

Dokumentacja wstępna

1. Treść zadania

System komunikacji pomiędzy bramą radiową a zbiorem kamer wykonujących zdjęcia.

Kamery wysyłają do bramy zdjęcia zgodnie z ustawionymi parametrami (rozdzielczość, ogniskowa, okres i zmienność opóźnienia «*jitter*»). Zaproponować procedury instalacyjne, testowe i konfiguracyjne. W warstwie transportowej używamy wyłącznie UDP. Niezawodność obsługiwana jest w warstwie aplikacji. Należy zaprojektować moduł do Wireshark umożliwiający wyświetlanie i analizę zdefiniowanych komunikatów.

2. Nazwa projektu: Camera-Link

3. Założenia funkcjonalne i нефunkcjonalne.

Założenia funkcjonalne:

- Administrator może dokonać instalacji, testowania lub konfiguracji kamer.
- Procedura instalacyjna pozwala na sparowanie kamer z bramą radiową.
- Procedura konfiguracyjna pozwala na przesłanie parametrów działania do kamer (rozdzielczość, ogniskowa, okres i zmienność opóźnienia «*jitter*»).
- Procedura testowa pozwala na sprawdzenie poprawności parowania, wykonanie i przesłanie testowego zdjęcia na żądanie.
- Kamery cyklicznie przesyłają zdjęcia do serwera.
- Możliwość podglądu komunikatów za pomocą Wiresharka.

Założenia нефunkcjonalne:

- Do komunikacji wykorzystuje się protokół UDP.
- System działa zarówno dla adresów IPv4 jak i adresów IPv6.
- Aplikacja sprawdza kompletność danych.
- Procedura instalacyjna musi uniemożliwiać lub utrudniać podłączenie fałszywych źródeł informacji podczas parowania urządzeń (uwierzytelnianie serwera oraz kamer).
- Komunikacja między kamerami i serwerem jest szyfrowana.
- Aplikacja sprawdza poprawność parametrów komunikatów odbieranych z sieci.
- Na serwerze i kliencie znajduje się klucz publiczny serwera.

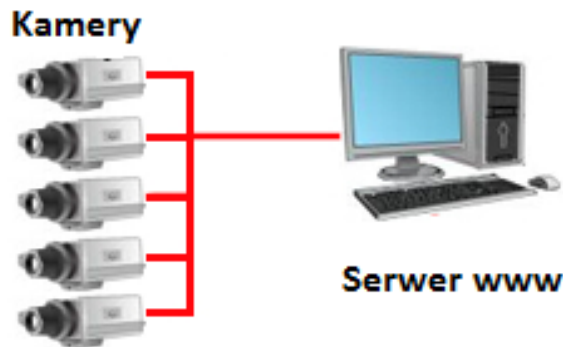
4. Podstawowe przypadki użycia.

- a. Instalacja systemu:
 - Administrator wprowadza adres IP kamery.
 - Serwer wysyła komunikat rozpoczynający połączenie do kamery.
 - Kamera odpowiada wysyłając do serwera słowo klucz zaszyfrowanym kluczem publicznym serwera.
 - Serwer odszyfrowuje przesłane słowo i jeśli się zgadza z oczekiwanym, to dodaje kamerę do swojej listy i przesyła potwierdzenie rozpoznania.
 - Kamera po otrzymaniu potwierdzenia rozpoznania, zapamiętuje adres IP serwera i przesyła komunikat potwierdzający instalację.
- b. Testowanie systemu:
 - Administrator wprowadza adres IP kamery.
 - Serwer wysyła zaszyfrowaną wiadomość testową do kamery i czeka na odpowiedź.
 - Kamera wysyła odszyfrowaną odpowiedź do serwera.
- c. Konfiguracja kamery:
 - Administrator wprowadza adres IP kamery oraz parametry konfiguracyjne.
 - Serwer przesyła parametry do kamery i czeka na potwierdzenie.
 - Kamera zapisuje parametry w pamięci i wysyła potwierdzenie konfiguracji do serwera.

5. Wybrane środowisko sprzętowo-programowe (systemy operacyjne, biblioteki programistyczne) i narzędziowe (debugowanie, testowanie).

- System operacyjny: ubuntu 16.04 wzwyż
- korzystanie z biblioteki do obsługi wątków : `pthread.h`
- korzystanie z bibliotek do testów jednostkowych Boost C++

6. Architekturę rozwiązania, tj. ilustrację i opis struktury logicznej systemu (konceptyjnych bloków funkcjonalnych).



Na serwerze pracuje jeden wątek/proces główny, który po odebraniu komunikatu od kamery (lub żądania od administratora) tworzy nowy wątek/proces który będzie obsługiwać połączenie z tą kamerą, aż do jego zakończenia lub do upływu ustalonego czasu(TIMEOUT).

7. Wstępna lista komunikatów z określeniem nadawców i odbiorców.

- a. Nawiązanie połączenia
Nadawca: Serwer
Odbiorca: Klient
- b. Uwierzytelnianie serwera
Nadawca: Klient
Odbiorca: Serwer
- c. Potwierdzenie rozpoznania kamery
Nadawca: Serwer
Odbiorca: Klient
- d. Potwierdzenie instalacji
Nadawca: Klient
Odbiorca: Serwer
- e. Test połączenia
Nadawca: Serwer
Odbiorca: Klient
- f. Potwierdzenie połączenia
Nadawca: Klient
Odbiorca: Serwer

- g. Konfiguracja
Nadawca: Serwer
Odbiorca: Klient
- h. Potwierdzenie konfiguracji
Nadawca: Klient
Odbiorca: Serwer
- i. Przesyłanie danych
Nadawca: Klient
Odbiorca: Serwer
- j. Potwierdzenie odbioru danych
Nadawca: Serwer
Odbiorca: Klient

8. Sposób testowania.

Przy każdym punkcie kontrolnym projektu będą dokonywane testy integracyjne pomiędzy poszczególnymi modułami projektu tworzonymi przez członków zespołu. W trakcie testów przesyłane komunikaty będą monitorowane za pomocą modułu Wiresharka w celu potwierdzenia ich poprawności.

Testy jednostkowe będą polegały na sprawdzeniu działania systemu dla przypadków typowych, skrajnych oraz złośliwych (takich jak np. celowo niepoprawnie wprowadzane dane, mające na celu zakłócenie działania systemu).

Testy które system przeszedł we wcześniejszym etapie projektu będą powtarzane przy dodawaniu nowych elementów, aby upewnić się, że nie powodują one nowych błędów.

9. Sposób demonstracji rezultatów, tj. scenariusze testów akceptacyjnych do zaprezentowania przy odbiorze projektu.

- Test zaniku napięcia dla serwera
- Test zaniku napięcia dla klienta
- Test obciążeniowy komunikacji między wieloma klientami a serwerem

10. Podział prac w zespole.

- Moduł do Wiresharka : Adam Bieniek
- Serwer WWW: Konrad Meysztowicz-Wiśniewski
- Klient: Maciej Puchalski
- Procedury: Łukasz Rombel

11. Harmonogram prac (minimum 1, zalecane 2 punkty kontrolne dla odbioru częściowych funkcji/modułów projektu).

- a. 17 kwietnia - wstępna wersja serwera i klienta umożliwiająca podstawowa komunikacje i testowanie przy pomocy Wiresharka
- b. 5 maja - implementacja procedury instalacyjnej i konfiguracyjnej
- c. 22 maja - złożenie wszystkich elementów systemu, końcowe testy integracyjne

12. Adres projektu na serwerze kontroli wersji.

https://github.com/koniczyn5/TIN_cameras