Fault-Tolerant Fog Computing Models in the IoT

Ryuji Oma[1(✉)], Shigenari Nakamura[1], Dilawaer Duolikun[1], Tomoya Enokido[2], and Makoto Takizawa[1]

[1] Hosei University, Tokyo, Japan
ryuji.oma.6r@stu.hosei.ac.jp, nakamura.shigenari@gmail.com, dilewerdolkun@gmail.com, makoto.takizawa@computer.org
[2] Rissho University, Tokyo, Japan
eno@ris.ac.jp

Abstract. A huge number of devices like sensors are interconnected in the IoT (Internet of Things). In order to reduce the traffic of networks and servers, the IoT is realized by the fog computing model. Here, data and processes to handle the data are distributed to not only servers but also fog nodes. In our previous studies, the tree-based fog computing (TBFC) model is proposed to reduce the total electric energy consumption. However, if a fog node is faulty, some sensor data cannot be processed in the TBFC model. In this paper, we propose a fault-tolerant TBFC (FTBFC) model. Here, we propose non-replication and replication FTBFC models to make fog nodes fault-tolerant. In the non-replication FTBFC model, another operational fog node takes over a faulty fog node. We evaluate the non-replication FTBFC models in terms of the electric energy consumption and execution time.

Keywords: Energy-efficient fog computing · IoT(Internet of Things)
Energy-efficient IoT · Tree-based fog computing model

1 Introduction

The Internet of Things (IoT) [1,4] is composed of not only computers like servers and clients but also devices like sensors and actuators. In the cloud computing model [2,6], sensor data obtained by sensors are transmitted to servers in a cloud and processed in servers. Then, servers send actions to actuators. Here, networks are congested and servers are overloaded due to heavy traffic of sensor data from sensors.

In the fog computing model [10] of the IoT, fog nodes are between clouds of servers and devices. A fog node receives sensor data, processes the data, and sends the processed data to another fog node. For example, an average value of a collection of sensor data is calculated on fog nodes and is sent to servers. Thus, data processed by a fog node is smaller than sensor data. Servers just receive data processed by fog nodes. Thus, data and processes to handle the data are

© Springer Nature Switzerland AG 2019
F. Xhafa et al. (Eds.): 3PGCIC 2018, LNDECT 24, pp. 14–25, 2019.
https://doi.org/10.1007/978-3-030-02607-3_2

distributed to servers and fog nodes. Since processed sensor data is transmitted to servers, the traffic of the network and servers can be reduced.

The linear fog computing (LFC) model [8] and the tree-based fog computing (TBFC) model [7,9] are proposed. Here, fog nodes are hierarchically structured in a tree. Sensors send sensor data to edge fog nodes and edge fog nodes generate output data obtained by processing the sensor data. A fog node processes input data received from other fog nodes and sensors. Then, a fog node sends processed output data to a parent fog node. Thus, each fog node sends processed data to a parent fog node. Finally, processed data is sent to servers in a cloud. The electric energy consumption and execution time of fog nodes are shown to be reduced in the TBFC model compared with the cloud computing model [7,9].

In the TBFC model, if some fog node is faulty, sensor data to be processed by the faulty fog node is not sent to the parent fog node. In this paper, we newly propose a fault-tolerant tree-based fog computing (FTBFC) model which is tolerant of faults of fog nodes. We newly propose a pair of non-replication and replication FTBFC models. In the non-replication model, another fog node takes over the faulty fog node. Child fog nodes of the faulty fog node communicate with the new parent fog node. Here, since the new parent fog node receives larger volume of input data, it takes longer time to process input data from the child fog nodes and the parent fog node consumes more electric energy. The output data of the parent fog node gets also larger and ancestor nodes receive more volume of input data and consume more electric energy. In the replication FTBFC model, every fog node is replicated. Even if a fog node is faulty, another replica receives input data and processes the input data. We evaluate the non-replication FTBFC model in terms of the electric energy and execution time.

In Sect. 2, we present a system model of the IoT. In Sect. 3, we propose the FTBFC model to make fog nodes fault-tolerant. In Sect. 4, we evaluate the FTBFC model.

2 System Model

2.1 TBFC Model

The fog computing model [10] of the IoT is composed of devices, fog nodes, and clouds. Clouds are composed of servers like the cloud computing model [2].

The device layer is composed of various devices, i.e. sensors and actuators. A sensor collects data obtained by sensing events occurring in physical environment [5]. Sensor data collected by sensors is delivered to servers in networks. For example, sensor data is forwarded to neighbor sensor nodes in wireless networks as discussed in wireless sensor networks (WSNs) [12]. Sensor data is finally delivered to edge fog nodes at the bottom of the fog layer. Based on the sensor data, actions to be done by actuators are decided in the IoT. Actuators receive actions from edge fog nodes and perform the actions on the physical environment.

Fog nodes are at a layer between the device and cloud layers [11]. Fog nodes are interconnected with other fog nodes in networks. In the cloud computing model, the fog layer is just a network of routers and each fog node is a router. A fog node also supports the routing function where messages are routed to destination nodes [12]. Thus, fog nodes receive sensor data and forward the sensor data to servers in fog-to-fog communication. In addition to the routing functions, a fog node does some computation on a collection of input data sent by sensors and other fog nodes. In addition, the input data is processed and new output data, i.e. processed data of the input data is generated by a fog node. For example, a maximum value d_k is selected by searching a collection of input data d_1, ..., d_l obtained from sensor nodes. The maximum value d_k is the output data and the collection of data d_1, ..., d_l is the input data of the fog node. Output data processed by a fog node is sent to neighbor fog nodes and servers finally receive data processed by fog nodes. In addition, a fog node makes a decision on what actions actuators have to do based on sensor data. Then, edge fog nodes issue the actions to actuator nodes. A fog node is also equipped with storages to buffer data. Thus, data and processes are distributed to not only servers but also fog nodes in the fog computing model while centralized to servers in the cloud computing model.

In the tree-based fog computing (TBFC) model [7,9], fog nodes are tree-structured as shown in Fig. 1. The root node f_0 denotes a cloud of servers. The root node f_0 has child fog nodes f_{01}, ..., f_{0l_0} ($l_0 \geq 1$). Here, each fog node f_{0i} also has child fog nodes f_{0i1}, ..., $f_{0il_{0i}}$ ($l_{0i} \geq 1$). Thus, each fog node has one parent fog node and child fog nodes. A notation f_R shows f_0, i.e. label R is 0 if f_R is a root node. If f_R is an ith child of a fog node $f_{R'}$, f_R is $f_{R'i}$, i.e. label R is a concatenation $R'i$ of labels R' and i. Suppose a fog node f_R is at level m of a tree and is an ith child of a fog node $f_{R'}$. The label R of a fog node f_R shows a sequence of labels $0r_1r_2$ ... $r_{m-1}i$ where the label R' of the parent fog node $f_{R'}$ is $0r_1r_2$ ... r_{m-1}. Here, each $1 \leq r_i \leq l_{0r_1 \ldots r_{i-1}}$ for each r_i. Thus, the label $R(= 0r_1r_2$ ... $r_{m-1}i)$ of a fog node f_R shows a path, i.e. a sequence of fog nodes f_0, f_{0r_1}, $f_{0r_1r_2}$, ..., $f_{0r_1r_2 \ldots r_{m-1}}$ $(= f_R)$ from a root f_0 to the fog node f_R. Here, the length $|R|$ of the label R is m. A fog node f_R is at level $|R| - 1(= m - 1)$ in the tree. Thus, each fog node f_R has l_R (≥ 0) child fog nodes f_{R1}, ..., f_{Rl_R} ($l_R \geq 0$) where f_{Ri} is an ith child fog node of the fog node f_R. In turn, f_R is a parent fog node of the fog node f_{Ri}. An *edge* fog node f_{Ri} is at the bottom level of the tree and has no child fog node ($l_{Ri} = 0$). A root fog node f_0 has no parent node. Suppose a sensor sends data to an edge fog node $f_{RR'}$. Here, the sensor is a descendant sensor of a fog node f_R.

A fog node f_{Ri} takes input data d_{Rij} sent by each child fog node f_{Rij} ($j = 1$, ..., l_{Ri}). A process p_{Ri} in the fog node f_{Ri} does the computation on a collection D_{Ri} of input data d_{Ri1}, ..., $d_{Ril_{Ri}}$ obtained from the child fog nodes f_{Ri1}, ..., $f_{Ril_{Ri}}$, respectively, and generates output data d_{Ri}. Then, the fog node f_{Ri} sends the output data d_{Ri} to the parent fog node f_R.

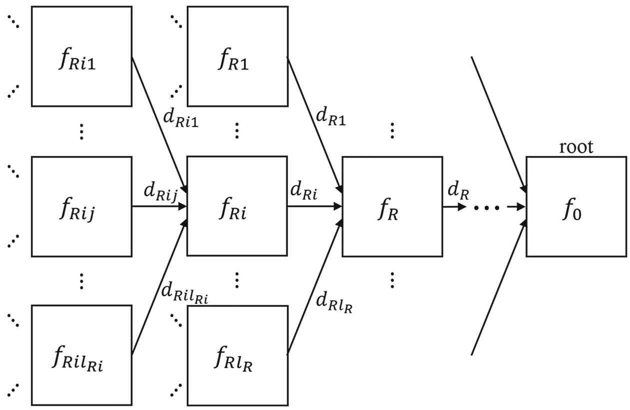


Fig. 1. TBFC model.

2.2 Model of a Fog Node

Each fog node f_{Ri} provides not only routing function but also computation on sensor data. Each process p_{Ri} of a fog node f_{Ri} is composed of four modules, an input I_{Ri}, computation C_{Ri}, output O_{Ri}, and storage S_{Ri} modules as shown in Fig. 2 [8]. The input module I_{Ri} receives data d_{Rij} from each child fog node f_{Rij} (j = 1, ..., l_{Ri}, $l_{Ri} \geq 0$). Then, the computation module C_{Ri} does the computation on the collection D_{Ri} of the input data d_{Ri1}, ..., $d_{Ril_{Ri}}$ and generates the output data d_{Ri}. The fog node f_{Ri} sends the output data d_{Ri} to the parent fog node f_R. For example, d_{Ri} is a maximum value d_{Rih} of the input data d_{Ri1}, ..., $d_{Ril_{Ri}}$. Then, the output module O_{Ri} sends the output data d_{Ri} to

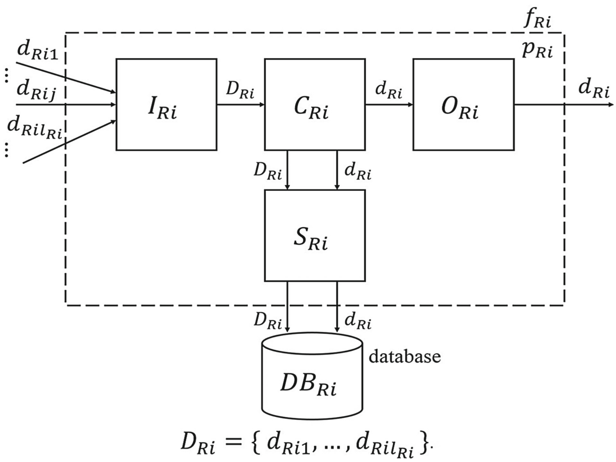
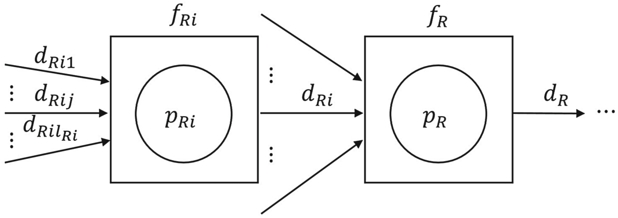


Fig. 2. Model of a process p_{Ri} on a fog node f_{Ri}.

a parent fog node f_R in networks. The storage module S_{Ri} stores the input data d_{Ri1}, ..., $d_{Ril_{Ri}}$ and output data d_{Ri} in the storage DB_{Ri}. For example, a collection of the output data d_{Ri} and input data d_{Ri1}, ..., $d_{Ril_{Ri}}$ are buffered in the storage DB_{Ri}. If the fog node f_{Ri} fails to deliver the output data d_{Ri} to the parent f_R, the fog node f_{Ri} retransmits the data d_{Ri} which is stored in the database DB_{Ri}.

A notation $|d|$ shows the size [bit] of data d. Thus, the size $|d_{Ri}|$ of the output data d_{Ri} is smaller than the input data $D_{Ri} = \{d_{Ri1}, ..., d_{Ril_{Ri}}\}$, $|d_{Ri}| \leq |D_{Ri}|$ $(= |d_{Ri1}| + ... + |d_{Ril_{Ri}}|)$. The ratio $|d_{Ri}|/|D_{Ri}|$ is the *reduction ratio* ρ_{Ri} of a fog node f_{Ri}. For example, let D_{Ri} be a set $\{v_1, v_2, v_3, v_4\}$ of four numbers showing temperature obtained by child fog nodes f_{Ri1}, ..., f_{Ri4}, respectively. If the output data d_{Ri} is a maximum value v of the values v_1, ..., v_4, the reduction ratio ρ_{Ri} of the fog node f_{Ri} is $|d_{Ri}|$ / $|D_{Ri}| = 1/4$. Here, $\rho_{Ri} \leq 1$. Suppose each of input data d_{Rih} from f_{Rih} is a sequence of values. If the output data d_{Ri} is obtained by taking the direct product of the input data d_{Ri1}, ..., $d_{Ril_{Ri}}$, the size $|d_{Ri}|$ of the output data d_{Ri} is $|d_{Ri1}| \cdot ... \cdot |d_{Ril_{Ri}}|$. Here, the reduction ratio ρ_{Ri} is larger than 1 as shown in Fig. 3.



$$|d_{Ri}| = \rho_{Ri} \cdot (|d_{Ri1}| + \cdots + |d_{Ril_{Ri}}|).$$

Fig. 3. Fog nodes.

2.3 Subprocesses on Fog Nodes

Let p be a process to handle sensor data. We assume a process p is realized as a sequence of subprocesses p_0, p_1, ..., p_m $(m \geq 1)$. The subprocess p_m takes sensor data from all the sensors and sends the output data to the subprocess p_{m-1}. Thus, each subprocess p_i receives input data from a preceding subprocess p_{i+1} and outputs data to a succeeding subprocess p_{i-1}, which is obtained by processing the input data. In the cloud computing model, the sequence of subprocesses p_0, p_1, ..., p_m are performed in a server. In the TBFC model [7,9], the subprocess p_m is performed on k^{h-1}edge fog nodes of level $h-1$. The subprocess p_{m-1} is performed on k^{h-2} fog nodes of level $h-2$. Thus, each fog node f_{Ri} of level l performs the same subprocess $p_{m-h+l+1}$ on k^l fog nodes. The subprocess p_{m-h+2} is performed on k fog nodes of level 1, one level lower than the root fog node, i.e. server f_0. A subsequence p_0, ..., p_{m-h} of subprocesses are performed on the root fog node f_0 while each subprocess p_l is performed on fog nodes at a level $l - m + h$ (for $l = m - h + 2$, ..., m) as shown in Fig. 4. In a tree of height h, there are totally $(1 - k^h)$ / $(1 - k)$ fog nodes.

Servers and devices are interconnected with networks in the cloud computing model. Here, each fog node does just the routing function. Thus, each fog node f_{Ri} is only composed of input I_{Ri} and output O_{Ri} modules. In the root node f_0, every computation on the sensor data is performed since f_0 has all the subprocesses p_0, p_1, ..., p_m.

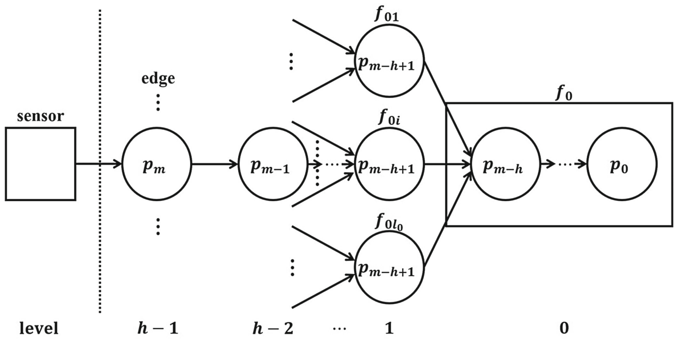


Fig. 4. Subprocesses.

3 Fault-Tolerant Fog Nodes

3.1 Non-replication Model

In the TBFC model, if a fog node f_{Ri} gets faulty, sensor data obtained by descendant sensors and processed by descendant fog nodes of the fog node f_{Ri} are unable to be delivered to the parent fog node f_R and the ancestor fog nodes of the fog node f_{Ri}. In this paper, we propose a fault-tolerant tree-based fog computing (FTBFC) model, i.e. non-replication and replication models to make fog nodes fault-tolerant in the TBFC model.

Suppose a fog node f_{Rij} is faulty in the FTBFC model as shown in Fig. 5. Here, f_{Ri} shows a parent fog node of the faulty fog node f_{Rij}. Fog nodes f_{Rij1}, ..., $f_{Rijl_{Rij}}$ ($l_{Rij} \geq 1$) are child fog nodes of the faulty fog node f_{Rij}. A fog node f_{Rip} is a child fog node where the parent fog node f_{Ri} is also the parent of the faulty fog node f_{Rij}. A fog node f_{Rmq} is a fog node which is at the same level of the faulty fog node f_{Rij}. This means, the fog nodes f_{Rip} and f_{Rmq} have the same subprocess as the faulty fog node f_{Rij}.

There are the following ways to be tolerant of the faults of the fog node f_{Ri}.

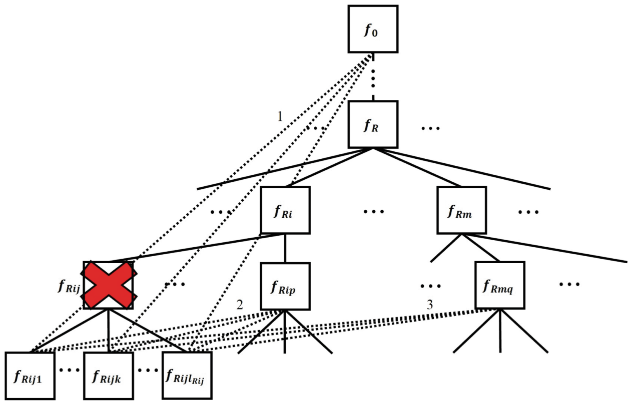


Fig. 5. Non-replication FTBFC model.

1. Each child node f_{Rijk} sends the output data d_{Rijk} to the root node f_0, i.e. the cloud of servers [Fig. 5].
2. Each child node f_{Rijk} takes one fog node f_{Rip} as a new parent fog node [Fig. 5]. The fog node f_{Rip} is a child node of the parent fog node f_{Ri} of the faulty fog node f_{Rij}.
3. Each child node f_{Rijk} takes one fog node f_{Rmq} $(m \neq i)$ as a parent node [Fig. 5]. The fog node f_{Rmq} is at the same level as the faulty fog node f_{Rij}.
4. Each child fog node f_{Rijk} takes one fog node $f_{R'}$ as a parent fog node, where $f_{R'}$ is at the same level as f_{Rij}.
5. One child fog node f_{Rijk} promotes to a parent node. Here, the process is transferred to the fog node f_{Rijk} from the sibling fog node f_{Rip} [Fig. 6].

In the way 1, every subprocess is installed in the root fog node, i.e. a server in a cloud. The root node f_0 can process the output data d_{Rijk} of every child fog node f_{Rijk}.

In the way 2, the fog node f_{Rip} has the same subprocess as the faulty fog node f_{Rij}. Here, the output data d_{Rijk} of every child fog node f_{Rijk} can be processed by the fog node f_{Rip} on behalf of the faulty fog node f_{Rij}.

In the way 3, the fog node f_{Rmq} has the same subprocess as the faulty fog node f_{Rij}. Differently from the way 2, the new parent fog node f_{Rmq} has a parent fog node f_{Rm} different from the fog node f_{Ri}.

In the way 4, the fog node $f_{R'}$ is at the same level as the faulty fog node f_{Rij}. The fog node $f_{R'}$ has the same subprocess as f_{Rij}. Let $f_{R''}$ be a least upper bound (lub) of the faulty fog nodes f_{Rij} and $f_{R'}$. In the way 2, $f_{R''}$ is f_{Ri}. In the way 3, $f_{R''}$ is f_R.

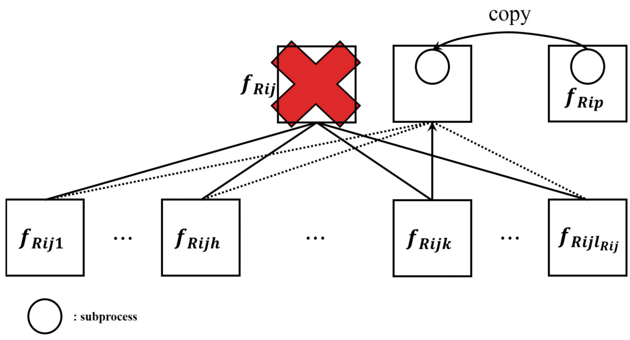


Fig. 6. Promotion.

In the way 5, one child fog node f_{Rijk} is promoted to a parent fog node of the other child fog nodes f_{Rij}, ..., $f_{Rijl_{Rij}}$. Since the fog node f_{Rijk} does not support the computation module of the faulty fog node f_{Rij}, the computation module is transmitted to the fog node f_{Rijk} from a fog node f_{Rip}. Here, the fog node f_{Rij} performs the computation modules itself and of both itself and the faulty fog node f_{Rij}.

If a fog node f_{Rij} is detected to be faulty, a new parent fog node of the child fog nodes f_{Rij1}, ..., $f_{Rijl_{Rij}}$ has to be selected. In this paper, a new fog node is selected so that the electric energy consumption of fog nodes can be reduced. In paper [7], the electric energy consumption $TE_{Rij}(x)$ [J] and execution time $ET_{Rij}(x)$ [sec] of a fog node f_{Rij} to receive and process an input data D_{Rij} of size x and send the output data d_{Rij}. For example, the electric energy consumption and execution time of a new parent fog node f_{Rip} increase to $TE_{Rip}(|D_{Rip}| + |D_{Rij}|)$ and $ET_{Rip}(|D_{Rip}| + |D_{Rij}|)$, respectively, in the way 2. In addition, the size of the output data d_{Rip} is $\rho_{Rip} \cdot (|D_{Rip}| + |D_{Rij}|)$. In the way 3, a parent fog node f_{Ri} does not receive output data d_{Rij} from the faulty fog node f_{Rij}. Hence, the electric energy consumption and execution time of the fog node f_{Ri} decrease to $TE_{Ri}(|D_{Ri}| - |d_{Ri}|)$ and $ET_{Ri}(|D_{Ri}| - |d_{Ri}|)$, respectively.

3.2 Replication Model

Every fog node f_{Ri} is replicated to replicas f_{Ri}^1, ..., $f_{Ri}^{r_{Ri}}$ ($r_{Ri} \geq 1$). There are the following replication schemes [3].

1. Active replication
2. Passive replication
3. Semi-active replication
4. Semi-passive replication

In the active replication, every replica f_{Ri}^h receives the same input data, does the same computation, and sends the same output data.

4 Evaluation

We evaluate the non-replication FTBFC model in this paper. We consider a balanced binary tree with height h, i.e. $\langle 2, h\rangle$ tree of the FTBFC model, where each fog node has 2 child fog nodes and every edge fog node is at level $h-1$. There are totally 2^{h-1} edge fog nodes. Each edge fog node receives the same volume of sensor data. Sensor nodes totally send x to 2^{h-1} edge nodes. For example, the total volume 1 [MB] (= 8,388,608 [bit]) of sensor data is sent to the edge fog nodes. Hence, each edge fog node receives sensor data of $8,388,608/2^{h-1}$ [bit]. In this evaluation, a process p is a sequence of subprocesses p_0, p_1, ..., p_m. The computation complexity of each subprocess is $O(x)$ or $O(x^2)$ for input data of size x.

In this paper, we evaluate the ways 2, 3, and 4. In the evaluation, one fog node is randomly selected to be faulty for each level k $(0 < k < h-1)$. Then, we calculate the total electric energy consumption and execution time of the fog nodes.

First, one fog node f_{Ri} in the tree is randomly selected as a faulty fog node. Then, we have to select a new parent fog node which is the same level of the faulty fog node f_{Ri}.

1. A sibling fog node f_{Rj} of f_{Ri} is selected. Since we consider a binary tree, the sibling fog node f_{Rj} is f_{R2} if f_{Ri} is f_{R1}, others f_{R1}.
2. A new parent fog node $f_{R'j}$ is randomly selected in fog nodes of the same level as the faulty fog node f_{Ri}.

For a fog node f_{Ri} and a new parent fog node $f_{R'j}$, the total electric energy TEE and execution time TET of the fog nodes are calculated in the simulation. Figures 7 and 8 show the total electric energy TEE for height h where the selection ways of a new parent node is 1 and 2 with computation complexity

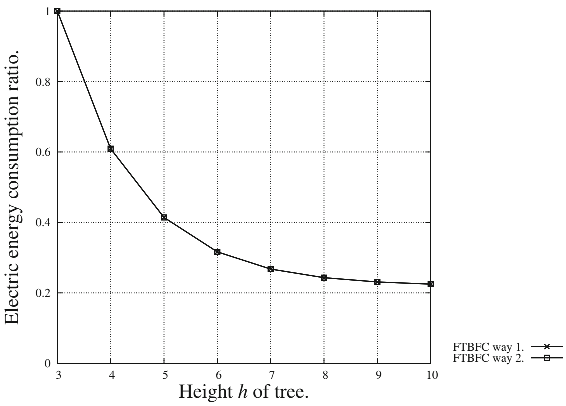


Fig. 7. Total electric energy consumption with computation complexity $O(x)$ for height h.

$O(x)$ and $O(x^2)$ of each fog node for size x of input data, respectively. As shown in Fig. 8, the TEE can be reduced if a sibling fog node is taken as a new parent fog node for $O(x^2)$.

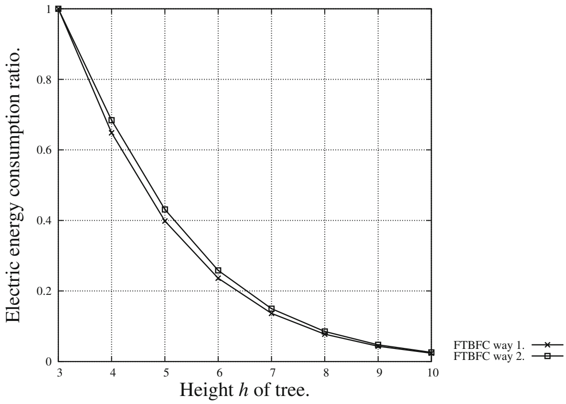


Fig. 8. Total electric energy consumption with computation complexity $O(x^2)$ for height h.

Figures 9 and 10 show the total execution time TET of the fog nodes for height h, where the selection ways of a new parent node is 1 and 2, with computation complexity $O(x)$ and $O(x^2)$, respectively. As shown in Figs. 9 and 10, the TET can be reduced if a sibling fog node is taken as a new parent fog node.

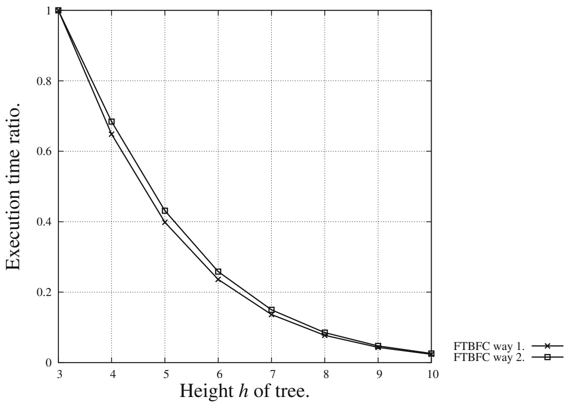


Fig. 9. Total execution time with computation complexity $O(x)$ for height h.

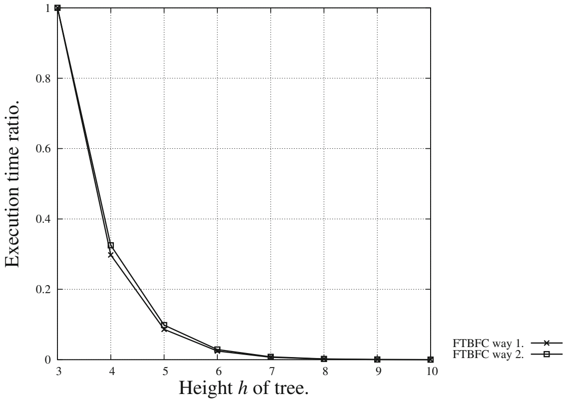


Fig. 10. Total execution time with computation complexity $O(x^2)$ for height h.

5 Concluding Remarks

The IoT is scalable and includes sensors and actuators in addition to servers. Processes and data are distributed to not only servers but also fog nodes in the fog computing model in order to reduce the delay time and processing overhead. In this paper, we proposed the fault-tolerant tree-based fog computing (FTBFC) model with non-replication and replication types. In the non-replication FTBFC model, another fog node which has the same subprocess as a faulty fog node supports child fog nodes of the faulty fog node. We evaluated the FTBFC model in terms of the electric energy consumption and execution time for computation complexity $O(x)$ and $O(x^2)$ of each fog node where x is size of input data. We showed the total electric energy consumption and total execution time can be reduced if a sibling fog node is selected as a new parent node.

Acknowledgments. This work was supported by JSPS KAKENHI grant number 15H0295.

References

1. Arridha, R., Sukaridhoto, S., Pramadihanto, D., Funabiki, N.: Classification extension based on IOT-big data analytic for smart environment monitoring and analytic in real-time system. Int. J. Space-Based Situated Comput. (IJSSC) **7**(2), 82–93 (2017). https://doi.org/10.1504/IJSSC.2017.10008038
2. Creeger, M.: Cloud computing: an overview. Queue **7**(5), 3–4 (2009)
3. Defago, X., Schiper, A., Sergent, N.: Semi-passive replication. In: Proceedings of the IEEE 17th Symposium on Reliable Distributed Systems, pp. 43–50 (1998)
4. Hanes, D., Salgueiro, G., Grossetete, P., Barton, R., Henry, J.: IoT Fundamentals: Networking Technologies, Protocols, and use Cases for the Internet of Things. Cisco Press (2018)

5. Ito, K., Hirakawa, G., Arai, Y., Shibata, Y.: A road condition monitoring system using various sensor data in vehicle-to-vehicle communication environment. Int. J. Space-Based Situated Comput. (IJSSC) **6**(1), 21–30 (2016). https://doi.org/10.1504/IJSSC.2016.076572
6. Messina, F., Mikkilineni, R., Morana, G.: Middleware, framework and novel computing models for grid and cloud service orchestration. Int. J. Grid Util. Comput. (IJGUC) **8**(2), 71–73 (2017). https://doi.org/10.1504/IJGUC.2017.10006830
7. Oma, R., Nakamura, S., Duolikun, D., Enokido, T., Takizawa, M.: Evaluation of an energy-efficient tree-based model of fog computing. In: Proceedings of the 21st International Conference on Network-Based Information Systems (NBiS-2018) (accepted) (2018)
8. Oma, R., Nakamura, S., Enokido, T., Takizawa, M.: An energy-efficient model of fog and device nodes in IOT. In: Proceedings of IEEE the 32nd International Conference on Advanced Information Networking and Applications (AINA-2018), pp. 301–306 (2018)
9. Oma, R., Nakamura, S., Enokido, T., Takizawa, M.: A tree-based model of energy-efficient fog computing systems in IOT. In: Proceedings of the 12th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2018), pp. 991–1001 (2018)
10. Rahmani, A.M., Liljeberg, P., Preden, J.S., Jantsch, A.: Fog Computing in the Internet of Things. Springer, Berlin (2018)
11. Yao, X., Wang, L.: Design and implementation of IOT gateway based on embedded μtenux operating system. Int. J. Grid Util. Comput. **8**(1), 22–28 (2017). https://doi.org/10.1504/IJGUC.2017.10008769
12. Zhao, F., Guibas, L.: Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann Publishers, Amsterdam (2004)