EXPERIMENTAL STUDY TO MAINTAIN STABILITY IN THE CASE OF INDUSTRIAL CLOUDLETS NETWORK

Amel Ben Lazreg

Higher Institute of Computer Science and Communication Technologies of Hammam Sousse, Prince Laboratory, Sousse University, Sousse, Tunisia

ABSTRACT

Cloud usage has increased rapidly, except when it comes to the notable integration of IOT with Industry 4.0.This study seeks to challenge the limits of Cloud uses. As a result, the integration of Cloudlet (middleware) in some critical applications is growing rapidly with the expansion of Cloud applications. The Cloudlet is being deployed as a necessary component in the industrial network to ensure data synchronization, security, and stability. Based on our recent studies, this work will focus on how to maintain stability in a Cloudlets network. In other words, the goal is to ensure that all nodes remain stable, meaning that all the Cloudlets are connected to the Cloud and able to transmit data independently, regardless of the constraints imposed by the industrial environment. The first part of this work is to define the problem, discuss how Cloudlet can resolve some Cloud limits. Then, we define the stability and its models. In effect, the experiments in a real industrial environment are presented. Finally, the results are analysed.

KEYWORDS

Cloudlets Network, Cloud Computing, Stability, Industrial Applications, Websockets.

1. INTRODUCTION

Nowadays, every user aims to benefit from the advantages of Cloud Computing while ensuring the data's security and confidentiality [1] . However, the limitations of using the Cloud in certain critical domains, especially in the industrial sector, continues to endure. These limits figure in Service discontinuity, latency, security, data localization and availability, etc. [2]

In fact, optimising public Cloud resources necessarily requires the implementation of middleware to minimise direct contact between the user and the public Cloud. These include:

- Making data available even when disconnected from the Cloud, given that this middleware is equipped with several types of connection: Wi-Fi, 3G/4G, ETHERNET, GPRS/GSM, etc. [3]
- Replicate updates requested by the user in real time, if the Cloud is available, otherwise the most recent version of the data is retrieved. This means that the information available to the user is the most accurate (the real information in the network).
- Maintain permanent connectivity between the user and his Cloud via a synchronous real-time copy of the services and data allocated. [4]
- Distribution of the computing load across the various network components (Cloud and other intelligent equipment). [5]

DOI: 10.5121/ijccsa.2023.13401

- The transfer of data and any other action carried out within the Cloudlet network requires security and traceability. We also need to plan updates and maintenance over time. To do this, we used the Blockchain technology. [6]

In fact, we have envisaged a hybrid solution enabling mechanisms to be implemented to ensure the autonomous transfer of data from the user layer to the Cloud server. To realise this goal, the industrial Cloudlet is integrated into various real applications. The installation of the Cloudlet network has solved several problems, but the behaviour of the nodes within this network still needs to be studied to guarantee better performance, which is the subject of this article.Therefore, our main aim is to guarantee the stability of this cloudlet network, even if there are moments of disconnection.

2. BACKGROUND

2.1. Embedded Cloudlet

An Industrial Cloudlet is considered a middleware layer between users and the Cloud. It provides a unified abstraction for all requests in a distributed environment and addresses various aspects of communication, synchronization, replication, persistence, and access control. [7]

More specifically, we will discuss an embedded Cloudlet, which is composed of both hardware and software. A Cloudlet is an embedded system that functions as a private cloud or embedded cloud, providing an individual user with access to the most up-to-date versions of data stored in the cloud. It also serves as a local backup solution for businesses, ensuring data transfer and overcoming the lack of internet coverage.

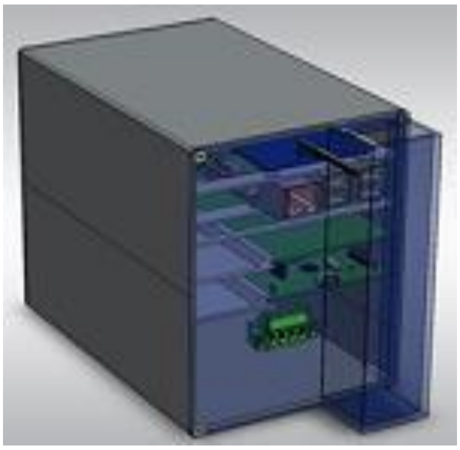

Figure 1. Embedded Cloudlet

2.2. Industrial Cloudlets Network

An Industrial Cloudlets Network (ICN) is a collection of dependable, high-capacity computers that are securely connected to the Internet. These nodes can be accessed by nearby mobile devices, which function as end users. Cloudlets do not need to be a fixed infrastructure close to the wireless access point; instead, they can be dynamically formed with any device in the local network that has available resources.

3. STABILITY OF A CLOUDLET NETWORK

The stability of a Cloudlet network necessarily involves verifying the stability of each individual Cloudlet over a specific period of time [8]. To ensure the stability of a Cloudlet, it is essential to have a consistent power supply, uninterrupted connectivity, and reliable transmission of accurate data to the Cloud server.

According to the approach proposed by previous studies, which assumes that the acquisition of any data should be considered essential, and whenever new data appears, it should be treated as such. Consequently, we can deduce that the Cloudlet network will only be perfectly stable when all nodes are connected to the Cloud and are able to transmit data independently, without being affected by the constraints of the industrial environment.

Each Cloudlet has its own characteristics, including connection type, RAM, processing power, and more. This allows the system to be subdivided into subsystems based on the type of connection. These sub-networks introduce the possibility of acquiring data from various connections and at various times.

By the deadline, during the "waiting phase," the information will be stored within the internal system of each Cloudlet, which can result in a state of global system instability.
Let “s” be the stability coefficient of the Cloudlet network.

To determine whether a system is stable or not, we define three states:

Table 1. Types of system stability.

parameter	System
$s = 1$	Perfectly stable
$0.5 < s < 1$	Moderately stable
$s \leq 0.5$	Unstable system

To effectively study network stability, we have utilized the concept of entropy, which is a state function. It can be interpreted as a measure of the degree of disorder in a system at the microscopic level. The second principle states that any transformation of a thermodynamic system occurs with an increase in the overall entropy, which includes the entropy of both the system and the external environment. This implies the creation of entropy and is associated with the impossibility of transitioning from disorder to order without external intervention.

According to Szilard [9], the entropy law allows for the quantification of information, particularly the optimal amount of information required to make a decision about a given variable. This new definition has attracted the attention of numerous researchers in various fields, particularly those interested in quantifying a decision parameter related to the flow of information.

The entropy law of an information item is considered as a function of the probability of a given random variable. It satisfies all of the following properties:

- The value of information from two events is greater than the value of information from one event.
- If two events are independent, the value of the information resulting from these two events is equal to the sum of their values.
- The value of information from any event is non-negative.

The mathematical functions that can satisfy all these properties are of the form:

$$H(X) = -\sum_{x} p(x) \log_2 p(x)$$

Where X is a discrete random variable and $x \in X$ and $p(x) = \Pr\{X = x\}$.

This equation will be written in our case as follows:

$$H(Rcl) = -\sum_{k=0}^{n} p(SClk) \log p(SClk)$$

Where n is the number of Cloudlets in the network

And $p(SClk)$ is the stability coefficient of each sub-network (per connection type) $= \frac{m}{n}$

Where m is the number of Cloudlets for that type.

Therefore, determining the entropy of each sub-network will allow measuring the “stability” of the overall network. In other words, entropy will enable a Cloudlet network to exhibit the behaviour of a node based on the transmitted data to and from the Cloud. In this case, the parameter “disorder” is inversely proportional to the entropy value. That is, the network is unstable when its entropy reaches its minimum.

4. APPLICATIONS

In order to showcase the various approaches already discussed in the preceding sections of this paper, we conducted experiments using our contributions in different industrial settings, including interurban and sub-Saharan industrial zones. In each case, we had to integrate the Cloudlet component into a platform that is already based on a Cloud. We would like to draw the reader's attention to the fact that our experiments were conducted within a highly professional framework. These experiments were based on international invitations to tender from Tunisia, Italy, Belgium, and Luxembourg, and were accompanied by rigorous specifications.

We took responsibility for setting up our platform to ensure a backbone of Cloudlets that are closely integrated with the local environment. These interconnected industries, such as phosphate production, industrial laundries, and online ordering, needed to continuously synchronize their data with the cloud server. In fact, these interconnected industries had to operate in real time, considering that the actors within the system utilize and manage the data during production. As a result, these actors have to operate even when there is no internet connectivity.

To meet the specifications, we developed our industrial Cloudlet as a hardware product and evaluated its efficacy in two major industries in Tunisia and Italy. In the remainder of this paper, we present the results of several experiments.

The “Lavapiù” laboratories in Italy are leaders in the field of industrial laundry services. They are best known for dry cleaning, especially delicate garments like silk, wool, lace, and cashmere. However, the equipment needed for this type of cleaning is very complex, including large washing machines and a high volume of customers. In fact, this group of companies utilizes the equipment and automated machines provided by the renowned “Metalprojetti”.

"Metalprogetti" offers all the equipment necessary to fully automate laundry laboratories.

Previously, "Lavapiù" organized all these tasks, but the process of fulfilling customer requests was traditional. When the customer arrived, a worker went to collect the clothes that were already prepared. In these situations, "Lavapiù" has encountered certain problems. Sometimes, customers arrive and their clothes are not yet ready or, due to the fault of the sales assistant, customers end up taking other clothes. This not only wastes their time but also leads to unnecessary searching for the correct items.

4.1. Implementation

To address these constraints, we proposed integrating the Cloudlet into the computer that contains the laboratory management software (cash register, garment list, pricing, etc.) while associating each customer with a unique barcode. This barcode is placed either on the customer's loyalty card or in their mobile application. In this case, the Cloudlet operates on a sampling period. Instead, actions are on-demand, meaning that when the garments are ready; the automated system sends a notification to the Cloudlet. This information is then broadcasted to the Cloud.

Consequently, the customer is notified that they can collect their garments. Upon arriving at the laboratory, the worker scans their barcode using a reader connected to the Cloudlet station. Subsequently, the Cloudlet records the transaction and notifies the automated system that it requests the garments for a specific customer. This notification triggers the operation of the automatic door, which brings the requested garments to the user.

4.2. Contribution

In the communication process explained above, we have elaborated the communication between the Cloudlet and the Cloud at the beginning in this way.

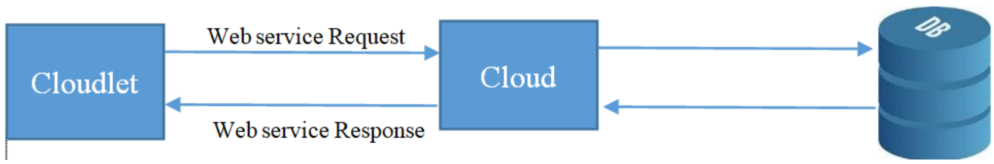

Figure 2. Web service communication between Cloudlet and Cloud

At the beginning, we implemented a system that follows the communication depicted in Figure 2, where the Cloudlet sends a membership request to the Cloud. The Cloud then saves a session (Cloudlet ID, port, etc.) in the database. It subsequently checks each time whether the Cloudlet is connected or not based on its corresponding data. In such cases, we found that the Cloudlet disconnects multiple times without the Cloud server being aware, as it relies on outdated session information.

To address these issues, we transformed the process into a self-verification mechanism using web sockets, as illustrated in Figure 3.

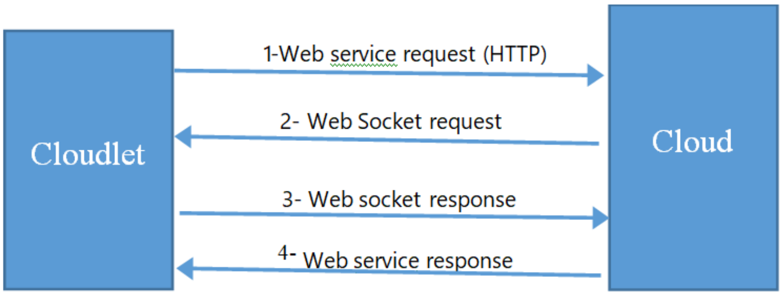

Figure 3. Web service communication between Cloudlet and Cloud

Web Sockets are a protocol that provides full-duplex communication channels over a single TCP connection. In contrast, HTTP provides half-duplex communication. The information exchange mode of Web Sockets is bidirectional, meaning that the server can push information to the client, which is not possible with direct HTTP communication. In our case, we utilized Web Sockets to push realtime update messages to a Cloudlet. One of the most interesting features is that most WebSocket libraries also support direct response to WebSocket messages from the Cloudlet, acting as acknowledgments in the message queue.

With this verification mechanism in place, we experienced fewer disconnections and improved connection stability. This will be further demonstrated through experiments in the following section.

4.3. Experimentations and Results

We installed Cloudlets in several industrial laundry and dry cleaning agencies, each with an appropriate type of connection (4G, Wi-Fi, Ethernet). During the installation of the Cloudlets using only web services (HTTP), we observed the following results. We implemented the upward communication (from the Cloudlet to the Cloud) in cases where the Cloudlet was connected via Wi-Fi and 4G.

As discussed in the connection request model, we relied on two approaches. The first approach was the standard one, where the Cloud contains a process which checked whether the Cloudlet was connected or not. The second approach involved self-verification by the Cloudlet itself. In this case, the Cloudlet checked its status (Connected/Disconnected) in relation to the Cloud at each sampling period (every 6 seconds).

We calculated the number of disconnections during a selected test period and computed the probability every 5 minutes.

$$P(CLcnx) = \frac{ncx}{nT}$$

Where ncx is the number of successful connections to the cloud server.

And nT is the number of total connection attempts. As the industrial Cloudlet is an embedded system with limited memory resources, so we thought of recording in the Cloudlet log file to save the disconnection attempts.

Hence

$$P\ (Cl_{cnx}) = 1 - \frac{ndx}{nT}$$

Where ndx is the number of disconnection times (failed connection attempt with the Cloud server).

These are some results of Log files of installed Cloudlet in the network that are considered as entries of mentioned results.

```
2023-07-06 21:25:56 =002%04002
2023-07-06 21:25:56 =002%0;succes
2023-07-06 21:26:02 =002%0#002
2023-07-06 21:26:02 =002%0;succes
2023-07-06 21:26:37 Reconnection
2023-07-06 21:26:44 =002%04002
2023-07-06 21:26:44 =002%0;succes
2023-07-06 21:26:50 =002%0#002
```

Figure 4. Log File in Cloudlet

When Cloudlet sends data to the Cloud server, it records the status of its connection in a log file. As the connection parameter is fundamental to the study of stability, the number of disconnection times is considered an important factor in the stability of Cloudlet.

```
2023-06-24 08:26:31 Cloudlet connected = 204
2023-06-24 08:26:31 Cloudlet connected = 401
2023-06-24 08:26:31 Cloudlet connected = 304
2023-06-24 08:26:31 Cloudlet connected = 207
2023-06-24 08:26:31 Cloudlet try to connect false
2023-06-24 08:26:31 Cloudlet connected = 302
2023-06-24 08:26:31 Cloudlet connected = 206
2023-06-24 08:26:32 Cloudlet connected = 3051
2023-06-24 08:26:33 Cloudlet5001 session Connection Closed......false
2023-06-24 08:26:33 Cloudlet201 session Connection Closed......false
2023-06-24 08:26:37 Cloudlet002 session Connection Closed......false
2023-06-24 08:26:41 message sended to Cloudlet 307 TextMessage payload= [=307%0#3071, byteCount=10, last=true
2023-06-24 08:26:41 message sended to Cloudlet 303 TextMessage payload=[=303%0#303], byteCount=10, last=true
2023-06-24 08:26:41 message sended to Cloudlet 104 TextMessage payload= [=104%0#104], byteCount=10, last=true
2023-06-24 08:26:41 message sended to Cloudlet 308 TextMessage payload= [=308%0#308], byteCount=10, last=true
2023-06-24 08:26:41 message sended to Cloudlet 305 TextMessage payload= [=305%0#305], byteCount=10, last=true
```

Figure 5. Log File in Cloudlet

In a Cloudlet network, the Cloud server saves all the states of the associated Cloudlets to give an overall view of the stability of its nodes.

Below are the results of the probabilities for Cloudlets connected in 4 G and Wi-Fi.

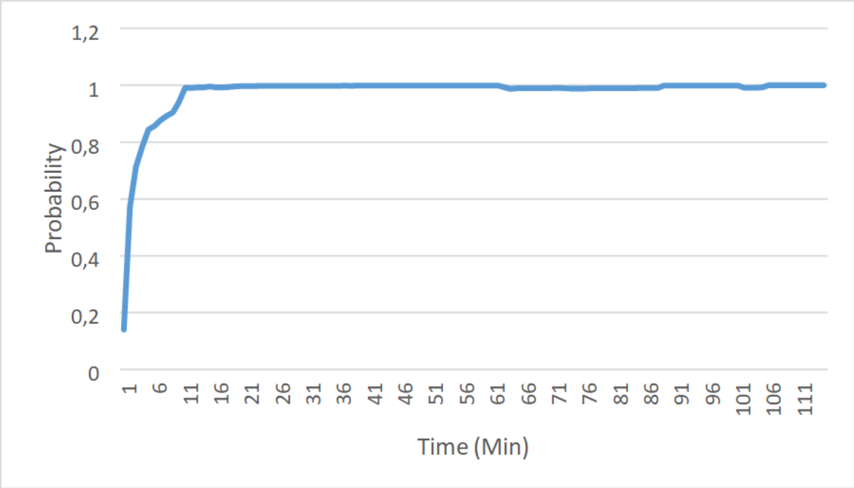

Figure 6. Probability of connection via 4G (Without verification)

In the case of 4G communication in a network consisting of 5 Cloudlets without self-verification by the Cloudlet, we observed that after an average time of 13 minutes, the probability of the network being connected converges to 1.

For this type of communication, the maximum upstream throughput varies between 21.69 and 28.18 Mbit/s, and the downstream throughput ranges from 48.72 to 87.09 Mbit/s. The time from initialization to network stability can be considered as the system's warm-up time or may be due to real-world conditions.

As mentioned before, we did not work with a simulator to achieve perfect conditions from the beginning of the experiment. The same stabilization duration is sufficient for the 4G system with Cloudlet self-verification.

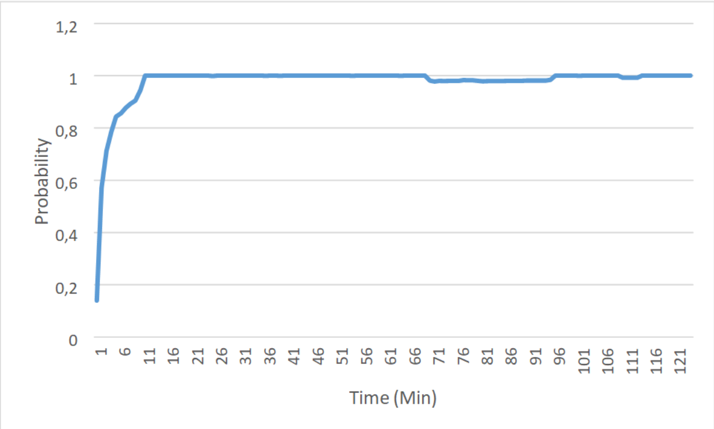

Figure 7. Probability of connection via 4G (With verification)

To see the degree of connection optimisation with the self-checking module in the Cloudlet network, we calculate the optimisation coefficient

$$\alpha = (1 - \frac{\sum_{i=0}^{N} P\,(Cl_{cnx})_{i\,without-verif}}{\sum_{i=0}^{N} P\,(Cl_{cnx})_{i\,with-verif}})$$

In the case of the 4G Cloudlet network, we got α = 1- 0.926= 0.074. We conclude that the cloudlet network with cloudlet verification optimized the probability of connection to the cloud server by 7.4

By testing the same experiments with the Cloudlet network via Wi-Fi, we got these probability results.

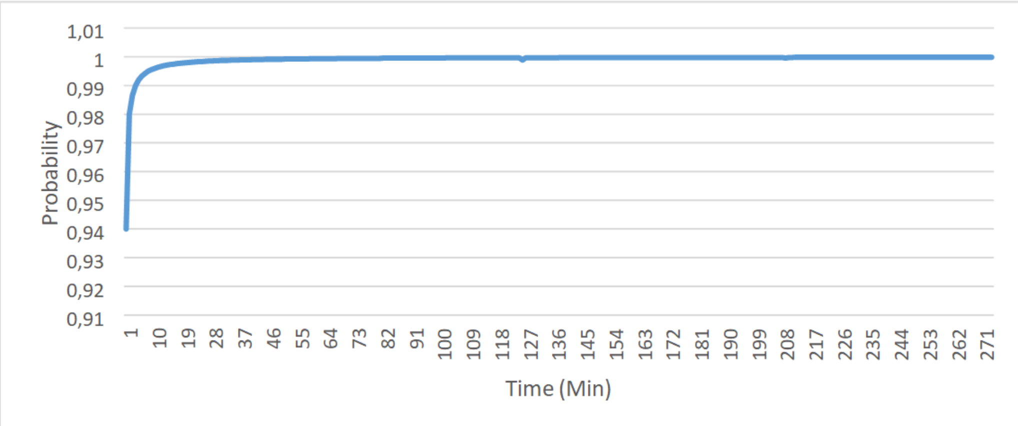

Figure 8. Probability of connection via Wi-Fi (Without verification)

The Wi-Fi network without verification takes an initialization period of almost 8 minutes.

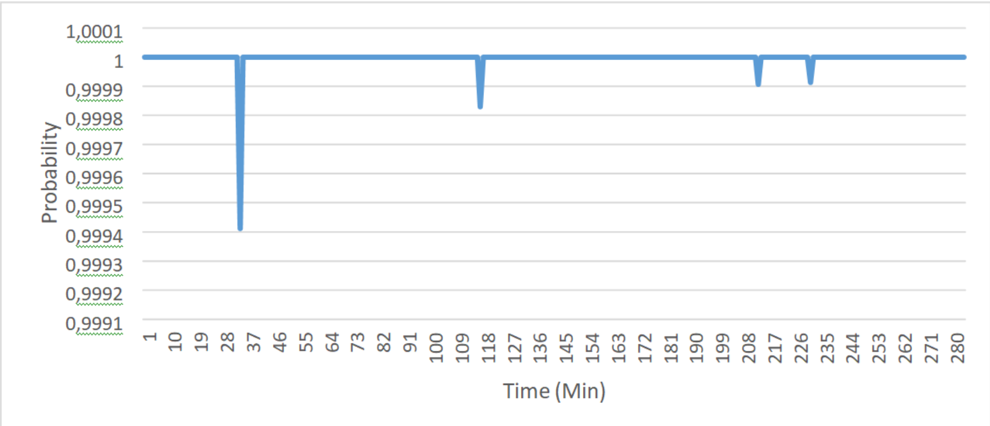

Figure 9. Probability of connection via Wi-Fi (With verification)

In the case of self-verification of a Wi-Fi connection where the throughput varies between 1.6 and 3.97 Mbit/s (low throughput, but these are the actual conditions of this laboratory in ViaLarga), we obtained α = 1 - 0.9673 = 0.032.

Given the linearity of the system, each state depends on the previous state and influences the following state. This is referred to as a memory system. Hence, the calculation of cumulative entropy imposes that over time, the system learns from connection states, which gradually increases the probability of connection. We highlighted the progression of entropies through the accumulation of k in this equation

$$\boldsymbol{H}(RCl) = -\sum_{k=0}^{n} p(SClk)\log p(SClk)$$

In fact, as if we have divided the large system into subsystems of the network. Below are the cumulative entropy curves for the same networks studied before.

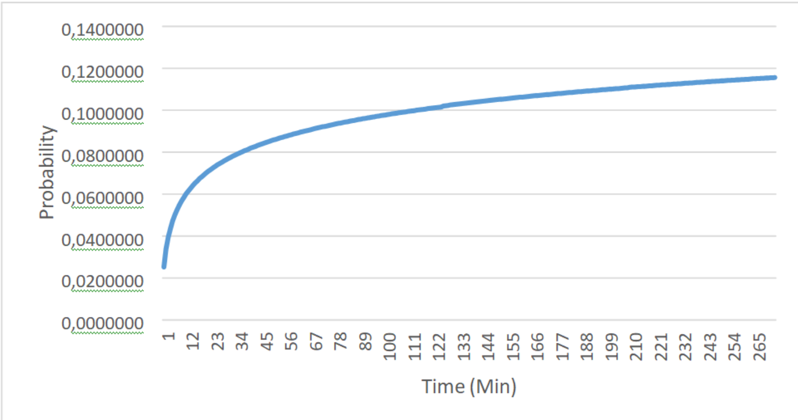

Figure 10. Entropy of connection via Wi-Fi (Without verification)

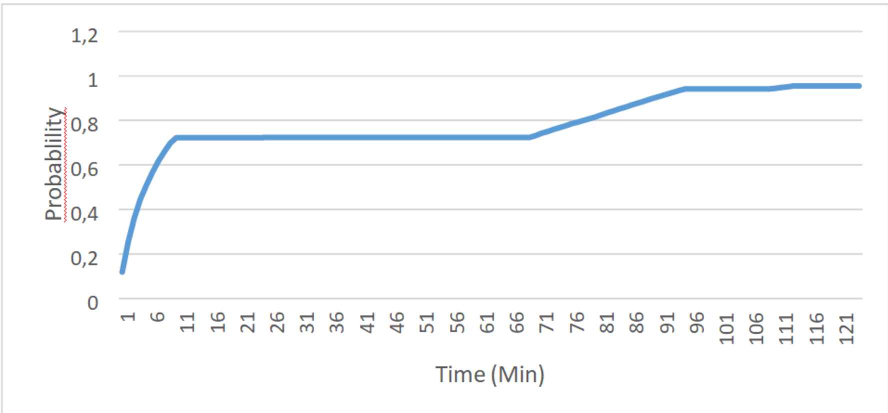

Figure 11. Entropy of connection via 4G (With verification)

For example, in the case of cumulative linear entropy of the 4G network with self-verification, which is the most suitable combination for data transfer in industrial environments, we observe that the system is unstable for the first 7 minutes after the process launch. Then it becomes moderately stable for about an hour, and finally, it maintains a high level of stability, reaching an H stability value of 0.94

5. CONCLUSIONS

In conclusion, these experiments have demonstrated the necessity of the Cloudlet component for data synchronization and optimization of data transmission. Integrating a Cloudlet layer has provided more opportunities to share the workload of the Cloud server and allow users to have their data readily available. This is a crucial parameter, as in the event of a failure or malfunction, the production process can continue without disruption (the Cloudlets replace the Cloud). In other cases, the Cloudlets serve as communication relays between themselves and between the end user and the Cloud. Basing on the stability of Industrial Cloudlet Network, we recommend to accord the 4G communication to the Cloudlet then Wi-Fi tom maintain more stability. The cost effectiveness of using Cloudlets is impeccable, as the aforementioned experiments have shown that equipping a network with a Cloudlet can significantly improve execution time and system responsiveness as a whole.

ACKNOWLEDGEMENTS

The author would like to thank EDUCANET Tunisian Company and La Milù for their support and for the opportunity; they gave me to carry out these experiments.

REFERENCES

[1] Kenji E. Kushida, Jonathan Murray,John Zysman, (2011) "Diffusing the Cloud: Cloud Computing and Implications for Public Policy", Journal of Industry, Competition and Trade, Vol. 11, pp 209–237

[2] Arooj Hassan, Sabeen Hussain Bhatti, Sobia Shujaat, Yujong Hwang (2022) "To adopt or not to adopt? The determinants of cloud computing adoption in information technology sector", Decision Analytics Journal, Vol.5

[3] Amel Ben Lazreg, Anis Ben Arbia, Habib Youssef, "A Synchronized Offline Cloudlet Architecture," ICEMIS, The International Conference on Engineering& MIS 2017. Monastir May 2017.

[4] Amel Ben Lazreg, Dhafer Ben Arbia, Anis Ben Arbia, Habib Youssef, " A Novel Cloudlet-based

[5] Communication Framework for Resource-Constrained Devices," DATA'18,International Conference on Data Science, E-learning and Information Systems 2018. Madrid Octobre 2018.

[6] Amel Ben Lazreg, Anis Ben Arbia, Habib Youssef, "A Distributed Decision Making Model For Cloudlets Network: A Fog To Cloud Computing Approach," AINA, The 34th International Conference on Advanced Information Networking and Applications 2020. Springer Book "Advanced Information Networking and Applications" pp.1126-1137

[7] Amel Ben Lazreg, Anis Ben Arbia, Habib Youssef, "Cloudlet-Cloud Network Communication Based on Blockchain Technology," ICOIN 2020, The 34th International Conference on Information Networking 2020. Barcelone Janvier 2020.

[8] E. Koukoumidis, D. Lymberopoulos, K. Strauss, J. Liu, and D. Burger, (2012) "Pocket Cloudlets," ACM SIGPLAN Notices, vol. 47, no. 4, pp. 171– 184

[9] Jannik Hahn, Olaf Stursberg, (2022) On Stability of Network Control Systems with Switching Transmission Probabilities, IFAC-PapersOnLine, Volume 55, Issue 13, Pages 276-281

[10] Parrondo, J.M.R. The Szilard engine revisited: Entropy, macroscopic randomness, and symmetry breaking phase transitions. Chaos 2001, 11, 725–733.

AUTHOR

Doctor in Sciences Computer, IT engineer, Completed her Ph.D from the Higher Institute of Computer Science and Communication Technologies of Hammam Sousse, University of Sousse , Tunisia. Her research interests include the distributed systems, Cloud Computing and Blockchain. Actually, she is university teacher in the same organization