An Energy Efficient Workload Offloading in Fog Computing

Akshada Yogesh Joshi[1], Poonam Suresh Khanvilkar[2]

[1]Akshada Yogesh Joshi, Department of Master of Computer Applications, ASM IMCOST, Mumbai, India
[2]Poonam Suresh Khanvilkar, Department of Master of Computer Applications, ASM IMCOST, Mumbai, India

---***---

Abstract - *The billions of connected devices in IoT are generating huge amount of data every second. Ultimately there is a need to store, process, analyze the generated data efficiently. The data-center in cloud plays vital role in IoT. But the utility of cloud computing gets affected by some parameters like delay in response, bandwidth, energy consumption, security etc. To overcome such limitations and enhance QoS, the concept called "Fog Computing" has been evolved. In Fog Computing the data is managednear to the devices. The implementation of Fog computing reduces the reliance on cloud based platforms. The requests from devices which are time sensitive and require quick response are processed by fog whereas the heavy requests with high processing requirements are handle by cloud server. By fog offloading the data can be store, manage and compute near to the device locations where data is generated. Fog computing reduces the network traffic of cloud system, but the total electric energy consumed by fog nodes increases to process sensor data. This paper contains the solution and methodology that helps to reduce the energy consumption by fog nodes.*

Key Words*: *Cloud Computing, Fog Computing, IoT, Energy Efficiency

1. INTRODUCTION

In Internet of Things (IoT) huge numbers of devices are connected in a network. IoT is a paradigm of physical devices that featured with network connectivity that enables devices intercommunication among each other and exchange data. These devices perform some function in physical world and generate huge size of data. To gain insights from these data, it needs to be processed, analysed and stored in a manner to use it further as per need.

The devices that can be connected in network and able to exchange data over this network are called Things. The data generated by these devices is sent to cloud servers for further actions. It may get processed, stored or analysed to take further actions. IoT devices can be control remotely which improves efficiency and reliability. Cloud provides various services to users over the internet such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS).But due some factors like high latency, bandwidth, security, and network congestion it makes cloud as non-feasible solution for IoT.
The data generation rate and size by IoT devices is in multiple of bytes, it results in network congestion and the servers get overloaded due to heavy traffic from sensors. This makes cloud servers work inefficiently and delay in response time. It is disadvantageous to use cloud platform for time sensitive requests. It is important to respond them quickly and efficiently. To solve such problems the Fog layer was introduced.

Fog layer is the middle layer between devices and cloud. Fog computing is an extension and not replacement for cloud. Fog computing or fogging is an architecture that enables data computation, storage at the edge of network and then routed over internet. The main objectives of fog computing are reduces the amount of data sent to cloud, decrease network & internet latency, and improve system response time.

Fig -1 displays the fog computing architecture where IoT devices send data to the fog nodes and then fog nodes can get connected to cloud data centers. Most of the requests from devices are handled by fog nodes and only critical request are sent to the cloud for processing. This Architecture has three levels. At base level various devices are connected, sensors from these devices collect data about activities in device. At middle level, there are multiple fog nodes that process requests from devices and also perform routing. At top level, cloud servers have storage, and computation capabilities.

Fog computing creates a platform virtually that provides networking, storage and computation services close to the devices. It acts as a middleware; as most of data is handle by fog nodes only summarized information is transferred to cloud. This helps to reduce latency and saves bandwidth to a large extent.

If we compare between cloud and fog computing, the evolution of fog computing concept which is an extension for cloud has various benefits over cloud computing. Such benefits are localized information services, low latency, location awareness, low network bandwidth, more security.

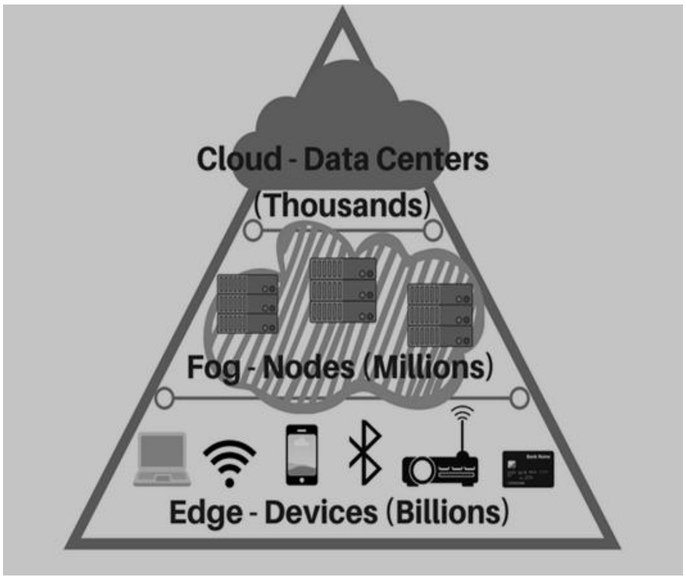

Fig -1: Fog Computing Architecture

Fog nodes interact with cloud server to provide better quality of service to end users. The paper has been divided into further more sections.

In Section 2, we overview related studies. In Section 3, we describe existing system. In Section 4, we review and describe the DTBFC model. In Section 5, we elaborate conclusion of this paper.

2. LITERATURE REVIEW

The Fog Computing has been evolved as a beneficiary solution to overcome the problems in cloud computing. Many researchers have done study on various aspects of fog computing. Although some areas that needs to be further studied and there is some related research which needs to be regarded.

Gollaprolu Harish et al. [1] proposed the IoT design victimization Fog Computing. They have done comparison between Fog and Cloud to know the various functioning capabilities in Fog computing. They have also explained the application, goals, challenges, benefits of fog computing.

Kun Ma et al. [2] proposed a multi-layer IoT-based fog computing model called IoT-FCM. This model uses a genetic algorithm for resource allocation between the terminal layer(devices) and fog layer and a multi-sink version of the least interference beaconing protocol (LIBP) called least interference multi-sink protocol (LIMP) to enhance the fault-tolerance/robustness and reduce energy consumption of a terminal layer.

Sindhu S, Dr. Saswati Mukherjee [4] in this work, emphasizes on importance of proper task scheduling in cloud computing. They have presented two scheduling algorithms for scheduling tasks taking into consideration their computational complexity and computing capacity of processing elements.

Shubha Brata Nath et al. [5] in their survey discuss the evolution of distributed computing from the utility computing to the fog computing, challenges for the development of fog computing environments, current status on fog computing research. They also focus on the architectures of fog computing systems, technologies for enabling fog, fog computing features, security and privacy of fog, the QoS parameters, applications of fog, and give critical insights of various works done on this domain. Lastly, discuss about different fog computing associations that closely work on the development of fog based platforms and services, and provide a thorough discussion on the future scopes and open research areas in fog computing.

Hasan Ali Khattak et al. [6] examined how utility of cloud and fog is affected by various parameters. More specifically, they have examined fog server utilization. They contribute to the capabilities of the iFogSim tool over the fog layer by balancing the load among the fog nodes. They have load balancing and latency in this work.

Mohammed Al-khafajiy et al. [7] proposed a fog computing architecture and framework to enhance QoS via request offloading method. The proposed work employ a collaboration strategy among fog nodes in order to permit data processing in a shared mode, hence satisfies QoS and serves largest number of IoT requests.

Raja Manish Singh et al. [9] have performed comparative study of the different algorithms of task scheduling problem in cloud computing for their suitability, feasibility, adaptability in the context of cloud scenario. They also try to propose the hybrid approach that can be adopted to enhance the existing platform further. So that it can facilitate cloud-providers to provide better quality of services.

The work in fog computing so far focuses on benefits of fog computing, reason for this evolution, solutions for emerging and possible problems in fog computing. In our work we have reviewed the possible solutions for low energy consumption by fog nodes that can additionally benefit the functioning of various fog nodes.

3. EXISTING SYSTEM

The concept of Cloud Computing was introduced to provide the computing services including storage, computation, analytics to remove the burden of on premise management of data. Fig – 2 shows main services provided by cloud platform i.e. Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS).

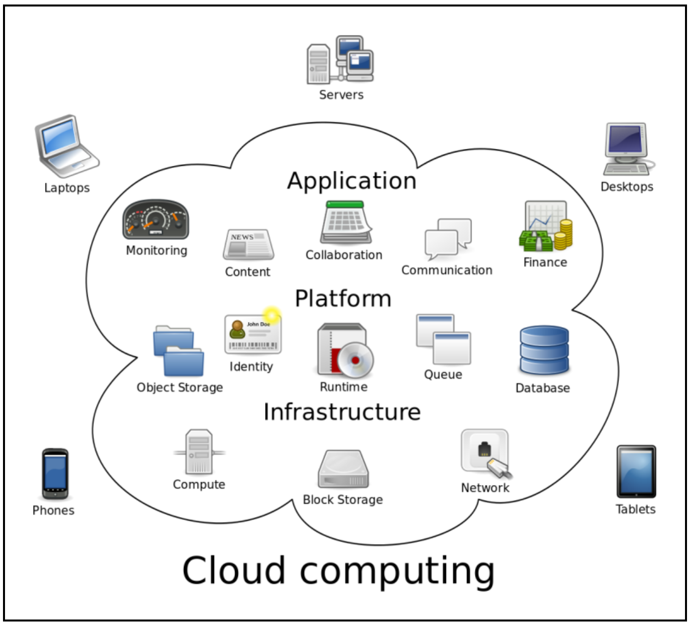

Fig -2: Services by Cloud Platform

IoT is something that generates incredible amount of data that places significant strain on infrastructure to handle it. Cloud computing with scalability and flexibility features can help with this. Even though IoT and Cloud Computing are different concepts, their collective implementation provides seamless services to users. Fig -3 shows how IoT and Cloud can get integrated.
The IoT is a concept deploying services that collect data from devices and transfer it to a particular location. The necessary actions can be taken on this data like sending alerts. IoT and cloud computing have complementary relationship. Thus integration of Cloud and IoT has increasing efficiency in our everyday tasks.

But this collaboration also has some limitations. As cloud computing is centralized structure, user is not aware about exact location of server. Due to long distance between user and respondent, the responses get delayed. As the numbers of servers are few, which results in network traffic, burden on servers, high energy consumption by servers.

Therefore to avoid such problems middle layer of fog nodes are used to efficiently work and respond to request. This increases quality of service and benefits the user.

4. PROPOSED SYSTEM

The evolution of Fog computing reduces the energy consumption by cloud servers. Since fog nodes perform computation on sensor data, they consume more energy. It is necessary to reduce energy consumption by not only servers but also by fog nodes. In this section we review and elaborate the technique to reduce energy consumption by fog nodes.

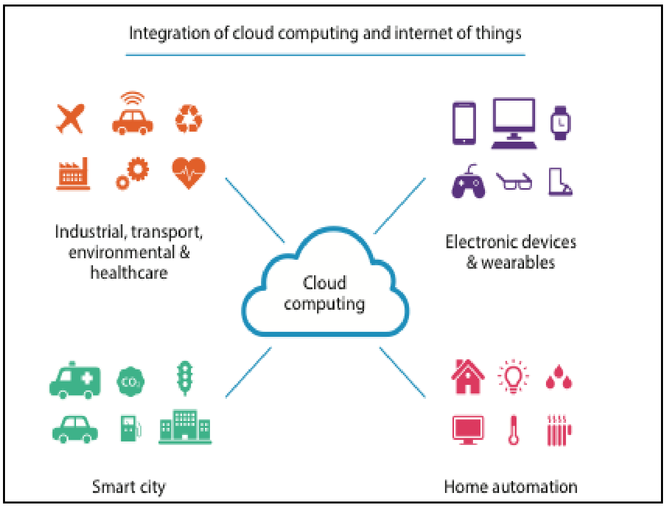

Fig -3: Integration of Cloud and IoT

The Dynamic Tree-based Fog Computing (DTBFC) model is beneficial to reduce energy consumption by fog nodes. In this model there is a root node which is cloud server denoted by 'f_0'. There is a hierarchy of fog nodes in tree based structure manner. The root node is Cloud server, whereas leaf node is edge node which collects the sensor data. Here we have considered k-ary tree of fog nodes with height h and every fog edge node is at (h-1) level. Every node receives input data from it's child node, process it and send output data to parent node. Likewise every fog node receives data from child fog node. In every fog node the input data is processed in addition to the routing of input and output data. The size of output data is less than input data. So output ratio = (size of output data/size of input data).

Fig -4 shows this hierarchy. So, input data for parent node is output data of child node.

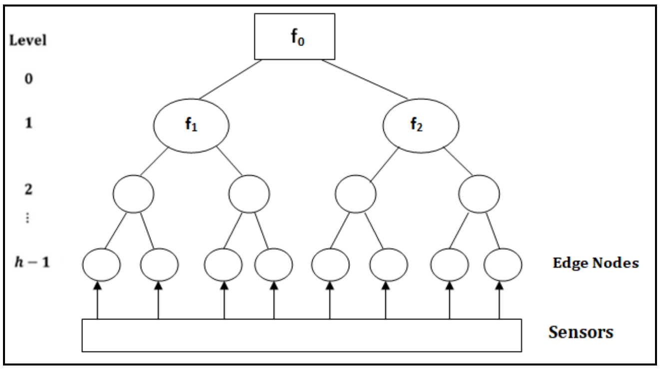

Fig -4: Hierarchy of Nodes

Every process P to handle data from sensors and to perform suitable actions is assumed to be a sequence of subprocesses $p_1,\ldots,p_m$ ($m \geq 1$). In this p_1 is the last subprocess and p_m is the first subprocess. p_m receives input data from sensors. After processing the input data subprocess p_m obtains the output data d_m. Then, subprocess p_m sends the output data d_m to the preceding subprocess p_m-1. So each subprocess receives input data $d_i + 1$ from a succeeding subprocess p_i+1 and sends output data d_i to a preceding subprocess p_i-1.

Subsequences $p_1,..., p_i$ and $p_j,..., p_m$ ($1 \le i, j \le m$) are a prefix and postfix of a sequence $p_1,....,p_m$. Initially all subprocesses $p_1,....,p_m$ are supported by one node f. Sensor sends the data to root fog node f which was processed by subprocesses $p_1,....,p_m$. As the amount of data received from it's child node increases, it gets heavily loaded. To solve this problem we have two ways i.e. Splitting and Replicating. Fig – 5 shows the replication and splitting process.

When node f gets heavily loaded, the subprocesses handle by it $p_1,....,p_m$ are splitted into two nodes f and f_1. The node f_1 is child node of f. The prefix subsequence $p_1,..., p_i$ ($i \le m$) is supported by node f and postfix subsequence $p_j,..., p_m$ is supported by node f_1. The node f_1 receives all the data from the child nodes and processes the data by the subprocess p_j, ..., p_m. The f_1 node then sends the output data d_j of the subprocess p_j to the node f. In this way the workload of node f gets divided into two nodes. If the amount of input data to nodes f and f_1 decreases then then f_1 gets again merged with f node.

In second way, if node f_1 gets heavily loaded and unable to handle data, then it is replicated to node f_2. The nodes f_1 and f_2 handle the same subsequence of $p_j,..., p_m$. The child nodes will send data to any one of the node f_1 and f_2. The nodes f_1, f_2 send output data to node f. The nodes f_1, f_2 are child nodes of node f. If the amount data to be processed by the nodes f_1, f_2 decreases, so that one node can handle data, the one of node, say f_2 gets dropped and all further data is processed by f_1.

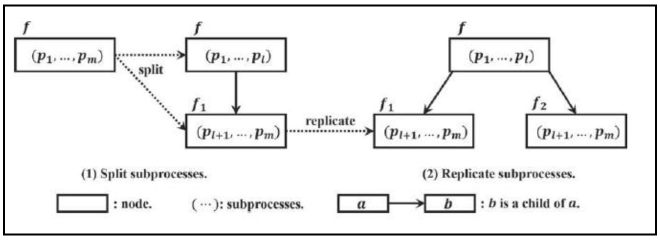

Fig -5: Splitting and Replication

4.1 Algorithm of splitting and replicating nodes

An application process P has subprocesses $p_1,....,p_m$. Suppose a node f_R supports a subsequence $SP(f_R) = (p_{sR}, p_{sR+1}, ..., p_{eR})$ of subprocesses of application process P. To minimize load of f_R, we assume that it has splitted and replicated to nodes f_R, $f_{R1}, ..., f_{R,cR}$. Now the node f_R supports the subprocesses (p_{sR}, ..., p_l) and child nodes after split and replica supports further subsquence ($p_{l+1}, ..., p_{eR}$). Fig – 6 shows that how the nodes f_R, f_{R1}..., $f_{R,cR}$ consumes energy to receive, process and sent data. Table -1 shows the list of abbreviations.

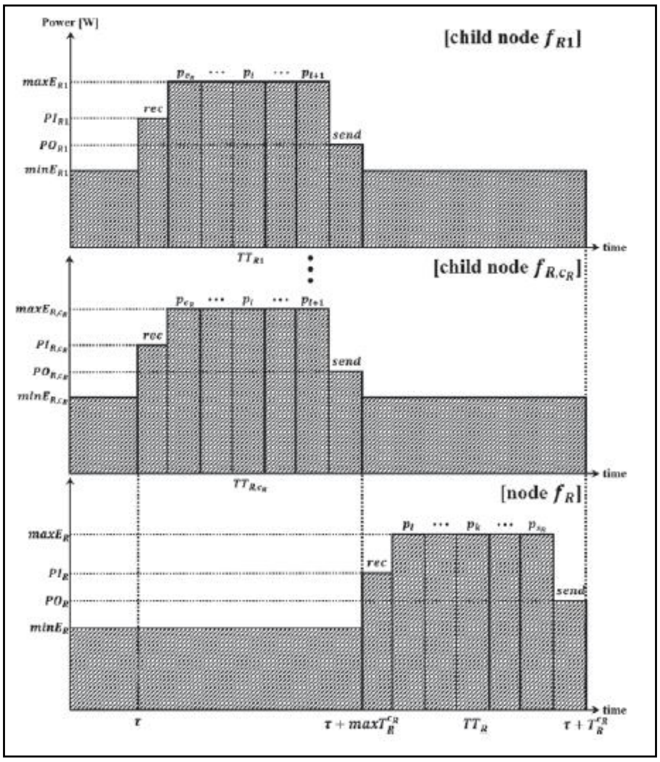

Fig -6: Energy Consumption by child and parent node

First, at time τ, every child node f_{Ri} receives the input data d_{Ri}. The node f_{Ri} consumes the power PI_{Ri} [W] for $TI_{Ri}(x_i)$ [sec]. Then subprocesses are performed by these child nodes, and it consumes power $maxE_{Ri}$ [W] to perform subprocesses for TC_{Ri} (p_{l+1}, ..., p_{eR}). And while sending output for $TO_{Ri}(\rho'_R \cdot x)$ [sec] it's power consumption is PO_{Ri} [W]. Before receiving input all these nodes consumes minE [W] power.

The Total execution time taken by each child fog node TT_{Ri} is addition of $TI_{Ri}(x_i) + TC_{Ri}((p_{l+1}, ..., p_{eR}), x_i) + TO_{Ri}(\rho'_R \cdot x_i)$ [sec]. And total energy consumption (EF) by these child nodes to perform subprocesses is addition of energy consumed by node while receiving (EI), computation(EC) and sending (EO) data. Let maxT be the largest execution time taken by child nodes to process the subprocesses (p_{l+1}, ..., p_{eR}). So, the parent node f_R starts taking input at time τ + maxT and at this point consumes minE [W] power. So energy consumption by parent node f_R is maxT · minE [W]. Then node f_R starts processing the input data. The response time ST and total energy consumption SE of node f_R and child nodes f_{R1}, ..., $f_{R,cR}$ are given as follows :

$$\mathbf{ST} = maxT + TT_R((p_{sR}, ..., p_l), \rho'_R \cdot x)$$

$$\mathbf{SE} = \sum_{i=0}^{cR} [EFRi((pl+1, \ldots, peR), xi) + \{maxT - TT_{Ri}((p_{l+1}, ..., p_{eR}), x_i) + TT_R((p_{sR}, ..., p_l), \rho'_R \cdot x)\} \cdot minE_{Ri}] + (EF_R((p_{sR}, ..., p_l), \rho'_R \cdot x) + maxT \cdot minE_R)$$

Whereas if node f_R supports all the subprocesses (p_{sR}, p_{sR+1}, ..., p_{eR}) then time ST_R energy consumption SE_R for time ST_R of node f_R is as follows :

$$\mathbf{ST_R} = TT_R((p_{sR}, ..., p_{eR}), x)$$
$$\mathbf{SE_R} = EF_R((p_{sR}, ..., p_{eR}), x)$$

Table -1: List of Abbreviations

List of Abbreviations	
x	size of data
rec	Energy consumed by f_{Ri} to receive input data
send	Energy consumed by f_{Ri} to send output data
PI	Power consumption while taking input
PO	Power consumption while sending output
maxE	Maximum Power consumption
TI	Total time to take input
TO	Total time to send output
TC	Total time to process/compute data
TT = TI +TC +TO	Total time taken
EI	Energy consumption to take input
EO	Energy consumption to send output
EC	Energy consumption to compute data
EF	Total energy consumption
ρ'_R	Output ratio

So when a single fog node is capable to support all subprocesses, it handles it solely. When the fog node gets heavily loaded, it gets splitted and replicated and after processing data child nodes may get dropped. Therefore by implementing this model, as every node supports a small amount of data it consumes less energy.

5. CONCLUSION

IoT is composed of various connected devices that generate data which needs to be processed. Fog computing was introduced to reduce the burden on cloud servers. Through fog computing only limited amount of data is placed in cloud for processing. In this paper, in order to implement fog model efficiently we reviewed and describe the Dynamic Tree-based Fog Computing (DTBFC) model. In this model the fog nodes are dynamically added and dropped as per need to utilize the resources. Therefore, it reduces the energy consumption of fog nodes while data handling.

REFERENCES

[1] Gollaprolu Harish, S.Nagaraju, Basavoju Harish, Mazeeda Shaik, "A Review on Fog Computing and its Applications", IJITEE,Volume-8, Issue-6C2, April 2019.
[2] Kun Ma, Antoine Bagula , Clement Nyirenda and Olasupo Ajayi, "An IoT-Based Fog Computing Model", June 2019.
[3] Hatem A. Alharbi, Taisir E.H. Elgorashi and Jaafar M.H. Elmirghani, "Energy Efficient Cloud-Fog Architecture", School of Electronic and Electrical Engineering, University of Leeds, LS2 9JT, United Kingdom.
[4] Sindhu S, Dr.Saswati Mukherjee, "Efficient Task Scheduling Algorithms for Cloud Computing Environment",January 2011.
[5] Shubha Brata Nath, Harshit Gupta, Sandip Chakraborty, Soumya K Ghosh, "A Survey of Fog Computing and Communication: Current Researches and Future Directions", April 2018.
[6] Hasan Ali Khattak, Hafsa Arshad, Saif ul Islam, Ghufran Ahmed, Sohail Jabbar, Abdullahi Mohamud Sharif and Shehzad Khalid, "Utilization and load balancing in fog servers for health applications",2019.
[7] Mohammed Al-khafajiy, Thar Baker, Atif Waraich, Dhiya Al-Jumeily, Abir Hussain, "IoT-*Fog* Optimal Workload via Fog Offloading", December 2018.
[8] Ryuji Oma, Shigenari Nakamur, Dilawaer Duolikun, Tomoya Enokido , Makoto Takizawa, " An energy-efficient model for fog computing in the Internet of Things (IoT)", 2018.
[9] Raja Manish Singh, Sanchita Paul, Abhishek Kumar, "Task Scheduling in Cloud Computing: Review", IJCSIT, Vol. 5 (6), 2014.
[10] Babur Hayat Malik, Muhammad Nauman Ali, Sheraz Yousaf, Mudassar Mehmood, Hammad Saleem, "Efficient Energy Utilization in Cloud Fog Environment", IJACSA, Vol. 10, No. 4, 2019.
[11] Thai T. Vu, Diep N. Nguyen, Dinh Thai Hoang, Eryk Dutkiewicz, Thuy V. Nguyen, "Optimal Energy Efficiency with Delay Constraints forMulti-layer Cooperative Fog Computing Networks", March 2020.
[12] Ehsan Ahvar, Anne-Cécile Orgerie, Adrien Lebre. Estimating Energy Consumption of Cloud, Fogand Edge Computing Infrastructures.IEEE Transactions on Sustainable Computing, IEEE, 2019.
[13] Fatemeh Jalali, "Energy Consumption of Cloud Computing and Fog Computing Applications", July 2015.
[14] Eric Bernardes C. Barros, Dionísio Machado L. Filho, Bruno Guazzelli Batista,Bruno Tardiole Kuehne and Maycon Leone M. Peixoto, "Fog Computing Model to Orchestrate theConsumption and Production of Energy in Microgrids", June 2019.
[15] Ryuji Oma, Shigenari Nakamura, Tomoya Enokido, and Makoto Takizawa, "A Dynamic Tree-Based Fog Computing (DTBFC) Model for the Energy-Efficient IoT", 2020.
[16] Fig -1 :"Fog Computing and Edge Computing: What You Need to Know", power-solutions.com , Url : https://www.power-solutions.com/industry-trends-

best-practices/industry-trends/fog-computing-and-edge-computing-what-you-need-to-know.
[17] Fig -2: "cloud computing", wikipedia.org, Url: https://en.wikipedia.org/wiki/Cloud_computing
[18] Fig – 3: "Integration of Internet of Things(IoT) and Cloud Computing Privacy Concerns and Possible solutions" internetinitiative.ieee.org, Url: https://internetinitiative.ieee.org/newsletter/september-2018/integration-of-internet-of-things-iot-and-cloud-computing-privacy-concerns-and-possible-solutions