A Novel Signaling Protocol (ARCSPXP): Case Study on Synchronization of Educational Data

Süleyman Eken[1], Fidan Kaya Gülağız[1(✉)], Ahmet Sayar[1], Adnan Kavak[1], Umut Kocasaraç[2], and Zana İlhan[2]

[1] Computer Engineering Department, Kocaeli University, Umuttepe Campus, 41380 Izmit, Turkey
{suleyman.eken, fidan.kaya, ahmet.sayar, akavak}@kocaeli.edu.tr

[2] Ardıç Arge Bilgi ve Teknoloji Cozumleri, Tübitak TEKSEB, 41470 Gebze, Kocaeli, Turkey
{umut.kocasarac, zana.ilhan}@ardictech.com

Abstract. To define the state of communication, a lot of signaling protocol has been studied by many researchers. In this paper, we firstly focus on a new hybrid optimized signaling protocol, ARCSPXP (ARDIC Cloud Service Platform Extension Protocol), which is specialized to mobile devices which are communicating with cloud based services via its internet connection. Then, we test usability and feasibility of ARCSPXP signaling protocol on synchronization of educational data (text, images, media, etc.) stored tablets, proxy servers, and cloud servers which are system actors of the most important educational project in Turkey. Experimental results show that ARCSPXP provides a more manageable and easy to use integration structure for mobile devices.

Keywords: ARCSPXP · Data synchronization · Machine-to-cloud signaling · M2M signaling · Proxy server

1 Introduction

Depending on cloud-based services, a lot of new technologies have begun to emerge. The basic need is to track mobile devices using services and provide access to these devices if necessary. With these needs, importance of signaling protocols has increased day after day. A signaling protocol is really just any protocol that can send a signal or message from one specific computer to another specific computer. The aim of our study is to evaluate the feasibility and efficiency of ARCSPXP signaling protocol on synchronization of educational data. Architecture, communication primitives and usage areas of ARCSPXP will be explained in the following sections.

Turkish Ministry of Education has recently launched FATIH (Movement to Increase Opportunities and Technology) project that is primarily based on employing tablets and smart boards in classes for students and teachers and using educational data stored in centralized cloud based servers. Currently, there are three main actors of the ecosystem within the scope of this system: (i) a cloud-based SaaS services, (ii) the smart boards in classes, and (ii) tablets at students and teachers/educators. These actors

R. Agüero et al. (Eds.): MONAMI 2016, LNICST 191, pp. 31–43, 2017.
DOI: 10.1007/978-3-319-52712-3_3

are foreseen to actively exchange data with cloud services within the project ecosystem. However, there exist limitations due to network traffic and infrastructure between the tablet clients and cloud servers that store educational data. Many schools have limited internet infrastructure. The limited network infrastructure and increase in educational data size, which are two major parameters affecting the system performance, cause increase in network delays and degradation of system performance in case of that a user in school network wants to access to the cloud servers to download educational data. FATIH project does not have school level (client-side) proxy servers. To solve these limitations and problems, a school level client-side proxy server and an extension framework for integrating it into the cloud system are proposed. This framework includes cloud signaling and synchronization functionalities in tablet client and proxy server. Cloud signaling module enables tablet-proxy-cloud communication. Synchronization module guarantees to keep the same data (i.e. educational files) at different locations in a consistent manner. So, proxy server is a solution approach to both decreasing network traffic and increasing the efficiency in data transfers between the end users (tablets) and cloud servers in FATIH project [1].

In this paper, we focus on cloud signaling module of the proxy server-based solution, especially ARCSPXP signaling protocol. Signaling module is developed by adapting ARDIC's ArCloud platform according to our requirements. ArCloud (ARDIC Cloud Services Platform) [2] is a high performance, extensible cloud platform designed for mobile devices to provide user, application, device and security management.

The remainder of this paper is organized as follows. Section 2 presents relevant research on mostly used signaling protocols. Section 3 introduces the architecture of ARCSPXP protocol. In this section, its communication primitives and application areas are also mentioned. Section 4 presents building an ARCSPXP application for data management and synchronization. Section 5 draws a conclusion and suggests some future works.

2 Related Works

With needs for tracking mobile devices and providing access to these devices if necessary, importance of signaling protocols has increased day after day. Three different types of signaling protocols are the most widely used signaling protocols in communication: H.323, SIP (Session Initiation Protocol), XMPP (Extensible Messaging and Presence Protocol). Detailed information about these protocols will be given in next paragraphs. H.323 is developed to enable multimedia communication over a computer network and to provide audio and video transmission. H.323 protocol is defined as a binary. Message format of H.323 protocol is determined as ASN.1 (Abstract Syntax Notation One). PER (Packet Encoding Rules) procedures are used for message encryption. These encoding rules belong to ASN.1 message format. ASN.1 is a standard that responsible for determining rules for preparing, transmitting, encrypting and decrypting data during the telecommunications on the computer network [3].

H.323 protocol has too many protocols except its components. These protocols have different processes like control of the record situation, control of the search signal, real time transfer etc. Some of these protocols are H.225 Registration, Admission and

Status, H.225 Call Signaling, Audio Processing, H.245 Control Signaling, Real Time Transport Protocol and Real Time Transport Control Protocol. A various applications of H.323 exist in corporate and home user environments such as IP Telephony, video conferencing, multimedia call centers, and telecommuting. Disadvantage of H.323 is to be VOIP (Voice over Internet Protocol) oriented. Its main task is to provide messaging before transferring audio and video, so it has limited messaging infrastructure.

The second communication protocol is SIP. It creates, sets up and finishes VOIP phone calls and has a text-based message structure and defines messages which will be sending between couples during a call. Also SIP can be used in many areas such as video conferences, fax transmission, file transfer and transferring status information [4]. Elements of the SIP protocols are defined by RFC 3261, which are divided into two types: user agents and servers. User Agents define endpoints, which are responsible for managing a SIP session and transmission of SIP messages. There are also four different server types in SIP protocol: redirect server, proxy server, registrar server, and location server [5]. Besides being a text-based, SIP is also http-like protocol. Its messages are similar to http messages. Request and response messages defined by RFC3261 are the two types of messages in SIP. An example of a SIP request message is below [6]. SIP also provides services for media and VOIP oriented, so it has limited messaging infrastructure.

The third one is XMPP (Whitepaper), originally named Jabber. It is an XML based signaling protocol for message oriented middleware. It enables two endpoints on the Internet to mutually transfer any structural information. Also, it allows message, file, and status transfer among more than one unit (user, device, etc.). Although XMPP is first intended for instant messaging, its XMPP has been improved for larger systems such as cloud computing in parallel with an increase in human requests for communication [7]. XMPP finds large areas of application such as instant messaging [8], interactive social media [9], and collective work flow [10], internet of Things, multi-agent systems, and cloud computing [11]. It conveys not only text or status information, but also voice and video messages in real-time is transmitted between users.

XMPP has three basic XML elements: status (presence), messages and iq (info/query). Notification mechanism that allows entities to acquire network usability information is provided by <presence> element from an entity which the other entities are member of it. An entity sends information to other entities by means of <message> element asynchronously. Request and responses are carried by <iq> element.

The fourth signaling protocol is MQTT [12, 13], a lightweight application layer protocol designed for processing and memory constrained devices. It utilizes topic-based pub/sub architecture allowing multiple clients can establish a connection. MQTT supports three QoS levels. QoS level 0 guarantees best-effort delivery service. No retransmission or acknowledgment is defined. QoS level 1 means that every message is delivered at least once and acknowledgement is required. In QoS level 2, a four-way handshake mechanism is used to ensure the delivery of a message exactly once.

The fifth one is WebSocket that lets clients and servers to communicate over the same TCP connection bi-directionally. Clients and servers can initiate a message and exchange any messages in any format such as JavaScript Object Notation (JSON) [14]. The last one is that SSE enables efficient server-to-client streaming of text-based event

data generated on the server. In this approach, servers send event data to clients using regular HTTP [15].

H.323 and SIP was wide widely used as the foundation to build VoIP services on desktops. XMPP, MQTT, WebSocket, SSE have been adopted to create the communication infrastructure between mobile devices and cloud services.

Besides the above-mentioned protocols, some researchers have studied on communication protocols and developed new ones to provide cloud based services. Bertacchi [16] proposed a method and system for providing compatibility between telecommunication networks using different transmission signal systems. Pospischil et al. [17] dealt with push location-based applications and therefore users need to explicitly subscribe to services which take advantage of location information. Their push service architecture is SMS, WAP (Wireless Application Protocol) and SIP based solution. Protocols such as SMTP, SMS and WAP have not extensible signaling infrastructure for cloud-based services, because they are acting independently of the message content. Similar architectures is developed using protocols such as H.323 and SMTP, but they are media and VOIP oriented.

ARCSPXP is a hybrid protocol and not a standard. Features of ARCSPXP, XMPP, SIP, and H.323 protocols are summarized by Kaya et al. [18]. However, we give all things about ARCSPXP in detail in this paper. If we compare the semantic structures and functional messaging background of protocols that are examined in our study, ARCSPXP signaling protocol has a significant difference in terms of integration support of SaaS services. This difference provides flexibility to ARCSPXP protocol's SaaS services for expanding message scale. XMPP and SIP protocols are general purposed protocols and these protocols try to solve the security problems at transport layer except protocol messaging. In addition to supporting functionalities of SIP, H.323 and XMPP, ARCSPXP also provides enhanced security solution with Authentication, Authorization and Accounting (AAA) support. ARCSPXP protocol is not a general purposed protocol so today's clients are not use this protocol widely, But ARCSPXP is specialized about large scale service layer signaling. So that ARCSPXP provides a more manageable and easy to use integration structure for mobile devices. H.323 was designed with a good understanding of the requirements for multimedia communication over IP networks, including audio, video, and data conferencing. SIP protocol is specialized in audio/video communication. It specialized about definition of VoIP session and life cycle of VoIP session. XMPP protocol is different from SIP protocol and care about the semantic message structure and offline state of client. XMPP has better message support for instant messaging and state communication. Nevertheless, XMPP uses too many port ranges so this may cause security vulnerabilities [19]. There is a possibility of contamination harmful files, viruses, trojan etc. during file sharing and data transfer between clients. Many advanced communication and collaboration system contains some of these protocols in itself. ARCSPXP cares about the lifecycle of message and related task that we want to monitor. ARCSPXP has ability to deliver the same message to multiple clients with structured data types using streaming structure.

3 Basics of ARCSPXP

ARCSPXP is an optimized signaling protocol that is specialized to mobile devices which are communicating with cloud based services via its internet connection. ARCSPXP is also used service level signaling for cloud internal signaling processes in the ArCloud. In this chapter, ARCSPXP's architecture, communication primitives and application areas are also mentioned. This structure consists of four layers: security layer, signaling layer (ARCSPXP), service layer and data layer.

- Security Layer: This layer obliged to provide necessary security infrastructure for ArCloud services. This security layer provides some security mechanisms for ArCloud's threats that are coming from internet. Some of these threats are Access Throttling, Deterrent Controls, Preventative Controls, Corrective Controls and Detective Controls etc.
- Signaling Layer: This layer enables the execution of ARCSPXP protocol and ArCloud signaling.
- Service Layer: This layer allows realize the ArCloud's services for mobile devices. Data synchronization and backup service that is provided for student's tablets will be implemented in this layer. Also this layer provides services such as detecting device location (at school or outside the school) etc. This layer inherently has a scalable architecture. Also the status and usage statistics of a service is provided by this layer. This layer allows use of more than one client simultaneous with Multi-Tenant architecture and this property provides isolation between clients in terms of data and service usage
- Data Layer: Services provided by ArCloud and mobile user data are stored at this layer. This layer is provided with the combine of different technologies data storage system to keep user and service data together. Many control processes are made at this layer. Some these controls are data storage, data analysis, data integrity and data information. The control operation is performed by this layer at periodic intervals.

ARCSPXP is a protocol created for facilitating communication between "client-server" and "server-to-server" and monitoring the connection between the client-server. Detailed information about this protocol will be given in following subtitles.

3.1 ARCSPXP Architecture

ARCSPXP is a hybrid protocol and it has improved in terms of the data representation. The semantic structure of XMPP protocol is preserved. Also XMPP uses XML data structure for data representation but ARCSPXP uses encapsulated JSON for data representation. With this data structure a new data structure oriented protocol is created, so XMPP protocol is optimized without changing the specific partitions of XMPP protocol. The critical part of XMPP optimization is expensive parsing cost of XML, which has been optimized with the defined JSON data structure.

ARCSPXP supports the plain text structure and Simple Authentication and Security Layer (SASL) authentication standards of XMPP. Clients are being included to

authentication process by ARCSPXP protocol with the approach of hard ware root of trust. Protocol is extended with using structures of hardware based such as MAC address and DMI (Desktop Management Interface) for hardware root of trust.

ARCSPXP protocol needs two types of server to run. The first of these is the server named as ARDIC Cloud Service Platform Database Management System (ARCSP DMS). This server makes the authentication process of between client and server and then manages connections and forwards the message to the clients. The second server is the server named as Message Server. This server is responsible for sorting server's messages and sending messages to client. ARCSPXP uses certificate which is generated with public-key infrastructure (PKI) for Authorization and Communication Security. ARCSPXP uses parameters such as user name, password, certificate and the identity of the client device for client authentication. However, certificate and device id are enough for a limited authentication at the first installation for the client.

3.2 Communication Primitives

The message format in ARCSPXP consists of two parts. Mandatory fields of message are available in the first part, and they are XML-based. In the second part has a parametric type of message. These sections are filled through Cloud APIs which are open to the outside and JSON format. An example of a message belonging to ARCSPXP protocol is following:

```
<message id="12-123" from="caller" to="callee">
  <command>
    <JSON AREA>
  </command>
</message>
```

Due to the fact that XML is powerful structurally (structure and namespace are extendible easily), the main structure of the message is expressed in XML. To take advantage of the ease of JSON data presentation, contents of the command is sent in JSON format.

It is necessary three types of processes to work ARCSPXP protocol: The process of verifying the identity of clients, management of connections and forwarding of messages to clients. There are three different types of message on ARCSPXP: (i) Simple Message, (ii) Mandatory Delivery Message (MD), and (iii) Hybrid Message.

3.2.1 Simple Message

This messaging is used for the management of "client-server" connection. In this type of messaging, when the client sends a message to the server or vice versa, delivery and receipt status of the message is important and is confirmed automatically.

3.2.2 Mandatory Delivery Message

In this type of messaging, the status of message transmission is important and the protocol guarantees that the message has been transmitted, parsed and processed as expected. Commands used for client management and policy management use this message type.

When a Mandatory Delivery message is sent to a device, the message is recorded into message server firstly. After processing and queuing the message, it is sent to the client through ARCSPDMS. After the message is sent, message server is waiting ACK message from the client. Message server waits the answer for over a predetermined time-out. If ACK message is received, next message to be sent to the client starts processing. If the message sending process fails, the message server signs this message to send later and it takes care of other client's messages. As the client and the server is working on independent networks in this type, the possibility of sending the same message more than one arise in some extreme cases. As a solution to this problem, ArCloud is designed the messages to support "Idempotent Message Pattern". Also, command has been in the message reports about their own status (the status of operation, what stage operation is, and etc.) asynchronously. Figure 1 shows that flow diagram of MD message.

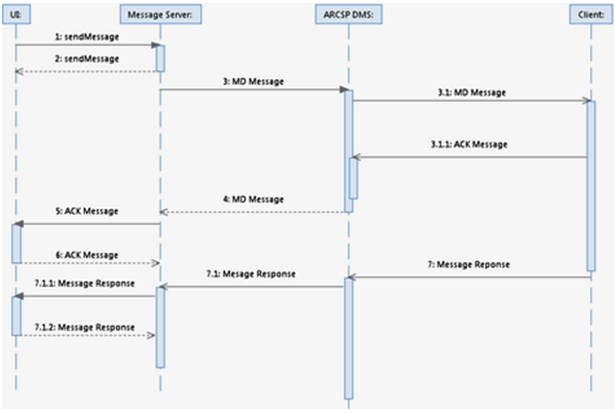

Fig. 1. MD message flow diagram

3.2.3 Hybrid Message

Hybrid message is a type that HTTP and ARCSPXP protocols are used together. These message types are derived from MD, so that it is guaranteed delivery status of message. Sync commands are sent with this message. It works same principle with the MD message, but information about-how to ensure binary data communication-exists in commands. When the client receives these messages, it does some of the operations related to the commands via the HTTP protocol. Flow diagram of hybrid message type is as shown in Fig. 2. Other messages can be concealed in ACK message in ARCSPXP protocol.

3.2.4 Application Areas

ARCSPXP has four main different application areas. These areas are Internet Enabled Sensors & Service Communication Signaling, Machine to Machine Communication Signaling (Tablet to IoT GW), Cloud Internal Server Signaling and Mobile OS to Cloud (Mobile to Cloud).

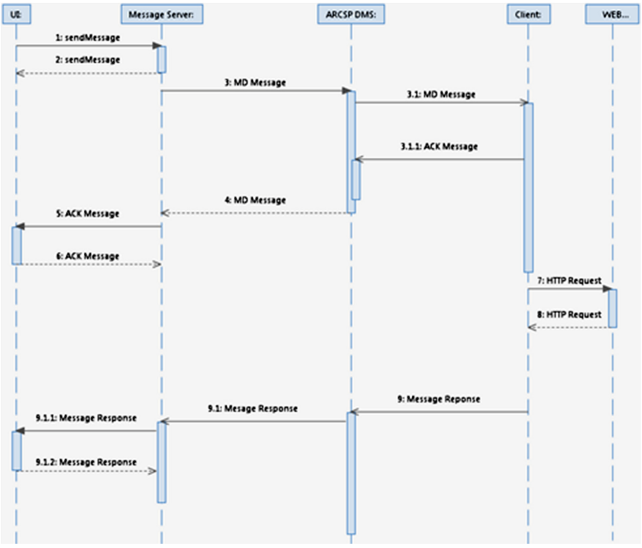

Fig. 2. Hybrid message flow diagram

- Internet of Things (IoT): The idea is to provide a security layer helping to define who can talk to whom and about what. Sensors and Actuators can talk each other and IoT gateways.
- Machine to Machine (M2 M): ARCSPXP protocol servers can be used as an IoT gateway to signal sensors and actuators in a local network.
- Server to Server: ARCSPXP protocol can be used for signaling of communication between the servers, The server signaling is different in some ways of the clients, these are relevant to theirs life cycles, ARCSPXP can be used as an abstract life-cycle management protocol to manage and monitor lifecycle of the servers and server clusters.
- Mobile to Cloud: In today's world the most critical thing is the presence/activity of the mobile clients which is using cloud based services. This approach is followed by many global service providers.

4 Building an ARCSPXP Application for Data Management and Synchronization

This section explains overall structure of proxy server-based approach as shown in Fig. 3. Tablets are connected with cloud from school or outside of the school. Tablets which are outside of the school access cloud server directly to access educational data. Tablets which are in school access data via proxy server. Educational data will be shared among tablets, will be organized according to the following scenarios with proxy server:

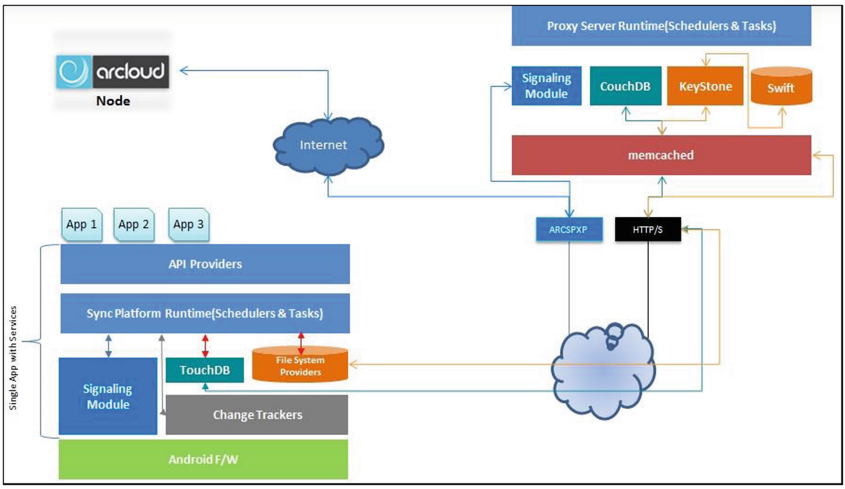

Fig. 3. High level communication between inter-components

- Getting the educational data that published via cloud server.
- Getting educational data from cloud server when the network traffic is low.
- Organizing the educational data that will be shared among student's tablets.
- Organizing the data requests when the network traffic is high at school.
- Transmitting the educational data updates from proxy server to cloud server.

Proxy server-based approach has generally two types of communication: (i) tablet-cloud server communication and (ii) tablet-proxy server communication. Tablet-cloud server communication scenario includes situations when tablet is used outside of the school or inside of the school without connected with proxy servers. If tablets are connected with the internet will also be connected to cloud, it does not matter whether tablets are in school or outside of the school. Cloud server always determines whether tablets will access educational data from cloud server or proxy server by identifying from where the tablet is connected. In other words, cloud server is decisive actor.

Figure 4 shows a flow diagram depicting that a tablet is connected to cloud server from outside of the school and accessing educational data from cloud server. Communication between tablet and cloud server will be provided via ARCSPXP protocol in this scenario. If tablet has internet connection, this communication would always be active. Downloading data from cloud server would be active only during content transfer, and the system has a structure that does not require continuous connection between cloud server and tablet.

The tablets which are connected with internet first time would be connected directly to the cloud server. For each new connection, a control will be started via cloud server. With this control, cloud server will decide whether tablet is an educational tablet or not. The cloud server will also decide whether tablet needs any adjustment or not. If a tablet wants to get any service from cloud server, it must be defined on cloud server with the following information before connection:

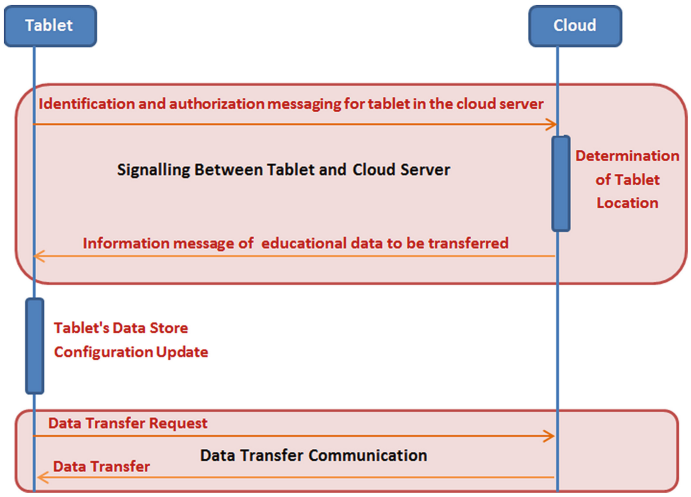

Fig. 4. Flow diagram of cloud signaling and data transfer when tablet is outside of the school

- Tablet's unique identity,
- User information assigned to the tablet,
- If the tablet assign a pre-defined group (e.g. group Istanbul), group assignment must be done,
- School ID, class and so on, which the tablet belonged.

After definition of required information, cloud server controls the tablet whether tablet is an educational tablet or not. If tablet is an educational tablet, cloud server will progress predefined operations. These operations are shown in Fig. 5. Detecting the location of the tablet (in school or outside of school) is one of these predefined operations. Later, if the tablet is in school, cloud server sends signaling message, including information about its proxy server to be able to upload and download data to this tablet.

It is possible minimizing the bandwidth usage between client-server and removing overload for creating message at client side and sending it. We can see the impact of this method to ARCSPXP bandwidth usage. Suppose that, a status message (ARCSPXP protocol monitors the activities of the clients via status messages) is transmitted at an average of 10 min and also a command message (ARCSPXP protocol transmits operations about functionality of ArCloud SaaS via command type messages) is transmitted at an average of 60 min. 168 ($1 \times 24 + 6 \times 24$) message per day are send with XMPP protocol for such communication. However, these status messages can conceal in ACK messages in the architecture developed for ARCSPXP. In other words, when the command message was sent to the client machine, status message have been cancelled in ACK messages arrived from client. So, it avoids the necessity to be sending some status messages at server side and to receive response from client. Thanks to this structure, this corresponds to possibility of eliminating 24 status messages to be sending and be received response daily. This provides decreasing bandwidth usage at server side by up to 14%. Also, Fig. 6 shows sync operation time according to document type and size depending on the number of users.

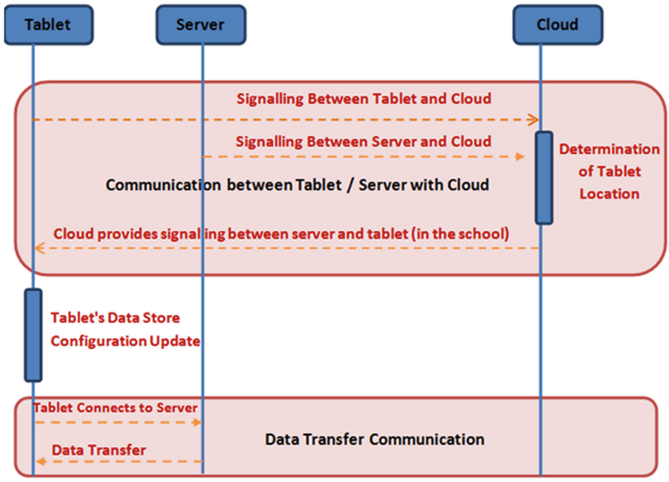

Fig. 5. Flow diagram of cloud signaling and data transfer when tablet is in the school

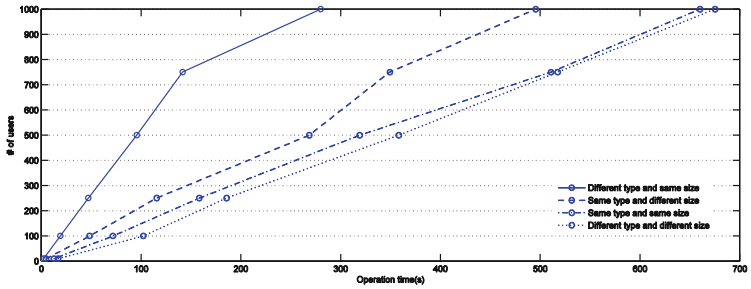

Fig. 6. Processing syncronization time depending on the number of users

5 Concluding Remarks

Currently due to the need of tracking mobile devices and providing access to these devices the importance of signaling protocols has also increased. In this paper, we explain a new hybrid optimized signaling protocol, ARCSPXP. It is specialized to mobile devices communicating with cloud based services by means of its internet connection. Results show that ARCSPXP provides a more manageable and easy to use integration structure for mobile devices. In the future, we will study optimization of messages sending and receiving between communication nodes.

Acknowledgments. This work is supported by the TUBİTAK under grant EEEAG 113E033 within 1003 Fatih Project Call.

References

1. Eken, S., Kaya, F., Sayar, A., Kavak, A., Şahin, S.: A method for localization of computational node and proxy server in educational data synchronization. In: Mumtaz, S., Rodriguez, J., Katz, M., Wang, C., Nascimento, A. (eds.) WICON 2014. LNICSSITE, vol. 146, pp. 180–190. Springer, Heidelberg (2015). doi:10.1007/978-3-319-18802-7_26
2. ArCloud. http://www.ardictech.com/indexphp/tr/cozumlerimiz/arcloud. Accessed 15 March 2015
3. Thom, G.A.: H.323: the multimedia communication standard for local area network. IEEE Commun. Mag. **34**(12), 52–56 (1996)
4. Rosenberg, J., Schulzrinne, H., Camarillo, G., et al.: SIP: Session Initiation Protocol. IETF RFC 3261. http://www.ietf.org/rfc/rfc3261.txt. Accessed 15 March 2015
5. Whitepaper, SIP Server Technical Overview. http://www.radvision.com/radvision/PDF/sip-server-platform/SIPServerTechnicalOverviewWhitepaper.pdf. Accessed 15 March 2015
6. Johnston, A., Donovan, S., Sparks, R., Cunningham, C., Summers, K.: Session initiation protocol (SIP) basic call flow examples. IETF RFC 3665. http://www.ietf.org/rfc/rfc3665.txt. Accessed 16 March 2015
7. Hornsby, A., Walsh, R.: From instant messaging to cloud computing, an XMPP review. In: IEEE 14th International Symposium on Consumer Electronics, pp. 1–6 (2010)
8. Saint-Andre, P.: Jingle: jabber does multimedia. IEEE Multimedia **14**(1), 90–94 (2007)
9. Hoekman, K., Ide, M., Deryckere, T., Martens, L.: XMPP and iDTV or how to make television a social medium. In: IEEE 4th Consumer Communications and Networking Conference, pp. 686–690 (2007)
10. Roczniak, A., Melhem,, J., Lévy, P.E., Saddik, A.: Design of distributed collaborative application through service aggregation. In: IEEE 10th International Conference on Distributed Simulation and Real-Time Applications, pp. 165–174 (2006)
11. Wagener, J., Spjuth, O., Willighagen, E.L., Wikberg, J.E.S.: XMPP for cloud computing in bioinformatics supporting discovery and invocation of asynchronous web services. BMC Bioinform. **10**, 279 (2009)
12. Dinesh, T., Xiaoping, M., Alvin, V., Hwee-Xian, T., Colin Keng-Yan, T.: Performance evaluation of MQTT and CoAP via a common middleware. In: IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP) Symposium on Sensor Networks, pp. 1–6 (2014)
13. Niccolò, D., Walter, C., Kris, S., Giuseppe, M., Gianluca, R.: Comparison of two lightweight protocols for smartphone-based sensing. In: IEEE 20th Symposium on Communications and Vehicular Technology in the Benelux, pp. 1–6 (2013)
14. RFC 6455. http://tools.ietf.org/html/rfc6455. Accessed 17 March 2015
15. Vinaski, S.: Server-Sent events with yaws. Internet Comput. **16**(5), 98–102 (2012)
16. Bertacchi, L.: Method and system for providing compatibility between telecommunication networks using different transmission signal systems. Patent no: 6625461 (2003)
17. Pospischil, G., Stadler, J., Miladinovic, I.: A location-based push architecture using SIP. In: IEEE 4th International Symposium on Wireless Personal Multimedia Communications (2001)

18. Kaya, F., Eken, S., Ilhan. Z., et al.: A comparative study of signaling protocols for data management and synchronization in fatih project with school level cloud proxy server deployment. In: IEEE 3rd Symposium on Network Cloud Computing and Applications, pp. 133–136 (2014)
19. Saint-Andre, P.: Extensible Messaging and Presence Protocol (XMPP): Core. http://xmpp.org/rfcs/rfc6120.html. Accessed 18 March 2015