EAI Endorsed Transactions
on Scalable Information Systems

Research Article EAI.EU

Software Defined Network-based Scalable Resource Discovery for Internet of Things

Mahbuba Afrin [1], Redowan Mahmud [1]

[1] Department of Computer Science and Engineering, University of Dhaka, Bangladesh,

email: m.afrin.ritu@gmail.com, ratul06oct@gmail.com

Abstract

Geo-distributed and heterogeneous Internet of Things (IoT) devices can generate huge amount of data. Inefficient management of IoT-data promotes network congestion and increases computational overhead on the data-processing entities. Traditional networking architecture, that is lack of functional abstraction and monitoring capabilities, often fails to meet the dynamics of IoT. Software Define Network (SDN) can be a viable alternative of the traditional networking architecture while dealing with IoT. In SDN, management, monitoring and context sensing of the connected components are simplifie and can be customized. In this paper, SDN-sensed contextual information of different components (computational entities, network, IoT devices) are combined together to facilitate scalable resource discovery in IoT. The proposed policy targets balanced processing and congestion-less forwarding of IoT-data. Through simulation studies, it has been demonstrated that the SDN-based resource discovery in IoT outperforms the traditional networking based approaches in terms of resource discovery time and Quality of Service (QoS) satisfaction rate.

Received on 09 April 2017; accepted on 13 July 2017; published on 25 September 2017

Keywords: Internet of Things, Resource Discovery, Software define network, Scalability , Service QoS

doi:10.4108/eai.25-9-2017.153149

1. Introduction

In recent years, the Internet of Things (IoT) has drawn significa t research interest. Due to rapid enhancement in hardware and communication technology, it is predicted that by 2020, there will be more than 50 billion active IoT devices [1]. IoT devices are geo-distributed, energy constrained and heterogeneous. The configu ations, applicability and sensing frequency of IoT devices are also diversified

Most of the IoT devices participate in real-time data sensing. As a consequence, the devices can generate huge amount of data within a minimal time. When a large number of IoT devices send data simultaneously towards the computational entities (e.g. Cloud, Fog nodes, Edge servers), it is more likely to create network congestion. Besides, random placement of IoT-data can increase processing overhead on the computational entities. In such scenario, efficiency of the underlying network in managing incoming IoT-data (data processing, data forwarding) is very crucial. However, due to lack of functional abstraction and inability in monitoring internal operations of the connected components (IoT devices, computational entities), the traditional networking architecture is not suitable for efficient IoT-data management. In this case, Software Define Network (SDN) can be adopted to overcome the shortcomings of traditional networking architecture in respect of IoT [2].

SDN is a very recent innovation in networking technology that operates through software system in place of specialized and dedicated hardware. It offers programmability of networking elements by decoupling network control plane and data forwarding plane [3]. In SDN, there exists a centralized entity that perceives the topology and status of the network. Based on perception, the centralized controller entity determines the data forwarding rules and notifie the rules to the data forwarding entities. Through abstraction of lower level

★Please ensure that you use the most up to date class file available from EAI at http://doc.eai.eu/publications/transactions/latex/

*Corresponding author. Email: m.afrin.ritu@gmail.com networking functionalities, SDN can set up, administrate, alter, and manage network behavior dynamically. In different computing paradigms (e.g. Cloud computing, Mobile edge computing), SDN based solutions have been explored extensively to meet auto-mated, on-demand service requests, handle mobility issues, ensure network reliability, etc. [4]. SDN-based solutions promotes virtualization of network, ensures flexibility in resource utilization, monitors internal operations of the connected components, senses contextual information, minimizes both capital and operational expenses. Although, networking among the sensors is the fundamental factor for IoT [5], SDN-based solutions for IoT have not been enlightened significantly. From the perspective of IoT, SDN-based solutions can play vital roles in resource discovery and load balancing.

Generally, resource discovery in IoT refers to find appropriate resources for processing IoT-data and its associate routing path to forward the data. In traditional networking architecture, the computational entities for processing IoT-data and the associate connections are predefined and static. Therefore, traditional static network architecture can not cope with the increasing number of IoT devices and their uncertain data load. As a result, QoS degradation in terms of network bandwidth and service delivery is widely observed. Taking cognizance of this fact, we investigate how SDN-based solutions can facilitate resource discovery in IoT. The proposed SDN-based solution incorporates contextual information from three different aspects (computational entities, network, IoT devices) while dealing with resource discovery in IoT to facilitate flexible data processing and congestion-less data forwarding. Besides, the proposed policy ensures dynamic management of IoT-data in SDN that can be scalable to certain extent according to the situation.

The major contribution of the paper are listed as:

- SDN-based solution for scalable IoT-resource discovery to facilitate uninterrupted data processing and data forwarding.
- Explored the applicability of SDN-sensed contextual information in managing uncertain load of IoT-data.
- Comparative study between SDN-based IoT resource discovery and traditional static network based approach in terms of resource discovery time and QoS satisfaction rate.

In the following section, several related works in this field are highlighted (Section. 2). In Section. 3 and 4 the system model and SDN-based IoT-resource discovery are discussed respectively. In Section. 5 performance evaluation is demonstrated. Section. 6 concludes the paper.

2. Related Works

Several research works on SDN has already been conducted in different areas of computation and networking. In [6], authors design a SDN-supported cloud computing environment through OpenFLow switches and controllers. They extend the features of OpenFLow controller in order to facilitate load balancing, less energy usage, and service monitoring. Besides, a queuing model is developed to claim the feasibility of the system. The SDN based solution aims at providing QoS satisfied cloud computing services.

In [7] some potential architectures of SDN-based Mobile Cloud has been proposed. The authors of the paper focus on identifying basic components of SDN-based Mobile Cloud that can deal with mobility and uncertain network status. Several frequency selection methods for data transmission have also been discussed. The feasibility of the SDN-based solution has been highlighted in terms of high packet delivery rate and system overhead.

The authors in [8] argued that with dense deployment of mobile devices and limited network bandwidth, it becomes difficult to assign radio resources for processing service requests. Besides, management of interference and load balancing between base stations get tough. To overcome these issues, authors propose a software defined radio access layer named "SoftRAN". It works as the centralized control plane for radio access network. According to the authors, SoftRAN can efficiently handle load distribution, manage interference within the network maximize the networking throughput.

In respect of scalability in SDN, the authors of [9], claimed that SDN scalability is free from inherent bottleneck. In that paper, the scalability of SDN controller has been discussed in details. Besides, the authors investigate the scalability in SDN in terms of overhead and fault tolerance. Since SDN reduces network programming and management complexity, SDN enhances the level of flexibility to accommodate network programming and management at any scale.

The impact of SDN in IoT has also been explored in several research works. In [10] a software defined framework is proposed that simplifies management of IoT-driven process and deals with dynamic challenging aspects of IoT in terms of forwarding, storing and securing sensed IoT-data. The framework integrates the software defined network, software defined storage, and software defined security into a single software defined based control model.

In [11] authors represent a software-defined IoT system for controlling flow and mobility in multi-networks named "UbiFlow". UbiFlow facilitates controllers entity to be placed distributively so that urban-scale SDN can be divided into different geographic partitions. In this case, a hash-based distributed overlay structure helps to maintain network scalability and consistency. Fault tolerance and load balancing are also handled by UbiFlow. Besides, it provides visibility over underlying network and optimizes the selection process of access points within multi-networks so that QoS satisfie IoT datafow can be ensured.

However, in the aforementioned works, the impact of SDN-sensed contextual information in IoT resource discovery has not been enlightened. Resource discovery plays an important role in not only ensuring QoS-satisfie processing of IoT-data but also managing network from being congested due uncertain load. Therefore, the paper aims at SDN-based resource discovery for IoT so that scalability in resource discovery for IoT-data processing and forwarding can be ensured.

3. System Model

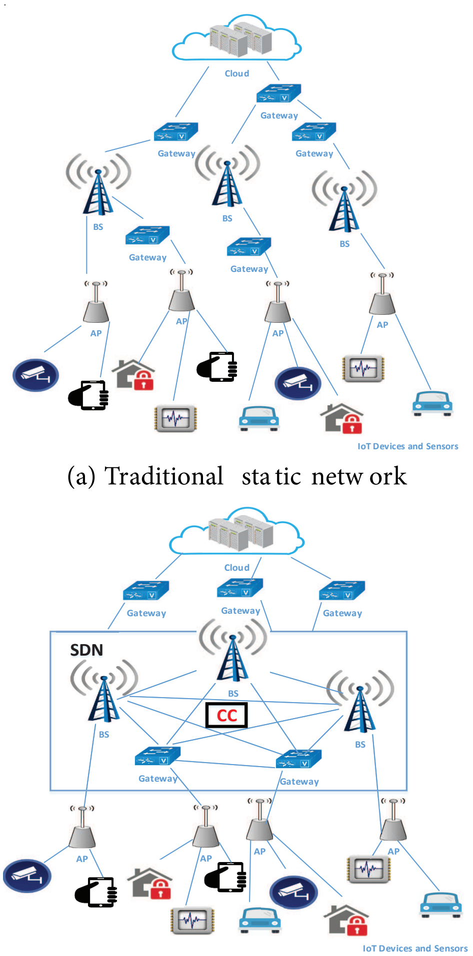
IoT-devices are geo-distributed and heterogeneous in terms of data sensing frequency and application-specific tion. Due to energy constraint, IoT-devices cannot process any sensed data but using communication protocols like Constrained Application Protocol (CoAP), Simple Network Management Protocol (SNMP), etc. can forward the sensed data towards Cloud or Fog for further processing. However, here we assumed that, the IoT-devices and the computational entities can interact through SDN.

Unlike traditional static networking architecture (as shown in Fig. 1.a), in SDN (as shown in Fig. 1.b), data forwarding plane is decoupled from the controller plane. Here, a Centralized Controller component (*CC*) determines the routing path and data forwarding rules. The other networking entities like switches, gateways, access points, base stations, etc. forwards the data according the guidelines of the *CC*.

In order to identify the efficient data routing path and computational entity, the *CC* senses the contextual information and monitors the internal operations of network, computational entities and IoT devices. In general, contextual information provides enriched perception regarding different system components [12]. Here, the contextual information includes:

- Current traffic load (network throughput) on different routing paths.
- Current data processing load (size of queued data) on each computational entity.
- Data sensing frequency (data transmission rate) of IoT devices.

As the components of the modelled system interacts with each other through SDN, it is possible to tarck the context of each components. Reference of several



(a) Traditional static network

(b) SDN-enabled network

Figure 1. Networking architecture for IoT.

context-sensing framework for SDN is available in the literature [13] [14]. Any of the frameworks can be applied to track the aforementioned contextual information of the computational entities, underlying network and IoT devices. The sensed contextual information helps *CC* to perceive the whole system efficiently and enhance the visibility over each of the components.

Due to software define architecture, SDN can dynamically activate any idle computational entity and routes towards the entity whenever the current system model becomes unable to meet the service demand. Moreover, in an SDN-based system, an IoT-device is unaware about the computational entity where associate IoT-data is going to be processed. In consequence, the system becomes able to provide virtualization in processing IoT data and can be managed according to the dynamics of the environment. Hence, an SDN-based system supports scalability to a certain extent. Conversely, in the traditional static network architecture, IoT-data cannot be migrated to other computational entity as it does not provide any virtualized settings. As a result it becomes very difficult to achieve scalability in the traditional network.

Necessary notation for modelling the system has been provided in Table. 1

Table 1. Notations

Symbol	Definitio
E	Set of all computational entities.
α_e	Data processing capacity of computational entity e, $e \in E$
ϕ_e	Current data processing load on computational entity e, $e \in E$
P	Set of all communication paths.
P_e	Set of all communication paths to computational entity e, $e \in E; P_e \subset P$
β_p	Data transmission capacity of communication path p, $p \in P$
ω_p	Current data transmission load of communication path p, $p \in P$
λ_n	Data transmission rate of any IoT device n.
μ_n	Sensed data by any IoT device n.

4. SDN-based IoT-resource discovery

The proposed SDN-based IoT-resource discover policy executes in the *CC*. Whenever an IoT-device n sensed any data μ_n from the external environment, it forwards the data μ_n through SDN to *CC*. Besides, the contextual information of IoT-device n regarding its data transmission rate λ_n is also sent to *CC*. Based on the received information, *CC* runs the *DiscoverResources* procedure as shown in Algorithm. 1.

The *DiscoverResources* procedure is consist of four basic steps. The steps can be describes as follows:

1. At firs, for each of the computational entity (line 4), it is checked whether the inclusion of μ_n to its current data processing load exceeds the capacity of the corresponding computational entity (line 5). If it satisfies then the computational entity with minimum data processing load is selected as the target entity for processing μ_n (line 6-8). This approach can be termed as the best-fi selection of computational entity.

2. Later, from the available routing paths the suitable routing path towards the selected computational entity is identifie (line 10-13). In this case, the firs route is selected that cannot be congested due to per unit time data transmission from the IoT device n (line 11). This is considered as the firs-fi selection of the routing path.

3. In this step the sensed data μ_n of IoT-device n is forwarded towards the selected computation entity through a congestion-less routing path (line 14-16).

4. If no feasible computational entity or routing path is found, *CC* can dynamically initiate any idle

---

Algorithm 1 Resource discovery algorithm

1: **procedure** DiscoverResources(n, λ_n, μ_n)
2:     $\eta \leftarrow maxLoad$
3:     $s_e \leftarrow null$
4:     **for** $e := E$ **do**
5:         **if** $\phi_e + \mu_n < \alpha_e$ **then**
6:             **if** $\phi_e < \eta$ **then**
7:                 $\eta \leftarrow \phi_e$
8:                 $s_e \leftarrow e$
9:     $s_p \leftarrow null$
10:     **for** $p := P_{s_e}$ **do**
11:         **if** $\beta_p - \omega_p > \lambda_n$ **then**
12:             $s_p \leftarrow p$
13:             **break**
14:     **if** $s_e \neq null$ **then**
15:         **if** $s_p \neq null$ **then**
16:             Forward μ_n to s_e through s_p
17:     **if** $s_e = null \quad or \quad s_p = null$ **then**
18:         Activate computational entity $a_e; a_e \in E$
19:         Identify route a_p towards $a_e; a_p \in P$
20:         Forward μ_n to a_e through a_p

---

computational entity and identify route towards the entity so that the sensed data μ_n can be forwarded for processing.

The *DiscoverResources* procedure combines best-fi and firs-fi selection approach (step 1-2) within it. Generally, the complexity of this algorithm will increase linearly as the number of computational entity increases. However, due to step 1 and 2, it becomes easier to identify appropriate computational resources and associated routing path (step 3). Moreover, due to basic features of SDN, it is also possible to accommodate increasing service demand to idle computation entities (step 4). As a result, scalability issues in resource discovery for IoT become attainable.

Since the *DiscoverResources* procedure facilitates resource discovery and scalability, it play crucial role in minimizing resource discovery time and in enhancing QoS satisfaction for increasing number of IoT-service requests. Besides, not only in IoT, the proposed SDN-based approach can be extended to any sort of operations [15] where real-time interactions are involved.

5. Performance Evaluation

In order to claim the feasibility of the proposed SDN-based IoT-resource discover policy, at firs, the system has been simulated and later the experimental results are analysed.

5.1. Simulation Environment

The system model has been simulated using *iFogSim* [16] simulation toolkit. *iFogSim* simulation toolkit has been developed upon the *CloudSim* framework which has been used extensively to simulate Cloud, Mobile Cloud, Vehicular Cloud environment.

In the simulation, Fog nodes are considered as the computational entities and *CC* is a specialized Fog node to conduct basic operations on SDN. In the modelled simulation environment, IoT-devices can be placed at any location and the devices can ask for processing their sensed data by following poisson distribution.

As the compatible real-world workload is not currently available, in the simulation, synthetic workload has been used. The workload and modelled system can be easily re-constructible. Simulation parameters and units are represented in Table. 2.

Table 2. Simulation parameters

Parameter	Value
Simulation Duration	100 s
Processing capacity of Fog nodes	20 - 30 Mbps
Service request size	0.5 - 1 Mb
Link bandwidth capacity	7 - 10 Mbps
Transmission rate of IoT devices	2 - 3 Mbps
Service delivery deadline	1 - 2 s

5.2. Simulation Results

The required time for identifying suitable computational resources is considered as one of the performance metrics. In order to model resource discovery time Eq. 1 has been applied. Here, the summation of data propagation time (δ_t) from source IoT device to target Fog node and waiting time (ν_t) in Fog node has been identified as total resource discovery time (RD_t) for a data processing request.

$$RD_t = \delta_t + \nu_t \quad (1)$$

Fig. 2 depicts that, resource discovery time for IoT in static network is higher compared to SDN-based policy. Although in SDN-based approach, a certain amount of time is required by *CC* to identify appropriate target Fog node and the associate routing path, the policy helps to reduce data processing waiting time and data propagation time to a great extent. In fact, the SDN-based solution selects that Fog node and that routing path as processing and communication medium in which processing load and network congestion is comparatively less. That's why in SDN-based solution resource discovery time gets minimized. Conversely, in static network based approach neither data processing overhead of Fog nodes nor network congestion is taken in to account. As a result, a high amount of time is required for resource discovery.

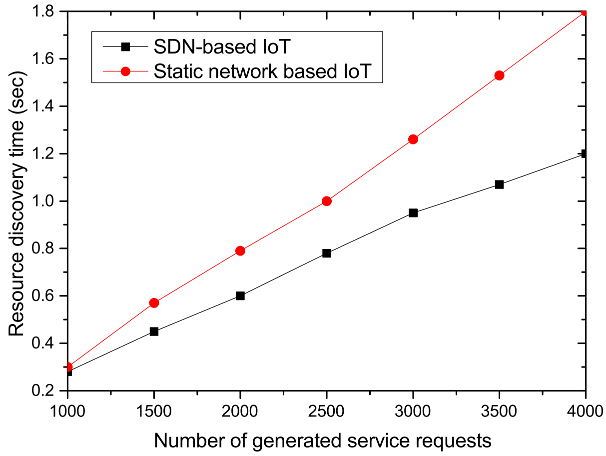


Figure 2. Resource discovery time vs number of service requests

In addition to resource discovery time, the percentage of QoS-satisfie data processing requests is considered as another performance metric. Here, the deadline satisfie service delivery is taken into account as a QoS parameter. A data processing request satisfie QoS when the following condition is satisfied here Δ_t is the service delivery deadline, τ_t is the service response time.

$$\Delta_t > \tau_t \quad (2)$$

Fig.3 represents that, the percentage of QoS satisfie service requests in static network decreases significatly as the number of service requests increases. In SDN-based IoT, as scalable resource discovery for increasing number of service requests is ensured, the percentage of QoS satisfie service requests always

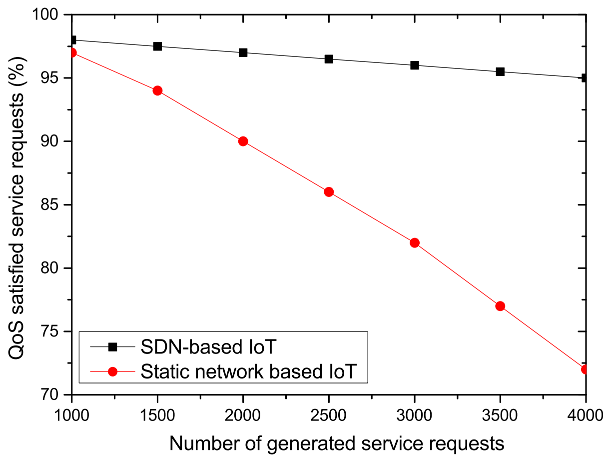


Figure 3. Percentage of QoS satisfied service requests vs number of service requests

remains in high. However, a very less amount of downfall in the percentage of QoS satisfie service request is also experienced in SDN-based IoT as the number of service request rises. It happens due to runtime activation of idle Fog nodes to meet the service demand. The required time for activating an idle Fog node has adverse effect of the QoS satisfie service delivery of some requests.

6. Conclusion

The domain of IoT is expanding at a great pace. It is also experiencing different type of challenges in its way of practical applicability . We have targeted one of such challenges of IoT in respect of scalable resource discovery. Here, the proposed SDN-based resource discover policy for IoT uses contextual information of computational entities, networks and IoT devices to identify suitable resources and routing path to process and forward IoT-data. The policy is independent of increasing number of data processing (service) requests that comes from geo-distributed IoT-devices. In consequence, the policy facilitates scalable resource discovery in IoT. Moreover, several simulation studies also claim the feasibility of the proposed policy in respect of resource discovery time and QoS satisfaction rate of service requests. The SDN-based solution is substantially efficient compared to the static network based resource discovery for IoT.

In future we aim at extending SDN-based solutions to other aspects of IoT such as SDN-based IoT network management, SDN-assisted content distribution in IoT, application deployment in SDN-enabled IoT.

References

[1] Qin, Y., Sheng, Q.Z., Falkner, N.J.G., Dustdar, S., Wang, H. and Vasilakos, A.V. (2014) When things matter: A data-centric view of the internet of things. *CoRR* **abs/1407.2704**. URL http://arxiv.org/abs/1407.2704.

[2] Liu, J., Li, Y., Chen, M., Dong, W. and Jin, D. (2015) Software-define internet of things for smart urban sensing. *IEEE Communications Magazine* **53**(9): 55–63. doi:10.1109/MCOM.2015.7263373.

[3] Kirkpatrick, K. (2013) Software-define networking. *Commun. ACM* **56**(9): 16–19.

[4] Govindarajan, K., Meng, K.C., Ong, H., Tat, W.M., Sivanand, S. and Leong, L.S. (2014) Realizing the quality of service (qos) in software-define networking (sdn) based cloud infrastructure. In *2014 2nd International Conference on Information and Communication Technology (ICoICT)*: 505–510.

[5] Zhang, Y., Shen, Y., Wang, H., Yong, J. and Jiang, X. (2016) On secure wireless communications for iot under eavesdropper collusion. *IEEE Transactions on Automation Science and Engineering* **13**(3): 1281–1293. doi:10.1109/TASE.2015.2497663.

[6] Yen, T.C. and Su, C.S. (2014) An sdn-based cloud computing architecture and its mathematical model. In *2014 International Conference on Information Science, Electronics and Electrical Engineering*, **3**: 1728–1731.

[7] Ku, I., Lu, Y. and Gerla, M. (2014) Software-define mobile cloud: Architecture, services and use cases. In *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*: 1–6.

[8] Gudipati, A., Perry, D., Li, L.E. and Katti, S. (2013) Softran: Software define radio access network. In *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, HotSDN '13 (New York, NY, USA: ACM): 25–30. doi:10.1145/2491185.2491207, URL http://doi.acm.org/10.1145/2491185.2491207.

[9] Yeganeh, S.H., Tootoonchian, A. and Ganjali, Y. (2013) On scalability of software-define networking. *IEEE Communications Magazine* **51**(2): 136–141.

[10] Jararweh, Y., Al-Ayyoub, M., Darabseh, A., Benkhelifa, E., Vouk, M. and Rindos, A. (2015) Sdiot: a software define based internet of things framework. *Journal of Ambient Intelligence and Humanized Computing* **6**(4): 453–461.

[11] Wu, D., Arkhipov, D.I., Asmare, E., Qin, Z. and McCann, J.A. (2015) Ubifow: Mobility management in urban-scale software define iot. In *2015 IEEE Conference on Computer Communications (INFOCOM)*: 208–216.

[12] Mahmud, M.R., Afrin, M., Razzaque, M.A., Hassan, M.M., Alelaiwi, A. and Alrubaian, M. (2016) Maximizing quality of experience through context-aware mobile application scheduling in cloudlet infrastructure. *Software: Practice and Experience* **46**(11): 1525–1545. doi:10.1002/spe.2392, URL http://dx.doi.org/10.1002/spe.2392. Spe.2392.

[13] d. Silva, M.P., NazÃąrio, D.C., Dantas, M.A.R., GonÃğalves, A.L., Pinto, A.R., Manerichi, G. and Vanelli, B. (2016) Context management and distribution architecture using software-define networking. In *2016 IEEE 25th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*: 185–190.

[14] Savarese, G., Vaser, M. and Ruggieri, M. (2013) A software define networking-based context-aware framework combining 4g cellular networks with m2m. In *2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC)*: 1–6.

[15] Afrin, M., Mahmud, M.R. and Razzaque, M.A. (2015) Real time detection of speed breakers and warning system for on-road drivers. In *2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*: 495–498. doi:10.1109/WIECON-ECE.2015.7443976.

[16] Gupta, H., Vahid Dastjerdi, A., Ghosh, S.K. and Buyya, R. 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* doi:10.1002/spe.2509. Spe.2509.