Journal for Re Attach Therapy and Developmental Diversities
eISSN: 2589-7799
2024; 7 (1): 26-40

Secure MFCC Architecture For Health Care Application

Ferdaous kamoun-abid[1*], Amel Meddeb-Makhoulf[2], Faouzi Zarai[3]

[1*,2,3]NTS'COM research unit Sfax, ENET'COM Sfax, Tunisia

***Corresponding Author:** Ferdaous kamoun-abid
*NTS'COM research unit Sfax, ENET'COM Sfax, Tunisia

Abstract

Introduction: The current field of which our day is cloud computing. It is used in several fields like medical field. Moreover, for several reasons, such as diversity and the rapid increase in the number of connected devices, Cloud Computing is unable to meet certain requirements such as support for mobility, a high level of scalability, low latency and real time. This creates many challenges for the traditional architecture of Cloud Computing. to meet its requirements, several paradigms have appeared in recent years, such as mobile edge computing, mobile cloud computing and fog computing. Based on our research, fog computing is complementary to the cloud and uses network devices to process the latency of data collected using end users. In addition, MCC (Mobile Cloud Computing) devices offer many advantages such as streaming services to Fog Nodes. Due to the open features and high scalability of these networks, security is not guaranteed, where most of the existing research focuses on protecting systems and their platforms against attacks from unauthenticated devices only on a peripheral paradigm. To answer these questions and secure the IT architecture, which combines the advantages of three emerging technologies: Cloud computing, Fog Computing and Mobile Cloud Computing.

Objectives: In this article, we provide a method called MFCC (Mobile Fog Cloud Computing) that is used to distribute and collaborate firewalls to prevent network-based attacks for healthcare application.

Methods: Different levels of collaboration, based on a model for assessing confidence in relation to risk, are introduced. In this article we have based on the access control based on the trust which is a module aims to determine if an event is suspicious. Moreover, the confidence estimation is useful for making decisions to avoid the malicious packet. In addition we have used the level cooperation method.

Results: This evaluation framework used the NeSSi² tool, where the results show that the proposed architecture is better in terms of transmission delay and blocking rate compared to related works. The most important result is that our proposal is able to prevent distributed attacks, such as DDoS.

Conclusions: This work is based on the security of an architecture combining the MCC, and the fog title MFCC dedicated for health applications. It is based on network-level distributed access control based on distributed firewalls/controllers that manage ACLs and blacklists.

Keywords: Cloud, Fog, Mobile, Cloud Computing, distribution, collaboration, firewall, collaboration levels, healthcare, trust, risk, model

1. Introduction

In recent years, the use of mobile computing has increased because of its ability to deliver high accurate services from any ware in the world. The MCC (Mobile Cloud Computing) platforms are growing because of their low cost. The MCC is nearly all use cloud-based architecture [1]. It enables to share storage, data and hardware. Accurately, User Equipment (UE) via a Wide Area Network (WAN) delivers its application to the cloud. UEs not only use the Virtual Machine (VMs) existing in the Cloud but also delivers commands to VMs passing through WANs [2].
Using Cloud Computing to deliver medical applications and provide healthcare services to the mobile devices is a paradigm of MCC, because of the importance of medical data that are related to human lives. This is presented by Alonso-Monsalve et al. in [3] to provide volunteer devices, which are a part of the resources of the Cloud. In this technology, the client runs the required applications in the Cloud infrastructure by using fewer resources in the user's mobile devices.
There are many problems unsolved that are related to the cloud. We find geo-distribution, mobility support and low latency. For that Fog Computing is proposed to enable computing directly at the edge of the network. This is used to deliver services for multiple devices [4]. Luan et al. in [5] provide a view of Fog computing and describe the basic system architecture of Fog computing. The authors aim to shape the key features of Fog computing and identify its main design

goals. They are interested in open-ended research issues around an effective mobile network system.
Although, cloud computing provides more distributed resources in the core networks not in the edge networks, fog computing provides more distributed resources in the edge networks. In fact, Fog Computing is a new paradigm that extends the computing infrastructure from the center to the edge of the network. In addition, it is an extension of the Cloud to provide services of storage, computing and networking at the edge of the network. Moreover, Fog Computing is a much-virtualized platform. It provides networking services, compute and storage between end devices and traditional Cloud Computing. It is extended to an edge network which includes many distributed endpoint and Fog Nodes (FN) to provide localization, which allows low latency and context awareness. For the MCC, Fog Computing is used as an intermediate level between mobile level users and cloud computing, in order to ensure low-latency high-rate services to mobile patients [6].
Researchers focus on how adopted cloud computing orchestration frameworks are customized to fog computing systems like in [7]. Fog Computing has to integrate mobility and scalability to be included in the MCC. Nonetheless, the mobility of MCC devices give such benefits as it increases the topological to the serving Fog Node. For that, we are interested to modify the architecture of MCC by adding fog computing.

Despite their features of scalability, low costs and mobility, these technologies suffer from lack of security due to the emergence of attacks. They face serious security challenges due to the lack of the network access control. These challenges become more important for medical services, because of the sensitivity of the provided health-care data and medical-related applications, where privacy and security are crucial for both patients and medical staff.
There exist several works working on the three-level architecture, such as [8, 9]. Researchers in [8] focus on presenting a secure data sharing with the CFDS "cloud-fog-device data sharing system" based on a new method of the cryptographic called "MABE: matching attribute-based encryption". This work offers a secure fine-grained sender and receiver access control between cloud end fog nodes. This work lacks an access control process between the fog node and cloud, where an attacker can target the cloud by using a fake access to the fog node.
The work of [9] proposed a design of a new security protocol by establishing secure communication between the three-tiers. Used methods are the System Login, the Key Agreement (user-fog) and the Communication establishment (Fog-Cloud).
We also find Zagrouba and Alhajri in [14] use the machine learning techniques (ML Machine learning) for detecting IoT botnet attacks. However, the problem is that they don't have work on how they can prevent these attacks.
These works present new solutions to secure communications, but, most of the users prefer to use their mobiles to access to the network, where these works lack the existence of Mobile level and the control between this level and the rest of the architecture.

Related work:
This section summarizes works related to our fields of interest. We have identified several works focusing separately on MCC and fog computing. In the field of access control, researches highlight only the user-level access control. The network access control for MCC and fog technologies has not been developed by researchers because of virtualization, where only distributed virtual firewalls are introduced. We have identified a lack of cooperation, synchronization and optimization between these firewalls.
Singh Sohal et al. in [12] proposed a cybersecurity framework. It aims to identify in fog computing environment the malicious edge devices, based on the Markov model, Intrusion Detection System (IDS) and Virtual Honeypot Device (VHD).
Researcher are shown that cloud computing is unable to meet some requirement like latency. Roman et al. in [13] presented a survey about Fog computing. They present the security threats and mechanisms in all edge paradigms, as well as potential synergies and venues of collaboration.

Security problems are difficult to solve because of dynamic creating, complex trust situation and dynamic of topology. Wherefore, we find Han et al. [14] proposed a new method to enable the client to eliminate connecting rogue Access Points (AP). This method is based on the round-trip time between DNS server and end users to detect malicious AP.
The main purpose of traditional firewalls is to analyse the traffic received by examining them and preventing them from continuing if a predefined rule is not respected. The firewalls examine each packet independently, and for that they are limited. This method is effective, fast and inexpensive in certain situations, as presented by Andrei-Daniel et a. in [15]. This work is based on configuring the topology into enterprise zones, where the network contains a mixture of different types of newer and older applications. They are proposed distributed firewalls which are composed of: a policy distribution scheme, a security policy and an encryption/authentication mechanism. The problem with this method is not applied to all topologies.
Kirak Hong et al. in [16] proposed the Mobile Fog Computing model. This method is not only based on providing a high–level programming to development on a large number of devices distributed in a large area; it also allows applications

based on demand of resources in the fog and in the cloud.
Puliafito et al. in [17] present a Fog Computing Platform. In this work, authors presented the Fog service to be migrated in order to be near the IoT device. Therefore, they present the CFC model (Companion Fog Computing). In addition, they introduce a CFC model and they derive a compromise of all the functionalities required in a platform in a reference architecture to make migration decisions.
Unfortunately, there are available works like in [16, 5] which are based on mobile fog. However, these works are missing both security at the user level and network level.
Because, existing research works lack precise definition of Mobile Fog Computing, works previously presented focus on the method of downloading/uploading information between cloud and fog across the mobile, optimization and management of resources and do not offer security solutions or access control. This remains a big challenge due to the severity of mobile fog attacks, dynamic and distributed access between mobile users, fog nodes and the cloud network.
Furthermore, MCC systems use heterogeneous network access technologies to achieve the requirements of modern services, including medical services with strict constraints.
Modi et al. in [18] presented the advantages of the Cloud based Healthcare system, where security of Healthcare data is important because of the high sensitivity of these data. This minimizes the acceptance of the Cloud based model in healthcare systems. Authors use in their work the re-encryption to exchange the encoding key matrix securely with the receiver and the linear network coding, which enhances security level.
This article is based on deep learning method. It is named Deep Neural Networks Energy Cost Effective Partitioning and Task Scheduling (DNNECTS). In addition, this way of working includes the following components: scheduling, task sequencing, and application partitioning. This scientific research is presented by Lakhan et al. in [19].
E-Healthcare in Cloud Computing is an active domain. Lately, access control user/network level, and stealing data of electronic health are challenges for health facilities The integration of healthcare in mobile cloud computing technologies with the Internet has guided more efficient processes in order to increase accessibility to healthcare providers and to offer a high quality of healthcare services [20, 21].
Zhen et al. in [22] proposed a mechanism that allow /deny the access user to the Online Healthcare Social Networks (OHSN) that they designed an Attribute-Based Encryption (ABE) scheme with the user revocation to combine the ciphertext and key update function. This ensures any authorized users in social assembly to access in the personal health information from other cloud data.
Several works have dealt with the access control user levels. They are designed of scheme based about user access like ABE (attribute-based encryption) in work [22]. The limit of this work is the lack of access control network level.
We find that in the works [18, 20, 22], the authors are interested only Healthcare service; hence the limits of these works are the lack of cooperation between the different levels of the network. Another limitation of existing solutions is the lack of attack filtering.
In this work, we are interested in overcoming the disadvantages of existing works and strengthening the security of mobile fog networks.

Three inter-levels and one intra-level of access control are introduced. The first inter-level is at its Mobile Node (MN). The second inter-level is executed when a mobile node tries to access the fog node. Due to lack of resources, the intra-level is executed when it moves to another fog node. Then the third inter-level is executed. Its control users accessing the cloud network is done via fog nodes.

2. Objectives

Our work proposes a protection of the VMs existing in the cloud; more than the mobile network from unauthorized access to hinder malicious accesses, where we have added the level of fog to add efficiency of the management of distributed resources. For this reason, distributed packet-based firewalls are used as a protection mechanism in three levels (cloud, fog and mobile).
Existing works are based on MCC or fog separately. These works proposed access control at the user-level or network-level one by one. Even if there is work that combines fog and mobile, they suffer from the lack of access control. To enhance the security of the MCC with an efficient management of distributed resources, we present a new architecture called MFCC (Mobile Fog Cloud Computing), based on a synchronization and collaborations inter networks and intra network, where we developed the intra network collaboration was developed by the MDFC (Mobile Distributed Firewall and Controller) in [10]. Moreover, distributed firewalls and rule migrations between firewalls are modeled using a Sequential Finite State Machine Se-FSM to propose the DCFC-cloud (Distributed and Cooperative Firewall/Controller in Cloud) published in [11].
In fact, this work is an extension of our paper, published in [10], where our objective is to ensure distributed network-based access control via a distributed and a cooperative firewall in MCC environment. In this work, we are interested to secure the architecture of the MFCC by using the advantage of fog computing features and the mobility of devices.

Our proposal increases the security of the network and user levels by introducing a distributed access control using firewalls. For that, we propose to:

- Ensure fourth access control levels by distributing and cooperating filtering and trusting processes among networks.
- Secure the MFCC topology (Mobile, Fog, cloud) based on distributed filtering. In fact, distributing sensitive data, like blocked packets, user's data and rules makes the network vulnerable to several attacks To satisfy these objectives, we propose to introduce for the 4 levels of computing in our architecture (Cloud, Fog, Mobile, User), 3 inter-level of cooperation and one intra-level in the fog computing. The cooperation aims to enforce both user and network access controls.

This paper is an enhancement of our work, published in [10] and [11], where we:

- Revise the method of computing the local Risk by adding risks related to medical staff.
- Revise the computation of the Total Risk to consider the output of the malicious packet detection module.
- Add an XACML model for the monitoring process, responsible for detecting intrusions
- Extend the network division into zones and master zones to Cloud and Fog computing layers.
- Add different levels of cooperation between architecture layers.
- Test the proposed solution in a sensitive environment, which is e-health.

This paper is organized as follows. In the second section, we introduce some related works. Section 3 presents the proposed architecture, MFCC. The simulation and evaluation results are presented in Section 4. Finally, Section 5 concludes our work.

3. Methods

MFCC Solution

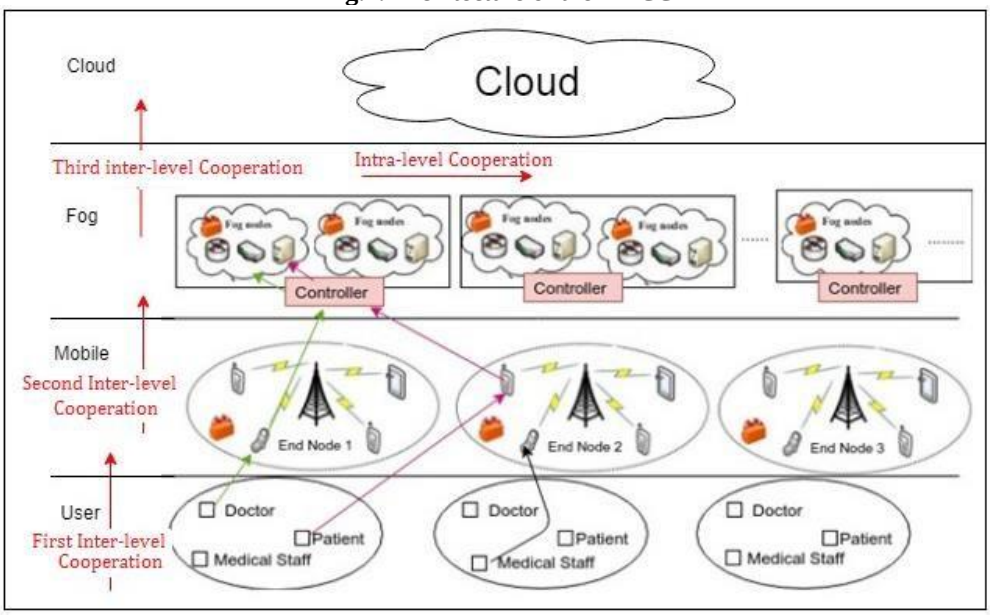
Mobile Cloud Computing (MCC) is a technique allowing each mobile user to be able to use the cloud services that enable flexibility and cost reduction. In fact, MCC, Cloud and Fog enable sharing many of the same attributes like virtualization, and distributed resources. They also manipulate the same resources like compute, storage and networking. Furthermore, fog computing is more geographically distributed than Cloud and it decreases latency. So, in our work, we propose to provide healthcare services via mobile devices by using the MFCC technology. The distributed and dynamic topology of the MFCC provides the service to medical end users with less latency and high security levels by deploying distributed Firewalls and adding access control levels.

3.1 MFCC Architecture

The architecture of MFCC is built on the edge servers. It is composed of three inter-levels and one intra-levels of access control, as presented in the proposed architecture of fig.1.

As presented in Figure 1, the proposed architecture of the MFCC is based on three types of network levels, where we admit that the mobile users are from healthcare systems, where doctors, patients and medical staff can profit from the cloud services using their mobile UE.

Fig.1. Architecture of the MFCC



The MFCC architecture is composed of a fog level. Whereas the Cloud level is composed of a set of Zones, controlled by firewalls and a set of Master zones, supervised by Manager Firewalls, responsible for exchanging rules. In addition, the fog computing is composed of a set of Fog Nodes (FN) using a set of firewalls. In addition, each firewall contains a list of ACLs in a file named blacklist containing the already blocked information. This level is divided on zones, where each one is supervised by a controller using a blacklist to facilitate rule migration between firewalls. As defined in our previous work [10, 11], the blacklist is a file containing information about malicious packets and the values of risk probabilities. It exists in the Controller. In order to prevent malicious connection, we add a firewall in each cellular node in the mobile network.

In our work, we investigate the trust management access control at both the network and the user levels. For the user-based access control, we based our work on [23], where additional analysis of the patient data, saved in the mobile devise and sent to the FN and to the cloud to be stored and consulted later.

3.2 Proposed Trust-based access control

The trust-based access control is a module aims to determine if an event is suspect. In addition, estimating trust is helpful in making decisions to avoid the mischievous situations.
At this level, we define the function of trust, which is based on the following metrics. This function is represented by Inequality

(1) where if the sum of risk probabilities multiplied by attack impact and the sum of metric multiplied by their weight are more than a threshold denoted S, the related event to the accessed user is trustable.

These metrics are:
Same-FN: It identifies the users of the same cell (first level in our architecture). It must connect to the same FN to deliver the same service. If this condition is satisfied, then the value of this metric is equal to 1, else it is equal to 0
Feature: This value takes into consideration the type of connected users identified by their ID (IDuser, IDdoctor, IDstaff-medical).
If ID (connected user) exists in a list (patient, or patient-Doctor or patient-staff) then Feature □ 1
Else
Feature ⇓ 0Attribute a trust value Tuser to the connected user. This is set by the security administrator, where $Tuser \in$ [0 1], according to his/his appreciation.

$$T = \sum^{n} (PRisk * Ij) + \sum^{k1} (Mk * Wp) < S \text{ with } \sum^{k1} Wp = 1 \quad i=1\ k=1,p \quad p=1 \tag{1}$$

With Ij: Impact: number of packet occurrences
Mk: The metric value k1: Number of metrics
W : Weight of each metric, reflecting its importance
PRisk: Probability value risk of attack
n: Length of Blacklist in the controller or ACL in the firewall
In the above, we propose three inter-levels and one intra-level of access control, which need cooperation between nodes fog (NF) via used firewalls (i) and (ii) and between fog and cloud in addition between Mobile user and fog via controllers (i) and (j).

As for the content of blacklist is in the case of blocked package, the following fields Information are considered: the number of attempts of a one user “nb”, IP source/destination and port source/destination.

In case a package is rejected by ACL or information is in the blacklist, this is dealt with by the following procedure “Rejecte- packet”. The packet containing (ID1, ID2, ID-Sour-Fw, Information, nb) is sent from the controller supervising the firewall source (with FW: Firewall).

Reject-packets: This procedure is responsible for deciding the deny/allow decision based on ACL and the information packet saved in blacklist. If the result of function return “Block packet with ACL” or “Block packet with existing information Blacklist” then the firewall exists.

```
1: procedure Reject-packets (ACL, Blacklist, packets: String;)
2: if packets ∈ACL then
3: display “Block packet with ACL”
4:// send ID1 , ID2 in broadcast to all FW exist in the same NF and to the NF and Controller neighbor.
5: else if packets ∈ Blacklist then
6:  display “Block packet exists in the Blacklist”
7:  end if
8:end if
9: end procedure
```

As shown in Fig.2, the process begins when a user requests a connection to a Mobile Node (MN).

In the first case, when ID1 and ID2 do not exist in Blacklist and ACL of the NM, the controller has to control the information about this user in this case it must to execute the function “verification_ID” and function “Trust”. It is the first inter-level of access control. Thereafter, the mobile device has to inform the associated controllers by sending their identifiers (ID, ID1 and ID2) (Step 5). It is the second inter-level of access control. When the associated Controller receives the ID1 and ID2, it checks if ID2 exists or not in his Blacklist (Step 6).
With

$$\begin{cases} ID_1 = Id\_patient\_Doct\ ID_2 = Id\_patient\ if\ user = patient \\ D_1 = Id\_Doct\_patient\ ID_2 = Id\_Doct\ if\ user = Docteur \\ ID_1 = Id\_staf\_patient\ ID_2 = Id\_staf\ if\ user = staf \end{cases}$$

Trust: This function makes a Value of trust. If the result of this function returns true, then the calculated Trust value is below the threshold. In this case, this information about Trust is sent to the associated nodes.

```
1: function Trust (PRisk, W, S: real; M: String ;)
2:  T⎸ ∑^n  (PRisk ∗ Ij)+ ∑^k1 (Mk ∗ Wp) // Equation (1)i=1  k=1,p
3: if T < S then
4: result⇓true
5: // send this information about Trust to the associated nodes and block this packet
6: else result⇓ false
7: end if
8: return result
9: end function
```

Verification_ID: This function is based on the id of the users who are logged in to visit the same information. This function consists of the followings:
a doctor can visit the information of one or more patient (//who is his patient) a user can visit only this information
a staff officer can visit his patient's information This function returns true in case of input ID of the patient is correct.

```
1: function Verification_ID (ID1, ID2, ID: Integer; list-patient-Doct, list-patient-Staf: table; user-type: String; result: Bool)
2: if (user-type== “patient”) then
3: { if ( ID2 == ID) then
4:result ⇓ true
5:else
6:result⇓ false
7: end if}
8: end if
9: if (user-type== “Docteur”) then
10: { if (list-patient-Doct [i]== ID) then
11: //i: counter table: “list-patient-Doct”
12: if ( list-patient-Doct [i]== ID1) then
13: result ⇓ true
14: end if
15: else
16: result⇓ false
17: end if}
18: end if
19: if (user-type== “Staf”) then
20:{ if (list-patient-Staf [j]== ID) then
//j: counter table: “list-patient-Staf”
21: if (list-patient-Staf [j]== ID1) then
22: result ⇓ true
23: end if
24: else
25: result⇓ false
26: end if}
27: end if
28: return result
29: end function
```

If he does not exist in his Blacklist and in his ACL (Step8 [case1]), it consults their Table-Zone (Table containing the ID of the neighboring Zone) (step 9) to send information packet, value of Trust function and ID1, ID2 to NF(i), NF(j), and Controller(j). If we need communication within the Fog level, data about blocked packet are sent to a different node. The latter executes the Trust function and the verification-ID function. Here, it is the intra-level of access control. In case of Trust function returns "True" (step 10), the information about the monitored packet is stored in the destination node.
If the associated controller consults its Table-Zone (step 9) and finds ID (Step 8 [case 2]), then it broadcasts information about this packet. The data are ID1, ID2, and information packet; step 11. After that, we execute Trust function to send and save this information to the Firewall Manager. This is the Third inter-level of access control.

3.3 Cooperation levels

The first, the second and the third inter-levels of cooperation in the cloud network and between the cloud and fog network have been discussed in our previous work [10].
The main idea of this work is to cooperate firewalls/controllers used in our topology in different networks levels (cloud network, fog network, mobile network, and user).The latter is a host that detects any abnormal user behavior in the first inter-level and the host in the other inter-level; it creates blocking policies on that host and sends them to its neighbors. Therefore, we propose two levels of cooperation.
Intra-level: Cooperation between firewalls/controllers in the same level as in cloud level and in Fog level.
Inter- level: Cooperation between different network levels firewalls/controllers.

3.3.1 Intra-level Cooperation (in Fog)

Intra-level cooperation at the Fog computing level is determined when the firewalls block a Mobile user based on their ACL or their Blacklist. Thus, it informs the firewalls neighbors to take into consideration the blocking information and to record it in the Blacklist of firewalls destination.
For the Intra-level cooperation at the Fog, the controller's existing in FN running proceeds in two phases, as follows: For the first phase:
The first phase should be periodically repeated in order to update the Table-Zone. In addition, this phase is executed once a rule is used to block traffic. This phase, called the initial phase, is used to discover zones. The controller makes a sweep of the firewalls located in the same Zone that it is taking in charge.
Each controller also has to discover its neighborhood. A controller (i) sends the ID of its firewalls with the used ACLs to the neighboring controller (j) based on the proximity parameters between controllers. Once the FWs ID and the ACLs are received, the controller (j) compares the ACLs of connected FWs with those received from other controllers.
In the second phase:

In the case of a blocking rule, a message is sent from the controller to which the firewall source belongs. The XACML language is used for this message. It extends the extensible Access Control Markup Language (XACML). This language belongs to OASIS Standard [24]. This message has the following format:

RN (8 bits)	ID firewall-Source (32 bits)	ID firewall-Destinations (32 bits)	ID Controller-Source (32 bits)	ID Controller- Destination (32 bits)	Nb-attempts (3 bits)	Blocking Information (with using ACL/ Blacklist) (120 bits)	Hash (160 bits)

Where:

- RN: (Random Number): It is a Nonce sent only once to guarantee anti-reply; 8 bits are assigned to this field.
- ID Firewall- Source: This is the identity of the firewall that has detected the blocking rule. Because the ID of the firewall can be an IP address, we use 32 bits of this field.
- ID Firewall-Destinations: It is the identity of the firewall that receive the Blocking packet Information from firewall-Source
- Destination controller ID is the identity of the controller. This controller is the equipment responsible for the migration of the rules between the firewalls. As the controller is a network component identified by its IP address, for that we assign 32 bits.
- Blocking packet information: It contains the information sent to the destinations firewall. We send in this field data from the packet header. Therefore, we assign 120 bits.
- Nb-attempts: This is the number of attempts executor by the same user refused by the FW. In our work, to code the attempt value, we need 3 bits.
- Hash: It is the result of applying the hash function on the entire message to guarantee authentication and integrity. For that, we propose the use of the HMAC (SHA1) scheme. That is why we need 160 bits.

To determine the destination Controller ID, the controller to which the firewall source belongs sweeps its Table-Zone.

The table contains firewall IDs belong to the same zone and the ID of the neighboring controller, then the message will be sent in broadcast to another FW and to neighboring controller. After that, the destination FW saves the blocked packet Information field in a blacklist, if the probability of matching between a rule and a part of it. It is more than a threshold or the priority value.

3.3.2 Inter-level Cooperation (User-Mobile or Mobile-Fog or Fog-Cloud)

These levels of cooperation are done once the Firewall denies the traffic destined to the next network. In this case, the Firewall has to identify the zone destination in Fog level of the denied traffic. This is retrieved using the destination IP-address existing in the ACL/Blacklist. The message data are sent between the Firewall and the controller of the interested zone in Mobile-Fog level and between two Firewalls in case of User-Mobile and Fog-Cloud levels. The former has the following format:

RN (8 bits)	ID firewall- Source (32 bits)	Destination (Controller-ID OR Firewall-ID)(32 bits)	Blocking Information (120 bits)	Function (hash) (160 bits)

Where:

Destination Controller-ID OR Firewall-ID is the identity of the controller or Firewall to which the message is sent. As the controller/Firewall are a network component identified by its IP address. (32 bits)

3.3.3 Cooperation specification

To model the inter/intra-level cooperation, we used the defined symbol provided as follows.

We employed notations $Event^s$,$ubject^s$,$Object^s$, and $Action^s$ to reference the set of events, subjects, objects, and actions, respectively. An event is the occurrence of an action on an object that is done by a subject during a period, where an event ei $\epsilon Event^s$ is defined by a quadruple:

$\forall$ ei ϵ $Event^s$, ei = <si, oi, ai, ti >, si ϵ $Subject^s$, oi ϵ $Object^s$, ai ϵ $Action^s$, ti ϵ $\mathbb{N}$

Where

ei: event: Cooperation Level (inter or intra); si : Subject such as: Controller or Firewalls; ti : Execution time; oi : Object like Destination-Controller or Destination-Firewall; ai : Action like Write Blocking information in a Blacklist of Destination- Firewall or Create new rule (ACL) in Destination-Firewall

In addition, we have presented in the following the properties of an event ei which is a four linear relationship, described as a quadruple where,

$ei = < si, oi, ai, ti > \Leftrightarrow \ < ei, si >, < ei, oi >, \ < ei, ai >, < ei, ti >$ (2) Next, we present the functions of subject, object, and actions, respectively:

$\forall$ ei ϵ $Event^s$, ei = <si, oi, ai, ti >$\Diamond$ subject (ei) = si

$\forall$ ei ϵ $Event^s$, ei = <si, oi, ai, ti >$\Diamond$ object (ei) = oi

$\forall$ ei ϵ $Event^s$, ei = <si, oi, ai, ti >$\Diamond$ action (ei) = ai

To define cooperation, the event ei is related to the event ej and this is expressed by (3), where when ei and ej are two events with inter-level cooperation

Then: ei and ej are two events in different network

(Like: ei Fog; ej in Cloud; Or ei in Mobile Cellular; ej in Fog; Or ei User level; ej in Fog) when ei and ej are two events with intra-level cooperation:

Then: ei and ej are two events in the same network

(Like: (ei and ej in Cloud) ; Or (ei and ej in fog) ; Or (ei and ej in Mobile Cellular)) ei : $\sim r$: ej (3)

Furthermore, we define the Access Control List Function ACLF: as:

ACLF is defined as ACL $\subseteq$ $Event^s\times$ $Subject^s\times$ $Object^s$ $\times Action^s$. To determine the set of rules related to existing events in the system, we define the rule pi = <si, oi, ai> in ACLF Where si is not authorized to do action ai in object oi at any time.

Besides, we define the Blacklist function FBlackList. Here, FBlackList is defined as heading- packet $\subseteq$ $Event^s\times$ $Subject^s\times Object^s$ $\times Action^s$. This determines the set of Blocked-Information events in the system. Moreover, we define the Blocked- Information BIj= <sj, oj, aj> in FB lack List Where sj is not authorized to do action aj in object oj at any time && (ei ϵ ACLF): $\sim r$: (ej ϵ FBlackList) In our case, a system that combines multiple networks (User, Mobile, Fog, Cloud) target to the explicit violation of security. So, for that we have to base on the Emergence Event Set EES (with EES $\subseteq$ $Event^s$) of cooperations between the levels of the proposed architecture.

We base this on the functions presented above, where we admit that the header information of the packet is similar to the information saved in ACL when object (ei) = object (ej). So, the associated action is the action of the (pi)

$\exists$ ei, pi □ (ei ϵ EES) ^ (pi ϵ ACLF) ^ subject(ei) = subject (pi) ^ object(ei) = object (pi) ^ action (ei) = action (pi)

When the information of the packet is similar to the information saved in blacklist, where object (ei) = object (ej) and/or

subject (ei) = subject (ej), the associated action is the action of the (BIj). This is used for the creation of new rules or the storage of the denied packet in the blacklist. Wherever, there is a new decision, it is necessary to send the information to neighboring firewall (Event ej), as follows:
∃ ej, BIj ^ (ej ϵ EES) ^ (BIj ϵ FBlackList) ^ subject(ej)=subject (pi) ^ object(ej)=object (BIj) ^ action (ej)=action (BIj)

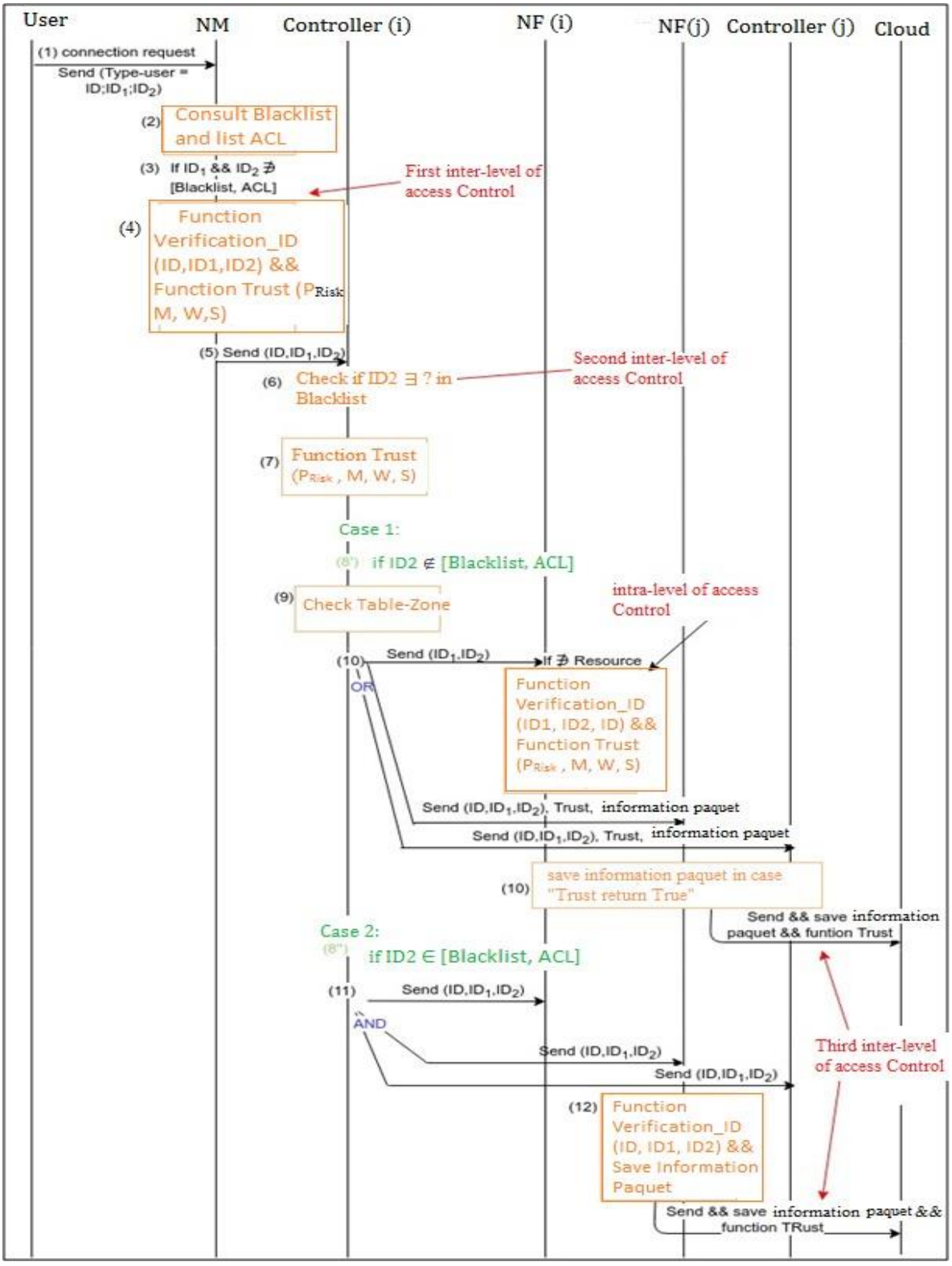


Fig 2. Flowchart of the proposed cooperation process
With Table-Zone which is aTable that contains the ID of the neighboring Zone i,j : Zone(i) and Zone(j)

4. Results

4.1 Achieved issues for healthcare application

In this section, it can be seen that the proposed model, MFCC, achieves in the following:

- Efficiency: The existing firewall in all inputs of FN in the fog level, the existing firewall in all inputs of MN in the Mobile level, together and in addition the existence of cooperative Controllers in each Zone increases the network security level by protecting the network from distributed attacks. Furthermore, we use TMAC scheme [23] to protect the user security levels. In addition, the use of the Trust Rejected-packets Verification_ID process to supervise the network and users' levels, which adds the efficiency of our MFCC model.
- Optimization: The use of FN allows us to minimize latency, and eliminates the network saturation during data exchanges; this is done because of the existence of the notion of cooperative controller in the fog level zone.
- Authentication of firewalls: The firewall identity is used to authenticate architecture components. Moreover, the verification of the hash of firewall identifier prevents spoofing attacks.
- Data integrity: The use of hash functions in message data exchanged between Firewalls allows us to guarantee the integrity of the data.

4.2 Performance evaluation

To evaluate the MFCC scheme, we first base our analysis on the delay parameter because Fog Computing manipulates some of the delay-sensitive requests and passes others to cloud computing. The fourth level as user interface that sends service request to FN. These request input to a set $\boldsymbol{K}$ of Fog devices through a Mobile Node (MN). We take note of the followings that Fog device i is corresponds to FN in our architecture. In addition, there is a set $\boldsymbol{S}$ of cloud servers, each of them lodging a number of Virtual Machine (VM). The process requests are forwarded from each fog device to each cloud server. The communication delay should be taken into consideration. In the following, we are principally interested in the delay of communication between the levels of our architecture. Moreover, we present in Table 1 the main notations used in this paper.

Table 1. Parameter Description.

Symbol	Definition	Unit
$\boldsymbol{K}$	Set of Fog devices	
$\boldsymbol{S}$	Set of Cloud servers	
qj	Integer: the number of VM at Cloud server j	
Sri , Srj	Service rate (Fog device i, Cloud server j)	
Tai , λ	Traffic arrival rate (Fog device i, Cloud server j)	(requests)/ s
D	Delay (Total)	Unit time
Dij	Communication Delay between Fog device i and Cloud server j	Unit time
$DMobile{-}Fog$	Communication Delay between Mobile device and Fog device i	Unit time
αj	Binary: on/off state of Cloud server j	

4.2.1 Delay-based evaluation

Delay for Fog Computing: We model our fog system by a queuing system. For each fog device I, the traffic arrival rate is Tai and the service rate is Sr . The calculation delay of Fog device (waiting time plus service time in fog) is equal to:

$$D_{i-fog} = \frac{1}{Ta_i + Sr_i} \quad (4)$$

Delay of Cloud Server: We characterize the Cloud Server by the model Erlang-C (M/M/q queueing). In the following, this model

introduces the delay as follows: $\left[\frac{C\left(\frac{(q,\lambda)}{\mu}\right)}{q\mu - \lambda} + \frac{1}{\mu} \right]$

Where q is the number of VM, λ is the traffic arrival rate, μ is the service rate and $\underline{c}$ $(q,)$ *is the* Erlang's C Formula [25]. In our case, we assume that each of all VM has the same service rate Srj.

From what previously presented, for the Cloud Server (j) with their sate "On" or "Off", αj and qj represent the running Machine. The calculation delay of Cloud server is equal to:

$$D_{j-Cloud-server} = \alpha_j \left[\frac{C\left(\frac{(q_j, Ta_j)}{Sr_j}\right)}{q_i Sr_j - Tr_j} + \frac{1}{Sr_j} \right] \quad (5)$$

Communication Delay between Mobile-Fog-Cloud-Server: This is based on QCI (QoS Class Identifier) [26]. It is preconfigured by the operator on one or more nodes (e.g, eNodeB). Furthermore, the characteristics of QCI describe the packet transfer processing in terms of the following performance characteristics (resource type (GBR or non-GBR); priority; Delay Budget Package; Packet error loss rate…). In addition, in our work we will choose the clause named "Packet Delay Budget" and we choose the resource type "GBR".
So, the total delay of transmission from the Mobile to Fog device i to Cloud Server j is presented by Equation (6):

$$D = (DFog{-}Cloud + DMobile{-}Fog) * TrMob{-}Fog(i){-}Cloud(j) \quad (6)$$

With
$DFog{-}Cloud = Di{-}Fog + Dj{-}Cloud{-}Server$ $DMobile{-}Fog$: The Packet Delay Budget [26].
$TrMob{-}Fo(i){-}Cloud(j)$: The traffic rate sent between Mobile to Fog device i to the Cloud Server j.
We propose in the following to simulate the proposed MFCC architecture to evaluate the introduced network overhead in terms of communication delay. Moreover, we show that our proposal consumes less delay compared to other works based on the equations previously presented. This is verified showing Fig.3, which illustrates that our approach presented in red is always below the works proposed in [6]. This shows that our proposal consumes less transmission delay with an increase of the security level. This is mainly due to the distribution and mitigation of used rules between firewalls and controllers.

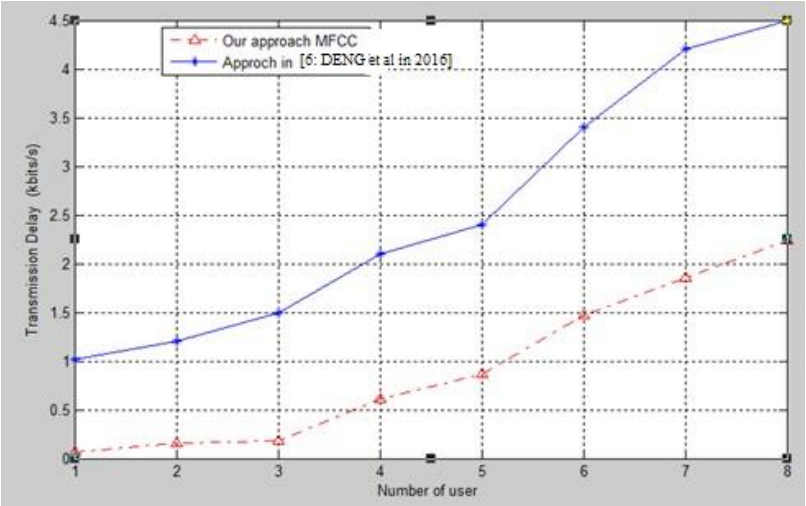


Fig 3. Delay with respect to the number of connected users.

Moreover, for more performance and to introduce our security analysis, we use in Mobile and Cloud level the tool Nessi2 to prove that our model MFCC is able to prevent complex attacks and estimating the blocked packets. This is done by simulating a DDoS via the profile of "bot" that already exists.
Thus, in this part, we have the goal of achieving the next point:

- evaluate the network's overheads in terms of blocked packets
- Prove that our proposed MFCC is able to prevent DDoS attacks.
- Prove that our MFCC have less delay compared to other works.

4.2.2 Nessi² based evaluation

To evaluate the architecture in the cloud level, we implement it in Nessi2. At this level, we expect to put access control with the ability to detect the malicious packets. Thus, we create cloud architecture and four sub-networks. At this level, we added a firewall for each FN using new profiles. Furthermore, we implement the "Trust" procedure, where retrieved results are illustrated in Table 2.

Table2. Trust function value

User	n $\sum_{i=1}$ PRisk	k1 $\sum_{k=1,p}$ Mk ∗ Wp	T (Trust)= T = $\sum_{i=1}^{n}$ P+ $\sum_{Risk\ k=1,p}^{k1}$ M ∗ W k p
1	**0.2**	**0.2**	**0.4**
2	**0.3**	**0.2**	**0.5**
3	**0.891**	**0.5**	**1.391**
4	**0.99**	**0.6**	**1.59**
5	**0.7**	**0.6**	**1.3**
6	**0.81**	**0.4**	**1.21**

The Blocking Rate is computed using the following Equation (7).

$$\text{Blocking rate} = \frac{\mathbf{NB_{Sent-packets}} - \mathbf{NB_{Blocked-packets}}}{\mathbf{NB_{Sent-packets}}} \quad (7)$$

With
NBSent−packets: The Number of sent packets.
NBBlocked−packets: The Number of Blocked. (Packets calculated by considering the values of the trust function presented in Table 2)

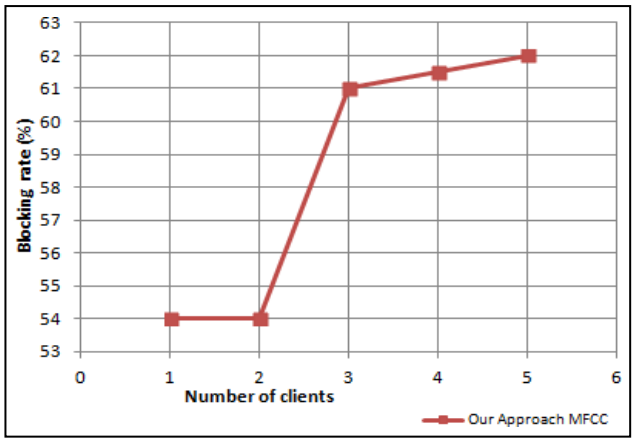


Fig 4. Blocking Rate versus Number of Clients.

From Figure 4, we notice that when we increase the number of clients that are sending packets, the number of blocking packets will be increased because of the increase in tracing information about already sending existing packets in the blacklist.
For this evaluation, we present the division in fog computing level by using firewalls in each input of FN and Controller in each Zone.
We will show that the proposed model MFCC can detect the attacks by simulating DDoS attacks with Nessi2. This is done by adding the new profiles in each firewall exists in the FN.
Table 3 presents the used simulation parameters, in our work.

Table 3. Simulation Parameters.

Parameter	Value
Simulator	Nessi[2]
Language of implementation	JDK 3.8.1
Number of zones	2
Number of FN	4
Number of firewalls	4
Number of clients	8
Number of Controller	2
Packet length	700 bytes
Tic	0.05 ms

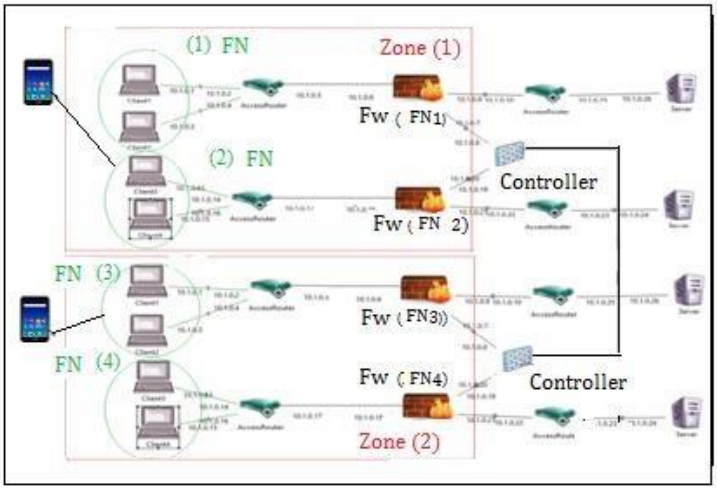
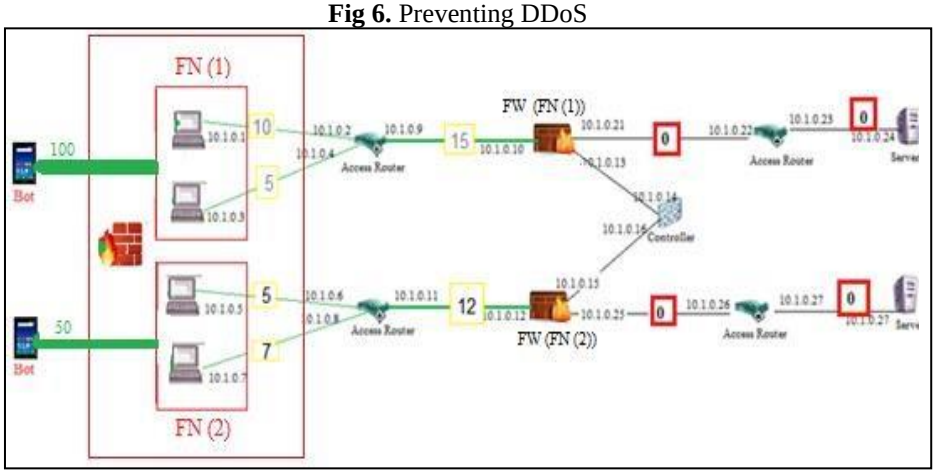


Fig 5. Proposed simulation MFCC level fog with Nessi².

As illustrated in Figure 5, the proposed MFCC architecture level Fog who is divided into FN and Zones. This architecture represents firewalls that are placed at the input of a FN and the Controllers are placed at the input of a Zone. The latter is composed of a set of instances, destination servers and controllers.
In the Zone 1, we visualize a set of FN each one contains a firewalls and a controller. In our simulation, we interested to implement a set of rules in each new firewall profiles.

Fig 6. Preventing DDoS



In this context, we have added a "Bot" profile that simulates a DDoS attack source. This is done to verify that our proposed approach, MFCC, is capable of preventing DDoS attack.

As it is presented in the Fig.6, the high number of packets is generated because of the DDOS attack in the link between the source "Bot" and the victim. Traditional firewalls are not able to block them. With the presence of our solution and the addition of our new function in the profile of firewall, we noticed that we can detect the presence of the DDoS attack and to make sure that there are no packets sent when the MFCC is being run. Therefore, at the level of Mobile-Fog we notice that our scheme increases the level of security.

5. Discussion

In this part, we summarize the achieved issues with our proposed MFCC framework compared with other work detailed in the section of related work. This is shown in Table 4, where MFCC is able to guarantee security by preventing DDoS (Distributed Denial of Service) with different network handsets (Cloud, Fog and Mobile). In addition, our proposal adds a mobile level, where medical related users, such as patients and medical staff can connect securely to the Cloud/fog network. The security is achieved by deploying two cooperation levels (inter, and intra).

Table 4. Comparison results.

works / Metric		[3], 2018	[10], 2018	[11], 2018	[12], 2018	[13], 2018	[14] 2021	[16], 2013	[18], 2019	[20], 2016	[22], 2017	[6], 2016	Proposed MFCC
Used of cooperation levels	Inter- level		*										*
	Intra- level		*	*									*
The existence of User, Mobile, Fog, Cloud at the same													*
The existence of MCC		*	*							*			*
The existence of MFC (Mobile Fog Computing)								*					*
The existence of Cloud, Fog												*	*
The existence of Fog					*	*							*
The existence of Cloud							*		*				
The existence of Healthcare services									*	*	*		*
Access Control level	Network		*										*
	User										*		*
Attack filtering				*									*
Attack detection					*	*	*						*

References

1. C. Luo, L. Yang, P. Li et al "A Holistic Energy Optimization Framework for Cloud-Assisted Mobile Computing" IEEE Wireless Communications, 2015, vol. 22, no. 3, p. 118—23
2. X.SUN, and N.ANSARI "Green cloudlet network: A distributed green mobile cloud network" IEEE Network, 2017, vol. 31, no 1, p. 64-70.
3. S.ALONSO-MONSALVE, F.GARCÍA-CARBALLEIRA and A.CALDERÓN "A heterogeneous mobile cloud computing model for hybrid clouds" Future Generation Computer Systems, 2018 vol. 87, p. 651-666.
4. F.Bonomi, R.Milito, J.Zhu and S.Addepalli "Fog computing and its role in the internet of things" Workshop on Mobile Cloud Computing, 2012, p.13-16
5. T.H.LUAN, L.GAO, Z.LI et al "Fog computing: Focusing on mobile users at the edge" arXiv preprint arXiv: 1502.01815, 2015.
6. R .DENG, R.LU, Ch.LAI et al "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption" IEEE Internet of Things Journal, 2016, vol. 3, no 6, p. 1171-1181.
7. Y.JIANG, Z.HUANG, and D.TSANG "Challenges and solutions in fog computing orchestration" IEEE Network, 2018, vol. 32, no 3, p. 122-129.
8. Sh.XU, J.NING, LI.Yingjiu et al "Match in my way: Fine-grained bilateral access control for secure cloud-fog computing" IEEE Transactions on Dependable and Secure Computing, 2020, p. 1-13.
9. R.AMIN, S.KUNAL, A.SAHA et al "CFSec: Password based secure communication protocol in cloud-fog environment" Journal of Parallel and Distributed Computing, 2020, vol. 140, p. 52-62.
10. F. Kamoun-Abid, A. Meddeb-Makhlouf and F.Zarai "Distributed firewall and controller for Mobile Cloud Computing" 15th ACS/IEEE International Conference on Computer Systems and Applications, 2018, Aqaba, Jordan.
11. F. Kamoun-Abid, A. Meddeb-Makhlouf, F.Zarai and M. Guizani "Distributed and Cooperative firewall/controller in cloud environments" 13th International Conference on Availability, Reliability and Security, 2018, Germany.
12. A.SOHAL, R.SANDHU, S.K.SOOD et al "A cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments" Computers & Security, 2018, vol. 74, p. 340-354.
13. R.ROMAN, J.LOPEZ and M.MAMBO "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges" Future Generation Computer Systems, 2018, vol. 78, p. 680-698.
14. R.ZAGROUBA and R.ALHAJRI "Machine Learning based Attacks Detection and Countermeasures in IoT" International Journal of Communication Networks and Information Security, 2021, vol. 13, no 2, p. 158-167.
15. A.D.TUDOSI, D.G.BALAN and A.D.POTORAC "Secure network architecture based on distributed firewalls" International Conference on Development and Application Systems, 2022. p. 85-90.
16. K.HONG, D.LILLETHUN, U.RAMACHANDRAN et al "Mobile fog: A programming model for large-scale applications on the internet of things" Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing, 2013. p. 15-20.
17. C.Puliafito, E.Mingozzi, C.Vallati, F.Longo and G.Merlino "Companion Fog Computing: Supporting Things Mobility Through Container Migration at the Edge" IEEE International Conference on Smart Computing, 2018, p. 97-105.
18. K.J.MODI and N.KAPADIA "Securing Healthcare Information over Cloud Using Hybrid Approach" Progress in Advanced Computing and Intelligent Engineering, 2019, Singapore, p. 63-74.

19. A.LAKHAN, Q.Ul.MASTOI, M.ELHOSENY et al "Deep neural network-based application partitioning and scheduling for hospitals and medical enterprises using IoT assisted mobile fog cloud" Enterprise Information Systems, 2021, p. 1-23.
20. M. Abomhara, H.K. Yang and M. Geir "Access control model for cooperative healthcare environments: Modeling and verification" International Conference on Healthcare Informatics, 2016, p. 46-54.
21. N. Kahani, K. Elgazzar and J R.Cordy "Authentication and Access Control in e-Health Systems in the Cloud" IEEE 2nd International Conference Big Data Security on Cloud, IEEE International Conference on High Performance and Smart Computing and IEEE International Conference on Intelligent Data and Security, 2016, p. 13-23.
22. Z.QIN, J.SUN, D.CHEN et al " Flexible and Lightweight Access Control for Online Healthcare Social Networks in the Context of the Internet of Things" Mobile Information Systems, 2017, vol. 2017.
23. B.W.DAOUD, A.MEDDEB-MAKHLOUF and F.ZARAI "A Trust-based Access Control Scheme for e-Health Cloud" IEEE/ACS 15th International Conference on Computer Systems and Applications, 2018, p. 1-7.
24. eXtensible Access Control Markup Language (XACML) Version 3.0. 22 January 2022. OASIS Standard http://docs.oasis- open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html
25. N. Gautam "Analysis of Queues: Methods and Applications" Boca Raton, FL, USA: CRC Press, 2012.
26. 3GPP TS 23.203 V15.4.0 (2018-09) 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Policy and charging control architecture (Release 15).