Mathematical Model of Fog Computing Architecture for IoT Micro-Services

Prasenjit Maiti, Jaya Shukla, Bibhudatta Sahoo, and Ashok Kumar Turuk

National Institute of Technology,
Rourkela, India
{pmaiti1287,jayashukla2192,bibhudatta.sahoo}@gmail.com,akturuk@nitrkl.ac.in

Abstract. Fog computing provides a distributed data capturing, processing and resource allocation approach in the Internet of Things (IoT). The Quality of Service (QoS) requirements of IoT services are service delay, energy consumption, resource provisioning, bandwidth utilization. We design a fog computing architecture based on QoS parameter. We represent the distributed Cloud-IoT solution where data optimally distributed among mini-clouds/Fog nodes. The processing of IoT traffic is taken care of by Virtual Machines(VMs) facilitated by distributed mini-clouds/Fog nodes and located within the edge devices. However, there has been little research on design a QoS aware fog computing architecture. We propose a mathematical formulation of fog computing model. We model for performance analysis of a system based on QoS metrics.

Keywords: Fog Computing; Quality of Services; Cloud; energy consumption-delay tradeoff

1 Introduction

The Internet of Things (IoT) depict a major change in data management. Real-time data management is associated with distributed objects and their associated smart sensors. Smart sensors data needs to be stored and retrieved efficiently on demand for IoT services. IoT devices are growing rapidly and it is anticipated that about 50 billion devices will be deployed in 2020. Existing cloud solution provides services for a large amount of data. But in some scenarios it can face limitations due to increased traffic of the entire network and thus delay in processing the services. Different IoT services such as Health-care, Face Recognition, Military, disaster management require real-time response with very low latency. To overcome this problem, a new architecture needs to be proposed and thus, fog computing emerged to take care of these challenges. Fog computing is a concept that provide services at the network edge and involves smart Gateways name Fog Smart Gateways(FSG). Fog nodes are deployed in the network near the users to handle the services. In this architecture the data is processed locally before sending it to the cloud. The major issues and challenges of architecture design for edge-centric IoT services are discovering fog nodes, data caching, partitioning and offloading tasks using fog nodes publicly and securely. Due to constrained resources , resource discovery and resource allocation are the challenges of architecture design which meets the Quality of Service (QoS) and Service Level Agreement (SLA). Fog node controller play a major role for identify the best candidate for placement a fog node and IoT services according to the resource availability. Sensor data virtualization is a requirement for IoT services that allows an application to retrieve and manipulate data with live data discovery and monitoring. Service allocation or service scheduling of all services to minimize the delay for each service allocation is done by considering both resource availability and the devices condition. To provide an efficient service, service nodes placement, service nodes selection, and service placement with a balanced and efficient pairing or matching strategy in a sensor-virtualization environment for edge nodes which is crucial for achieving Service Level Agreement (SLA) and Quality of Services (QoS). Containers handle processing of IoT traffic hosted by distributed fog nodes.Only a few literature are available for edge centric architecture design for IoT services with above parameters. There is a need to develop an efficient architecture that reduces service response time, overall network traffic and is economically feasible.

We organized the rest of the paper as follows. Prerequisites for our work are in Section-2 as related work. Section-3 describes the details of fog computing architecture and mathematical model. Section-4 presents QoS metrics of fog computing architecture. We have concluded our work of this paper in Section-5.

2 Related Work

A Sensor-Fog architecture provides a platform to the users to easily provide services. Bonomi et al. [1] proposed that fog computing is a distributed, highly virtualized platform that provides compute, storage and networking services. A fog network is a collection of fog nodes and each fog node resides in any edge devices like router, switch, base station, access point, gateway or smart phone. Many researchers proposed a three-layer architecture, consisting of cloud data centers, fog nodes at the edge of the network and devices as end points. Depending on the definition, the fog nodes that may be router in the core network, switch in the WAN, and even wireless Access Points(APs) and smart phones are included in the fog layer. The smart gateway is proposed as a fog node in [2], [3], the micro data centers proposed in [4], or the proposal of fog nodes serving as caches in Information Centric Networking in [5]. Fog nodes as mini-clouds proposed by [6], [7]. Now, the key aspect is where fog nodes are located. Some of the authors[2], [3], [4], [8] proposed to locate fog nodes in highly capable devices, such as routers or smart gateways. Bonomi et al. [9], [10], [11] proposed intermediate compute nodes as fog nodes which has no dependency on specific devices. Tang et al. [7] proposed the use of three fog computing layers for big data analysis in smart cities. In a different application [12], fog computing was applied in the industrial environment. In the paper, fog computing had been implemented in Cisco edge routers, as it was first proposed by Cisco in [13]. Abdullahi et al. [5]

and Skala et al. [6] proposed routers as a candidate for deployment of fog nodes. In [14], an interesting example was given of sharing smart phone computation resources only when phones are connected to the grid and it is not enough to handle highly demanding scenarios. In the Fog to Cloud (F2C) scenario, whenever a node in the IoT layer submits a job , the node first contacts the fog it is connected with, instead of submitting the job directly to the cloud. The fog then decides whether to undertake the job itself or forward it to the cloud for processing. The fog has lower storage and processing capabilities than the cloud, it might at times refuse to undertake new jobs if it had reached its maximum capacity, hence the concept of fog availability is introduced in [15]. In this paper author discussed that fogs are not limited anymore to either execute a task or forward it to the cloud, but also to communicate with other fogs to process the job request. This minimizes the overall network delay. Kumar et al. [16] developed a distributed cloud data centers, called mini-clouds, among which data can be replicated. Masip-Bruin et al. [17] introduced a Fog-to-Cloud architecture which consists of a layered management structure that integrates different heterogeneous fog layers into hierarchal architecture. A management system responsible for discovering a set of available fogs is developed to select the best fog available to meet certain service requirements but authors have not explained the criteria used by the management system in choosing the optimal fog. Souza et al. [18] proposed a QoS- aware service distribution strategy in Fog-to-Cloud scenarios. The work aims at achieving low delay on service allocation by using service atomization in which services are decomposed into distinct sub-services called atomic services to enable parallel execution. A control plane within the Fog-to-Cloud architecture exists that is responsible for distribution of the atomic services among the available edge nodes. The service allocation process aims at reducing the service allocation delay, providing load balance and energy-usage balance among the distinct fogs. Although there exists extensive research in the area of cloud resource sharing, work in fog resource sharing and cooperation is still premature.It is quite natural that the service latency is drastically reduced as compared to the service processing using the cloud when fog computing is applied, but it is possible to further reduce the service response time depending on which nodes is deployed as fog nodes. Also, the network traffic can be reduced according to the deployment position of the fog node. Narendra et al. [19] proposed optimal mini-cloud placement to minimize latency of data collection from IoT devices; and data migration among mini-clouds with a view towards addressing storage capacity issues while minimizing access latency. Malandrino et al. [20] work shows high server utilization and low application latency, but the best approach depends on the individual network operators deployment strategy and geographic specifications of the cities. Most of the research issues in fog computing [21],[22],[23], [24], the service latency , network traffic and power consumption are reduced by fog computing architecture. But it is conceptual to deploy the fog nodes near the user only , and there is no consideration what devices actually fog server should be deployed on. Luan et al. [25] and Hong et al. [26] described the concept of fog computing as mobile fog and showed that mobile users can use fog computing to improve QoS, reduce bandwidth and energy consumption, end to end delay and network traffic. But, there is no consideration where to deploy or place the fog nodes. A fog node has capability to run multiple Virtual Machines (VMs) on its own physical machine. The VMs can be flexible placed in fog network, based on the traffic distribution and moving pattern of mobile users. Dynamic fog node placement in fog network systems incurs a significant cost on latency, energy and bandwidth consumption of the network links.

3 Fog Computing Architecture and Mathematical Model

In our architecture (Figure 1) , the IoT service network consists of four layers. The networking elements of the architecture perform the tasks of data aggregation and processing of the traffic produced by IoT devices.

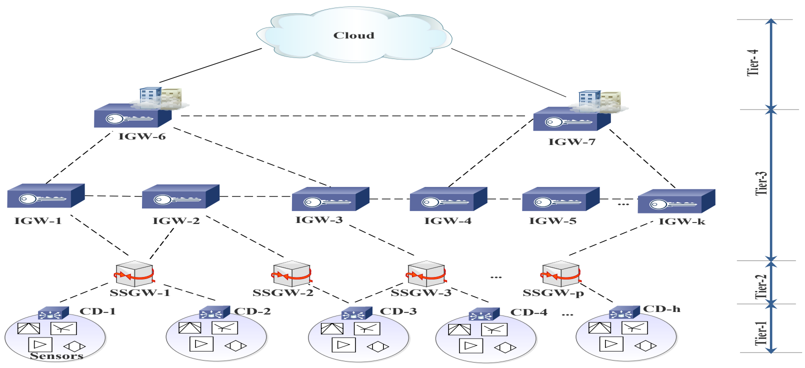

Fig. 1. Edge Centric Architecture for IoT Services

(a) Tier 1: This is the ground-level layer encompassing all the smart sensor nodes (SSNs) that are assigned unique IPv6 addresses, suitably compressed according to the 6LoWPAN protocol and form a mesh network. SSN is a collection of sensors and actuators. SSN sense environmental data and send to the upper layer. There can be instructions from the upper layer to the actuator to perform an action. IoT devices or IoT nodes is a collection of SSN (mobile phones,smart vehicles, and smart meters etc.). SSN are distributed uniformly at random . A coordinate value is assigned to each SSN. We assume the transmission range as a circle of SSN in a smart city scenario. A typical smart city scenario has hundreds of networks, pertaining to different domains , deployed all over its geographical area. Each of these networks is coordinated by a Coordinating Device(CD). A CD is known differently in different networks namely Cluster Head(CH) in sensor networks, Access Point(AP) in WiFi networks and Reader in Radio-Frequency Identification(RFID) network etc. S is a set of n number of static smart sensor nodes distributed uniformly at random in the area of $p \times q$. A SSN denoted by $\chi \in S$, is defined as a eleven-tuple.

$$\chi =< S_{id}, S_{st}, S_{\max\_v}, S_{\min\_v}, S_{ts}, S_e, S_l, S_h, S_{cid}, S_k, S_B >$$

S_{id} is representing the unique IP address of the sensor. A status of a SSN, S_{st} is represented by a boolean value $S_{st} = \{0, 1\}$, which defines the sensor node is in active state(value 0) or dead state (value 1). The maximum and minimum value that sensor can return(in the appropriate SI unit)is described by $S_{\max\_v}$ and $S_{\min\_v}$. S_{ts} represents time in millisecond when the sensor value sends. One SSN can sense multiple environment events. S_e and S_l indicate the type of event and node location of a SSN. The tuple S_h express the specifications of a SSN which includes its hardware details. S_k represents the encryption key. S_B represents the battery level of a SSN. If the value of S_B is less than the threshold value then SSN close its communication. S_{cid} indicates the cluster id of a SSN. SSNs send data to the CD. CD is a set of coordinating devices. Each CD have a certain transmission range r_{c_d}. The IoT network consists of different technologies(RFID,ZigBee,Bluetooth,BLE,802.11 a/g ,WiFi etc.). TL is a set of technologies which are associated with CDs.

(b) Tier 2: CDs need to transmit their data to the Internet for efficient execution of their corresponding applications. This transmission of data is facilated by device known as Solution Specific Gateways($SSGW$) or IoT Gateway(IGW) . $SSGW$ is a set of solution specific gateway and G is a set of IoT gateway (IGW) . r_{sg} is a range of each $SSGW$. CDs can only communicate through one specific technology and are connected to at least one SSGW/IGW. However , an SSGW is a wireless device which supports technologies of all the CDs associated with it.Two SSGW to be connected if and only if they are in each other's range and support at least one mutually common technology, else, they are connected through an IGW. $SSGW$s route the data received from CDs associated with them to the IGWs. The $SSGW$ should also ensure the coverage of the CDs. Wireless Mesh Network is as close as it can get to the IoT network with one fundamental difference. All gateway in a wireless mesh network support the same set of technologies whereas $SSGW$ in IoT support different sets of technologies. Each IGW has a wired connection to the Internet and sends the data received from the $SSGW$s to the upper layer. $SSGW$ is describe by three typle.

$$SG =< SG_{id}, SG_{d[m]}, SG_{sp} >$$

Each $SSGW$ has unique IP address. SG_{id} represents ID of a $SSGW$. Each SSGW maintains a neighbor list that records the connected CDs. $SG_{d[m]}$ is non-empty 1D array of size m which stores the IDs of connected CDs. SG_{sp} dictates the hardware specification like processor,memory, wireless communication technologies involved such as RFID,ZigBee,Bluetooth,WiFi etc.

(c) Tier 3: This layer consists of set of edge devices, such as gateway, router, switch etc. A Fog Node(FN) is placed within an IGW specific to a geographic location and other IGW is served by the co-located FN. FNs are capable of load balancing and service orchestration. A Fog device , F_d , represent by the three tuple. Each fog device id F_{id} is different from each other. The type (such as gateway, router,switch etc.) of the fog computing device represented by F_{type}. F_{sp} is the hardware specification of the device. An IoT service SE_{IoT} is defined as a four-tuple.The service is the main motivation of the IoT.

$$SE_{IoT}{=}< SE_{id}, SE_{type}, SE_{sp}, SE_{req} >$$

The identification must be unique for each instance of the IoT service.$SE_{id}$is the service ID. The IoT service has to specify its purpose, i.e, what its functionality is or what the responsibilities of this service. Service functionality is a set of operations that can be provided to the test of the IoT services.SE_{type} type denote the purpose for which the service is used (such as medical, education, finance,entertainment, utility, and gaming). SE_{sp} manages the base framework particulars that are required to run the application including the processor, memory, and operating system. SE_{req} is the resource requirement(such as bandwidth, computation capability, storage) to run a service. An application A is represent as a three-tuple which is running at the end of a user.

$$A =< A_{id,} A_{sreq} >$$

A_{id} is the unique ID of an application.REQ is a set of requests from the users. A_{sreq} is an user request for a particular service.

(d) Tier 4: The cloud computing layer is top-most layer. A data center has several physical servers and there is an interconnection of high speed LAN-network and high bandwidth link to the Internet from each physical server. Each IGW conneted to a cloud data center by a wired network. The cloud computing environment is with number of heterogeneous physical hosts in a data center.

4 QoS Metrics

4.1 Service latency

The service delay is the requested transmission delay and processing delay. We assume that the communication delay between SSNs is considered insignificant. Let $\Delta_{cd\_}sg$ and Δ_{sg_igw}, Δ_{igw_sfg} be the delays in transmission of a data packet from a CD to the corresponding SSGW, from a SSGW to the corresponding IGW, and from IGW to a smart fog gateway respectively. η_{sg}, η_{igw}, and η_{sfg} are the processing latency of SSGW, IGW and smart fog gateway for a data packet. Thus, the mean transmission latency, σ_{sfg}, for the data packets of req_i request running within mc_i is given by

$$\sigma_{sfg} = (\Delta_{cd\_sg}\mu + \Delta_{sg\_igw}\theta + \Delta_{igw\_sfg}\tau) + (\eta_{sg}\mu + \eta_{igw}\theta + \eta_{sfg}\tau) \quad (1)$$

where, μ , θ, and τ ($\mu > \theta > \tau$) are the total number of packets sent by CD, SSGW , and IGW.

4.2 Energy consumption

Since the energy consumed by the data from CD to SSGW and from the SSGW to IGW is represented by λ_{cd_sg} and λ_{sg_igw}, respectively.λ_{igw_sfg} is the energy expansion from the IGW to the intelligent fog gateway for unit byte data transfers. The energy demand to process unit byte of data within the SSGW ,gateway, and smart fog gateway are represented by ω_{sg}, ω_{igw}, and ω_{sfg} respectively. Total energy consumption of a data packet is transmission energy and processing energy of a data packet. The rate of energy dissipation of a data packet represent as

$$\xi(t) = \left(\begin{array}{l} \left(\lambda_{cd\_sg} \sum_{i=1}^{h} \sum_{j=1}^{p} \gamma_{i,j} + \lambda_{sg\_igw} \sum_{i=1}^{p} \sum_{j=1}^{k} \alpha_{i,j} + \lambda_{igw\_sfg} \sum_{i=1}^{k} \sum_{j=1}^{t} \beta_{i,j} \right) + \\ \left(\omega_{sg} \sum_{i=1}^{h} \sum_{j=1}^{p} \gamma_{i,j} + \omega_{igw} \sum_{i=1}^{p} \sum_{j=1}^{k} \alpha_{i,j} + \omega_{sfg} \sum_{i=1}^{k} \sum_{j=1}^{t} \beta_{i,j} \right) \end{array} \right) \tag{2}$$

where $\gamma_{i,j}$, $\alpha_{i,j}$ and $\beta_{i,j}$ $(\gamma_{i,j} > \alpha_{i,j} > \beta_{i,j})$ be the total number of bytes being transmitted from cd_i to sg_j, sg_i to igw_j, and igw_i to sfg_j at time t.

5 Conclusion and Future Work

With the rapid growth of IoT services; service management, QoS, SLA is becoming a critical issues. Efficient and in time scheduling and management of resources, minimum energy consumption and service latency not only allows Fog nodes to perform according to the situations, but also, helps customer satisfaction. In this paper, we have presented a model for reduce service latency and energy consumption through Fog computing. Our future work involve design an architecture based on efficient resource utilization , service composition and orchestration , and sensor virtualization environment.

Acknowledgment

This work has been supported by Media Lab Asia (Visvesvaraya Ph.D.Scheme for Electronics and IT, Project Code-CS-VSE) under the department of MeitY government of India. The work is done at Department of Computer Science and Engineering , NIT Rourkela, India.

References

1. Bonomi, F., Milito, R., Zhu, J., Addepalli, S. : Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, pp. 13-16, ACM, Helsinki, Finland (2012). doi:10.1145/2342509.2342513

2. Aazam, M., Huh, E. N. : Fog Computing and Smart Gateway Based Communication for Cloud of Things. In: IEEE International Conference on Future Internet of Things and Cloud, pp. 464-470, IEEE, Barcelona (2014). doi: 10.1109/Fi-Cloud.2014.83

3. Gia, T. N., Jiang, M., Rahmani, A. M., Westerlund, T., Liljeberg, P., Tenhunen, H. : Fog Computing in Health-care Internet of Things: A Case Study on ECG Feature Extraction. In: IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, pp. 356-363, IEEE, Liverpool (2015). doi: 10.1109/CIT/IUCC/DASC/PICOM.2015.51

4. Aazam, M., Huh, E. N. : Fog Computing Micro Datacenter Based Dynamic Resource Estimation and Pricing Model for IoT. In: IEEE 29th International Conference on Advanced Information Networking and Applications, pp. 687-694 , IEEE, Gwangiu (2015). doi: 10.1109/AINA.2015.254

5. Abdullahi, I., Arif, S., Hassan, S. : Ubiquitous Shift with Information Centric Network Caching Using Fog Computing. In: Computational Intelligence in Information Systems. Advances in Intelligent Systems and Computing, vol. 331, pp. 327-335 , Springer (2014). doi: org/10.1007/978-3-319-13153-5-32

6. Skala, K., Davidovic, D., Afghan, E., Sojat, Z. : Scalable Distributed Computing Hierarchy: Cloud, Fog and Dew Computing. In: Open Journal of Cloud Computing (OJCC), Vol. 2, Issue 1 , pp. 16-24, (2015).

7. Tang, B., Chen, Z., Hefferman, G., Wei, T., He, H., Yang, Q. : A Hierarchical Distributed Fog Computing Architecture for Big Data Analysis in Smart Cities. In: Proceedings of the ASE BigData and Social Informatics, , ACM , Kaohsiung, Taiwan (2015). doi: 10.1145/2818869.2818898

8. Rahmani, A., Thanigaivelan, N., Gia, T., Granados, J., Negash, B., Liljeberg, P., Tenhunen, H. : Smart eHealth Gateway: Bringing Intelligence to Internet-of-Things Based Ubiquitous Healthcare Systems. In: 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), pp. 826-834, IEEE, Las Vegas, USA (2015). doi: 10.1109/CCNC.2015.7158084

9. Bonomi, F. : The Smart and Connected Vehicle and the Internet of Things.

10. Bonomi, F., Milito, R., Zhu, J., Addepalli, S. : Fog Computing and Its Role in the Internet of Things. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, pp. 13-16, ACM, Helsinki, Finland (2012). doi: 10.1145/2342509.2342513

11. Bonomi, F., Milito, R., Natarajan, P., Zhu, J. : Fog Computing: A Platform for Internet of Things and Analytics. In: Bessis N., Dobre C. (eds) Big Data and Internet of Things: A Roadmap for Smart Environments, vol. 546 , pp. 169-186, Springer (2014).

12. Gazis, V., Leonardi, A., Mathioudakis, K., Sasloglou, K., Kirikas, P., Sudhaakar, R. : Components of Fog Computing in an Industrial Internet of Things Context. In: 12th Annual IEEE International Conference on Sensing, Communication, and Networking - Workshops (SECON Workshops), pp. 1-6, IEEE, Seattle, USA(2015). doi: 10.1109/SECONW.2015.7328144

13. Cisco Fog Computing Solutions: Unleash the power of the Internet of Things $http : //www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing - solutions.pdf$

14. Busching, F., Schildt, S., Wolf, L. : DroidCluster: Towards Smartphone Cluster Computing- The Streets are Paved with Potential Computer Clusters. In: 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW), pp. 114-117, IEEE, Macau, China (2012). doi: 10.1109/ICDCSW.2012.59

15. Masri, W., Ridhawi, I. A., Mostafa, N., Pourghomi, P. : Minimizing delay in IoT systems through collaborative fog-to-fog (F2F) communication. In: 9th IEEE International Conference on Ubiquitous and Future Networks (ICUFN), pp. 1005-1010,IEEE, Milan (2017). doi: 10.1109/ICUFN.2017.7993950

16. Kumar, A., Narendra, N. C., Bellur, U. : Uploading and Replicating Internet of Things (IoT) Data on Distributed Cloud Storage. In: IEEE 9th International Conference on Cloud Computing (CLOUD), pp. 670-677,IEEE, San Francisco, CA (2016). doi: 10.1109/CLOUD.2016.0094

17. Masip-Bruin, X., Marn-Tordera, E., Tashakor, G., Jukan, A., Ren, G. J. : Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems. In: IEEE Wireless Communications, vol. 23, no. 5, pp. 120-128 ,IEEE(2016). doi: 10.1109/MWC.2016.7721750

18. Souza, V. B., Masip-Bruin, X., Marin-Tordera, E., Ramirez, W., Sanchez, S. : Towards Distributed Service Allocation in Fog-to-Cloud (F2C) Scenarios. In: IEEE Global Communications Conference (GLOBECOM), pp. 1-6, IEEE, Washington, DC (2016). doi: 10.1109/GLOCOM.2016.7842341

19. Narendra, N. C., Koorapati, K., Ujja, V. : Towards Cloud-Based Decentralized Storage for Internet of Things Data. In: IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), pp. 160-168, IEEE, Bangalore (2015). doi: 10.1109/CCEM.2015.9

20. Malandrino, F., Kirkpatrick, S., Chiasserini, C. F. : How Close to the Edge? Delay/Utilization Trends in MEC. In: Proceedings of the ACM Workshop on Cloud-Assisted Networking, pp. 3742, ACM, CA, USA (2016).

21. Hu, P., Ning, H., Qiu, T., Zhang, Y., Luo, X. : Fog Computing Based Face Identification and Resolution Scheme in Internet of Things. In: IEEE Transactions on Industrial Informatics, vol. 13, no. 4, pp. 1910-1920 , IEEE (2017). doi: 10.1109/TII.2016.2607178

22. Chen, N., Chen, Y., You, Y., Ling, H., Liang, P., Zimmermann, R. : Dynamic Urban Surveillance Video Stream Processing Using Fog Computing. In: 2nd IEEE International Conference on Multimedia Big Data (BigMM), pp. 105-112, IEEE, Taipei (2016). doi: 10.1109/BigMM.2016.53

23. Brogi, A., Stefano, F. : QoS-aware Deployment of IoT Applications Through the Fog. In: IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1185-1192, IEEE (2017). doi: 10.1109/JIOT.2017.2701408

24. Sarkar, S., Misra, S. : Theoretical modelling of fog computing: a green computing paradigm to support IoT applications. In: IET Networks, vol. 5, no. 2, pp. 23-29, IEEE (2016). doi: 10.1049/iet-net.2015.0034

25. Luan, T. H., Gao, L., Li, Z., Xiang, Y., Wei, G., Sun, L. : Fog computing: Focusing on mobile users at the edge (2015). In: arXiv preprint arXiv:1502.01815

26. Hong, K., Lillethun, D., Ramachandran, U., Ottenwlder, B., Koldehofe, B. : Mobile fog: A programming model for large-scale applications on the internet of things. In: SIGCOMM workshop on Mobile cloud computing, pp. 15-20, ACM , Hong Kong, China (2013). doi: 10.1145/2491266.2491270