RA-FSD: A Rate-Adaptive Fog Service Delivery Platform

Tiehua Zhang(✉), Jiong Jin, and Yun Yang

School of Software and Electrical Engineering,
Swinburne University of Technology, Melbourne, Australia
{tiehuazhang,jiongjin,yyang}@swin.edu.au

Abstract. As the Internet of Things (IoT) technologies permeate people's daily lives, the sheer number of IoT applications has been developed to provide a wide range of services. Among all, real-time IoT services start to draw increasing attentions. Conventionally, cloud plays the role as the service provider in IoT but is no longer considered as the rational option for the real-time services due to service transmission latency and communication overhead. Therefore, we propose a novel rate-adaptive fog service delivery platform, namely RA-FSD, aiming at real-time service provisioning and network utility maximization (NUM) of the underlying IoT resources based on the newly emerged fog computing paradigm. The platform leverages fog nodes as either fog service provider to offer timely services for end users, or service intermediaries to help track network conditions and mitigate communication overhead. By doing so, service consumers would always benefit from the fact that services produced by IoT applications are in their proximity and thus delivered to destination in a prompt manner. A service rate-adaptive algorithm is also developed as the key component of the RA-FSD platform to handle the abrupt changes happened in IoT network, dynamically adjust service delivery rate based on the network condition while retaining satisfactory Quality of Service (QoS) to each service consumer, and support both elastic and inelastic network services from heterogeneous IoT applications.

Keywords: Fog computing · Internet of Things (IoT)
Service-oriented networking · Network utility maximization (NUM)
Quality of Service (QoS)

1 Introduction

The proliferation of IoT technology has brought the unprecedented convenience to people and draw widespread attentions from both academia and industry [3]. Currently, cloud plays a major role in providing these highly personalized, context-aware IoT services because of the advantages like cost reductions, easy deployment and strong reliability. However, the new challenges presented by

C. Pahl et al. (Eds.): ICSOC 2018, LNCS 11236, pp. 246–254, 2018.
https://doi.org/10.1007/978-3-030-03596-9_16

real-time IoT applications like stringent latency and Quality of Service (QoS) requirement are not well addressed by the standalone cloud computing paradigm.

In order to address aforementioned issues and overcome inadequacy of the cloud, fog computing is introduced. The idea of extending cloud to the edge of network and closer to end users has been viewed as an alternative with the overarching goal of "off-loading" from the cloud. Fog nodes, acting as the proxy of both cloud and end devices/users (*things*), could be equipped with computing, storage and networking resources to accommodate various IoT applications. Therefore, these applications could be deployed in fog rather than the conventional approach on either resource-constrained IoT devices or remote cloud. In this regard, fog and cloud complement each other to form a service continuum that distributes respective services to end users [2].

Along with the rapid growth of IoT applications, heterogeneous services are tailored for service consumers with certain QoS guarantee. In reality, stable service delivery rate is a key component to meet the QoS and it could act as a major part in service consumers perception with regard to overall performance of service invocation [1]. In our work, the utility function is used to model the QoS performance, which increases as the increasing of service delivery rate. From the utility point of view, the services can be categorized into two main groups, i.e., traditional elastic services (e.g., file transfer, data analysis and web browsing), in which each service attains a strictly increasing and concave utility function to measure its QoS performance, and real-time inelastic services provided by audio, video and multimedia real-time applications. Such inelastic services have an intrinsic bandwidth threshold in nature and adopt the sigmoid-like functions to describe the corresponding QoS [5].

There are several previous efforts made towards developing service delivery architecture to connect service consumers and providers in IoT environment. A vehicular data cloud platform is proposed in [4] to provide real-time information, yet concerns like service transmission latency and transportation cost are not discussed. Some service platforms aim to utilize underlying IT resources to achieve good QoS through NUM, e.g, the automatic service routing platform proposed in [1] and the multicast multirate service delivery platform in [8], but their works either are not designed for IoT environment, or fail to support inelastic, real-time IoT services. Given that, the advantages of RA-FSD platform are highlighted as follows: (1) it seamlessly integrates fog computing into IoT environment, and utilizes fog node as service provider equipped with more computation and storage resources than traditional IoT devices; (2) it offers services that are generated in the proximity of users, and both service transportation cost and delivery latency are largely reduced; (3) beside elastic services, it also supports real-time inelastic services.

2 Architecture of RA-FSD Platform

In this section, we introduce the architecture of RA-FSD platform and the components inside. The platform is composed of heterogeneous *things*, fog nodes and

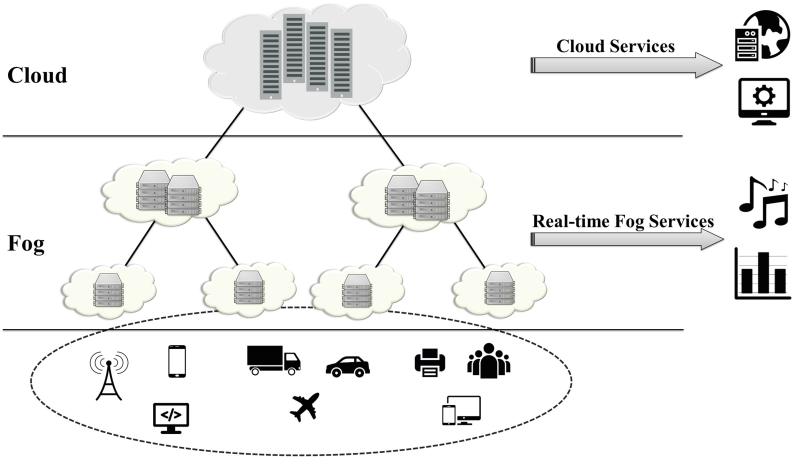

Fig. 1. Fog computing architecture, and example services provided by cloud and fog nodes

the cloud. From the service-oriented computing perspective [7], *things* normally play the role of service consumers/requesters, fog nodes equipped with computation, storage and networking power could serve as either service providers or service intermediaries, 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 in the edge network, it is noticeable that the use of cloud is no longer mandatory under this platform, but one could choose to continue using cloud as service provider for particular services, e.g., advanced analytic service for big data.

Figure 1 presents a simple fog architecture with a multi-level hierarchy, including different types of service provisioned inside. The bottom layer consists of the end devices or users. The fog nodes have been positioned in the middle two layers, and similar to the traditional IoT network, the cloud is at the top layer. Taking advantage of the geographical locations, IoT applications could deploy on the fog nodes in the vicinity of *things* and fog node is thus able to serve as the real-time service provider in IoT network. In addition, fog node selected as provider would form the corresponding service group in which a particular service could be disseminated. Specifically, the provider in each group will gather feedback regarding downstream network conditions reported by service intermediaries to adjust transmission rate for different service receivers. Service intermediaries, on the other hand, collaborate with providers and could be converted to the providers if needed, which increases the scalability and flexibility of the platform. In general, the cloud is only used as service provider if that service associates with a large volume of data processing and storage, or service requested is delay-tolerant.

3 Analytic Framework and Optimization Problem

The NUM resources allocation problem in the fog-based IoT environment is formulated in this section to make it support both elastic and real-time inelastic services. In addition, we characterize the utility in terms of allocated service transmission rate deriving from the underlying bandwidth of IoT network. Consider a fog service delivery network consisting of a set of links $L = \{1, 2, ..., l\}$, each of which has respective capacity c_l. There is a set of $S = \{1, 2, ..., s\}$ service groups, and each service group is devoted to providing one 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 in service group s could be defined 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 for 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 nodes.

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. Utility function $U_s(x_{s,i})$ has been selected on per-service basis, which is customized to describe the QoS requirement. In addition, utility function should be strictly increasing and continuously differentiable, but needs not be concave in our work. $x_{s,i}$ represents the service delivery rate to receiver i in service group s.

We then formulate the following optimization problem, on the basis of multicast, multirate model similar to [8]:

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

$$\text{subject to} \quad \sum_{s \in S} x_s^l \leq c_l, \quad \forall l \in L \tag{2}$$

$$x_s^l = \max_{\{i | l \in L_{s,i}\}} x_{s,i} = \lim_{N \to \infty} \left(\sum_{\{i | l \in L_{s,i}\}} x_{s,i}^N \right)^{\frac{1}{N}} \tag{3}$$

In Eq. (3), $\{i | l \in L_{s,i}\}$ 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 (2) in this optimization problem means that the aggregate service rate on link l across all service groups should not exceed the link capacity (network condition). Then we replace (3) in (1) and the Lagrangian multiplier could be yielded:

$$L(x, p) = \sum_{s \in S} \sum_{i=1}^{n_s} U_s(x_{s,i}) - \sum_{l \in L} p^l \left[\sum_{s \in S} \left(\sum_{\{i | l \in L_{s,i}\}} x_{s,i}^N \right)^{\frac{1}{N}} - c_l \right] \tag{4}$$

In order to make our platform support both elastic and inelastic services, a pseudo utility function [5] is constructed as (5) to substitute the original optimization problem $P1$:

$$\mathcal{U}_s(x_{s,i}) = \int_{m_s}^{x_{s,i}} \frac{1}{U_s(y)} dy, \quad m_s \le x_{s,i} \le M_s \tag{5}$$

where m_s and M_s represent minimum and maximum service delivery rate, respectively. Based on the characteristic of original utility function $U_{s,i}(x_{s,i})$, this pseudo utility function must be strictly increasing and concave as $\mathcal{U}_s''(x_{s,i}) < 0$. By solving the optimization problem, the derived results could be further incorporated into the service rate-adaptive algorithm.

4 Service Rate-Adaptive Algorithm and Implementation

The algorithm is devised to take advantage of the architectural support from the platform and is deployed on all fog nodes to periodically check the network conditions as well as adapt the service delivery rates accordingly. By choosing appropriate fog node as service provider, the corresponding service generated could traverse less hops than the cloud scheme and thus make it suitable for delay-sensitive IoT applications.

```
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: Establish Service group $S = [s^1, s^2 .... s^n]$ and gather link capacity $C =$
    $[c^1, c^2 .... c^l]$
 3: $W, P \leftarrow R^S \times L$ matrix,where $w^l_{s,i} = \frac{1}{\{j|l \in L_{s,j}\}}, p^l_{s,i} = 0$ at initial stage
Phase 2: Adjusting service rate
 1: Starting at bottom fog nodes at every interval $t$
 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 $i$, whose service transverses link $l$
12:             $w^l_{s,i} = [w^l_{s,i} + \lambda(x_{s,i} - x^l_s)]^+$
13:         if receiver $i$ receives service s at rate $x^l_s$ then
14:             $w^l_{s,i} = 1 - \sum_{\{j|j \neq i, l \in L_{s,i}\}} w^l_{s,j}$
15:         end if
16:         5. calculate link price $p^l_{s,i}$ for each downstream receiver $i$ on link $l$
17:             $p^l_{s,i} = w^l_{s,i} p^l$
18:         6. update corresponding $w^l_{s,i}$ in $W$ and $p^l_{s,i}$ in $P$, respectively
19:         if $f$ is a provider of service s then
20:             for each receiver $i$ that receives service s do
21:                 7. calculate relevant path price
22:                     $p_{s,i} = \sum_{l \in L_{s,i}} p^l_{s,i}$
23:                 8. adjust service rate
24:                     $x_{s,i} = U_s^{-1}\left(\left[\frac{1}{p_{s,i}}\right]_{U_s(m_s)}^{U_s(M_s)}\right)$
25:             end for
26:         end if
27:     end for
28:     if there is any upstream service coming to $f$ then
29:         9. propagate network condition upward, and repeat phase 2
30:     end if
31: until (all providers have been reached)
```

Algorithm 1 presents a summary of the algorithm, which incorporates the analytic results derived in Sect. 3 and comprises two phases. The service intermediaries would firstly gather relevant downstream links information, then report it upwards recursively to form the fog service continuum (lines 2–3). 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 detect 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 links 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 11–17, which implies that this coefficient would continue to increase for receiver with the largest service rate, while decreasing among other receivers.

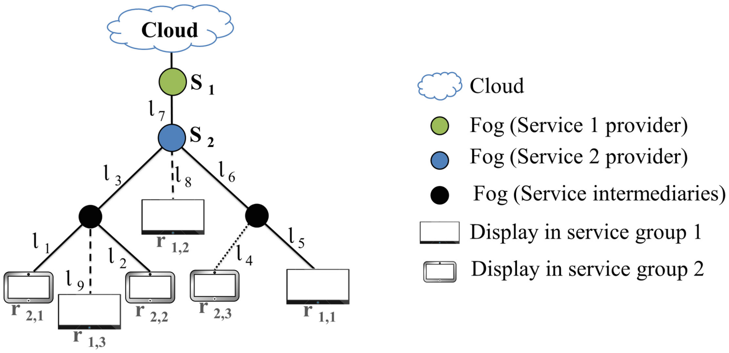

Fig. 2. Fog architecture in the shopping mall use case

Lines 18–27 handle the service rate adjusting process, if and only if current node f is a service provider.

5 Performance Evaluation on a Case Study

In this section, we verify the feasibility of proposed platform with the algorithm through a real-world shopping mall use case. Besides, the numerical results are meanwhile used to demonstrate the applicability and robustness of the RA-FSD platform. It is worthwhile mentioning that, as elastic service is relatively easier to be accommodated, our focus on the use case is to implement inelastic services originated from real-time applications.

Figure 2 illustrates the topology of the IoT network empowered by fog architecture in the shopping mall. In this topology, all fog nodes represented as the dot points are placed inside the shopping mall, along with two different sizes of digital displays acting like *things*. The topmost fog node is configured as the most powerful node among all, and operates as the main gateway of this autonomous network. In addition, cloud is only used for data backup purpose.

The performance of RA-FSD platform is evaluated through simulations. The fog network originally contains 7 links labeled as $l_1, l_2, ..., l_7$ with relevant capacities c = (6, 4, 10, 8, 8, 12, 18) (in Mbps), and these links have been shared between two service groups s_1, s_2. In addition, two types of screens require different services transmission rates to satisfy respective QoS requirement. The utility function $U_1(x_{s=1,i}) = \frac{1}{1+e^{-2(x-6)}}$ is used to reflect the QoS performance of inelastic service 1, and $U_2(x_{s=2,i}) = \frac{1}{1+e^{-2(x-4)}}$ for service 2.

The simulations start at time $t = 0$, and service group 2 contains relatively small screens $r_{2,1}, r_{2,2}, r_{2,3}$, while service group 1 only has one big screen $r_{1,1}$ at the start. The minimum and maximum service delivery rates to each screen are set to be 0 and 10 Mbps but will be adapted quickly based on the feedback of network condition. The platform triggers the algorithm at the bottom-level fog node at every time interval of 0.1 s and experimental results including service

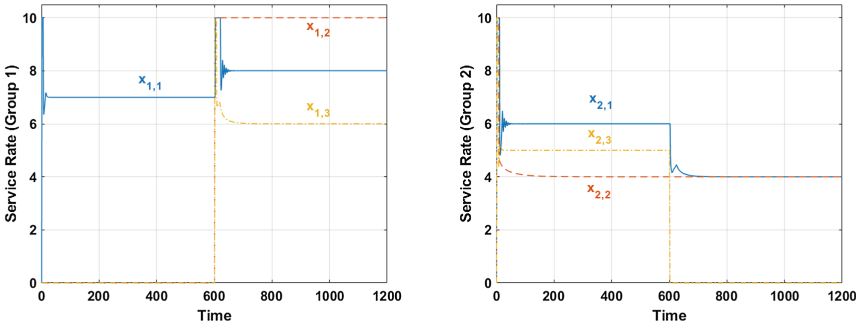

Fig. 3. The service delivery rates in different service groups

rate and utility are expected to reach a stable state rapidly to demonstrate the applicability of the platform. It is also noticeable that at $t = 60$ s, two new displays $r_{1,2}$ and $r_{1,3}$ have joined service group 1 and establish connections to fog nodes through links l_8 and l_9 (dashed-line links) with capacities of 10 and 8, respectively, while $r_{2,3}$ has suddenly left service group 2 along with a disconnection of link l_4. The abrupt changes of topology is not uncommon in real-world situation, which is hereby used to validate the robustness of the platform.

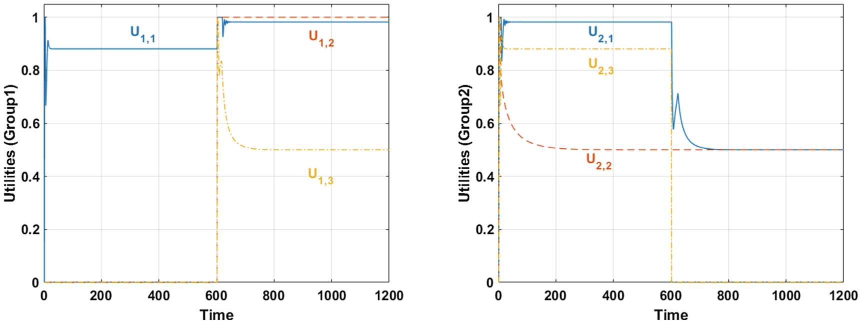

Fig. 4. The utility results in different service groups

The simulation results of service delivery rates ($x_{s,i}$) and utility results (QoS) in service groups 1 and 2 are shown in Figs. 3 and 4, respectively. It is clearly observed that all service rates converge to the optimum under the complex IoT network conditions even with the abrupt network changes. The platform is capable of eliminating the instability, and relevant fog service providers can concretely adapt service rate for each receiver to maintain a relatively good QoS. The minimum service delivery rate achieved in this scenario is 4 Mbps and the overall QoS achieved by the platform retains the value of more than 0.5, which substantially suffices the service requirements of displays in the shopping mall use case [6].

To conclude, the simulation results reconfirm that our proposed platform is applicable and robust in real-world scenario, and also capable of handling abrupt network changes. Furthermore, the convergence of service delivery rates and corresponding utilities clearly demonstrate that the platform could support real-time inelastic services offered by the fog service providers.

6 Conclusion and Future Work

In this paper, we have proposed a novel rate-adaptive fog service delivery platform that is applicable for current IoT network. Important issues in traditional service-oriented IoT network such as service transmission latency and huge bandwidth waste have been well addressed by applying RA-FSD. Heterogeneous IoT services now offered in the vicinity of *things* as fog node could serve as providers that are only a few hops away. Additionally, fog nodes in our platform work collectively to maintain the stability of the IoT network.

Our case study verifies that, on the basis of fog architecture, RA-FSD seamlessly integrates service rate-adaptive algorithm, and copes with real-world scenarios effectively even with the abrupt changes occurred in the IoT network (new joiner or leaver). Moreover, both elastic and inelastic IoT services are well supported in the platform. Our next phase of research will focus on developing the provider "pick up" strategy, in which RA-FSD platform should dynamically select a fog node as the service provider based on its characteristic such as computing power, geographic location or network condition.

Acknowledgements. This work is partly supported by Australian Research Council Discovery Project Grant DP180100212 and Australian Government Research Training Program Scholarship.

References

1. Callaway, R.D., Devetsikiotis, M., Viniotis, Y., Rodriguez, A.: An autonomic service delivery platform for service-oriented network environments. IEEE Trans. Serv. Comput. **3**(2), 104–115 (2010)
2. Chiang, M., Zhang, T.: Fog and IoT: an overview of research opportunities. IEEE Internet Things J. **3**(6), 854–864 (2016)
3. Guinard, D., Trifa, V., Karnouskos, S., Spiess, P., Savio, D.: Interacting with the SOA-based internet of things: discovery, query, selection, and on-demand provisioning of web services. IEEE Trans. Serv. Comput. **3**(3), 223–235 (2010)
4. He, W., Yan, G., Da Xu, L.: Developing vehicular data cloud services in the IoT environment. IEEE Trans. Ind. Inform. **10**(2), 1587–1595 (2014)
5. Jin, J., Wang, W.H., Palaniswami, M.: Application-oriented flow control for wireless sensor networks. In: International Conference on Networking and Services, p. 71. IEEE (2007)
6. Karam, M., Payne, T., David, E.: Evaluating bluscreen: usability for intelligent pervasive displays. In: International Conference on Pervasive Computing and Applications, pp. 18–23. IEEE (2007)

7. Tsai, W.T., Sun, X., Balasooriya, J.: Service-oriented cloud computing architecture. In: International Conference on Information Technology: New Generations, pp. 684–689. IEEE (2010)
8. Wang, W.H., Palaniswami, M., Low, S.: Necessary and sufficient conditions for optimal flow control in multirate multicast networks. IEE Proc.-Commun. **150**(5), 385–390 (2003)