HBRP PUBLICATION

Journal of Advance Research in Mobile Computing
Volume 1 Issue 3
DOI: http://doi.org/10.5281/zenodo.3591228

Fog-Mobile Edge Performance Evaluation and Analysis on Internet of Things

Wasswa Shafik, S. Mojtaba Matinkhah*, Mohammad Ghasemazade
Computer Engineering Department, Yazd University, Yazd, Iran
****Corresponding Author***
***E-mail Id:-** matinkhah@yazd.ac.ir*

ABSTRACT
Network analysis and optimization are emerging research that calls for the number of entities like increased availability, resources management, performance analysis, privacy-oriented evaluation among others. The dynamic advancements in technological edges increase availability, connectivity of the devices that provides and controls the network resource management, reduced latency and bandwidth varieties that contribute to the smart innovation including smart city, smart homes, among others. In this paper, we analyze the performance of mobile edge computing of the Internet of things (IoT) that determines the service quality of the network as seen in other computing arenas. More still, we propose a performance model called Erlang Performance Model (EPM) Erlang formulas basing different assumptions during data transmission. The article provides a deep understanding of queuing theories concerning the discipline of First-In-First-Out (FIFO) in mobile edge networks through availing mathematical calculations of virtual parameters. The paper proved that Erlang formulas could be alternatively used in designing performance models as we illustrated. The simulated MATLAB results showed that analyzing the performance creates reduces congestion/ traffic during data transmission and better determination of grade and quality of service to the users and smart communication providers.

Keywords:-*Blocking Probability, Erlang Formulas, Requests Evaluation, Internet of things*

INTRODUCTION

In a few years ago, Fog-Mobile Edge computing (FMEC) is one of the popular transits to smart cities and fifth-generation computation that provides large variety of services and virtually unrestricted accessible resources for users like new applications, such as virtual reality and smart building control, smart homes, among others have emerged due to the large number of resources and services conveyed by cloud computing and increased connectivity of device over internet (IoT) [1].

Nevertheless, the delay-sensitive applications face the problem of large latency, specifically once several smart devices and things are getting involved in human's un enabling capacities to meet the requirements of low latency, location consciousness, and flexibility support as in [2], [3]. With an attempt to solve these challenges, surveys have suggested a dependable solution through the FMEC to put the services and resources of the cloud closer to users, which facilitates the leveraging of available services and resources in the edge networks [4]. Through this, we are moving from the core (cloud data centers) to the edge of the network closer to the users. FMEC dependability is based on providing a user-centric service [5].

Purposely, FMEC is to run the heavy real-time applications over network edge directly using the billions of connected mobile devices like MIMO in [6]. Moving to the edge enables pioneering applications for example Cloud Radio Access Networks (C-RAN) IoT, connected cars, E-health among others [7]. We are only just beginning to uncover the true power of the edge and how to deliver applications in an efficient, cost-effective way. The IoT paradigm is incorporating "things" from the physical world into the Internet environment to enhance the monitoring and intelligent control of physical, digital and social systems [8]. In cloud-centric IoT applications, the spotting streams from these things at the edge of the network are accumulated as depicted [9]. Processed centrally at private clouds and the responses are consistent back to the things, leading to significant latencies and bandwidth costs [10].

To satisfy the ever-increasing demand for computing resources from emerging applications such as IoT advocates to supply large cloud data centers with microdata centers [11]. These micro data centers, also called Fogs, are located at the edge of the network, closer than cloud data centers for instance in [12]. Further still, edge devices such as smartphones and gateways themselves have non-trivial compute volume and are even closer to the user [13]. As a result, it is possible to utilize such edge and fog resources to off-load reckoning that would conventionally have been carried out at the network [14]. On the one hand, the developing of mobile edge computing infrastructure requires examining operating systems with virtualization and middleware techniques for fabric management, which requires extensions to current programming models that can benefit from such massively distributed systems [15, 16].

This also opens up other challenges including security, privacy discussed in details in [17]and generating of trust of the edge and fog resources [18], resource management and performance analysis for IoT applications shown in [19], transient and constrained resources where lots of metrics and models are illustrated in [20]. Fundamentally, the main objective of the Internet of Things (IoT) is to address security issues, obtain and analyze data from assets that were previously disconnected from most data processing tools [21]. IoT simply defined as a system of interrelated computing devices, mechanical and digital machines, or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a fog/edge network without requiring human-to-human over human-to-computer interaction with limited blockages [22,23].

In appreciation of the current efforts, many efforts seen creating many technological platforms used over the internet through the introduction of intermediate and advance algorithms [24]. Network computing makes great use of the Erlang to understand traffic patterns within the network and they use the figures to determine the capacity that is required in any area of the network following technological rules that ease the data transfer like inference [25]. Regardless of the pattern mentioned, this entity of the setup leaves the gap of determining the quality and grade of the performance of services obtained by the users, so by all this, a performance model is needed to determine the performance [26]. This will help to measure offered loads, multi-cells discussed in [27], random loads at times least loads on service-providing entities like in buffer circuits where some requests handled in different ways leading to excess costs for unused capacities [28].

These devices perform a task in the physical world such as pumping water, switching electrical circuits, or sensing the world around them but at times delayed depending on various factors. Fog and edge computing are effectively the same in perspective to the resource availability to the users, both are concerned with

Journal of Advance Research in Mobile Computing
Volume 1 Issue 3
DOI: http://doi.org/10.5281/zenodo.3591228

leveraging the computing capabilities within a local network to carry out computation tasks that would ordinarily have been carried out in the cloud since basic entities are implemented.

Notably, the IoT was initially most interesting to business and manufacturing, where its application is on occasion like machine-to-machine (M2M) like illustrated in [29]. However, the emphasis is now on filling our homes and offices with smart devices, transforming it into something relevant to almost everyone regardless of distance and location. Same functionalities in terms of pushing both data and intelligence to analytic platforms that are situated either on, or close to where the data originated from, whether that's screens, speakers, motors, pumps or sensors hence efficient data request answering within the nodes [30]. Edge computing usually occurs directly on the devices to which the sensors are attached or a gateway device that is physically "close" to the sensors and enabling various technologies with the IoT systems. Fog computing moves the edge computing activities to processors that are connected to the LAN or into the LAN hardware itself so they may be physically more distant from the sensors and actuators and by so doing.

This attempt will help in making hard real-time computations on the generated IoT data analytics for smart homes, without having to send the data to the cloud that may even cause the denial of services. Physical assets generate this data or things deployed at the very edge of the network that is to say motors, light bulbs, generators, pumps, and relays among others that perform specific tasks to support the desired privacy process. The IoT Things is about connecting these unconnected devices and sending their data to the cloud or Internet to be analyzed putting security and privacy in mind [32].

Fog Computing seamlessly extends cloud computing into the edge for secure control and management of domain-specific hardware, software, and standard computes, storage and network functions within the domain and enable secure rich data processing applications across the domain since all data from physical assets or things are transported to the cloud for storage and advanced analysis. Once in the cloud, the data is used for cognitive prognostics and utilization factor approaches [33].

The IoT bridges the gap between the digital world and the physical world, which means that hacking into devices can have dangerous real-world consequences, this issue made contributions to the methodological approach to cognitive IoTs. Hacking into the sensors controlling the IoTs could trick the operators into making a catastrophic decision that will detection malware. However, there is a key difference between both concepts of fog computing and edge computing involve pushing intelligence and processing capabilities down closer to where the data originates at the network edge. The key difference between the two architectures is exactly where that intelligence and computing power is placed. With fog, pushes intelligence down to the local area network (LAN) level of network architecture, processing data in a fog node or IoT gateway and edge pushes the intelligence, processing power, and communication capabilities of an edge gateway or appliance directly into devices like programmable automation controllers (PACs).

The rest of this paper is structured as follows. In section 2, we describe related work and problem analysis. In section 3, illuminates model methodology, queuing analysis in context to the model, section 4 explains Erlang formulas with associated functionality and extended understanding

Journal of Advance Research in Mobile Computing
Volume 1 Issue 3
DOI: http://doi.org/10.5281/zenodo.3591228

of the equations. In section 6, we explicate numerical results performance analysis, section 7 with the conclusion and our future studies.

RELATED WORK AND PROBLEM ANALYSIS

In this section, we provide an attempt for a performance model that is discussed through a brief ground back of current studies that were done providing a clear problem analysis.

Related Work

To Begin with, Mobile edge Computing brings the much-required data collection, processing, and analysis closer to the data sources at the edge enabling both edge and fog analytics. Sometimes it is more effective to analyze data locally, however, in some cases, the data may need to go to the cloud [34]. As for edge computing, the data is processed on the device or sensor itself without being transferred anywhere leading to increased privacy within the IoT. Within this section, we show some of the most efforts carried out in this regard together with the great analysis of their models [35]. It can become a complex issue for brands to handle, as data sets that require more sophisticated algorithms are better handled in the cloud, whereas simpler analytical processes are best kept at the edge.

The most proposed models to evaluate the performance of infrastructure as a service, data centers by assuming all Virtual Memory System (VMS) /servers are homogeneous; it combined with the transform-based analytical approach. The study that was made depicts that smart product needs evaluation in terms of performance and sustainable behavior within its capacities like monitoring, control, autonomy, and optimization as well as detailed in [36]. Security basing a major component we can use to determine the performance; researchers proposed a new theorem called the Chinese Remainder Theorem (CRT)-based data storage mechanism that was focusing on the data storage mechanism without any possible data linkages. This proposed CRT-based secured storage scheme adopted two encryption schemes, which use new formulas for performing the first and second encryption, and introduced a new formula for decrypting the cloud data for best evaluation and performance [37].

Regardless of the efforts, in case IoTs cannot harmonize in all possible applications, their conclusion of saying that it is better than the current security measures because of this failure to have device harmonization. In appreciation that to existing systems, like that is discussed, IoTs still have outstanding security, privacy, and performance issues altogether. For instance, the problem in was solved by modeling heterogeneous VMs but by assuming that there existed only one type of resources in the system clearly in other Scalable modeling approaches, for instance, interacting stochastic model approaches caused by the monolithic model were further proposed like importantly the RBC system well discussed in [38].

Most currently, the proposed a novel semi-supervised learning approach clearly shows that increasing tendency toward different edge computing prototypes requires the need for the evaluation and performance models, proposing different designs and approaches in [39]. The cost of experimenting on real edge/cloud environments is now a factor that pushes the researchers to evaluate their proposals with new performance models and simulators. Performance-related service latency of different cloud services is modeled and evaluated based on it.

Mathematical Parameters and Notation

In section, we demonstrate all computation

HBRP PUBLICATION
Journal of Advance Research in Mobile Computing
Volume 1 Issue 3
DOI: http://doi.org/10.5281/zenodo.3591228

that is within the limits of the model, we begin with our main mathematical and virtual notation in Table 1. As identified, basing on a recent literature assessment, a performance model to be used in the evaluation for IoTs tasks is not inexistent, the basic points, given by to depict the arrival time, will depict service rate, Constant (deterministic) D be service rate are depicted, we explain in details all related concepts to the mode, Quotient rule formula for *variable*, more detailed below in Table 1.

***Table.1:**-Mathematical parameters and notation*

Symbol	Quantity
Λ	Request arrival rate
A *Avg* Ii	Traffic Intensity in Erlangs Average The inverse of Erlang B
N P_b	Number of channels Request probability of blocking
λ/μ	All servers being busy
C(n,rho)=p	Intensity
M	Service rate per server
A	Conditional traffic intensity was a* d
D	Average request duration. Most literature defines it as μ =1/d
C	Erlang C
$C <= p$	Loop recursion of the routine
AHT *NB* *NS* *U* *N*	**A**verage **H**andle **T**ime (average duration of service) The average **n**umber of **b**usy servers The average **n**umber of users requesting Utilization fraction (occupancy, the average fraction of time that each server is busy) Traffic offered to the buffer circuits
M	*Poison random arrivals*

Problem Description and Analysis

A midst limited strategic focus on mobile edge computing, timely, meaningful feedback, appropriate recognition and rewards, proper training and communication over network widening. It should be noted that the essentials of mobile edge computing service providers have to think of performance evaluation models that can be used to evaluate the performance quality and grade of services to diminish various management and investment costs. Basing on the currently related efforts and challenges. We lastly observe that it is significant for mobile edge computing to have performance evaluation to the avoidable phenomena in smart generations. The contribution to the paper is as follows.

1. The analytical approach and analysis of how quality and grade of IoTs is determined on mobile edge devices through virtual real assumptions of IoT.
2. Provide a model that can be used to evaluate the performance of all IoTs on the network.

PROPOSED MODEL AND ANALYSIS

In section, a simplified mathematical expression is presented line to performance analysis. The model has precise formulas that describe it completely. Through divergence, our understanding of a real-world smart city is used in computing but the research in implement is in small numbers and still in its initial stages. Smart systems are mysterious, some are messy, with great complexity. We begin by illustrating the virtual parameter used within the model. Applying the queuing theory facilities the

DOI: http://doi.org/10.5281/zenodo.3591228

decision in design modeling to use with which systems, and how to perform the mapping three above to glean helpful information about real-world smart city systems.

Queuing Theory concerning the Model

Queuing theory (Qt) defined as the study of "the phenomena of standing, waiting, and serving". The systematic application of queuing theory to telecommunications began years ago. In this section, we introduce the queuing theory that is complete enough so that a specialist can understand the theoretical basis for the functionality of the Erlang. Mathematically, this can be understood as systems representing "standing, waiting, and serving", which we refer here to a queuing model, in a real-world system. For example, a requesting node in the fog networks is where real computation is carried out and lastly mapping between the queuing model and the real-world system since it is a simplified system that exists only in the idealized world of mathematics.

Queuing Analysis (M/M/ …)

Let us take the first approach as (M/D/1) considering random arrival, Constant (deterministic) service rate, with one channel. Therefore, we find out that the expected Avg queue length E (m) is expressed as.

$$E(m) = \frac{\left(2\rho-\rho^2\right)}{2} \quad (1-\rho) \qquad (1)$$

From the (1), the expected Avg total time E (v) is obtained as follows

$$E(v) = 2 - \frac{\rho}{2\mu} \quad (1-\rho) \qquad (2)$$

Then remain question is on waiting time, it is E(w) that can be given us

$$E(w)=\frac{\rho}{2\mu} \ (1-\rho)$$

Now considering (2), to check the probability, considering the assumption that (M/M/1) as the random arrival, random service with one service channel. The probability of having zero requests in the system channel *Pb* is given by

$$P_{b0} = 1-\rho \qquad (3)$$

Meaning that probability of having n request of resources, (3) in the system channel as

$$P_{bn} = \rho^n P_{b0} \qquad (4)$$

So the expected Avg queue length E(m) is given as

$$E(m) = \frac{\rho}{(1-\rho)} \qquad (5)$$

Observing (5) ρ the half of the ρ will be the E(V) expressed mathematically is

$$E(v)=\frac{\rho}{\lambda} \ (1-\rho)$$

Then the E(W) is observed by

$$E(w) = E(v) - \frac{1}{\mu}$$

Using the expression, the $\rho/c<1$ (M/M/c) were, random arrival, random services, c channel. The probability of having *quotient* in mind is going to be given as considering 0 requests.

$$P_{b0} = \left[\sum_{n=0}^{c-1}\frac{\rho^n}{n!}+\frac{\rho^c}{c!\left(1-\frac{\rho}{c}\right)}\right]^{-1} \qquad (7)$$

From the same layout of (7), our E (m) is stated as

$$E(m) = P_{b0}\,\frac{\rho^{c+1}}{cc}\,\frac{1}{\left(1-\frac{\rho}{c}\right)^2} \qquad (8)$$

E(n) is obtained as

$$E(v)=\frac{E(n)}{\lambda}$$

Taking E(n) is the expected Avg number in the system.

$$E(n)=E(m)+\rho$$

E(w) is illustrated as

$$E(v)-\frac{1}{\mu}$$

HBRP PUBLICATION

Journal of Advance Research in Mobile Computing
Volume 1 Issue 3
DOI: http://doi.org/10.5281/zenodo.3591228

Combining the above equation approached lets us take the (M/M/c/f) in case that random arrival, random services, and c service channel, f are the maximum number of the requests.
Considering the situation to be the

$$\frac{\rho}{c} = 1$$

so, the chances are given by.

$$P_{b0} = \left[\sum_{n=0}^{c-1} \left(\frac{1}{n!} \rho^n \right) \left(\frac{\rho^c}{c!} \right) \left(\frac{1 - \frac{\rho}{c}^{F-c+1}}{1 - \frac{\rho}{c}} \right) \right]^{-1} \quad (9)$$

This result and gives us

$$P_{b0} = \left[\sum_{n=0}^{c-1} \left(\frac{1}{n!} \rho^n \right) \left(\frac{\rho^c}{c!} \right) (F-c+1) \right]^{-1}$$

This is seen from the above equation that it is obtained as

$$\frac{\rho}{c} = 1$$

let us know to consider our n now.

$$P_{bn} = \frac{1}{n!} \rho^n P_{b0} \quad | \, 0 \le n \le c$$

Then the

$$P_{bn} = \left(\frac{1}{c^{n-c} c!} \right) \rho^n P_{b0} \, | \, c \le n \le F$$

So, the E (m) is given as

$$\mathrm{E}(\mathrm{m}) = \frac{P_{b0} \rho^c \left(\frac{\rho}{c} \right)}{c! \left(1 - \frac{\rho}{c} \right)^2} * \quad (10)$$

$$\left[\left(\frac{\rho}{c} \right)^{F-c+1} - \left(1 - \frac{\rho}{c} \right) (F-c+1) \left(\frac{\rho}{c} \right)^{F-c} \right]$$

So, from (10) above, helps us to obtain E (n) as

$$\mathrm{E}(\mathrm{n}) = \mathrm{E}(\mathrm{m}) + c - P_{b0} \sum_{n=0}^{c=1} \frac{(c-n) \rho^n}{n!} \quad (11)$$

After that, (11) we take the E (v) to be

$$\mathrm{E}(\mathrm{v}) = \frac{\mathrm{E(n)}}{\lambda (1 - P_{bF})} \quad (12)$$

Now, we need to know that distribution of Erlang taking the Erlang (E) to be (n, x) for scale and shape parameters respectively since there are no waiting positions, these lost customers are also said to experiencing blockage or to be blocked taking the random variable to be r**.**

$$f_{(r)} = \frac{r^{x-1} e^{-r/n}}{n^r (x-1)!} \quad (13)$$

Cumulatively, support E the above distribution function F in the expression can be stated as

$$F_{(r)} = P_{b0} (E \le r) = 1 - \sum_{i=0}^{n-1} \frac{e^{-r/n} r^x}{n^r n!} \quad (14)$$

Basing on the observation from (13) and (14) two equations are obtained as > 0, the functionality of surviving S of E

$$S_{(r)} = P_{b0} (E \ge r) = 1 - \sum_{i=0}^{n-1} \frac{e^{-\frac{r}{n}} r^x}{n^r n!}$$

Generally, the combination of the functionality of hazard (Hd) is simply given by the

$$Hd_{(r)} = \frac{F_{(r)}}{S_{(r)}} = \frac{r^{x-1} e^{-x/n}}{n^x (x-1)!} \sum_{i=0}^{n=1} \frac{n^x n!}{e^{-r/n} n^x}$$

So, the characteristics of Erlang in general is

$$\varnothing t = E\left[e^{itE} \right] = (1 - it\alpha)^{-n} \quad (15)$$

Under the (15), the functionality m of Erlang is

$$m_{(t)} = E\left[e^{ir} \right] = (1 - t\alpha)^{-n}$$

Therefore, (12) is minus $\frac{1}{\mu}$.

DESCRIPTIVE MODEL ILLUSTRATION

In this section, we present a descriptive format of the proposed model Figure 1, we illustrate the physical model layout that may be used with the implementations of Erlang formula functionality during the

Journal of Advance Research in Mobile Computing
Volume 1 Issue 3
DOI: http://doi.org/10.5281/zenodo.3591228

mobile edge computing for smart city performance analysis with detailed functionality.

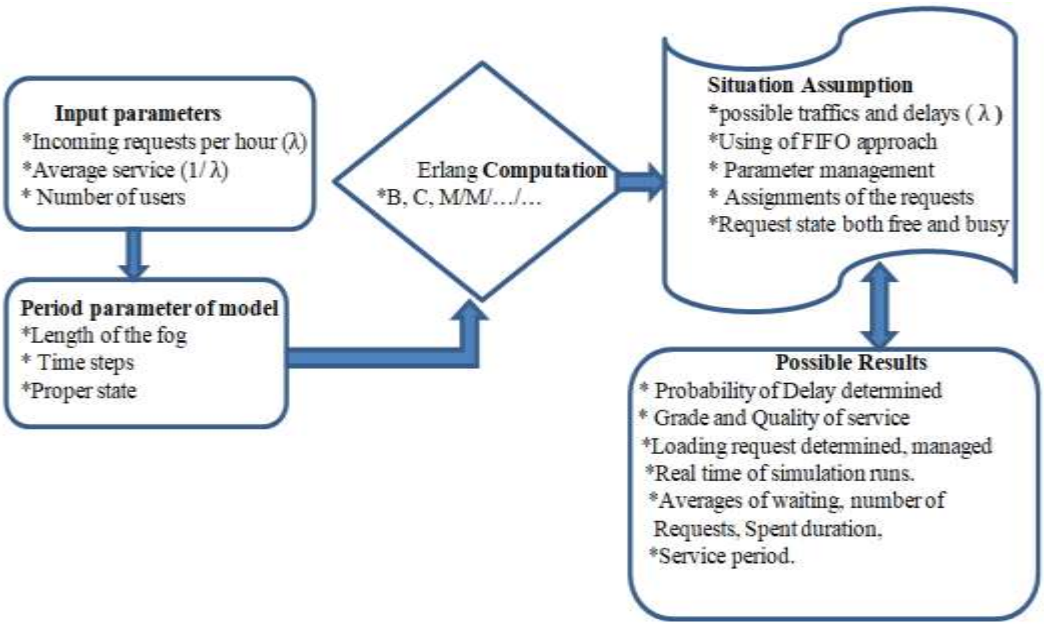

Fig.1:-*Physical representation and illustration of the Erlang Performance model.*

Discussion of Model and functionality

Having the got expression and model above, in this section, we discuss the implementations of most important functionalities considering the model assumptions as illustrated in the Erlang during the computing at least once and sometimes many times.

Model Loss Functionality

Firstly, let us begin with symbols and simple function to understand the formula where our E is our Erlang, λ becomes the mean arrival rate of the new request and holding duration.

$E=\lambda * h$

Let us introduce,

$$P_b = \frac{\frac{n^x}{x!}}{\sum_{i=0}^{x} \frac{n^i}{i!}} \quad (16)$$

The summation is got as seen the formula

i=0 to x

Originally, Erlang B assumed that traffic is brought due to various attempts. In simplification, B (n, m) where B is the number of channels below

$$P_b = B(n,x) = \frac{\frac{n^x}{x!}}{\sum_{i=0}^{x} \frac{n^i}{i!}} \quad (17)$$

To analyze this (16 and 17) we take in this regard x to be traffic intensity measured in Erlang to use the above notation, we take our x to be

$$x = \frac{\lambda}{\mu},$$

Basing on the function, $P_b = B(n,x)$

Besides, the AHT simply is given by

$AHT = 1/\mu$

From the equation above, our NB is obtained by

$NB = (1-B)*x$

If we pay attention to our NB, you will find out that NS is given the same as illustrated in that last equation. As the utilization (U) of given by the

$$U = \frac{NB}{n}$$

$$U = (1-B) * \frac{x}{n}$$

The general formula as stated above give us an attempt to calculate the probability

HBRP PUBLICATION
Journal of Advance Research in Mobile Computing
Volume 1 Issue 3
DOI: http://doi.org/10.5281/zenodo.3591228

only if-then and x as defined. Simplifying the (17) the function $B(n,x)$ defined by

$$B(n,x) = \frac{\left(\frac{x^n}{n!}\right)}{\left(1+x+x^2 2!+x^3 3!+\frac{x^n}{n!}\right)} \tag{18}$$

Computes the probability that an arriving customer will be blocked in the Erlang B model may be obtained in the assumption that λ/μ. From mostly used mathematical proofed equation about the maximization that is given by.

$$f'\left(\frac{u(x)}{v(x)}\right) = \frac{u'(x)v(x)-u(x)v'(x)}{v^2(x)} \tag{19}$$

To proof the truthiness of the (19) we simplified it that that

$$f_{(x)}=\left(\frac{u(x)}{v(x)}\right)$$

Then, this provides

$$f_{(x)}=u(x).v^{-1}(x)$$

$$f'(x)=u'(x).v^{-1}(x)+-v^{-2}(x).v^1(x).u(x)$$

This produces to

$$f'(x)=u'(x).\frac{1}{v(x)}-\frac{1}{v^2(x)}-v^1(x).u(x)$$

From the above expression,

$$f^1(x)=\frac{u'(x)}{v(x)}-\frac{v^1(x)u(x)}{v^2(x)}$$

This result

$$f^1(x)=\frac{u'(x)}{v(x)}-\frac{v^1(x)u(x)}{v^2(x)}$$

This simply means that

$$f'(x)=\partial/\partial x\left\{\frac{u(x)}{v(x)}\right\}$$

At this point, our final expression is going to be

$$f'(x)=\partial/\partial x\left\{\frac{u(x)}{v(x)}\right\}=\frac{u'(x)v(x)-v'(x)u(x)}{v^2(x)}$$

Taking us back to our original quotient rule formula for variable this (19), therefore if we consider our assumption to be

$P_b(x,\mu,\lambda)$

Subsequently,

$$P_b(\lambda,\mu,x) = \frac{\left(\frac{\lambda}{\mu}\right)}{x!}\frac{1}{\Sigma_{i=0}^{x}\left[\left(\frac{\lambda}{\mu}\right)^i\frac{1}{I!}\right]} \tag{20}$$

Mathematically, we usually refer to this function the Erlang Loss Function and the valve can complement by the assumption of n'. Therefore, the completed complement as

$$P'_b = B'(n',x') = \frac{\frac{n'^{x'}}{x'!}}{\Sigma_{i=0}^{x'}\frac{n'^{i}}{i!}} \tag{21}$$

With the assumption that recall factors in (20) are considered in early efforts. It is particularly important to understand the traffic volumes at peak times of the day. This formula can be recursive.

$$\frac{1}{P_b}=\frac{\Sigma_{i=0}^{x}\frac{n^i}{i!}}{\frac{n^x}{x!}} = \frac{x!}{n^x}\sum_{i=0}^{x}\frac{n^i}{i!} = \sum_{i=0}^{x}\frac{x!n^x}{n^i i!} \tag{22}$$

For traffic variations let us compare take our formula (22) to be

$$\frac{1}{P_b} = \sum_{i=0}^{x}\frac{x!}{n^{x-i}i!} \tag{23}$$

From equation.23, therefore,

$$\frac{1}{P_b(x+1)}=\sum_{i=0}^{x+1}\frac{(x+1)!}{\left(n^{x+1-i}i!\right)}=\frac{(x+1)!}{(x+1)!}+\sum_{i=0}^{x}\frac{(x+1)!}{n^{x+1-i}i!}$$

This results in

$$1+\frac{x+1}{n}\sum_{i=0}^{x}\frac{x!}{n^{x-1-i}i!}$$

In supplementary,

$$\frac{1}{P_b(x+1)}=1+\frac{x+1}{n}\frac{1}{P_b(x)}$$

When we take our x to be 0, it same as saying that

$1/(P_b(0))=0!/(n^0 0!)=1.$

It was necessary to understand the maximum communications traffics at the peak times of the day and to be able to determine the acceptable level of service required. This can only be analyzed when we take the x to be constant.

Meaning that

Journal of Advance Research in Mobile Computing
Volume 1 Issue 3
DOI: http://doi.org/10.5281/zenodo.3591228

$$P_b(n) > 0$$

Simply by applying the commonly used (21), systems try estimating the maximum number of traffics required connections or private wire connections use essentially the traffic model.

$$P_b'(n)=\frac{\frac{xn^{x-1}}{x!}\sum_{i=0}^{x}\frac{n^i}{i!}-\frac{n^x}{x!}\sum_{i=0}^{x}\frac{in^{i-1}}{i!}}{\left(\sum_{i=0}^{x}\frac{n^i}{i!}\right)^2}$$

By observing the downer part of the equation is squared, putting that in mind mathematically meaning is greater than 0 and it is so in case our n equals 0, so we need to prove the above equation.

$$\frac{xn^x}{x!n}\sum_{i=0}^{x}\frac{n^x}{x!}-\frac{n^x}{x!}\sum_{i=0}^{x}\frac{in^{i-1}}{i!}=\frac{n^x}{x!}\left(\sum_{i=0}^{x}\frac{xn^{i-1}}{i!}-\sum_{i=0}^{x}\frac{in^{i-1}}{i!}\right)$$

In simplifying traffics represented in brackets as is the number of hours of request traffic there are during the busiest hour of operation as

$$\sum_{i=0}^{x}\frac{xn^{i-1}-xn^{i-1}}{i!}=\sum_{i=0}^{x}\frac{n^{i-1}(x-i)}{i!}$$

Finally, to have the above, we consider the assumption that it's greater than 0, Mathematically

$$\frac{n^x}{x!}>0$$

Therefore, it was given by

$$\sum_{i=0}^{x}(x-i)=(x+1)x-\sum_{i=0}^{x}i=x(x+1)-\frac{x(x+1)}{2}>0$$

Analyzing Model Base Formulas

The Extended Erlang B is closely similar to Erlang B, but it can be used to factor in the number of requests that are blocked and immediately tried again. The Erlang B looks at traffic loading in peak loading times and the Erlang C refines further elements of this by looking at queuing aspects. More still, The Erlang C model assumes that not all requests may be handled immediately and some requests are queued until they can be handled.

The Erlang C model is used by fog networks as well to determine how many staff or request gateways are needed, based on the number of requests per hour, the average duration of the request and the length of time requests are left in the queue. Its approach is difficult to determine because there are more interdependent variables. The Erlang C is nevertheless very important to determine if a request Centre is to be set up, as technologies do not like being kept waiting interminably, as so often happens. Since there are infinitely many waiting positions, the waiting positions can never be exhausted. Part of the definition of the Erlang C queue specifies how waiting customers are selected for service: we assume that whenever there exist waiting customers, the longest-waiting customer will always be the next customer to begin service. In other words, we assume a *FIFO* queueing discipline. Originally, Erlang C defined as the formula that can all you analyze to deal with the probabilities of the fog networks setup depicted with the traffic intensity (A) following the number of inputs (N) around.

$$P_b=\frac{\frac{A^N}{N!}\frac{N}{N-A}}{\left(\sum_{i=0}^{N-1}\frac{A^i}{i!}\right)+\frac{A^N}{N!}\frac{N}{N-A}} \qquad (24)$$

The functionality of Erlang C taken to be $\mathbf{C(n,x)}$ is defined by

$$C(\mathbf{n},\mathbf{x}) = \mathbf{n}*\mathbf{B}(\mathbf{n},\mathbf{x}) \ / \ (\mathbf{n}-\mathbf{x}*(1-B(\mathbf{n},\mathbf{x})))$$

Where,

$$B(n,x)=\left(x^n/n!\right)/\left(1+x+\frac{\mathbf{x}^2}{2!}+\frac{\mathbf{x}^3}{3!}+...+\frac{x^n}{n!}\right)$$

is the Erlang B functionality, The Erlang C

Journal of Advance Research in Mobile Computing
Volume 1 Issue 3
DOI: http://doi.org/10.5281/zenodo.3591228

function computes the probability that an arriving customer in the Erlang C queuing model will find that all servers are busy? The same as the fraction of arriving request that is delayed before beginning service for instance in Table 2.

Table.2:-*Mathematical understanding the basic erlang functionality*

Shrinkages	**30 (measured in %ages)**
Maximum occupancy	85 (measured in %ages)
Target Answer Duration	each 20 (measured in seconds)
Average handling Duration	180 seconds (3 minutes)
Number of requests	100 requests in 1 hour
Number of Incoming trunks	107

To use Erlang B to estimate trucking blockage (also called resource congestion) using Table 2. Let us consider the assumption further that requests arrive in a Poisson process and those requests lengths are exponentially distributed. Finally, assume that if a user finds that all trunks are busy, then the requests hang up and go away. We use the Erlang B function as follows. First, we compute the offered load x in Erlangs.

$$x = \frac{(\text{requests per period})*(\text{requests length})}{(\text{period length})}$$

Then the Erlang blockage b is going to be b=ErlbBlockage(nsrv, trafficInErlangs), let us simplify Erlang C more mathematically, let us use Q and P, separate the formula into segments (complement is needed on the denominator)

$$P_b = \frac{Q}{(P+Q)} \qquad (25)$$

Then, (complement is needed on the denominator A)

$$Q = \frac{A^N}{N!}\frac{N}{N-A},$$

$$P = \left(\sum_{i=0}^{n-1}\frac{A^i}{i!}\right)$$

$$P_b = \frac{\frac{A^N}{N!}\frac{N}{N-A}}{\left(\sum_{i=0}^{n-1}\frac{A^i}{i!}\right)+\frac{A^N}{N!}\frac{N}{N-A}}$$

$$P_b = \frac{Q}{P+Q}$$

Take a look at Q,

$$Q = \frac{A^N}{N!}*\left[\frac{N}{N-A}\right] = \frac{10^{11}}{11!}*\left[\frac{11}{11-10}\right] = \frac{10^{11}}{11!}*\left[\frac{11}{1}\right]$$

Now, we take it time to have the summation, P means that the total is

$A^! / i!$

from 0, so we take this N-1

$$P = \left(\sum_{i=0}^{n-1}\frac{A^i}{i!}\right) = \frac{A^0}{0!}+\frac{A^1}{1!}+\frac{A^2}{2!}+\frac{A^3}{3!}+\frac{A^4}{4!}+---\frac{A^{10}}{10!}$$

Table.3:-*Confirmation results obtained from (27)*

i	i!	A^i	$\frac{A^i}{i!}$	$\sum\frac{A^i}{i!}$
0	1	1	1	1
1	1	10	10	11
2	2	100	50	61
3	6	1000	166.7	227.7
4	24	10000	416.7	644.3
5	120	100000	833.3	1477.7
6	720	1000000	1388.9	2866.6

So, by the completion of all Table 3, we are going to involve Q and P into (26), (complement is needed on the denominator)

$$P_b = \frac{Q}{(P+Q)}$$

$$P_b = \frac{27557}{(12842+27557)}$$

The service level is easily obtained by

$$1 = \left[P_b * e^{-\left[(N-A)*(\text{Target Duration}/AHT)\right]} \right]$$

This is 68.2 % when converted to a percentage. We need to prove these two formulas

$$P_b = \frac{\frac{A^N}{N!}\frac{N}{N-A}}{\left(\sum_{i=0}^{N-1}\frac{A^i}{i!}\right) + \frac{A^N}{N!}\frac{N}{N-A}}$$

And our second is given is

$$P_b = \frac{\frac{n^x}{x!}}{\frac{n^x}{x!} + (1-\rho)\left(\sum_{i=0}^{N-1}\frac{n^i}{i!}\right)}$$

At the condition that

$$\rho = \frac{n}{x}$$

Let's us interchange the formula provided above as

$$P_b = \frac{\frac{n^x}{x!}}{\frac{n^x}{x!} + \left(1-\frac{n}{x}\right)\left(\sum_{i=0}^{N-1}\frac{n^i}{i!}\right)}$$

Keeping our A and N as early defined, we can substitute the equation

$$P_b = \frac{\frac{A^N}{N!}}{\frac{A^N}{N!} + \left(1-\frac{A}{N}\right)\left(\sum_{i=0}^{N-1}\frac{A^i}{i!}\right)}$$

Notably, we may write our denominators and numerators as also part of the equation too as

$$\left(1-\frac{A}{N}\right) = \frac{N}{N} - \frac{A}{N} = \frac{N-A}{N}$$

Lastly, on this formula, we can divide the top with the bottom half of the equation simply by

$$\frac{(N-A)}{N}$$

Therefore,

$$P_b = \frac{\frac{A^N}{N!}}{\frac{A^N}{N!} + \left(\frac{N-A}{N}\right)\left(\sum_{i=0}^{N-1}\frac{A^i}{i!}\right)}$$

To have this result,

$$P_b = \frac{\frac{A^N}{N!}/\frac{N-A}{N}}{\frac{A^N}{N!}/\frac{N-A}{N}\left(\sum_{i=0}^{N-1}\frac{A^i}{i!}\right)}$$

This means that we divided the top and the bottom by

$$\frac{N-A}{N}$$

Since the dividing by

$$P_b = \frac{\frac{A^N}{N!}\frac{N}{N-A}}{\frac{A^N}{N!}\frac{N}{N-A} + \left(\sum_{i=0}^{N-1}\frac{A^i}{i!}\right)}$$

To obtain our original formula as stated above, since P+Q is the same to say that Q+P, from the same perspective, we can provide our formula as

$$P_b = \frac{\frac{A^N}{N!}\frac{N}{N-A}}{\left(\sum_{i=0}^{N-1}\frac{A^i}{i!}\right) + \frac{A^N}{N!}\frac{N}{N-A}}$$

Generations of telephone engineers have successfully used the Erlang B model to predict blockage in trunk groups, although no telephone trunk group exactly matches the assumptions of Erlang B. Perhaps the most questionable assumption is that when a request is blocked, the customer then goes away, never to return.

RESULTS AND DISCUSSION

In section, we provide result simulated in MATLAB for equations expressions are presented line to performance analysis. The model has precise formulas that describe it completely.

Journal of Advance Research in Mobile Computing
Volume 1 Issue 3
DOI: http://doi.org/10.5281/zenodo.3591228

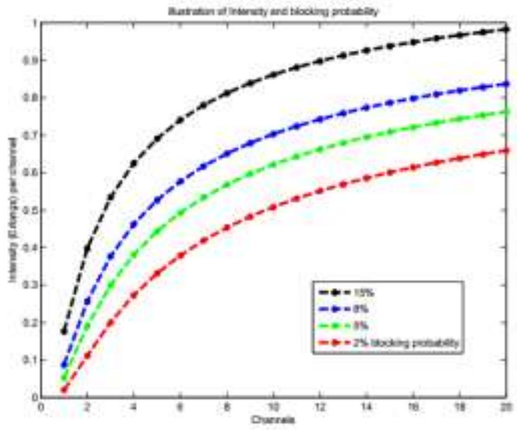

***Fig.2:**-Performance Model using intensity and channels for the city resource.*

Model Analysis on Delay and Waiting Duration

The probability of delay for incoming requests in a network. Suppose that we have requested in the network with these parameters during a certain period as early showed [40]. Assume further that the request arrives in a Poisson process, that the handle times are exponentially distributed, and that the queue of waiting requests is processed in a FIFO manner.

Exponentially distributed, and that the queue of waiting requests is processed in a FIFO manner. Finally, assume that if all agents are busy, and then the users will wait until an agent picks up the request. We are interested in estimating the average wait time t for all users, where they who are served immediately are included in the average with their wait time taken to be 0.
The average wait time or duration is,
t = ErlcWait4 (nsrv, seconds per period, requests PerPeriod, ahtSeconds)
The illustrations of model below show that use the queuing in the EPM model to help understand the functioning of a group of agents taking incoming requests. As in any application of queuing theory including a queuing model as illustrated in [41]. in Figure 2, a real-world system, and mapping of the queuing model to the real-world system, the model computation resulted in this conclusion.

Basing on the results, it shows that the performance evaluation of the mobile edge IoTs decreases with the increasing number of channels, at the level of 5 channels illustrates the better performance and beyond this, level the performance degrades. The increase of channels without an increase in resource allocation will not increase traffic intensity per channel. The Erlang is a particularly important element however; one must be aware of its limitations and apply the findings of any work using Erlangs or functions with a certain amount of practical knowledge. While in the model system, Users either occupy a waiting position or shares a server position with a server where service is shared. When requests arrive at the system, the technology users look to see if any server positions have no customers in them as illustrated in Figure 3.

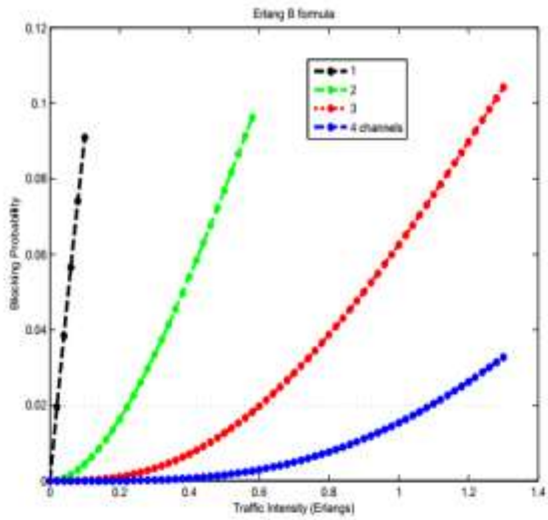

***Fig.3:**-Model Analysis using traffic intensity and blocking possibilities of resources.*

Some of the assumptions of the model covered also a finite number of trunks, hence only a finite number of places where requests can be parked while waiting for service [42]. Nevertheless, if the number of trunks is quite large, then as a practical matter, there may be no situation when all trunks are busy, hence the assumption of

Journal of Advance Research in Mobile Computing
Volume 1 Issue 3
DOI: http://doi.org/10.5281/zenodo.3591228

infinitely best performance may be true as similarly, the assumptions of Poisson arrivals and exponential service basing on the model at times will not hold exactly.

From the illustration still, conclusively depicts that requests of mobile edge computing arrive at the buffer circuits, other user receiving service occupy all server positions. In that case, the arriving request looks to see if any waiting positions are unoccupied implying that, they are going to be given access to the cloud and in the due, cause it evaluates performance [43].

There is more need to have device (IoTs) harmonization in technologies as an attempt to ensure secure communication since today some devices do not sense others on the fog/edge. There is a great chance of requests being queued within the buffer channels, blocked or even lost during busy duration when the channel is fully used-up. The integration of fog computing and other IoT devices will be advantageous to different IoT application platforms since provided results are made based on assumptions of different conditions of the equations as illustrated.

CONCLUSION

In computing, the discipline the queuing dates back years with FIFO approach, ideally**,** the servers become idle and wait until an arriving request joins the server to begin service from the cloud. It is revealed further that the Erlang formulas are still essential in communication areas that handle communication probabilities and rates despite the widespread use of Erlang concepts and formulae, it is necessary to remember that there are limitations to their use. More still, the servers operate inside circuits containing a finite number of *positions.* Each server occupies positions not containing a server are called waiting for positions or buffers, the proposed model is depicted as optimized computation. Within our future work, this study has inspired us to work on the communication efficiency is in wireless computing including Unmanned Ariel Vehicles in hard to reach areas with the idea of securing safe data transfer and better beam selections.

ACKNOWLEDGEMENT

This research has been supported by Yazd University. The authors would like to acknowledge the Department of Computer Engineering for the support and sharing of their commentaries for us to attain this paper quality.

REFERENCES

1. Z. Ning, X. Kong, F. Xia, W. Hou, and X. Wang, "Green and a sustainable cloud of things: Enabling collaborative edge computing," *IEEE Communications Magazine*.2019.57(1):72–78p.
2. T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Transactions on Vehicular Technology*.2019.68(1): 856–868p.
3. C. Li, J. Bai, and J. Tang, "*Joint optimization of data placement and scheduling for improving user experience in edge computing,"* Journal of Parallel and Distributed Computing.2019.125:93–105p.
4. A. Brogi, S. Forti, and A. Ibrahim, "Predictive analysis to support fog application deployment,*"* Fog and Edge Computing: Principles and Paradigms.2019.191–221p.
5. S. Li, D. Zhai, P. Du, and T. Han, "*Energy-efficient task offloading, load balancing, and resource allocation in mobile edge computing enabled IoT networks*," Science China Information Sciences.2019.62(2):29307p.
6. T. T. Nguyen, L. Le, and Q. Le-Trung, "*Computation offloading in MIMO based mobile edge computing systems*

under perfect and imperfect CSI estimation," IEEE Transactions on Services Computing, 2019.
7. J. Ren, Y. He, G. Huang, G. Yu, Y. Cai, and Z. Zhang, "*An Edge-Computing Based Architecture for Mobile Augmented Reality*," IEEE Network, 2019.
8. S. Ammirato, F. Sofo, A. M. Felicetti, and C. Raso, "*A methodology to support the adoption of IoT innovation and its application to the Italian bank branch security context*," European Journal of Innovation Management.2019.22(1):146–174p.
9. B. Afzal, M. Umair, G. A. Shah, and E. Ahmed, "*Enabling IoT platforms for social IoT applications: vision, feature mapping, and challenges,"* Future Generation Computer Systems,2019.92:718–731p.
10. V. V. Sood, S. Sharma, and R. Khanna, "*Performance Evaluation of Cognitive Internet of Things in Asynchronous Distributed Space-Time Block Codes over Two-Wave Diffuse Power Fading Channel*," in *Engineering Vibration, Communication and Information Processing*, Springer, 2019,. 131–141p.
11. D. Dutta, "IEEE 802.15. 4 as the MAC Protocol for the Internet of Things (IoT) Applications for Achieving QoS and Energy Efficiency," in *Advances in Communication, Cloud, and Big Data*, Springer, 2019, 127–132p.
12. P. Maiti, J. Shukla, B. Sahoo, and A. K. Turuk, "Mathematical Modeling of QoS-Aware Fog Computing Architecture for IoT Services," in *Emerging Technologies in Data Mining and Information Security*, Springer, 2019, 13–21p.
13. A. A. Multan, M. K. A. Ghani, N. Arunkumar, M. A. Mohamed, and O. Mohd, "Enabling technologies for fog computing in healthcare IoT systems," *Future Generation Computer Systems*, 2019.90:62-78p.
14. W. Wang, G. Wu, Z. Guo, L. Qian, L. Ding, and F. Yang, "Data Scheduling and Resource Optimization for Fog Computing Architecture in Industrial IoT," in *International Conference on Distributed Computing and Internet Technology*, 2019, 141–149p.
15. F. A. Teixeira, F. M. Pereira, H.-C. Wong, J. M. Nogueira, and L. B. Oliveira, "*SIoT: Securing Internet of Things through distributed systems analysis,"* Future Generation Computer Systems.2019.92:1172–1186p.
16. M. G. Borlick, L. M. Gupta, R. G. Hathorn, and K. A. Nielsen, "*Method, system, and computer program product for distributed storage of data in a heterogeneous cloud*," Jan-2019.
17. M. Dabbagh and A. Rayes, *"Internet of things security and privacy*," in the Internet of Things From Hype to Reality, Springer, 2019:211–238p.
18. X. Li, Q. Wang, X. Lan, X. Chen, N. Zhang, and D. Chen, "*Enhancing Cloud-Based IoT Security through Trustworthy Cloud Service: An Integration of Security and Reputation Approach," IEEE Access*, 2019.
19. M. Jhamb, T. Dhall, and T. Verma, "Performance Analysis of Comparator for IoT Applications," in *Advances in Signal Processing and Communication*, Springer, 2019,501–509p.
20. M. Martonosi, "*New Metrics and Models for a Post-ISA Era: Managing Complexity and Scaling Performance in Heterogeneous Parallelism and Internet-of-Things*," ACM SIGMETRICS Performance Evaluation Review, 2019.46(1):20–20p.
21. S. Mishra, S. Sahoo, and B. K. Mishra, "*Addressing Security Issues and Standards in Internet of Things,"* in Emerging Trends and Applications

in Cognitive Computing, IGI Global, 2019.224–257p.
22. F. Al-Turjman, "*5G-enabled devices and smart-spaces in social-IoT: an overview*," *Future Generation Computer Systems*, 2019,92:732–744p.
23. S. Williams, C. Hardy, and P. Nitschke, "Configuring the Internet of Things (IoT): A Review and Implications for Big Data Analytics," in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.
24. R. Casadei, G. Fortino, D. Pianini, W. Russo, C. Savaglio, and M. Viroli, "*Modelling and simulation of Opportunistic IoT Services with Aggregate Computing,"* Future Generation Computer Systems.2019.91:252–262p.
25. G. Casalino, G. Castellano, V. Pasquadibisceglie, and G. Zaza, "Contact-Less Real-Time Monitoring of Cardiovascular Risk Using Video Imaging and Fuzzy Inference Rules," *Information*, 2018.10(1):9p.
26. R. K. Ambhati, S. Y. Chaudhari, and M. Jain, "LD-IoT: Long-Distance Outdoor Networking for 802.11 ah-Based IoT," in *Recent Findings in Intelligent Computing Techniques*, Springer, 2019:375–384p.
27. K. Poularakis, J. Llorca, A. M. Tulino, I. Taylor, and L. Tassiulas, "Joint Service Placement and Request Routing in Multi-cell Mobile Edge Computing Networks," *arXiv preprint arXiv:1901.08946*, 2019.
28. M. G. R. Alam, M. M. Hassan, M. Zi. Uddin, A. Almogren, and G. Fortino, *"Autonomic computation offloading in mobile edge for IoT applications,"* Future Generation Computer Systems, 2019.90:149–157p.
29. S. SOO, C. Chang, S. W. Loke, and S. N. Srirama, "*Dynamic Fog Computing: Practical Processing at Mobile Edge Devices," in* Algorithms, Methods, and Applications in Mobile Computing and Communications, IGI Global, 2019.24–47p.
30. J. Luo *et al.*, "*Container-based fog computing architecture and energy-balancing scheduling algorithm for energy IoT*," Future Generation Computer Systems, 2019.
31. A. Yassine, S. Singh, M. S. Hossain, and G. Muhammad, "*IoT big data analytics for smart homes with fog and cloud computing,"* Future Generation Computer Systems.2019.91:563–573p.
32. A. Viejo and D. Sánchez, "Secure and privacy-preserving orchestration and delivery of fog-enabled IoT services," *Ad Hoc Networks*, 2019.82:113–125p.
33. M. S. Mekala and P. Viswanathan, "*Energy-Efficient virtual machine selection based on resource ranking and utilization factor approach in cloud computing for IoT*," Computers & Electrical Engineering.2019.73:227–244p.
34. D. Kim, K. Park, Y. Park, and J.-H. Ahn, "*Willingness to provide personal information: Perspective of privacy calculus in IoT services,"* Computers in Human Behavior.2019.92:273–281p.
35. S. Ganapathy, "*A Secured Storage and Privacy-Preserving Model Using CRT for Providing Security on Cloud and IoT based Applications*," Computer Networks, 2019.
36. N. Bocken, E. Ingemarsdotter, and D. Gonzalez, "*Designing Sustainable Business Models: Exploring IoT-Enabled Strategies to Drive Sustainable Consumption*," in *Sustainable Business Models: Innovation, Implementation and Success*, A. Aagaard, Ed. Cham: Springer International Publishing, 2019:61–88p.
37. S. Ganapathy, "A Secured Storage and Privacy-Preserving Model Using CRT for Providing Security on Cloud and

Journal of Advance Research in Mobile Computing
Volume 1 Issue 3
DOI: http://doi.org/10.5281/zenodo.3591228

IoT based Applications," *Computer Networks*, 2019.

38. W. Wang, Q. Zhang, H. Lin, M. Liu, X. Liang, and Q. Liu, "*Wireless Energy Transmission Channel Modeling in Resonant Beam Charging for IoT Devices,"* IEEE Internet of Things Journal, *2019.*

39. Y. Yang *et al.*, "*GAN-based Semi-supervised Learning Approach for Clinical Decision Support in Health-IoT Platform*," IEEE ACCESS, 2019.

40. W.Shafik, S.M.Matinkhah, "*Admitting New Requests in Fog Networks According to Erlang B Distribution*," 27th Iranian Conference on Electrical Engineering *(ICEE)*, 2019,.2016–2021p.

41. S. Mostafavi, W. Shafik. *Fog Computing Architectures, Privacy and Security Solutions*. J. Commun. Technol. Electron. Comput. Sci. 2019. 24:1–14p.

42. W.Shafik, S.M.Matinkhah. *How to use Erlang B to Determine the Blocking Probability of Packet Loss in a Wireless Communication.* Presented at the 13th Symposium on Advances In Science & Technology. 2018.

43. W.Shafik, S.M.Matinkhah. *Privacy Issues in Social Web of Things*. 5th International Conference on Web Research (ICWR). 2019:208–214p.

Cite this article as: Wasswa Shafik, S. Mojtaba Matinkhah, & Mohammad Ghasemazade. (2019). Fog-Mobile Edge Performance Evaluation and Analysis on Internet of Things. Journal of Advance Research in Mobile Computing, 1(3), 1–17. http://doi.org/10.5281/zenodo.3591228