International Journal of Pure and Applied Mathematics
Volume 120 No. 5 2018, 855-868
ISSN: 1314-3395 (on-line version)
url: http://www.acadpubl.eu/hub/
Special Issue

http://www.acadpubl.eu/hub/

A Framework for Optimization of Location of Fog Servers and Fog Network Formation to Minimize Latency

Hussein Ali Ghadhban Alsalman and Jaber Ibrahim Naser
Msc(IS), Directorate Of Education in Basrah.
Msc(IS), Directorate Of Education in Al-Qadisiyah .
Hussein.alsalman85@gmail.com,aberalidami@gmail.com

Abstract: For computing is an extension to cloud computing which exploits computing resources of an enterprise's network. It is also named as fogging or edge computing. It is a phenomenon in which computing infrastructure is decentralized between the cloud and the edge of the cloud. This will facilitate many applications and services that do not really fit into public cloud for reasons like infrastructure limitations or technical. It enables provision of services immediately by passing the overhead of electronic super highway or Internet. Some of the applications of fog computing include healthcare, smart cities, smart grids and connected cars. Here location fog servers is an important as the data availability drives decision making in fog computing. Both end user devices and cloud servers involve in computing thus increasing speed and minimizing latency. In order to optimize performance of fog computing, it is important to know the location of fog nodes. To solve this problem, a framework is proposed and implemented that continuously considers the location of fog nodes and make well informed decisions in distributing workload in order to minimize latency and improve the performance of applications that run on top of fog computing resources. We built a prototype application to demonstrate proof of the concept. Our empirical results revealed that the proposed framework shows significant performance improvement over other state-of-the-art approaches found in the literature.

Keywords –Cloud computing, fog computing, optimization of fog computing, network formation

1. INTRODUCTOIN

Internet of Things (IoT) has emerged as combination of diversified technologies to enable connectivity between things and computing devices. Due to IoT realization billions of connected devices and things are operating in the real world. When IoT is integrated with IT infrastructure of any organization, it generates huge amount of data known as big data and that cannot be maintained in the local IT infrastructure. It needs support from cloud computing [1]. When large number of IoT devices depends on cloud computing infrastructure, the cloud may not be able to provide required latency. In order to reduce transmission latency, fog computing came into existence. With fog computing many network functions are moved to edge of network where fog devices are involved in processing instead of sending data directly to public cloud. Thus IoT applications gain advantage from fog computing in terms of reducing latency.

With fog computing in place, the computational tasks are allocated to fog nodes intelligently. With fog computing integrated with IoT, three layers are required. The bottom layer is known as edge layer where IoT devices operate and data is originated. The middle layer is known as fog layer

where fog nodes or fog servers are maintained. They are in the local area network with capabilities of data analytics. The top layer is known as cloud layer where huge amount of data is stored and processed. The fog layer is faster than the cloud layer. Therefore fog computing is given significance in the context of IoT applications. In other words response time of IoT applications is less when compared with that of directly used public cloud. This is the rationale behind the growth of fog computing.

In this context, it is important to locate fog servers in order to exploit its services to offload computation and reduce latency time further. Many researchers contributed towards this kind of study. They include [1], [5], [6], [7], [11], [12], and [16]. Particularly research on latency minimization is found in [1], [8], [15] and [22]. In many of the existing works, it is assumed that formation of for network is known to devices. However, it is not true in the real word as fog nodes may discover other fog nodes dynamically to form fog computing network. This is the reason that there is need for locating fog server nodes and estimating the possibility of reducing latency in order to form fog network that can minimize latency. The contributions of this paper are as follows.

1. We proposed a framework that is used to optimize location of fog servers and form a suitable fog network in order to minimize overall latency required by IoT applications.
2. We proposed an algorithm to achieve fog network formation that leads to minimal latency.
3. We built a prototype application to demonstrate proof of the concept. Our empirical results revealed that fog computing network that is formed based on the location of servers that are capable of reducing latency has its impact on the performance of IoT applications.

The remainder of the paper is structured as follows. Section 2 presents review of literature related to fog computing. Section 3 presents the need for locating fog servers and form a fog network. Section 4 presents the proposed framework used to minimize latency experienced by IoT applications. Section 5 presents experimental results while section 6 concludes the paper and provides directions for future work.

2.RELATED WORK

This section reviews literature on the fog network formation and its related aspects. Lee et al. [1] proposed a framework for discovering fog network information dynamically at runtime by using online secretary approach to achieve minimal latency. Dawy et al. [2] studied cellular communications that are in massive scale and useful for distributed networks. Mozaffari et al. [3] on the other hand explored the machine to machine (M2M) communications and the tradeoffs in performance. Park et al. [4] investigated the heterogeneity nature of Internet of Things (IoT) and its resources and communication techniques. There is relationship between fog computing and IoT. This relationship is explored by Chiang and Zhang [5] and provided insights related to various opportunities of IoT and its integration with fog computing. The way cloud is extended to the things in the network is studied by Cisco [6] to the extent of integration between fog computing and IoT.

Peng et al. [7] studied the radio access networks that are based on fog computing. ElBamby et al. [8] proposed a methodology for edge computing in proactive fashion in fog networks that are latency constrained. Fog network formation is dynamic in nature. Towards this end, Lee et al. [9] investigated online optimization techniques that help in bringing about effectiveness in fog computing under uncertain situations. Fog computing has many ingredients. They are explored by Yannizzi et al. [10] in terms of cloud computing and fog computing. IoT has its relationship with fog computing when latency improvement is concerned. This relationships studied by Bonomi et al. [11]. M2M fog platforms and M2M communications are investigated by Vallati et al. [12] in order to know LTE M2M communications in the context of fog computing. M2M communications are also explored by Khelil and Soldani [13] for ensuring road safety.

Luan et al. [14] proposed a framework that is used to have media rich content distribution network where vehicles are also involved. Nishio et al. [15] studied the need for SOA for sharing resources in heterogeneous environments. Sharma et al. [16] investigated the need for Service Oriented

Architecture (SOA) to realize optimization in latency of mobile clouds. Mobile fog servers are explored in the context of IoT for having self-aware communication architecture. When it comes to mobile edge computing with distributed architecture, resource allocation and task scheduling are explored by Zhao et al. [17]. The concept of cooperative resource management is investigated in mobile cloud computing (MCC) in [18]. In the same fashion profitability in task allocation is studied by Khaledi et al. [19]. Using mobile edge computing the process of offloading communications and computations is explored by Ketyk et al. [20]. It was studied in mobile edge computing with 5G technology.

Souza et al. [21] explored fog-cloud scenarios and investigate on handling service allocation. Cloud and edge processing with combination is studied in fog radio access networks by Park et al. [22]. In [23] mobile edge computing with the concept of CPU time allocation and joint sub carrier allocation are investigated. The tradeoffs in the delay and power consumption are studied by Deng et al. [24] in the context of cloud-fog computing. Similar kind of work is carried in multi-user environments in [25]. In the literature it is found that fog computing and IoT integration is very important to have better solutions in distributed computing. However, there is need for further investigation in the context of dynamic formation of fog computing. The assumption of identified fog nodes will not be useful in all scenarios. Towards this end, in this paper we proposed a framework that takes care of forming fog network dynamically.

3. PROBLEM DEFINITION

As fog network can reduce latency of IoT applications, it is important to have fog computing for offloading computations. However, sometimes fog network formation is dynamic without any pre-defined knowhow on the location of nodes. In such context, it is important to locate neighbor fog nodes and estimate the latency benefits they can give in order to form a network that really optimizes the latency experienced by IoT applications. Figure 1 shows the context of this problem where locating fog nodes that can save bandwidth and give real-time response to

IoTapplications.

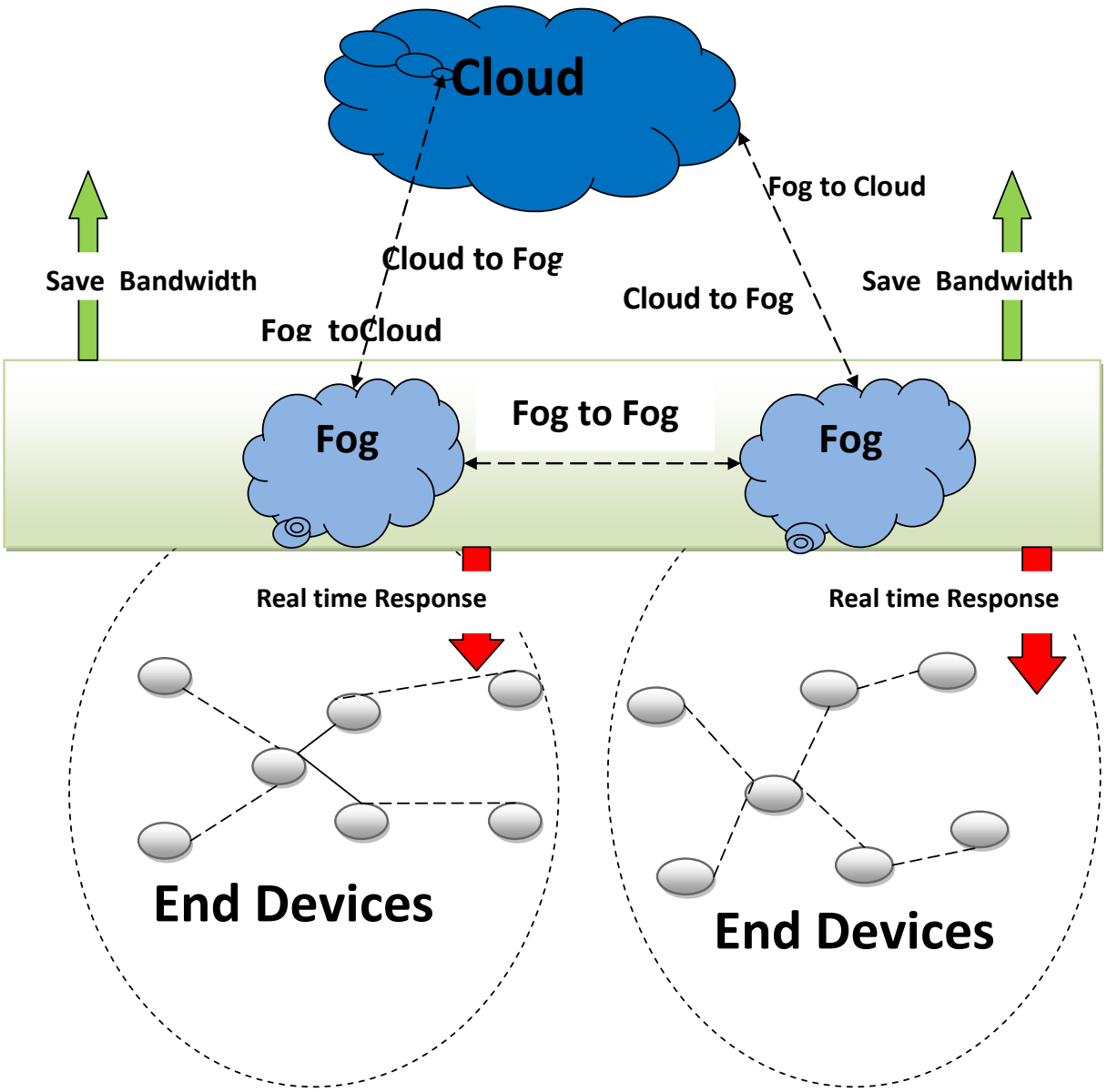

Figure 1**:** IoT with fog computing integration

As shown in Figure 1, the end devices, for nodes and cloud are involved in processing IoT applications. The problem is that cloud is slow with respect to serving IoT applications as the cloud was not optimized for IoT applications. The fog computing network is faster than that of cloud. In this context, further optimization is the problem considered in this paper. It makes use of the proposed framework to determine neighboring fog nodes to form a fog network dynamically while minimizing latency by careful selection of fog nodes after analyzing expected latency.

4. PROPOSED SYSTEM

The proposed system considers a system model containing sensor layer, fog layer and cloud layer as usual. The fog layer may be IoT devices or fog nodes that can perform storage and computing. Each fog node may maintain computation queue. The tasks are sent to fog node. Then the service rate is defined as in Eq. (1).

$$\mu_{mn} = \frac{B_l}{S} log_2\left(1 + \frac{g_{mn}\, p_{tx,m}}{B_l N_0}\right) \quad (1)$$

Where m and n are fog nodes, $d_{mn}$is the distance between them, g_{mn} is the channel gain between the fog nodes, $P_{tx,m}$ is the transmission power of fog node, B_l is the bandwidth per node. N_0 is the noise power spectral density

$$T_n(\lambda_{mn}(\beta_{mn}), \mu_{mn}) = \frac{\lambda_{mn}(\beta_{mn})}{2\mu_{mn}(\mu_{mn} - \lambda_{mn}(\beta_{mn}))} + \frac{1}{\mu_{mn}} \quad (2)$$

The latency of transmission queue is modeled as in Eq. (2) where task arrival is assumed as per the Poisson process. Then the transmission queue delay is computed as in Eq. (3).

$$T_c(\lambda_c(\beta_c), \mu_c) = \frac{\lambda_c(\beta_c)}{2\mu_c(\mu_c - \lambda_c(\beta_c))} + \frac{1}{\mu_c} \quad (3)$$

Afterwards the computation queue is modeled as in Eq. (4).

$$D_n\left(\lambda_{mn}(\beta_{mn})\right) = \frac{\lambda_{mn}}{2\mu_n(\mu_n - \lambda_{mn})} + \frac{1}{\mu_n} + \omega_n \lambda_{mn}(\beta_{mn}) \quad (4)$$

When the fog node computes assigned tasks locally, the latency is modeled as in Eq. (5).

$$D_m\left(\lambda_m(\beta_m)\right) = \frac{\lambda_m(\beta_m)}{2\mu_m(\mu_m - \lambda_m(\beta_m))} + \frac{1}{\mu_m} + \omega_m \lambda_m(\beta_m) \quad (5)$$

When tasks are computed at the cloud, the computing delay is computed as in Eq. (6).

$$D_c\left(\lambda_c(\beta_c)\right) = \omega_c \lambda_c(\beta_c) \quad (6)$$

When a task is routed to public cloud denoted as c, the latency is computed as in Eq. (7).

$$L_c(\lambda_c(\beta_c), \mu_c) = T_c(\lambda_c(\beta_c), \mu_c) + D_c\left(\lambda_c(\beta_c)\right) \quad (7)$$

When a task is offloaded to fog node, the latency is computed as in Eq. (8).

$$L_n(\lambda_{mn}(\beta_{mn}), \mu_{mn}) = T_n(\lambda_{mn}(\beta_{mn}), \mu_{mn}) + D_n\left(\lambda_{mn}(\beta_{mn})\right) \quad (8)$$

When the fog node computes it's given tasks locally, the latency is computed as in Eq. (9).

$$L_m\left(\lambda_m(\beta_m)\right) = D_m \lambda_m(\beta_m) \quad (9)$$

With dynamic fog network formation, the new task arrived at fog node; the maximum latency of fog node is computed as in Eq. (10)

$$D_m\left(\lambda_m(\beta_m)\right) = \frac{\lambda_m(\beta_m)}{2\mu_m(\mu_m - \lambda_m(\beta_m))} + \frac{1}{\mu_m} + \omega_m \lambda_m(\beta_m) \quad (10)$$

The framework used to form a fog network dynamically is as in Figure 1.

Notation	Description
m,n	Fog nodes
g_{mn}	channel gain between the fog
$P_{tx,m}$	transmission power of fog node
B_l	bandwidth per node
N_0	Noise power spectral density.
$\lambda_{mn}(\beta_{mn})$	Taskarrival rate offloaded from fog node m to fog node n
T	Transmission queue
μ_{mn}	Fog transmission service rate from m to n
S	Size of a task packet
C	cloud
$T_n(\lambda_{mn}(\beta_{mn}), \mu_{mn})$	Transmission queue from fog node m to n
$T_c(\lambda_c(\beta_c), \mu_c)$	Transmission queue when tasks are offloaded to cloud
μ_c	Cloud transmission service rate
μ_n	service rate of fog node n
$\omega_m \lambda_m(\beta_m)$	the fog node m's computing time
$L_c(\lambda_c(\beta_c), \mu_c)$	Latency of cloud
$L_n(\lambda_{mn}(\beta_{mn}), \mu_{mn})$	Latency from fog node m to n
$L_m\left(\lambda_m(\beta_m)\right)$	Latency of fog node m
$D_c\left(\lambda_c(\beta_c)\right)$	Delay of cloud
$D\left(\lambda_m(\beta_m)\right)$	Delay of fog node m
$D_n\left(\lambda_{mn}(\beta_{mn})\right)$	Delay of node m to n

Table 1: Notations used in the paper

Table 1 shows notations used in the paper and their description. The dynamic fog network formation is presented in Figure 2.

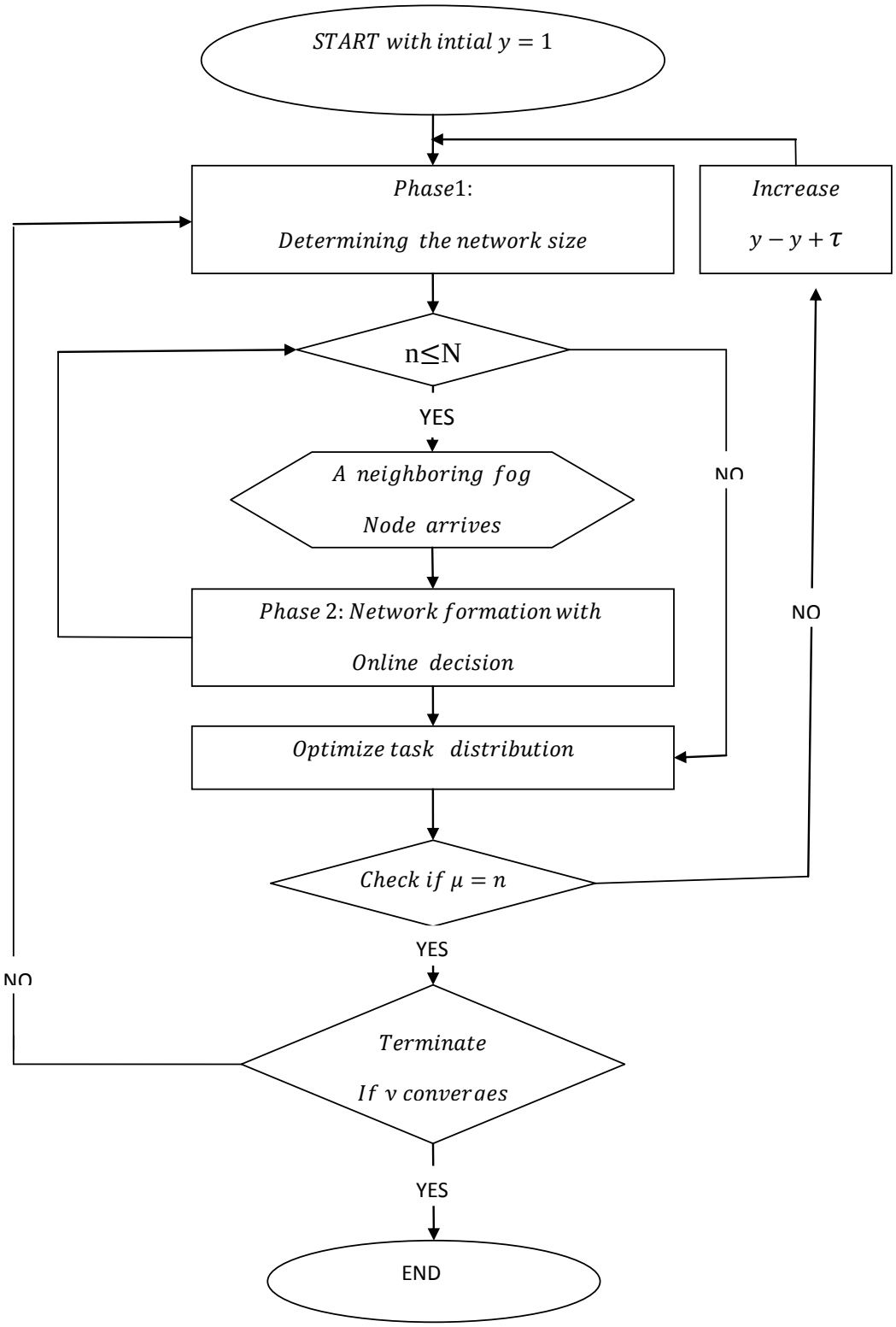

Figure 2**:** Fog network formation dynamically

As shown in Figure 2, it is evident that the flow of the proposed framework ensures the formation of fog server network dynamically. The fog node location finding and estimation of latency in order to select the node dynamically to form a network is the essence of the framework. Network size

determination and network formation with decision taking dynamically to minimize latency are the two important phases in the proposed framework.

Dynamic Fog Network Formation Algorithm

This algorithm is meant for forming fog network dynamically in order to minimize latency of IoT applications deployed in cloud-for network.

Algorithm: Dynamic Fog Network Formation Algorithm

1. Initiate γ =1

   Phase 1: Determining the network size :-
2. If(n≤N)
3. A neighboring fog Node arrives
4. Else
5. Optimize task distribution
6. If($|J| = \hat{J}$)
7. If(v conveges)
8. END
9. Else
10. repeat from step2
11. Else
12. increase $\gamma \leftarrow \gamma + \tau$
13. Repeat step1

    Phase2:-Network formation with online decision
14. If(n≤N)
15. A neighboring fog Node arrives
16. Else
17. Optimize task distribution
18. If($|J| = \hat{J}$)
19. If(v conveges)
20. END
21. Else
22. repeat from step2
23. Else
24. increase $\gamma \leftarrow \gamma + \tau$
25. Repeat from step1

The algorithm performs the process of fog network formation in two phases. In the first phase, it determines the network size with which it can serve with low latency and in the second phase, it makes decisions online to form a fog network. The network is formed in such a way that the service will be better with decreased latency. The experimental results are presented in Section 5.

5. EXPERIMENTAL RESULTS

This section presents results of experiments in terms of Latency for different task arrival rates at the initial fog node m and with the proposed framework.

Task Arrival Rate (Packet/second)	**Latency (seconds)**					
	Baseline, d_c= 140	**Baseline,d_c= 120**	**Baseline, d_c= 100**	**Proposed, d_c= 140**	**Proposed, d_c= 120**	**Proposed, d_c= 100**

100	180	190	185	155	150	145
150	210	200	200	170	165	160
200	230	220	210	190	185	180
250	250	245	240	210	200	195

Table 1: Latency for different task arrival rates at the initial fog node m

As shown in Table 1, task arrival rate and the latency provided by different algorithms with d_c 140, 120 and 100 are presented.

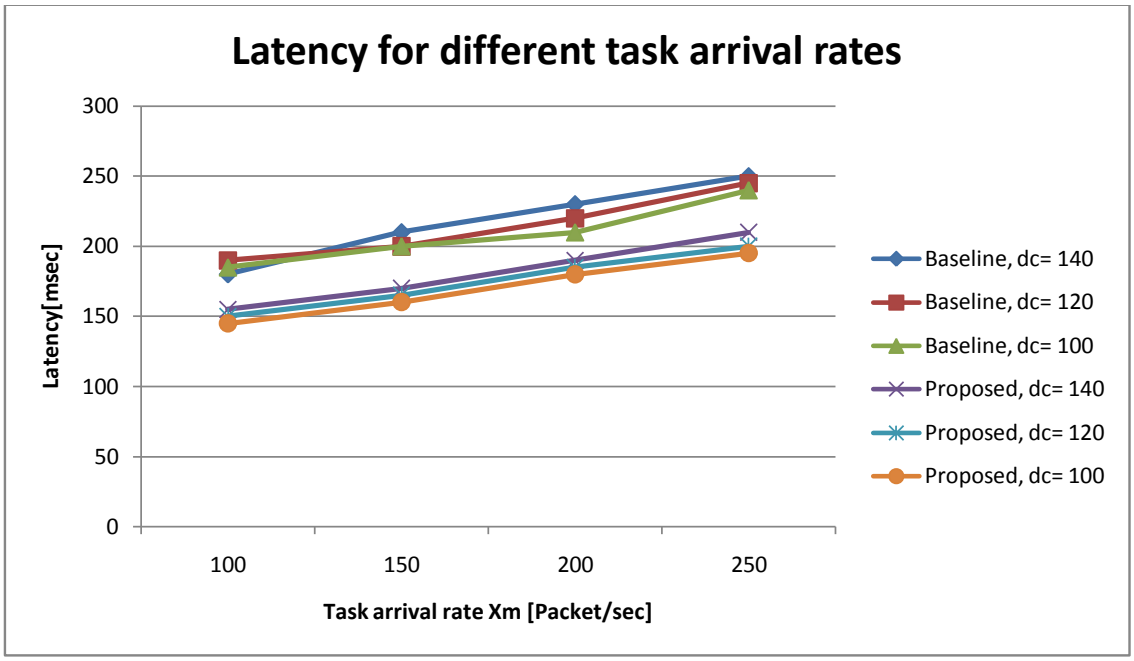

Figure 3: Latency for different task arrival rates at the initial fog node m

As shown in Figure 3, the task arrival rate and latency are presented in horizontal and vertical axes respectively. As the task arrival rate is increased, the latency is also increased with linear relationship. The proposed method with dc values 100 and 200 outperformed other methods.

Number of Neighbor Nodes	Latency (seconds)			
	x_m=10,ω_m = ω_n=0.05	x_m=13,ω_m = ω_n=0.05	x_m=10,ω_m = ω_n=0.03	x_m=13,ω_m = ω_n =0.03
100	155	175	130	145
140	145	165	125	138
180	142	162	115	132

Table 2: Latency for different number of neighboring nodes

As shown in Table 2, number of neighbor nodes and the latency provided by different algorithms with different values of x_m, $\omega_m$and $\omega_n$are presented.

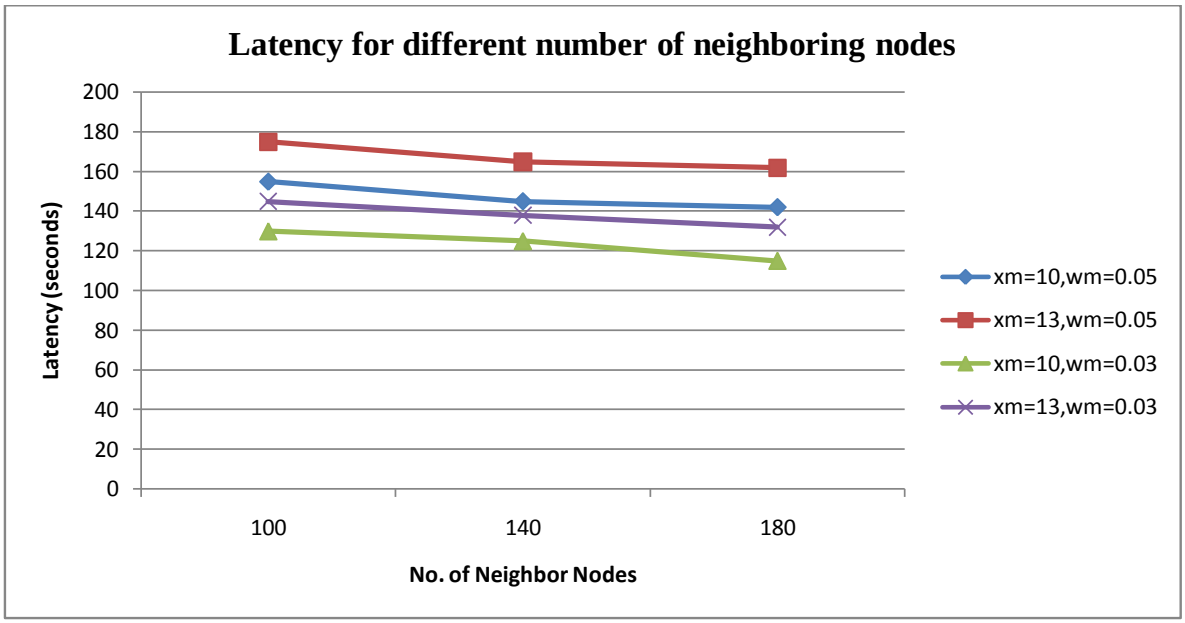

Figure 4: Latency for different number of neighboring nodes

As shown in Figure 4, the number of neighboring nodes and latency are presented in horizontal and vertical axes respectively. As the number of neighbor nodes is increased, the latency is also decreased with linear relationship. With different xm and wm values, the observations are made.

Target competitive ratio y	d_c**=120m**	d_c**=100m**
0	0	0
0.2	1.12	1.16
0.4	1.11	1.18
0.6	1.1	1.19
0.8	1.08	1.2
1	1.09	1.21
1.2	1.1	1.22
1.4	1.12	1.24
1.6	1.13	1.25

Table 3:Changes in the target competitive ratio

As shown in Table 3, number of iterations and the target competitive ratio are provided with different values of d_c.

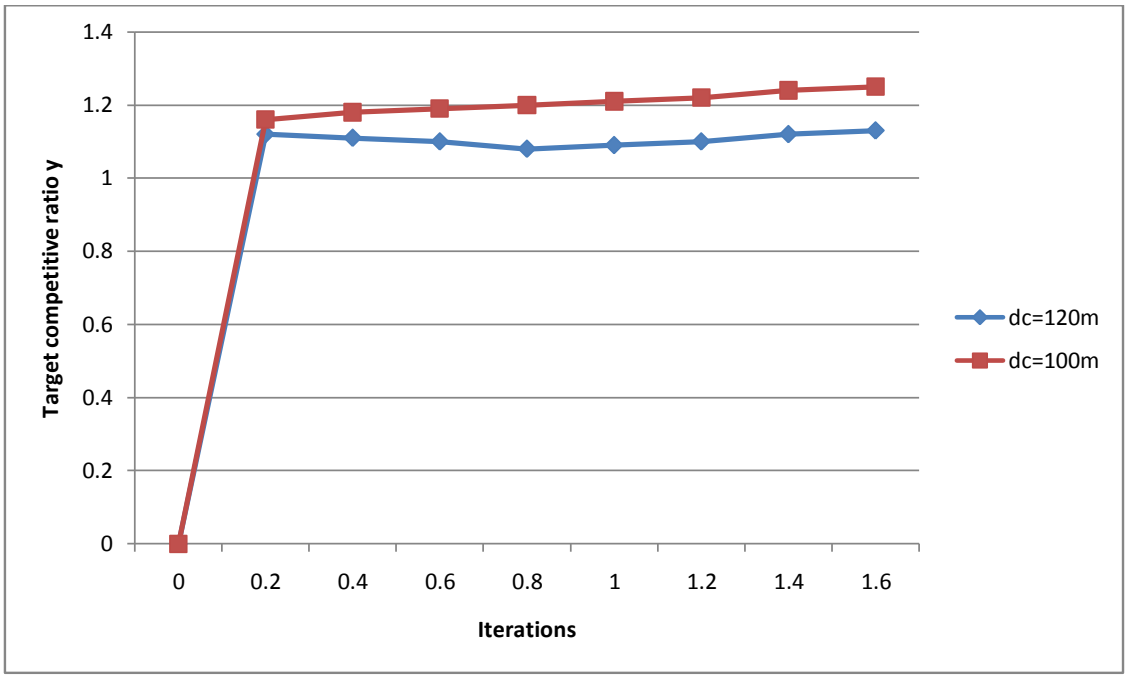

Figure 5: Changes in the target competitive ratio

As shown in Figure 5, the number of iterations and target competitive ratio are presented in horizontal and vertical axes respectively. As the number of iterations is increased, the latency is also increased. Observations are made with different d_c values.

Number of Neighbor Nodes	Percentage of offloaded tasks[%]					
	Neighboring nodes(Equal bandwidth)	Neighboring nodes(Cloud-centric)	Cloud(Equal bandwidth)	Cloud(Cloud-centric)	Fog node i(Equal bandwidth)	Fog node i(Cloud-centric)
0	65	52	20	30	19	18
20	70	59	10	25	20	18
40	73	62	8	20	20	18
60		60		22		19
80		52		28		20
100		40		40		22

Table 4:Fog transmission service rate with respect to the number of neighboring nodes

As shown in Table 4, number of neighboring nodes and percentage of offloaded tasks are provided with different kinds of nodes.

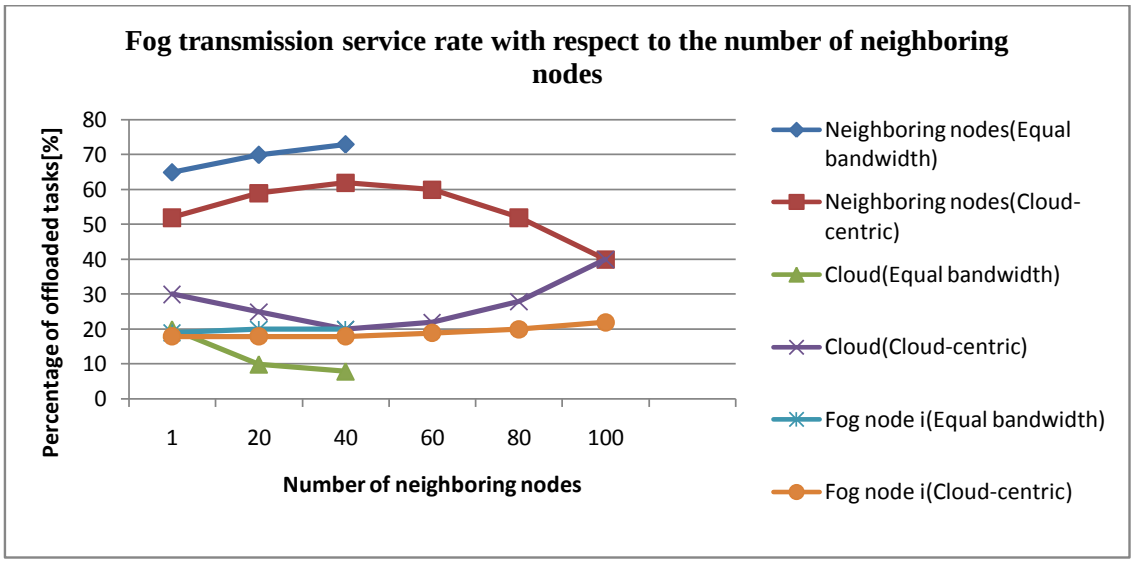

Figure 6: Fog transmission service rate with respect to the number of neighboring nodes

As shown in Figure 6, the number of neighboring nodes and % of offloaded tasks are presented in horizontal and vertical axes respectively. The dependency between number of neighboring nodes and the percentage of offloaded tasks is very dynamic based on the kind of node to which the task is offloaded.

Latency[msec]	**Baseline (d_c=100m)**	**Existing (d_c=100m)**	**Proposed(d_c =100m)**	**Baseline(d_c =120m)**	**Existing(d_c =120m)**	**Proposed(d_c=120)**
140	180	143	141	183	149	147
160	180	152	150	183	157	155
180	180	158	156	183	163	161
200	180	163	161	183	168	166

Table 5: Task distribution with respect to the number of neighboring nodes

As shown in Table 5, target competitive ratio and latency are provided for different algorithms with different values for d_c.

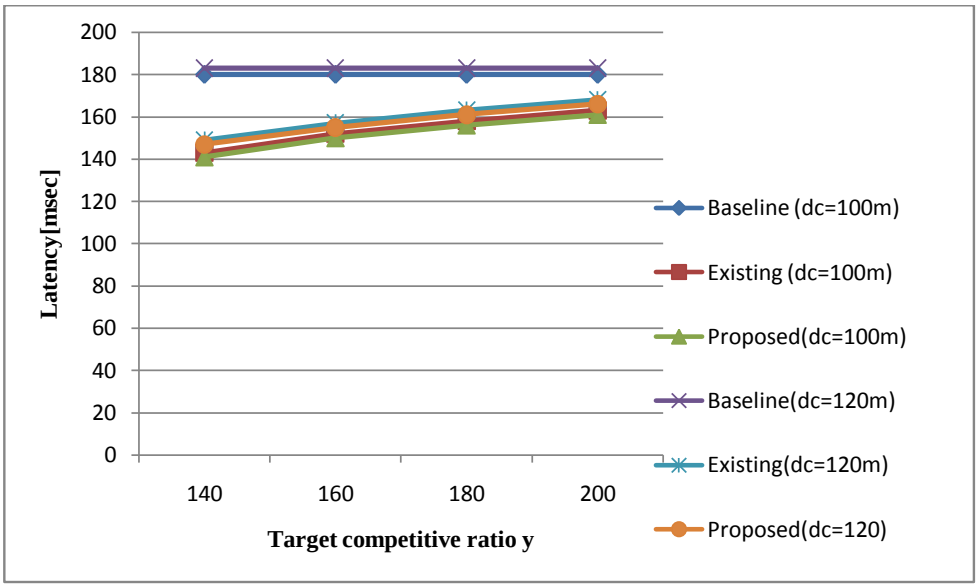

Figure 7: Latency against target competitive ratio

As shown in Figure 6, the target competitive ratio and latency are presented in horizontal and vertical axes respectively. The effect of target competitive ratio is observed with different algorithms and different d^c values. The proposed algorithm showed less latency comparatively consistently with different d_c values.

CONCLUSIONS AND FUTURE WORK

In this paper, we studied the need for fog computing to solve latency problems in IoT applications that delay on public cloud. When fog computing is used the computational work is offloaded in order to speed up processing and minimize latency. From the literature it is understood that most of the fog computing researches assumed the knowledge of fog nodes when network is formed. However, in the real world, it is possible that a fog node may need to form a network with other fog nodes dynamically. We proposed a framework that determines network size and makes decisions to form a fog network dynamically by considering latency estimation. Experiments are made with a prototype application suing baseline and proposed approach. The results revealed that the latency is minimized when the proposed framework is employed. In future we intend to improve the latency performance further by considering new algorithms in node selection process.

References

[1] G. Lee, W. Saad, and M. Bennis, "An online secretary framework for fog network formation with minimal latency," in Proc. IEEE Int. Conf. on Commun. (ICC), Paris, France, May 2017, pp. 1–6.
[2] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, "Toward massive machine type cellular communications," IEEE Wireless Communications, vol. 24, no. 1, pp. 120–128, Feb. 2017.
[3] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," IEEE Trans. Wireless Commun., vol. 15, no. 6, pp. 3949–3963, Jun. 2016.
[4] T. Park, N. Abuzainab, and W. Saad, "Learning how to communicate in the Internet of Things: Finite resources and heterogeneity," IEEE Access, vol. 4, pp. 7063–7073, Nov. 2016.
[5] 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, Dec. 2016.
[6] Cisco, "Fog computing and the Internet of Things: Extend the cloud to where the things are," Cisco white paper, 2015.
[7] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: issues and challenges," IEEE Network, vol. 30, no. 4, pp. 46–53, Jul. 2016.
[8] M. S. ElBamby, M. Bennis, and W. Saad, "Proactive edge computing in latency-constrained fog networks," in Proc. European Conference on Networks and Communications (EuCnC), Oulu, Finland, May 2017, pp. 1–6.
[9] G. Lee, W. Saad, and M. Bennis, "Online optimization techniques for effective fog computing under uncertainty," MMTC Communications-Frontiers, vol. 12, no. 4, pp. 19–23, Jul. 2017.
[10] M. Yannuzzi, R. Milito, R. Serral-Graci, D. Montero, and M. Nemirovsky, "Key ingredients in an iot recipe: Fog computing, cloud computing, and more fog computing," in Proc. IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Athens, Greece, Dec 2014, pp. 325–329.
[11] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in Proc. 1st MCC workshop on Mobile cloud computing. Helsinki, Finland: ACM, Aug. 2012, pp. 13–16.
[12] C. Vallati, A. Virdis, E. Mingozzi, and G. Stea, "Exploiting LTE D2D communications in M2M fog platforms: Deployment and practical issues," in Proc. IEEE 2nd World Forum on IoT, Milan, Italy, Dec. 2015, pp. 585–590.
[13] A. Khelil and D. Soldani, "On the suitability of device-to-device communications for road traffic safety," in Proc. IEEE World Forum on Internet of Things (WF-IoT), Seoul, Korea, Mar. 2014, pp. 224–229.
[14] T. H. Luan, L. X. Cai, J. Chen, X. Shen, and F. Bai, "Vtube: Towards the media rich city life with autonomous vehicular content distribution," in Proc. 8th Annual IEEE Commun. Society Conference on Sensor, Mesh and Ad Hoc Commun.and Networks, Salt Lake City, UT, USA, Jun. 2011, pp. 359–367. 30
[15] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, "Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud," in Proc. 1st Int. Wksh. on Mobile Cloud Comput.Netw., Bangalore, India, Jul. 2013, pp. 19–26.
[16] V. Sharma, J. D. Lim, J. N. Kim, and I. You, "SACA: Self-aware communication architecture for IoT using mobile fog servers," Mobile Information Systems, vol. 2017, pp. 1–17, Apr. 2017.
[17] T. Zhao, S. Zhou, X. Guo, and Z. Niu, "Tasks scheduling and resource allocation in heterogeneous cloud for delay-bounded mobile edge computing," in Proc. IEEE Int. Conf. on Commun. (ICC), Paris, France, May 2017, pp. 1–7.
[18] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative resource management in mobile cloud computing," IEEE J. Sel. Areas in Commun., vol. 31, no. 12, pp. 2685–2700, Dec. 2013.
[19] M. Khaledi, M. Khaledi, and S. K. Kasera, "Profitable task allocation in mobile cloud computing," in Proc. 12th Int. Symp.onQoS and Security for Wireless and Mobile Networks, Malta, Nov. 2016.

[20] I. Ketyk, L. Kecsks, C. Nemes, and L. Farkas, “Multi-user computation offloading as multiple knapsack problem for 5G mobile edge computing,” in Proc. European Conference on Networks and Communications (EuCnC), Athens, Greece, Jun. 2016, pp. 225–229.
[21] V. B. C. Souza, W. Ramirez, X. Masip-Bruin, E. Marn-Tordera, G. Ren, and G. Tashakor, “Handling service allocation in combined fog-cloud scenarios,” in Proc. IEEE Int. Conf. on Commun. (ICC), Kuala Lumpur, Malaysia, May 2016, pp. 1–5.
[22] S.-H. Park, O. Simeone, and S. Shamai, “Joint cloud and edge processing for latency minimization in fog radio access networks,” in Proc. IEEE 17th Int. Workshop on Signal Process. Adv. in Wireless Commun., Edinburgh, UK, Jul. 2016, pp. 1–5.
[23] Y. Yu, J. Zhang, and K. B. Letaief, “Joint subcarrier and CPU time allocation for mobile edge computing,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Washington DC, USA, Dec. 2016.
[24] R. Deng, R. Lu, C. Lai, and T. H. Luan, “Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing,” in Proc. IEEE Int. Conf. on Commun. (ICC), London, UK, Jun. 2015, pp. 3909–3914.
[25] Y. Mao, J. Zhang, S. Song, and K. B. Letaief, “Power-delay tradeoff in multi-user mobile-edge computing systems,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Washington DC, USA, Dec. 2016.