Rate Adaptive Fog Service Platform for Heterogeneous IoT Applications

Tiehua Zhang, Jiong Jin, *Member, IEEE,* Xi Zheng, *Member, IEEE,* and Yun Yang, *Senior Member, IEEE*

***Abstract*—With the advancement of the Internet of Things (IoT) technologies, the number of heterogeneous IoT applications requiring a variety of resources and services is increasing dramatically. Recently, the introduction of fog computing has further unlocked the potential of real-time services within the IoT context. On the basis of fog architecture, we herein propose a novel rate adaptive fog service platform [1] aiming at heterogeneous services provisioning and optimized service rate allocation. By forming several service groups in the fog network in which each service could be adequately provisioned, service consumers would always benefit from the fact that the majority of services produced by IoT applications are in their proximity and thus are delivered to the destination promptly. Taking advantage of the well-known network utility maximization (NUM) approach, a service rate adaptive algorithm is developed to empower fog nodes working together to adjust service delivery rate dynamically. Throughout this process, the algorithm takes the current network condition and constraint into account to ensure the rate is calibrated in favor of providing satisfactory Quality of Service (QoS) to each service receiver at the same time. Compared to other resource allocation strategies that mainly focus on allocating resources for a single network service, our proposed platform is capable of not only dealing with both elastic and inelastic services but also handling the abrupt network changes and converging back to the global optimum rapidly.**

***Index Terms*—Fog Computing; Internet of Things (IoT); Service-Oriented Networking; Network Utility Maximization (NUM); Quality of Service (QoS).**

I. Introduction

INTERNET of Things (IoT) is a growing topic of interest and has already attracted widespread attention from both academia and industries. The consensus on the definition of IoT is a network composed of heterogeneous devices (or things) that are equipped with computation and communication capabilities, some of which are able to interact with the cloud to complete tasks collaboratively. Because of the communicating abilities with the cloud, these *things* become "much smarter" since the data collected from physical surroundings could be further analyzed [2].

The proliferation of IoT technology in service-oriented computing has unleashed the great potential in many areas, especially for the service provisioning companies in the software industry seeking to leverage the advancement of IoT and provide a wide range of real-time services so as to cater for the growing needs from users [3]. According to the Internet of Services (IoS) vision, these prevalent IoT applications rely on the process of collecting and analyzing users' data in order to offer highly personalized, context-aware services [4]. During this process, the interaction, communication and collaboration between *things* and the cloud are inevitable [5], [6].

Undoubtedly, the use of well-known cloud computing paradigm demonstrates the benefits in many aspects, e.g., the provision of enterprise-level computing, storage and networking capabilities in a "Pay-As-You-Go" fashion to reduce the cost of individuals and organizations [7]. Apart from that, these coarse-grained, discoverable application entities could be centralized at the cloud to take advantage of convenient, low-cost manageability and strong reliability [8].

Emerging IoT applications, nevertheless, have more stringent latency requirements and mostly expect a timely response. Therefore, waiting for services to be transferred from the cloud is no longer efficient and effective due to issues like communication overhead and service delivery latency. Besides, the privacy and security of user data is another big challenge [9]. IoT applications in cloud platform usually trade-off data privacy for service quality by storing and retrieving sensitive data in the cloud. Even though some mechanisms have been developed for confidentiality purpose, it could still cause problems like colossal bandwidth waste and energy consumption [10]. These issues essentially suggest that the reliance on traditional IoT-Cloud schema is no longer an efficient approach, and it is imperative to come up with an alternative computing paradigm that could seal the gap.

To address these issues mentioned above and cope with the inadequacy of the cloud, fog computing has been introduced. Initially proposed by Cisco, fog computing is introduced to empower the computing directly at the edge of the network to host different IoT applications and provide services in this regard [5]. In this ecosystem, facilities located at the edge of the network and capable of providing resources for services are called fog nodes, which are considered as an extension of the cloud at the edge with the overarching goal of "off-loading" from the cloud. Fog nodes, like the proxy of the cloud, could be equipped with not only computation power, but also storage and networking resources required by a variety of IoT applications so that the deployment no longer needs to happen on either resource-constrained IoT devices or remote cloud. In this sense, fog and cloud complement each other to form a service continuum from which end users could seamlessly receive the particular service [6].

Along with the rapid growth of IoT applications, heterogeneous services are tailored to meet the needs of service consumers with certain QoS guarantee. In reality, stable

Tiehua Zhang, Jiong Jin and Yun Yang are with the School of Software and Electrical Engineering, Swinburne University of Technology, Melbourne, Australia (e-mail: tiehuazhang, jiongjin, yyang@swin.edu.au).

Xi Zheng is with the Department of Computing, Macquarie University, Sydney, Australia (e-mail: james.zheng@mq.edu.au).

[1]Preliminary version of this paper appeared in a conferences [1].

service delivery rate is a crucial component to achieve the desirable QoS, and it could act as a significant part in service consumers perception with regard to the overall performance of the service invocation [11], [12]. In our work, the utility function is used to measure user's satisfaction and to model the QoS performance, which increases as the increase of service delivery rate. Thus, the use of network utility maximization (NUM) sheds light on maximizing the allocation of service delivery rate based on residual bandwidth under current IoT network and therefore benefits QoS simultaneously. From the utility point of view, services provisioned by different IoT applications can be categorized into two main groups, i.e., traditional elastic services and real-time inelastic services [13]. The former usually refers to the ones like file transfer, data analysis and web browsing services, etc., where each service attains a strictly increasing and concave utility function to reflect its QoS performance. In comparison, real-time inelastic services are generally provided by real-time applications such as audio, video and multimedia delivery applications. Such services have an intrinsic bandwidth threshold in nature and adopt the sigmoid-like functions to describe the particular QoS [14].

In our work, we seek to leverage the advancement of fog computing by accommodating fog nodes as the service holders for heterogenous IoT applications. The proposed fog service provision platform enables fog nodes to collaborate vertically to adjust service transmission rates in real time under the resources-constrained IoT network.

The main contributions of this paper are as follows:

1) To better serve for services through IoT network, fog architecture has been applied under the service-oriented computing context. Based on the fog architecture, a fog service platform is developed to support both elastic and inelastic IoT services. Also, fog nodes, which can be easily deployed in the proximity of end users/devices, complement the cloud as the role of the service provider in the delay-sensitive service spectrum.
2) An analytical framework, including a mathematical-proven theorem, is generalized and ready to use. This framework is guaranteed to support both elastic and inelastic services and capable of allocating the underlying IoT resources to each service type both fairly and optimally. In other words, the unstable oscillation problem happened when simultaneously allocating resource to different service types no longer exists.
3) With the help of the analytical framework, a service rate adaptive algorithm is devised from an engineering point of view, and distributedly runs on each fog node in the platform. The algorithm enables the fog nodes to 1) recursively collaborate with its parent node to calibrate the service transmission rate to each requester based on current network conditions and constraint; 2) handle the abrupt changes of the IoT network and stabilize the affected service rate rapidly; and 3) ensure the fairness and global optimum with regards to the rate adaptation.
4) The platform is adopted in a shopping use case and modelled using a fog deployment simulator to mimic the real-world deployment closely. Both service delivery latency and energy consumption are then studied to verify its effectiveness.

The rest of paper is organized as follows. We review the related work on both service delivery architecture and NUM-based service rate allocation in Section II. In Section III, we introduce the architecture of the proposed platform in detail and formulate the optimisation problem in Section IV. We then develop the service rate adaptive algorithm in Section V, followed by a shopping mall case study along with the experiments to illustrate the practicality and effectiveness of the platform in Section VI. We draw the conclusion and point out the future work in Section VII.

II. Related Work

There are several previous efforts made towards developing service delivery architecture to connect service consumers and providers in IoT environment. In [3], the service-oriented architecture (SOA) is embedded onto IoT devices to provide on-demand web services to facilitate the service querying and discovery process. However, some critical issues, e.g., limited computing capabilities of IoT devices and complex IoT network conditions, are not discussed in this paper. Apart from that, the authors of [15] propose a vehicular data cloud platform to provide real-time information such as traffic control and management, car location tracking and monitoring, and road condition to different receivers in IoT environment, but service transmission latency and underlying transportation cost are not considered in their model. In addition, authors in [16] offer several schemes to reduce power consumption by hard real-time services and power-aware profitable provisioning of soft real-time services. The traditional cloud data centre is selected as the service provider in which a real-time virtual machine model is devised to handle the real-time service requests, and power-aware provisioning of virtual machines for real-time services is mainly studied. Similarly, the work in [17] develop a real-time cloud services framework to Vehicular Clients (VCs) aiming to cope with delay and delay-jitter issues.

A group of efforts has been spent on the research of leveraging the fog computing paradigm in different aspects. In [18], authors propose an adaptive fog configuration strategies to dynamically configure fog nodes to host services for sensors deployed in an industrial environment. The work in [19] focuses on solving the load balancing issue so as to achieve resource efficiency and avoid bottlenecks. Likewise, the bandwidth resource allocation problem is studied in [20] concerning the scale of IoT devices that are connected into the fog network, which is then solved using the analytic hierarchy process (AHP). However, there is no guarantee for the global optimum allocation, and the service types from different IoT devices are not taken into account.

By comprehending the service delivery delay caused by relying solely on the cloud data centre, authors in [21] present a service provision framework incorporating both cloud and mobile edge computing. In this work, the cloud plane is used to process large-scale, long-term, global data, which can be used to obtain decision making information such

as feature, law, or rule sets. In contrast, the edge plane is used to process small-scale, short-term, local data, which is used to present a real-time situation. The adoption of this framework essentially gives the alternative that offloads the computational workload from the cloud by assigning certain processing tasks to the mobile edge devices. However, it is very likely that this framework could malfunction as most of the real-time applications are resource/energy consuming. In other words, various resources other than computational power should be made available and put in the edge so that heterogeneous services could be provisioned both robustly and promptly. Similarly, the work in [22] focuses on allocating resources for microservices in the edge cloud environment, and an online auction-based mechanism is proposed so that the edge cloud platform can reclaim the allocated resources and reallocate them to other microservices waiting in the line. Also, authors in [23] study on minimizing the end-to-end service latency and service completion time through a latency-oblivious distributed task scheduling scheme to improve the QoS.

To better utilize underlying IT resources and achieve good QoS, an autonomic service platform is proposed in [11]. The core component of this platform is a service routing protocol that makes use of NUM, allowing service intermediaries to route the service request from consumers to providers dynamically. Unfortunately, this platform is not devised for IoT networking environment and does not take network changes into consideration either. To enhance the quality of experience across all users, authors in [24] devote to designing the utility-based framework within a total network transit cost budget, and with the same philosophy of our work, maximizing the utility would lead to a better user QoS experience.

The work in [25] focuses on achieving good QoS by adjusting service transmission rate that maximizes the total receiver utilities in a multicast multirate network setting. However, one serious limitation of their approach is that it can only handle elastic network services, meaning the utility function selected must be strictly concave and thus are not suitable for real-time IoT services. Authors in [26] raise concerns on QoS as well, thus a joint optimisation problem regarding minimization of the service latency, optimal revenue maximization while keeping an acceptable QoS is solved through the proposed adaptive service offloading scheme.

III. The Architecture of Fog Service Platform

In this section, we introduce the architecture of this fog platform as well as the components inside. The platform is composed of end devices/users (*things*), fog nodes and the cloud. From the service-oriented computing perspective [8], *things* generally constitute service requesters/receivers. Fog nodes equipped with computation, storage and networking power could serve as either service providers or service intermediaries/forwarders, in which service intermediaries help collect service requests from bottom-level *things*, track network conditions, cooperate with providers to adapt service transmission rate, and forward services back to requesters. Since the cloud treats fog as the proxy at the edge of the network, it is noticeable that the use of cloud is no longer mandatory in this platform, but one could choose to continue using the cloud as a service provider for some energy consuming, delay tolerant services, e.g., large scale data backup service.

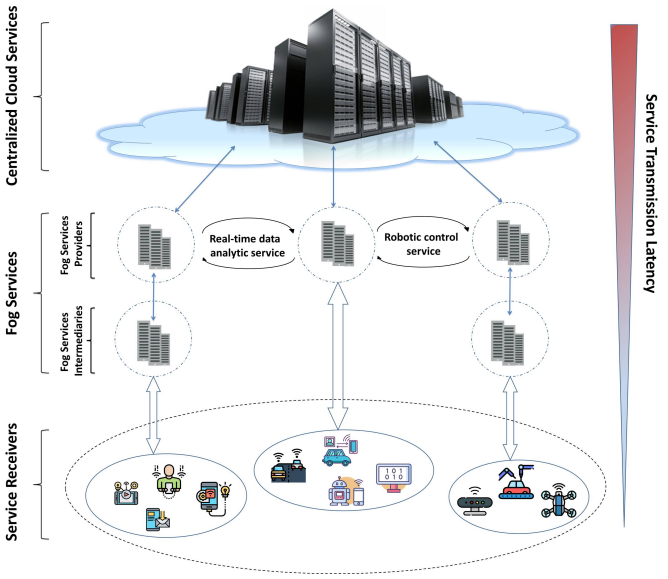

Fig. 1: The architecture of fog service platform, and different services provided by cloud and fog nodes

In order to mitigate service delivery latency and obtain a good service quality, IoT applications could deploy on fog nodes in the vicinity of *things* [27]. When it comes to the user data privacy, fog node that has more storage capacity could serve as the user data repository, essentially giving an alternative to the network manager who concerns about data privacy issues in the cloud [10].

An example of the fog service provisioning platform is presented in Fig. 1. It shows that a variety of end devices/users plays the role of service requesters at the bottom layer. Fog nodes, as the placeholder for different IoT applications in middle layers, put efforts together to establish several fog service groups to facilitate the fog service generation and distribution, where each group distributes a particular type of service. Similar to the traditional IoT network, the cloud stays at the top layer. Regrading the service groups, it is worth noting that each service provider could reside in multiple groups, i.e., a fog node could essentially provision multiple services if condition allowed. For instance, one fog node provides real-time data analytics service and robotic control service at the same time in Fig. 1. Fog service provider in each group takes up the responsibility to gather feedback regarding downstream network conditions reported by service intermediaries, or even collaborate with each other in the process of generating services. Needless to say, fog node as a service provider could empower real-time IoT applications to give the timely response to receivers (following the decreasing service transmission latency trend pointed out in the figure). When it comes to the service intermediaries, the platform allows these fog nodes to be converted to the providers if equipped with enough resources, which increases the scalability and

flexibility of the platform substantially.

To build this platform, the following assumptions are reasonably made: 1) the bottom level *things* are connected into the network through nearby fog nodes; 2) fog nodes are interconnected vertically and aware of both downstream and upstream links.

IV. The Analytical Framework and optimisation Problem

The service transmission rate allocation problem in the fog-based IoT environment is formulated in this section to make it support both elastic and real-time inelastic services along with the generalization for different fog architecture designs. Additionally, we characterize the utility in terms of allocated service transmission rate deriving from the underlying bandwidth of fog network.

Consider a fog service delivery network consisting of a set of links $L = \{1, 2, ..., l\}$, each of which has capacity c_l. There is a set of $S = \{1, 2, ..., s\}$ service groups, and each service group is devoted to providing one particular service. For each service group s, there is only one unique service provider, which is either a fog node or the cloud. A set of receivers that is in service group s could be noted as $R_s = \{r_{s,1}, r_{s,2}, ..., r_{s,n}\}$, and along with a set of links $L_s \subset L$, they together form the corresponding service delivery tree of that service group, where the provider stays at the root of the tree, and each receiver in R_s is connected to the IoT network through the leaf fog node.

For each service receiver $R_{s,i} \in R_s$ in a service group, $L_{s,i} \subset L_s$ describes the service delivery path from the provider of service group s to relevant receiver i. Say $x_{s,i}$ represents the service delivery rate to receiver i in service group s. Then the set of service rates for respective receivers is defined as:

$$x = [x_{1,1}, ..., x_{1,n_1}, ..., x_{2,n_2}, ..., x_{s,1}, ..., x_{s,n_s}]$$

As stated earlier, utility function $U_s(x_{s,i})$ has been modelled on per-service basis to describe its QoS requirement. The original utility function $U_s(x_{s,i})$ is non-negative, continuous and strictly increasing over the range $x_{s,i} \in [m_s, M_s]$, where m_s and M_s represent the minimum and maximum service delivery rate, respectively. As it fails to guarantee the concavity in the inelastic scenario, a "pseudo utility", denoted as $\mathcal{U}_s(x_{s,i})$, is defined to generalize both elastic and inelastic services [14], where $U_s(x_{s,i})$ needs not be concave in the "pseudo utility" context.

Considering the characteristics of utility functions for both elastic and inelastic services, $\mathcal{U}_s(x_{s,i})$ should be modified to be increasing and strictly concave under any service types. The rationale is that we expect to form a convex optimisation problem so that a global optimal value could be obtained. Therefore, it is crafted to relate to original utility function as [28]:

$$\mathcal{U}_s(x_{s,i}) = \int_{m_s}^{x_{s,i}} \frac{1}{U_s(y)} dy, \quad m_s \leq x_{s,i} \leq M_s \tag{1}$$

Since we focus on making our platform support both elastic and inelastic services, the "pseudo utility" function is used, and it leads to the optimisation problem $P1$:

$$P1: \quad \max_{x \geq 0} \mathcal{U}_s(x_{s,i}) = \sum_{s \in S} \sum_{i=1}^{n_s} \mathcal{U}_s(x_{s,i}) \tag{2}$$

$$\text{subject to} \quad \sum_{s \in S} x_s^l \leq c_l, \quad \forall l \in L \tag{3}$$

$$x_s^l = \max_{\{i | l \in L_{s,i}\}} x_{s,i} \tag{4}$$

In equation (4), $\{i | l \in L_{s,i}\}$ is a set of receivers that uses link l to receive the corresponding service in service group s. This equation states that in service group s, the service rate on link l is the same as the rate of the fastest downstream receiver in this group. In addition, constraint (3) in this optimisation problem suggests that the aggregate service rate on link l across all service groups should not exceed the link capacity (network condition). Since equation (4) contains the maximum discrete function that is not continuous and differentiable, it is difficult to solve the problem by traditional optimisation methods. We thus make an approximate solution as follows:

$$x_s^l = \max_{\{i | l \in L_{s,i}\}} x_{s,i} = \lim_{N \to \infty} \left(\sum_{\{i | l \in L_{s,i}\}} x_{s,i}^N \right)^{\frac{1}{N}} \tag{5}$$

Therefore, the maximum function in equation (4) could be approximated by:

$$x_s^l = \left(\sum_{\{i | l \in L_{s,i}\}} x_{s,i}^N \right)^{\frac{1}{N}} \tag{6}$$

where N is a sufficiently large integer. After the transformation, the original problem $P1$ could be re-formulated by the following optimistion problem:

$$P2: \quad \max_{x \geq 0} \mathcal{U}_s(x_{s,i}) = \sum_{s \in S} \sum_{i=1}^{n_s} \mathcal{U}_s(x_{s,i}) \tag{7}$$

$$\text{subject to} \quad \sum_{s \in S} \left(\sum_{\{i | l \in L_{s,i}\}} x_{s,i}^N \right)^{\frac{1}{N}} \leq c_l, \quad \forall l \in L \tag{8}$$

Clearly when N goes to ∞, $P2$ is equivalent to the original problem $P1$.

In order to solve $P2$, the Lagrangian problem is then derived as:

$$L(x, p) = \sum_{s \in S} \sum_{i=1}^{n_s} \mathcal{U}_s(x_{s,i}) - \sum_{l \in L} p^l \left[\sum_{s \in S} \left(\sum_{\{i | l \in L_{s,i}\}} x_{s,i}^N \right)^{\frac{1}{N}} - c_l \right] \tag{9}$$

Theorem 1. *For service receiver requesting heterogeneous IoT services, the optimal service transmission rate is under condition that each Lagrangian multiplier $p^i = [p^1, p^2 ... p^l] \geq 0$, and each service receiver should equip with a price weighting coefficient $w_{s,i}^l$ in relation with link l, such that:*

$$x_{s,i} = U_s^{-1} \left(\left[\frac{1}{p_{s,i}} \right]_{U_s(m_s)}^{U_s(M_s)} \right) \tag{10}$$

$$p^l = \left[p^l + \lambda(\sum_{s \in S} x_s^l - c^l) \right]^+ \tag{11}$$

$$w_{s,i}^l = \frac{x_{s,i}^N}{\sum\limits_{\{j|l \in L_{s,j}\}} x_{s,j}^N} \tag{12}$$

$$p_{s,i} = \sum_{l \in L_{s,i}} w_{s,i}^l p^l \tag{13}$$

Proof. The proof and derivation of Theorem 1 can be found in the Appendix. □

This theorem reveals that, in the steady state, the associated utility U_s is simply equal to $\frac{1}{p_{s,i}}$, in which $p_{s,i} \in [\frac{1}{U_s(M_s)}, \frac{1}{U_s(m_s)}]$.

As stated above, N should always be a sufficiently large integer; however, it is worth noticing that in equation (12), if $x_{s,i}$ has a relatively large value, then the corresponding $w_{s,i}^l$ will encounter a sudden change that further results in an unstable condition for equation (13), and ultimately affect rate adapting process, hence the following modifications have been made to improve the robustness:

$$Initiate: \quad w_{s,i}^l = \frac{1}{|\{j|l \in L_{s,j}\}|} \tag{14}$$

$$w_{s,i}^l = [w_{s,i}^l + \lambda(x_{s,i} - x_s^l)]^+ \tag{15}$$

$$w_{s,i}^l = 1 - \sum_{\{j|j \neq i, l \in L_{s,i}\}} w_{s,j}^l \tag{16}$$

Equation (14) implies that w has a value within a range of $w_{s,i}^l \in [0, 1]$. Equations (15) and (16) indicate that $w_{s,i}^l$ will continue to increase to its boundary for the fastest receiver in service group s, while decreasing among other slow receivers in the same service group. Finally, the receiver with the largest service rate will have $w_{s,i}^l = 1$ on link l.

Since equations (11) and (15) have been updated with a step size λ, it is important to select the value of parameter λ, which has a critical impact on the convergence speed. Similar to other gradient projection algorithms, when λ is selected appropriately and not larger than some positive λ^*, the service delivery rate will converge smoothly to the optimal value [14].

From the flow control aspect, our analytical framework emphasizes the relationship between bandwidth allocation and QoS performance of applications. It is implicitly assumed that the service will be served timely and reliably if sufficient bandwidth is allocated and service providers are only a few hops away from the receiver. Given the link capacity constraint, the only way to ensure that no receiver has been left behind is to allocate the underlying resources both fairly and optimally. By doing so, one can at least ensure that the delay is decreased owing to the abundant bandwidth support for that particular service. In an extreme case where the bandwidth supply is much less than needed due to the communication overhead caused by colossal receivers, the admission control over the number of users being allowed to connect in can be the most effective solution for sufficient allocation and less latency.

However, especially for real-time applications, it will be more challenging to explicitly consider the packet delay effects and solve it as the convex optimisation problem regarding the increase of the number of users and change of network situation. One possible extension in this direction is to follow the work suggested by [29] in which a new utility function for receivers could be defined to incorporate the delay into the analytical framework as:

$$\mathcal{U}_s^{new}(x_{s,i}) = \mathcal{U}_s(x_{s,i}) - \beta_s \sum_{\{i|l \in L_{s,i}\}} d_s\left(x_{s,i}^l\right) \tag{17}$$

where $d_s\left(x_{s,i}^l\right)$ represents the average delay happened by a packet of service s to receiver i on link l. Therefore, the summation $\sum_{\{i|l \in L_{s,i}\}} d_s\left(x_{s,i}^l\right)$ calculates the end-to-end delay of a particular service for that receiver. The tuning parameter $\beta_s > 0$ reflects the relative importance of the service versus delay.

To summarize, the derived analytical framework takes advantages of both link price p^l and price weighting coefficient $w_{s,i}^l$ to adjust the service transmission rate for receivers on that specific link. Whenever a link exceeds its capacity constraint, these two parameters (equations 11 and 15) get adapted accordingly, which ultimately lead to the adaptation of service rate, as indicated in equation (10). The analytical framework is integrated into the fog platform through the implementation of the algorithm detailed in the next section.

V. Service Rate Adaptive Algorithm and Implementation

We now present the service rate adaptive algorithm adopted by the platform in this section, and more importantly, demonstrate how to deploy it in the fog architecture.

A. Motivation

As discussed previously, the overarching goal of the proposed platform is to allow fog nodes to work together and reach a consensus. To achieve that, fog nodes playing in different roles should comprehend the responsibility it should carry out and respond properly. For instance, service providers should calculate the corresponding service delivery rate based on the received feedback concerning the downstream link conditions, whereas service forwarder merely calculates its link condition and report it upwards. Herein, the algorithm is developed from an engineering perspective to instruct all fog nodes to work towards that goal, where every service provider in the platform could distribute respective service at the optimal rate eventually based on the feedback of network condition recursively passed by the downstream fog nodes (the bottom-up approach). It substantially benefits from the results derived in the analytical framework in Section IV to guarantee the global optimum of resource allocation.

B. Overview of the Algorithm

Algorithm 1 displays a summary of the algorithm, which consists of two phases. The bottom-level fog nodes would firstly gather relevant information such as the types of service requested, downstream links information, then forward these to either the topmost fog node or the cloud to form the service

tree (lines 2 - 4). Afterwards, several service groups have been established, and whenever a service requester joins or leaves a service group (network changes), the bottom-level fog node is able to sense the change and report it upwards, which consequently starts another round of Phase 1.

In Phase 2, fog nodes would firstly iterate through each downstream link and calculate the current link status. More specifically, lines 5 - 10 deal with the link price updating process, followed by the calculation of price weighting coefficient in lines 11 - 16, which implies that this coefficient would continue to increase for the receiver with the largest service rate while decreasing among other receivers. Lines 17 - 27 handle the service rate adapting process, if and only if the current node f is a service provider. Lines 29 - 31 indicate the bottom-up approach of this algorithm, in which all service providers would be reached and informed at the end with the latest network condition.

C. Computational Complexity Analysis

This part discusses the computational complexity of the adaptive service delivery rate algorithm. Since the most time-consuming operations reside in Phase 2, we thus mainly focus on the analysis of this part. We define some notations here for convenience. Say L represents the whole set of links and F representing all fog nodes in the platform, and the downstream links that a fog node f possesses are defined as $\{l_f \mid f \in F,\ l_f \in L\}$. As noted in the service rate adaptation process, each link under the control of a fog node will iterate through every service being delivered on it, and services on that link could be approximated by $\{s_l \mid l \in l_f\}$. For each service s on link l, there is a constant number of operations (n) on calculating the link price p along with its coefficient w, as well as adjusting the respective service transmission rate x. Therefore, the total number of operations could be calculated as follows:

$$\text{total number of operations} = \sum_{f \in F} \sum_{l \in l_f} n * s_l$$

Since the value of n is a constant and thus can be ignored, the computational complexity of this algorithm is $\mathcal{O}(F{*}l_f{*}s_l)$.

VI. Performance Evaluation on a Case Study

In this section, we evaluate the performance through a numerical experiment for the proposed fog service platform. The experiment not only validates the feasibility of the algorithm designated for the platform but also demonstrates its flexibility to adapt the service delivery rates for both elastic and inelastic IoT services. Most importantly, the scalability of the fog platform is reflected in the simulation as well in which service requester may intermittently join or leave the network.

A. Shopping Mall Use Case

In order to compete with online shopping and e-commerce, shopping malls nowadays employ a wide range of approaches to stimulate customer's shopping desires. One method emerged recently is to harness IoT technologies to excel in providing

Algorithm 1 Service Rate Adaptive Algorithm

Phase 1: *Initialization*

1: Fog or cloud will collect network condition data reported from child nodes, update relevant W and P, then communicate back.
2: Service group $S = [s^1, s^2 .... s^n]$
3: Link capacity $C = [c^1, c^2 .... c^l]$
4: $W, P \leftarrow R^S \times L$ matrix,where $w^l_{s,i} = \frac{1}{\{j | l \in L_{s,j}\}}, p^l_{s,i} = 0$ at initial stage

Phase 2: *Service Rate Adaptation*

1: Bottom-level fog nodes trigger the algorithm at every interval t, each node is aware of the services that traverse through it, as well as the current service transmission rate
2: **repeat**
3: $f \leftarrow$ current node
4: **for** each downstream link l of f **do**
5: 1. select largest service delivery rate of each service on that link
6: $$x^l_s = \max_{\{i | l \in L_{s,i}\}} x_{s,i}$$
7: 2. aggregate delivery rate of each service on link l
8: $$x^l = \sum_{s \in S} x^l_s$$
9: 3. calculate the current link price of l
10: $$p^l = [p^l + \lambda(x^l - c^l)]^+$$
11: 4. calculate the price weighting coefficients for each downstream receiver i,
12: whose service transverse link l
13: $$w^l_{s,i} = [w^l_{s,i} + \lambda(x_{s,i} - x^l_s)]^+$$
14: **if** receiver i receives service s at rate x^l_s **then**
15: $$w^l_{s,i} = 1 - \sum_{\{j | j \neq i, l \in L_{s,i}\}} w^l_{s,j}$$
16: **end if**
17: 5. calculate link price $p^l_{s,i}$ for each downstream receiver i on link l
18: $$p^l_{s,i} = w^l_{s,i} p^l$$
19: 6. update corresponding $w^l_{s,i}$ in W and $p^l_{s,i}$ in P, respectively
20: **if** f is a provider of service s **then**
21: **for** each receiver i that receives service s **do**
22: 7. calculate relevant path price
23: $$p_{s,i} = \sum_{l \in L_{s,i}} p^l_{s,i}$$
24: 8. adjust service rate
25: $$x_{s,i} = U_s^{-1}\left(\left[\frac{1}{p_{s,i}}\right]^{U_s(M_s)}_{U_s(m_s)}\right)$$
26: **end for**
27: **end if**
28: **end for**
29: **if** there is any upstream service coming to f **then**
30: 9. propagate network condition upward, and repeat phase 2
31: **end if**
32: **until** (all providers have been reached)

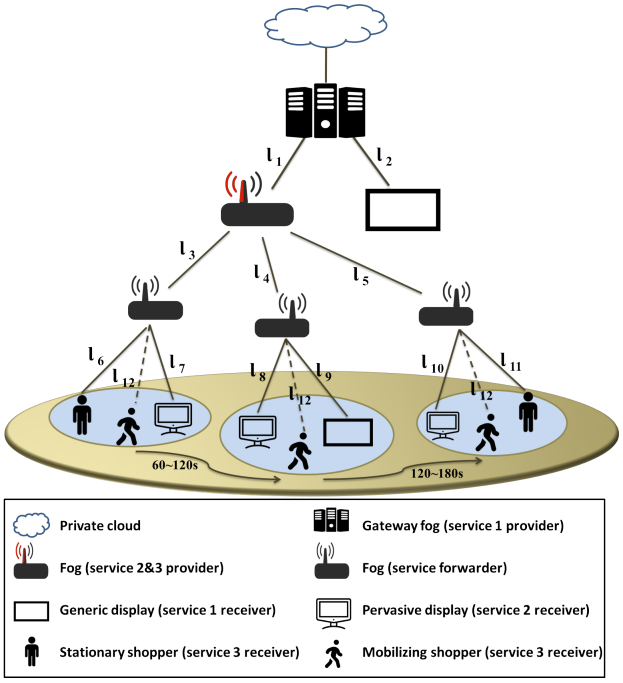

Fig. 2: Fog platform architecture in the shopping mall use case

highly related and attractive services to accommodate different shoppers. Apart from that, it catches some attention that increasing the usage of assistant objects such as digital signage deployed inside the mall to maximize the fine-grained branding and advertising opportunities could be beneficial to improve the overall shopping experience [30]. As stated previously, the challenge stands when it comes to providing the service both stably and promptly to shoppers. Herein, we apply the shopping mall use case as the "experimental field" for the proposed fog platform, where digital displays and shoppers get connected in the IoT network as service requesters that are under control of the platform. More specifically, these digital displays are in different shapes and sizes for varying purposes, referred as the conventional digital displays and the advanced pervasive displays, respectively.

The versatility of pervasive displays makes it an excellent candidate to support IoT applications in shopping mall scenario. These displays are capable of interacting with shoppers and pushing the relevant information of shopper's interest [30], [31]. Based on these attributes, these displays could be deployed at different shopping districts so that highly personalized store information such as personal preference or discount will be exhibited on the screen when a shopper approaches nearby. These displays mostly have relatively smaller screen size and could comfortably achieve better QoS with limited service delivery rate.

In contrast to pervasive displays, conventional displays have characteristics of big screen size, less interactive requirement and primarily with commercial-driven purpose. They could be placed at the noticeable spots such as the main foyer, central areas of each floor or food courts, and the contents pushed to this kind are related to the generic information of this shopping mall as well as video data with recreational and commercial purposes. The related QoS requirement for this type would be stringent as more video data along with the stable delivery rate is needed for better pixel quality. One similar feature shared between these two types of displays is that the service required is characterized as real-time, inelastic video streaming service. Therefore, the sigmoidal function should be chosen accordingly to describe the QoS.

However, it is unrealistic to consider inelastic services only in the shopping mall use case. Nowadays, people tend to spend more time on their phone to keep updated on the latest news while taking a break from shopping or tracking the discount information as interested. These service requests generally involve the elastic services like web browsing or mobile coupon searching from IoT applications. Thus, the logarithmic utility function is adopted to approximate these elastic IoT services.

Fig. 2 illustrates the topology of the IoT network empowered by the fog platform in the shopping mall. In this topology, all fog nodes are placed inside the shopping mall, in which an autonomous network has been formed so that administrators could easily monitor the network status. Various displays, as well as shoppers requesting for heterogeneous services, act as *things* in this network, and fog nodes are equipped with the different level of computational, storage and networking capabilities. In particular, the topmost fog node is the most powerful among all and operates as the main gateway of this autonomous network. Apart from that, the gateway fog node also controls the communications with the Internet outside and only uploads filtered data to the remote cloud for backup purpose.

Although the cloud is drawn in Fig. 2, it is worth mentioning that the cloud is not mandatory to become the service provider because of concerns such as high service transmission latency, communication overhead or data security issues. However, it could still provision time-tolerant service for backup purpose. One may notice that the fog deployment in the shopping mall use case appears to resemble the tree-topology, and it is notable that, since the service providers undertake more computing tasks than intermediaries and normally control more than one service intermediary for the ease of management in an autonomous network, the structure merely coincides with the appearance of tree-based topology. Apart from that, as the most prevalent and dominant topology in fog architecture, the tree-resembled topology in fog computing benefits many fields of research, including privacy preserving [32], autonomous vehicles [33] or many other use cases [34]. Our optimal service rate adaptation problem could thus contribute substantially to this active research area.

A number of IoT applications are deployed in the fog network to cater for different service requirements. We refer the service required by generic, commercial-driven displays as **Service 1** and pervasive screen as **Service 2**. These two services are considered to be real-time, inelastic video services, whereas services related to web browsing and coupon search requested by shoppers are categorised as elastic **Service 3**. It is worth pointing out that considering **Service 3** as elastic service here is merely to test the fairness and robustness

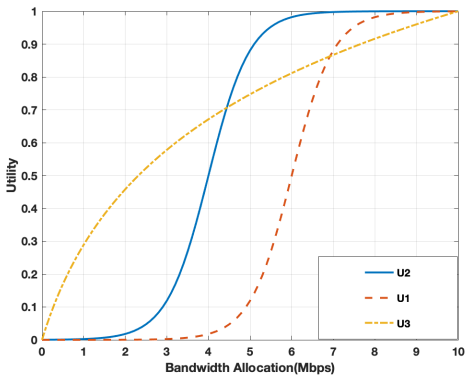

Fig. 3: Utility functions employed for service groups 1, 2 and 3

of allocating network resources for different service types (both elastic and inelastic), and **Service 3** could easily be realised as inelastic service if needed owing to the support of the analytical framework in Section III and the proof in the Appendix. The convergent nature and speed of the algorithm are not affected in this matter.

Considering the resource capacity for each fog node in this fog platform, the gateway fog node is the most powerful among all and thus capable of deploying the computationally expensive IoT applications. In other words, **Service 1** equips the purpose of expressing more general information of the mall, and the process of generating a large amount of video data is considered to be expensive, it is therefore more reasonable to be deployed on the gateway fog node. In contrast, **Service 2** relates to the interactive video streaming service, and end displays that request this service are located at different shopping districts to provide customers with the highly-personalized shopping information. Furthermore, these pervasive displays need not have as much video data transmitted as generic displays. Hence **Service 2** is derived from the connected child node of the gateway fog node. Apart from that, we tend to make the same fog node as the **Service 3** provider so that the service generated could arrive in requested shoppers with fewer hops. The design of this fog architecture meets the real-world scenario, and the selection of these service providers makes good use of the flexibility of the fog service platform.

Given the analysis of these three different types of IoT services, it is of importance to cast appropriate utility function to approximate the QoS. Explicitly, utility functions should be modelled on a per-service basis to better: 1) describe the corresponding QoS requirements, i.e., with the same service delivery rate, small displays tend to get "satisfied" much easier than large screens as its utility gets closer to 1; 2) reflect the nature of real-time inelastic service and other elastic services, respectively. We then select these utility functions as follows:

$$U_1(x_{s=1,i}) = \frac{1}{1+e^{-2(x-6)}} \tag{18}$$

$$U_2(x_{s=2,i}) = \frac{1}{1+e^{-2(x-4)}} \tag{19}$$

TABLE I: MATLAB simulation setup

Link capacity (Mbps)					
l_1	l_2	l_3	l_4	l_5	l_6 to l_{12}
16	12	12	11	10	10
Service delivery rate range (Mbps)					
0 (m_s) to 10 (M_s)					
Gradient-based step size λ (observed to converge both rapidly and smoothly)					
0.01					

$$U_3(x_{s=3,i}) = \frac{lg(x+1)}{lg11} \tag{20}$$

Fig. 3 is the visual representation of these three utility functions.

B. Experimental Setup

As illustrated in Fig. 2, the topology of this fog network originally contains 12 links labeled as $l_1, l_2, ..., l_{12}$. Since the network traffic that happens in this IoT network naturally is in the bottom-up convergent manner, and the connection between the gateway fog node and its child node is most likely to result in the communication overhead. Thus, it is rational to assign this link with the highest link capacity (16Mbps). The link capacities of other connections are generally in the decreasing manner through this top-down tree structure. By following this experiment design, we can divide the links into several levels based on their capacities, where l_2 ranks the second (12Mbps), and l_3 to l_5 are set to be degrading gradually to increase the randomness (12Mbps, 11Mbps and 10Mbps, respectively) whereas things/users connected to IoT network through l_6 to l_{12} have the same values (10Mbps). It is noticeable that the status of the bottom-level link depends on things/users. In other words, links could be broken from or re-connected to the network if the user mobilizes from one fog controlled area to another. The detailed setup could be found in Table I

Furthermore, these links have been shared among service requesters residing in three service groups s_1, s_2 and s_3 with generic displays in service group 1, pervasive displays as the members of service group 2, and shoppers (both stationary and mobilizing) requesting service 3, where each receiver has 0 and 10Mbps as the minimum and maximum service rate (corresponding to m_s and M_s in Algorithm 1).

It could be clearly observed in Fig. 2 that there are eight service receivers. Since each fog node covers the limited area (each department) to alleviate the overall communication overhead, and despite the fixed-position displays, one could move around the mall to different shopping departments in reality, yet good QoS is expected to retain regardless. Hence, a designated moving trajectory of a customer is also considered in the experiment, i.e., shopper stays at the woman's department at first, starts walking and taking a rest at the food court, and finally arrives at the children's department.

To be more specific, the customer labelled as $r_{3,3}$ gets connected to the network through the bottom-left fog node at the beginning, requesting the web browsing service from the service provider 3 up to a timestamp. Then she starts moving to other different areas as time elapsed. The detailed

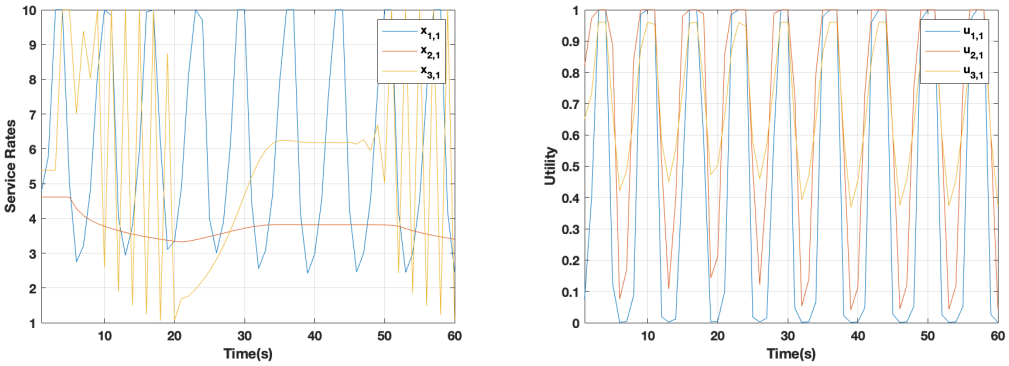

Fig. 4: Simulation results from the original, unmodified algorithm

moving trajectory can be found in Fig.2 (arrowed line). The rationale behind this setting is to verify the stability and adaptability of the proposed platform under this commonly happened situation in the real-world shopping mall.

C. Experimental Results

The performance of the rate adaptive fog service platform is evaluated through MATLAB simulations. The results focus on demonstrating the adaptability, expandability, and robustness of the fog platform. Herein, the negative and positive experiment groups are considered separately for the comparison purpose.

1) Case 1: We start off pointing out the ineffectiveness of applying state-of-the-art resource allocation strategy in [25] through our proposed fog platform for heterogeneous IoT applications.

Three service receivers, randomly picked from different service groups, are taken out to show the corresponding service delivery rates as well as QoS. For simplicity, the mobilizing shopper scenario is not considered in this comparison group. As seen from Fig. 4, three receivers, $r_{1,1}$, $r_{2,1}$ and $r_{3,1}$, are connected to the network since the beginning of the simulation, yet are not able to receive the expected service with sable transmission rate (corresponding to $x_{1,1}$, $x_{2,1}$ and $x_{3,1}$, respectively) and satisfied QoS until the end of the simulation (at 60s in this case). There is a clear oscillation observed throughout the whole period of Case 1 simulation, indicating that the conventional algorithm in the literature is unable to support both general elastic services and inelastic real-time services at the same time, even though it works for sole elastic services.

2) Case 2: The Case 2 simulation is considered as the positive example demonstrating the effectiveness of our fog platform in the shopping mall use case.

The simulations start at time $t = 0$, and each service group contains several receivers at the beginning. More specifically, groups one and two are to deliver inelastic real-time video services containing generic digital displays $r_{1,1}$, $r_{1,2}$ and pervasive displays $r_{2,1}$, $r_{2,2}$, and $r_{2,3}$, respectively. In contrast, group three serves the purpose of provisioning elastic services, thus covers the situation of both stationary shoppers ($r_{3,1}$, $r_{3,2}$) and a moving shopper ($r_{3,3}$). In the beginning, the service delivery rates to each receiver are randomly set to be a value in the range between 0 and 10Mbps but is expected to be adapted by providers rapidly based on feedback of the network condition. The platform triggers the algorithm at the bottom-level fog node so that the network status could be collected and properly initialized. Since then, all fog node would constantly monitor the network in a collaborative manner, and important factors such as service transmission rates and utility are expected to reach a stable state promptly. It is also noticeable that as receiver $r_{3,3}$ leaves area 1 and enters area 2 at $t = 60s$, the network encounters the link breaking and recovery situation (dashed-line link in Fig. 2), which is the same when she keeps moving to the last area, i.e. area three (at $t = 120s$). The abrupt changes of topology are not uncommon in the real world, which is hereby used to validate the expandability and robustness of the platform.

The simulation results of service delivery rates ($x_{s,i}$) in these three service groups are shown in Fig. 5. We can discover that all service rates converge to the global optimum under the complex IoT network conditions, which indicates that the globally optimal allocation of service rates is well accomplished. Moreover, even with the abrupt network changes (with the shopper $r_{3,3}$ moving to different areas), the platform is capable of eliminating the instability, and relevant fog service providers will swiftly adapt service rate for each receiver to maintain relatively good QoS. The minimum service delivery rate achieved in this scenario is around 4Mbps, which substantially suffices the majority of service needs in the shopping mall use case [35]. Apart from that, the generic digital display $r_{1,1}$ is designed to be isolated with which no resource competition happens. It represents one variation of experimental setting under the platform and attains the highest service transmission rate throughout the whole period.

The utility results in Fig. 6 are used as an indicator of the overall user's satisfaction and QoS achieved by the platform in which all utilities are more than or around 0.4 even with the fierce resource competition among bottleneck links. In particular, the underlying network resources, bandwidth in this case, have been optimally allocated to each receiver to accomplish the stability of QoS promptly. Among all, $u_{1,1}$ enjoys the highest satisfaction owing to no resource competition happened in its connected link. Another interesting

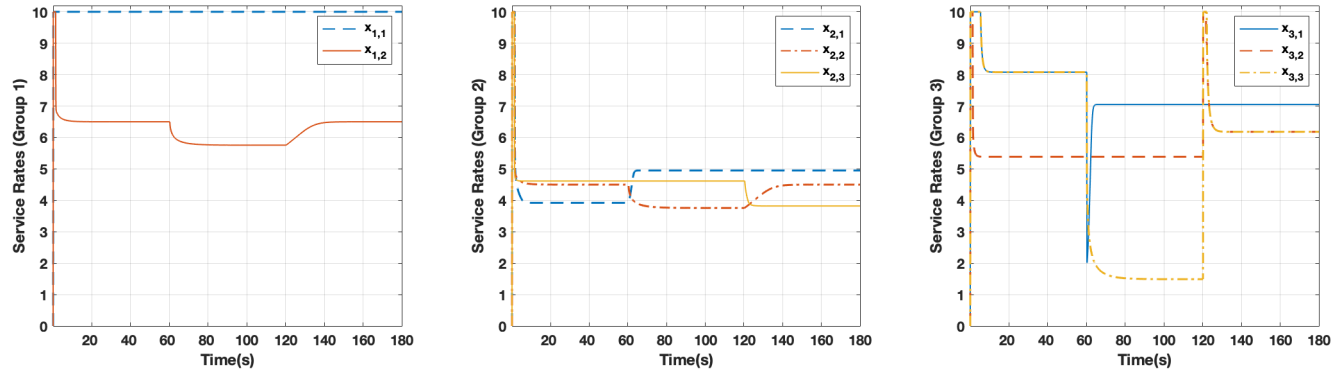

Fig. 5: Simulation results of service rates: Rate changes for receivers in group 1 (left), group 2 (middle) and group 3 (right). $x_{3,3}$ is the rate of the mobilizing shopper, and it affects the rates of other receivers when it joins/leaves the network from that area.

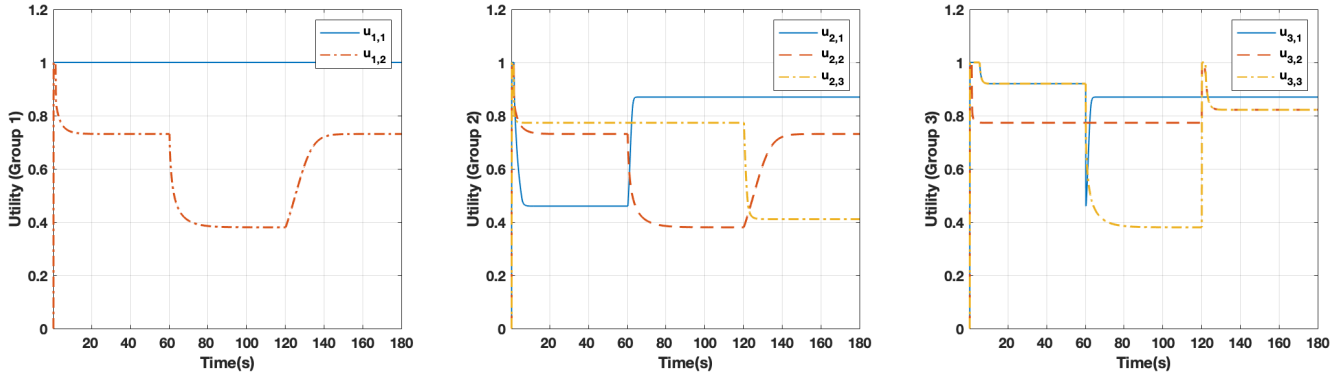

Fig. 6: Simulation results of utilities (QoS): The impact on QoS for each service receiver when there is a new joiner/leaver in that area. The QoS is maintained to a large extent owing to the resource allocated both fairly and optimally.

result shown in Fig. 6 regards the fairness which ensures that most service competitors at least will achieve the same level of utility values (QoS). As pointed out in [14], the utility maximization derived OFC approach used in work like [25] can lead to a seriously unfair situation for network resource allocations (oscillations of utility as observed in Fig. 4), yet our work allows a fair traffic distribution to receivers who are even under the most tense resource competitions, in which at least the same QoS could be accomplished. Apart from that, the QoS of each receiver will achieve a higher value as of the increase of link capacity or decrease of the total number of receivers, which means that by enhancing the throughput from link l_1 to l_{12}, e.g., using fibre optical communication links in fog-to-fog connection instead, one can easily observe the increase of QoS to a large extent. Equivalently, the preliminary work in [1] shows the decreasing number of connected receivers could lead to the same goal (at least 0.5 of QoS in that work).

To conclude, the simulation results re-confirm that our proposed platform is both practical and robust in a real-world scenario. This platform can be naturally integrated into the IoT environment, and our algorithm clearly demonstrates its performance in dealing with heterogeneous IoT applications. Furthermore, the convergence of service delivery rates and corresponding utilities prove that, under complex network conditions, the platform could help distribute the service, adapt the service rate, and be able to expand smoothly.

TABLE II: iFogSim simulation setup

Configuration of the running PC		
OS	CPU	RAM
mac OS	2.6 GHz Intel Core i5	8 GB
Configuration of each fog node		
Device Type	CPU	RAM	POWER
Service 1 provider	3.0 GHz	8 GB	214.678(M) 106.82(I) W
Service 2&3 provider	2.0 GHz	4 GB	107.339(M) 83.433(I) W
Other service forwarders	1.6 GHz	4 GB	107.339(M) 83.433(I) W

D. The Study on Real-world Deployment

To explore further on the aspects of service transmission latency and energy consumption, verify the effectiveness of supporting delay-sensitive applications as well as the plausibility of real-world deployment of our platform, we adopt the fog simulator, namely iFogSim [36], to model the IoT and fog environment that our platform is built upon. We also incorporate the algorithm to observe the impact on the IoT network.

In iFogSim, we customize the fog structure the same as the one in shopping mall use case and deploy three IoT applications at different fog nodes to provide the corresponding real-time services as described earlier (**Services 1, 2** and **3**). To be more technically specific, each application essentially

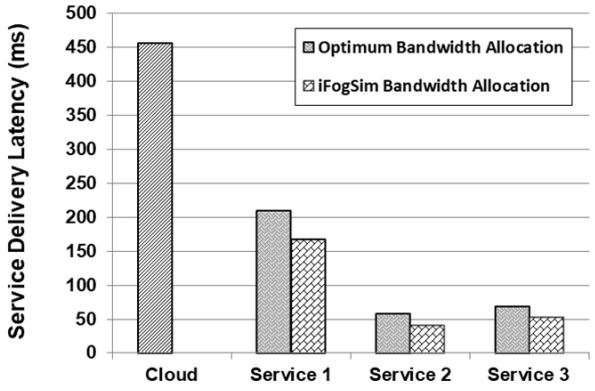

Fig. 7: Service delivery latency for services 1, 2 and 3

creates an **Application Model**, which can be instantiated and placed inside the fog. When the simulation starts, iFogSim will monitor the service delivery latency through **AppModule** happening in the service continuum (service requester - service forwarder - service provider then sends the service back to the requester in the reversed path). One advantage of adopting this simulator is that it not only calculates the service transmission latency but also takes into account the service execution time from each provider, so the total service delivery latency observed through iFogSim is thus more reflective of the real-world deployment. When it comes to the evaluation of our proposed service rate adaptive algorithm, it allocates the bandwidth resources both fairly and optimally, which is also used as an indicator to analyze our algorithm's impact on service transmission latency. The **Power Monitoring Module** in the simulator toolkit, on the other hand, continues to record the power consumption status for each fog node involved in operating the service continuum. It is worth mentioning that, as pointed out in [36], the simulator utilizes a model named PowerModelLinear to continuously cumulate the energy usage of each fog node instance based on the configuration of each node, including the CPU, RAM, power usage for both busy and idle states.

To demonstrate the superiority of our platform with respect to facilitating the real-time service delivery, we seamlessly integrate our proposed algorithm into the simulator and compare the corresponding service delivery latency with the ones generated by the default strategy concerning the link resource allocation in iFogSim and the traditional cloud approach (configuration details in Table II), respectively [36]. Fig. 7 concludes the service delivery latency brought from requesting to each service provider. It could be observed that owing to the optimum bandwidth allocation derived from our algorithm, the caused service delivery latency exhibits an evident decline as opposed to the latency caused by the default resource allocation strategy in the simulator. It shows that Service 2 and Service 3 have relatively low service delivery latency (40.4ms and 52.6ms, respectively), and Service 1 comes later yet at the same scale (164.7ms). The service latency from the cloud, on the other hand, is almost 9 to 10 times higher than that in the fog platform, which confirms that the fog-enabled service platform can cater for delay-sensitive applications effectively.

TABLE III: Energy consumption status

Energy consumed for service provisioning	MegaJoules
Service 1 provider	0.87
Service 2&3 provider	0.98
Other service forwarders	0.64
Fog platform in total	2.49
Cloud datacenter	8.695

From the energy consumption perspective, Table III lists the energy consumed by each service provider and other service forwarders in our platform, the overall energy consumed by the platform, and total consumption if all applications are simply deployed at the cloud. It shows that our proposed platform excels in saving energy consumption as well.

VII. Conclusion and Future Work

In this paper, we develop a novel fog service platform that highlights the capabilities of supporting heterogeneous IoT applications and service delivery rate adaptation. Issues in the traditional service-oriented network such as service transmission latency, huge bandwidth waste and sole support for elastic service have been addressed through the proposed platform. More specifically, various IoT services now are offered in the vicinity of end users/devices, as fog node could serve as providers that are only a few hops away. Additionally, fog nodes in this platform work collaboratively to maintain the stability of the IoT network. Our case study verifies that building on the top of fog architecture, the fog service platform seamlessly integrates service rate adaptive algorithm, and copes with real-world scenarios effectively even with the abrupt change of IoT network (new joiner or leaver). Moreover, the exploration of real-world deployment through iFogSim re-assures the effectiveness of our platform.

We also believe that our proposed platform brings up some exciting research opportunities in the area of service computing under the fog architecture that we will investigate further as our future work. One example is the service provider migration scheme, where the platform can handle the failure of the service provider and dynamically migrate the service provision task to nearby fog node based on features such as residue resources and network conditions.

VIII. Acknowledgement

This work is partially supported by Australian Research Council Discovery Project Grants DP190102828 and DP180100212, and Australian Government Research Training Program Scholarship. We thank all the anonymous reviewers for their valuable comments and suggestions.

References

[1] T. Zhang, J. Jin, and Y. Yang, "RA-FSD: A rate-adaptive fog service delivery platform," in *International Conference on Service-Oriented Computing*. Springer, 2018, pp. 246–254.

[2] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112–121, 2014.

[3] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the SOA-based Internet of Things: Discovery, query, selection, and on-demand provisioning of web services," *IEEE Transactions on Services Computing*, vol. 3, no. 3, pp. 223–235, 2010.

[4] J. V. Nguyen, "System and method for designing, developing and implementing internet service provider architectures," Jul. 21 2015, US Patent 9,087,319.

[5] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for Internet of Things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer, 2014, pp. 169–186.

[6] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[7] Q. Duan, Y. Yan, and A. V. Vasilakos, "A survey on service-oriented network virtualization toward convergence of networking and cloud computing," *IEEE Transactions on Network and Service Management*, vol. 9, no. 4, pp. 373–392, 2012.

[8] W.-T. Tsai, X. Sun, and J. Balasooriya, "Service-oriented cloud computing architecture," in *International Conference on Information Technology: New Generations*. IEEE, 2010, pp. 684–689.

[9] D. Yu, Y. Jin, Y. Zhang, and X. Zheng, "A survey on security issues in services communication of microservices-enabled fog applications," *Concurrency and Computation: Practice and Experience*, pp. 4436–4436, 2018.

[10] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.

[11] R. D. Callaway, M. Devetsikiotis, Y. Viniotis, and A. Rodriguez, "An autonomic service delivery platform for service-oriented network environments," *IEEE Transactions on Services Computing*, vol. 3, no. 2, pp. 104–115, 2010.

[12] T. Wang, Z. Yao, B. Zhang, C. Li, and K. Hao, "Adaptive flow rate control for network utility maximization subject to QoS constraints in wireless multi-hop networks," *Peer-to-Peer Networking and Applications*, vol. 11, no. 5, pp. 881–899, 2018.

[13] E. E. Tsiropoulou, P. Vamvakas, and S. Papavassiliou, "Joint customized price and power control for energy-efficient multi-service wireless networks via s-modular theory," *IEEE Transactions on Green Communications and Networking*, vol. 1, no. 1, pp. 17–28, 2017.

[14] J. Jin, "Flow control and performance optimization for multi-service networks," Ph.D. dissertation, University of Melbourne, 2010.

[15] W. He, G. Yan, and L. Da Xu, "Developing vehicular data cloud services in the IoT environment," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1587–1595, 2014.

[16] K. H. Kim, A. Beloglazov, and R. Buyya, "Power-aware provisioning of virtual machines for real-time cloud services," *Concurrency and Computation: Practice and Experience*, vol. 23, no. 13, pp. 1491–1505, 2011.

[17] M. Shojafar, N. Cordeschi, and E. Baccarelli, "Energy-efficient adaptive resource management for real-time vehicular cloud services," *IEEE Transactions on Cloud computing*, 2016.

[18] L. Chen, P. Zhou, L. Gao, and J. Xu, "Adaptive fog configuration for the Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4656–4664, 2018.

[19] X. Xu, S. Fu, Q. Cai, W. Tian, W. Liu, W. Dou, X. Sun, and A. X. Liu, "Dynamic resource allocation for load balancing in fog environment," *Wireless Communications and Mobile Computing*, 2018.

[20] S. F. Abedin, M. G. R. Alam, S. A. Kazmi, N. H. Tran, D. Niyato, and C. S. Hong, "Resource allocation for ultra-reliable and enhanced mobile broadband IoT applications in fog network," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 489–502, 2019.

[21] X. Wang, L. T. Yang, X. Xie, J. Jin, and M. J. Deen, "A cloud-edge computing framework for cyber-physical-social services," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 80–85, 2017.

[22] A. Samanta, L. Jiao, M. Mühlhäuser, and L. Wang, "Incentivizing microservices for online resource sharing in edge clouds," in *IEEE International Conference on Distributed Computing Systems*, 2019.

[23] A. Samanta, Z. Chang, and Z. Han, "Latency-oblivious distributed task scheduling for mobile edge computing," in *IEEE Global Communications Conference*. IEEE, 2018, pp. 1–7.

[24] T. K. Phan, D. Griffin, E. Maini, and M. Rio, "Utility-centric networking: Balancing transit costs with quality of experience," *IEEE/ACM Transactions on Networking*, vol. 26, no. 1, pp. 245–258, 2018.

[25] W.-H. Wang, M. Palaniswami, and S. Low, "Necessary and sufficient conditions for optimal flow control in multirate multicast networks," *IEEE Proceedings in Communications*, vol. 150, no. 5, pp. 385–390, 2003.

[26] A. Samanta and Z. Chang, "Adaptive service offloading for revenue maximization in mobile edge computing with delay-constraint," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3864–3872, 2019.

[27] C. Wöbker, A. Seitz, H. Mueller, and B. Bruegge, "Fogernetes: Deployment and management of fog computing applications," in *Network Operations and Management Symposium*. IEEE, 2018, pp. 1–7.

[28] J. Jin, W.-H. Wang, and M. Palaniswami, "Utility max–min fair resource allocation for communication networks with multipath routing," *Computer Communications*, vol. 32, no. 17, pp. 1802–1809, 2009.

[29] Y. Li, A. Papachristodoulou, M. Chiang, and A. R. Calderbank, "Congestion control and its stability in networks with delay sensitive traffic," *Computer Networks*, vol. 55, no. 1, pp. 20–32, 2011.

[30] N. Davies, S. Clinch, and F. Alt, "Pervasive displays: Understanding the future of digital signage," *Synthesis Lectures on Mobile and Pervasive Computing*, vol. 8, no. 1, pp. 1–128, 2014.

[31] Y. Zhang, X. Ai, Q. He, X. Zhang, W. Dou, F. Chen, L. Chen, and Y. Yang, "Personalized quality centric service recommendation," in *International Conference on Service-Oriented Computing*. Springer, 2017, pp. 528–544.

[32] M. Yang, T. Zhu, B. Liu, Y. Xiang, and W. Zhou, "Machine learning differential privacy with multifunctional aggregation in a fog computing architecture," *IEEE Access*, vol. 6, pp. 17 119–17 129, 2018.

[33] D. Roca, R. Milito, M. Nemirovsky, and M. Valero, "Tackling IoT ultra large scale systems: Fog computing in support of hierarchical emergent behaviors," in *Fog Computing in the Internet of Things*. Springer, 2018, pp. 33–48.

[34] H. Atlam, R. Walters, and G. Wills, "Fog computing and the internet of things: a review," *Big Data and Cognitive Computing*, vol. 2, no. 2, p. 10, 2018.

[35] M. Karam, T. Payne, and E. David, "Evaluating bluscreen: Usability for intelligent pervasive displays," in *International Conference on Pervasive Computing and Applications*. IEEE, 2007, pp. 18–23.

[36] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.

[37] M. A. Hanson, "On sufficiency of the Kuhn-Tucker conditions," *Journal of Mathematical Analysis and Applications*, vol. 80, no. 2, pp. 545–550, 1981.

APPENDIX A
PROOF OF THEOREM 1

In this appendix, we will list out the step-by-step derivation of our analytical framework, which will lead us to the final results that are further used in the rate adaptive algorithm.

All the math notations remain consistent with Section IV, and we start off looking at the Lagrangian optimisation problem formed:

$$L(x,p) = \sum_{s\in S}\sum_{i=1}^{n_s}\mathcal{U}_s(x_{s,i}) - \sum_{l\in L} p^l\left[\sum_{s\in S}\left(\sum_{\{i|l\in L_{s,i}\}} x_{s,i}^N\right)^{\frac{1}{N}} - c_l\right] \quad (21)$$

In order to solve this constrained optimisation problem, Kuhn-Tucker theorem [37] is then applied:

$$\frac{\partial L(x,p)}{\partial x} = 0 \quad (22)$$

$$p_l\frac{\partial L(x,p)}{\partial p_l} = 0, \forall l \in L \quad (23)$$

By solving the partial derivatives equations (21) and (22), we have the optimal solution of P2:

$$\begin{aligned}\mathcal{U}_s^{'}(x_{s,i}) &= \frac{\partial\Big(\sum_{l\in L} p_l\Big(\sum_{s\in S}\Big(\sum_{\{j|l\in L_{s,j}\}} x_{s,j}^N\Big)^{\frac{1}{N}} - c_l\Big)\Big)}{\partial x_{s,i}} \\ &= \sum_{l\in L_{s,i}} p_l\Big(\sum_{\{j|l\in L_{s,j}\}} x_{x,j}^N\Big)^{\frac{1-N}{N}} x_{s,i}^{N-1} \\ &= \sum_{l\in L_{s,i}} p_l\Big(\frac{x_{s,i}^N}{\sum_{\{j|l\in L_{s,j}\}} x_{s,j}^N}\Big)^{\frac{N-1}{N}}\end{aligned} \tag{24}$$

$$p_l\Bigg[\sum_{s\in S}\Big(\sum_{\{j|l\in L_{s,j}\}} x_{s,j}^N\Big)^{\frac{1}{N}} - c_l\Bigg] = 0, \forall l \in L \tag{25}$$

Based on (23), we approximate the result as:

$$p_{s,i} = \sum_{l\in L_{s,i}} p_l\Big(\frac{x_{s,i}^N}{\sum_{\{j|l\in L_{s,j}\}} x_{s,j}^N}\Big)^{\frac{N-1}{N}} \tag{26}$$

where $p_{s,i}$ represents the path price of receiver $r_{s,i}$ to source s, leading to:

$$\mathcal{U}_s^{'}(x_{s,i}) = p_{s,i} \tag{27}$$

As we have used the redefined pseudo utility function so far to ensure its concavity, the global optimal value could be achieved, and it could be transformed in the format of original utility function by combining (1) and (26). Therefore, (10) could be easily derived.

As mentioned in Section IV, when N is a big enough number and goes to ∞, problem $P2$ ultimately coverts to the original problem $P1$. Under this condition, we define the variable, namely price weighting coefficient, $w_{s,i}^l$ of the receiver $r_{s,i}$ at link l as:

$$\begin{aligned}w_{s,i}^l &= \lim_{N\to\infty}\Big(\frac{x_{s,i}^N}{\sum_{\{j|l\in L_{s,j}\}} x_{s,j}^N}\Big)^{\frac{N-1}{N}} \\ &= \lim_{N\to\infty}\frac{x_{s,i}^N}{\sum_{\{j|l\in L_{s,j}\}} x_{s,j}^N}\end{aligned} \tag{28}$$

from which the path price (13) could be inferred along with (25). It then completes the proof of Theorem 1.