1. Piotr POWROŹNIK[1], 2. Miłosz ĆWIETNIEWICZ[1]

Uniwersytet Zielonogórski, Instytut Metrologii, Elektroniki i Informatyki (1)
ORCID: 1. 0000-0001-7485-9959; 2. 0000-0002-7239-6257

doi:10.15199/48.2022.11.22

Wpływ zastosowania technologii VPN na czas przesyłania danych pomiędzy urządzeniami zgodnymi z koncepcją Internetu Rzeczy

Streszczenie. *W artykule zaprezentowano wyniki badań i analiz dotyczących protokołu OpenVPN dla zaprojektowanej sieci VPN. W projekcie uwzględniono dopasowanie węzłów pomiarowo – sterujących do koncepcji Internetu Rzeczy. W założeniach projektowych zawarto model komunikacyjny oparty na zasadzie publikacji i subskrypcji danych w poszczególnych tematach. Jakość poziomu usług zapewniono na podstawie właściwości protokołu MQTT i komunikacji poszczególnych węzłów z MQTT brokerem.*

Abstract*. In this paper the results of research and analysis on the OpenVPN protocol for the designed VPN network are presented. The project takes into account that the measurement and control nodes fit into the concept of the Internet of Things. The design assumptions include a communication model based on the principle of publication and subscription of data in individual topics. Service level quality was ensured based on the characteristics of the MQTT protocol and the communication of individual nodes with the MQTT broker. (**The impact of the use of VPN technology on the time of data transfer between devices compliant with the Internet of Things concept**).*

Słowa kluczowe: Internet Rzeczy, Wirtualne Sieci Prywatne, MQTT broker, rozproszony sieciowy system pomiarowo – sterujący.
Keywords: Internet of Things, Virtual Private Network, MQTT broker, distributed network measurement and control system.

Wstęp

Rozwój technologiczny oraz pandemia koronawirusa SARS-CoV-2 wymusiła rozwój usług dostępu zdalnego do danych [1]. Wszelkie podejmowane działania związane z rozszerzaniem funkcjonalności usług sieciowych miały na celu łagodzenie skutków społeczno – ekonomicznych pandemii. W wyniku zwiększenia zapotrzebowania na dostęp zdalny wymagane stało się rozważenie w jaki sposób możliwe byłoby zwiększenie poziomu zabezpieczeń dostępu do danych, przy jednoczesnym zachowaniu akceptowalnego poziomu uproszczenia wymiany danych pomiędzy użytkownikami i urządzeniami.

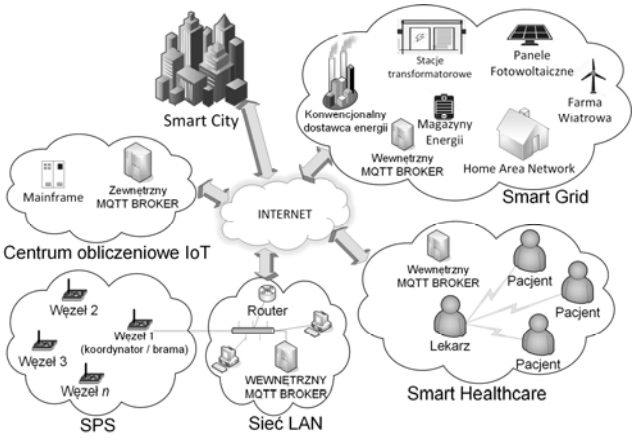
Zróżnicowanie urządzeń sieciowych, pod względem sprzętowym i programowym, daje możliwość korzystania z wymiany danych nie tylko w zastosowaniach komercyjnych, ale i również domowych. Sprzęt wraz z oprogramowaniem tworzą w ten sposób rozproszony sieciowy system pomiarowo – sterujący (RSSPS).

Programowa konfiguracja prostych urządzeń sieciowych realizowana jest za pomocą kreatorów lub wcześniej predefiniowanych szablonów. Takie podejście w większości przypadków upraszcza i przyspiesza proces konfiguracyjny.

Opracowaniem automatyzacji procesu konfiguracji najczęściej zajmują się dostawcy usług internetowych (ang. Internet Service Provider - ISP). ISP dokładają wszelkich starań, by dbać o bezpieczeństwo swoich użytkowników w RSSPS. Niestety, najczęściej to sami użytkownicy, w wyniku braku specjalistycznej wiedzy, przyczyniają się do generowania zagrożeń. Sytuacja taka występuje w przypadku małych firm jak i dużych korporacji.

W XXI wieku kluczową gałąź rozwoju technologicznego stanowi Internet Rzeczy (ang. Internet of Things – IoT) [2], który niesie za sobą konieczność przekazywania danych w RSSPS. W tym przypadku źródłem danych są nie tylko sami użytkownicy, ale i również dedykowane proste urządzenia, np. systemy wbudowane z dostępem do Internetu, takie jak procesory, czy czujniki zbierające dane i elementy wykonawcze. Połączenie poszczególnych elementów RSSPS realizowane jest za pomocą IoT hubów oraz bram IoT.

Na rysunku 1 przedstawiono przykładowy schemat wymiany danych w RSSPS dla kilku wybranych aspektów. W skład tego systemu wchodzi lokalny system pomiarowo – sterujący (SPS) oraz funkcje inteligentnych rozwiązań. W szczególności chodzi o funkcje dotyczące inteligentnych miast (ang. Smart City [3]), ochrony zdrowia (ang. Smart Healthcare [4]) i inteligentnych sieci elektroenergetycznych (ang. Smart Grid [5]). Wymiana danych w poszczególnych częściach składowych RSSPS oraz pomiędzy nimi stanowi istotną funkcjonalność. Brak wymiany danych, przykładowo w Smart Gridach, zaburzyłby prawidłowe funkcjonowanie algorytmów zarządzania energią. W algorytmie elastycznego zarządzania energią (EEM) [6, 7] brak możliwości odczytu danych z inteligentnych odbiorników energii, odnawialnych źródeł energii oraz operatora systemu przesyłowego (OSP) oraz operatora systemu dystrybucyjnego (OSD) uniemożliwiłby między innymi reakcję na zjawisko zwiększonego zapotrzebowania na energię elektryczną (ang. peak demand).



Rys.1. Schemat wymiany danych w RSSPS

Na rysunku 1 centralny punkt RSSPS stanowi infrastruktura związana z siecią Internet. Przez tą infrastrukturę będą przekazywane wszelkiego rodzaju dane pomiarowo – sterujące. Symbolicznie oznaczone zostało również centrum obliczeniowe IoT, w którym dane pomiarowo – sterujące będą przechowywane i przetwarzane. W tym przypadku rolę tę będzie pełnić MQTT broker i komputer o ultra wysokiej wydajności

Mainframe. MQTT broker [8] pełniąc funkcję serwera odpowiada za wysyłanie danych w postaci wiadomości pomiędzy nadawcami, a odbiorcami. Podczas przesyłania wiadomości nadawca nie musi znać liczby oraz lokalizacji odbiorców. MQTT broker po odebraniu danych od nadawcy zajmuje się dystrybucją wiadomości do wszystkich odbiorców, którzy zasubskrybowali dany temat z wiadomością. Tematy stanowią zatem formę adresowania, która umożliwia klientom MQTT udostępnianie informacji. Tematy mają strukturę hierarchiczną. Taki sposób opisywania danych daje możliwość tworzenia przyjaznych dla użytkownika i samoopisujących się struktur nazw. Przy tak zdefiniowanej infrastrukturze analiza danych wykonywana może być przez systemy back-endu lokalnie lub w rozwiązaniach chmurowych, np. na potrzebę rozwiązań biznesowych. Dla RSSPS zdefiniowanego na rysunku 1 przesyłane wiadomości dotyczyć będą wrażliwych danych, które podczas wymiany pomiędzy zewnętrznym (centralnym) MQTT brokerem, a wewnętrznymi brokerami nie mogą zostać zmodyfikowane lub odczytane w celu ich późniejszych analiz.

W literaturze [9] dokonany został podział w zależności od wykonywanej analizy funkcji. W tym przypadku wyróżniono: sieci sensorowe, systemy alarmowe, systemy analizy, systemy sterowania oraz systemy reaktywne, które wskutek analizy danych z czujników wyzwalają określone elementy wykonawcze. Wymiana danych pomiędzy urządzeniami, przy jednoczesnej minimalizacji zużycia energii, stanowi środek do realizacji jednego z celów IoT. W raporcie firmy Ericsson [10] oszacowano, że po 2022 roku liczba urządzeń IoT na świecie będzie wynosiła 29 bilionów, co niewątpliwie ma związek z prężnym rozwojem tego środowiska i rosnącą popularnością.

Każdy użytkownik jest zagrożony przejęciem, modyfikacją lub utratą danych. Nieautoryzowany dostęp do danych może się przyczynić do wymiernych konsekwencji np. finansowych. Szereg zastosowań, gdzie wymieniane są poufne dane, otwiera możliwość wystąpienia nadużyć. W szczególności na uwadze należy mieć wszelkie obszary, gdzie zdrowie i życie ludzkie mogłoby zostać narażone. Nie bez znaczenia będą również kwestie stosowania zabezpieczeń w rozwiązaniach biznesowych.

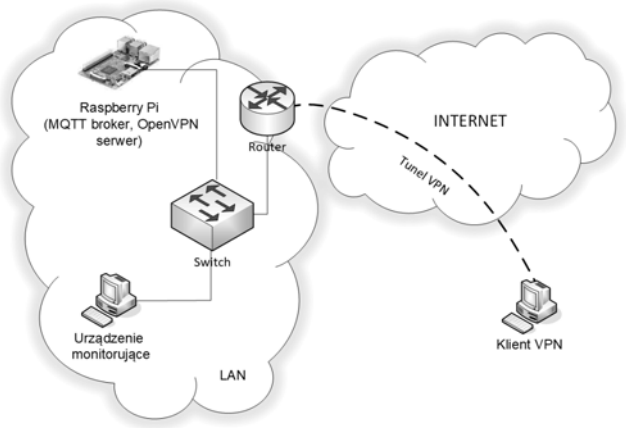
W tej sytuacji zwiększenie poziomu bezpieczeństwa przesyłanych danych w RSSPS może zostać zrealizowane za pomocą wirtualnych sieci prywatnych (ang. Virtual Private Network – VPN) [11]. Przedstawione wyniki stanowią rozszerzenie badań, które zaprezentowano w [12].

Wirtualne sieci prywatne

Sieci VPN umożliwiają użycie publicznej infrastruktury sieciowej wzbogaconej o protokoły tunelowania. Przykładowy schemat uproszczonej wersji RSSPS przedstawiono na rysunku 2.

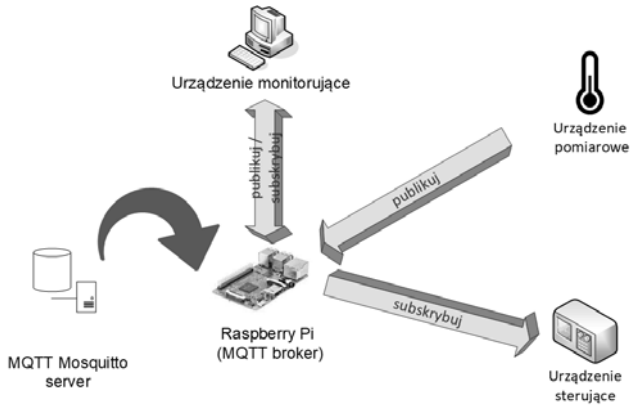
Klient VPN, poprzez tunel VPN, uzyskuje dostęp ze zwiększonym poziomem bezpieczeństwa danych do zasobów wydzielonej sieci LAN. W sieci takiej mogą być urządzenia o dedykowanych funkcjach (w szczególności funkcji serwera). Określenie liczby urządzeń i ich zasobów zależeć będzie między innymi od liczby klientów, którzy będą wysyłać żądania obsługi w tym samym czasie. W przypadku prostych rozwiązań, gdzie nie jest wymagana obsługa wielu jednoczesnych żądań obsługi, wystarczający może okazać się dedykowany system wbudowany taki jak Raspberry Pi. Na rysunku 2 Raspberry Pi pełni rolę MQTT brokera oraz serwera VPN.

Zasada działania MQTT brokera opiera się w głównej mierze na odczytywaniu i zapisywaniu przesłanych wiadomości z danymi pomiarowymi z urządzeń pomiarowych.



Rys.2. Schemat przykładowej uproszczonej wersji RSSPS

Na rysunku 3 przedstawiono ideę użycia MQTT brokera z funkcjami publikacji i subskrypcji danych w wiadomościach.



Rys.3. Idea publikacji/subskrypcji w SPS.

Rolę MQTT brokera pełni Raspberry Pi. MQTT broker może być oparty na serwerze MQTT Mosquitto [13]. MQTT Mosquitto implementuje protokół MQTT w wersjach: 5.0, 3.1.1 i 3.1. Protokół ten może być stosowany do przesyłania danych w wiadomościach na urządzeniach o niskim poborze mocy (np. urządzenia mobilne) i urządzeniach typu Mainframe.

Funkcjonalność przesyłania danych w tematach odbywa się za pomocą funkcji „publikuj". Subskrypcja tematów z danymi pomiarowymi daje możliwość urządzeniom sterującym otrzymanie powiadomienia o zaistniałych zmianach w danym temacie. Takie podejście jest korzystne ze względu na przekazywanie danych tylko wtedy, kiedy jest to konieczne, czyli w momencie zmiany wartości. Niezawodność dostarczania wiadomości została rozwiązana poprzez określenie poziomu jakości usług (ang. Quality of Service - QoS) [14]. Określono trzy poziomy QoS. Poziom QoS = 0 nie daje gwarancji dostawy, ze względu na brak potwierdzeń. Jednak brak potwierdzeń skutkuje szybszym działaniem. Dla poziomu QoS = 1 zapewnione jest, że przynajmniej raz wiadomość zostanie dostarczona do odbiorcy. W tym wariancie nadawca przechowuje wiadomość do momentu potwierdzenia przez odbiorcę. Wiadomość może być wielokrotnie dostarczona. QoS = 2 zapewnia, że dana wiadomość zostanie odebrana tylko raz przez odbiorców, którzy zasubskrybowali dany temat. Takie podejście jest najbezpieczniejsze. Tym razem zwiększenie poziomu bezpieczeństwa skutkuje wolniejszym działaniem

ze względu na zastosowane przynajmniej dwukrotnego przepływu żądania/odpowiedź.

Sieci VPN stanowią skuteczną ochronę dzięki zaawansowanym algorytmom zamieniającym tekst jawny na tekst zaszyfrowany, a co najważniejsze, pozwalają na oddzielenie sieci wewnętrznej od sieci publicznej. Dzięki tej koncepcji urządzenia w RSSPS mogą być ukryte przed niepowołanym dostępem z zewnątrz. Przekazywanie danych wewnątrz takiej sieci realizowane jest w sposób zaszyfrowany. Sieci VPN zapewniają poufność, autentyczność i integralność danych, stając się tym samym pierwszą linią obrony przed potencjalnymi atakami skierowanymi przeciwko RSSPS.

Implementację sieci VPN można zrealizować za pomocą zestawienia połączenia serwer – klient w oparciu o technologię OpenVPN [15]. Technologia ta używa protokołu o tej samej nazwie lub protokołu Wireguard, który jest nowym podejściem do usług VPN. Pomimo faktu, że Wireguard w teorii oferuje większą prędkość transmisji, to posiada wadę w postaci braku możliwości elastycznego wyboru algorytmów kryptograficznych. Niewątpliwą zaletą OpenVPN jest to, że przeszedł już wiele niezależnych audytów potwierdzających jego bezpieczeństwo [15].

W dalszej części artykułu opisano wyniki badań i analiz dotyczących technologii OpenVPN z poziomem szyfrowania określonym przez AES-256. Postawiony cel badań i analiz można podsumować jako ustalenie wpływu stosowania technologii VPN na czas przesyłania danych w RSSPS, gdzie zastosowano MQTT brokera.

W celu podniesienia poziomu bezpieczeństwa zostanie użyta dodatkowo aplikacja Pi-hole [16]. Rozwiązanie to zostało wybrane ze względu na funkcję blokowania żądań DNS dla sieci prywatnej. Pi-Hole ze swoim interfejsem pozwala na filtrowanie ruchu sieciowego serwera VPN, dzięki czemu stanowi zaporę sieciową dla RSSPS. W głównej mierze blokowane są zdefiniowane na liście domeny śledzące i reklamowe.

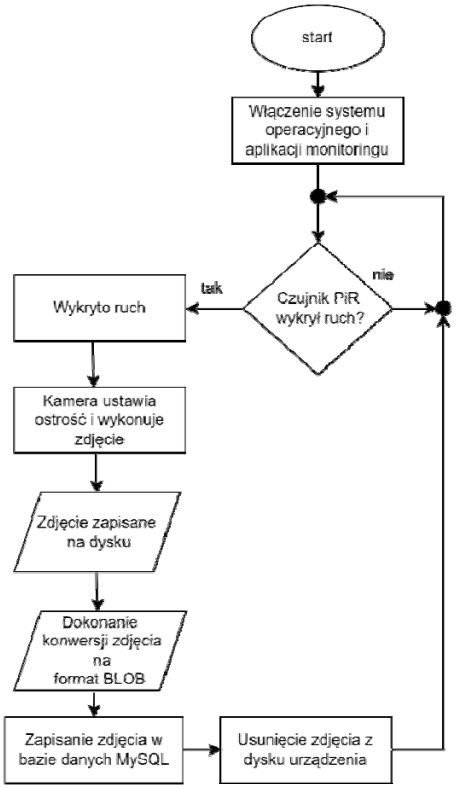
Projekt RSSPS

Na potrzebę przeprowadzenia badań opracowany został testowy RSSPS. System ten miał na celu zapewnienia funkcjonalności domowego systemu monitoringu. Rozwiązanie sprzętowe oparto na:

- systemie wbudowanym Raspberry Pi 3B+,
- czujniku ruchu PIR HC-SR501,
- kamerze ArduCam OV5647.

Należy podkreślić, że zgodnie z przyjętymi założeniami RSSPS miał być niedrogi i bazować na prostych komponentach, które w razie awarii można byłoby szybko i łatwo wymienić na nowe. Dodatkowo w tym przypadku postawiono na rozwiązanie open-source'owe systemu operacyjnego w postaci Raspberry Pi OS. Oprogramowanie dla poszczególnych funkcji RSSPS zrealizowano w oparciu o skrypty w języku Python. Algorytm prezentujący działanie testowego RSSPS przedstawiono na rysunku 4.

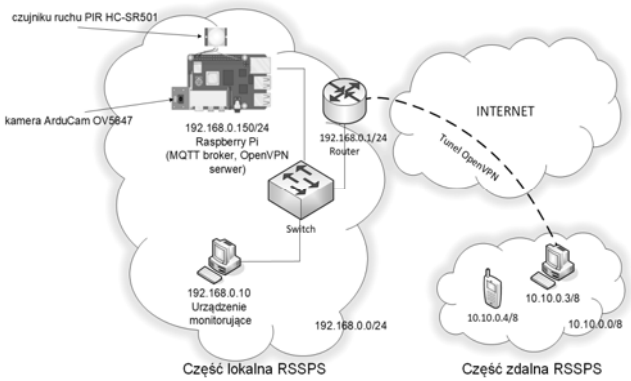
Testowy RSSPS został zaprojektowany w trybie pracy ciągłej. Po podłączeniu Raspberry Pi zaimplementowana aplikacja automatycznie włącza się i zaczyna monitorowanie ruchu rejestrowane przez czujnik PiR. Jeśli ruch zostanie wykryty, to w pierwszej kolejności ustawiana jest ostrość na wykrytym obiekcie, a następnie wykonane zdjęcie zapisane jest na dysku. W celu zapewnienia dostępu do zrobionego zdjęcia użyto pakietu LAMP [17]. Przechodząc do dalszych operacji realizowanych w algorytmie, zdjęcie jest poddawane konwersji do formatu BLOB (ang. Binary Large Object) umożliwiającego przechowywanie danych jako pojedynczego obiektu w bazie danych. Na tym etapie algorytm kończy proces przetwarzania po detekcji ruchu i przechodzi w stan dalszego monitorowania. Zdjęcia z monitoringu dostępne są poprzez podanie właściwego adresu IP serwera Raspberry Pi 3B+ w dedykowanej aplikacji. Aplikacja ta, oprócz samej wizualizacji zdjęć, umożliwia również sprawdzenie dodatkowych parametrów. Wyświetlana jest data oraz godzina wykonanego zdjęcia.



Rys.4. Algorytm prezentujący działanie testowego RSSPS

Analizy zaprojektowanego RSSPS

Dostęp z sieci Internet do testowego RSSPS został zapewniony poprzez konfigurację usługi serwera VPN bazującego na technologii OpenVPN uruchomionego na Raspberry Pi 3B+ z protokołem OpenVPN. Konfiguracja została zrealizowana w wariancie podłączenia klient – klient (tryb mostu ang. bridge mode) [18] lub klient – router (tryb routingu) [18]. Schemat konfiguracji w trybie routingu przedstawiono na rysunku 5.

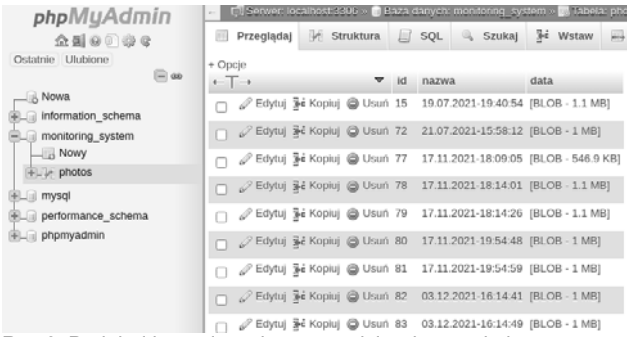


Rys.5. Schemat sieci dla zaprojektowanego RSSPS – tryb routingu

Różnica pomiędzy trybami konfiguracji VPN polega między innymi na modyfikacji sposobu adresacji urządzeń w części zdalnej RSSPS. W przypadku trybu bridge urządzenia w części zdalnej zostałyby zaadresowane w sieci 192.168.0.0/24. Dostęp do sieci Internet został

zrealizowany za pomocą asymetrycznej cyfrowej linii abonenckiej jednego z operatorów dostawców usług internetowych ISP.

W celu weryfikacji poprawności wdrożonych metod zabezpieczeń w zaprojektowanym RSSPS przeprowadzone zostały testy. Celem pierwszego testu było sprawdzenie konfiguracji platformy Docker [19] przy użyciu mechanizmu WSL, który miał na celu weryfikację wstępnej konfiguracji na urządzeniu monitorującym. Mechanizm ten pozwala na użycie jądra Linux'a w systemie Windows, przez co proces konfiguracji aplikacji staje się znacznie szybszy. Dodatkowo zastosowanie platformy Docker dało możliwość użycia wcześniej opisanego rozwiązania Pi-hole w kontenerze w platformie Docker. W drugim teście została przetestowana aplikacja Pihole zainstalowana na Raspberry Pi. Kolejny test dotyczył przekierowania pakietów przychodzących na poszczególne porty. Przetestowane zostały również poszczególne funkcje testowego RSSPS. Sprawdzono między innymi czy obiekty monitorowane i pokazane przez system znajdują się w bazie danych. Zastosowano do tego celu narzędzie *phpmyadmin* (rys. 6).



Rys.6. Podgląd bazy danych w narzędziu *phpmyadmin*

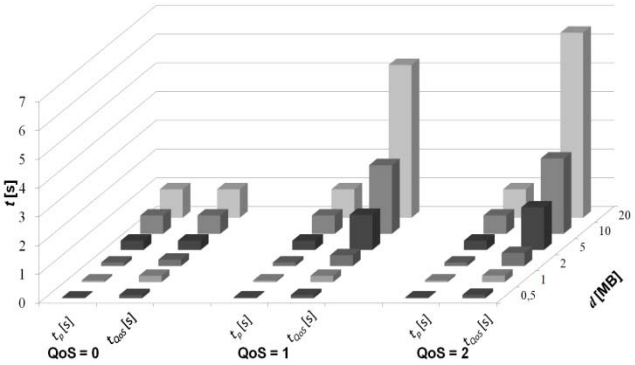
Po weryfikacji poprawności zaprojektowanego RSSPS przeprowadzono analizy, które miały na celu określenie czasów publikacji wiadomości (t_p) oraz czasów potwierdzenia otrzymania danych w wiadomości (t_{QoS}) od odbiorcy. Pomiar czasów t_p i t_{QoS} był realizowany za pomocą narzędzia MQTTBox [20] w wersji 0.2.3. Poszczególne analizy wykonano dla kilku wariantów scenariuszy pomiarowych (sc). Zestawienie sc wraz z opisem poszczególnych wariantów przedstawiono w tabeli 1.

Tabela 1. Zestawienie wybranych sc wraz z opisem

sc	Opis szczegółowy danego sc
sc_1	Określenie wartości czasów t_p i t_{QoS}, gdzie podłączony do MQTT brokera jest jeden klient bez zabezpieczeń w postaci sieci VPN
sc_2	Podobnie jak w sc_1, ale zastosowany jest VPN w trybie rutowalnym
sc_3	Podobnie jak w sc_1, ale zastosowany jest VPN w trybie mostu – bridge
sc_4	Podobnie jak w sc_2, ale tym razem połączonych do MQTT brokera jest dwóch klientów
sc_5	Podobnie jak w sc_3, ale tym razem połączonych do MQTT brokera jest dwóch klientów

Wszystkie dalsze analizy zostały wykonane dla sześciu wariantów wypełnienia pola danych w wiadomości (ang. payload). Pole to było wypełniane danymi (d) o rozmiarze: 0,5, 1, 2, 5, 10 i 20 MB. Warianty d zostały dobrane tak, aby odzwierciedlić najczęściej występujące w praktyce sytuacje. Od przesyłania periodycznie małych porcji danych (np. dane pomiarowe w formacie JSON) do przesyłania aperiodycznie dużych zbiorów danych np. serializowanych danych pochodzących z kamer internetowych. Dodatkowo analizy poszerzono o uwzględnienie trzech poziomów jakości usług QoS.

Na rysunku 7 przedstawiono wyniki pomiaru czasów t_p i t_{QoS} dla sc_1.



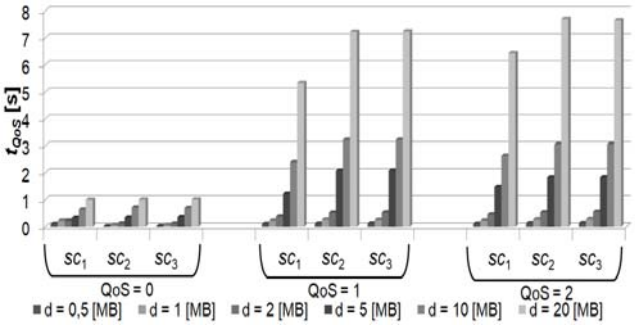
Rys.7. Analiza porównawcza t_p i t_{QoS} dla różnych wartości QoS

Na podstawie analizy wyników przedstawionych na rysunku 7 wartości t_p są porównywalne dla trzech wariantów QoS. Publikacja danych za każdym razem zachodziła w taki sam sposób, co świadczy o powtarzalności badań. Dodatkowo zauważalne są dla QoS = 0 zbliżone wartości czasów t_p i t_{QoS}. Sytuację taką można uzasadnić założeniami QoS. W przypadku, gdy QoS = 0, to nie ma potrzeby oczekiwania na potwierdzenie dostarczenia danych. Zwiększenie poziomu niezawodności dostarczenia danych (QoS = 2 względem QoS równego 0 i 1) powoduje zwiększenie wartości t_{QoS}. Prawidłowość ta zachodzi kolejno dla wszystkich d.

W trakcie prowadzenia kolejnych badań zauważano, że wartości t_p cechują się taką samą zależnością jak to miało miejsce w sc_1.

Zatem, w dalszych analizach będą prezentowane tylko wartości czasów t_{QoS}, na podstawie których można wyciągnąć będzie bardziej szczegółowe wnioski dotyczące wpływu zastosowania technologii VPN na czas przesyłania danych w RSSPS, gdzie zastosowano MQTT brokera.

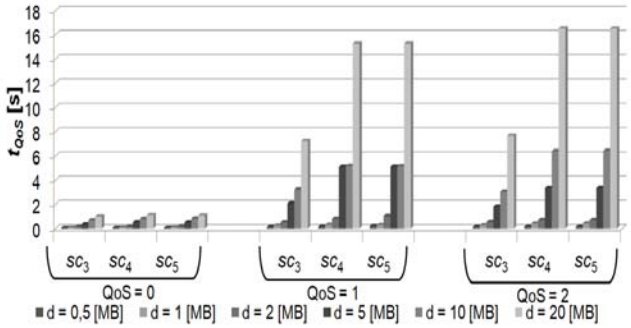
Kolejne analizy przeprowadzono porównując sc_1, sc_2 i sc_3. Wyniki z przeprowadzonych badań zaprezentowano na rysunku 8.



Rys.8. Czasy t_{QoS} podczas przesyłania danych z i bez VPN dla jednego klienta MQTT brokera

Na podstawie wartości przedstawionych na rysunku 8 można stwierdzić, że zastosowanie VPN w trybie routingu i mostu nie wpływa na samo publikowanie danych (dla QoS = 0). Dopiero wprowadzenie mechanizmów potwierdzeń w QoS równego 1 i 2 zwiększa czas t_{QoS}. Nastąpił średnio wzrost wartości t_{QoS} o 20%. Porównując ze sobą sc_2 i sc_3 można natomiast stwierdzić, że w tym przypadku tryb konfiguracji VPN nie miał wpływu na t_{QoS}.

W kolejnych badaniach postanowiono sprawdzić jaki wpływ na t_{QoS} ma zwiększenie liczby klientów VPN korzystających z MQTT brokera. W celach porównawczych na rysunku 9 zestawiono ze sobą wartości czasów t_{QoS} dla sc_3, sc_4 i sc_5.



Rys.9. Czasy t_{QoS} podczas przesyłania danych z i bez VPN dla dwóch klientów MQTT brokera

Na podstawie analizy wartości czasów przedstawionych na rysunku 9 można ponownie potwierdzić brak wpływu trybu konfiguracji VPN na czas $t_{QoS.}$ Dodanie kolejnego klienta VPN spowodowało zwiększenie wartości czasu t_{QoS} średnio o 38 %. Czasy t_{QoS} są tym większe im większa jest wartość d.

Podsumowanie

Obecnie korzystanie z możliwości wymiany danych pomiędzy ludźmi i urządzeniami w RSSPS stało się codziennością. Natomiast zapewnienie poufności, integralności, niezawodności, bezpieczeństwa i prywatności danych musi zostać uwzględnione podczas projektowania nowoczesnych RSSPS. Poszczególne kwestie mają znaczenie nie tylko od strony ekonomicznej, ale i również mogą mieć wpływ na zdrowie i życie ludzkie. Jedną z metod zwiększenia poziomu zabezpieczeń danych jest stosowanie sieci VPN w rozwiązaniach biznesowych jak i domowych. Na podstawie przedstawionych informacji w artykule implementacja sieci VPN może zostać zrealizowana przy użyciu nawet prostych systemów wbudowanych, które umożliwią zwiększenie poziomu bezpieczeństwa.

Na podstawie przeprowadzonych badań i analiz, dla zaprojektowanego RSSPS, stwierdzono że wdrożone rozwiązania przyczyniły się do podniesienia poziomu zabezpieczeń podczas wymiany danych. Na czas t_{QoS} ma wpływ stosowanie technologii VPN, w szczególności dla znacznych wartości parametru d. W szczególnych przypadkach zmian może okazać się konieczne przeniesienie funkcjonalności serwera VPN na osobne urządzenie o większych zasobach sprzętowych. W prostych rozwiązaniach Rasperry Pi może być rozwiązaniem akceptowalnym w przypadku, gdy publikacja danych w tematach będzie realizowana aperiodycznie i z małą częstotliwością.

Na czas t_{QoS} może mieć również wpływ przepustowość zastosowanego łącza od ISP. Dlatego podczas projektowania RSSPS należy uwzględnić przepustowość łącza, do którego podłączony jest serwer VPN i MQTT broker. Sama prędkość wysyłania danych w tematach podczas publikacji i subskrypcji może ulec zmniejszeniu ze względu na fakt, że całość komunikacji musi odbywać się przez serwer, który obsługuje równocześnie wysyłanie jak i odbieranie danych.

Kolejnym etapem prac będzie zwiększenie liczby klientów podłączonych do MQTT brokera i ustalenie granicy kiedy wymagane byłoby zwiększenie zasobów sprzętów serwera tak, aby zminimalizować czasy t_{QoS}. Rozważane jest również zweryfikowanie użytej technologii VPN. W tym przypadku zgromadzone wyniki dla protokołu OpenVPN zostałyby porównane z protokołem WireGuard.

***Autorzy**: dr inż. Piotr Powroźnik, Uniwersytet Zielonogórski, Instytut Metrologii, Elektroniki i Informatyki, ul. prof. Z. Szafrana 2, 65-516 Zielona Góra, E-mail: p.powroznik@imei.uz.zgora.pl; inż. Miłosz Ćwiertniewicz, Uniwersytet Zielonogórski, Instytut Metrologii, Elektroniki i Informatyki, ul. prof. Z. Szafrana 2, 65-516 Zielona Góra, E-mail: cwiertniewicz.milosz@gmail.com.*

LITERATURA

[1] Boonsong W., Senajit N., Remote Patient Body Temperature Monitoring Based-IEEE802.11a Internet of Things (IoT), *Przegląd Elektrotechniczny*, 98 (2022), nr 6, 95-98
[2] Motlagh H. N., Mohammadrezaei M., Hunt J., Zakeri B., Internet of Things (IoT) and the Energy Sector, *Energies*, 13 (2020)
[3] Burlacu M., Boboc R. G., Butilă E. V., Smart Cities and Transportation: Reviewing the Scientific Character of the Theories. *Sustainability*, 14(13) (2022)
[4] Mohanty M. D., Das A., Mohanty M. N., Altameem A., Nayak S. R., Saudagar A. K. J., Poonia R. C., Design of Smart and Secured Healthcare Service Using Deep Learning with Modified SHA-256 Algorithm, *Healthcare*, 10(7):1275 (2022)
[5] Tamay P., Inga E., Charging Infrastructure for Electric Vehicles Considering Their Integration into the Smart Grid, *Sustainability*, 14(14):8248 (2022)
[6] Powroźnik P., Szcześniak P., Turchan K., Krysik M., Koropiecki I., Piotrowski K., An Elastic Energy Management Algorithm in a Hierarchical Control System with Distributed Control Devices, *Energies*, 15(13):4750 (2022)
[7] Powroźnik P., Szcześniak P., Piotrowski K., Elastic Energy Management Algorithm Using IoT Technology for Devices with Smart Appliance Functionality for Applications in Smart-Grid, *Energies*, 15(1):109 (2022)
[8] D'Ortona C., Tarchi D., Raffaelli C., Open-Source MQTT-Based End-to-End IoT System for Smart City Scenarios, *Future Internet,* 14(2):57 (2022)
[9] Serpanos D., Wolf M.: Internet-Of-Things(IoT) Systems, *Springer*, (2018)
[10] Collela P.: Unshering In a Better Connected Future, https://www.ericsson.com/en/about-us/company-facts/ericsson-worldwide/india/authored-articles/ushering-in-a-better-connected-future (dostęp 12.07.2022)
[11] Trombeta L., Torrisi NM.: DHCP Hierarchical Failover (DHCP-HF) Servers over a VPN Interconnected Campus, *Big Data and Cognitive Computing*. 3(1):18 (2019)
[12] Powroźnik P., Ćwiertniewicz M., Wirtualna sieć prywatna w usługach wymiany danych w sieciowych systemach pomiarowo - sterujących, *W: Systemy pomiarowe w badaniach naukowych i w przemyśle* - SP'2022: XIV Konferencja Naukowa. *Łagów, Polska, 2022 .- Zielona Góra:* Uniwersytet Zielonogórski, Instytut Metrologii, Elektroniki i Informatyki, (2022), 77-80
[13] Eclipse Mosquitto, https://mosquitto.org/ (dostęp 11.07.2022)
[14] Karakus M., Durresi A., Quality of Service (QoS) in Software Defined Networking (SDN): A survey, *Journal of Network and Computer Applications*, 80 (2017), 200-218
[15] Gentile A.F., Fazio P., Miceli G.: A Survey on the Implementation and Management of Secure Virtual Private Networks (VPNs) and Virtual LANs (VLANs) in Static and Mobile Scenarios. Telecom. 2(4) (2021),430-445
[16] Pi-hole® Network-wide Ad Blocking, https://pi-hole.net/ (dostęp 15.07.2022)
[17] Ubuntu, https://ubuntu.com/server/docs/lamp-applications (dostęp 15.07.2022)
[18] OpenVPN, https://openvpn.net/ (dostęp 12.07.2022)
[19] Docker, https://www.docker.com/ (dostęp 12.07.2022)
[20] MQTTBox, https://github.com/workswithweb/MQTTBox (dostęp 12.07.2022)