Poprawa bezpieczeństwa w komunikacji miejskiej przy zastosowaniu serwera obrazu

JEL: R41 DOI: 10.24136/atest.2018.359

Data zgłoszenia: 19.11.2018 Data akceptacji: 15.12.2018

W artykule omówiony został aspekt poprawy bezpieczeństwa w komunikacji miejskiej przy synchronizacji standardowego monitoringu spotykanego w autobusach miejskich z centrum powiadamiania o sytuacjach niebezpiecznych – dyspozytornią, policją lub innymi służbami. Kierowca za pomocą wydzielonego przycisku może uruchomić powiadomienie oraz rozpocząć transmisję online obrazu lub obrazu i dźwięku do służb odpowiedzialnych za bezpieczeństwo. Zapewnić to ma zwiększenie ogólnego poziomu bezpieczeństwa w komunikacji publicznej.

Słowa kluczowe: monitoring, bezpieczeństwo, transmisja danych.

Wstęp

Artykuł prezentuje koncepcję opracowania mobilnego systemu monitoringu komunikacji miejskiej, który zapewnia możliwość podglądu online bieżącej sytuacji w wybranym środku transportu. Temat został częściowo skonsultowany z kierownictwem Miejskiego Przedsiębiorstwa Komunikacji w Lublinie, którego autobusy posiadają wewnętrzny monitoring. Jednak w dotychczas stosowanym systemie monitoringu obraz może być odtworzony dopiero po powrocie autobusu do zajezdni [3].

Tematyka zwiększania bezpieczeństwa jest zagadnieniem ciągle aktualnym. Zmieniające się ciągle sytuacje geopolityczne i związane z tym widoczne migracje ludności skłaniają do zastosowania w komunikacji publicznej ciągłego monitoringu, który pozwoli na bieżącą ocenę sytuacji, pozwalając na podjęcie odpowiednio szybkiej reakcji na ewentualne niebezpieczeństwo.

W czasie przeprowadzonych konsultacji z Zarządem Miejskiego Przedsiębiorstwa Komunikacji w Lublinie ustalono, że aktualna metoda rejestrowania obrazu wnętrza autobusów nie spełnia wymagań zapewniających optymalne warunki bezpieczeństwa pasażerów. Zaproponowane rozwiązanie może stać się także pomocne w ograniczaniu sytuacji wandalizmu i niszczenia urządzeń w pojazdach MPK oraz podwyższyć poczucie bezpieczeństwa wśród pasażerów korzystających z komunikacji miejskiej [3].

Podstawowym założeniem projektu mobilnego monitoringu jest wykorzystanie architektury klient-serwer, gdyż serwer jest systemem pasywnym, oczekującym na żądania klienta, które przetwarza i następnie wysyła odpowiedź. Klientem w tym systemie jest serwer webowy, który nawiązuje połączenie z platformą komputerową, na której zaimplementowana jest usługa pełniąca rolę serwera. Strukturę proponowanego systemu monitoringu przedstawiono na rysunku 1. Użytkownik loguje się na serwer WWW, który zarazem pełni rolę klienta usługi mobilnej. Po wybraniu numeru pojazdu, klient łączy się z serwerem tej usługi, który po prawidłowym zalogowaniu się rozpoczyna czynności związane z obsługą kamery oraz strumieniowaniem obrazu.

Projekt mobilnego monitoringu komunikacji miejskiej dotyczy przede wszystkim struktury systemu, wymagań niezależnych od sys-

temu, tj. niefunkcjonalnych, a także wymagań funkcjonalnych. Umożliwia to właściwy wybór platformy komputerowej, stanowiącej najistotniejszy element mobilnego monitoringu.

Rys. 1. Projekt struktury systemu mobilnego monitoringu komunikacji publicznej

Propozycja rozbudowy systemu monitoringu o omawiane funkcje związana jest także z faktem, że aktualne używane do rejestracji dyski magnetyczne, na które zapisywany jest obraz z kamer, ulegają uszkodzeniu pod wpływem drgań w pojazdach podczas jazdy. Unie-możliwia to późniejsze odtworzenie zarejestrowanego materiału, który niekiedy stanowi jedyny dowód zaistniałych zdarzeń. Zdarza się, że nagrania z komunikacji miejskiej są wykorzystywane w postępowaniach prowadzonych przez służby porządkowe. Aktywacja monitoringu w czasie rzeczywistym może pomóc chociażby policji w zlokalizowaniu poszukiwanych sprawców zgłoszonych przestępstw.

1. Przegląd wybranych platform sprzętowych i programowych

Wybór odpowiedniego urządzenia oraz konkretnego modelu sprzętu stanowi często fundamentalne podłoże postawionych wymagań wydajności w przetwarzaniu obrazu, a także w przesyłaniu go do dedykowanej kamery. Na podstawie dostępnych wymagań dotyczących systemów monitoringu w komunikacji publicznej [9] system taki powinien spełniać minimalne wymagania:

- platforma powinna przetwarzać obraz z jednej kamery w rozdzielczości nie mniejszej niż 1280 x 720 pikseli w kolorze [6],
- przesyłanie cyfrowego sygnału wideo z najlepszą jakością obrazu ma być możliwe tylko przy najniższym współczynniku transferu danych. Aby uzyskać ten proces, kompresja próbkowania musi osiągnąć poziom YUV (Y-luminancja; U, V-dwa poziomy chrominancji) 4:2:0 bez wielokrotnego kopiowania i sklejanie tych samych ramek [5]. W związku z tym system powinien mieć możliwość kompresji 4:2:0 YUV, by strumieniowanie klatek poprzez sieć i przeglądarkę internetową odbywało się bez zakłóceń,
- ze względu na minimalizację kosztów wdrożenia systemu, oprogramowanie samego urządzenia monitoringu powinno być możliwie bezpłatne – platforma powinna być wyposażona w ogólnie dostępny system operacyjny.

Systemy sprzętowe ze względu na obszary zastosowania w monitoringu muszą wywodzić się z obszarów, gdzie zostały już odpo-

wiednio przetestowane: wykazały się stabilnością i możliwie bezawaryjnym działaniem przez zdany okres czasu. Błędy w nowym oprogramowaniu mogą być katastrofalne w skutkach dla odczytu danych poddawanych obróbce, zbieraniu danych oraz ich późniejszym przechowywaniu. Takie podejście narzuca dość wysokie wymagania co do projektów systemów operacyjnych wbudowanych, jakie mogą być zastosowane w poszukiwanych urządzeniach do mobilnego monitoringu.

Powyższe czynniki mają na celu przybliżenie problematyki projektowania systemów wbudowanych, jakie mają funkcjonować w monitoringu wizyjnym.

1.1. Dostępne platformy sprzętowe

Rodzina procesorów DaVinci

Pierwszą rodziną układów, jakie brane są pod uwagę przy wyborze, są procesory DaVinci (Texas Instruments). Jest to kompilacja rdzenia ARM9 (*ang. Advanced RISC Machine*) i rdzenia DSP (*ang. Digital Signal Processing*) w jednej obudowie, która umożliwia szybkie kodowanie i dekodowanie sygnału wideo dzięki zastosowaniu bezpośredniej magistrali danych łączących wspomniane oba rdzenie. Przy zastosowaniu tego rozwiązania powstaje problem wynikający z założeń do proponowanego systemu: osobne programy, potrzebne do kompilacji kodu niskopoziomowego nie są bezpłatne, co pociąga za sobą zwiększenie kosztów opracowywanego projektu.

Mikrokomputery ARM

Kolejną rodziną mikrokomputerów branych pod uwagę jako potencjalna platforma sprzętowa jest rodzina mikrokomputerów ARM. Ich zaletą jest dostępność darmowego oprogramowania przy stosunkowo dużej wydajności najnowszych układów z tej rodziny. Kolejnym aspektem brany pod uwagę przy wyborze platformy sprzętowej jest niska awaryjność oraz konieczność serwisowania urządzenia. Komputery z rodziny ARM, powinny być odpowiednio chłodzone. Aktualnie dostępne aktywne chłodzenia dla tej platformy generują także koszty, a ich gabaryty zwiększają zapotrzebowanie na miejsce w porównaniu z systemami dedykowanymi dla tej platformy. Po dokonaniu pierwszych testów mikrokomputery ARM zostały odrzucone ze względu na duży pobór mocy oraz wspomniane wcześniej dodatkowe koszty.

Platforma Raspberry Pi

Platforma Raspberry Pi, jaka jest zastosowana w procesorze o architekturze ARM7 i układy SoC (*ang. System-on-a-chip*), zawierające mikroprocesor Broadcom BCM2835, są szeroko stosowane wszędzie tam, gdzie wymagany jest niski pobór energii, przy stosunkowo dużej mocy obliczeniowej. Dzięki możliwości wykonywania obliczeń w sposób równoległy, układy BCM2835 dobrze nadają się do implementacji algorytmów przetwarzania obrazu. Zaletą jest duży wybór języków programowania (m. in.: Java, C, C++, C#, Asembler, Python), a narzędzia wspomagające programowanie nie generują dodatkowych kosztów. Dużym udogodnieniem jest powszechna dostępność dokumentacji, co znacząco przyspiesza etap prototypowania. Architektura ARM - wielu producentów mikroprocesorów oferuje swoje implementacje z rdzeniem BCM2835. Charakteryzują się one dużą wydajnością obliczeniową w porównaniu do innych rodzin mikroprocesorów. Dużą zaletą jest także dostępność darmowych narzędzi programistycznych (np. GNU GCC) oraz oprogramowania typu Open Source, które są dostępne na powyższej platformie. Dostępne na platformie procesory Arduino są procesorami uniwersalnymi. Nie posiadają sprzętowych kodeków, trzeba je zaimplementować i uzupełnić o dodatkowe skrypty wspomagające obsługę przetwarzania strumienia wideo.

Biorąc pod uwagę wyżej wymienione wymagania projektowe oraz analizę przedstawionych wcześniej platform sprzętowych, zdecydowano, że projektowany system będzie zbudowany w oparciu o procesor ARM7 firmy BROADCOM - Raspberry Pi przedstawionej na rysunku 2.

Rys. 2. Ogólny widok platformy Raspberry Pi2 Model B

Sercem wybranego urządzenia jest procesor Broadcom BCM2836 z czterema rdzeniami ARM Cortex-A7. Układ graficzny platformy to VideoCore IV, z wbudowaną pamięcią RAM 1 GB. W urządzeniu znajdują się również 4 porty USB, złącze GPIO, port microUSB 2.0, służący do zasilania, wyjście HDMI oraz złącza CSI i DSI [4].

1.2. Wybrane aspekty oprogramowania

System operacyjny dla Raspberry Pi

Wybrany systemem operacyjnym została jedna z dystrybucji Linuksa. Debian Linux został napisany specjalnie dla komputerów klasy PC, dla firmy Intel. Wykorzystuje architekturę procesorów wspomnianej firmy. Dzięki jądrze Debiana zaszyte sterowniki umożliwiają pracę na platformie Raspberry Pi2, która obsługuje architekturę procesorów ARM.

Zaletą jest także obsługa pamięci RAM (*ang. random-access memory*) przez ten system. Linux używa tak zwanego stronicowania na zadanie, polegającego na dynamicznej alokacji pamięci RAM w stosunku do potrzeb aplikacji. Aby wydajniej wykorzystać pamięć RAM, Linux używa całej wolnej pamięci na pamięć podręczną dla dysku, skracając czas dostępu do programów i danych. Rozmiar pamięci podręcznej, jak już wcześniej zostało wspomniane, jest dynamicznie zmniejszany wprost proporcjonalnie do zajętości pamięci przez aplikacje. Wybrany tu Linux Debian w wersji Wheezy pozwala na używanie bibliotek współużytkowanych dynamicznie, obsługuje wiele różnych systemów plików, w tym kompatybilne z systemem DOS i OS/2. Do wersji Debiana Wheezy dołączono dodatkowe biblioteki, umożliwiające kompilowanie na poziomie języków C, C++, Fortran, Pascal, LISP, Ada, BASIC oraz Smalltalk, co daje duże możliwości w pisaniu skryptów i programów pod różne platformy [8].

Strumieniowanie obrazu

Przesyłanie obrazu tak zwanym strumieniem wymaga dużego pasma dla przesyłanych danych. Jest odporne na zakłócenia utraty ramek informacyjnych, szczególnie dla kodeka kompresji obrazu JPEG (*ang. Joint Photographic Experts Group*), który jest tu proponowany. MJPEG (*ang. Motion JPEG*) należy do rodzaju kodeków, gdzie kolejne, poszczególne poddawane są pojedynczo kompresji i kodowaniu standardowym algorytmem JPEG. Przez to jedna ramka może być rozpatrywana niezależnie i jest pozbawiona uzależnienia od innych ramek, co stanowi bardzo ważny czynnik, jaki skłania do

wyboru tego kodeka. Równie istotnym aspektem jest bezpłatna dostępność – kodek używa standardowego i otwartego kodu JPEG, który obsługiwany jest przez wszystkie wiodące przeglądarki internetowe. Obniża to w znacznym stopniu koszty przy produkcji i wdrożeniu projektu [7].

Klatki w MJPEG są kompresowane niezależnie, więc nie jest wymagana przy tym duża moc obliczeniowa i wykorzystywana jest mała wielkość pamięci operacyjnej na etapie kodowania. Kodowanie przy wykorzystywaniu różnych transformat jest współczesną techniką stratnego kodowania obrazów i sekwencji wizyjnych. Zasada kodowania kanałów głównych, na którym opiera się kodowanie obrazów za pomocą transformaty, zakłada, że po wykonaniu na bloku treści obrazu pewnej transformacji wielowymiarowej, energia sygnałów w nowej dziedzinie skupia się w niewielkiej liczbie kanałów, to znaczy współczynników takiej transformacji. Wystarczające jest kodowanie i przesyłanie tylko współczynników, których wartości są znacznie przybliżone do odtworzonej treści filmu przesyłanego jednak pojedyncze zdjęcia sklejone są wysokiej jakości [4,5].

2. Projekt systemu opartego o platformę Raspberry Pi2

Na wybranej platformie sprzętowej testowane były różne biblioteki wspomagające kompresję obrazu:

- libjpeg - najpopularniejsza biblioteka kompresji obrazu dla środowiska Linux. Pozwala się skompilować na Raspberry Pi2 i jest wygodna w użyciu. Niestety prędkość jej pracy jest bardzo mała. Kompresja pojedynczej ramki JPEG w formacie 4:2:2 i rozdzielczości D1 (720 x 576) trwa średnio 0,8 sekundy co dyskwalifikuje to rozwiązanie dla potrzeb proponowanej aplikacji przesyłania obrazu ruchomego [1, 6],
- Ffmpeg/libavcodec - pakiet ten zawiera kilkanaście kodeków wideo. Testowana była wersja zoptymalizowana pod kątem procesora ARM. Przetestowany został kodek MJPEG oraz MPEG4 (ang. *Moving Picture Experts Group*). Okazuje się, że najwyższa wydajność kodeka MJPEG to 8 klatek na sekundę w formacie koloru 4:2:0. Kodek MPEG4 umożliwiał kompresję obrazu z prędkością 6 klatek na sekundę w rozdzielczości 320 x 240 pikseli [1, 10], co również nie jest zadowalające dla potrzeb niniejszego systemu.
- libjpeg62-dev - jedna z bardzo popularnych bibliotek. Umożliwia dobre dopasowanie do wybranej platformy i systemu operacyjnego, jakim jest w badanym przypadku Raspbian, ponieważ posiada biblioteki pod architekturę procesorów ARM, wykorzystywanych w Raspberry Pi2. Biblioteka przetestowana była pod kątem szybkości i szerokości pasma internetowego. Główny czynnik to mała obciążalność obliczeniowa procesora [10]. Testowana przy rozdzielczości 1280 x 720 pikseli przy prędkości przesyłu 25 klatek na sekundę wykazała się obciążeniem rzędu 12% pracy zasobu mikrokontrolera.

2.1. Usługa serwerowa na Raspberry Pi2

Na skonfigurowanej platformie Raspberry Pi2 umieszczonej w pojeździe komunikacji miejskiej z podłączoną kamerą będącą częścią standardowego monitoringu pojazdu, zaimplementowana została usługa serwerowa, w tym wypadku mobilna, ponieważ jej komunikacja na zewnątrz odbywa się za pośrednictwem sieci LTE (ang. *Long Term Evolution*) [2]. Komunikacja usługi z serwerem WWW odbywa się za pomocą protokołu HTTP, metodą POST. Głównym modulem jest klasa rozszerzająca HttpServlet. Servlety są modułami, które uruchamiane są wewnątrz serwerów przetwarzających zapytania i generujących odpowiedzi, takich jak np. rozszerzone o obsługę Javy serwery WWW. Rozszerzają one funkcjonalność tych serwerów. Servlety są ponadto szybko wykonywane, ponieważ wywołanie ich nie odbywa się poprzez uruchomienie nowego procesu, co okazuje się

kosztowne ze względu na zajmowany czas pracy procesora oraz zasoby pamięciowe, lecz jako wątek. Kod wykonywalny dla servletu jest ładowany do serwera tylko raz w chwili, gdy po raz pierwszy żądana jest usługa oferowana przez dany servlet lub automatycznie, kiedy zostanie zmieniony kod servleta. Następnie servlet pozostaje w pamięci serwera i może obsługiwać równolegle wiele zapytań i umożliwiać komunikację pomiędzy nimi [3].

Transmisja strumienia jest podtrzymywana okresowo. Jeżeli nie nastąpi jej podtrzymanie, następuje zatrzymanie usługi.

W komunikacji między serwerami Raspberry Pi2 a serwerem WWW wykorzystywany jest format JSON (ang. *JavaScript Object Notation*). W związku z tym zachodzi konieczność przetworzenia obiektu JSON na obiekt Javy i na odwrót.

2.2. Moduł obsługi systemu od strony użytkownika zewnętrznego

Użytkownik, do którego docierają komunikaty z systemów mobilnych, które w chwili wywołania sygnalizują i przekazują obraz ze zdarzeń nagłych w pojeździe lub wywołanych zagrożeń obsługuje swoją część systemu, która obsługiwana jest przez przeglądarkę internetową (Rys.3).

Numery autobusów

2279	2280	2281
2282	2283	2284
2285	2286	2287
2288	2289	2290
2291	2292	2293
2294	2295	2296
Użytkownicy		
Wyślij		

Rys. 3. Wygląd strony po zalogowaniu z numerami poszczególnych pojazdów komunikacji miejskiej

Moduł komunikacji z mobilnym serwerem, strumieniującym obraz podczas wywołania przez operatora systemu zewnętrznego, jest odpowiedzialny za nawiązanie, podtrzymanie i zakończenie transmisji. Nawiązanie komunikacji odbywa się poprzez wysłanie komunikatu wywołania wraz z zapamiętanym kluczem API (ang. *Application Programming Interface*) i danymi uwierzytelnienia użytkownika. Po poprawnej weryfikacji otrzymywany jest od serwera token, który jest dołączany do każdego kolejnego wysyłanego komunikatu. Następnym etapem jest wysłanie polecenia inicjującego strumieniowanie sygnału wizyjnego. Niezbędne jest podtrzymywanie transmisji poprzez wysyłanie do serwera komunikatów w interwałach dziesięciosekundowych, gdyż w przeciwnym przypadku zostanie ona zakończona.

Dodatkową funkcjonalnością, która została zaimplementowana jest moduł umożliwiający przeglądanie historii komunikatów na wybranym serwerze usługi strumieniowania obrazu - moduł dziennika zdarzeń. Klient wysyła żądanie do wybranej mobilnej platformy zainstalowanej na wywoływany pojeździe komunikacji miejskiej, po czym otrzymuje historię logów. Pojedynczy wpis zawiera czas zarejestrowanego polecenia, nazwę polecenia, żądanie i odpowiedź, a także wyjątek, o ile wystąpił.

Widok jaki otrzymuje operator po zgłoszeniu od kierowcy zdarzenia przedstawia rysunek 4.

Autobus nr 2279

Rys. 4. Strona podglądu wnętrza wywołanego autobusu w czasie rzeczywistym

Do testów sprawdzających sprawność strumieniowania kodeka MJPEG został użyty standardowy modem LTE Huawei E-3372, pozwalający na transfer teoretycznie od 100 do 150 Mb/s. Wykorzystywany mobilny monitoring w komunikacji miejskiej przez operatorów systemu wymaga przydzielenia pasma między autobusy a serwer macierzysty, przechwytyjący obraz. Ograniczenia, jakie wynikają z zastosowanej modulacji, warunki techniczne jakie będą występować i narzucać wybór parametrów w transmisji strumienia obrazu o kompresji MJPEG będą stanowiły główne czynniki jakości przesyłanego obrazu wideo [3].

Podsumowanie

Testy dwóch bibliotek (libjpeg, Ffmpeg/libavcodec) okazały się niezadowolające z powodu małej prędkości działania. Konieczna okazała się optymalizacja gotowych bibliotek pod kątem zastosowanej architektury sprzętowej i szerokości pasma medium internetowego, potrzebnego do sprawnego strumieniowania sygnału video. Biblioteka libjpeg62-dev dobrze została dopasowana do potrzeb projektowanego i częściowo testowanego systemu.

Mobilny monitoring w komunikacji miejskiej projektowany i testowany jest etapami. Pierwszym było przedstawienie założeń projektowych oraz wymagań systemu. Na tej podstawie oraz na podstawie analizy możliwości dostępnych platform komputerowych, dokonano wyboru sprzętu. Następnie zaimplementowano usługę na serwerze mobilnym oraz klienta na serwerze WWW. W trakcie przygotowywania projektu zostało podjętych kilka ulepszeń, związanych z mobilnością platformy Raspberry Pi2 oraz z podłączeniem do mobilnego Internetu.

Zaprojektowany mobilny monitoring może zostać wdrożony w przyszłości jako system wspierający pracę dyspozytora komunikacji miejskiej. Praktyczną stroną projektowanego systemu jest użycie urządzenia Raspberry Pi2 w roli serwera wideo, wysyłającego do urządzenia klienckiego strumień wizyjny, pochodzący z zainstalowanych wewnątrz pojazdu kamer wideo. Jak wskazano możliwe jest poddanie transmisji kompresji i wysyłka sygnału poprzez sieć rozległą WAN.

Istnieje wiele innych możliwości zastosowania monitoringu – szczególnie tam, gdzie jest niezbędna kontrola w czasie rzeczywistym w celu podnoszenia bezpieczeństwa. W związku z tym, że jest to projekt mobilnego monitoringu, poszerza to skalę jego zastosowania, gdyż jedynymi wymogami użycia platformy Raspberry Pi2 w tym wypadku jest zapewnienie źródła zasilania oraz dostępu do sieci rozległej.

Mobilny monitoring może być zastosowany nie tylko w komunikacji miejskiej, ale również w systemach alarmowych jak również w systemach zdalnego nadzoru bezpieczeństwa. System mógłby współpracować z dowolną kamerą po uprzedniej implementacji dodatkowych płytek rozwojowych ogólnie dostępnych na rynku lub specjalną kamerą np. termowizyjną lub na podczerwień, co powiększa jego zakres zastosowań. Po dokonaniu modyfikacji oprogramowania system monitoringu może współpracować z innymi systemami zabezpieczeń, gdzie wymagane jest powiadamianie, a także może zapisywać zarejestrowane sekwencje wideo na zdalnych dyskach sieciowych. Może zatem być częścią elektronicznego systemu nadzoru bezpieczeństwa w szerokim znaczeniu tego pojęcia.

Bibliografia:

1. Bartosz A., Transmisja internetowa danych multimedialnych w czasie rzeczywistym. WKŁ, 2010.
2. Czejdo B., Baszun M., Remote medical monitoring and advisory using artificial intelligence techniques to support operators of transportation equipment, *Autobusy – Bezpieczeństwo i ekologia* - 2016, nr 12.
3. Minke R., Ślodyczka M., Węgrzyn D., Mobilny monitoring komunikacji miejskiej, Praca Inżynierska, Politechnika Lubelska, Lublin 2016.
4. Pande A., Zambreno J., Embedded Multimedia-Security System. Algorithms and Architectures. Springer, 2013.
5. Przelaskowski A., Techniki Multimedialne. Wydawnictwo PW – Okno, 2011.
6. Wieczorkowska A., Badanie metod kompresji obrazu. Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej, 1996, nr 10.
7. <http://www.bigasoft.com/articles/avi-to-mjpeg.html>, dostęp maj 2018.
8. <https://www.debian.org/releases/wheezy/>, dostęp maj 2018.
9. <https://polgard.com/pl/aktualności/5-monitoring-wizyjny-w-autobusach-i-tramwajach.html>, dostęp maj 2018.
10. <https://sourceforge.net/projects/mjpg-streamer/>, dostęp maj 2018

Improvement of safety in public transport using image server

The article discusses an aspect of improving safety in public transport in the situation of synchronising standard monitoring, found in city buses, with a centre notifying about dangerous situations - the control room, police or other services. A driver can use a special button to launch the notification and start online image or image and sound transmission to the services responsible for safety. It is expected to increase the general level of public transport safety.

The designed mobile monitoring can be implemented in the future as a system supporting the work of the public transport controller. A practical aspect of the designed system is the use of the Raspberry Pi2 device as a video server, which sends a video stream to the client's device from the video cameras installed inside a vehicle. As it was indicated, it is possible to transmit compressed data and send a signal through a wide area network (WAN).

Keywords: monitoring, security, data transmission.

Autor:

dr inż. Dariusz Gutek – Politechnika Lubelska, Instytut Informatyki, ul. Nadbystrzycka 36b, 20-618 Lublin, d.gutek@pollub.pl