Human-centric Computing and Information Sciences

RESEARCH Open Access

TACRM: trust access control and resource management mechanism in fog computing

Wided Ben Daoud[1*], Mohammad S. Obaidat[2,3,4], Amel Meddeb-Makhlouf[1], Faouzi Zarai[1*] and Kuei-Fang Hsiao[5]

*Correspondence: wided.bendaoud@gmail.com; faouzifbz@gmail.com
[1] (NTS'Com) Research Unit, ENET'COM, University of Sfax, Sfax, Tunisia
Full list of author information is available at the end of the article

Abstract

Fog computing network is designed as an extension of the cloud due to the need for a supporting platform capable of ensuring the requirements of Internet of Thing (IoT). The growth of fog based fifth generation mobile communication (5G) system is challenged by the need for data sharing security. In fact, without properly securing access to Fog node resources in IoT network, services providers may not be able to achieve the desired performance. Indeed, fog computing obviously confront numerous security and privacy risks, due to its features, such as huge scale geolocation, heterogeneity and mobility. Thus, we propose a security model that is based on cooperation between IoT and fog. This model integrates an efficient access control process associated with a monitoring scheme to ensure secure cooperation between diverse resources and different operational parts. Indeed, a comprehensive scheduling process and resource allocation mechanism using our security model is proposed to improve the intended performance of the system. In fact, our main contribution is to introduce a distributed access control based on security resource management framework for fog-IoT networks, and proactive security scheme under ultra-trustworthiness and low-latency constraints. After evaluation based on iFogSim, we have proved that our scheme not only provides low latency with high security and privacy, but also reduces the complexity of administration and management of security and resources mechanisms.

Keywords: Fog computing, IoT network, Security, Privacy, Access control, Trust, Resource management, Scheduling

Introduction

Despite the fact that cloud computing is an effective solution for handling data in distributed environments, it is considered as an appropriate way to efficiently process the mass data generated by IoT devices [1]. It delivers centralized resources for data computation and storage, which can affect metrics like delay and bandwidth limitation [2, 3]. Inherently, Fog nodes are distributed within the proximity of users; a characteristic that reduces latency and establishes adjacent localized connections.

Recently, the combination of cloud/fog, and IoT communication networks has received a great attention and widely emerged [4]. IoT exploits the fog computing capacities for virtualizing the tasks of IoT devices, but it still has restricted capability and acquires long delay [5].

Though the primary purpose of Fog paradigm is to achieve all tasks with high performance, the security features must be considered as part of the Fog system to guarantee the Confidentiality, Integrity, and Authentication (CIA) of all types of data [6]. Figure 1 illustrates the major interaction between cloud computing, fog nodes, and the IoT devices.

However, by using wireless communications to interconnect nodes, this can make the system vulnerable to attacks, such as sniffing, spoofing and Denial of Service (DoS) attacks [7]. That is, the amount of vulnerabilities in Fog computing is in height because it resides between the end IoT devices and the cloud data centers [8].

In the context of the cloud, storing data and hosting multiple users are significant security risks. Strong tools are currently available in the cloud to protect the user data. This is becoming more complex in the fog computing environment, as the additional security risks related to the traffic carried over nodes. For instance, a hacker could deploy malicious applications, which could in turn exploit a vulnerability that may damage or reduce the quality of service of the network [9]. In fact, a single compromised Fog node can produce the potential access point for a Man-in-the-Middle attack (MITM) and disturb all attached users, leak data, overuse the service and damaging the data in the fog nodes. A MITM attack can be initiated by a malicious internal user and can threat the Fog network by sniffing, hacking, injecting and filtering end user incoming data [10, 11].

As a result, the design and the built of a strong system that can effectively provide security and privacy without sacrificing and losing performance, is the primary challenge of wireless fog/cloud based IoT network. Protection against these attacks is important because human life is at stake. Potentially, the greatest common technique to eliminate such problems is to control the user access and monitor the entire systems. That is why; the security and privacy methods need to be reviewed to facilitate computing with high performance.

Since fog nodes have limited resources, it is very challenging for them to process a huge number of simultaneous requests. In this case, the Fog computing performance can be decreased to a great level. In addition, without properly securing access to Fog node resources in an IoT network, providers may not be able to achieve the desired performance.

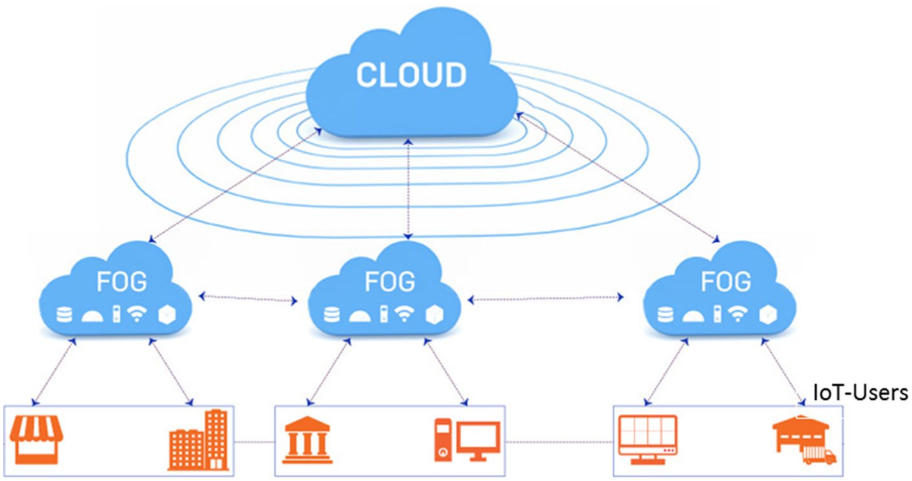


Fig. 1 Interaction Cloud-Fog-IoT

Currently, we cannot anticipate new and evolving dangers that IoT networks will have to protect against, but we have the basis necessary to build autonomous network management solutions/methods that must deal with them. Besides, we should be fed with insights provided by real-time controlled analysis systems, to decrease or stop malicious threats.

As a result, the design of a combined wireless fog-cloud computing approach based IoT network becomes essential for network control security and resource management.

To address the above problems, a comprehensive security and quality management system, guided by a set of defined security policies and a set of resource management strategies, is necessary to improve the intended security and performance of the system. To supervise the overall traffic types in fog computing, which are combined to serve IoT devices, collaborative clusters are introduced. A cluster will manage a known number of fog nodes. The proposed component has a monitoring agent that evaluates and update the trust level of connected users to protect the network from attackers during user access. Moreover, if the trust level is maintained during user access, then a certificate is delivered to this user as a trust authorization. Furthermore, a resource management is proposed to efficiently use the resources, hence, guaranteeing the Quality of Services (QoS) of the interacted wireless fog in IoT network.

To present our risk-based access control solution, this paper is organized as follows. "Related works" section presents some related works. "Trust access control and resource management mechanism (TACRM)" section describes the combined fog based IoT network architecture. Then, simulation results and discussions are given in "Evaluation results and discussions" section. We conclude our work by a conclusion in "Obtained results" section.

Related works

Much research on task scheduling and resource allocation are available for cloud and fog computing, attempting to resolve the abovementioned concerns. Bousselmi et al. [12], focused on QoS-based workflow planning. Their scheduling algorithm leads to increase the quality of service (QoS) in cloud computing, based on certain metrics like resources availability, cost, and data transmission time. In Agarwal et al. [13], proposed an algorithm for resource provisioning in fog computing. In their work, the fog layer contains functional components like Fog Server (FS), Fog Server Manager (FSM), and virtual machines (VMs).

Verma et al. [14] suggested a load balancing algorithm between the client, fog and cloud layers. They tried to solve the problems related to delays, the execution time and the allocation of resources. A threshold is assigned to the fog layer to control the number of tasks performed. Once this threshold is exceeded, the requests for tasks will be transferred to the cloud layer [14].

More recently, Xu et al. [15] have studied the allocation of computer resources in the fog (edge). They proposed the model of Zenith, which allows service providers to use a resource scheduling algorithm that takes latency into account in established contracts.

The authors of [16] presented a new design of collaboration between IoT, Fog, and Cloud computing for IoT service assignment and resource allocation. They introduced an algorithm that aims to generate decision rules of linearized decision tree based on three conditions (services size, completion time, and VMs capacity). The authors presented simulation results showing a better performance by optimizing the resource distribution in fog/cloud environments. Their work consists of dividing the network architecture based on 1/m/m1 model, where (1) refers to cloud broker, (m) refers to many paths, (m) refers to many fog brokers in fog environments, and (1) refers to IoT devices users. In fact, the algorithm process was presented as follows. The IoT devices send service request to fog broker in fog. Fog broker divides data into multiple blocks where they are assigned to certain VMs. Each block is divided into multiple chunks, which are sent to multiple processors. After receiving the processed data, the processor combines them again into one big data and returns the result to user IoT devices. The authors guarantee service, especially, the availability of servers processing in fog or in cloud. However, an extra time for the procedure is needed due to the division procedure. Then the combination of data is introduced and can add a delay-based overhead. In addition, from a security point of view, they did not consider the risks of insider attacks in the fog/cloud networks.

The authors of [17] proposed a method that aims to schedule the tasks in cloud system, considering the QoS constraints. The proposed algorithm, named the grouped tasks scheduling (GTS), is based on distributing the tasks into five classes; each class has tasks with similar attributes (user type, task type, task size, and task latency). Then, the scheduling of tasks is transformed on available resources. The scheduling method depends on the attributes of the tasks that belong to each category, and also, on the execution time of task.

Yet, the GTS algorithm uses only four attributes to achieve quality of service. In addition, this proposed algorithm only works for independent tasks and requires first queue all jobs for classification.

For the purpose of improving fog network performance and reducing cost, the authors of [18] proposed a task scheduling algorithm based on priority levels. In fact, they ensure that the communication between fog nodes in the fog layer can lead to efficient resource allocation and load balancing. The proposed algorithm processes all the user requests, and assigns services for them based on their priority levels.

In addition, a dynamic resource allocation method, named DRAM, to perform the load balance for the fog systems is proposed by the authors of [19]. They presented a system framework for IoT applications in fog, and designed a static resource allocation and dynamic service migration strategy. However, the authors need to analyze the negative impact of the service migration, including the traffic, the cost for service migration, and especially, the decrease of performance of services. Furthermore, it becomes more important to consider some security methods to avoid some threats and privacy breaches, either in a distributed architecture or even in a centralized system. In this regard, there is some research that depicted the security aspect in their works.

Thus, we propose, in this paper, a novel approach for securing data in the fog using a distributed strategy based on trust estimation. We recommend to monitor data access in the fog and detect abnormal user's behavior and deactivate illegitimate anomaly actions. Additionally, data owners in fog nodes need to monitor the whole system to enforce the security through access policies in order to minimize risk and enhance trust. The proposed mechanism enhances data security process and thus allows the detection of illegal access. Hence, it allows a valid distribution of resources. We also introduce an efficient algorithm for service allocation based on resource availability.

Trust access control and resource management mechanism (TACRM)

The poor implementation of the security system can lead to serious issues. Therefore, it is essential to carefully select the security measures to be incorporated, the required modules/elements, and the defined performance criteria.

To the best of our knowledge, it is not always true that enhancing the security of a system does automatically means compromising on performance. For instance, security schemes should be developed as a basic element of Fog ecosystem, for the reason that if they were not secure, their performance could possibly be reduced due to attacks such as malware, misuse of resources, etc.

Fog systems are essentially composed of IoT devices and wireless sensors. If it is not hidden and secure, the wireless network offers attackers exceptional freedom to interrupt and capture sensitive exchanged data. Thus, by taking into account heterogeneous and flexible tasks, we propose fog computing security strategies and resource management techniques to optimize the execution delay and then minimize the total price of management in the system. The proposed TACRM is a security access control scheme based on trust assessment and monitoring user's activities, where a resource management strategy in integrated wireless fog in IoT networks is applied to improve the resource utilization and reduce the transmission latency.

Fog systems that continuously manage personal data from end user to the cloud paradigm and vice versa, should supervise and detect abnormal network actions through automated application of security and performance rules and policies. Consequently, each Fog system should apply security-efficient network monitoring methods. They must be implemented as a key part of every Fog node, hence, any malicious action can be identified and completed before any real destruction happens.

The first step of the proposed scheme is the clustering, where a Machine Learning technique involves the grouping of data points. Given a set of FN, we divide them into small cells. Firstly, the number of FN to use in each cell are randomly selected. Then, each cell is managed by a fog node manager (FNM) that plays the role of the cluster head. Hence, when an IoT-User tries to access a FN, the access request will be handled by this cluster head.

TACRM architecture

Our work is based on dividing the fog computing system on small cells that contains a number of fog nodes (FNcell). Every cell is managed by a FNM as a cluster head. We consider that each FNcell contains different services categories. In fact, when the IoT user requests such service, it is served as long the demanded service is available.

The first task of this latter is to control the access of every new IoT user. It computes the trus level of such an access request to the system. After that, an authorization is allocated for every user containing especially his trust level, assessed by the cluster head. The authorization is used after the resource allocation procedure (its affectation to the FN). In his next access to such a FN in that small cell, this user doesn't repeat the procedure of the access, since he/she has an authorization; quite simply, need to present it.

The second task of our FNM is to play the role of a resources classifier. In fact, by knowing the amount of resources available at every moment, it schedules the IoT users depending on the types of their requests. After that, these users are affected to a specific FN based on the type of requested resources.

If the resources are not available, the user has to wait. It will be served based on FIFO (First In, First Out) scheduling method.

Indeed, the FNM monitors the whole system to supervise the user activities. Consequently, for the purpose of collaboration, the incorporated monitoring agent in the FNM reports the user's behaviors. In addition, this agent is responsible for filling the field of the authorization, in particular, the level of the trust of this user. As soon as the user reaches the FN, he/she presents the authorization to have the required amount of resources. Within the fog environment, the monitoring is a functional until the end of the activity of the user. If the system detects any malicious behaviors, the FNM has the ability to revoke the authorization, and that IoT user will be rejected from the network. This procedure is repeated for each user.

The architecture of the network, illustrated in Fig. 2, presents different IoT users requesting the access to the FNcell to gain computing services. The proposed network architecture comprises two layers: IoT layer and the fog layer, and three other steps.

The fog layer comprises several fog nodes. These nodes make available data services independently, to users, with reduced latency.

TACRM functionalities

The main proposed functionalities of the TACRM approach are summarized below:

1. Access control management: The FNM computes the trust of each IoT-user. We assume that the user's trust is assigned according to the behaviors of each user to be considered in the access control to the FNcell. It changes based on the user's profile. Each user's task is associated with a weight, which is attributed by the security system, and which reveals the importance of the activity of this user.

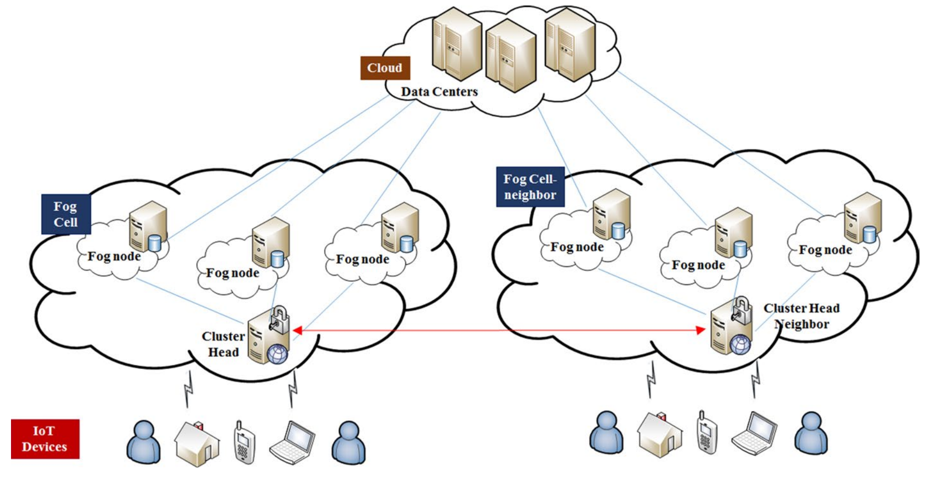


Fig. 2 TACRM architecture

The trust computation consists of an input vector $X = x_1, x_2, \ldots$ representing the metric identifying each user; a weight vector $W = w_1, w_2, \ldots$. The result of applying the weights $W_{i(i=[1..m])}$ to the inputs $x_{j(j=[1..n])}$ is the trust value.

Thus, trust is computed based on the formula (1).

$$\mathrm{Trust} = \sum_{i=1}^{n} \mathrm{w_i x_i}. \tag{1}$$

The access decision function is specified by the inequality (2).

$$\mathrm{Ac\_Des} = \begin{cases} 1 & \mathit{if} \sum_{i=1}^{n} \mathrm{w_i x_i} \geq \theta \\ 0 & \mathit{else} \end{cases}. \tag{2}$$

With, θ is a predefined threshold.

As cited in our previous work in [20, 21], the metric used for the computing of the user's trust, are factors identifying such a user, such as access historic, type of the used host and the user's localization.

Finally, the FNM has to return a response to have an authorization containing the trust level of the connected user, and announcing to users receiving their requested services from a precise FN. Eventually, the trust is assigned according to the following 4 levels:

1 = Highest trust level

2, 3 = Medium trust level

4 = Lowest trust level.

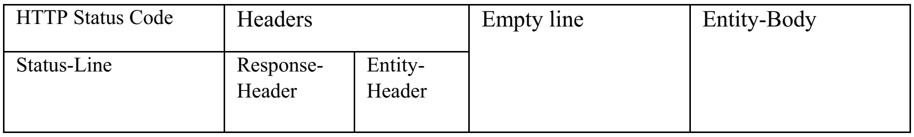
Because the most used secure IoT user's connections are based on the HTTPS protocol, we present in the following the authorization format using the structure of a HTTPS response, which contains the following components (Fig. 3).

Response header fields

The response header fields allow the server to pass additional information about the response, which cannot be placed in the Status-Line. These header fields give information about the server and about further access to the resource identified by the Request-URI.

```
Response-Header = Location            ;
          | Server                ;
          | WWW-Authenticate       ;
```

WWW-Authenticate = In this field, we set the trust value of that user, which indicates his trust level.



HTTP Status Code	Headers		Empty line	Entity-Body
Status-Line	Response-Header	Entity-Header		

Fig. 3 Authorization format-HTTPS response protocol

Entity header fields

Entity-header fields define optional meta information about the resource identified by the request.

```
Entity-Header  = Allow                ;
            | Content-Encoding       ;
            | Content-Length         ;
            | Content-Type           ;
            | Expires                ;
            | Last-Modified          ;
            | Extension-header       ;
```

Content-length: It contains information about the resource allowed by the cluster head, after the achievement of the scheduling and the resource allocation procedures, based on the resources available in the system.

2. Monitoring process: To supervise the overall traffic between cloud and fog computing, which are combined to serve the IoT devices, collaborative agents/controllers are introduced. In fact, the monitoring process detects malicious behaviors of connected users and reports alerts, which are shared between different FN in the same small cell, leading to a more effective use of the network infrastructure.

The exchange of alert messages between FNM and the monitoring agents, communicating in the same FNcell, and between FNM over different FNcell is based on the Intrusion Detection Message Exchange Format (IDMEF) [2]. Furthermore, by the user of the IDMEF, the FNM performs a correlation between them in order to send and receive reports about users who cannot achieve their resource requested.

In fact, when a user does not perform his requested services (not available at this cell), the FNM redirects the user to the neighbor cell and sends an IDMEF message to the neighbor FNM about the status and the information about the user as his ID, and his trust level, in order to not repeat the whole access control process.

As illustrated in Fig. 4, in the 'source' case, the FNM would give information about the user that commits the suspicious event. Then, in 'additional data', the FNM gives the user's trust level previously computed.

3. Resource management: Since the wireless network varies over time, it should be noted that the resource allocation process needs to be dynamically adjusted to meet quality of service requirements. In fact, if the user is trusted, the classifier allocates appropriate communications and computing resources to each user. Additionally, since the IoT-user has a strict latency requirement, resources must be efficiently utilized while guaranteeing the QoS of the IoT-users. Then, the Resource Management, including Scheduling and Resources Allocation, is responsible to determine user's priority that will be benefited by the requested resources and to allocate resources to users.

In fact, IoT-users are selected according to the security trust computation to be arranged during the scheduling process based on the priority scheduling method.

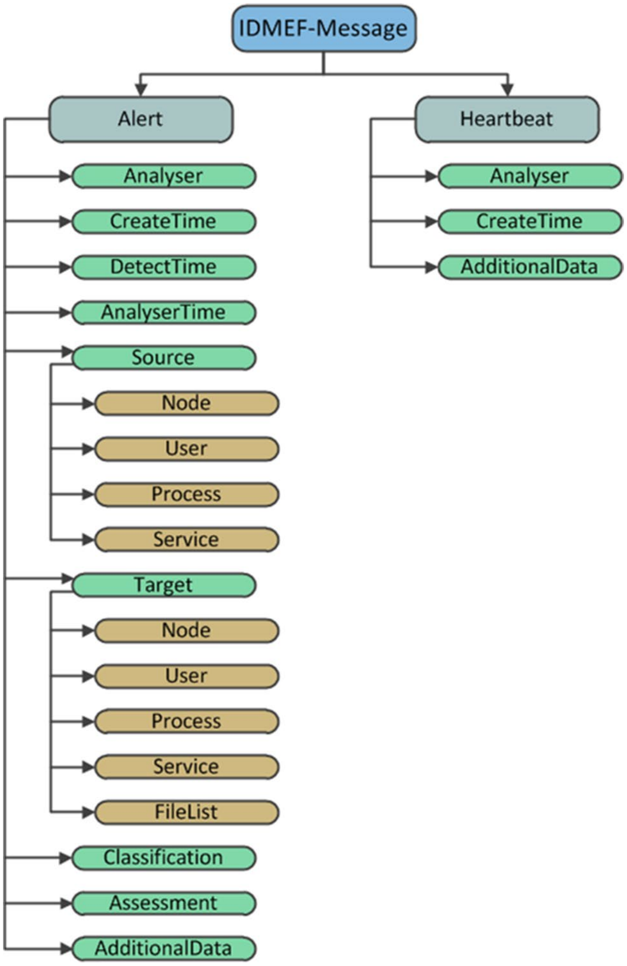


Fig. 4 IDMEF format

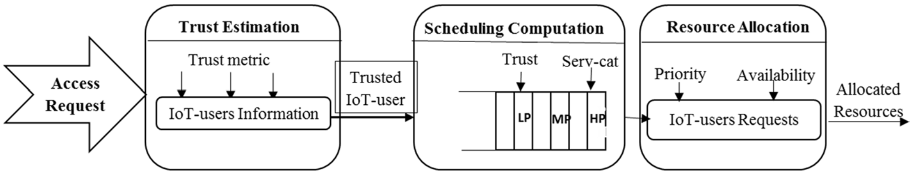


Fig. 5 Proposed process for TACRM solution

All these functionalities defined and explained above, are described in Fig. 5, where the entire process starts when an access request will be handled by the trust estimation to decide whether the IoT-requester is trusted or not. After that, the access request of these trusted users will be scheduled based on trust level and service type, in order to finally, allocate resources based on the priority algorithm and the availability of the requested services.

As shown in Fig. 6, the user's access is granted according to the priority level based on the trust value and the service category $\mathrm{Serv}_{\mathrm{CAT}}$ to which user belongs.

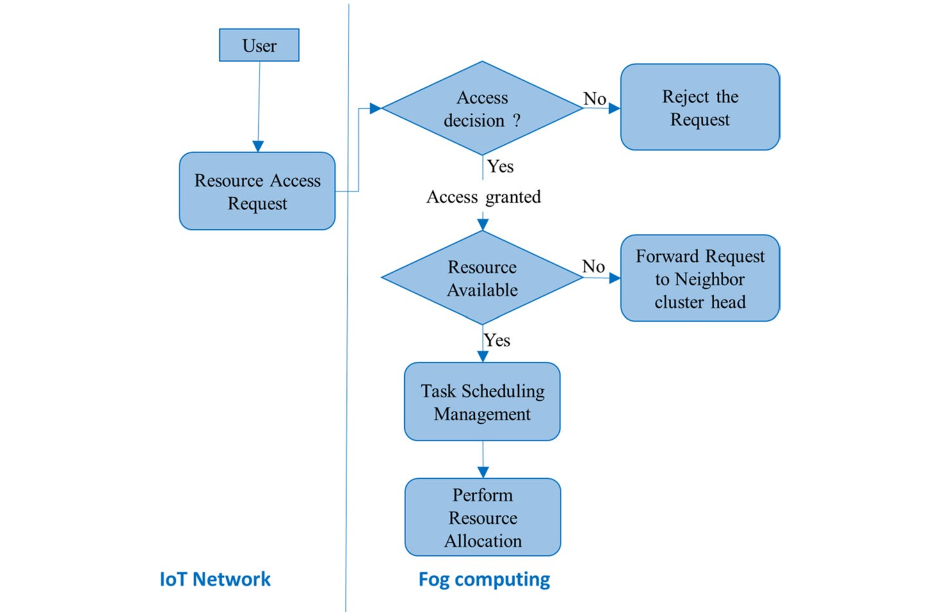


Fig. 6 Flowchart of task offloading process between IoT and Fog networks

```
{     If (Trust == 1)
            and if ServCAT ==H or M)
               Give HP to the ac-reqi
            and if ServCAT ==L)
               Give MP to the ac-reqi
      If (Trust ==2 )
            and if ServCAT ==H)
               Give HP to the ac-reqi
            and if ServCAT ==M or L)
               Give MP to the ac-reqi
    If (Trust == 3)
           and if ServCAT ==H or M)
              Give MP to the ac-reqi
           and if ServCAT ==M)
              Give LP to the ac-reqi
    If (Trust == 4)
           and if ServCAT ==H or M or L)
              Give LP to the ac-reqi
}
```

where
Ac-reqi – access request from user i
Pri: Priority
servCAT - Service category
H, M, L: High, Medium, Low
HP: high priority
MP: medium priority
LP: low priority]

To define serv_{CAT} and to categorize services, every user must assess the amount and the category of the requested services, using one of the 3 following service categories (serCAT): SerCAT—high, SerCAT—medium or SerCAT—low.

To categorize the types of services that could be allowed by the Fog/cloud networks, we base our analysis on the standard 3GPP [22]. Table 1 contains more details about the priorities of services.

Then, according to the computed trust level, the available resources in the system, and SerCAT of ith access-request (ac-reqi), attained access-request is placed in one buffer, which will be served based on FIFO (First In, First Out) scheduling algorithm.

In the ideal case, when the user's trust value is very high and the requested resources (Res.Requested) are available, users should have the maximum of the requested resource (Res.max) whenever possible. However, if the trust value is very low, the user might receive a resource lower than the requested privileges (Res.Required).

After evaluating the access request and estimating the trust value, the FNM decides to grant or deny the permission. The permission granted represent a set of privileges, which are required to access the service requested.

In fact, when a request for a service from the user i (ac-reqi) arrives at the FNcell (service provider), the Classifier checks whether the access-request can be accepted or not by computing the trust value for that request:

Table 1 Standardized priorities services

Arranged classes	Priority level	Example services
HP: high priority	1	Mission critical delay sensitive signalling (e.g., MC-PTT signalling, MC video signalling) Mission critical user plane push to talk voice (e.g., MCPTT) IMS signalling
	2	Mission critical video user plane Conversational voice Non-mission-critical user plane push to talk voice
	3	V2X messages Real time gaming, V2X messages Electricity distribution—medium voltage Process automation—monitoring
MP: medium priority	4	Conversational video (live streaming)
	5	Non-conversational video (buffered streaming)
	6	Mission critical data Video (buffered streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
LP: low priority	7	V2X messages Low latency eMBB applications (TCP/UDP-based) Augmented reality Voice, video (live streaming) Interactive gaming
	8	Video (buffered streaming)
	9	TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)

```
if ( Trust<=Threshold)
    Allow the access-request;
Else
    Reject the access-request;
Classifier assigns the priority by calling the scheduler module;
Based on priority and the availability of the resources in FNcell, service scheduling happens as follows
{ (Pri) = Assign-priority (Trust, $Serv_{CAT}$)
      { For each user i having HP or MP, or LP
          Classifier Checks for the resource availability by communicating with the FNs $\epsilon$ same FNcell.
              if (required resources are available)
                  Schedule the ac-req i for the service and initiate the servicing of user i }
```

Evaluation results and discussions

To evaluate TACRM scheme, we propose to implement it on the case of the architecture shown in Fig. 7, where we assume that we have 10 users who want to access to the fog network, and where *RA* designates Resource Available in that FN. The users $U_{i(i=[1..10])}$ are arranged based on arrival time, where U1, U2, U3 arrived first. Then, after 5 min, U4, U5, and U6 arrived. After that, U7, U9, and U10 arrived the last one at the same time.

As illustrated in Table 2, we compute the trust value for the 10 users.

Based on our analysis, after the access control process, we still have only 7 trusted users that are allowed to access the network. Now, the scheduler algorithm is running, in order to give priorities to trusted users.

U2 is served first, then U2 and U3 the last.

U5 is served after that U2 and U3 achieved his tasks.

U7, U9, and U10 are scheduled based on their priorities; U10 will be served firstly, then, U9, and U7 the last one served.

Finally, as illustrated in the proposed architecture, resources are allocated to users based on resources available as in following;

U2 is served from FN1;

U1 is served from FN2;

U3 is served from FN2, because, FN2 still have resource available.

U5 is served from FN2, because, FN2 still have resource available.

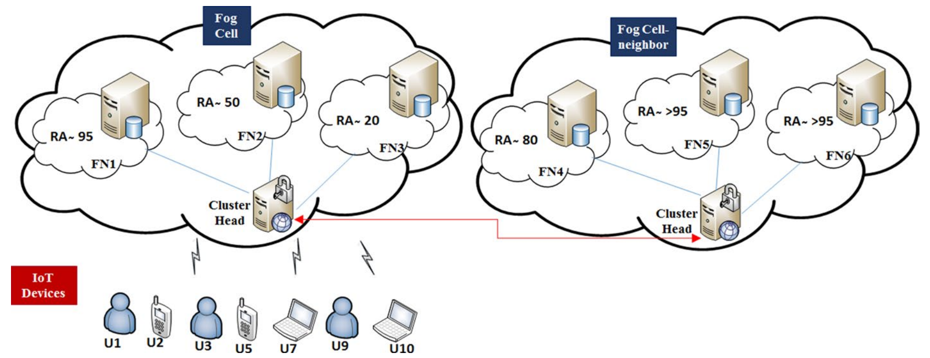


Fig. 7 Suggested case study architecture

Table 2 Results for priority allocation based on trust level and services types

Users	Trust value	Trust level	Requested services	ServCAT	Priority level
U1	3	3	Monitoring (20 Gb)	H	MP
U2	2	2	Mission critical (95CPU)	H	HP
U3	2.5	3	Conversational video (live streaming) (2 Gb)	M	MP
U4	7	Not trusted	Real time gaming (10 CPU)	–	–
U5	1	1	V2X messages (1 Gb)	L	MP
U6	9	Not trusted	Interactive gaming (4 Gb and 5CPU)	–	–
U7	4	4	Chat (76 Gb)	L	LP
U8	5	Not trusted	Conversational voice (3 Gb)	–	–
U9	1.5	2	Internet, e-mail, (100 Gb)	M	MP
U10	0.7	1	ftp (120 Gb)	M	HP

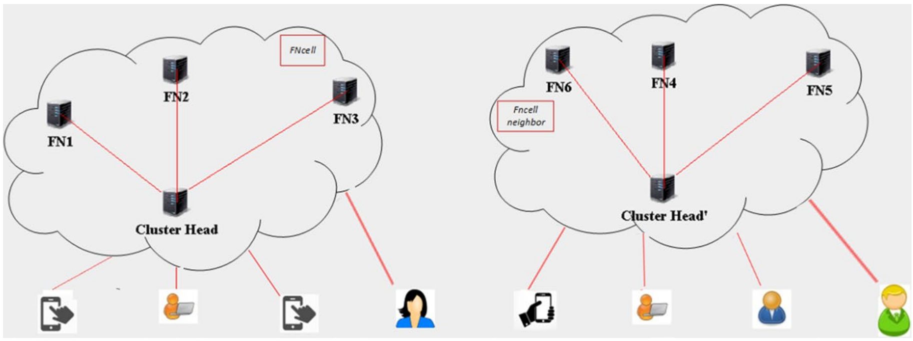


Fig. 8 Overall simulated architecture

U10 is served from FN5 from FNcell neighbor. In fact, FNM forwards the request for the neighbor FNM that executes resource allocation to FN5, without repeating the trust computing.

U9 is served from FN6 using FNcell neighbor.

U7 checks if FN1 still has resources available. Otherwise, he is served from FN4 from FNcell neighbor.

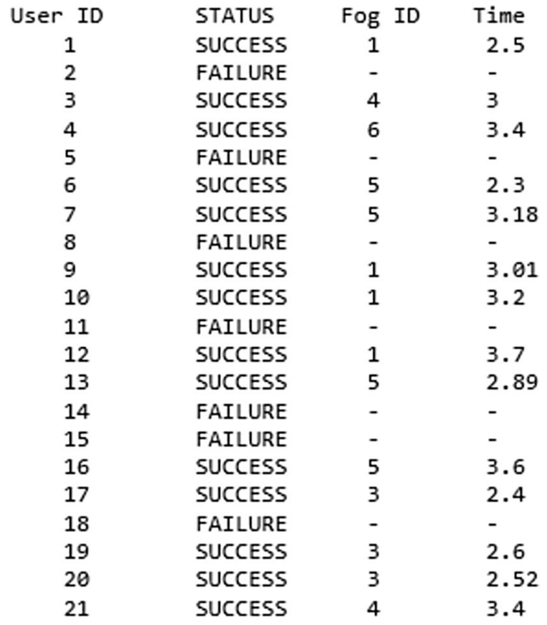
After analysis, we propose to examine the efficiency of access control and resource management based on priority scheduling using simulation analysis by means of the iFogSim [23] tool, which is a framework for modelling and simulation of infrastructures and services in Java jdk-8u191 and Eclipse-IDE.

Hence, to test the proposed algorithm, we drew our network topology as a Java application. The graphical view of the proposed architecture is generated by iFogSim as indicated in Fig. 8. The experiment was done with 2 cells using one FNM for each cell; each having 3 FN and 4 IoT devices per cell.

The experimental network configurations are illustrated in Table 3. In our PC, we used one Intel Pentium CPU N3540 and 4 GB RAM. It shows the default configurations for the fog nodes and IoT devices in terms of million instructions per second (MIPS), and RAM in gigabyte. In our simulation, we used the default values of

Table 3 Default entity configurations in iFogSim

Attribute	FN1, FN3, FN4, FN6	FN2, FN5	IoT devices
MIPS	1000	20,000	1500
RAM (GB)	10	4	2
upBw	10,000	100	10,000
downBw	270	10,000	10,000



```
User ID      STATUS     Fog ID    Time
   1         SUCCESS      1        2.5
   2         FAILURE      -        -
   3         SUCCESS      4        3
   4         SUCCESS      6        3.4
   5         FAILURE      -        -
   6         SUCCESS      5        2.3
   7         SUCCESS      5        3.18
   8         FAILURE      -        -
   9         SUCCESS      1        3.01
   10        SUCCESS      1        3.2
   11        FAILURE      -        -
   12        SUCCESS      1        3.7
   13        SUCCESS      5        2.89
   14        FAILURE      -        -
   15        FAILURE      -        -
   16        SUCCESS      5        3.6
   17        SUCCESS      3        2.4
   18        FAILURE      -        -
   19        SUCCESS      3        2.6
   20        SUCCESS      3        2.52
   21        SUCCESS      4        3.4
```

Fig. 9 Example of the various metrics reported by iFogSim

iFogSim, where the cloud and proxy connection has a latency of 100 ms, and fog to the end devices has 2 ms latency.

The proposed TACRM approach (user's security and resource management Algorithm) was compared with the GTS approach in terms of Application Latency (Response Time). Then, we compared our algorithm applied in fog computing with the existing traditional methods of resource management implemented in cloud only, in term of Network Usage.

Figure 9 presents an example of various metrics reported by iFogSim. The performance metrics used include the access control algorithm execution time calculated in milliseconds (ms). Some IoT users are rejected from the network. By using the access control algorithm, these users are not permitted to access the FN. The other users accessed to the network, will take an extra time to have their requested resources, due to the security access control and the delivering of the authorization. This calculated time is extended between 2.3 ms and 3.7 ms that is the maximum elapsed time by an IoT-user for having an access authorization. Yet, this extra time will be eliminated thanks to the resource allocation algorithm based on clustering. In all cases, this time is not long, compared to the achieved results.

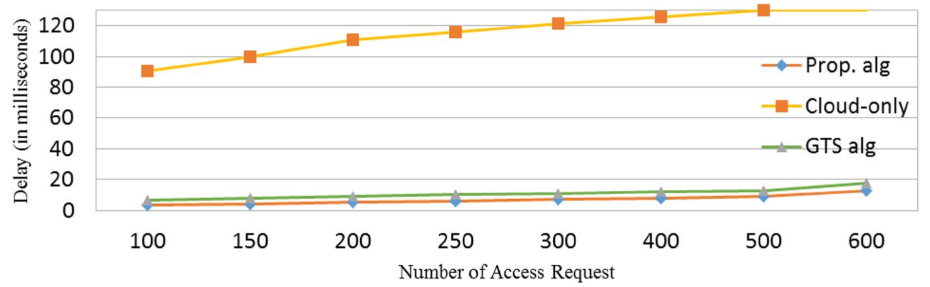


Fig. 10 Average latency of users services allocation vs. number of requests

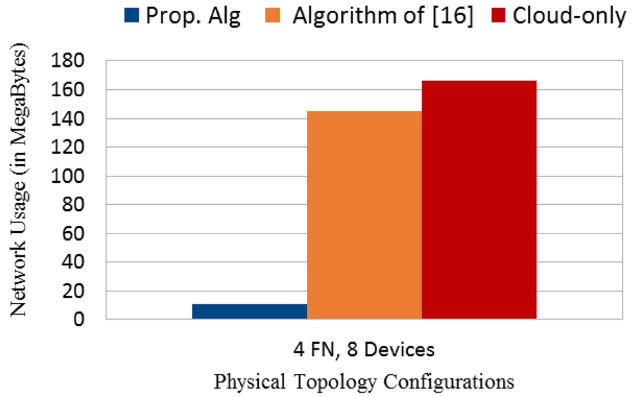


Fig. 11 Network usage vs. physical topology configuration

The results of simulation analysis demonstrate a favorable impact on network usage, and average latency of our proposed algorithm, where Figs. 10 and 11 show that the proposed algorithm provides lower execution time to all user access requests, compared to the recent algorithms proposed in [16, 17].

Figure 10 shows the system response time after running the process for a period of time. It describes the latency for getting each service from the fog/cloud servers; by varying the number of users, the results show that the latency increases as the number of users increases. This is due to the growing contention under a large number of users.

Figure 10 represents the response time in three scenarios; when connecting to cloud server, when introducing our approach in FN, and when applying the GTS algorithm to have services.

The results show that the services provided by the cloud server takes a long time with a minimum range of 90 ms, while the services provided by the FN with the application of the proposed TACRM algorithm have a maximum response time of 12.46 ms, with a gap time of 5.0328 ms with that of GTS.

GTS algorithm was suggested to schedule tasks into services based on 4 types of task attributes [17], which are used to measure priority of tasks. From a security point of view, the authors do not take into account the security risks related to connected users. In addition, our algorithm is based on clustering idea. So, the response time of users will be reduced than by using the GTS scheduling algorithm.

As shown in Fig. 11, there is an astounding reduction in network usage for the proposed algorithm, compared to traditional scheme using the cloud network, and the algorithm proposed in [16]; thanks to the use of fog networks. This is important because there is always a service in a certain fog node that is available. However, in [16], the division of data into multiple blocks engenders additional overhead in the network. Moreover, the combination of data adds a delay-based overhead.

Nowadays, many applications require to process data closer to its source to decrease latency and network traffic, and then, powerfully proceed with the data explosion. That is why, for the sake of optimization, we have chosen the fog paradigm to apply our proposed TACRM scheme.

Obtained results

In this section, we discuss and analyze the obtained results of TACRM scheme, where the following features are satisfied by our proposal:

Security

Ensuring homogeneous security in a heterogeneous fog environment using the trust based access control approach would be a necessity. Furthermore, authorization issues would arise as fog resources would be shared by various users. Without proper security measures, the network will be vulnerable to many threats and become easily compromised.

Intrusion detection and prevention

The monitoring process is functioning continuously, even after the gain of permission from the user. Moreover, a deactivating function is triggered when detecting abnormal activities and reporting them.

Scalability/resilience to network changes/failures

The user's information is vulnerable to network failures mostly at the Edge level (fog, IoT). Moreover, having hundreds of billions of connected devices and nodes forming the IoT paradigm, needs to overcome compute and store limits and improve performance. The proposed system has a global view of the network to observe and control the activities of every element in the network and to update trust according to the supervised metrics. In fact, the proposed FNM communicates, continuously, with the neighbors' FNM. We presented monitoring agents that are able to control and supervise a large number of resources and cooperate them via the notion of small cells supervised by cluster. This ensures the deactivation action in case of detecting changes in the user's behaviors. Hence, the proposed approach is applied for any fog network architecture.

Availability

The availability can be affected by failure of connectivity, hardware, or software matters. Hence, it must be guaranteed. The implementation of an efficient access control and resource allocation based priority scheduling approach in a highly distributed and heterogeneous fog environment means that availability will be a challenge. In fact, the deployment of our resource allocation approach and the information exchanged between cluster heads from various FNcells increases the availability of the system, hence, the users are served with minimum introduced delay. Moreover, the delivered authorization is used after the resource allocation procedure (its affectation to the FN). In his next access to such a FN in that small cell, this user doesn't repeat the procedure of the access, since he/she has an authorization; quite simply, need to present it. Eventually, we assure the availability of server in fog nodes to handle huge number of requested resources at any time, with considering to offer fast respond time, and to guarantee the satisfaction of QoS.

Network bandwidth

The moving of the processing from cloud to fog would reduce the network bandwidth usage. Moreover, the use of the clustering method in the fog domain would decrease network communication overhead as well. Hence, our proposed strategy has led to the optimization of distribution of large data in the fog.

Conclusions

In IoT-Fog, there is a huge amount of data that may need to be executed and manipulated. In such environment attacks may occur. Such attacks may occur due to inefficient and insufficient resource policies, as well as from a lack of monitoring of user activity.

The cooperation between these paradigms requires a robust security system to cope with expected attacks. An efficient resource management strategy to improve the performance of the environment is needed as well. Carrying forward our work reported in [20, 21], here we proposed a clustering algorithm for security and resource allocation based on priority. To this end, we suggested a cooperative access control scheme based on user's trust assessment and monitoring process in order to ensure high security level by inserting real-time constraints. Moreover, we described a scheduling and resource allocation scheme to guarantee a lowest latency level and better performance.

Furthermore, we illustrated the result of the efficient resources deployment for network environment. From the simulation results, we proved the impact of the implementation of our algorithm in the Fog paradigm towards solving the problem of latency that is a critical factor in IoT applications. Potentially, we succeeded to provide a proactive security scheme under ultra-trustworthiness and low-latency constraints.

Finally, this study has revealed insightful tracks for working in the future, such as the application of mobility on the fog nodes and understanding the manner of resource allocation in a dynamic environment supporting mobility.

Acknowledgements
Not applicable.

Authors' contributions

WBD and AM conceived of the presented idea. WBD developed the theory, performed the computations, and performed the numerical simulations. FZ and MO verified the analytical methods. All authors discussed the results and contributed to the final manuscript. WBD wrote the manuscript with support from AM All authors provided critical feedback and helped shape the research, analysis and manuscript. All authors read and approved the final manuscript.

Funding
No funding.

Availability of data and materials

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details
[1] (NTS'Com) Research Unit, ENET'COM, University of Sfax, Sfax, Tunisia. [2] Department of ECE, Nazarbayev University, Astana, Kazakhstan. [3] University of Jordan, Amman, Jordan. [4] University of Science and Technology Beijing, Beijing, China. [5] Ming Chuan University, Taipei, Taiwan.

Received: 3 April 2019 Accepted: 4 July 2019
Published online: 22 July 2019

References

1. Guan Z, Zhang Y, Wu L, Wu J, Li J, Ma Y (2019) APPA: an anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT. J Netw Comput 125:82–92
2. Bangui H, Rakrak S, Raghay S, Buhnova B (2018) Moving to the Edge-Cloud-of-Things: recent advances and future research directions. Electronics 7(11):309
3. Riad K (2018) Secure storage and retrieval of IoT data based on private information retrieval. Wirel Commun Mob Comput 2018:8
4. Tanwar S, Kumar N, Tyagi S, Obaidat MS (2019) Tactile internet-based ambient assistant living in fog environment. Future Gener Comput Syst J 98:635–649
5. Alrawais A, Alhothaily A, Hu C, Cheng X (2017) Fog computing for the Internet of Things: security and privacy issues. IEEE Internet Comput 21(2):34–42
6. Tanwar S, Kumar N, Obaidat MS (2019) Fog computing for smart grid systems in 5G environment: challenges and solutions. IEEE Wirel Commun Mag. https://doi.org/10.1109/MWC.2019.1800356
7. Xiao M, Zhou J, Liu X, Jiang M (2017) A hybrid scheme for fine-grained search and access authorization in fog computing environment. Sensors (Switzerland) 17(6):1–22
8. Puthal DD, Obaidat MS, Nanda P, Prasad M, Mohanty SP, Zomaya AY (2018) Secure and sustainable load balancing of edge data centers in fog computing. IEEE Commun Mag. 56(5):60–65
9. Jaïdi F, Labbene Ayachi F, Bouhoula A (2018) A methodology and toolkit for deploying reliable security policies in critical infrastructures. Secur Commun Netw 2018:22
10. Ni J, Zhang K, Lin X, Shen XS (2018) Securing fog computing for Internet of Things applications: challenges and solutions. IEEE Commun Surv Tutor 20(1):601–628
11. Zhang J, Chen B, Zhao Y, Cheng X, Hu F (2018) Data security and privacy-preserving in edge computing paradigm: survey and open issues. IEEE Access 6:18209–18237
12. Bousselmi K, Brahmi Z, Gammoudi MM (2016) QoS-aware scheduling of workflows in cloud computing environments. In: IEEE 30th international conference on advanced information networking and applications (AINA) 2016. pp 737–745
13. Agarwal S, Yadav S, Yadav AK (2016) An efficient architecture and algorithm for resource provisioning in fog computing. Int J Inf Eng Electron Bus 8(1):48–61
14. Verma M, Bhardwaj N, Yadav AK (2016) Real time efficient scheduling algorithm for load balancing in fog computing environment. Int J Inf Technol Comput Sci 8(4):1–10
15. Xu J, Palanisamy B, Ludwig H, Wang Q (2017) Zenith: utility-aware resource allocation for edge computing. In: 2017 IEEE international conference on edge computing (EDGE) 2017. pp 47–54
16. Alsaffar AA, Pham HP, Hong CS, Huh EN, Aazam M (2016) An architecture of IoT service delegation and resource allocation based on collaboration between fog and cloud computing. Mob Inf Syst 2016:1–15
17. Ali HG, Saroit IA, Kotb AM (2017) Grouped tasks scheduling algorithm based on QoS in cloud computing network. Egypt Inform J 18(1):11–19
18. Choudhari T, Moh M, Moh TS (2018) Prioritized task scheduling in fog computing. In: Proceedings of the ACMSE 2018 conference 2018. pp 1–8
19. Xu X et al (2018) Dynamic resource allocation for load balancing in fog environment. Wirel Commun Mob Comput. https://doi.org/10.1155/2018/6421607
20. Ben Daoud W, Meddeb-Makhlouf A, Zarai F, Obaidat MS, Hsiao KF (2018) A distributed access control scheme based on risk and trust for fog-cloud environments. In: Proc. 15th Int. Jt. Conf. E-bus. Telecommun. vol 1, no. Icete. pp. 296–302
21. Ben Daoud W, Meddeb-Makhlouf A, Zarai F (2018) A model of role-risk based intrusion prevention for cloud environment. In: 2018 14th international wireless communications & mobile computing conference (IWCMC) 2018. pp 530–535
22. T. Specification and G. Services (2018) 3gpp ts 23.203. Release 15
23. Mahmud R, Ramamohanarao K, Buyya R (2017) Latency-aware application module management for fog computing environments. ACM Trans Embed Comput Syst 9(4):1–21

Publisher's Note