JOURNAL of COMPUTER SCIENCES INSTITUTE

JCSI 8 (2018) 247-251
Wysłane: 2018-03-17
Przyjęte: 2018-03-28

Porównanie wybranych metod komunikacji sieciowych na platformie Android

Przemysław Żydek*, Jakub Smołka
Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Artykuł poświęcony jest porównaniu trzech metod komunikacji aplikacji mobilnej z serwerem. Analizie poddano wyniki sześciu badań przeprowadzonych przy użyciu protokołów HTTP, HTTPS oraz technologii gniazd serwera. Zbadano czas przesłania danych o różnych rozmiarach pomiędzy aplikacją klienta a serwerem oraz wpływ operacji na użycie procesora i zużycie baterii. Doświadczenia polegały na wysłaniu i odebraniu formularza, wysłaniu i odebraniu dużego zdjęcia oraz zapewnieniu komunikacji ciągłej w celu zbadania użycia urządzenia. Do przeprowadzenia analizy stworzono aplikację klienta na platformę Android obsługującą badane technologie. Stronę serwera stanowiły programy napisane w technologii Java uruchomione na serwerze Tomcat. Przeprowadzone badania umożliwiły wskazanie faworyta i jest nim technologia gniazd serwera.

Słowa kluczowe: http; HTTPS; gniazdo serwera; Android; komunikacja sieciowa

*Autor do korespondencji.

Adres e-mail: przemekkzydek@wp.pl

Comparison of selected network communication methods on the Android platform

Przemysław Żydek*, Jakub Smołka
Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. This paper is devoted to comparing three communication methods between mobile applications and servers. The analysis encompassed the results of six tests conducted using HTTP and HTTPS protocols, and server-socket technology. All sending times of data with various sizes between the client application and the server, and the impact of this operation on the processor load and battery use, were evaluated. The experiments consisted of sending and receiving a form, sending and receiving a large photo, and ensuring continuous communication to assess device use. To perform the analyses, an Android application was created to support the researched technology, while the server side was composed of programs written in Java, running on a Tomcat server. The conducted research made it possible to establish the best solution, which is server-socket technology.

Keywords: HTTP; HTTPS; socket; Andoid; network communication

*Corresponding author.

E-mail address: przemekkzydek@wp.pl

1. Wstęp

Duża liczba możliwości wykorzystania urządzeń mobilnych sprawiła, że znajdują one zastosowanie w prawie każdej dziedzinie. Dostarczane funkcjonalności realizowane są za pomocą aplikacji zainstalowanych na urządzeniu. Bardzo ważnym aspektem w działaniu tych rozwiązań jest rodzaj komunikacji sieciowej stosowany do wymiany danych z serwerem. Wykorzystywany w tych rozwiązaniach model klient-serwer zazwyczaj składa się z jednego centralnego serwera i wielu punktów klienckich [1]. Przed wyborem odpowiedniej technologii należy odpowiedzieć na pytanie, czy operacja przesłania danych ma być jak najbardziej wydajna, a może informacje na których operuje mają być bezpieczne i odporne na ataki.

Komunikacja sieciowa jest tematem wielu prac. w [2] Autorzy artykułu opisują tworzenie aplikacji sieciowych w środowisku Java przy pomocy gniazd serwera. Program uruchomiony po stronie klienta inicjuje połączenie poprzez wysłanie żądania na określony port, serwer interpretuje otrzymane dane i jeśli jest to możliwe zwraca zasób. Serwer centralny może działać w dwóch modelach TCP/IP. Pierwszy z nich to model iteracyjny, jest dość mocno ograniczony ponieważ jego architektura obsługuje tylko jedno żądanie, kolejne nadchodzące żądania są kolejkowane, co oznacza opóźnienia w obsłudze klientów. Drugi model współbieżny umożliwia obsługę wielu żądań jednocześnie. Gdy pojawia się połączenie z drugim klientem następuje rozwidlenie procesów. Zwiększa to znacznie wydajność obsługi większej ilości klientów w tym samym czasie. Kolejne opisane aspekty towarzyszące szyfrowaniu to konieczność nabycia certyfikatu, obciążenie sprzętu czy nakład pracy sieci.

W artykule [3] Autorzy poruszyli aspekt wydajności protokołu HTTP oraz jego szyfrowanej wersji HTTPS. Zwrócono uwagę na to, że każdej technologii zabezpieczającej dane towarzyszy koszt wykonania dodatkowych operacji. Wszystkie koszty są do zaakceptowania jeśli dane którymi operuje aplikacja są poufne. w artykule zaprezentowano ewolucję udziału przepływu komunikacji za pośrednictwem protokołu HTTPS, od kwietnia 2012 r. do września 2014 r. na przykładzie portali Facebook i YouTube zanotowano

dwukrotny wzrost popularności tego rozwiązania. Zjawisko to jest podyktowane ochroną prywatności użytkowników. Popularność technologii ma też wpływ na zmniejszenie kosztów infrastruktury. Do porównania technologii przeprowadzono badania, w których zmierzono czasy ładowania stron internetowych. Zbadano komunikację z wykorzystaniem protokołów HTTP oraz HTTPS. Otrzymane wyniki wykazują na to, że użycie HTTPS znacznie wydłuża ładowanie strony. Wydłużony czas jest spowodowany głównie nawiązywaniem połączenia szyfrowanego za pomocą protokołu TLS. Bardzo ważnym aspektem takiej komunikacji w urządzeniach mobilnych jest wpływ na żywotność baterii. Przeprowadzone badania nie wykazały znacznego wzrostu kosztów energii przy testowaniu protokołu HTTPS.

Artykuł [4] zawiera informacje o procesie badania komunikacji pomiędzy urządzeniem mobilnym, opartym na architekturze ARM a serwerem. Autor prezentuje jakich narzędzi należy użyć do przechwycenia ruchu sieciowego. Programy Tcpdump oraz Burp umożliwiają wygenerowanie pliku z całym ruchem sieciowym, który w dalszej części badań może posłużyć do analizy komunikacji.

Badanie w artykule [5] miało na celu zbadanie aplikacji mobilnych pod kątem zaimplementowanych rozwiązań sieciowych. Badana próbka to 639 283 aplikacji, z czego w 573 258 wykryto połączenia sieciowe. Analiza wykazała, że znaczna większość połączeń realizowana jest za pomocą protokołu HTTP, 414 194 aplikacji korzysta z tej technologii. Szyfrowana odmiana tego protokołu została znaleziona w 229 317 aplikacjach.

2. Przedmiot badań

Przedmiotem badań były trzy popularne metody komunikacji sieciowej. Zbadano protokoły HTTP, HTTPS oraz gniazdo serwera. Pierwszy z nich to natywny protokół znajdujący się w warstwie aplikacji modelu OSI, głównie wykorzystywany przez serwery WWW oraz przeglądarki internetowe. Dostęp do zasobów realizowany jest za pomocą żądań w postaci tekstu, wysyłanych do serwera [6, 7]. Drugi jest odmianą pierwszego z tą różnicą, że przesyłane dane są szyfrowane przez protokół TLS. HTTPS zapewnia w pełni poufność przesyłanych danych [8, 9]. Technologia gniazd opiera się na protokole TCP lub UDP. Transport danych odbywa się za pomocą strumienia, daje to możliwość odczytu i zapisu bajtów danych do wskazanego adresu IP i portu [10,11].

3. Przeprowadzone badania

Do przeprowadzenia analizy metod komunikacji zaimplementowano (1) aplikację mobilną na platformę Android pełniącą rolę klienta oraz (2) stronę serwera obsługującego żądania i zapisującego czasy doświadczeń w bazie danych. Programy obsługują wszystkie trzy opisywane sposoby komunikacji. Badania zostały przeprowadzone w środowisku lokalnym, w którego skład wchodzi komputer pełniący rolę serwera, ruter Wifi oraz urządzenie klienckie typu smartfon. Przyjęty format w jakim przesyłane są dane to JSON, składający się z trzech elementów. Pierwszy z nich zawiera informacje o przesyłanych danych, drugi właściwe dane, trzeci czas operacji wyrażony w ms. Wszystkie operacje poddane analizie wymieniono w kolejnym punkcie.

3.1. Badane operacje

Do analizy metod komunikacji klienta z serwerem przeprowadzono następujące operacje:

1) Wysłanie formularza składającego się z 20 pól opisowych, dane mogą odzwierciedlać zamówienie towaru lub przeprowadzenie badania poza biurem (jest to około 15 tys. znaków). Próbę wykonano 100 razy.
2) Pobranie formularza, aplikacja klienta wysyła żądanie o udostępnienie zasobu. Serwer w odpowiedzi zwraca formularz identyczny jak w poprzednim punkcie. Próbę wykonano 100 razy.
3) Wysłanie z pamięci lokalnej urządzenia zdjęcia o rozdzielczości 6000 x 4000 pikseli. Rozmiar pliku wynosi ok 7 MB o rozszerzeniu JPG. Próbę wykonano 100 razy.
4) Pobranie zdjęcia z serwera. Dane wykorzystane z poprzedniego punktu. Próbę wykonano 100 razy.
5) Pomiar zużycia baterii oraz użycia procesora podczas ciągłej komunikacji w interwale z pół sekundową przerwą. Użyto aplikacji *Monitor Systemu* oraz *CoolTool.*

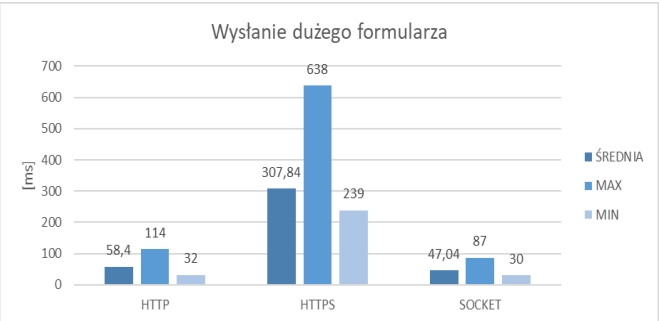
3.2. Platforma testowa

Wszystkie badania przeprowadzono na tych samych urządzeniach. Aplikacja kliencka została stworzona w środowisku programistycznym Android Studio w wersji 2.3.3, z wykorzystaniem narzędzi programistycznych SDK w wersji 26. Urządzenie klienta stanowi smartfon Samsung Galaxy J3 2016 z systemem Android w wersji 5.1.1 Lollipop. Stronę serwera zapewnia komputer typu laptop Acer Swift 3 z systemem operacyjnym Windows 10 Professional.

4. Wyniki przeprowadzonych badań

4.1. Wysłanie formularza

Rysunek 1 przedstawia wykres z czasem średnim, maksymalnym oraz minimalnym dla każdej technologii.

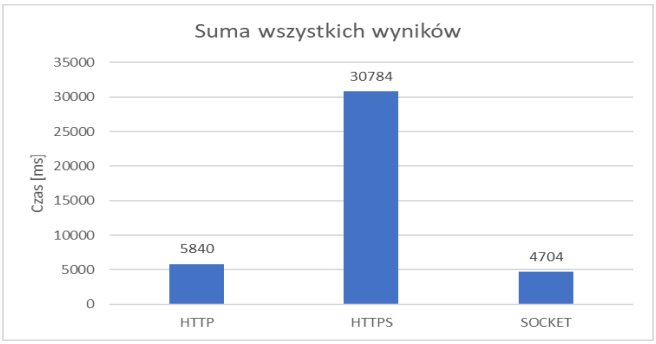


Rys. 1. Wykres z czasami wysłania formularza

Średni czas operacji wysłania formularza do serwera oraz odebrania odpowiedzi dla protokołu HTTP wyniósł 58 ms,

szyfrowane połączenie na wykonanie tej operacji potrzebowało 308 ms, a technologia gniazda osiągnęła wynik 47 ms.

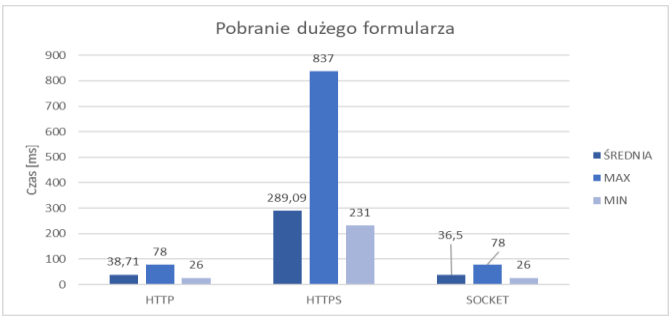
Analiza sumy wszystkich czasów osiągniętych przy 100 wykonanych operacjach, wskazuje bezkonkurencyjną technologię gniazd serwera z wynikiem 4707 ms. Najdłużej tą operację wykonuje protokół HTTPS z sumarycznym czasem 30 784 ms (rys 2).



Rys. 2. Suma wszystkich zgromadzonych wyników

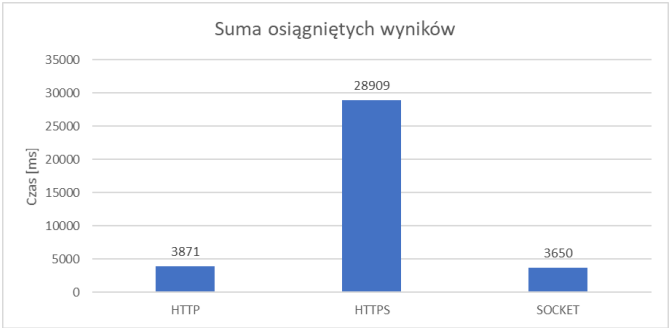
4.2. Pobranie formularza

Średni czas wykonania operacji dla badanych technologii HTTP, HTTPS i gniazda serwera wynoszą odpowiednio 39 ms, 289 ms i 37 ms. Zebrane dane prezentuje rysunek 3.



Rys. 3. Wykres z czasami pobrania formularza

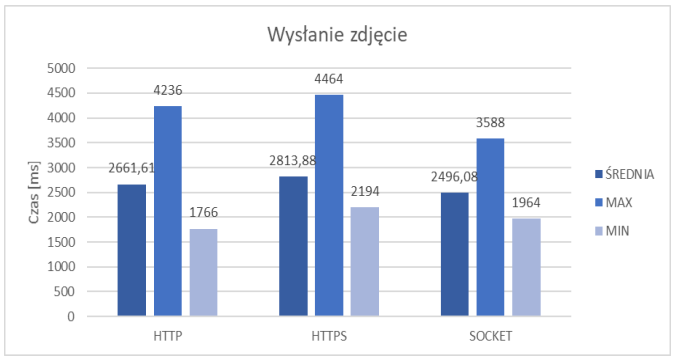
Łączny czas badanych operacji jest krótszy niż w poprzednim punkcie. Technologia gniazd bardzo zbliżyła się wydajnością do protokołu HTTP. Uśredniony czas obydwu technologii wynosi 3760 ms. Najdłuższy łączny czas o wartości 28909 ms zanotowano podczas badania protokołu HTTPS (rys. 4).



Rys. 4. Suma wszystkich zgromadzonych wyników

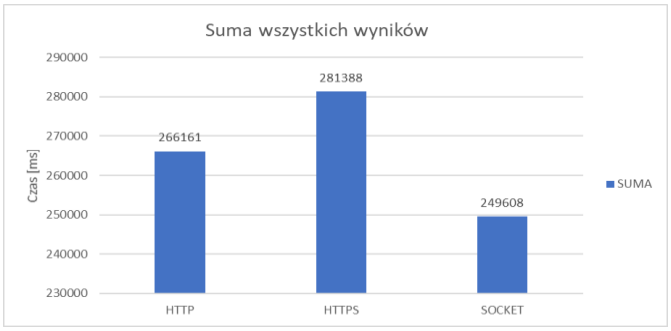
4.3. Wysłanie zdjęcia

Kolejnym badanym aspektem jest wysłanie zdjęcia o rozmiarze 7 MB do serwera. Najgorszy średni wynik uzyskał protokół HTTPS z czasem 4464 ms. Drugi w zestawieniu uplasował się protokół HTTP z rezultatem 2662 ms. Najkrótszy czas uzyskała technologia gniazd, osiągnęła wynik 2496 ms. Analizowane czasy umieszczono na rysunku 5.



Rys. 5. Wykres z czasami wysłania zdjęcia

Najlepszy łączny czas w analizowanym badaniu osiągnęła technologia gniazd z wynikiem 249 s. Zaraz za nią znajduję się rozwiązanie wykorzystujące protokół HTTP, 266 s. Ostatnie miejsce w zestawieniu z czasem 281 s zajęła metoda HTTPS. Wyniki przedstawia rysunek 6.

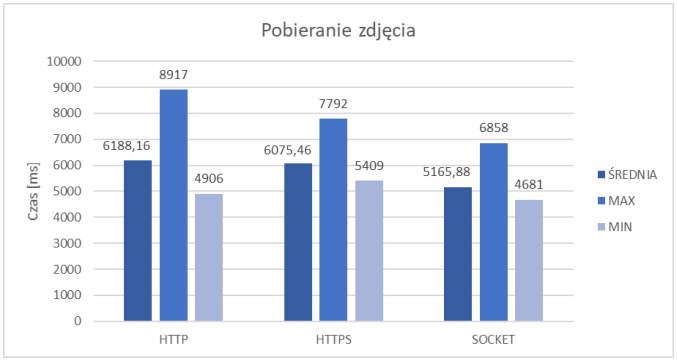


Rys. 6. Suma wszystkich zgromadzonych wyników

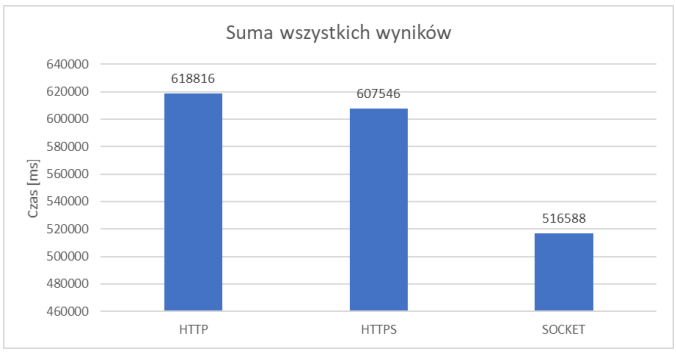
4.4. Pobranie zdjęcia

Analizowany eksperyment opiera się na operacji odebrania zasobu (zdjęcia) od serwera. Faworytem w tym zestawieniu okazała się technologia gniazd, osiągając najkrótszy czas 5165 ms. Drugim czasem wynoszącym 6075 ms może pochwalić się protokół HTTPS. Najgorzej w zestawieniu wypadło rozwiązanie HTTP z czasem 6188 ms. Omawiane wyniki zaprezentowano na rysunku 7.

Najkrótszy czas potrzebny do wysłania 100 dużych zdjęć do serwera to 517 s, osiągnęła go technologia gniazd. Drugi odnotowany czas to 608 s, przy analizie protokołu HTTPS. Najwięcej czasu potrzebował protokół HTTP z czasem 619 s. Przedstawione wyniki padań zaprezentowano na rysunku 8.



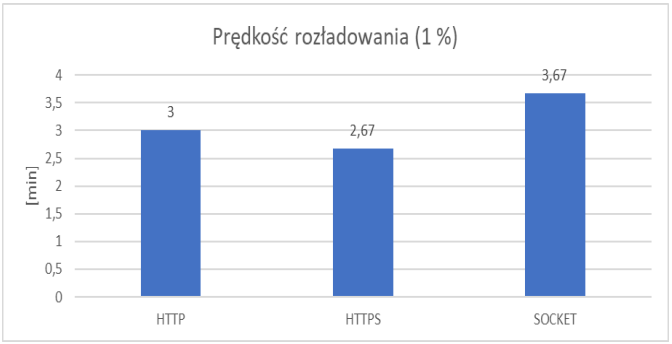
Rys. 7. Wykres z czasami pobrania zdjęcia



Rys. 8. Suma wszystkich zgromadzonych wyników

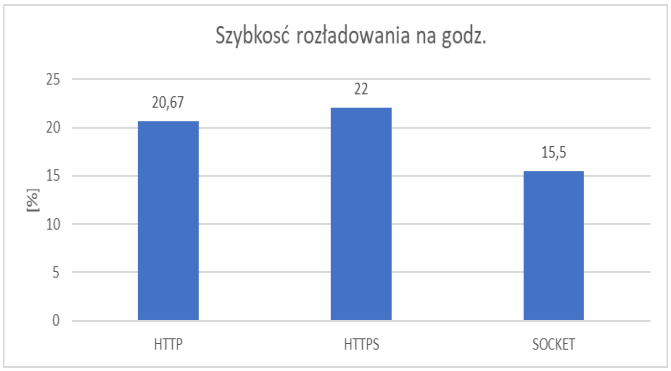
4.5. Pomiar użycia baterii

Pierwszym analizowanym przypadkiem jest czas w jakim bateria rozładuje się o 1%. Najlepszy wynik osiągnęła technologia gniazd, która potrzebowała 3,67 minuty do rozładowania analizowanej próbki. Pośrodku zestawienia znajduje się protokół HTTP z czasem 3 minut. Najszybciej rozładowuje się bateria przy komunikacji z wykorzystaniem protokołu HTTPS, w przeciągu 2,67 minuty rozładuje się 1% baterii na urządzeniu. Opisane wyniki przedstawiono na rysunku 9.



Rys. 9. Prędkość rozładowania 1% baterii

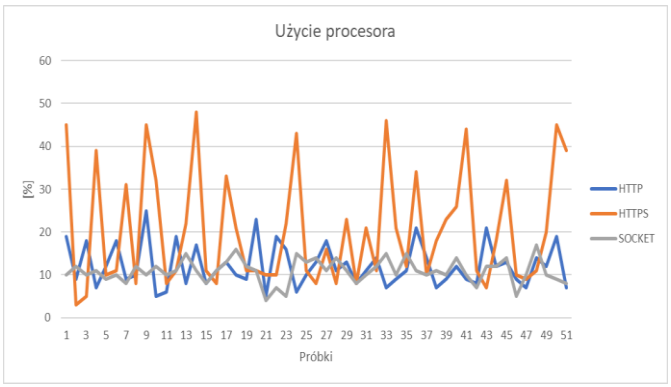
Drugim badanym aspektem jest ilość rozładowania wyrażona w procentach w przeciągu godziny. Najbardziej zostanie rozładowana bateria przy pracy z protokołem HTTPS, jest to około 22%. Zbliżoną wartość 21% osiągnięto przy komunikacji HTTP. Najmniejszą ilość zużycia baterii wynoszącą 15,5% odnotowano przy analizie technologii gniazd. Wyniki pokazano na rysunku 10.



Rys. 10. Szybkość rozładowania baterii

4.6. Pomiar użycia procesora

Zebrane wyniki oscylowały w okolicy 15% dla wszystkich badanych metod komunikacji. Najlepszy średni wynik użycia procesora uzyskała technologia gniazd z wartością 11%. Drugie miejsce zajął protokół HTTP z wynikiem 13%, rezultat protokołu HTTPS wyniósł 20% i stanowi największą wartość. Najbardziej stabilne wartości prezentuje rozwiązanie gniazd, różnica pomiędzy wartością maksymalną a minimalną wynosi 13% jest to o 246 % więcej od porównywanego protokołu HTTPS. Wykres prezentujący wyniki przestawia rysunek 11.



Rys. 11. Wykres użycia procesora

5. Wnioski

Analizując zebrane wyniki należy stwierdzić, że najbardziej wydajną technologią spośród testowanych są gniazda serwera. Osiągnęły we wszystkich eksperymentach najlepsze rezultaty. Różnica względem drugiego rozwiązania w zestawieniu (protokołu HTTP) jest niewielka i jej średnia ze wszystkich badań wynosi 27%. Porównanie technologii gniazda serwera do najwolniejszego rozwiązania wypada znacznie gorzej na rzecz protokołu HTTPS, tutaj różnica wynosi 510%.

Protokół HTTPS okazał się najwolniejszą metodą komunikacji. We wszystkich eksperymentach osiągnął najdłuższe czasy. Jedynym zadaniem w którym metoda była lepsza od protokołu HTTP, było pobranie dużej ilości danych w postaci zakodowanego zdjęcia. Wpływ na słabe wyniki ma ustanawianie przez protokół TLS połączenia szyfrowanego.

Najmniej czasu na rozładowanie 1% baterii potrzebuje aplikacja korzystająca z protokołu HTTPS (tj. 2,67 minut), faworytem w tym zestawieniu są gniazda z czasem 3,67 minuty. Najbardziej rozładuje się bateria w przeciągu godziny podczas pracy z technologią szyfrowaną (22%), drugi w zestawieniu z wartością 20,67% jest protokół HTTP, na pierwszym miejscu uplasowała się metoda gniazd serwera. Wyniki otrzymane podczas badania obciążenia procesora oscylowały w granicach 15%. Najmniej stabilne wyniki osiągnął protokół HTTPS, związane jest to z ustanawianiem połączenia szyfrowanego.

W nawiązaniu do powyższych wniosków można stwierdzić która z technologii jest najwydajniejsza, lecz to nie powinno mieć kluczowego wpływu na wybór metody komunikacji. Jeżeli projektowana aplikacja nie działa w sieci Intranet, a dane którymi operuje są wrażliwe powinno się przede wszystkim zadbać o bezpieczeństwo danych bez względu na koszty z tym związane.

Literatura

[1] http://zstzbaszynek.pl/blog/2021,era-systemow-klient-serwer/ [11.01.2018]

[2] A. Sawant, B. Meshram: Network programing in Java using Socket, Google Scholar, 2013.

[3] D. Naylor, A. Finamore, I. Leontiadis, Y. Grunenberger, M. Mellia, M. Munafo, K, Papagiannaki, P. Steenkiste: The Cost of the „s" in HTTPS, Google Schoolar, 2015.

[4] A. Serafinowicz: Android – historia prawdziwa, http://softonet.pl/publikacje/poradniki/Android-historia.prawdziwa, 1162, 2015.

[5] M. Oltrogge, Y. Acar, s. Dechand, M. Smith, s. Fahl: To Pin or Not to Pin – Helping App Developers Bullet Proof Their TLS Connections, Google Schoolar, 2015.

[6] J.Smołka: Programowanie aplikacji dla systemu Android, Politechnika Lubelska, 2014.

[7] B. Sosinsky: Networking Bible, Wiley Publishing, Inc.2009.

[8] T. Dierks, E. Rescorla: The Transport Layer Security (TLS) Protocol, RFC5246, 2008.

[9] http://edu.pjwstk.edu.pl/wyklady/mpr/scb/W8/W8.htm [20.01.2018]

[10] W. Frank Ableson, R. Sen, C. King, C. Enrique Ortiz: Android in action. Third editon, Manning Publications Co., 2011

[11] https://www.tutorialspoint.com/unix_sockets/what_is_socket.htm [01.01.2018]