Offload as a Service in Fog computing Environment
A novel approach
Dai Hoang Tran, Tra Huong Thi Le, Oanh Tran Thi Kim,
Nguyen H. Tran, Eui-Nam Huh, Choong Seon Hong
Emails: {dai.tran, huong_tra25, ttkoanh, nguyenth, johnhuh, cshong}@khu.ac.kr

Abstract
Recently the concept and vision of Fog Computing to enable applications on billions of devices that already connected and form the Internet of Things (IoT) was introduced to run directly at the edge of network. With the vision that fog computing paradigm will be the future of the computing technology, we look at its characteristics and propose a novel approach to enable a new kind of service, that is Offload As A Service or OaaS. Offload computation has been an active research area for many years. It is exciting technology, in which it provides capability to extend mobile resources over their limitation in term of CPU, storage and battery energy. Fog computing paradigm has clear synergy with offload computation technology because of its low delay and close proximity features. To realize the enabling of OaaS in fog computing environment, we propose a novel approach to solve many challenging problems. Our framework comprises a testbed that represents a small scale fog network, a set of applications for communication and offloading mechanism between different layers of fog architecture, and a matching algorithm to handle the fair mapping between user and service providers. The result of our work shows great potential and value.

1. Introduction
Cloud computing is a mature technology that provides a huge leap in elastic computation [1]. And new development trends are shaped to compliment the cloud computing paradigm. Cisco recently introduced the concept and vision of Fog Computing to enable applications on billions of devices that already connected and form the Internet of Things (IoT) [2], to run directly at the edge of network [3–5]. There are similarities between cloud and fog in which they both provide on-demand data, compute, storage and application services to end-users. What differentiates fog characteristics are its close proximity to the end-users, its strong support for mobility and its dense geographic distribution. Thus, now services can be provided at places nearby the end-users such as access gateway, or set-top boxes. To the end-users, Fog computing provides more advantages in comparison to the cloud such as location awareness, low latency, better Quality of Service (QoS) for streaming, real-time applications and strong heterogeneity support.

Computing technology advancements overtime has dramatically changed the shapes and physical characteristics of the computing devices. This has led to a rapid gap in demand for complex software programs and the availability of hardware limited resources in mobile devices. One of the most effective method to augment these mobile devices’ capabilities is to use offloading computation. Offloading by definition is to unload, transfer data to other devices. So in a general view, offloading computation (OC) is to migrate the computation to stronger, resourceful computing devices such as a cloud server. At first, it may seem this is the same as the traditional client–server architecture, but it is not. In the client/server model, the client always asks for results of a computational task from its server, whereas in OC model, the client only asks for offload when it needs. In other literature, researchers also use different terms to describe OC likes “surrogate computing” and “cyber foraging”.

There is a strong connection between the new fog computing paradigm and the well–researched offload computation methods. Within the fog–computing environment, we have thousands if not millions of surround mobile devices that are each has some capabilities in processing computational tasks. Either they are some extra storage, or idle CPU clocks, etc. that most often are not fully utilized. This has given a new opportunity for a new kind offload computing aspect, where the surrounded fog devices, given the abundant of resources in their infrastructure, will provide OC services to the one who needs. And we give it the name “Offload as a Service”, similar to other acronyms in cloud computing such as SaaS, PaaS and IaaS. But as intuitive as it sounds, there are many challenging that we need tackle for this new offloading model. Thus, in this paper, we propose a framework to provide the managing of offload services across multiple fog service providers and its users.

The remaining of the paper is organized as follows: section 2 and 3 show the scenario and implementation of our framework. Section 4 and 5 shows the evaluation results and our conclusions.
2. Scenario
The scenario for our research model is illustrated in Figure 1. The surrounded fog environment is comprised of many IoT connected devices such as pc terminal, gateway, server, smartphone/sensor, and they are provided by different service providers (SPs). From the Figure 1, we have A, B, C as an example of SPs. In the middle are our users’ devices (UD). In this case, we have UD1 and UD2. While these users are in this fog environment, they want to use the Offloading service (OaaS) that is provided by these SPs on their devices. To be able to utilize this OaaS, users activate the application of our framework, and it will register/receive user’s preferences, get current status of its devices (such as battery level, network bandwidth, CPU speed, etc.). And based on these information, the framework will find the best fog devices around to offload the computation and orchestrate the communication until the service is served.

3. OcOc framework
3.1 Overview
The framework has three components that are distributed deployed on different layers in the fog environment infrastructure. Figure 2 illustrates our proposed framework. On the top layer, which is the Data Center or the Cloud of the SPs, each will have their own database of customers (the users of the fog services), which can be tied into their mobile plan (Subscriber model). We call these databases the Oc-DB. Below the Cloud layer, come the Edge layer, where a dense network of Fog devices are Geo-graphically distributed, and we put our framework agent in these devices, and we call it Oc-Agent. Finally, at the lowest layer, which is the Smart Things Network, where million of smart devices/sensors are distributed, we put our framework client on the capable running devices (i.e. mobile phone and tablet), and we call it the Oc-Client. As you can see, components of each layer are all connected with each others. Thus, our framework mechanism to handle all the interaction and connectivity between nodes in different layers is very important.

3.2 Prototype implementation
With three main components and each stays in different layer, they will form the communication path during the processing of a user’s service request. The architecture of the framework is detailed in Figure 3. First, user inputs his initial preferences into the Oc-Client in the first time usage. These preferences will be the constraints for our matching algorithm to determine the best fog device in the surrounded environment to serve him. Upon invocation of Oc-Client for an offload computation request, the Oc-Discovery and Oc-Learner will get activated. The Oc-Discovery sub module will broadcast offload request and then wait for various acknowledgment of offerings from the Oc-Receiver. These offering will be passed to the Oc-Core to make the final assessment either from user or automatically. Upon agreement of the chosen service, the user will get authenticated by the Oc-Core reside in the Fog device. It will check with Oc-DB in the SP’s cloud, since the Edge Network layer usually connects to a strong backbone network, the process will get completed very fast. Another important sub-module that is the Oc-Learner, it will record the final decision, the status of the devices, and how good the offload service is, what is the cost for that piece of computation, etc. All of these metrics will be evaluated and get added to the list of constraints in our matching algorithm.

4. Evaluation
We simulate the fog environment as in Figure 4. One strong
Desktop is used as the cloud central, one tablet and one android board are used to act as the fog devices, and the android mobiles are used to act as user’s devices. The communication involves wired and Wi-Fi and Wi-Fi Direct between different fog layers.

![Simulated Fog computing environment](image)

**Figure 4: Simulated Fog computing environment**

We perform the evaluation by offloading the calculation of Fibonacci and bubble sort algorithm. These algorithms are implemented in an ineffective way to make calculation would take longer to finish, thus the benefit of offloading would be valued. As we can see from the figures, the offloading provides good results as expected. This shows us the feasibility of our OaaS novel approach.

![Bubble Sort offload computation result](image)

**Figure 5: Bubble Sort offload computation result**

### 5. Conclusions and Future Work

Fog computing paradigm and offload computation open many new exciting technologies to be explored in the computing world. As a successor of cloud computing, offload computation can now reply on fog instead of cloud environment to enhance its capabilities and provide better user’s experiences. We realize that Offload as a Service in fog computing environment can be achieved. Our work has primary focused on how feasible is the approach and how practical it is. For that purpose, we propose a novel framework with methodology prototype implementation and algorithms to seek a proof of concept for OaaS in fog computing environment. Following the model introduced by Cisco, we choose Android platforms as a mean to simulate of novel framework. By trying to simulate different fog environment layers in similar fashion, we form a working infrastructure testbed that can behave the way we would expect. And by provide matching algorithms to perform real-time mapping of user devices to fog devices, we can confirm the accuracy of proposed novel communication model. Finally, various testing results give us confident in believing that OaaS in fog computing environment is possible, and can provide many great new application for users in future technology, where demand in smoothly interaction and low energy consumption play a central role.

In the future, we would like to improve our novel framework in many aspects. We plan to incorporate more communication protocols into the testbed instead of just Wi-Fi Direct at this moment, to increase range of testing devices, which can be real IoT devices such as sensors or Pan–tilt–zoom cameras, etc. Other aspect to be improved is the OcOc framework components. We would like to add more offloading functions into the testbed such as video streaming, gaming offloading and more interactive tasks to see how low delay characteristic of fog computing plays out.

### 6. Acknowledgement

This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2015-H8501-15-1015) supervised by the IITP(Institute for Information & communications Technology Promotion). *Dr. CS Hong is the corresponding author*

### References


