A Review on Fog Computing Technology

Firas Al-Doghman[1], Zenon Chaczko[2], Alina Rakhi Ajayan[3], Ryszard Klempous[4]
[123] Centre for Innovation in IT Services and Applications (iNEXT), School of Computing & Communications, Faculty of Engineering & IT, University of Technology, Sydney – 2007, NSW, Australia
[4] Faculty of Electronics, Wroclaw University of Technology, Poland
Email: [1] FirasQaisMohammedSaleh.Al-Doghman-1@student.uts.edu.au, [2] zenon.chaczko@uts.edu.au, [3] Alina.Ajayan@uts.edu.au, [4]ryszard.klempous.@pwr.edu.pl

***Abstract* – Out of the many computing and software oriented models that are being adopted by Computer Networking, Fog Computing has captured quite a wide audience in Research and Industry. There is a lot of confusion on its precise definition, position, role and application. The Internet of Things (IOT), todays' digitized intelligent connectivity domain, demands real time response in many applications and services. This renders Fog Computing a suitable platform for achieving goals of autonomy and efficiency. This paper is a justification of the concepts, interest, approaches, and practices of Fog Computing. It describes the need for adopting this new model and investigate its prime features by elucidating the scenarios for implementing it, thereby outlining its significance in the IoT world.**

***Index Terms* - Fog Computing, Cloud Computing, Internet of Things, Autonomics, Computational Intelligence.**

I. Introduction

The Internet of Things (IoT) epitomizes the cutting edge technology to realize theoretical and practical facets of computer networking, and has initiated a prominent transition in the how communication and interactions of our world. The strong need to remedy the growing concerns regarding dealing with the huge data influx in real-time and functioning the available bandwidth bounds led to the birth of Fog computing, which, intensively but not exclusively, operate along the network edge. Fog computing is a novel paradigm realizing distributed computing, network services and storage from beyond Cloud Computing Data Centers up until the devices along the network edge. This notion extends the inherent operations and services of Cloud computing, thus enabling a new breed of application. The primary function is to filter and aggregate data for Cloud data centers and apply logical intelligence to end devices. Fog Computing is similar to Cloud computing in many traits. However, it may be differentiated from the former by its ingrained end devices, the intensive spatial distribution achieved and offered mobility support. As Fog-based processing occurs along the network edge, the end-results reflect highly improved location awareness, low latency and Quality-of-Service (QoS) in streaming and real time applications [1]. Fog nodes are heterogeneous devices, ranging from high-end servers, access points, set-top boxes, edge routers to the end devices such as mobile phones, smart watches, and sensors etc. [2].

In the light of recent research, Fog computing has been effectively demonstrated using several techniques. The study in [3] developed an architecture for connecting vehicles to a "Fog" platform deployed at Road Side Units (RSUs) and M2M gateways. Such a system design, in regard of its characteristics, enables consumer centric services like M2M Data Analytics with Semantic Web Technologies, IoT services discovery and Connected Vehicles Management. Another method for implementing Fog computing was introduced by [4], wherein an android-like Appstore is to be applied on network devices, downgrading computations from occurring at Cloud level to the network, while in [5] an "IoT Hub" which is essentially a generic Fog node positioned along the edges of multiple networks, enhance the networks' capabilities through the employment of the following entities: border router; cross-proxy; cache; and resource directory. However, the article [6] addresses the utility based matching problem within the IoT domain using Irving's matching algorithm under node-specified preferences for efficient node pairing.

The Internet of Things and Connected Smart devices have had an exponential growth in terms of involved technologies, market participation and consumer approval, paving the way to the evolution of Fog computing principles, gradually amassing productive opportunities in various domains such as Vehicular networks, Body Area Networks (BAN), and the Smart Grid. The advantages of this computing procedure for services in several domains are needed to be investigated [1]. Fog Computing allows greater support and better response time to the Internet of things environment, it is suitable for real-time service requests, and to share resource efficiently an efficient and cooperative utility based pairing strategy between the high-end IoT nodes is needed [6]. The Fog acts as a link between IoT and the Cloud to induce the necessary extra functionalities for application-specific processing like filtering and aggregation before transferring the data to the Cloud. It should be able to decide what is to be sent (the content), how (data format) and when to send (time). During this process, it also needs to delete some redundant or invalid data, and aggregate the complementary data in the space and time dimensions [2].

The rest of this paper is structured as follows: Section II highlighting the previous work done in some literatures. Section III provides some methodologies used to describe Fog Computing practices. Section IV presents a evaluation scenarios for two use cases, followed by deployment case for Fog. Finally, Section V concludes the paper and provides directions for future work.

978-1-5090-1897-0/16/$31.00 ©2016 IEEE

II. REVIEW OF RELEVENT LITERATURE

In this section, some of the research conducted so far related to Fog Computing is introduced, which helps to build and support this paper.

A significant number of characteristics helps us identify the Fog as an inconsequential extension of the generic 'Cloud'. In terms of IoT, the work in [7] condenses this vision and defines the key features of Fog Computing which make it the appropriate platform for a number of critical Internet of Things (IoT) services and applications. It defines the Fog node aspects such as: a) Mobility; b) Low latency and location awareness; c) Wide-spread geographical distribution; d) Very large number of nodes; e) the predominant role of wireless access; f) Strong presence of streaming and real-time applications, and g) Heterogeneity. Similarly, [8] discloses the reasons as to why Fog Computing is the most-fitting natural computing platform for IoT through the main requirements for designing and building a scalable and adaptable IoT platform which include support of rapid mobility patterns, support of systems requiring reliable sensing, analysis, control and actuation and management of a vast amount of geographically distributed "things" whilst in [1] the many advantages offered by Fog computing for services in various areas is examined along with an analysis of the cutting-edge developments and security issues of this paradigm. It also includes the ideation that some inventions in computation and storage may be inspired in future to address data intensive services based on the interplay between Fog and Cloud. Moving onto [2], the characteristics and prospects of Fog computing and services in regard of healthcare systems is studied. Here the Fog functions in focus are Switching Networks, Pushing Services and Core Services and the need to support protocols such as ZigBee, WiFi, 2G/3G/4G, WiMax, 6LOWPAN, in contrast to the sole support offered by the Cloud for TCP/IP. The various communication protocols transfer data in different formats so the first thing the Fog should do is network switching between IoT and the Cloud. Here the Fog will act as an intermediary between end devices and the Cloud, and should provide pushing service to both, while ingesting acquired data and updating processed data onto the Cloud for long-time storage and deeply digging in parallel. Thus the prime function of the Fog is to achieve local data processing, storing and computing in devices of weak performance metrics.

Owing to its recent introduction and emergence, there is no available standard architecture regarding Fog-based resource management. [9] presents a simple model for this purpose, by taking into account resource prediction, resource allocation, and pricing in a realistic and dynamic way, while also considering customers' type, traits, and characteristics. This model is adaptable to the requirements of different CSPs

Investigated Aspect	Traditional Cloud	Fog Computing
Prediction Latency	5 seconds	1.5 seconds
Webpage display latency	8 seconds	3 seconds
Internet Traffic	75 Kbps	10Kbps
Hardware used	Amazon Web Server	Raspberry Pi

Table 1. Comparison of Results between Cloud and Fog

thanks to its' flexibility and dynamics, and so may be implemented in different environments with varied conditions. The authors in [10] describe an Adaptive Operations Platform (AOP) to provide an end-to-end manageability for enabling Fog Computing infrastructure according to the operational requirements of industrial processes, while [4] outlines a method for transforming the computation environment from the Cloud to the Fog by presenting an Appstore kind of system on network devices wherein the user can choose which data should be processed at the edge and which ones on the cloud. This involves tagging packets which are to be processed on the network device. Untagged packets are then sent directly to the cloud without any intermediate processing. However, the work in [11] presents the 'reliability defiance' posed by the current computing paradigms, and expands the debate towards reliable Fog platforms encompassing smart devices' networks communicating amongst themselves and with the Cloud. [6] delivers an efficient IoT-node pairing scheme between the same-domain IoT nodes in the Fog context. The utility based matching or pairing problem within the domain of IoT nodes is addressed utilizing Irving's matching algorithm under the node specified preferences, to ensure stable IoT node pairing. Interoperability, the keystone of IoT systems and defined as the ability in heterogeneous objects to inter-operate dynamically with minimal human intervention, together with IoT nodes' strong-fort – the diverse range of network and physical resources, allow their pairing for sharing resources amongst each other cooperatively to achieve user specific requirements. A refinement is proposed in the classical stable matching or pairing algorithm, which incites Edge computing with a better utility factor, thereby configuring more proficient device-to-device communication. The shared parking model created in [12] depicts deployment of the Fog solution and Roadside-Cloud concepts into the parking problem. The status of a parking slot - vacant or reserved - will be updated in the Fog server installed in local areas, which in turn delivers information on the managed empty spaces to Road Side Units or RUSs. At each RSUs, the RFPARK communicates with these fog servers in order to direct drivers to an optimal space. [13] surveys the expansion of the Fog concept onto the decentralized smart building control, recognizes Cloudlets as a distinctive case of Fog computing, in conjunction with the Software-Defined Networking (SDN) scenarios. Cooperative data scheduling and adaptive Traffic Light problems in the SDN based Vehicular Networks scene, as well as Demand Response in Macrostation and Microgrid based Smart Grids are discussed.

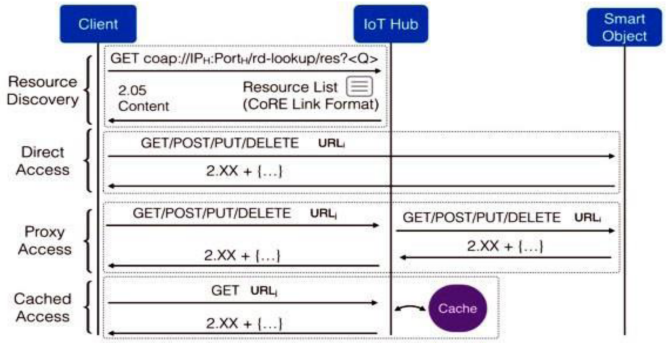

Fig. 1. Interaction between clients and heterogeneous smart objects with the mediation of the IoT Hub [5]

The evolution of IoT, according to its heterogeneous and dynamic nature, had undergone paradigm shifts many times over in terms of the design of network architectures and many approaches have been presented and proposed. An important design and implementation of a new Fog node was presented in [5], where it is placed along the edge of multiple physical networks with the aim of creating an IP-based IoT network to be used as the infrastructure for deploying IoT applications. Performance evaluation shown that the IoT Hub successfully had managed a number of heterogeneous physical networks, each with several connected devices, with limited resource usage, in terms of processing capacity and memory, thus expediting the deployment an IoT Hub even on low-end devices, such as RPis. The article [3] examines an architecture for connected vehicles with RSUs and M2M gateways including the Fog Computing Platform. M2M data processing with semantics, discovery and management of connected vehicles are also briefly examined likewise as Consumer-Centric IoT services enabled by the prominent features of Fog Computing while [14] had the goal of creating a mathematical model for Fog computing and assessing its applicability in the IoT context where it is pivotal to meet the demands of the latency-sensitive applications running along the network-edge for analyzing its suitability within the framework of IoT. The work also conducts a case study on comparative performance evaluation of Cloud Computing with that of the Fog for an environment involving a high number of Internet-connected devices demanding real-time services. It proves that for the SDN based Vehicular Networks scene, as well as Demand high number of latency-sensitive IoT applications the presence of Fog Computing enhances network performance in regard of power consumption, decreasing service latency as well as cost

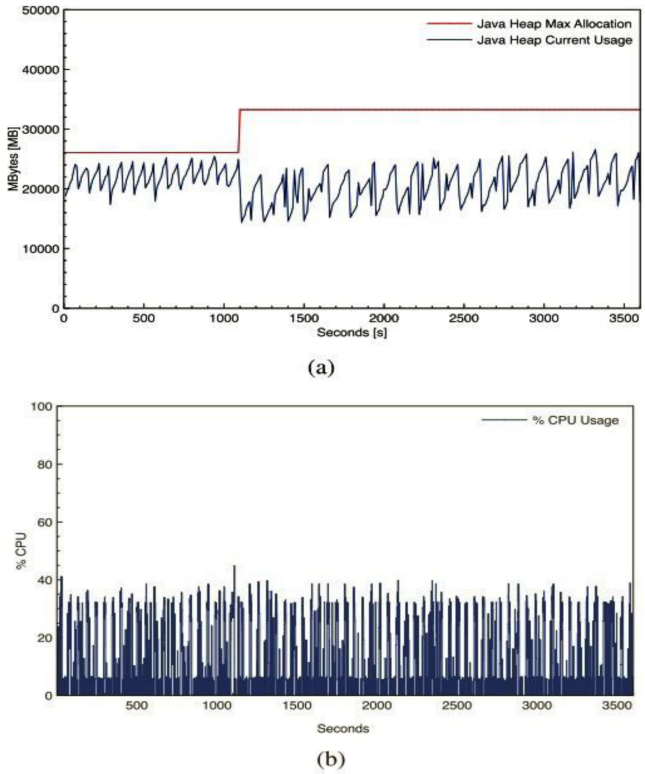

Fig. 2. Performance evaluation: (a) Heap memory used (dimension: [MB]); (b) CPU usage [5].

and overall network performance.

However, [8] examines some of the IoT's promising and challenging scenarios, discusses the inevitable chemistry between the Fog and the Cloud Computing in the near future and review some of the technologies that will require significant improvement, so as to bolster the applications scope for the IoT market, whereas [10] presents a deployment case for Adaptive Operations Platform (AOP), followed by evaluation scenarios for two use cases. It also develops an industrial solution in the context of predictive maintenance applications, using the Fog computing paradigm and the DMo technology. Finally, the studied article [13] states the prospective role of Fog computing in the important class of Cyber-Physical Systems (CPS), in the light of stimulating scenarios, demonstrated by Smart Grids, Vehicular Networks, Wireless Sensor and Actuator Networks, Smart Building Control and that Cloudlets are an important special case of Fog Computing..

III. METHODOLOGY

In this section we will introduce a review of the methodologies used to describe Fog computing practices.

The publishing [12] addresses the problem of Car Parking especially along the choice of an ideal slot; a matching method is used at RFPARK as the problem's solution. This article has the proposed Parking slots association problem formulated in the Fog-Cloud environment as a many-to-one matching game in which a set of vehicles 'V' has been designated towards a set of parking lots 'L'. The preferences concept formulated the model in terms of common and conflicting interest. A parking slot association algorithm detects a stable matching of parking slots for the vehicles extending requests.

The methodology used by [4] to propose a method of moving the computation from the cloud towards the network was to use a Raspberry Pi based Fog device by connecting three Arduino boards with temperature sensors to the Wi-Fi of the Raspberry Pi, they then measure the temperature of the surroundings every 5 seconds and send it to the router (the raspberry), the raspberry invokes a python script once it receives the temperature data that writes the received temperature values in different files (one file per Arduino board). The time series prediction is applied to the data in each file, the result of the time series prediction is then written into a MySQL database instance running on the AWS cloud, the PHP instance on the cloud reads the values from MySQL and displays it on a webpage. The results (Table 1) showed that Fog based architecture has a better response time compared to the cloud architecture.

The "IoT Hub" proposed by [5], with a Fog node deployed along multiple networks' edges is implemented using a Java-based implementation of CoAP called Californium and other related drafts. The IoT Hub has been deployed to a Raspberry Pi (RPi) Model B single board computer and used to manage resources hosted by heterogeneous SOs within a real-world IoT testbed within our department. The added value of the IoT Hub is its capability to hide completely the diverse nature of smart objects which can interact with them using uniform interfaces and without requiring any prior configuration.

Smart objects' interaction through the IoT Hub occurs as shown in Fig.1. The performance evaluation has shown that the IoT Hub is able to manage a number of heterogeneous physical networks, each with several devices, with a limited use of resources such as processing capacities and memory requirements, thus allowing the deployment of an IoT Hub even on low-end devices such as RPis. The results are shown in Fig. 2. Meanwhile the model introduced by [9] delivering an effective proficient resource management IoT framework the Fog entails, was thereafter implemented with Java/NetBeans 8.0 tools, later evaluated and analyzed with the help of using CloudSim 3.0.3 toolkit. The choice was due its capability to adapt to the dynamic requirements of dissimilar CSPs. It is applicable in various environments of widely different scenarios with respect to the resource prediction, allocation and pricing in in a dynamic and flexible way.

The proposed matching algorithm had its performance criteria addressed in [6] through simulation. The core concept in Irving's matching algorithm deals with solving the stable roommate problem for the purpose of construction of a one-to-one stable matching scenario. The refinement may then initiate a one-to-many cooperative pairing or matching amidst the nodes. Refining the Irving's procedure was to sustain quota-based nodal pairing on either one-to-one or one-to-many pairing. The corresponding performance measures are portrayed in Fig 3. Comparison results shows the efficiency of imparted by the Irving's algorithm for 5 node pairs with quota. It outperforms the greedy algorithm where the nodes are paired by considering the neighboring nodes available for pairing. Such levels of efficiency is the combination of quota based approach and the best utility based selection of nodes. It also that the cooperation between IoT nodes tremendously increases the total utility of such pairing actions. This phenomena also indicates that with a large number of nodes' sets and quota based pairing, overall utility of the entire set of node domain will increase drastically.

The work indexed in [10] defines an Adaptive Operations Platform (AOP) to address key limitations that an industrial infrastructure poses to data-intensive IoT applications. AOP is structured upon the service capabilities of the following layers: the Fog Infrastructure that includes networking equipment with particular Fog capabilities and provides for end-to-end communication services and the Operational Support System (OSS) that leverage the Fog Infrastructure to provide the customary asset management and business support functions (e.g., inventory, maintenance, provisioning, etc.). To efficiently leverage key features of the Fog Infrastructure, AOP includes several functional elements. The Model Building (MB) functional element combines static information about the failure models of the equipment types found in the industrial site along with dynamic data collected during the latter's operation. The Rule Mapper (RM) functional element tasked with mapping the fused model to a set of traffic handling rules understood by the SDN infrastructure. The Rule Deplorer (RD) functional element which, given a set of traffic handling rules and a description of capabilities in the SDN infrastructure, computes the deployment plan to apply this set of rules on the suitable elements. This model can be modified as in Fig. 4 to describe a more complete IoT model.

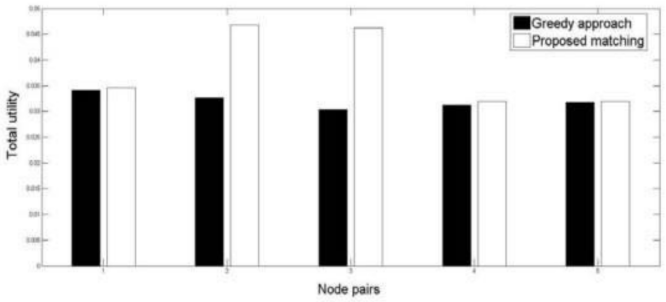

Fig. 3. Example comparison result showing how proposed matching algorithm can improve total utility of node pairs [6].

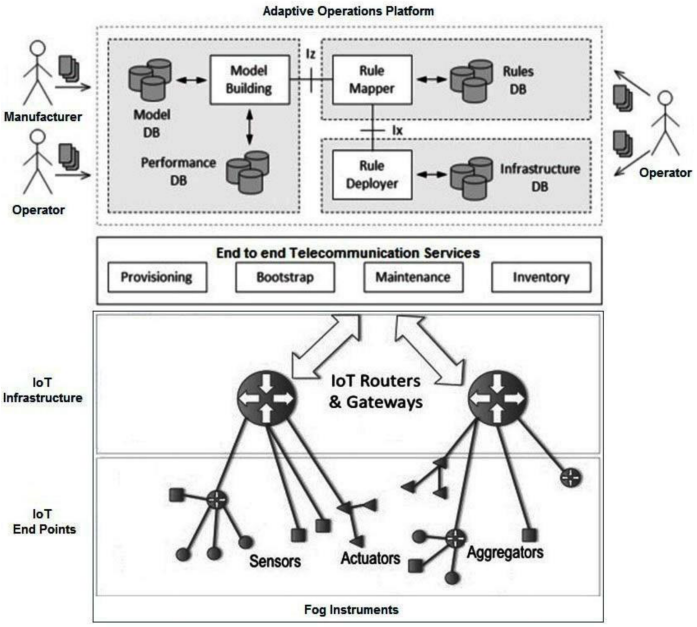

Fig. 4. Architecture of the Adaptive Operations Platform (AOP).

IV. ASPECTS AND SENARIOS OF FOG COMPUTING

Since its introduction, many proposals were made to implement Fog Computing. The following are some scenarios which were been introduced in order to implement it.

A. Scenario 1: Monitoring the status of a deployed machinery

Paper [10] introduced a scenario consist of N sensors monitoring the status of the deployed machinery to evaluate the performance of Fog components deployed in the network. The sensors report the captured values regularly through a router to a central server, as displayed in Fig. 5. The measurements are stored in a database and are illustrated to an operator. The objective of deploying the aggregators and the Sensor measurements is to investigate potential anomalies in the monitored machinery. The investigated scenario is tested using from the one hand a common router (see Fig. 5) and from the other a router/gateway supporting DMo (see Fig. 6), with the aim to compare the centralized scheme to the APO approach. In this scenario, as a first use case the reduction in data received by the operator, stored in the Historian, using a

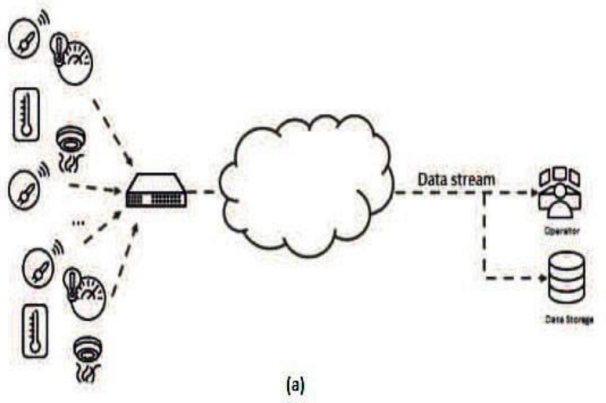

Fig. 5. Scenario using a normal router [10].

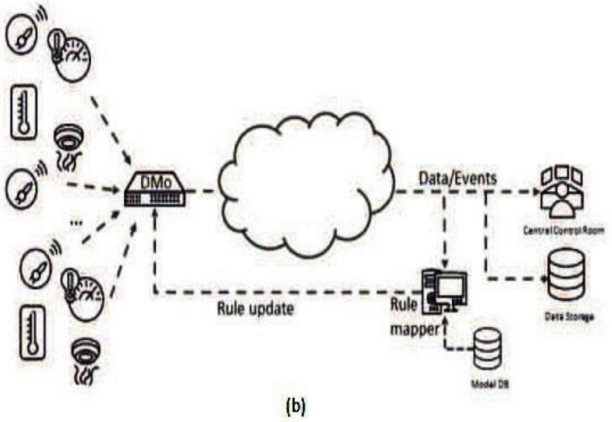

Fig. 6. Scenario using a router supporting DMo [10].

router supporting DMo is introduced. One of the N sensors was considered and supposed that this sensor is monitoring some operational behavior of a machine (e.g. its operational temperature). Looking at the traffic generated in a time interval equal to 120s, it is possible to see that the measurements are distributed as a Gaussian distribution. Specifically, most of the values are in the range [50, 70] which corresponds to the normal behavior of the specific monitored machine in the scenario. The values out of that range correspond to potential anomalies of the machine.

Using a common router (Fig.5), all the measurements gathered by the sensor are received by the server and stored, filling in most of the cases, the database with redundant information. This also leads to increased bandwidth usage for transferring the data but also it consumes resources and space in the database. Moreover, to detect the anomalies, a process has to take into consideration a large amount of collected data due to the large number of sensors. In order to reduce the amount of data stored, it is possible to apply the AOP approach. In this case, the server runs a Ma- chine Learning (ML) algorithm (e.g., k-Means) that is trained to understand the normal behavior of the appliance. Then, using the Rule Mapper (RM), it sets a rule to receive only data outside of the normal range in order to identify anomalies. These data will be sent as events to the operator. In this case, the ML indicates that the average of the received values is close to 60, which corresponds to the normal behavior, and sends to DMo a new rule that specifies to the router to forward to the server only values that represent anomalies, e.g., in the range [0, 50] and [70, 100]. Obviously, the range can be set differently according to thresholds. The results obtained using this approach are illustrated in Fig. 7. As expected, we obtain a huge decrease in the number of transmitted packets. Fig. 7, also depicts the CPU utilization of the router. We observe that the use of DMo claims only 8% of the CPU for a 1/3 reduction in the received traffic.

A second use case aims to demonstrate the capability of the proposed system to change dynamically the rules when needed. Specifically, the temperature of N = 6 appliances inside a rack is wanted to be measured and created an alert to the operator only when the temperature of some appliances is above a certain threshold (anomaly). We need also to identify the appliance that has the anomaly. We use 6 sensors, one for each appliance, and one sensor which measures the temperature inside the rack. As in Use Case 1, the normal behavior of the appliance is when the average value calculated in 120s is equal to 60. In order to save bandwidth and resources in the server we have created a rule in the DMo router to send to the server only measurements from the rack sensor. Looking at the Fig. 8, in normal conditions we have an amount of packets received equal to 200. Let us consider that after a while (at time 720s), the temperature monitored by the sensor rack increases (from 60 to 80). This provides an estimate that an abnormal event occurs inside the rack without designating which appliance has an anomaly.

In this case, the server receives values above the threshold (set to 60) and using the RM it sends a new rule to the DMo for the router to forward the data coming from all the 6 sensors. The result will lead to a temporary increase in the number of received packets, in order to identify the appliance with the anomaly. At this point, the ML identifies the sensor which is sending data above the threshold and sets a new rule which instructs the DMo router to send data only about this sensor (i.e., in addition to the rack sensor). The new rule applies until the situation in the rack goes back to the normal conditions. After that, the operator will receive again the measurements of the rack sensor only.

B. *Senario 2 : a Fog computiong platform*

[15] builds a proof-of-concept fog computing platform, consisting of two fog sub-systems where OpenStack was installed on each of them. Each of the fog sub-systems possesses one router and three servers. The routers are connected to the Amazon EC2 cloud through WAN, as well as connected with each other through LAN. The routers are also integrated with Wireless AP function, so that mobile devices can access the fog as well as the Amazon EC2 cloud through them. Four OpenStack modules were installed: Keystone, Glance, Nova, and Cinder. Keystone is for authentication and authorization; Glance is for VM image management; Nova is a compute module with simple network functionality; and Cinder is the block-level storage module. A VM offloading scheme was implemented which can migrate one VM to another fog cluster. The latency and bandwidth provided by fog and cloud were compared and fog computing has stronger advantages for clients. VM migration is essential in fog computing; its function was implemented in two ways. First, Fog 1 takes a snapshot of the VM to be migrated, compresses

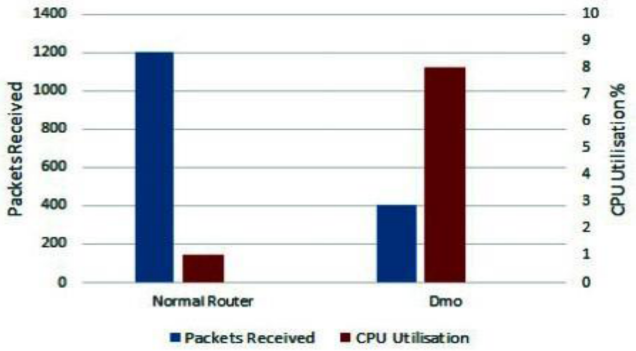

Fig. 7. Router CPU utilization and total received packets [10].

Fig. 8. Packets received and measured values vs. time from different sources [10].

it, and then transfers the compressed data to Fog 2. Fog 2 then decompresses the data and re-launches the VM by using the snapshot. In the second way, the VM has a "base" snapshot stored on both fogs. The incremental part of the VM's user snapshot will be transferred instead of the snapshot itself. The will only perceive "Transmission time" + "Post-transmission time" and the incremental way is better in this experiment. Also a face recognition application running across a smartphone and a fog was implemented, with an app for the user's smartphone to capture a face photo and transmit it to a remote server which will then try to recognize the face by matching it to the local face photo database. The same tasks were run on fog as well as on the Amazon EC2 cloud.

V. CONCLUSIONS

A review of the Fog Computing paradigm is introduced in this paper with the aim to resolve some confusion about what it is and the methodologies used to represent it. The introduction of Fog Computing in IoT enhances the its performance and increases the network resources' efficiency. Fog computing is defined as a computation and services extension of the cloud which covers the area from the front-end up to the cloud using heterogeneous devices communicating via wireless radio waves or by wired connection as in the case of fiber optics. This paper presents some previous work analyzing Fog characteristics, resource management models and the design of Fog network architectures and approaches. It describes some methodologies and case studies for implementing Fog Computing. The repercussion is that presence of Fog Computing within the IoT environment improves the performance of the network in many aspects. The potentiality of self-management resources paved the way to autonomic Fog Computing which could be an imperative aspect in implementing Fog Computing.

REFERENCES

[1] I. Stojmenovic and S. Wen, "The Fog Computing Paradigm: Scenarios and Security Issues," vol. 2, pp. 1–8, 2014. [Online]. Available: https://fedcsis.org/proceedings/2014/drp/503.html
[2] Y. Shi, H. Wang, H. E. Roman, and S. Lu, "The Fog Computing Service for Healthcare," 2015.
[3] S. Datta and B. Christian, "Fog Computing Architecture to Enable Consumer Centric Internet of Things Services," vol. 85, pp. 6–7, 2015.
[4] Y. N. Krishnan, C. N. Bhagwat, and A. P. Utpat, "Fog Computing- Network Based Cloud Computing," no. Icecs, pp. 250–251, 2015.
[5] S. Cirani, G. Ferrari, N. Iotti, M. Picone, G. Srl, and R. Emilia, "The IoT Hub : a Fog Node for Seamless Management of Heterogeneous Connected Smart Objects," 2015.
[6] S. F. Abedin, G. R. Alam, N. H. Tran, and C. S. Hong, "A Fog based System Model for Cooperative IoT Node Pairing using Matching Theory," pp. 309–314, 2015.
[7] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," Proceedings of the first edition of the MCC workshop on Mobile cloud computing, pp. 13–16, 2012. [Online]. Available: http://doi.acm.org/10.1145/2342509.2342513$\delimiter"026E30F$npapers2://publication/doi/10.1145/2342509.2342513
[8] M. Yannuzzi, R. Milito, R. Serral-Gracia, D. Montero, and M. Nemirovsky, "Key ingredients in an IoT recipe: Fog Computing, Cloud computing, and more Fog Computing," 2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pp. 325–329, 2014. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7033259
[9] M. Aazam and E.-N. Huh, "Fog Computing Micro Datacenter Based Dynamic Resource Estimation and Pricing Model for IoT," 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, pp. 687–694, 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7098039 [10] V. Gazis, A. Leonardi, and K. Mathioudakis, "Components of Fog Computing in an Industrial Internet of Things Context," 2015.
[11] H. Madsen, G. Albeanu, B. Burtschy, and F. Popentiu-Vladicescu, "Reliability in the utility computing era: Towards reliable fog computing," International Conference on Systems, Signals, and Image Processing, pp. 43–46, 2013.
[12] O. Tran, T. Kim, N. D. Tri, V. Nguyen, N. H. Tran, and C. S. Hong, "A Shared Parking Model in Vehicular Network Using Fog and Cloud Environment," pp. 321–326, 2015.
[13] I. Stojmenovic, "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks," 2014 Australasian Telecommunication Networks and Applications Conference (ATNAC), pp. 117–122, 2014. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7020884
[14] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the Suitability of Fog Computing in the Context of Internet of Things," IEEE Transactions on Cloud Computing, vol. 7161, no. c, pp. 1– 1, 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/ wrapper.htm?arnumber=7286781
[15] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog Computing: Platform and Applications," 2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), pp. 73–78, 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7372286 [16] R. Jhawar and V. Piuri, "Fault Tolerance and Resilience in Cloud Computing Environments," Computer and Informaion Security Handbook, vol. 2, 2013. [Online]. Available: http://spdp.di.unimi.it/ papers/JPCISWeb.pdf
[17] F. Zhu, G. Li, Z. Li, C. Chen, and D. Wen, "A case study of evaluating traffic signal control systems using computational experiments," IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 4, pp. 1220–1226, 2011.
[18] A. Tsitsigkos, F. Entezami, T. Ramrekha, C. Politis, and E. Panaousis, "A case study of Internet of Things (IoT) based on Wireless Sensor Networks (WSNs) and Smartphone (iPhone)," 2012.
[19] H. Shi, N. Chen, and R. Deters, "Combining Mobile and Fog Computing: Using CoAP to Link Mobile Device Clouds with Fog Computing," 2015 IEEE International Conference on Data Science and Data Intensive Systems, pp. 564–571, 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7396558