Implementation and Evaluation of IoT System Using Cloud Storage Platform

Abd Qawiyyel Mateen
Universiti Brunei Darussalam
Jalan Tungku Link
Gadong BE1410
mateenrajid@gmail.com

Ong Wee Hong
Universiti Brunei Darussalam
Jalan Tungku Link
Gadong BE1410
weehong.ong@ubd.edu.bn

ABSTRACT

As the internet has become more accessible, the use of Internet of Things (IoT) systems is increasing. An IoT system can be accessed either by directly connecting to the network configured with external access and appropriate port forwarding; or through a cloud server. Direct access through network may not be possible for networks with restriction on external access, and requires technical know-how to configure the port forward. The use of cloud server is convenient that the users simply have to sign up an account at and use user friendly application to setup the system. This approach however requires the manufacturer to main the cloud server, has the concern of having users' data accessible to the owner of the cloud server, and does not allow interoperability of devices and systems across different manufacturers. In this paper, we have proposed the use of publicly available cloud storage services to provide the accessibility of the IoT devices without relying on a specific cloud server, and without requiring the manufacturer to maintain their own cloud server. The proposed approach will allow interoperability between devices or systems from different manufacturers. This paper describes the concept, implementation and evaluation of the proposed cloud storage based IoT system. The proposed IoT system has been implemented to work with Google Drive, Dropbox and Microsoft OneDrive. The performance of the proposed system has been evaluated against dedicated IoT cloud servers including Microsoft Azure IoT, Google IoT, CloudMQTT (CMQTT) and Eclipse IoT. The results show that can be effectively used in systems that are not time critical.

CCS Concepts

• Networks → Network protocols. **• Networks → Network performance analysis**

Keywords

Internet of Things (IoT); Cloud; Protocols; Internet; Network; Gateway; IP; Storage

1. INTRODUCTION

Internet of Things (IoT) system allows users to connect and interact with devices over the internet. As the internet has become more accessible, the use of IoT systems is becoming convenient and common. Various commercial IoT systems have been produced in the market. Traditionally, an IoT system can be accessed by directly connecting to its server or gateway at its locality through its network. This approach can be considered a distributed approach whereby each IoT system runs its own server, usually referred as the gateway. This approach requires that the network has external access, i.e. a static public Internet Protocol (IP), and that its network has been configured with appropriate port forwarding. This is certainly a difficult task for a non-IT user. Furthermore, certain places have restricted network where it prohibits user to open port. The current trend is to use a centralized cloud server. The user of an IoT system creates a user account in its manufacturer's cloud server and set up the IoT system using the user-friendly application of the manufacturer. The centralized approach demands the manufacturer of the IoT system to maintain its server to meet the increased number of IoT systems. Not all IoT system creators have the resources to maintain a server to scale with the increased use of their systems. This limits the production of IoT system to large organizations, and individual technoprenuers has to deploy their systems through major IoT cloud servers such as Microsoft Azure IoT and Google IoT. Nowadays there are many IoT companies providing IoT servers but IoT systems configured in these servers are restricted to operate with the specific server only. If the chosen server discontinued its operation, users can't easily switch to another IoT cloud providers without learning and setting up in a different server. In terms of privacy, centralized IoT cloud servers store all their users' data in their own format. Users will never know if their activity and data in these IoT cloud are being accessed without their permission.

In this research, cloud storage based IoT system is proposed to resolve the problems that have been mentioned above. In the proposed system, a file is used to represent each IoT device. Most cloud storage platforms allocate certain amount of space to their user when they sign up. Creator of the IoT system does not need to maintain a server and can rely on the well-established cloud storage service provider. Accessing cloud storage is done through the internet without requiring direct network or IP access. Such system can be used in most organization that can access cloud storage services such as the Dropbox, Google Drive and Microsoft OneDrive. With the availability of different cloud storage services, it is easy to change the cloud storage server without having to change the IoT system which has been represented as a file system. The IoT system provider does not have access to the cloud storage server. The IoT system can incorporate file encryption to enforce privacy to prevent cloud storage providers from sniffing into the users' data.

2. BACKGROUND

IoT is a concept where objects transmit its data in digitize format over the internet so it can be used for its intended purposes. [1]. Right now, there are nine billion active IoT devices and it's predicted to increase to twenty four billion by 2020 [2]. As the internet becomes more accessible, applications of IoT systems have expanded in various areas such as health, home-security and industry which lead to the increase in the number of IoT devices [3-4]. IP-based is one of the connectivity in IoT device where several methods of defining IP address were applied such as dedicated IP stack and micro IP [5]. In the article on *an IP-Based Wireless Sensor Network Approach to The Internet of Things* (2010) stated the issues of IP-based connectivity which are IT device migrating to IPv6, integration of web application, mobility, global time synchronization and security [5]. IP-based access to IoT systems directly expose the system to internet network without proper security layer where internet-based attack can happen [5]. Furthermore, IP-based access require some technical expertise to configure IP port forwarding and port restriction.

As cloud computing emerges, it offers several beneficial services to address the technological restrictions of IoT systems such as processing power, storage and power consumption [6]. Cloud-based IoT systems also offer the ease of using the systems without requiring configuration of a local network for external access. Despite overcoming some of the previous known constraints, cloud-based IoT systems have their own issues such as privacy handled in the cloud, limitation of identities, unable to handle different types of network and identifying data priority [7]. Furthermore, maintaining server is much more costly than maintaining IoT devices [8]. This can hinders the progress of a developers as well as restricting their technological creativities. Different IoT cloud companies don't use a common integration system where there is no flexibility to interchange from one cloud to another [8]. This requires users to set up entirely new IoT system when they have to migrate to a different cloud service. In this research paper, public cloud-storage based IoT system were proposed to address most of the issues stated. It provides the ease of setting up the internet access without having to configure a local network for external access, and avoid the issues of cloud-based systems in terms of being tied to a specific cloud service provider or manufacturer. Cloud storage services were meant for storing and accessing data from a remote storage disk [9]. Even though it's not intended to be used for IoT systems, IoT devices can store their data in cloud storage. At the very least, one data server is enough for the cloud storage platform to operate their service where users can store, retrieve and manipulate their files [9]. The proposed approach in this paper will allow a developer to create IoT systems that can be used with different cloud services. Since data servers are provided by the cloud storage provider, an IoT creator does not need to maintain a server. In 2014, on average there were 300 million users on Dropbox, 250 million users on OneDrive, and 200 million users on Google Drive [10]. The cloud storage stated are popular services that offer user friendly UI and adequate spaces. The availability of several cloud storage allow an IoT system using the proposed approach in this paper to easily change from one cloud storage provider to another without reconfiguring the system.

3. CLOUD STORAGE IOT SYSTEM

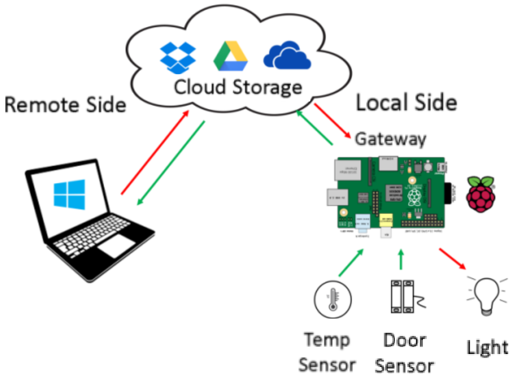

Figure 1. Overview and dataflow of cloud storage based IoT system

Figure 1 shows the overview of the proposed cloud storage based IoT system as well as how data are transferred within the system. It consists of two sides, local and remote sides At the local side, a local server or gateway manages and configures all of the IoT devices and communication with the cloud storage servers. Each IoT device is represented as a file in the gateway. Configuration Information and data for an IoT device are written in a file created by the gateway. The files for all devices are stored in the gateway and synced to cloud storage. Most cloud storage platforms provide Application Programming Interface (API) for the basic functions such as upload and download that can be used to achieve the data transaction of an IoT system. At the remote side, user can interact with an IoT device and, view or change the state of the device by using upload and download API functions respectively. The files' content can be kept in sync at both sides. The states of an IoT actuator device are changed by uploading changes to the device file from the remote side. Both remote and local sides can detect changes of the cloud storage by using the API of the selected cloud storage platform. The only requirement for users to use this IoT system is by having the cloud storage account and have internet access. To access the cloud API, authentication of the user account in the cloud server is required. There are four main API functions that are used to mimic the process of a mainstream IoT system. They are upload, download, polling and authentication**.** These basic functions have similar process amongst different cloud storage platform. Each file inside the cloud storage is given an ID where it is used by the API functions to access the file. Since the synchronization process is independent of the files being processed, the same IoT system can be used on different cloud storage servers. This also means the local IoT system is independent of the choice cloud storage service provider. The proposed IoT system uses fat-client model where the metadata of each device is stored in itself. This allows the creation of new IoT device to be added to the system without manually modifying the script and database at the gateway as long the metadata in the IoT device is in an agreed format. The IoT system can easily switch between different service providers. Furthermore, the local system can continue to function without the cloud. Changes to the IoT system does not affect the choice of the cloud storage. Likewise, changes to the cloud storage service does not affect the local IoT system.

By separating the data storage provider from the IoT system provider, this approach provides a layer of privacy protection as well as allow IoT system provider to develop IoT system without having to maintain a server and without having to tie to a specific IoT cloud. A second layer of privacy protection can be achieved by the IoT system provider with encryption of the device files to prevent the files being sniffed by the cloud provider. In the end, the device files are well secured because they are encrypted by both cloud storage and IoT system where both respective developers cannot decrypt the files independently.

4. EXPERIMENT

4.1 Setup

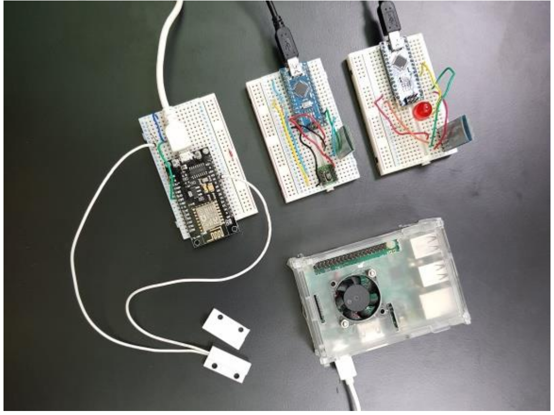

Figure 2. Implementation of IoT device

The purpose of the experiment was to find out whether the cloud storage was able to implement a real IoT system. An IoT system has been implemented using a Raspberry Pi as the gateway with three Arduino based IoT devices. In figure 2, The devices implemented are a temperature sensor, a door sensor and a light that represent an actuator. Local and remote side software have been developed to remotely interact with the devices over the internet. The implemented system has successfully operated as an IoT system using cloud storage. The remote side application could view the states of the sensors and could control the state of the actuator.

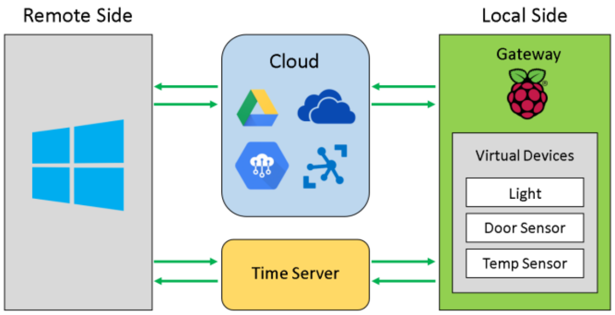

Figure 3. Experiment setup and data flow

Experiments were then conducted to evaluate the real-time performance of the proposed system against that of the dedicated IoT cloud platforms. The experiments conducted were to measure the response time and data received rate for the data transfers between the local and remote sides. Figure 3 shows the setup and the transmission flow of the data when the experiments were conducted. The cloud platforms used in these experiments were popular dedicated IoT cloud platforms and cloud storage platforms. The dedicated IoT cloud platforms consist of two types, top tier and free tier platforms. The top tier provides many features but payment is required. The top tiers tested in this work are Microsoft Azure IoT and Google IoT. The free tiers tested are CloudMQTT (CMQTT) and Eclipse IoT. The cloud storage used in this work are OneDrive, Google Drive and Dropbox. These three are the major cloud storage providers in the market. All of these platforms provide API to programmatically control their data transfer. The IoT system consists of two sides, the local side where IoT devices and gateway are connected and the remote side where monitoring and controlling IoT devices happened. The gateway is a Raspberry Pi3 and a PC is used at the remote side. To facilitate the measurements in the experiment, three virtual IoT devices were created to simulate temperature sensor, door sensor and light. Data from the temperature and door sensors were transmitted from the local to the remote side to emulate the actual IoT sensors transmitting data from where they are placed to a viewer at remote location. The light emulated an IoT actuator where it was controlled by receiving data from the application at remote side to local side where the light was placed. Therefore, in this experiment, remote and local side can be a sender or receiver depending on the IoT device. A time server was implemented inside the sender's machine (whether it's remote or local) so that the receiver can get a synchronized timestamp from the sender.

4.2 Experiment details

Measurements of response time and data receive received rate were conducted on IoT systems implemented with the following seven cloud platforms:

- Microsoft Azure IoT
- CMQTT
- Eclipse IoT
- Google IoT
- Dropbox
- Google Drive
- Microsoft OneDrive

Note that the later three are cloud storage and are used based on the proposed approach in this paper.

The experiments were conducted for seven days because the platform's server may have different response time each day based on traffic. On each platform, a device sent an average of 50 data with different intervals each day. Five different delay intervals were set in between data sent. Those intervals were 5, 3, 1, 0.5, 0.25 and 0 second. The flow of data sent was from local to remote side and vice-versa. Throughout the duration of the experiment, the environment was in the lab using the same machine and network.

An IoT system needs to respond to the changing state of the IoT devices and display the new state to the receiving side as fast as possible. In this experiment, the response time of each platform was measured by capturing the timestamp of the sender when it sent the data and the timestamp of the same time server when the data is received. Thus, the response time can be obtained by subtracting the timestamp value of receiver and sender that were fetched from the same time server. For the sensors, the sender would be at the local side whereas for the actuator, the sender would be at the remote side.

The successful rate of data transfer from sender to receiver is an important aspect of IoT system. Loss in data means loss in valuable information. In this experiment, the data received rates were measured by counting the number of received data over the total number of data sent.

5. RESULT AND DISCUSSION

5.1 Response time

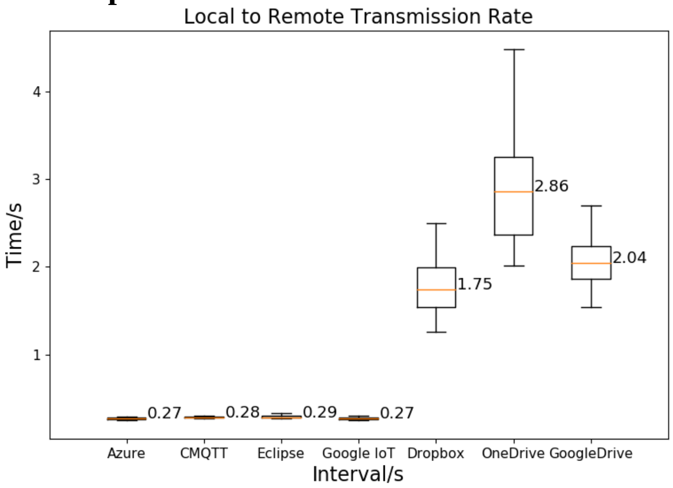

Figure 4. Box and whiskers plot of response time of local to remote side

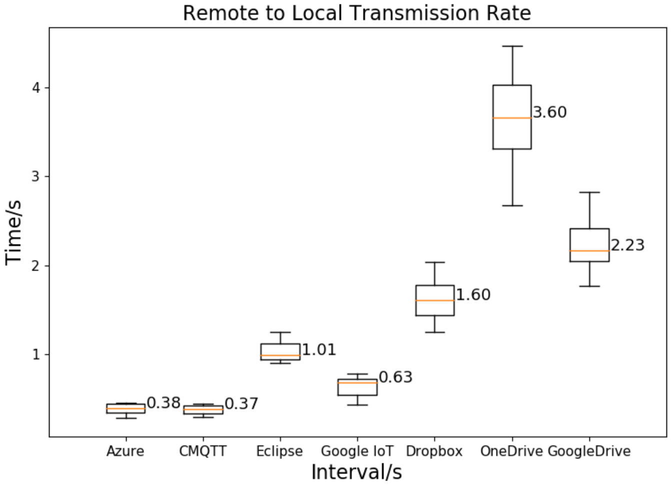

Figure 5. Box and whiskers plot of response time of remote to local side

The measurement of response time is represented in box and whiskers plot because it can layout the means as well as the deviation. The response time of the IoT system implemented on the seven cloud platforms are shown in Figure 4 and Figure 5. The value in Figure 4 indicate the average response time of 50 average measurements over seven days for the two sensors on each of the cloud platforms. Figure 5 shows the figures for data transfer from remote to local for the actuator. The overall higher response time in remote to local side transmission than that of local to remote transmission is suspected to due to the download mechanism was heavier than upload at the local side since downloading requires detecting changes at the server. Furthermore, when the downloading role (receiver) is on a machine with slower processing power, it affects the response time of the cloud. Raspberry Pi (1.4 GHz quad-core ARM Cortex-A53) has lower processing speed than a laptop (Intel Core i7-3770 3.40 GHz) at the remote side used in this experiment. Most dedicated IoT cloud platforms have more or less the same average response time value at around 0.28 second when data transfer take place from local to remote. Meanwhile the fastest average response time for cloud storage was Dropbox with average response time value at around 1.60 second and the worst average response time was OneDrive with its value clocked around 3.60 second. For data transfer from remote to local, the slowest average response time for IoT cloud was 1.01 seconds, and the fastest average response time for cloud storage based system was 1.60 seconds. Cloud storage response times were significantly slower than dedicated IoT cloud platform. This is expected as the cloud storage servers handle data transmission differently in comparison to dedicated IoT cloud server.

5.2 Data received rate

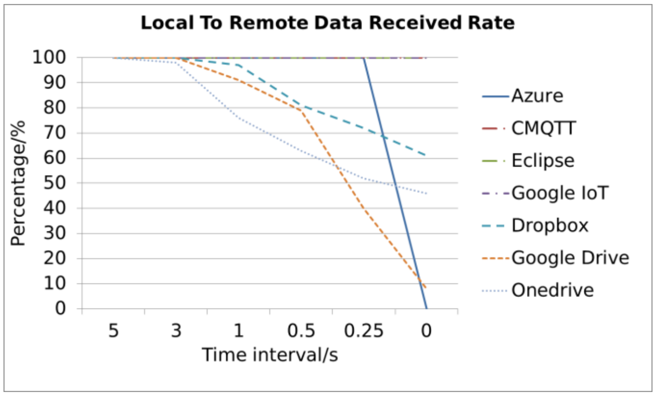

Figure 6. Data received rate of local to remote side

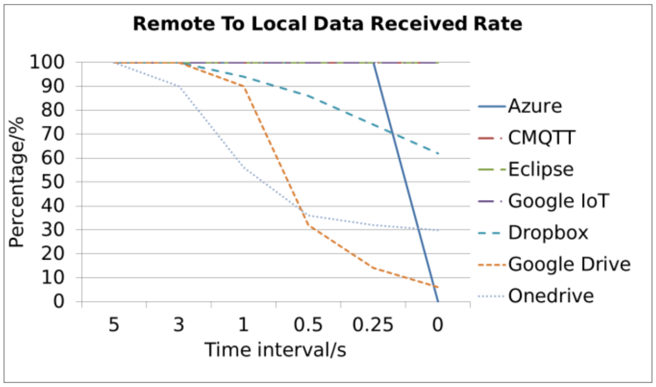

Figure 7. Data received rate of remote to local side

Table 1. Data received rate of remote to local side. For brevity, G-IoT denotes Google IoT and G-Drive denotes Google Drive in the table

Cloud	***5s***	***3s***	***1s***	***0.5s***	***0.25s***	***0s***
Azure	100%	100%	100%	100%	100%	0%
CMQTT	100%	100%	100%	100%	100%	100%
Eclipse	100%	100%	100%	100%	100%	100%
G-IoT	100%	100%	100%	100%	100%	100%
Dropbox	100%	100%	94%	86%	74%	62%
G-Drive	100%	100%	90%	32%	14%	6%
OneDrive	100%	90%	56%	36%	32%	30%

Table 2. Data received rate of local to remote side. For brevity, G-IoT denotes Google IoT and G-Drive denotes Google Drive in the table

Cloud	*5s*	*3s*	*1s*	*0.5s*	*0.25s*	*0s*
Azure	100%	100%	100%	100%	100%	0%
CMQTT	100%	100%	100%	100%	100%	100%
Eclipse	100%	100%	100%	100%	100%	100%
G-IoT	100%	100%	100%	100%	100%	100%
Dropbox	100%	100%	97%	81%	72%	61%
G-Drive	100%	100%	91%	79%	40%	8%
OneDrive	100%	98%	76%	63%	52%	46%

Figure 6, Figure 7, Table 1 and Table 2 show the data received rate of the different cloud platforms as the time interval between data transmission was decreased from 5 seconds to 0 second. All dedicated IoT cloud platforms could receive all the data even when the time interval was shorter than their average response time as measured in Figure 4 and Figure 5. This indicates that the IoT cloud servers did not overwrite data when the receiver had not read the data. The cloud server appeared to queue the data until the receiver acknowledged that they had received the data. In the case for Microsoft Azure IoT at the 0 second time interval, the receiver did not receive any data. This could be that the Azure server refuses to receive streaming of data from one source to prevent denial of service (DOS). As for the cloud storage, when the time interval was below their average response time as measured in Figure 4 and Figure 5, data loss occurred. The lower the time intervals from their average response value, the more data were loss. The cloud storage server did not “queue” the file change but instead it update with the latest change. Based on the response time experiment, data transmitted from remote to local side were having slightly slower response time compared to local to remote side. Hence, it also affect data received rate of cloud storage platform on data transmitted from remote to local side.

6. CONCLUSION

This paper has proposed an implementation of IoT system using cloud storage in which files are used to represent devices. The proposed system was implemented and successfully operate as an IoT system. However, experimental measurements have shown that the response time to the changes of the states of the devices using cloud storage was significantly slower than dedicated IoT cloud platforms. The results show that dedicated IoT cloud platforms have short response time making them capable to handling time critical system. On the other hand, the cloud storage based IoT system would have high data loss when data transmission is faster than its capability to respond. The proposed system could be used in non-time critical IoT systems. In many cases, such limitation is tolerable. For example, in a home IoT system, a delay of a few seconds for state update is acceptable. With the advantages that the proposed system offers, it opens up opportunities for under resource technopreneurs to create IoT systems with high degree of privacy protection.

7. REFERENCES

[1] Rolf H. Weber and Romana Weber. 2014. *Internet of Things Legal Perspectives*, Berlin: Springer Berlin 9

[2] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems* 29, 7 (2013), 1646. DOI:http://dx.doi.org/10.1016/j.future.2013.01.010

[3] Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. The Internet of Things: A survey. *Computer Networks*54, 15 (2010), 2787. DOI:http://dx.doi.org/10.1016/j.comnet.2010.05.010

[4] Xu Li, Rongxing Lu, Xiaohui Liang, Xuemin Shen, Jiming Chen, and Xiaodong Lin. 2011. Smart community: an internet of things application. *IEEE Communications Magazine* 49, 11 (2011), 68. DOI:http://dx.doi.org/10.1109/mcom.2011.6069711

[5] Sungmin Hong et al. 2010. SNAIL: an IP-based wireless sensor network approach to the internet of things. *IEEE Wireless Communications* 17, 6 (2010), 34–36. DOI:http://dx.doi.org/10.1109/mwc.2010.5675776

[6] Alessio Botta, Walter De Donato, Valerio Persico, and Antonio Pescape. 2014. On the Integration of Cloud Computing and Internet of Things. *2014 International Conference on Future Internet of Things and Cloud* (2014), 23–24. DOI:http://dx.doi.org/10.1109/ficloud.2014.14

[7] Mohammad Aazam, Imran Khan, Aymen Abdullah Alsaffar, and Eui-Nam Huh. 2014. Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved. *Proceedings of 2014 11th International Bhurban Conference on Applied Sciences & Technology (IBCAST) Islamabad, Pakistan, 14th - 18th January, 2014*(2014), 416. DOI:http://dx.doi.org/10.1109/ibcast.2014.6778179

[8] Alessio Botta, Walter De Donato, Valerio Persico, and Antonio Pescapé. 2016. Integration of Cloud computing and Internet of Things: A survey. *Future Generation Computer Systems* 56 (2016), 34. DOI:http://dx.doi.org/10.1016/j.future.2015.09.021

[9] Jiyi Wu, Lingdi Ping, Xiaoping Ge, Ya Wang, and Jianqing Fu. 2010. Cloud Storage as the Infrastructure of Cloud Computing. *2010 International Conference on Intelligent Computing and Cognitive Informatics*(2010), 382. DOI:http://dx.doi.org/10.1109/icicci.2010.119

[10] Tony Danova. 2014. Most People Are Still Confused About Cloud Storage, And No One Service Is Winning The Race To Educate And Acquire Users. (July 2014). Retrieved August 20, 2018 from https://www.businessinsider.com.au/people-use-the-cloud-and-dont-even-realize-it-2014-7.