UDC 004.415.2

Volodymyr Kazymyr, Andrii Mokrohuz

SELECTING A BASE APPLICATION LAYER PROTOCOL TO TRANSFER DATA IN THE CLIENT-SERVER ARCHITECTURE

Володимир Казимир, Андрій Мокрогуз

ВИБІР БАЗОВОГО ПРОТОКОЛУ ПРИКЛАДНОГО РІВНЯ ДЛЯ ПЕРЕДАЧІ ДАНИХ В АРХІТЕКТУРІ КЛІЄНТ-СЕРВЕР

Владимир Казимир, Андрей Мокрогуз

ВЫБОР БАЗОВОГО ПРОТОКОЛА ПРИКЛАДНОГО УРОВНЯ ДЛЯ ПЕРЕДАЧИ ДАННЫХ В АРХИТЕКТУРЕ КЛИЕНТ-СЕРВЕР

This article presents a system of criteria to determine the optimal application-level protocol for communication between a client and a server when mobile devices are involved in the communication. The criteria focused on the abilities of protocols to support different types of content, structure features, the ability to expand, header size and complexity of usage by software developers. According to the system of the criteria, the most popular application-level protocol HTTP (S) and other alternative protocols like SPDY and COAP have been analyzed and researched.

***Key words**: HTTP, CoAP, SPDY, mobile device, protocol.*

Fig.: 3. Tabl.: 5. Bibl.: 8.

Представлено систему критеріїв для визначення оптимального протоколу прикладного рівня при організації зв'язку між клієнтом і сервером у процесі використання мобільних пристроїв. Запропоновані критерії зосереджені на таких можливостях протоколів, як підтримка різних типів контенту, особливостях структур, здатності до розширення, розміру заголовка і складності використання розробниками програмного забезпечення. За обраною системою критеріїв були проаналізовані та досліджені найпопулярніший протокол прикладного рівня HTTP(S) та інші альтернативні протоколи, такі як COAP і SPDY.

***Ключові слова:** HTTP, COAP, SPDY, мобільний пристрій, протокол.*

Рис.: 3. Табл.: 5. Бібл.: 8.

Представлена система критериев для определения оптимального протокола прикладного уровня при организации связи между клиентом и сервером в процессе использования мобильных устройств. Предложенные критерии сосредоточены на таких возможностях протоколов, как поддержка различных типов контента, особенностях структур, способности к расширению, размеру заголовка и сложности использования разработчиками программного обеспечения. По выбранной системе критериев были проанализированы и исследованы популярный протокол прикладного уровня HTTP (S) и другие альтернативные протоколы, такие как COAP и SPDY.

***Ключевые слова**: HTTP, COAP, SPDY, мобильное устройство, протокол.*

Рис.: 3. Табл.: 5. Библ.: 8.

Urgency of the research. Developers have been developing their software during many years by using different programming languages, different platforms, approaches and technologies. Selection of an application level protocol is one of the issues, which developers had to overcome on their way to successful applications. Design and creation of application level protocol is not an easy task. Many things should be considered during these processes, for instance, protocols should be unambiguous and precise, allow future extensions, do not replicate services provided by lower layer protocols etc.

Application protocols are different. Application-level protocols are designed to target specific application tasks. They determine both the procedure of interaction between a specific type of application processes and the presentation of information in this interaction.

The functions associated with the application layer protocols enable our human network to interface with the underlying data network. When we open a web browser or an instant message window, an application is started, and the program is put into the device's memory where it is executed. Each executing program loaded on a device is referred to as a process. Applications and services are two forms of software programs or processes that provide access to the network within the application layer.

Total number of application layer protocols is large and continues to increase steadily. Some protocols have existed since the very beginning of the development of the Internet. For example, TELNET and FTP are quite old protocols. Others have appeared recently such as X-Window, SNMP or SPDY.

Despite the fact that protocols are designed to be as optimal as possible, developers can misuse each application level protocol. For instance, developers can generate technical data, which is not required, when transferring text data types or select the technology, which transfers unnecessary data. [1]

In the paper, we are considering the most popular application level protocols and identifying which are the best protocols to work with, when developing a client server application.

Target setting. Large variety of application layer protocols makes it hard to analyze them and make some conclusions. Therefore, we need to identify the group of application layer protocols, which will help us to identify the protocols for our research. We decided to create the criteria list to distinguish the protocols. The next criteria were defined to identify the most appropriate application layer protocols. For simplicity, we call this criteria "Identification criteria".

- Protocol must support the client-server communication.
- Protocol must be available for usage in mobile networks.
- Protocol must have development tools, which help to use it.
- Protocol must be popular and used in the Internet.

In addition, we defined the criteria to compare selected protocols. For simplicity, we call this criteria "Comparison criteria". Every application layer protocol, which satisfies identification criteria, will be compared using comparison criteria.

- Amount of a technical data transferred.
- Protected data transfer reliability and performance.
- Popularity of the protocol in terms of software development tools.
- Overall performance.
- Client server communication drawbacks and characteristics.

Actual scientific researches and issues analysis. End users use application layer protocols in form of software applications. Sandvine report [2] shows overall usage of applications and application layer protocols by mobile devices in mobile networks. For instance, Table 1 shows percentage of usage of the most popular application layer protocols for mobile access.

Table 1

Mobile access traffic share for Europe

	Upstream		Downstream		Aggregate	
Rank	Application	Share	Application	Share	Application	Share
1	Facebook	17.93%	HTTP	17.65%	HTTP	16.92%
2	HTTP	13.45%	YouTube	16.54%	YouTube	15.15%
3	SSL	8.63%	Facebook	12.85%	Facebook	13.72%
4	YouTube	8.25%	SSL	5.68%	SSL	6.17%
5	BitTorrent	5.00%	MPEG	4.23%	MPEG	3.85%
6	Skype	4.60%	Netflix	3.89%	Netflix	3.53%
7	iTunes	3.01%	iTunes	3.48%	iTunes	3.40%
8	Instagram	2.07%	Google Market	2.66%	BitTorrent	3.02%
9	MPEG	2.05%	BitTorrent	2.60%	Google Market	2.43%
10	Snapchat	1.86%	Instagram	1.92%	Skype	1.93%
		64.99%		69.59%		70.13%

sandvine

As it can be seen, HTTP protocol takes leading position for mobile devices in the Internet for Europe. For Northern America HTTP takes 3rd place with share of 12.59 %, Latin America has aggregated value 13.04 %, Asia Pacific has 19.14 % of HTTP and Africa has 26.15 % [2].

That means that HTTP is the most popular protocol for users of mobile devices. In some world regions, it has lower value, but the general picture shows domination of HTTP over other application layer protocols in the Internet. Thus, we need to consider different application layer protocols, which can replace HTTP in some cases. However, they must be similar to HTTP.

Uninvestigated parts of general matters defining. It is very hard to select appropriate application layer protocols to analyze, without distinguishing required protocols from a large set. Identification criteria are presented in Table 2 for application layer protocol HTTP.

Table 2

Identification criteria for HTTP

Criteria	Value
Protocol must support the client-server communication	Yes
Protocol must be available for usage in mobile networks	Yes
Protocol must have development tools which help to use it	Yes
Protocol must be popular and used in the Internet	Yes

Mobile devices are different and developed for different purposes. It can be powerful smartphone with multicore CPU and couple GB of RAM, or it can be small board with some sensors and small amount RAM up to 100 MB. HTTP has relatively big amount of technical data transferred from a server to a client and back as it was shown in the paper "In HTTPS potential traffic overhead for mobile devices" [1]. For not very powerful devices, additional overhead can be not acceptable.

HTTP is the most popular application layer protocol in the Internet, thus we need to consider only the protocols, which are similar to HTTP and can satisfy the criteria.

One of these protocols is CoAP (constrained application protocol) lightweight alternative of HTTP. The protocol works similar to HTTP, as it shown on Figure 1.

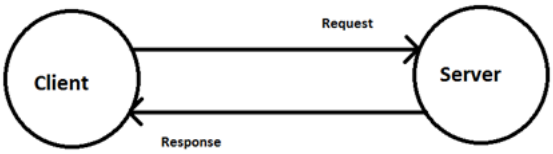

Fig. 1. Client-server communication in CoAP

CoAP supports GET, POST, PUT and DELETE methods [3]. This protocol can be successfully used with IoT (internet of things) devices. These mobile devices usually have not very powerful hardware and cannot have fully functional operational system installed. Identification criteria for CoAP application layer protocol is presented in Table 3.

Table 3

Identification criteria for CoAP

Criteria	Value
Protocol must support the client-server communication	Yes
Protocol must be available for usage in mobile networks	Yes
Protocol must have development tools which help to use it	Yes
Protocol must be popular and used in the Internet	Not very popular

CoAP is relatively new protocol and it becomes popular for IoT. It can replace HTTP in some cases. For instance, CoAP protocol was used for healthcare monitoring sensors to view the status of patients via browser [4]. Therefore, it can be also considered as reliable application protocol as well.

Another protocol, which can be considered as alternative for HTTP is SPDY (pronounced SPeeDY). This protocol claims to be faster than HTTP. On Figure 2 the average page load for website over 3G (mobile) networks has been shown [5]. The authors have shown that SPDY is a little bit faster than HTTP in 3G networks.

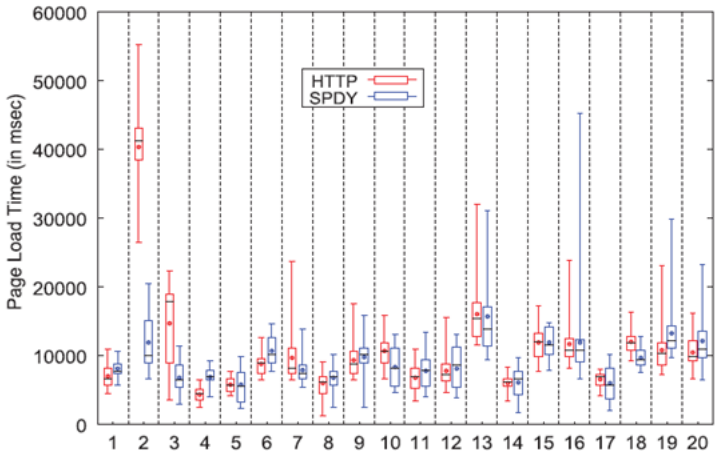

Fig. 2. Page load time for different web sites with HTTP and SPDY over a 3G network [5]

However, the results shown on Figure 2 give improvement from 4 % to 56 % which depends on a website. A previous result, which was achieved by Google, has been shown improvement between 27–60 % [6].

In Table 4, the identification criteria has been shown for SPDY protocol.

Table 4

Identification criteria for SPDY

Criteria	Value
Protocol must support the client-server communication	Yes
Protocol must be available for usage in mobile networks	Yes
Protocol must have development tools which help to use it	Yes
Protocol must be popular and used in the Internet	Not very popular

The research objective. Our objective is comparison of application layer protocols, which are popular and satisfy our identification criteria. The result of this comparison should give us an idea, which protocols are better to use. It is also possible to identify which protocol more suitable for specific application types.

The statement of basic materials. All these protocols have similar principle of work and can be used for communication between a client and a server. We were trying to compare similar to HTTP protocols, because HTTP is the most popular protocol in the internet [2]. In Table 5 we presented comparison criteria for CoAP and SPDY protocols, which, we think, can be used as an alternative to HTTP protocol.

Table 5

Comparison criteria for HTTP, SPDY and CoAP protocols

Criteria	HTTP	SPDY	CoAP
Amount of a technical data transferred	High	High	Low
Protected data transfer reliability and performance	HTTPS (TSL/SSL)	TSL/SSL	DTLS (RSA and AES or ECC and AES)
Popularity of the protocol in terms of software development tools	Very popular, a lot of tools available for software development	Tools are available for software development	Tools are available for software development
Overall performance	Good	Good	Good
Client server communication drawbacks and characteristics	Can be used for communication for different clients and servers. Very popular. High amount of technical data can be transferred	Protocol meant to replace some parts of HTTP, reduces loading time of web pages. Makes almost no difference between data transferred in mobile networks. Not very popular	Protocol Used for IoT applications, low amount of technical data, but has limitations

Http protocol

Amount of technical data transferred between a client and a server can be high. This depends on type of information (text, image, sound or video) which is transferred and also depends on developers who write the software [1; 7].

HTTP has secure alternative, which uses SSL/TSL to protect data transferred. Many programming languages like Java, C #, C++, support HTTP.

Overall performance of applications, which use HTTP(S), depends on particular applications and technologies which developers decided to apply. HTTP protocol was designed to be reliable, expandable and all-purpose protocol. Any type of information can be sent by using HTTP.

SPDY protocol

SPDY targets some parts of HTTP protocol where cannot deliver better performance. Google's SPDY uses some techniques (header compression, pushing and multiplexing) to decrease amount of data transferred from a client to a server and as a result improve speed communication. Different researches have shown ambiguous results. On Figure 2 the page load is shown for different web sites using HTTP and SPDY and its only 4 % of improvement for mobile networks [5]. Another research, which was carried out by C. Roseti et al. have shown different result. They were using latest SPDY version to measure the page load time over satellite network. As it can be seen on Figure 3, amount of data transferred by using SPDY is almost 60 % less than the amount of data transferred by using HTTP [8].

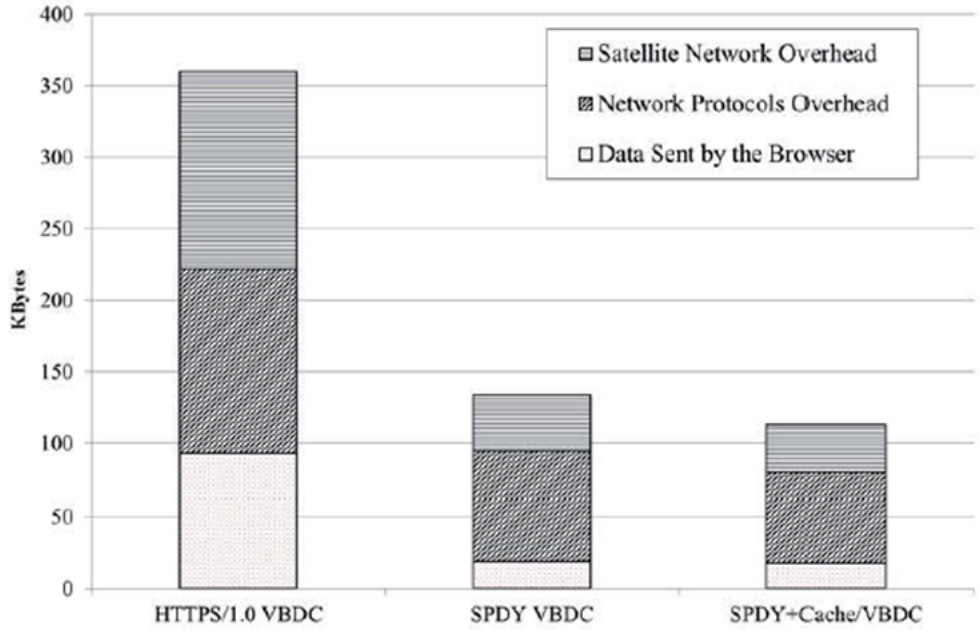

Fig. 3. Amount of data transferred using satellite networks

SPDY is not very popular and there are much more less development tools and examples how to use SPDY.

CoAP protocol

CoAP is a protocol which is used by IoT devices to communicate with servers. Usually, IoT devices are devices with limited battery, memory and processing capabilities and CoAP can decrease amount of data transferred for these devices.

For developers CoAP is something that can be used very easily because of variety of tools and examples for many programming languages (Java, C#, C++, Javascript etc.). Moreover protocol is similar to HTTP from software development point of view, which makes it even more attractive for developers.

Conclusions and propositions. In the paper, we identified the most popular protocol in the Internet. As a result, HTTP is the most popular protocol; therefore, we considered protocols, which are similar to HTTP. Identification criteria were created to identify protocols, which can be used instead of HTTP in some cases. There only two application level protocols passed identification criteria. They are CoAP and SPDY. To compare protocols between each other we created comparison criteria.

After comparison and analysis of previous researches, it is clear that CoAP and SPDY perform better than HTTP giving up to 50% performance increase. However, areas where CoAP can be applied are restricted. In addition, SPDY positioned himself as experimental protocol from Google. SPDY is not popular nowadays and developers prefer using HTTP. Popularity of HTTP, large amount of examples for different programming languages and amount of application, which already have been created, are the most important factors, which influence on selection of an application level protocol for software development.

For the future researches, consideration of how developers using application level protocols can be priority direction. Usage of HTTP by software developers can be improved to increase performance of applications they create.

References

1. Kazymyr, V., Mokrohuz, A. (2016). *In HTTP(S) potential traffic overhead for mobile devices*. International Journal "Information Theories and Applications", vol. 23, no. 4, pp. 383–393.

2. *Global internet phenomena report* [Online]. Available: https://www.sandvine.com/downloads/general/global-internet-phenomena/2014/1h-2014-global-internet-phenomena-report.pdf.

3. Shelby, Z., Hartke, K., Bormann, C., and Frank B. (2014). *The Constrained Application Protocol (CoAP)*, IETF RFC7252, June 2014.

4. Ugrenovic, D. and Gardasevic, G. (2015). *CoAP protocol for Web-based monitoring in IoT healthcare applications. Telecommunications Forum Telfor (TELFOR),* 23rd, Belgrade, pp. 79–82.

5. Erman, J., Gopalakrishnan, V., Jana, R. and Ramakrishnan, K.K. *Towards a SPDY'ier Mobile Web?* In IEEE/ACM Transactions on Networking, vol. 23, no. 6, pp. 2010–2023, Dec. 2015.

6. Google, *"SPDY: An experimental protocol for a faster Web"* [Online]. Available: http://www.chromium.org/spdy/spdy-whitepaper.

7. Pengcheng Jiang; Fang Liu; Huan Wang; Chenyu Li. *Characterizing HTTP Traffic of Mobile Internet Services in Provincial Network. I*n Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2014 Sixth International Conference on, vol. 1, pp.78–81, 26-27 Aug. 2014.

8. Roseti, C., Salam, A. A., Luglio, M. and Zampognaro, F. (2015). *SPDY over satellite: Performance optimization through an end-to-end technology*. Telecommunications and Signal Processing (TSP), 2015 38th International Conference on, Prague, pp. 1–6.

Kazymyr Volodymyr – Doctor of Technical Sciences, Professor, Chernihiv National University of Technology (95 Shevchenka Str., 14027 Chernihiv, Ukraine).
Казимир Володимир Вікторович – доктор технічних наук, професор, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна).
Казимир Владимир Викторович – доктор технических наук, профессор, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).
E-mail: vvkazymyr@gmail.com
ORCID: http://orcid.org/0000-0001-8163-1119
Scopus Author ID: 56644727300

Mokrohuz Andrii – PhD student, Chernihiv National University of Technology (95 Shevchenka Str., 14027 Chernihiv, Ukraine).
Мокрогуз Андрій Олександрович – аспірант, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна).
Мокрогуз Андрей Александрович – аспирант, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).
E-mail: myworkingadr@gmail.com

Kazymyr, V., Mokrohuz, A. (2016). Selecting a base application layer protocol to transfer data in the client-server architecture. *Technical sciences and technologies*, no. 3 (5), pp. 182–187.