poprawności przesyłania danych w warstwie wymiany komunikatów (szyfrowanie wyłączone).

   REZULTAT: Komunikaty wymieniane są zgodnie z opracowaną specyfikacją i protokół poprawnie realizuje swoje funkcje użytkowe.

Analiza działania protokołu

3. CEL: Zbadanie zachowania protokołu w przypadku błędnego hasła użytkownika.

   REZULTAT: Serwer zgłasza błąd podczas uwierzytelniania użytkownika i zamyka połączenie.

4. CEL: Zbadanie poprawności podziału list na fragmenty.

   REZULTAT: Lista została podzielona na kilka odrębnych fragmentów, przesłana w oddzielnych komunikatach, a następnie scalona po stronie klienta.

Analiza działania protokołu

5. CEL: Zbadanie poprawności procedury zmiany kluczy w trakcie połączenia.

   REZULTAT: Po wymianie określonej liczby komunikatów serwer zażądał zmiany kluczy. Procedura została wykonana w sposób niewidoczny dla użytkownika.

6. CEL: Zbadanie działania algorytmu obliczania czasu retransmisji.

   REZULTAT: W przypadku braku odpowiedzi komunikaty są retransmitowane w poprawnych odstępach czasowych.

Podsumowanie

- Protokół działa zgodnie z opracowaną specyfikacją i poprawnie realizuje przyjęte założenia;
- Oparcie się na protokole UDP pozwoliło na stworzenie stosunkowo lekkiego i niewymagającego protokołu IM;
- Napisanie rozszerzenia dla programu Wireshark znacznie przyśpieszyłoby proces analizy działania protokołu;
- Analizując długości przesyłanych komunikatów można domyślić się jaki typ komunikatu jest prawdopodobnie przesyłany;
- Powiadamianie o zmianie stanu wysłanej wiadomości generuje znaczny ruch;
- Algorytm wyznaczania czasu retransmisji dla protokołu TCP sprawdza się również w przypadku protokołu IPK;
- Konieczne jest ciągłe weryfikowanie protokołu pod względem zapewnianego poziomu bezpieczeństwa.

Dziękuję za uwagę.