

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/308861149>

Fog Computing Micro Datacenter Based Dynamic Resource Estimation and Pricing Model for IoT

Conference Paper · March 2015

DOI: 10.1109/AINA.2015.254

CITATIONS

44

READS

171

2 authors:



Mohammad Aazam

Carnegie Mellon University Qatar

50 PUBLICATIONS 478 CITATIONS

[SEE PROFILE](#)



Eui-nam Huh

Kyung Hee University

304 PUBLICATIONS 2,273 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Real-Time Mobile Cloud Service Platform [View project](#)



Development of Modularized In-Memory Virtual Desktop System Technology for High Speed Cloud Service [View project](#)

All content following this page was uploaded by **Mohammad Aazam** on 28 January 2017.

The user has requested enhancement of the downloaded file.

Fog Computing Micro Datacenter Based Dynamic Resource Estimation and Pricing Model for IoT

Mohammad Aazam
Computer Engineering Department
Kyung Hee University, Suwon
South Korea
aazam@ieee.org

Eui-Nam Huh
Computer Engineering Department
Kyung Hee University, Suwon
South Korea
johnhuh@khu.ac.kr

Abstract— Pervasive and ubiquitous computing services have recently been under focus of not only the research community, but developers as well. Prevailing wireless sensor networks (WSNs), Internet of Things (IoT), and healthcare related services have made it difficult to handle all the data in an efficient and effective way and create more useful services. Different devices generate different types of data with different frequencies. Therefore, amalgamation of cloud computing with IoTs, termed as Cloud of Things (CoT) has recently been under discussion in research arena. CoT provides ease of management for the growing media content and other data. Besides this, features like: ubiquitous access, service creation, service discovery, and resource provisioning play a significant role, which comes with CoT. Emergency, healthcare, and latency sensitive services require real-time response. Also, it is necessary to decide what type of data is to be uploaded in the cloud, without burdening the core network and the cloud. For this purpose, Fog computing plays an important role. Fog resides between underlying IoTs and the cloud. Its purpose is to manage resources, perform data filtration, preprocessing, and security measures. For this purpose, Fog requires an effective and efficient resource management framework for IoTs, which we provide in this paper. Our model covers the issues of resource prediction, customer type based resource estimation and reservation, advance reservation, and pricing for new and existing IoT customers, on the basis of their characteristics. The implementation was done using Java, while the model was evaluated using CloudSim toolkit. The results and discussion show the validity and performance of our system.

Index Terms—IoT; Cloud of Things; Fog computing; Edge computing; Micro Data Center; resource management.

I. INTRODUCTION

Internet of Things (IoT) has revolutionized the future of connectivity and reachability. ‘Things’ in IoT represent any object on the face of the Earth, which may or may not be able to communicate on its own. From a smart device to a leaf of a tree or a bottle of beverage, anything can be part of Internet with IoT. Dumb objects become communicating nodes over the Internet, through data communication means, mainly through Radio Frequency Identification (RFID) tags. IoT also includes smart objects. Smart objects are those objects that are not only physical entities, but digital as well, capable of performing some tasks for

humans and the environment. This is why, IoT is not only hardware and software paradigm, but also comprises interaction and social aspects [1].

IoT works on the basis of Machine-to-Machine (M2M) communications, but not limited to it. M2M refers to communication between two machines, without human intervention. In IoT, even non-connected entities can become part of it, with a data communicating device, like a bar-code or an RFID tag, sensed through a device (which may be a smart phone sensing it), which eventually is connected to the Internet.

IoT oriented services are gaining momentum rapidly. Number of connected devices already exceeded the number of people on Earth since 2011. Connected devices have reached 9 billion and are expected to grow more rapidly and reach 24 billion by 2020 [2]. With increasing number of heterogeneous devices connected to IoT and generating data, it is going to be impossible for a standalone IoT to efficiently perform power and bandwidth hungry tasks. IoT and cloud computing amalgamation has been envisioned in this regard [3] [4]. There comes a situation when cloud is connected with an IoT that generates multimedia data. Visual Sensor Network or CCTV connected to cloud can be examples of such scenario. Since multimedia content consumes more processing power, storage space, and scheduling resources, its management in cloud will become inevitable. Besides, mission critical and latency sensitive IoT services require a very quick response and processing. In that case, it is not feasible to communicate through distant cloud, over the Internet. Fog computing plays a very vital role hereby [5]. Fog Computing brings networking resources near the underlying networks, by extending the traditional Cloud Computing paradigm to the edge of the network, enabling creation of refined and better applications or services [6]. Fog is an Edge Computing and Micro Datacenter (MDC) paradigm, suitable for IoTs and WSNs.

In this paper, we present a service oriented resource management model for IoT devices, through Fog, which can help in efficient, effective, and fair management of resources. We have categorized IoT devices into three, based on the nature of device and mobility factor, and provide resource management accordingly. Our work is mainly focused on customer type and device based resource estimation and pricing. We have

considered different traits and characteristics of customers in this regard, which makes our model more flexible and scalable.

In rest of the paper, existing works are discussed in section II. Section III explains about Fog computing. Section IV presents resource management model. Section V is on performance evaluation. Section VI concludes the paper.

II. RELATED WORK

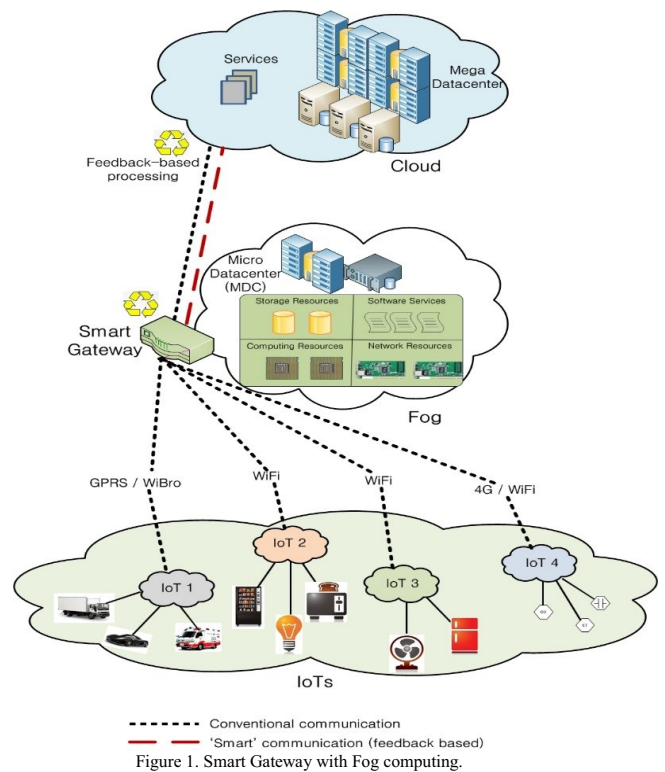
Fog computing is a newly introduced paradigm, therefore, no standard architecture is available regarding management of resources. Already done studies are limited to cloud-only scenarios. Fog or Cloud of Things (CoT) oriented resource management is not yet addressed.

Rogers Owen et al. present a resource allocation mechanism [7], but resource prediction and detailed billing, along with refunding issue is not considered. Their study is also only limited to basic cloud resource management. Park Ki-Woong et al. [8] present a billing system with some security features. To resolve different types of disputes in future, a mutually verifiable billing system is presented. Their work only focuses on the reliability of transactions made in purchasing and consuming resources. They do not focus on the overall resource management, pricing, refunding, or similar important feature, specially for CoT. Wang Wei et al. [9] propose a brokerage service for reservation of instances. The authors propose a brokerage service for on-demand reservation of resources, for IaaS clouds. Their work is limited to only on-demand jobs and they do not present anything beyond that. Yang Yichao et al. present resource allocation algorithm in a simplistic way [10]. Deelman Ewa et al. present performance tradeoffs of different resource provisioning plans. They also present tradeoffs in terms of storage fee of Amazon S3 [11]. Their work does not take into account pricing strategies and other resource management tasks. Shadi Ibrahim et al. present the concept of fairness in pricing in respect of micro-economics [12], not discussing how pricing should be done for different types of services. Their work is only limited to micro-economics pricing aspect. Kan Yang et al. present [13] a dynamic auditing protocol for ensuring the integrity of stored data in the cloud. They present an auditing framework for cloud storage. Zhen Xiao et al. present [14] a resource allocation system that uses virtualization technology to dynamically allocate resources, according to the demands of the service. In their study, they present measuring the unevenness in resource utilization. IoT based environment is not considered in this study. D. Cenk Erdil, in [15], presents an approach for resource information sharing through proxies. Situations where clouds are distant and there is no direct control, proxies can be used to make resource information available to them. This study only focuses on the importance of resource information sharing. Rakpong et al. consider resource allocation in mobile cloud computing environment in their work [16]. They discuss about communication/radio resources and computing resources, but their work only focuses on decision making for coalition of resources, to increase service provider's revenue.

Flavio Bonomi et al. present [6] basic architecture of Fog computing, which does not include its practical implications and resource management for IoT. Similarly, Salvatore J. Stolfo et al. present [18] data protection through Fog computing, but not going into resource management and related matters.

III. FOG COMPUTING

Fog computing, a Micro Datacenter paradigm, is a highly virtualized platform, which provides computation, storage, and networking services between the end nodes in an IoT and traditional clouds [6]. In contrast to the cloud, which is more centralized, Fog computing targets the services and applications with widely distributed deployments. As shown in the overall architecture in figure 1, Fog will be able to deliver high quality streaming to mobile nodes, like moving vehicles, through proxies and access points positioned accordingly, like, along highways and tracks. Fog suits applications with low latency requirements, emergency and healthcare related services, video streaming, gaming, augmented reality, etc. For smart communication, Fogs are going to play an important role. For many of the tasks a gateway has to perform, it is not possible for it to do effectively being standalone.



The underlying nodes and networks are not always physical. Virtual sensor nodes and virtual sensor networks are also requirements for various services. Similarly, temporary storage, preprocessing, data security and privacy, and other such tasks can be done easily and more efficiently in the presence of a Fog, co-located with the Smart Gateway. Based on the feedback from application, gateway must decide the timings and type of data to be sent. This kind of a gateway, we refer it as “Smart Gateway” [4], [5], would help in better utilization of network and cloud resources. The data collected from WSNs and IoTs will be transmitted through gateways to cloud. The received data is then stored in the cloud and from there, it is provided as a service to the users.

Being localized, Fog provides low latency communication and more context awareness. Fog computing allows real-time delivery of data, specially for delay sensitive and healthcare related services. It can perform preprocessing and notify the cloud, before cloud could further adapt that data into enhanced services. With heterogeneous nodes, heterogeneous type of data would be collected. Interoperability and transcoding becomes an issue then. Fog plays a very vital role in this regard. Likewise, IoT and WSN federation, in which two or more IoTs or WSNs can be federated at one point, can be made possible through the Fog. This will allow creation of rich services.

Keeping in view the basic tasks Fog can provide, its overall layered architecture is presented in figure 2.

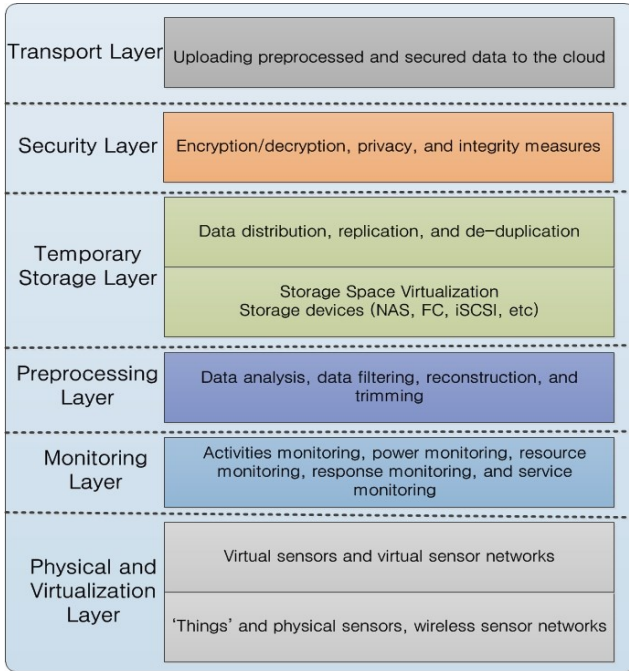


Figure 2. Layered architecture of Fog.

At Physical and Virtualization layer, physical nodes, WSNs, virtual nodes, and virtual sensor networks are managed and maintained, based on their types and service requirements. Monitoring layer monitors the activities of the underlying nodes and networks. Which node is performing what task, at what time, and what is required from it next, is monitored here. Other than this, the power constrained devices or nodes are monitored on their energy consumption basis as well, so that effective measures can be taken in time. Preprocessing layer performs data management related tasks. It analyzes the collected data, performs data filtering, trimming, and in the end, more meaningful and necessary data is generated. Data is then temporarily stored in the Fog resources. Once the data is uploaded in the cloud and it is no more required to be stored locally, that data is then removed from the storage media. IoTs and WSNs may generate some private data as well. Ubiquitous healthcare and smart healthcare services generate private data of the patients. Similarly, location aware data may also be sensitive in some cases, which requires to be made secure. This is where Security layer comes into play. In the end, at Transport layer, the ready-to-send data is sent to the cloud, burdening the core to the minimum and allowing cloud create more useful services.

IV. FOG-BASED IOT RESOURCE MANAGEMENT MODEL

Sensors, IoT nodes, devices, and Cloud Service Customers (CSCs) contact Fog to acquire the required service(s) at best price. CSCs perform the negotiation and service level agreement (SLA) tasks with Fog. Once the contract is agreed upon, the service is provided to the customer. In this regard, Fog not only provides services on ad hoc basis, but also, it has to estimate consumption of resources, so that they can be allocated in advance. Estimating resources beforehand creates efficiency and fairness in the management of resources. As mentioned, the requests can be made from objects or nodes as well as devices operated by people. Therefore, prediction and pre-allocation of resources also depend upon user's behavior and its probability of using those resources in future. For this purpose, Fog performs pricing and billing accordingly, which is also presented in this section.

We formulate the estimation of required resources as:

$$\mathfrak{R} = \sum_{i=0}^n ((U_i * ((1 - \bar{x}(P_i(L|H)_s)) - \sigma^2) * (1 - \Omega_i) * \Phi) \quad (1)$$

$$\mathfrak{R} \in \{CPU, memory, storage, bandwidth\}$$

Where \mathfrak{R} represents required resources, U_i is the basic price of the requested service i . In most of the cases, U_i is decided at the time contract is being negotiated. $\bar{x}(P_i(L|H)_s)$ is the average of service oriented relinquish probabilities of a particular customer of giving up the same resource which it has requested now. In case the customer is requesting this service for the first time, the default value set for $\bar{x}(P_i(L|H)_s)$ is 0.3. Because, the average of

low relinquish probability (0.1 to 0.5, from complete range of 0.1 to 0.9) is 0.3. For simplicity, we have categorized customers into two types, one having low (L) giving up probability and the other having high (H) giving up probability. But this grouping is applicable only when the customer has requested for a service for the first time and no probability record is available. Where,

$$0 < L \leq 0.5, 0.5 < H \leq 1 \quad (2)$$

$$\Omega_i = \begin{cases} \bar{x} (\bar{x} (\sum_{k=0}^n P(L|H)_k)), P(L|H)_{last} & \text{if } n > 0, \\ 0.3 & \text{if } n = 0 \end{cases} \quad (3)$$

σ^2 is the variance of service oriented relinquish probabilities. CSCs, specially mobile users, can have a very fluctuating behavior in utilizing resources, which may lead to deception, while making decision about resource allocation. That is why, in our model, we have taken into account variance of relinquish probabilities, which helps determining the actual behavior of each customer.

Ω is a constant decision variable value, which is assigned by the Fog resource manager to each user, according to CSC's historical record of overall relinquish probabilities. Here, it should be noted that $P_i(L|H)_s$ determines probability of the same service, which customer is requesting currently, while Ω is overall probability, including all activities a particular customer has been doing. Most recent behavior is determined from the last relinquish probability. That is why, it is given more importance and the average is taken again, by adding last relinquish probability. In case of a new user, when there no previous data for that user, this value is set at default low relinquish probability 0.3.

ϕ represents the type of accessing device. Accessing device is only meant to access a particular service, therefore, device capabilities, like CPU and memory are not currently considered in our model. However, for multimedia services, display size has a role to play, because the Fog has to allocate resources according to the display size and appropriate required quality. Device's mobility also comes into play here. A mobile device would require more resources from the Fog, because it is in motion and requires quick response. Relatively more resources are required in this case, so that efficient transcoding and data delivery is made possible. Laptop can be used in static mode as well as mobile. In our model, we have taken into account the access network in case of laptop, to make sure which mode, static or mobile, is being used. Because, if laptop is always considered as a mobile device while it is being used in static mode, precious datacenter resources would be wasted. Based on our real experiments in different wired and wireless networks (broadband, WiFi, WiBro, 3G, and 4G LTE-A) [17], we came to the conclusion that smartphone and similar devices would require approximately 1.25 of the resources reserved for static

device (desktop computer or laptop in static mode). On the other hand, large mobile device (tablet and laptop) requires approximately 1.5 times of such resources. Main focus should be to give priority to service accessed by mobile devices, instead of doubling the resources. Therefore, the value of ϕ would accordingly be:

$$\phi = \begin{cases} \phi_{m_s} = 1.25, \\ \phi_{m_l} = 1.5, \\ \phi_s = 1 \end{cases} \quad (4)$$

Where ϕ_{m_s} represents small mobile device, ϕ_{m_l} represents large mobile device, and ϕ_s represents static device.

With this formulation, Fog can determine future resource requirements. It is important for Fog to rightly decide about reserving resources and prevent precious resources go waste. It will also help power consumption management, which is becoming a point of concern for datacenters.

An ongoing service can be discontinued at any stage by the customer. Specially with mobile and handheld devices, the service consumption is more random. At that point, Fog has to halt the service and refund the remaining amount to the customer. In this case, Fog has to take into account the utilized resources or consumed services and the remaining service value of the decided total initial service. This can be formulated through the following equations.

$$\partial_t = \partial_{un} + \partial_{deg} \quad (5)$$

$$\lambda_{un} = 1 - \frac{\alpha}{100} \quad (6)$$

$$\partial_{un} = \begin{cases} \partial_{unAPP} & \text{if } \alpha \geq 60\%, \\ \partial_{unDEP} & \text{if } \alpha < 60\% \end{cases} \quad (7)$$

In eq. 5, ∂_t is the total amount to be refunded. ∂_{un} is the refund amount of unutilized resources. ∂_{deg} is the refund factor to be paid on quality degradation. During service delivery, it is not always possible to deliver the service exactly according to the promise made during SLA. ∂_{un} is further calculated through equations 8 to 11, while ∂_{deg} , through equation 12. λ_{un} represents unutilized resources, while α represents utilized resources.

Our model also incorporates the concept of incentive for a better customer. For those customers who have used more service, Fog has earned more money from them. When they quit the service, they can be provided with some appreciation amount ∂_{unAPP} , while refunding. We call it Appreciation Index ω . For example, customers who have used 60% or more service are eligible for this. In that case, the refund amount should linearly increase,

encouraging the customer, which in turn works as an appealing factor and allows customer to return to that service provider again and again and consume more services. In this case, the formulation would be:

$$\partial_{unAPP} = ((\lambda_{un} * U_i) - \frac{f}{\alpha}) + \omega \quad (8)$$

$$\omega = \log \alpha \quad (9)$$

$$\partial_{unDEP} = ((\lambda_{un} * U_i) - \frac{f}{\alpha}) + \varepsilon \quad (10)$$

$$\varepsilon = \ln \left(\frac{\alpha}{100} \right) \quad (11)$$

ε is the depreciation index, which deducts some amount, based on business policy, from those customers who used very less service. In our model, we apply Depreciation Index when resource utilization is less than 60%. The reason why we have used natural log \ln is that it produces negative value in accordance with the utilization factor α . Therefore, the final refund amount would actually be depreciated. f is Fog's service ratio (e.g., 10% of the total amount). $\frac{f}{\alpha}$ is used to determine Fog's service ratio according to utilized resources, which is then deducted while refunding the remaining amount.

$$\partial_{deg} = \left(\frac{Q_{SLA}}{Q_a} * \frac{f}{100} \right) * ((U_i - (Q_a * U_i)) * \left(\frac{\alpha}{100} \right)^2) \quad (12)$$

Where Q_a is the acquired quality of service and Q_{SLA} is the promised quality of service, during SLA. $Q_a * U_i$ will produce acquired quality factor according to the utilized service, which is then subtracted from the actual price U_i . The factor varies according to the type and price of service, generating quality degradation factor accordingly. $\left(\frac{\alpha}{100} \right)^2$ determines the amount of utilized resources.

V. VALIDITY AND PERFORMANCE EVALUATION

In this section, we present validity of our proposed service model. We defined our model through algorithm to evaluate the effectiveness in CoT business. Our main objective is to observe the influence of performance factors on the systems and test the feasibility of our method.

A. Evaluation Setup

Table 1 shows the evaluation environment for our model's test. We have considered different parameters to estimate the required resources, pricing and billing for different types of users, and refund amount calculation. Table 2 shows the setting

of basic parameters. Since implementation on real test-beds limits the extent to the scale of the test-bed, which consequently makes it difficult to reproduce the result and analyze in varied scenarios, we chose simulation instead.

TABLE 1: SIMULATION SETUP

System	Intel(R) Core(TM) i5-M430, 2.27 GHz
Memory	4 GB
Implementation language	Java using NetBeans 8.0
Simulator	CloudSim 3.0.3
Operating System (OS)	Window 7 Home Premium

TABLE 2: KEY PARAMETERS' SETTING FOR EVALUATION

Parameters	Range
Service Level Agreement (Q_{SLA})	0.9
Acquired service quality (Q_a)	0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1
Service Price (p)	100,150, 200,250,....,1000
User characteristic (L or H)	$L > 0$ & ≤ 0.5 , $H > 0.5$ & ≤ 1
Unutilized resources (α)	90% ~ 10%
Service utilization	10% ~ 90%
Number of registered services	10
Fog service ratio (f)	10%

B. Resource estimation for an absolutely new IoT customer

When different CSCs are requesting for a particular service, the Fog has to analyze what number of resources have to be allocated for that service, based on the type of customer. For CSCs having low relinquish probability, priority in resource allocation is given. For those customers, who are absolutely new and Fog has no past record for them, default probability is used. In other words, it is assumed that this new customer will be 'somewhat' loyal. That is why, relinquish probability is set to 0.3. While perfectly loyal customer would be having a probability of 0.1. Since cloud and Fog resources are precious and it is not advisable to take risk, hence, instead of assigning 0.1 probability value, we have assigned 0.3, which is the average probability of low relinquish. Figure 3 shows the unit of resources, we call it virtual resource value (VRV), being predicted for new customers, for different types of registered services and devices. This unit is then mapped to actual resources (memory, CPU, storage space, bandwidth, etc.), according to the type of service being offered and policies of a particular CSP. For example, a USD 100 cloud storage collaboration service is more I/O intensive. It requires more CPU as well as storage space. The CSP will map VRV 9 to level one of its resource allocation actual mappings. In case the USD 100 service is related to database queries, then only I/O is intensive, not storage, because it requires read-only process. The CSP will perform mappings accordingly. This is how different units of resources are mapped to actual resources, based on the type of service. For USD 100 service, if the accessing device is static, 9 units of resources are reserved. While, in case of small mobile device, 11 resources, and for large mobile device, 14 resources are reserved. This is how each type of device is catered, according to its requirements. Similarly, for a USD 150 service, if a static device is requesting, 14 units of resources are reserved. On the other hand, small mobile device

gets 18 and large mobile device gets 21 units of resources. Fog is then able to handle different types of IoT devices accordingly.

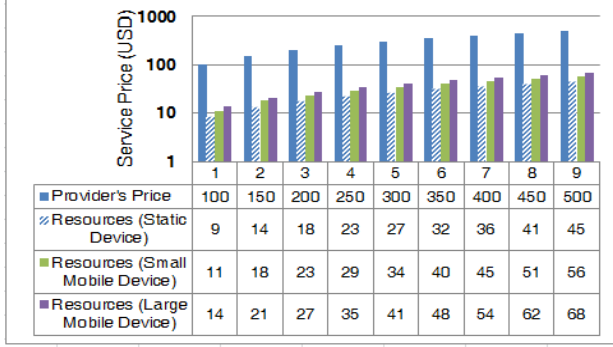


Figure 3. Resource estimation for new CSCs, for different requested services.

Illustrative Scenario:

Figure 4 shows the illustrative scenario, as an example of how mapping can be performed by the CSP, according to its resource pool and the type of service being provided. For YouTube service, S1, VRV 9 is mapped to corresponding resource pool level (RPL). Then according to the type of service being provided, the mapping is performed to the actual resource pool. Among the available resources for the service 1, CSP allocates 10% of CPU, 8% of memory, 0% of storage space, since storage is not required here, and data rate of 300Kbps. The guarantee of allocation of these resources is 80%, which means, at least 80% of resources from the mapping are guaranteed. This is only an example. This mapping would vary according to the type of service and available resource pool of CSP. For the same service requested by small mobile device, 1.25 times resources are increased for each type of resource.

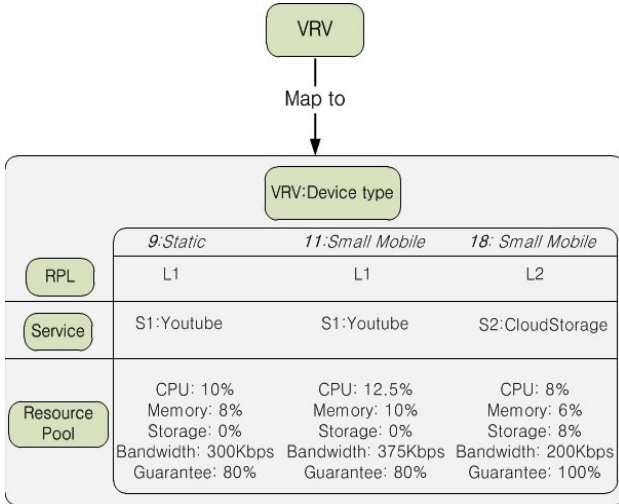


Figure 4. Illustrative scenario of mapping of virtual resource value to the resource pool, according to the type of service.

C. Resource estimation for an existing customer

For the returning/existing customers, Fog already has a historical record of its past activities and probabilities (overall probabilities and service oriented probabilities) with which CSC has been consuming resources. When characteristic of a particular customer is known, it is more justified and fair to determine and allocate resources accordingly. In this way, Fog will be able to reserve right amount of resources and would be having least number of chances to lose profit. Figure 5 shows five different types of CSCs, having different Service Oriented Probability (SOP) and Average Overall Probability (AOP), requesting a particular service S . Comparison is shown on the basis of devices, when each type of CSC requests the same service having different devices. In this example, the result is presented for service price USD 100. The unit is greater for L customers, while it is smaller for H customers, because of their behavior. Since there are more chances of an H customer to relinquish the service(s), hence, more priority and quality is provided to the more loyal customer, having L probability. In case of CSC 1, having SOP = 0.1 (bold font in the figure) and AOP = 0.7, 22 unit of resource are reserved for USD 100 service, when it is requesting from a static device. In case of small mobile device, 28 units of resources would be reserved and for large mobile device, 33 units will be reserved. In case of CSC 2, SOP = 0.2 and AOP = 0.4, 47, 59, and 71 resources are reserved respectively for static, small mobile, and large mobile devices. Even though CSC 2 has relatively higher SOP, as compared to CSC 1, but since its AOP is lower than CSC 1, therefore, it gets more resources. CSC 3 has same AOP (0.7) as that of CSC 1, but has a comparatively higher SOP (0.3). That is why, it gets a smaller amount of allocated resources. CSC 4 and 5 both have an overall perfect loyalty and stability record, having AOP = 0.1. Their resource allocation differ on the basis of SOP. On the basis of currently requested service S , CSC 4 is comparatively loyal, having SOP = 0.4, while CSC 5 has SOP = 0.5. This shows that both these types of probabilities have their impact and final decision is made accordingly, which makes it sure that a CSC who has generally been loyal, but not so in case of some particular service, or vice versa, gets treated in view of that.

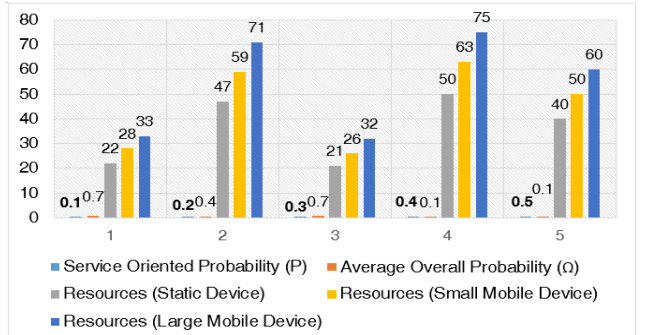


Figure 5. Resource estimation for different types of CSCs, for USD 100 service.

D. Price estimation for new customers

Pricing is an important element in resource management. Due to heterogeneity in types of CSCs and the services offered, pricing has to be dynamic. Pricing cannot be too generic, because it will create unfairness. Based on characteristics of a CSC and type of service requested, prices are determined. CSC can be altogether new, existing but requested service S for the first time, and existing and requested for service S before as well. The third scenario is discussed in subsection E. In case of scenario 1, since broker does not have any previous record of service utilization, it treats the CSC as somewhat reliable and determines pricing on the basis of low relinquish probability. In case of scenario 2, broker has the record of CSC's previous activities and its AOP can be determined. But CSC has requested for current service S for the first time. On the basis of AOP, pricing is done, categorizing CSC in either for Price-L (for low relinquish probability CSC) or Price-H (for high relinquish probability CSC). As shown in figure 6, if CSP has set basic price as USD 100, Price-L for that service would be USD 103.3 (because CSC would get different added services, like refunding, etc.), while USD 107.3 for Price-H. Therefore, CSC is treated accordingly, which in the end creates fairness. This strategy makes sure that if there is enough risk that a particular CSC would relinquish the services, the broker should have earned enough profit, such that when the resources are relinquished, broker does not have to bear loss.

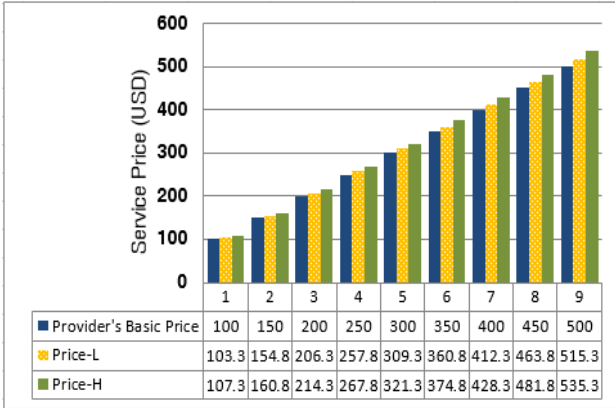


Figure 6. Price estimation, for new customers.

E. Price estimation for existing customer, on the basis of characteristic

In scenario 3, when a CSC has been requesting this service S before as well, price determination is on the basis of profit earned as well, other than relinquish probabilities. The idea is to give more incentive to a CSC from whom CSP has earned more profit. This encourages CSC to utilize more and more service(s) and be more loyal to the CSP. In figure 7, horizontal axis shows different instances of CSCs, while vertical axis shows price in USD. In case of CSC 1, the relinquish probability is high, 63%,

and profit earned so far is low, USD 21. Therefore, for a USD 100 service, this CSC has to pay USD 109.3. Since CSP has the historical data of this CSC, the price is determined in a more realistic and fair way. If we compare this CSC 1 with CSC 1 of subsection D (figure 6), it shows that when a CSC is new, it has to pay lesser amount, even if it lies in the category of Price-H. But in the case presented here, we know more chronological details about CSC 1, therefore, the pricing is more convincing, based on CSC's characteristic. In case of CSC 2, CSP has earned the same profit value USD 21, as that of CSC 1, but since CSC 2 has a relatively low relinquish probability, 37%, therefore, the payable price is USD 105.47. In case of CSC 3, relinquish probability is 70%, but profit earned so far is USD 80, higher than the previous two cases. In this case, CSC 3 has to pay USD 107.8. If we compare the case of CSC 3 with CSC 1, they do not have a big difference in terms of relinquish probabilities and in fact, CSC 3 has a higher relinquish probability. Still, CSC 3 has to pay a lesser amount, because of the higher profit value, which CSP has earned from it. CSC 4 is the best case in the presented result; having comparatively lowest relinquish probability and highest earned profit for CSP, has to pay USD 103.6.

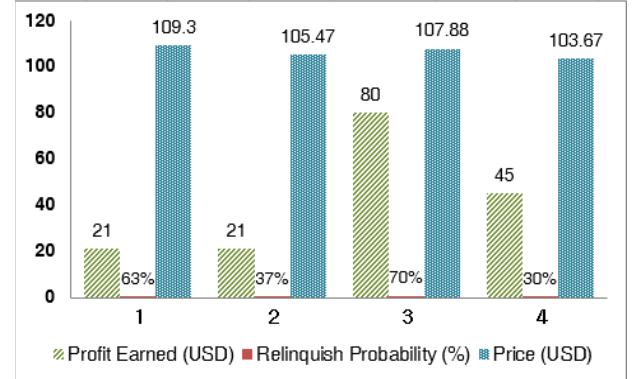


Figure 7. Price estimation according to customer characteristic, for USD 100 service.

VI. CONCLUSION AND FUTURE WORK

With rapidly increasing IoT services; service management, quality of service, efficiency, and user's satisfaction is becoming a crucial task. The future is of CoT, in which IoTs are amalgamated with cloud computing for better resource managements and service provisioning. In case of multimedia content, a lot of resources are required. Besides, emergency, healthcare, and other latency sensitive as well as security/privacy sensitive service require Fog, as a Micro Datacenter, to be present between the underlying nodes and the distant cloud. Efficient and in time scheduling and management of resources not only allows data centers to perform according to the situations, but also, helps customer satisfaction. In this paper, we have presented a model for management of resources, through Fog computing. The model takes into account resource

prediction, resource allocation, and pricing all in a realistic and dynamic way, also considering customer's type, traits, and characteristics. The model was then implemented using Java/NetBeans 8.0 and evaluated using CloudSim 3.0.3 toolkit. We believe that this work can be a good benchmark towards more realistic research and development, related with IoTs and Fog computing. Because of being dynamic and flexible, this model is capable to adapt according to the requirements of different CSPs. Therefore, it is implementable in different environments with varied scenarios. We intend to extend our work in respect of more heterogeneous services and QoS, keeping in view Fog computing concept. We would also incorporate device mobility factor in our extended work.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(No.NRF-2013R1A1A2013620). The corresponding author is Prof. Eui-Nam Huh. The corresponding author is Prof. Eui-Nam Huh.

REFERENCES

- [1] Gerd Kortuem, Fahim Kawsar, Daniel Fitton, and Vasughi Sundramoorthi, "Smart Objects and Building Blocks of Internet of Things", *IEEE Internet Computing Journal*, volume 14, issue 1, pp. 44-51, Jan.-Feb., 2010.
- [2] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions", *Technical Report CLOUDS-TR-2012-2*, July 2012.
- [3] Mohammad Aazam, Pham Phuoc Hung, Eui-Nam Huh, "Cloud of Things: Integrating Internet of Things with Cloud Computing and the Issues Involved", in the proceedings of 11th IEEE IBCAST, Islamabad, Pakistan, 14-18 January, 2014.
- [4] Mohammad Aazam, Pham Phuoc Hung, Eui-Nam Huh, "Smart Gateway Based Communication for Cloud of Things", In the proceedings of 9th IEEE ISSNIP, Singapore, 21-24 April, 2014.
- [5] Mohammad Aazam, Eui-Nam Huh, "Fog Computing and Smart Gateway Based Communication for Cloud of Things", in the proceedings of IEEE Future Internet of Things and Cloud (FiCloud), Barcelona, Spain, 27-29 August, 2014.
- [6] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, Sateesh Addepalli, "Fog Computing and Its Role in the Internet of Things", in the proceedings of ACM SIGCOMM, August 17, 2012, Helsinki, Finland.
- [7] Rogers, Owen, and Dave Cliff. "A financial brokerage model for cloud computing." *Journal of Cloud Computing* 1.1, 1-12, 2012.
- [8] Park, Ki-Woong, et al. "THEMIS: A Mutually verifiable billing system for the cloud computing environment." *Services Computing, IEEE Transactions on* 6.3, 300-313, 2013.
- [9] Wang, Wei, et al. "Dynamic cloud resource reservation via cloud brokerage", 33rd IEEE ICDCS 2013.
- [10] Yang, Yichao, et al. "A service-oriented broker for bulk data transfer in cloud computing.", 9th IEEE International Conference on Grid and Cooperative Computing (GCC), Nanjing, Jiangsu, China, 01-05 November, 2010.
- [11] Deelman, Ewa, et al. "The cost of doing science on the cloud: the montage example." *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, 2008.
- [12] Shadi Ibrahim, Bingsheng He, Hai Jin, "Towards Pay-As-You-Consume Cloud Computing", *IEEE International Conference on Services Computing*, Washington, USA, July 4-9, 2011.
- [13] Kan Yang, Xiaohua Jia, "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing", *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, September 2013.
- [14] Zhen Xiao, Weijia Song, and Qi Chen, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment", *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, June 2013.
- [15] D. Cenk Erdil, "Autonomic cloud resource sharing for intercloud federations", *Future Generation Computer Systems* vol. 29 (2013) 1700–1708.
- [16] Rakpong Kaewpuang, Dusit Niyato, Ping Wang, and Ekram Hossain, "A Framework for Cooperative Resource Management in Mobile Cloud Computing", *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, Vol. 31, No. 12, December 2013.
- [17] Mohammad Aazam, Eui-Nam Huh, "Inter-Cloud Architecture and Media Cloud Storage Design Considerations" *proceedings of 7th IEEE CLOUD*, Anchorage, Alaska, USA, 27 June – 02 July, 2014.
- [18] Salvatore J. Stolfo, Malek Ben Salem, Angelos D. Keromytis, "Fog Computing: Mitigating Insider Data Theft Attacks in the Cloud", *Security and Privacy Workshops (SPW)*, 2012 IEEE Symposium on. IEEE, 2012.