Cognitive Management Framework for Fog Computing In IOT Case Study: Traffic Control System

Prof. Mostafa Abdel Azim Mostafa
Department of Computer Science. Arab Academy for Science, Technology & Maritime Transport
Arab Academy for Science, Technology & Maritime Transport
Cairo, Egypt
melbaqary@gmail.com

Ahmad Mohammad Khater Mohammad
Department of Computer Science. Arab Academy for Science, Technology & Maritime Transport
Arab Academy for Science, Technology & Maritime Transport
Cairo, Egypt
Ahmed_khater20@hotmail.com

***Abstract*—Fog computing is a new computational paradigm proposed by Cisco that subdues the shortcomings of cloud computing by transferring some of the core functions of cloud towards the network edge. Internet of Things (IOT) is a very promising technology, which are used heavily in many areas of life and with such a large and rapid spread produces a lot of challenges in IOT. As reported by Cisco, by the year 2020, the next generation Internet-connected devices would comprise of around 50 billion Internet-connected devices which are located at the edge of the network and require support for mobility, low latency, real-time, and location-aware services. The Cloud computing is used as a main component in IOT architecture to provide processing and storage services for IOT system but the conventional cloud computing architecture is completely centralized in nature, and therefore, fails to provide real-time and low latency services to billions of IOT devices at the edge of the network, simultaneously. The main objective of this research is to combine the capabilities of Fog Computing , Cognitive and Context-Aware IOT to build a cognitive management framework for Fog computing capable of adopting it's behavior in a context-aware fashion to maximize the number of functions that can be carried out in The Fog and to provide a solution for determining which functions should be carried out in the Fog and which should be carried out in the cloud and how the Fog should interact with the Cloud, The same function in some situations can be carried out in the fog while in other situations can be carried out in the cloud according to its context.**

Keywords—Cognitive IOT; fog computing; cloud computing; traffic control system component

I. INTRODUCTION

The first question needed to be answered, Is there any assessment of the suitability of Fog Computing in the context of Internet of Things? After research results show that as the number of applications demanding real-time service increases, the Fog Computing paradigm outperforms traditional Cloud Computing. However, it is mention worthy that for an environment with less percentage of applications demanding for low-latency services, Fog Computing is observed to be an overhead compared to the traditional Cloud Computing. Therefore, researches show that in the context of IOT, with high number of latency-sensitive applications Fog Computing outperforms Cloud Computing. To maximize the number of function that can be carried out in the Fog there is a need for a cognitive management framework for Fog resources capable of adopting it's behavior in a context-aware fashion. The main objective of this research is to Combine the capabilities of Fog Computing, Cognitive and Context-Aware IOT to build a cognitive management framework for Fog Computing capable of adopting it's behavior in a context-aware fashion to maximize the number of functions that can be carried out in The Fog and to provide a solution for determining which functions should be carried out in the Fog and which should be carried out in the cloud and how the Fog should interact with the Cloud, The same function in some situations can be carried out in the fog while in other situations can be carried out in the cloud according to its context. So, The Fog architectures should allow computing, storage, networking ...etc. tasks to be dynamically relocated among the Fog, the Cloud, the Things and between the fog Instances in the Fog taking in consideration the limited processing and storage capabilities of the fog instances and how to well utilize these resources. The Fog architectures should consider the decentralized dependent data distributed across multiple dynamic fog instances for real-

978-1-5090-6332-1/17/$31.00 ©2017 IEEE

time and low latency services. The framework will face a lot of challenges and limitations:

i. Performance; the system must respond in a real-time fashion to support large number of devices, sensors and applications with low latency and real-time services.

ii. Accuracy; the system should consider the decentralized nature of Fog Computing to be able take accurate decisions specially while serving a single heavy request in a parallel fashion through multiple fog instances.

iii. The system must handle the dynamic and mobile nature of the Fog Computing.

iv. The low processing power and low storage capabilities of the fog instances must be well utilized.

v. Where and when to execute the cognitive tasks of the system without affecting the system's performance.

The main objectives of this research to, minimize these challenges and limitations and maximize the throughputs. The implementation of this framework will apply for a Car accident scenario as shown in fig. (2).

II. BACKGROUND

A. Internet of Things(IoT)

The internet of things (IoT) is the network of physical devices, vehicles, buildings and other items—embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect, exchange data, see, hear, and smell the physical world for themselves.

B. Cognitive Internet of Things(CIoT)

Cognitive Internet of Things (CIoT) [1] is a new network paradigm, where (physical/virtual) things or objects are interconnected and behave as agents, with minimum human intervention, the things interact with each other following a context-aware perception-action cycle, use the methodology of understanding-by-building to learn from both the physical environment and social networks, store the learned semantic and/or knowledge in kinds of databases, and adapt themselves to changes or uncertainties via resource-efficient decision making mechanisms, with two primary objectives in mind:

- bridging the physical world (with objects, resources, etc) and the social world (with human demand, social behavior, etc), together with themselves to form an intelligent physical-cyber-social (iPCS) system;
- enabling smart resource allocation, automatic network operation, and intelligent service provisioning.

C. Cloud Computing

Cloud Computing [2], the recent trend in IT, takes computing from desktop to the whole World Wide Web and yet, the user does not need to worry about maintenance and managing all the resources. User has to bear only the cost of usage of service(s), which is called pay-as-you-use in Cloud Computing terms. With this Cloud Computing, a smart phone can become an interface to large data center. Cloud Computing is extended form of distributed computing, parallel computing, and grid computing. Cloud Computing provides ubiquitous access to the content, without the hassle of keeping large storage and computing devices. Sharing large amount of media content is another feature that Cloud Computing provides. Cloud Computing recently has emerged and advanced rapidly as a promising as well as inevitable technology.

D. Fog Computing

The conventional Cloud Computing architecture is completely centralized in nature, and therefore, fails to provide real-time and low latency services to billions of IoT devices at the edge of the network, simultaneously. Cisco proposed a new computational paradigm, termed as Fog Computing, that subdues the shortcomings of Cloud Computing by transferring some of the core functions of cloud towards the network edge.

It should be reiterated that Fog Computing is not a substitute of Cloud Computing; rather anticipating the next generation IoT applications and their huge demand of real-time services, Fog Computing, in collaboration with the traditional Cloud Computing model, will serve as a greener computing platform.

III. RELATED WORK

Fog Computing is a new computational area and there are a lot of challenges and researches trying to find solutions for these challenges, The first question need to be answered is related to the suitability of Fog Computing in IoT and when to use. Authors of [3], [4] focus on analyzing the suitability of Fog Computing within the framework of IoT. The goal of these papers is to develop a mathematical model of Fog Computing and assess its applicability in the context of IoT where it is pivotal to meet the demands of the latency-sensitive applications running at the network-edge. The Fog Computing paradigm is modeled by mathematically characterizing the Fog Computing network in terms of power consumption, service latency, CO_2 emission, and cost, and evaluating its performance for an environment with high number of Internet-connected devices demanding real-time service. A case study is performed with traffic generated from the 100 highest populated cities being served by eight geographically distributed data centers. Results show that as the number of applications demanding real-time service increases, the Fog Computing paradigm outperforms traditional Cloud Computing. For an environment with 50% applications requesting for instantaneous, real-time services, the overall service latency for Fog Computing is noted to decrease by 50.09%. However, it is mention worthy that for an environment with less percentage of applications demanding for low-latency services, Fog Computing is observed to be an overhead compared to the traditional Cloud Computing. Therefore, the work shows that in the context of IoT, with high number of latency-sensitive applications Fog Computing outperforms Cloud Computing.

Authors of [5] performed an analysis of the trade off between accuracy and network traffic for the distributed learning solution based on the Hypothesis Transfer Learning framework. The key findings from this study is that a decentralized data analysis solution, along the line of a Fog Computing paradigm, is the most suitable one, as it

- drastically reduces the network traffic, and
- still guarantees a very high accuracy of the final data model.

Instead of training a model on the whole training set, multiple parallel models are trained on disjoint subsets, and then the partial models are combined to obtain a single final model. Results show that the accuracy of the solution does not significantly vary with the number of Data Collectors (DCs). This is quite interesting, as it shows that it is possible to use a completely decentralized solution where data analysis is done entirely on edge devices, according to a Fog Computing paradigm. Comparison with the two extreme cases (completely decentralized HTL and centralized cloud solutions, respectively) show some interesting features

- Reducing the number of DCs only provides a marginal advantage in terms of accuracy over a completely decentralized HTL solution.
- On the other hand, the cost in terms of network traffic does increase as the number of DCs decreases.

Authors of [6] focus on implementing a Fog Computing platform, which dynamically pushes programs to the devices. The programs pushed to the devices pre-process the data before transmitting them over the Internet, which reduces the network traffic and the load of data centers. The authors propose an efficient heuristic deployment algorithm to solve the deployment problem. Also implement an optimal algorithm for comparisons and conduct experiments with a real test-bed to evaluate the algorithms and Fog Computing platform. The proposed algorithm shows near-optimal performance, which only deviates from optimal algorithm by at most 2% in terms of satisfied requests. Moreover, the proposed algorithm runs in real-time, and is scalable. More precisely, it computes 1000 requests with 500 devices in < 2 seconds. Last, the implemented Fog Computing platform results in real-time deployment speed: it deploys 20 requests < 10 seconds.

Authors of [7] Design a management architectural pattern for adaptation system in Internet of Things based on modeling of control loops. With this pattern will be possible to design an architecture for self-adaptive systems in the IoT. Besides, the programmer can use that pattern in other IoT systems and worrying about only functional issues of such systems. Combine two patterns which are master/slave pattern and regional planning pattern. In the master/slave pattern, the control loop components are organized with one (centralized) master component that is responsible for the analysis and planning of control and multiple slave components that are responsible for monitoring and executing activities. The regional planning provides one component called regional planner for each region. The regional planner collects the necessary information from the managed system under its supervision to plan the adaptations. The regional planners interact with one another to coordinate the adaptations that span multiple regions.

The SWIFT architecture introduced by the authors of [8] in an earlier work provides a ubiquitous platform for seamless interaction of various smart objects, devices and systems, and hence may prove to be an ideal architecture to capture, process and assimilate information from Big data. In this paper the authors discuss issues related to implementation of SWIFT architecture for handling Big data for traffic management in smart cities. Various strategies to provide Big data solutions for smart traffic in terms of profiling traffic density, traffic signaling, managing the parking lots, smart navigation and monitoring vehicular pollution are discussed in the paper. Although the proposed model is very similar to idea behind the Fog Computing but the model doesn't consider the dynamic and mobility nature of the fog instances.

an advanced high level formalization—Performance Evaluation Process algebra (PEPA), a kind of stochastic process algebra (SPA), is adopted by the authors of [9] to model the process of real-time traffic status query in intelligent traffic system (ITS). Meanwhile, by using fluid approximation approach to analyze performance of the model and then the performance parameters of the system in practical application can be achieved accurately. This paper employs PEPA to model the process of real-time traffic status query in ITS. Because in real life, an efficient ITS is of great significance for the management of depth understanding the specific performance parameters of the system.

The related work proves the effectiveness and the importance of Fog Computing specially in IoT and shows that there are a lot of researches in this area with different perspectives, Some were concerned with the suitability of Fog Computing in IoT and showed that Fog Computing is very promising in IoT while other researches studied the impact of Fog Computing on network traffic and system performance.

But no management framework introduced to manage the fog instances (FIs) in the fog layer trying to well utilize the available resources to keep the computation as much as possible in the fog specially in critical situations to continue providing the advantages of Fog Computing and this is the purpose of this work.

As an example of critical situations when a FI is unable to serve a large number of requests or the FI is down for some reason or there is a heavy request that can’t be served by one FI. the case study introduced by this paper describes in details one of these critical situations and how the framework should act to handle such situation without going back to the cloud.

At the level of smart traffic systems, Fog Computing will be very promising as long as its possible to keep the computation in the fog layer. Some architectures like the SWIFT architecture introduced for handling big data for traffic management in smart cities but these architectures are not dynamic in nature like the Fog Computing.

IV. PROPOSED FRAMEWORK

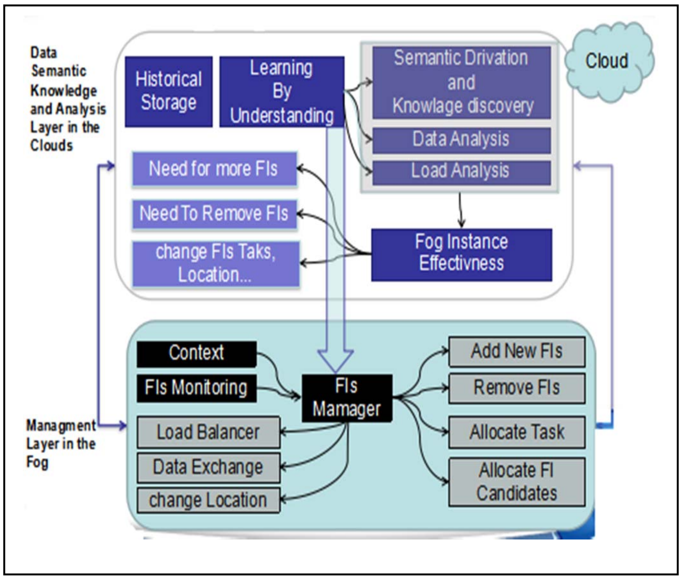


Fig. 1. A framework architecture for cognitive Fog Computing in IOT

It should be reiterated that the proposed framework is a generic framework for managing the fog layer in a any system not only the traffic control system which used as a case study in this paper.

The framework is fully independent from the managed system, it should be able to manage such system without any knowledge about the nature or the purpose of the system through integration points as will be discussed later. The framework provides a set of services and functions which are must for any systems using the Fog Computing for real-time services, so instead of re-implementing these function with each system this work provide an intelligent generic framework to provide such functions.

Fig 1. Describes the proposed framework which consists of two main layers.

- Data Semantic Knowledge and Analysis layer on the cloud;
- Management Layer on the Fog.

A. Data semantic Knowledge and Analysis layer

This layer resides on the cloud and it's main function is to store, analyze and learn from the managed fog instances (FIs) data like

- The number or requests served by the FI, the response time, the height load times…etc.
- The context data like the time, the location, the source of the served requests, …etc.
- The meta data of the FI like it's geographic location, the storage capabilities, the processing capabilities, …etc.

The output of this layer is to

i. Measure the effectiveness of each FI and decides if there is a need for a certain location in the network to add new, remove, changes the location, the capabilities or the task of some FIs.

ii. Learn which functions should be carried out in the Fog and which should be carried out in the cloud, The same function in some situations can be carried out in the fog while in other situations can be carried out in the cloud according to the context.

iii. Semantic derivation and knowledge discovery which instructs the system how to act in different situation.

So the first input required from the managed system is the performance data of the FIs, by analyzing these data the framework will be able to learn how to keep the computation in the fog layer as much as possible and will be able to propose solutions for handling critical situations as will be discussed in the case study, the required data not related to the nature of managed system but related to the performance of the fog layer and will be exchanged between the framework and the manages system through predefined data contracts.

As the framework propose solutions for different critical situations it should analyze the effectiveness of these solutions and enhance them for future similar situations.

B. Management Layer

This layer resides on the fog. It should be reiterated that this layer isn't a layer between the fog and the cloud layers of the managed system, but its another FIs in the fog layer to manage the FIs of the managed system.

This layer responsible for managing the FIs of the managed system to maximize the number of function that can be carried out in the fog.

This layer represents the system data collector and actuator which uses the knowledge gained by the Data Semantic Knowledge and Analysis layer to decide how to act in different situations. It consists of two types of components as FIs.

i. Monitoring Component, each monitoring FI is responsible for monitoring some FIs of the managed system and it should raise an event when some action happened, the actions and their related events are configured and predefined by the managed system for example when the response time of some FI exceeds 5 milliseconds then raise the event Response Delay with the FIs as input another example when there is no response from some FI raise the event Down FI with the FI as input. The monitoring FI raises requests periodically to the managed FIs to monitor their states and when an actions happened it raise the related event to the Managing FIs which will be described in the next section.

ii. Managing Component, each managing FI is responsible for managing some FIs of the managed system.

I. It should listen to the events raised by the monitoring FIs and act according to the action and the context in which the action happened.

II. Finding the proper solution for handling different situation specially critical situation to keep the computations on the fog, this is the key feature of the framework because if its not possible to handle the critical situations on the fog layer without going back to the cloud like managing the high load on FIs in this case the main advantages of using the Fog Computing are lost.

III. Decides carefully and quickly if there is a solution for the situation or going back to the cloud will be better for this situation.

IV. To propose a solution for some situations there is a wide range of choices to handle the situations for example.

- Change the task of some FIs for example make some FI works as a load balancer for the requests while another FIs works as a database engine for managing the storage and so on.
- Heavy requests can be executed in parallel across multiple FIs.
- Large number of requests can be divided into smaller sets and served in parallel across multiple FIs.
- Decide to execute some function on the cloud not in the fog.

V. The solution may be composed from one or more choices and its clear that there are a lot challenges with each choice for example in the case of task re-allocation or parallel processing the framework should decide which FI is the best one for the task without affecting the system, the solution shouldn't handle some situation and raise another problem as a result.

VI. The tasks of the FIs are handled as a software modules which are loaded to the FIs according to the task as propose by Hua-Jun Hong [6]. The algorithms of these modules are choose or defined by the managed system not the framework, they are black boxes for the framework, the framework just know which module for which task.

VII. The main goal of the managing component is how to choose the proper task for the proper FIs to handle the situation for example if it decided to make some FI as a load balancer it doesn't care how the load balancer module works or manages the request , it cares about what is task needed? and which FI should be selected for this task? and does this re-allocation solved the problem or not and why?.

VIII. So the details of the manged system and it's modules is completely hidden from the framework just the mapping between the tasks and the modules is required.

So composing the best solution for handling some situation and in a real-time fashion is the main challenge of this work, and to achieve this the framework should

I. Consider the knowledge gained from the analysis of the historical data of the FIs of the manages system, This will help to choose the best appropriate FI for some task without affecting the system for example having a knowledge about the times when some FI is loaded and when its not loaded helps to well utilize this FI in time in which its not loaded.

II. Consider the knowledge gained from analyzing the previously proposed solutions for different situations, this will help the framework to learn how to propose a better solutions.

III. After proposing some solution for some situation the framework should ensure that this solution solves the situation otherwise finds an alternative solution.

IV. Suggest adding, removing, changing location or changing tasks of some FIs of the manages system to avoid the occurance of some situations in the future .

V. Assessment for the performance of the software modules of the manages system to consider this in the composition of the solution, may be the problem is in the performance of the software modules of the managed system, this is a very important information for the managed system.

As stated earlier the proposed framework is generic so there are integration points with the managed system containing the following

i. Data contracts to exchange the data of the FIs of the manages system like the number of requests served , the response time …etc.

ii. Interfaces for managing the FIs of the managed system like changing the task of some FI.

iii. The way of communication between the framework and the managed system.

V. CASE STUDY :ACCIDENT SCENARIO

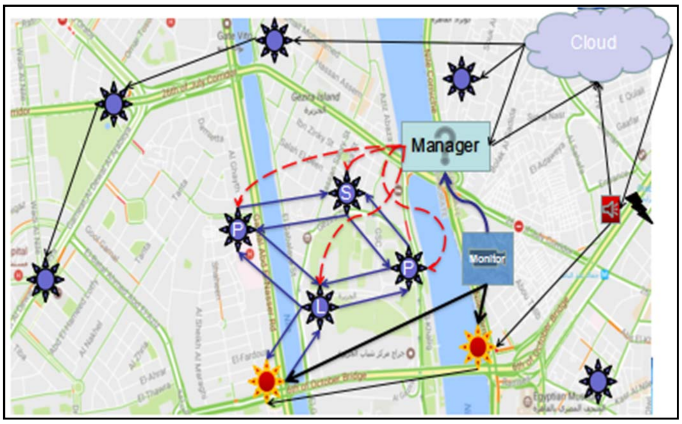


Fig. 2. Car accident scenario for certain region at Big Cairo Map

This section describes the case study which more clarify the different components of management layer and the role of each component, there are assumption that the Traffic Control System (TCS) can do the following.

i. The system can detect that some accident happened and able to determine the location of this accident.

ii. The system should stop the flow of the traffic to the location of the accident.

iii. The system has software modules to find alternative routes.

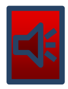
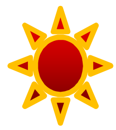
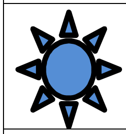
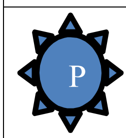
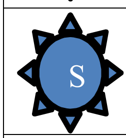
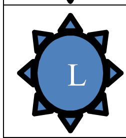
iv. The system has software modules for load balancing.

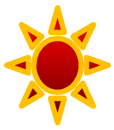
v. The system has software modules for data storage and retrieval.

vi. The system able to determine the FIs affected by the accident and which should stop the traffic flow to the accident location.

This section demonstrates how the proposed framework should act while an accident happened in some location as shown in fig 2. The different components of the figure are described in table 1.

TABLE I. CASE STUDY DESCRIPTION

Item	*Description*
	Accident happened in some location.
	Fog instance detected the accident, It should do the following I. Determine and report the nearest affected FIs to stop the traffic flow to the accident location. II. Report the accident to the cloud to report faraway FIs to give info about the traffic status in this area. III. Looking for the nearest ambulance and traffic police and find the best route to reach the the location.
	FIs affected by the accident, should stop the traffic flow to the accident location. So they will be very loaded by a huge amount of requests to find alternative routes which should be calculated in a real-time fashion.
	FIs not affected by the accident so they are less loaded.
	Monitoring unit detects highly loaded FIs and reports the manager with the loaded FIs.
	Manager Unit once reported with the loaded FIs it will try to handle the situation without depending on the cloud to proceed providing a real-time services for the applications depending on these FIs.It will look for near less loaded FIs and utilize them to handle the situation by reassigning new tasks for them using the program loader module. The manager will utilize the less loaded FIs as follows taking in consideration not affecting the areas these FIs serve.
	FIs Allocated for processing only by the manager to find alternative routes.
	FIs Allocated for storage only by the manager to keep the calculated routes to be reused instead of recalculating them for each request.
	FIs Allocated as a load balancer to balance the load on the processing FIs.

Item	*Description*
	Will not do any processing just pass the request to the load balancer FI and keep track of requests

Once an accident happened the TCS should stop the traffic flow to the location of the accident so it starts to determine the FIs which should stop the flow, once determined these FIs will be be very loaded and they will be requested to find alternative routes by a huge number of clients and it should responds in a real-time fashion to avoid any traffic problems.

Once the affected FIs becomes unable to handle the huge amount of requests the monitoring component should raise an event to the managing component with the FIs loaded.

The managing component once received the event from the monitoring component it will starts to handle the situation, first it should determine if it's possible to handle the situation in the fog layer or not, in case of not it will direct the loaded FIs to direct the requests to the cloud by loading the software module for this task on them.

For the managing component to be able to decide it should propose a solution for the situation and determine the resources for the solution so actually the fist step is to determine the solution that handle the situation efficiently then check if the required resource available or not.

In the accident scenario the managing component proposed a solution to distribute the requests across four FIs.

- Two FIs will be allocated for finding alternative routes.
- One FIs for storing the calculated routes to retrieve them if required again instead of re-calculating.
- One FIs will act as a load balancer to balance the load on the two FIs for finding the routes.
- The loaded FIs will just pass the requests to the load balancer and retrieve the response back to the client.

After proposing the solution the managing component will look for the resources required for the solution, it should select carefully the FIs that can execute the proposed solution without affecting the main tasks of these FIs.Selecting the appropriate FIs depends on the current state which can be obtained from the monitoring component and the knowledge gained from analyzing of the the historical data of the managed FIs.

After proposing and executing the solution the managing component must ensure the proposed solution solved the situation other wise it should find an alternative solutions.

All these steps should be performed in a real-time fashion without affecting the managed system to gain the benefices of the Fog Computing model.

VI. Simulation

Currently we are examining different simulators to decide which one will meet the research requirements, there are different candidates simulators like iFogSim which meets most of the requirements.

iFogSim enables modeling and simulation of Fog computing environments for evaluation of resource management and scheduling policies across edge and cloud resources under different scenarios. The simulator supports evaluation of resource management policies focusing on their impact on latency (timeliness), energy consumption, network congestion and operational costs. It simulates edge devices, cloud data centers, and network links to measure performance metrics. The major application model supported by iFogSim is the Sense-Process-Actuate model. In such models, sensors publish data to IoT networks, applications running on Fog devices subscribe to and process data coming from sensors, and finally insights obtained are translated to actions forwarded to actuators.

VII. Conclusion

The new framework will overcome most of the challenges which faced by previous research using cloud and Fog Computing. This framework must respond in a real-time fashion to support large number of devices, sensors and applications with low latency and real-time services. Also, should consider the decentralized nature of Fog Computing to be able take accurate decisions specially while serving a single heavy request in a parallel fashion through multiple fog instances.

References

[1] Qihui Wu, Guoru Ding, Shuo Feng, Zhiyong Du, Jinlong Wang and Keping Long. "Cognitive Internet of Things: A New Paradigm Beyond Connection." IEEE Internet of Things Journal (Volume: 1, Issue: 2, April 2014). IEEE,2014.

[2] Mohammad Aazam and Eui-Nam Huh. "Fog Computing and Smart Gateway Based Communication for Cloud of Things." Future Internet of Things and Cloud (FiCloud), 2014 International Conference on. IEEE,2014.

[3] Subhadeep Sarkar, Subarna Chatterjee and Sudip Misra. "Assessment of the Suitability of Fog Computing in the Context of Internet of Things." IEEE Transactions on Cloud Computing (Volume: PP, Issue: 99).IEEE,2015.

[4] Subhadeep Sarkar and Sudip Misra. "Theoretical modelling of Fog Computing:a green computing paradigm to support IoT applications." IET Networks (Volume: 5, Issue: 2, 3 2016). IET,2016.

[5] Lorenzo Valerio, Andrea Passarella and Marco Conti. "Accuracy vs. traffic trade-off of learning IoT data patterns at the edge with hypothesis transfer learning." Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2016 IEEE 2nd International Forum on. IEEE,2016.

[6] Hua-Jun Hong, Pei-Hsuan Tsai and Cheng-Hsin Hsu. "Dynamic module deployment in a fog computing platform." Network Operations and

Management Symposium (APNOMS), 2016 18th Asia-Pacific. IEEE,2016.

[7] Admilson de Ribamar Lima Ribeiro, Fernando Mendonça de Almeida, Edward David Moreno and Carlos A. E. Montesco. "A management architectural pattern for adaptation system in Internet of Things." Wireless Communications and Mobile Computing Conference (IWCMC), 2016 International. IEEE,2016.

[8] Satyanarayana V Nandury and Beneyaz A Begum. "Strategies to handle big data for traffic management in smart cities." Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on. IEEE,2016.

[9] Jie Ding, Rui Wang and Xiao Chen. "Performance modeling and evaluation of real-time traffic status query for intelligent traffic systems." Communications (APCC), 2016 22nd Asia-Pacific Conference on. IEEE,2016.

[10] Mohammad Aazam and Eui-Nam Huh. "Fog Computing and Smart Gateway Based Communication for Cloud of Things." Future Internet of Things and Cloud (FiCloud), 2014 International Conference on. IEEE,2014.