Analysis of Fog Model Considering Computing and Communication Latency in 5G Cellular Networks

Krittin Intharawijitr*, Katsuyoshi Iida†, and Hiroyuki Koga‡
*Graduate School of Science and Engineering, Tokyo Institute of Technology, Tokyo, Japan
Email: intharawijitr@netsys.ce.titech.ac.jp
†Global Scientific Information and Computing Center, Tokyo Institute of Technology, Tokyo, Japan
Email: iida@gsic.titech.ac.jp
‡ Graduate School of Environmental Engineering, University of Kitakyushu, Fukuoka, Japan
Email: h.koga@kitakyu-u.ac.jp

***Abstract*—Challenging 5G requirements have been proposed to enable the provision of more satisfactory services to users. One of the most critical 5G enhancements is extremely low latency compared with that of 4G. Various 5G applications will be supported by Cloud computing services. However, Cloud services sometimes have a long communication distance because of the physical location of data centers. *Fog computing architecture* is being introduced in 5G cellular networks to shorten the latency arising from the location of data centers. The communication delay as well as the computing delay must be considered for latency-sensitive applications. A Fog server serving many users will probably have a longer computing delay than a Fog server with a lighter workload. To clarify the computing delay and communication delay in Fog architecture, we define a mathematical model of a Fog network and the important related parameters. We also analyze results from a model used to evaluate three different policies for selecting the target Fog server for each task. We conducted simulations to evaluate this model.**

I. Introduction

Mobile communication has become an important part of daily life for many people, and millions of users now connect with each other via mobile network systems. Mobile networks have been continuously improved over the years to meet newly developing demands, and fifth-generation (5G) mobile networks are expected to provide a better *Quality of Service* (QoS) than what has been previously available, even though the 5G requirements will challenge network developers [1] [2].

The very low latency boundary of 5G networks will enable unprecedented mobile services. For example, *Tactile Internet*, an application allowing the instantaneous interaction of objects in both real and virtual environments [3], will enhance user activities in many fields such as healthcare, traffic control, industry, and education. One specific application is a form of augmented reality (AR) where an overlay of a virtual environment is displayed on a real environment. However, this requires the completion of heavy computing tasks within a short time, and AR applications typically sends such tasks from mobile devices to Cloud-base servers[4].

Unfortunately, if the physical location of the Cloud data center is far away, there will be a long communication delay. The *Fog computing service* has been introduced to provide a low latency network architecture for mobile applications [5]. Figure 1 shows the Fog network architecture. The Fog computing model replaces edge routers, where a router first attaches to users, with Fog servers that are each called a Fog node [6]. Mobile users send their packets to Fog servers, but the packets do not pass through the core network as for a Cloud service. The Fog physical location will be significantly closer to users, so it can provide a quick response to the source and thus overcomes the latency issue of Cloud computing service in 5G.

Survey results [7] have indicated that users of Fog computing face various issues that need to be resolved. For example, the capacity and latency, in terms of QoS, are important metrics with regard to latency-sensitive applications such as AR and high-quality real-time video. In this research, we have studied how the computing time and communication time in the Fog computing architecture changes according to the demand from mobile users and the supply of Fog servers in the network system. A heavy workload in one Fog node leads to a longer processing time. Even though resources are probably available in neighboring nodes, using these resources could lead to a much longer transmission delay despite the shorter computation time. Fog nodes can be predetermined as candidates to deliver better QoS. If we set an optimal target for the completion of the resultant workloads, the whole system can achieve maximum efficiency.

II. Mathematical Model

Here, we define the problem model regarding the Fog architecture. All entities and their relationship are considered. To find the optimal solution of the defined problem, we construct a mathematical model and constraints for the problem.

A. Problem definition

To design a low latency network architecture with acceptable performance, we defined a problem model and all important elements. We consider a group of users in the same area as one *Source node* and a Fog server as one *Fog node*. The relation of Fog nodes and Source nodes can be shown as a complete bipartite graph (Fig. 2). A Source node S_i connects to a Fog node F_j where $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, M\}$. Each Source node S_i produces a workload w_k with λ_i rate of Poisson process. Each Fog node F_j holds n_j

978-1-5090-1941-0/16/$31.00 ©2016 IEEE

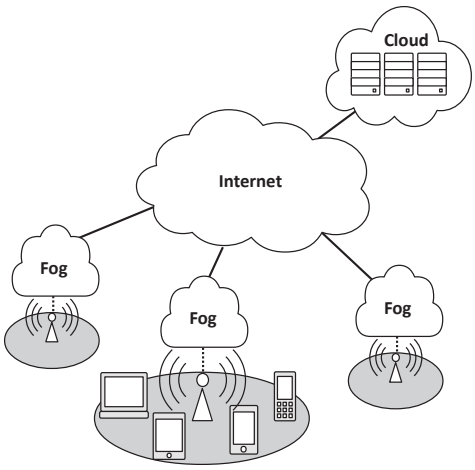


Fig. 1. Fog Architecture.

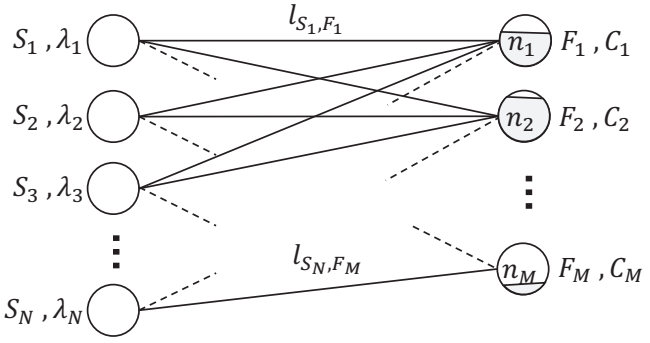


Fig. 2. Problem model

workloads at a time. A nodes maximum acceptable workload is C_j. The workload w_k will be sent to F_j. When it arrives at F_j, it requires a computing time that depends on n_j. The transmission latency between S_i and F_j is defined as l_{S_i,F_j}. This latency represents the round-trip time excluding the computing time in a Fog node. We assume that all Source nodes can access all Fog nodes in a network.

B. Latency functions

Taking the total latency into consideration, the following can be stated:

$$f_c(n) = \alpha(n) + \beta \tag{1}$$

$$f(w_k) = \begin{cases} l_{S_i,F_j} + f_c(n_j + w_k), & \text{accept } w_k \quad (2a) \\ 0, & \text{reject } w_k \quad (2b) \end{cases}$$

The function (1) is to estimate the processing time of all current workloads n_j held in the Fog node F_j. In fact, f_c is not easily defined because it is influenced by many factors. However, f_c is deliberately described here as a linear function to simplify the analysis of the beginning step. In addition, we assume that all Fog nodes use an identical function to determine the computing time.

The total latency can be determined using function (2). In the case of (2a), w_k can be accepted by F_j. When w_k initiated from the Source node S_i is offered to some Fog nodes, the total latency will become equal to the sum of the computing latency and the communication latency. On the other hand, if w_k is not executed by any Fog node, it will be discarded without creating any latency (2b).

C. Mathematical formulation

The objective function of the problem used to evaluate each policy is

$$\min_{w} \quad \frac{\sum_{r=1}^{R} w_r}{\sum_{k=1}^{K} w_k} \tag{3}$$

$$\text{subject to} \quad f(w_a) \leq \text{threshold} \tag{4}$$

$$0 \leq n_j \leq C_j$$

$$C_j,\ l_{S_i,F_j},\ w_k > 0$$

This objective function (3) tries to find the minimum of the *blocking probability* represented as the ratio between a number of rejected workloads w_r and the total number of all workloads w_k in the entire system. Before meeting the objective, the network system is required to first satisfy all constraints. Here, (4) requires that the total latency of an accepted workload w_a, composed of the computing delay and the communication delay, must not surpass the accepted threshold which is the maximum tolerant latency of a service.

D. Policies

All Fog nodes which can satisfy the system constraints given in the previous subsection II-C will be included in a candidate list. The workload w_k will select one target Fog node from among those candidates. In this research, we have considered three policies.

- *Random policy* (Random): The simplest way to pick the target node from a candidate list. One Fog node is randomly selected from a uniform distribution to execute a workload.
- *Lowest latency policy* (Lowest latency): The Fog node providing the lowest total latency given the current state of the system is selected.
- *Maximum available capacity policy* (Max capacity): The Fog node that has the maximum remaining resource in a candidate list is selected.

All of these policies can be used to find the most suitable Fog node for one workload, but a particular policy might not be the best solution for the whole system. We need to consider the results from all policies to determine which one is most appropriate under prevailing system conditions.

III. Simulation

After the problem and mathematical model are defined, simulation evaluations of the problem are needed. For this research, we developed a simulation using C++ programming language to demonstrate how the problem behavior varied according to different parameter values. The parameters were

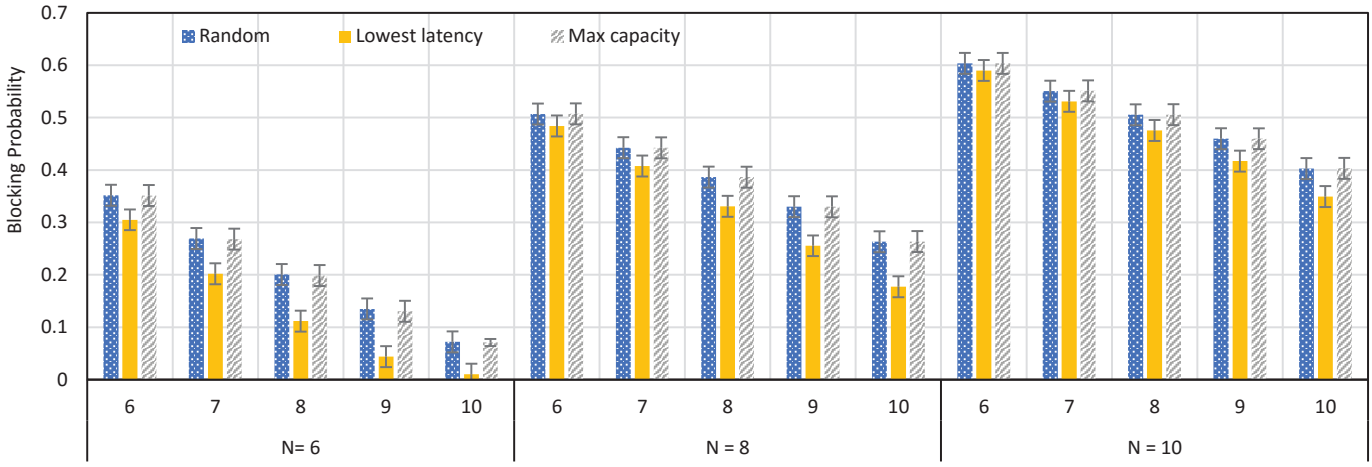
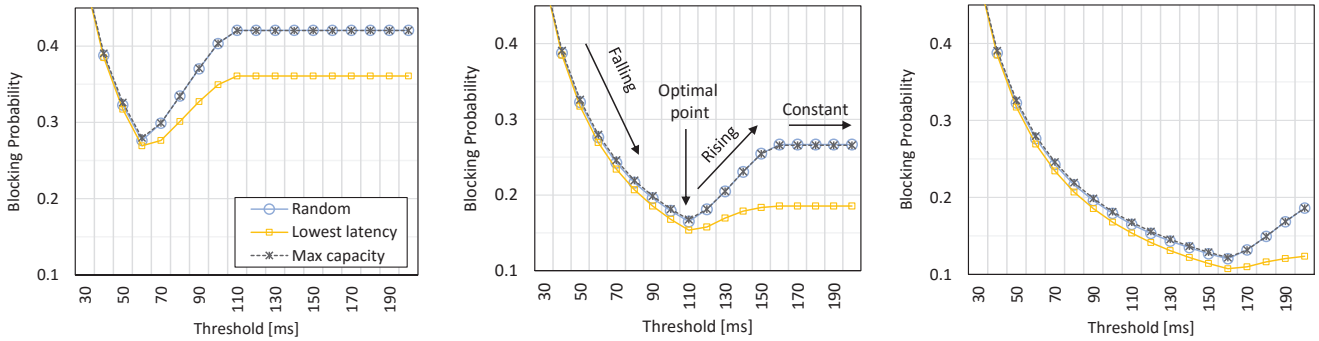


Fig. 3. Impact of number of Fog nodes (M), $C = 5$, threshold $= 100$



(a) $C = 5$ (b) $C = 10$ (c) $C = 15$

Fig. 4. Impact of threshold, $N = M = 10$

not altered during a simulation being run. Eventually, the system will reach a steady state; the result remains unchanged even if the simulation time continues. The final outcome of the simulation therefore shows the blocking probability of each policy along with its confidence interval.

Each source node generates a workload with a constant rate of a Poisson process. It produces the next workload at random times with an exponential distribution. When one workload has been launched, the system determines which Fog nodes can be included in the candidate list regarding constraints and each nodes current status. After that, the workload will select the destination depending on the predefined policy. A Fog node processes the workload with a computing time and then it releases the workload and sets the resource free.

The computing time in the simulation is described as the computing function (1). The communication time is considered to be the hop count. A source node and a Fog node with the same ID have a 1-hop distance. The hop distance will expand depending on the difference between the node IDs. For example, F_i is located 1 hop away from S_i and 2 hops away from $S_(i+1)$, and $S_(i-1)$.

In this research, we set the same λ, C, α, and β for every node. We, even more, suppose that all links have the same hop delay l_h. The simulation parameters are listed in Table I. .

TABLE I
SIMULATION PARAMETERS

Parameter	Description	Value
N	Number of Source nodes	6, 8, 10
M	Number of Fog nodes	6 – 10
α	Gradient of Computing function	10
β	Constant of Computing function	0
l_h	Hop delay	10 [ms/hop]
C	Maximum capacity of one Fog node	5, 10, 15 [workloads]
λ	Producing work rate	100 [workloads/s]
threshold	An upper bound of latency	30 – 200 [ms]

IV. Analysis of Results

The brief numerical results from the simulation provide many interesting points to be discussed and analyzed. Y axis of all results represents the blocking probability

A. Number of Nodes

Figure 3 shows results for the three policies with different numbers of nodes. The *lowest latency policy* always provides the least blocking probability when there are fewer source nodes than Fog nodes. When $N = 6$ and $M = 10$, the confidence interval for the lowest latency policy shows that this policy completely outperforms the other policies. In this case, the system has sufficient supply and very little demand from users. By using the lowest latency policy, a Fog node can finish a workload in a shorter time and release resource sooner than with the other policies. The earlier availability of a free slot allows a Fog node to accept more workloads while other policies would waste time waiting for an available buffer.

Comparing the *random policy* and the *maximum available capacity policy* further, both policies provide a nearly equal blocking probability in most cases, even though the number of nodes is changed. These policies can distribute a workload to any of the candidate Fog nodes. The random policy balances the amount of workload through randomness with equal probability. The maximum available capacity policy dispatches a workload to the node with the most remaining resource. In the steady state, the output from both policies will always stay in a closely similar state.

In addition, increasing the number of source nodes means that more demand is inserted into the network. A Fog node must be able to receive more tasks with the same storage or the number of rejected workloads must rise. Incrementally increasing the number of Fog nodes gives the network system more capability to support the demand from source nodes. As a result, the frequency of rejected workloads is reasonably reduced.

B. Threshold

In Fig. 4, the blocking probability decreases as we increase the threshold. Soon after reaching the optimal point, the probability surprisingly rises. The probability with the lowest latency policy rises more gradually than those with the other policies which dramatically increase. After the probability rises, though, it reaches a constant phase and remains at the same value even when the threshold is increased.

There are three phases as labels in Fig. 4(b), The falling phase occurs when the threshold prevents a Fog node attracting enough workload to operate at its maximum capacity. Expanding the threshold in this phase helps a Fog node stockpile a greater workload, so the number of rejections will go down. During the rising phase, within the threshold, the workload can fully occupy all resources of a Fog node. When the system is in the steady state, the computing time is usually equal to the time needed to compute C workloads. Thus, the higher threshold allows a workload from a more distant source node to be attracted and leads to a larger volume of work flowing to the Fog node. In other words, the Fog node receives more demand with the same limitation, so it must deny an excessive amount of workload. After the rising phase, even if the threshold is extended, the result will not change and will continue in the constant phase. Because the excessive threshold already covers the maximum latency of the whole system, the longest propagation delay and the longest processing delay results. Each Fog node can receive a workload from any source node in the system with its maximum capacity, so increasing the threshold never affects the outcome.

The optimal threshold, which provides the minimum blocking probability, shifts to a higher threshold depending on capacity. A larger capacity allows more acceptable workloads. The falling phase must become larger because it requires an increased threshold to achieve absolute effectiveness. Thus, the optimal point before the transition to the rising phase will also move.

V. Conclusion and Future work

The Fog computing concept is attractive for a low-latency 5G network architecture. In this research, we have constructed a mathematical model of the Fog architecture. A source node, a Fog node, the other necessary elements and their relationships are addressed in this model. To analyze our model, we examined the selection of a target Fog node when using a random policy, a lowest latency policy, and a maximum available capacity policy.

Simulation results show that the lowest latency policy provides significantly better performance due to its quickly available resource. Furthermore, we found there was an optimal value for the latency threshold in terms of the blocking probability.

Our research presented here is just a beginning step. We have developed the Fog model as an ideal model to simply evaluate the impact from both computing and communication latency. In the future, we intend to analyze results regarding other parameters in our model. Ultimately, the simulation will also have to be developed with more complexities concerning the different applications running on the source to enable more accurate analysis of a real network environment.

Acknowledgements

This work was supported in part by JSPS KAKENHI Grant-in-Aid for Scientific Research (B) Number 25280028.

References

[1] J. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. Soong, and J. Zhang, "What will 5G be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[2] D. Soldani and A. Manzalini, "A 5G infrastructure for Anything-as-a-Service," *Journal of Telecommunications System & Management*, vol. 3, no. 2, Jan. 2014.

[3] G. Fettweis, "The tactile internet: Applications and challenges," *IEEE Vehicular Technology Magazine*, vol. 9, no. 1, pp. 64–70, Mar. 2014.

[4] M. Chen, C. Ling, and Z. Wenjun, "Analysis of augmented reality application based on cloud computing," in *Proc. IEEE Intn'l Congress on Image and Signal Processing (CISP2011)*, vol. 2, Oct. 2011, pp. 569–572.

[5] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. ACM MCC Workshop on Mobile Cloud Computing*, Aug. 2012, pp. 13–16.

[6] I. Stojmenovic, "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks," in *Proc. IEEE Conference on Telecommunication Networks and Applications (ATNAC2014)*, Nov. 2014, pp. 117–122.

[7] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. ACM Workshop on Mobile Big Data (Mobidata'15)*, Jun. 2015, pp. 37–42.