Warszawa, 20.11.2015r.

System dystrybucji i uruchamiania spotów reklamowych - projekt wstępny

Zespół: *Damian Ostrowski, Marek Bednarski, Cezary Grunwald, Piotr Rżysko*
Lider projektu: *Piotr Rżysko*

1. Treść zadania

W systemie pracuje stacja zarządzająca i zbiór sterowników paneli reklamowych. Stacja multicastowo dystrybuuje filmy i harmonogramy ich wyświetlania. Sterowniki unicastowo raportują swój stan i zgłaszają błędy. Błąd w transmisji multicastowej obsługiwany jest retransmisją multi- lub unicastową w zależności od liczby otrzymanych komunikatów NAK. System ma pracować w sieci IPv6 i IPv4 (bez NAT). Należy też zaprojektować moduł do Wireshark umożliwiający wyświetlanie i analizę zdefiniowanych komunikatów.

2. Nazwa projektowanego systemu

AdvertCast

3. Przyjęte założenia funkcjonalne i niefunkcjonalne

1. Założenia funkcjonalne

- Administrator serwera w pliku konfiguracyjnym będzie mógł zdefiniować listę filmów wraz z harmonogramem ich wyświetlania.
- Plik konfiguracyjny będzie w formacie JSON.
- Opis pojedynczego filmu będzie składała się ze ścieżki na dysku lokalnym serwera do pliku z filmem oraz znacznikiem czasu
- Administrator będzie mógł zdefiniować adres grupy multicastowej do której wysyłane będą filmy.
- Administrator sterowników paneli reklamowych będzie mógł zdefiniować adres serwera oraz port służący do odbierania komunikatów.

2. Założenia niefunkcjonalne

- Maksymalny rozmiar pliku z filmem to 15MB.
- Minimalna długość filmu to 1min.
- Akceptowane formaty filmów to: .avi, .mp4, .mov.
- System powinien oferować niezawodność transmisji na poziomie 90%.
- Wszystkie pracujące w systemie stacje powinny mieć zainstalowany system Linux.

- Wszystkie pracujące w systemie stacje powinny mieć dostęp do Internetu.
- Wszystkie pracujące w systemie stacje powinny mieć włączoną synchronizację czasu z Internetem.
- Oddzielne kanały komunikacji dla danych oraz komunikatów.
- Komunikaty wysyłane przez TCP.
- Wykrywanie błędów z wykorzystaniem CRC.

4. Podstawowe przypadki użycia

1. Uruchomienie serwera

1. Administrator wprowadza do pliku konfiguracyjnego dane dotyczące plików filmów i harmonogramy wyświetlania ich.
2. Administrator uruchamia program serwera podając ścieżkę do pliku konfiguracyjnego
3. Aplikacja stwierdza poprawność wprowadzonych danych
4. Administrator podaje adres grupy multicastowej
5. Aplikacja stwierdza poprawność adresu
6. Uruchamiany jest proces dystrybucji filmów

Alternatywnie:

3a. Aplikacja stwierdza niepoprawność danych
    1. Aplikacja kończy działanie

5a. Aplikacja stwierdza niepoprawność adresu
    1. Aplikacja kończy działanie

7. Wybrane środowisko sprzętowo-programowe i narzędziowe

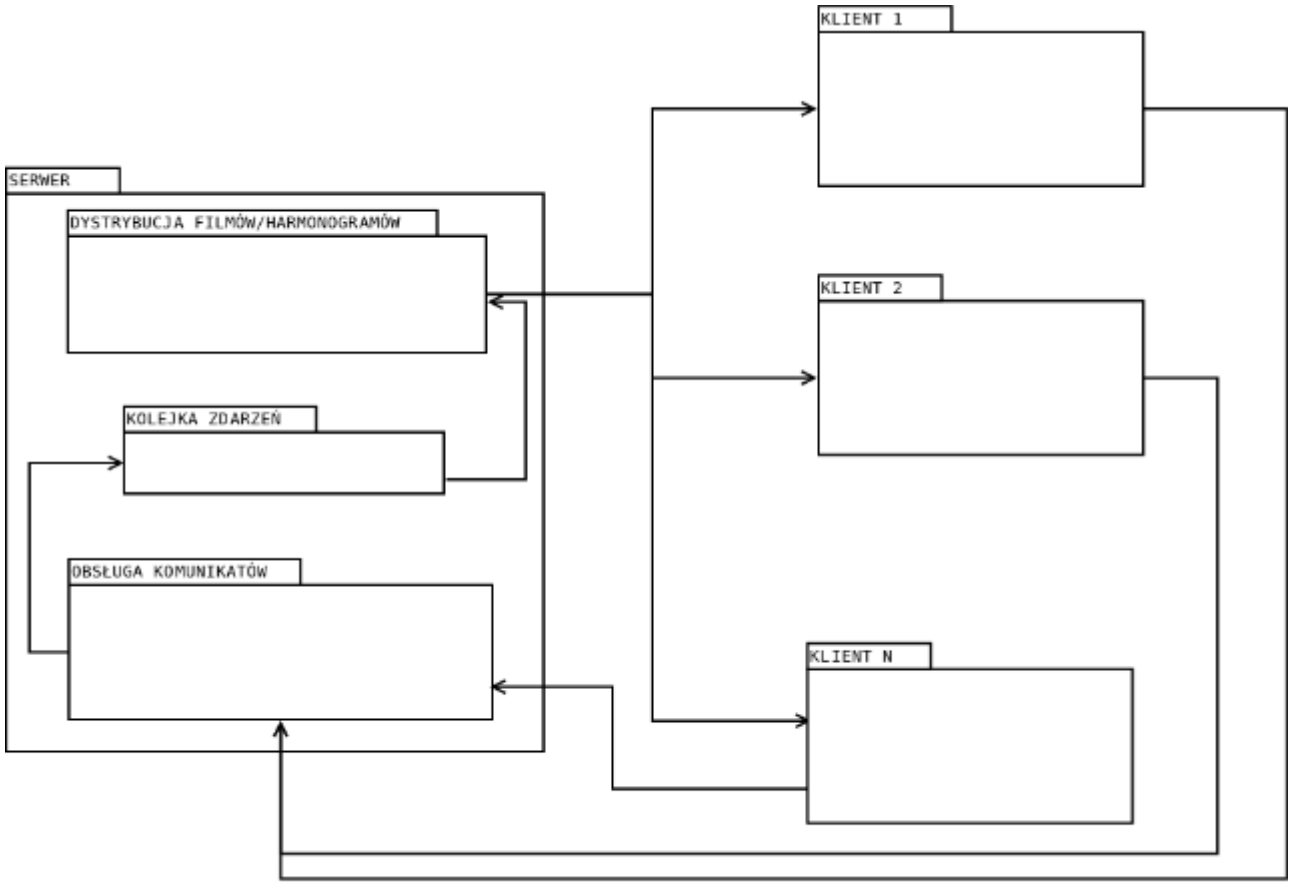
1. Środowisko sprzętowo-programowe

- system zostanie zrealizowany w środowisku uniksowym (Linux)
- aplikacja zostanie napisana w języku C++ z wykorzystaniem API gniazd BSD
- w projekcie będzie wykorzystywana biblioteka STL oraz Boost

2. Środowisko narzędziowe

- kod systemu będzie tworzony z wykorzystaniem IDE CLion
- debugowanie będzie prowadzone z wykorzystaniem narzędzi dostarczonych przez środowisko CLion oraz programu Wireshark
- do budowy aplikacji wykorzystany zostanie system CMake, który jest zintegrowany ze środowiskiem CLion (opcjonalnie może zostać wykorzystany uniksowy (g)make)

8. Architektura rozwiązania



- **Kolejka zdarzeń**

  Jest to kolejka priorytetowa, której elementami są kolejne filmy które powinny zostać wysłane do sterowników paneli reklamowych. Filmy w kolejce są uporządkowane według zaplanowanego czasu ich odtwarzania. Początkowo w kolejce znajdują się filmy które zostały wprowadzone jako konfiguracja serwera.

- **Obsługa komunikatów**

  Moduł ten odpowiedzialny jest za odbiór i przetwarzanie komunikatów pochodzących od sterowników paneli reklamowych.
  Moduł ma za zadanie oczekiwać przez określony czas (ustalony podczas testów) na komunikaty od wszystkich sterowników. Jeżeli po ww. czasie nie uzyska odpowiedzi od danego sterownika uznaje go jako nieaktywny i w kolejnych iteracjach ignoruje komunikaty od niego.
  Moduł na podstawie liczby otrzymanych komunikatów NAK decyduje w jaki sposób należy obsłużyć retransmisję. W przypadku wystąpienia błędów dodaje do kolejki film do wysłania z informacją o sposobie retransmisji.

Po wykonaniu ww. działań moduł wznawia działanie dystrybucji filmów i harmonogramów, sam natomiast zostaje zawieszony.

- **Dystrybucja filmów/harmonogramów**

  Moduł ten odpowiedzialny jest za dostarczenie do zdefiniowanej w elemencie kolejki grupy odbiorników filmu który znajduje się na początku kolejki.
  Moduł powinien odrzucać filmy których czas wyświetlania już minął. W przypadku braku filmów w kolejce kończy działanie serwera.
  Po wysłaniu filmu wznawia działanie obsługi komunikatów, sam natomiast zostaje zawieszony.

- **Klient**

  Aplikacja klienta będzie odpowiedzialna za odbieranie filmów oraz sprawdzanie błędów transmisji. W przypadku wystąpienia błędu będzie wysyłała komunikat NAK do serwera przez specjalnie przygotowany kanał. W przypadku braku błędu musi wysłać komunikat ACK.

9. Sposób testowania

Program testowany będzie z pomocą serwisu TravisCI. Będą to testy jednostkowe napisane dla każdego elementu składowego z pomocą biblioteki Boost (konkretnie Boost.Test). Wstępnie planowane jest także badanie pokrycia kodu testami z pomocą biblioteki gcov. Testy będą uruchamiane dla każdego commita.

10. Sposób demonstracji rezultatów

Testy akceptacyjne będą polegały na uruchomieniu całego systemu w środowisku produkcyjnym i przetestowania kluczowych funkcjonalności:

- poprawności transmisji
- przestrzegania harmonogramów
- poprawnej interpretacji plików konfiguracyjnych przekładającej się na właściwe wprowadzenie harmonogramu (wraz z plikami) do systemu

11. Podział prac w zespole

Projekt jest jeszcze na zbyt wczesnym etapie by móc przypisać konkretne osoby do konkretnych modułów. Można jednak wyszczególnić pewne ogólne zadania oraz role w zespole.

- Damian Ostrowski
  - Organizacja dokumentacji kodu
  - Programowanie modułów stacji zarządzającej oraz sterowników

- Marek Bednarski
  - Organizacja testów, obsługa CI
  - Programowanie modułów stacji zarządzającej oraz sterowników

- Cezary Grunwald
  - Organizacja dokumentacji projektowej
  - Programowanie modułów stacji zarządzającej oraz sterowników

- Piotr Rżysko
  - Organizacja pracy zespołu
  - Moduł do Wireshark umożliwiający wyświetlanie i analizę zdefiniowanych komunikatów.
  - Programowanie modułów stacji zarządzającej oraz sterowników

12. Adres projektu na serwerze kontroli wersji

https://github.com/piotreq53/TIN

Powyższy ustalenia są ustaleniami wstępnymi i zastrzegamy sobie prawo do zmian w trakcie rozwoju projektu.