# Data Exchange Algorithm at Aggregate Level in the TWTBFC Model

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

[1] Hosei University, Tokyo, Japan
{yinzhe.guo.3e,ryuji.oma.6r}@stu.hosei.ac.jp, makoto.takizawa@computer.org
[2] Rissho University, Tokyo, Japan
eno@ris.ac.jp

**Abstract.** In the TBFC (Tree-Based Fog Computing) and TWTBFC (Two-Way TBFC) models the electric energy consumed by fog nodes and servers can be reduced in the fog computing (FC) model. Here, fog nodes are hierarchically structured in a height-balanced tree, where a root node is a cloud of servers, leaf nodes are edge nodes which communicate with devices, and each node receives data from child nodes and sends the processed data to a parent node. In the TWTBFC model, nodes send processed data to not only a parent node but also each child node. In order to reduce the network traffic in the TWTBFC model, only aggregate nodes at some level collect the output data of every other aggregate node, i.e. aggregate data. Since only target actuators are to be activated, the aggregate data has to be only delivered to target actuators. Nodes whose descendant actuators are target ones are relay nodes. On receipt of aggregate data, only relay nodes forward the aggregate data to the child nodes. We evaluate the new TWTBFC model in terms of energy consumption of nodes and number of messages transmitted to deliver aggregate data to edge nodes.

**Keywords:** Energy-efficient fog computing · IoT (Internet of Things) · Two-way TBFC (TWTBFC) model · Aggregate node

## 1 Introduction

The IoT (Internet of Things) [5,7] is composed of not only computers like servers and clients but also millions of devices, i.e. sensors and actuators installed in various things like glasses and cars [11,14]. Compared with traditional information systems like the cloud computing (CC) model [4], the IoT is more scalable and huge amount of data from sensors are transmitted in networks and are processed by application processes on servers. The fog computing (FC) model [16] is proposed to reduce the network and server traffic of the IoT (Internet of Things). On the other hand, huge amount of electric energy is consumed by nodes. In order to not only increase the performance but also reduce the electric energy


L. Barolli et al. (Eds.): BWCCA 2019, LNNS 97, pp. 114–124, 2020.
https://doi.org/10.1007/978-3-030-33506-9_11

consumption of the IoT, the TBFC (Tree-based Fog Computing) model is proposed in our previous studies [3,11,12,15]. Here, fog nodes are hierarchically structured in a height-balanced tree. A root node is a cloud of servers and leaf nodes are edge nodes which receive sensor data from sensors and send actions to actuators. Each fog node has one parent node and child nodes. Each node receives input data from the child nodes. Then, the fog node processes the input data and sends the output data, obtained by processing the input data to a parent node. A server in a cloud finally receives data processed by fog nodes. Then, the servers delivers actions to actuators through networks of fog nodes. While the traffic of servers and networks can be reduced, it takes time to deliver actions to actuators.

The TWTBFC (Two-Way TBFC) model [9,10] is also proposed to reduce delay time to deliver actions to actuators. Here, a node not only sends output data to a parent node in a same way as the TBFC model but also forwards the output data to the child nodes. In addition, some level is taken as *aggregate* level. Nodes at aggregate level are *aggregate* nodes [8]. Each aggregate node collects the output data from every other aggregate nodes. Then, each aggregate node obtains aggregate data which is a collection of output data of all the aggregate nodes. Then, the aggregate data is transmitted from each aggregate node down to the descendant edge nodes. Then, edge nodes make a decision on actions and activate child actuators by sending the actions. Since the aggregate data is transmitted to every edge node, more number of messages are transmitted in networks. On the other hand, only some edge node is required to activate its child actuators. Actuators to be activated for the aggregate data are *target* ones. Nodes whose descendant actuators are target ones are referred to as *relay* ones. In order to reduce the network traffic, we propose a new model where only relay nodes forward the aggregate data to the child nodes. In the evaluation, we show the number of messages and energy consumption of nodes to obtain the aggregate data and deliver the aggregate data to target edge nodes.

In Sect. 2, we present the TWTBFC model of the IoT. In Sect. 3, we present the power consumption and computation module of a fog node. In Sect. 4, we evaluate the TWTBFC model.

## 2 Two-Way Tree-Based Fog Computing (TWTBFC) Model

### 2.1 TBFC Model

The fog computing (FC) model [16] to efficiently realize the IoT [11] is composed of sensor and actuator devices, fog nodes, and clouds. Clouds are composed of servers like the cloud computing (CC) model [4]. In the TBFC (Tree-Based Fog Computing) model [12,15], fog nodes are hierarchically structured in a height-balanced tree as shown in Fig. 1. Here, the root node $f$ denotes a cloud of servers. Fog nodes at the bottom level are *edge* nodes which communicate with sensors and actuators.

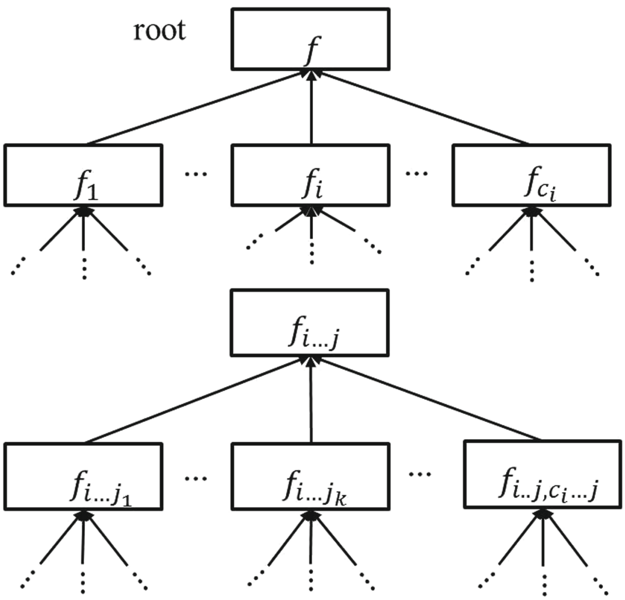


**Fig. 1.** TBFC model.

Each node $f_R$ has $c_R$ ($\geq 0$) child nodes $f_R, \ldots, f_{R,c_R}$. Here, $f_{Ri}$ shows the $i$th child node of the fog node $f_R$ and in turn $f_R$ is a parent node of the node $f_{Ri}$. $ch(f_R)$ is a set $\{f_{R1}, \ldots, f_{R,c_R}\}$ of child nodes of a node $f_R$. $pt(f_{Ri})$ is a parent node $f_R$ of a node $f_{Ri}$. For example, the second child node of a root node $f$ is $f_2$, and the first child node of the node $f_2$ is $f_{21}$. Thus, the label $R$ of a fog node $f_R$ is a sequence of numbers and shows a path from the root node $f$ to the fog node $f_R$. Let $an(f_R)$ be a set of ancestor nodes of a node $f_R$. $dn(f_R)$ shows a set of descendant nodes of a node $f_R$ and $sn(f_R)$ is a set of nodes which are at the same level as a node $f_R$.

In the cloud computing (CC) model, an application process $p$ is performed on servers to process sensor data sent by sensors in networks. In this paper, an application process $p$ is assumed to be linear, i.e. a sequence of subprocesses $p_0, p_1, \ldots, p_{h-1}$. The edge subprocess $p_{h-1}$ takes input data from sensors. The root subprocess $p_0$ is performed on a root node $f$, i.e. servers. Each subprocess $p_i$ takes data from a subprocess $p_{i-1}$ and gives the processed data to a subprocess $p_{i+1}$. In the TBFC tree of height $h$, each subprocess $p_i$ is performed on nodes of level $i$. Let $p(f_R)$ show a subprocess to be performed in a node $f_R$.

A node $f_R$ takes input data $d_{Ri}$ from each child node $f_{Ri}$ ($i = 1, \ldots, l_R$). $D_R$ shows a collection of the input data $d_{R1}, \ldots, d_{R,l_R}$ from child nodes $f_{R1}, \ldots, f_{R,l_R}$, respectively. The node $f_R$ obtains output data $d_R$ by doing the computation $f(p_R)$ on the input data $D_R$. Then, the node $f_R$ sends the output data $d_R$ to a parent node $pt(f_R)$.

A notation $|d|$ shows the size [Byte] of data $d$. Let $i_R$ and $o_R$ be the size of $|D_R|$ and $|d_R|$ of input data $D_R$ and output data $d_R$, respectively. The ratio $|o_R|/|i_R|$ is the *output ratio* $\rho_R$ of a node $f_R$. Here, $o_R = \rho_R \cdot i_R$. For example, if a fog node $f_R$ obtains an average value of the input data $d_{R1}, \ldots, d_{R,l_R}$, the output ratio $\rho_R$ is $1/l_R$.

## 2.2 Aggregate Nodes

Each fog node $f_R$ receives input data $D_R = \{d_{R1}, \ldots, d_{R,l_R}\}$ from the child nodes $f_{R1}, \ldots, f_{R,l_R}$ and obtains output data $d_R$ by processing the output data $D_R$. At some level $l$ of the tree, each fog node sends the output data $d_R$ to every other node $f_v$ and receives output data $d_v$ from every other fog node $f_v$ in addition to sending the output data $d_R$ to the parent node $pt(f_R)$. Then, each fog nodes $f_R$ obtains a collection $AD_l$ of output data from every node at the same level, i.e. here, nodes of level $l$ are referred to as *aggregate nodes* and the level $l$ is *aggregate level* in the tree. Let $AN_l$ be a set of aggregate nodes of level $l$ in the tree. The data $AD_l$ is an *aggregate data* which is a set of output data obtained by all the aggregate nodes of the level $l$, i.e. $AD_l = \{d_s \mid f_s \in AN_l\}$. By using the aggregate data $AD_l$, actuators to be activated are decided. An actuator to be activated for the aggregate data $AD_l$ is referred to as *target* actuator. Here, a node which is an ancestor of a target actuator is referred to as *relay* node. Each aggregate node $f_R$ sends the aggregate data $AD_l$ to each relay child node. Let $RN_R(\in ch(f_R))$ be a subset of relay child nodes of a node $f_R$. Even if a non-relay child node $f_{Ri}$ receives the aggregate data $AD_l$, the node $f_{Ri}$ does not forward $AD_l$ to any child node $f_{Rij}$. Thus, only a relay node $f_{Ri}$ forwards the aggregate data $AD_l$ to child nodes $f_{Rij}$. Eventually, a relay edge node $f_R$ receives the aggregate data $AD_l$. The relay edge node $f_R$ makes a decision on actions to be performed on child target actuators and issues the actions to the target actuators.

Target actuators are localized in some area for the aggregate data $AD_l$ as shown in Fig. 2. A node $f_R$ is referred to as *broadcast* node if every descendant edge node is a relay one. This means, actuators in an area covered by a broadcast node are activated. On receipt of the aggregate data $AD_l$, a broadcast node $f_R$ forwards the aggregate data $AD_l$ to every child node. Every descendant node of a broadcast node $f_R$ is a broadcast node. An aggregate node forwards the aggregate data $AD_l$ to relay nodes. A relay node forwards the aggregate data $AD_l$ to relay nodes. Eventually, a relay node $f_R$ forwards the aggregate data $AD_l$ to every child node. Here, $f_R$ is a broadcast node. A level at which the node $f_R$ exists is a *broadcast* ($b$) level. At higher level than the $b$ level, a relay node send the aggregate data $AD_l$ to only relay nodes, i.e. unicasts $AD_l$ to each relay nodes

Let us consider a node $f_R$ which has child fog nodes $f_{R1}, \ldots, f_{R,c_R}$. Let $x_R$ stand for the size $|d_R|$ of the output data $d_R$. The size $x_R$ of the output data $d_R$ of a node $f_R$ is given as $x_R = \rho_R \cdot (\Sigma_{i=1}^{c_R} x_{Ri})$. Here, $\rho_R$ is the output ratio of the node $f_R$. If $f_R$ is an edge node, each size $x_{Ri}$ shows the size of the sensor data $d_{Ri}$ from a child sensor $s_{Ri}$. Thus, the size $x_{Ri}$ of the output data $d_{Ri}$ of each child

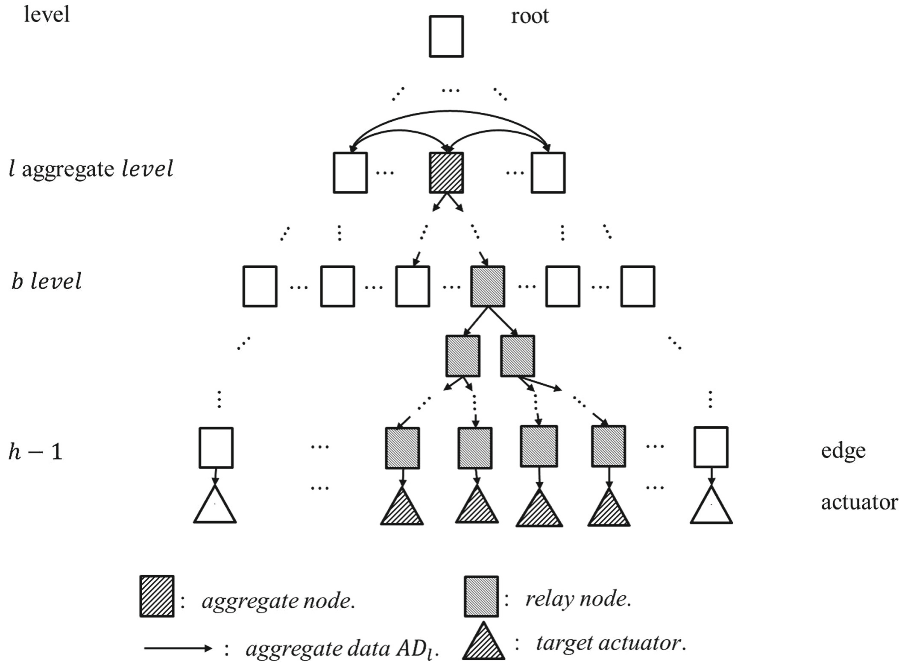


**Fig. 2.** Relay nodes and target actuators.

node $f_{Ri}$ of level $l$ can be obtained, and then the size $x_R$ of an aggregate node $f_R$ is calculated. Each aggregate node $f_R$ of aggregate level $l$ obtains the aggregate data $AD_l$ whose size $as_l (= |AD_l|)$ is $\Sigma_{f_s \in AN_l} (\rho_s \cdot \Sigma_{i=1}^{c_s} x_{si})$. The aggregate data $AD_l$ of size $as_l$ is forwarded to target edge nodes. On receipt of the aggregate data $AD_l$, a relay edge node $f_R$ decides on actions and sends the actions to the target actuators $a_{R1}, \ldots, a_{R,al_R}$.

# 3 Power Consumption and Computation Models of a Fog Nodes

## 3.1 Upward Transmission

A fog node $f_R$ is assumed to be implemented to be a sequence of input ($I_R$), computation ($C_R$), and output ($O_R$) modules. The input module $I_R$ receives input data $d_{Ri}$ from a child node $f_{Ri}$ and the output module $O_R$ sends output data $d_R$ to a parent node $pt(f_R)$. The computation module $C_R$ is a subprocess $p(f_R)$ which generates the output data $d_R$ by processing input data $D_R = d_{R1}, \ldots, d_{R,C_R}$. In this paper, we assume the $I_R$, $C_R$, and $O_R$ modules are sequentially performed in a fog node $f_R$ on receipt of the input data $D_R$.

It takes time to perform the $I_R$, $C_R$, and $O_R$ modules of a node $f_R$. Let $TI_R(x)$, $TC_R(x)$, and $TO_R(x)$ show the execution time [sec] of the input $I_R$, computation $C_R$, and output $O_R$ modules of a node $f_R$ for data of size $x$, respectively. The execution time $TC_R(x)$ depends on the computation complexity of a subprocess $p(f_R)$. In this paper, the computation complexity of the subprocess $p(f_R)$ is assumed to be $O(x)$ or $O(x^2)$. That is, the execution time $TC_R(x)$ of the computation module ($C_R$) is $ct_R \cdot C_R(x)$ where $C_R(x) = x$ or $C_R(x) = x^2$ and $ct_R$ is a constant. A pair of execution time $TI_R(x)$ and $TO_R(x)$ to receive and send data of size $x$, respectively, are proportional to the data size $x$, i.e. $TI_R(x) = rt_R \cdot x$ and $TO_R(x) = st_R \cdot x$, where $st_R$ and $rt_R$ are constants. Thus, the execution time $TC_R(x)$, $TI_R(x)$, and $TO_R(x)$ are given as follows:

$$TC_R(x) = ct_R \cdot C_R(x). \tag{1}$$

$$TI_R(x) = rt_R \cdot x. \tag{2}$$

$$TO_R(x) = st_R \cdot x. \tag{3}$$

It takes time $TF_R(x)$ [sec] for each node $f_R$ to receive and process input data $D_R$ of size $x$ and send the output data $d_R$ to a parent node $pt(f_R)$:

$$TF_R(x) = TI_R(x) + TC_R(x) + \delta_R \cdot TO_R(\rho_R \cdot x). \tag{4}$$

Here, if $f_R$ is a root node, $\delta_R = 0$, else $\delta_R = 1$. The execution time $TI_R(x)$ of the $I_R$ module realized in a Raspberry Pi 3 model B [2] node is five times longer than the execution time $TO_R(x)$ of the $O_R$ module, i.e. $rt_R = 5 \cdot st_R$ and $ct_R = rt_R/2$ [13]. That is, $ct_R : st_R : rt_R = 1 : 2.5 : 0.5$.

$EI_R(x)$, $EC_R(x)$, and $EO_R(x)$ show the electric energy [J] consumed by the input $I_R$, computation $C_R$, and output $O_R$ modules [11] of a node $f_R$ for data of size $x$, respectively. In this paper, we assume each node $f_R$ follows the SPC (Simple Power Consumption) model [5–7]. The power consumption of a node $f_R$ to perform the computation module $C_R$ $(= p(f_R))$ is $maxE_R$ [W]. In a Raspberry Pi Model B, node $f_i$, $maxE_i = 3.7$ [W]. The energy consumption $EC_R(x)$ [J] of the computation module $C_R$ of a node $f_R$ to process data of size $x$ $(> 0)$ is $EC_R(x) = maxE_R\,[\mathrm{W}] \cdot TC_R(x)$ [sec].

A pair of the electric power $PI_R$ and $PO_R$ [W] are consumed to perform the input $I_R$ and output $O_R$ modules, respectively [5–7]. $PI_R$ and $PO_R$ are $re_R \cdot maxE_R$ and $se_R \cdot maxE_R$, respectively, where $0 < se_R \leq re_R \leq 1$. For example, $se_R = 0.676$ and $re_R = 0.729$ in the Raspberry Pi 3 model B node $f_R$ [13]. The energy consumption $EI_R(x)$ and $EO_R(x)$ [J] to receive and send data of size $x$ $(> 0)$ are $EI_R(x) = PI_R[\mathrm{w}] \cdot TI_R(x)[\mathrm{sec}]$ and $EO_R(x) = PO_R[\mathrm{w}] \cdot TO_R(x)[\mathrm{sec}]$, respectively.

Each node $f_R$ consumes the energy $EF_R(x)$ to reduce and process the input data $D_R$ of size $x$ and send the processed data $d_R$ of size $\rho_R \cdot x$:

$$\begin{aligned} EF_R(x) &= EI_R(x) + EC_R(x) + \delta_R \cdot EO_R(\rho_R \cdot x) \\ &= (re_R \cdot TI_R(x) + TC_R(x) + \delta_R \cdot se_R \cdot TO_R(\rho_R \cdot x)) \cdot maxE_R \\ &= (re_R \cdot rt_R \cdot x + ct_R \cdot C_R(x) + \delta_R \cdot se_R \cdot st_R \cdot \rho_R \cdot x) \cdot maxE_R. \end{aligned} \tag{5}$$

### 3.2 Downward Transmission

Each aggregate node $f_R$ consumes electric energy and takes time to collect the aggregate data $AD_l$ from other aggregate nodes of aggregate level $l$ as shown in Fig. 2. $AN_l$ is a set of aggregate nodes at level $l$. Each aggregate node $f_R$ of level $l$ sends the output data $d_R$ to and receives the output data $d_s$ from every other aggregate node $f_s$. Let $o_s$ be the size $|d_s|$ of the output data $d_s$. Then, the aggregate node $f_R$ forwards the aggregate data $AD_l$ to the child nodes $f_{R1}, \ldots, f_{R,c_R}$. The aggregate data $AD_l$ is a set $\{d_s \mid f_s \in AN_l\}$ of output data of every aggregate node. The size $as_l (= |AD_l|)$ of the aggregate data $AD_l$ is:

$$as_l = \Sigma_{f_s \in AN_l} |o_s| = \Sigma_{f_s \in AN_l} (\rho_s \cdot \Sigma_{i=1}^{c_s} o_{si}). \tag{6}$$

It takes time $AEX_R$ of an aggregate node $f_R$ to send the output data $d_R$ and to receive the output data $d_s$ from every other aggregate node $f_s$:

$$AEX_R = TO_R(o_R) \cdot |AN_l| + \Sigma_{f_s \in AN_R} TI_R(o_s). \tag{7}$$

Then, a relay aggregate node $f_R$ of aggregate level $l$ sends the aggregate data $AD_l$ to relay child nodes. Let $RN_R$ $(\subseteq ch(f_R))$ be a set of relay child nodes of a node $f_R$. The total time $ATO_R$ [sec] of a relay aggregate node $f_R$ is given as follows:

$$ATO_R = TO_R(o_R) \cdot |AN_l| + \Sigma_{f_s \in AN_l} TI_R(o_s) + TO_R(as_l) \cdot |RN_R|. \tag{8}$$

The relay aggregate node $f_R$ consumes the energy $AEO_R$ [J] as follows:

$$\begin{aligned} AEO_R = (&se_R \cdot TO_R(o_R) \cdot |AN_l| + se_R \cdot TO_R(|as_l|) \\ &+ re_R \cdot \Sigma_{f_s \in AN_l} TI_R(o_s)) \cdot maxE_R. \end{aligned} \tag{9}$$

A descendant relay node $f_R$ of the aggregate nodes receives the aggregate data $AD_l$. If $f_R$ is a relay node, the node $f_R$ forwards the aggregate data $AD_l$ to the relay child nodes. The execution time $ATO_R$ of a relay node $f_R$ is as follows:

$$\begin{aligned} ATO_R &= TI_R(as_l) + TO_R(as_l) \cdot |RN_R| \\ &= rt_R \cdot as_l + st_R \cdot as_l \cdot |RN_R|. \end{aligned} \tag{10}$$

Each node $f_R$ of level $k$ $(< l)$ consumes energy $AEO_R$ to forwards the aggregate data $AD_l$ to the descendant edge nodes.

$$\begin{aligned} AEO_R &= (re_R \cdot TI_R(as_l) + se_R \cdot TO_R(as_l)) \cdot maxE_R \\ &= (re_R \cdot rt_R \cdot as_l + se_R \cdot st_R \cdot as_l \cdot |RN_R|) \cdot max_R. \end{aligned} \tag{11}$$

The higher the aggregate level $l$ is, the smaller size of the aggregate data $AD_l$ and the fewer number of messages are exchanged among the aggregate nodes. However, the more number of messages are transmitted to deliver the aggregate data $AD_l$ to edge nodes.

## 4 Evaluation

We evaluate the TWTBFC model of the IoT in terms of electric energy consumption of fog nodes and number of messages transmitted by fog nodes. The TWTBFC model is composed of fog nodes structured in a tree. In this paper, we consider a height-balanced $k$-ary tree of fog nodes, whose height is $h$. The output ratio $\rho_R$ of each fog node $f_R$ is assumed to be the same $\rho$, i.e. $\rho_R = \rho$. We assume a root node is a server $f$ with a pair of Inter Xeon E5-2667 CPUs [1], where the minimum electric power consumption $minE_0$ is 126.1 [W] and the maximum electric power consumption $maxE_0$ is 301.3 [W]. Each fog node $f_R$ is realized by a Raspberry Pi 3 Model B [2]. Here, the minimum power $minE_R$ is 2.1 [W] and the maximum power $maxE_R$ is 3.7 [W] [3]. The computation ratio $CR_R$ of each fog node $f_R$ is 0.879/4.75 = 0.185, where the computation rate of the root node $f$ is 1. This means, the computation speed of the node $f_R$ is 18.5 [%] of the root node $f$.

$AN_l$ is a set of aggregate nodes at aggregate level $l$. There are $k^l$ $(= |AN_l|)$ aggregate nodes at aggregate level $l$ in the tree. As presented in the preceding section, each aggregate node $f_R$ exchanges the output data $d_R$ with every other aggregate node and obtains the aggregate data $AD_l$ $(= \bigcup_{f_s \in AN_l} d_s)$. Each aggregate node $f_R$ sends the output data $d_R$ to $(k^l - 1)$ aggregate nodes and receives output data from the other $(k^l - 1)$ aggregate nodes. Hence, totally, $k^l \cdot (k^l - 1)$ messages are transmitted. Here, the size of sensor data which each edge node receives from the child sensors is assumed to be one. A node of level $h-1$ receives data of total size $k$ from $k$ edge nodes and sends the output data $d_R$ of size $\rho \cdot k$. Thus, each aggregate node $f_R$ receives the output data $D_R$ of size $(\rho k)^{h-1-l} \cdot k$ and generates the output data of size $(\rho k)^{h-l-2}$. Hence, the total size $k^l \cdot (k^l - 1) \cdot (\rho k)^{h-l-1}$ of data is exchanged among the aggregate nodes.

For the aggregate data $AD_l$, the target actuators are in some area. In this paper, we assume there is one broadcast node at broadcast level $b$ and every descendant edge node of the broadcast node is a target one. At level $q$ $(l \leq q < b)$, one relay node sends the output data $AD_l$ to one child relay node. Then, a broadcast node sends the output data $AD_l$ to $k$ child nodes. Thus, $(b - l)$ messages are transmitted to deliver the aggregate data $AD_l$ to the broadcast node. The node $f_R$ sends the aggregate data $AD_l$ to $k$ child nodes and each child node forwards the message $AD_l$ to $k$ child nodes. Thus, totally, $k + k^2 + \ldots + k^{h-1-b} = k \cdot (1 - k^{h-1-b})/(1-k)$ messages are transmitted. For a broadcast node $f_R$ of broadcast level $b$, there are $k^{h-1-b}$ descendant edge nodes. Here, totally $(b-l) + k \cdot (1 - k^{h-1-b})/(1-k)$ messages are transmitted. The total size of data transmitted is $[(b - l) + k \cdot (1 - k^{h-1-b})/(1 - k)] \cdot k^l \cdot (k^l - 1) \cdot (\rho k)^{h-1-l}$.

We assume $k = 2$, the height $h$ of tree is 10 $(h = 10)$ and the output ratio $\rho = 0.5$ in the evaluation.

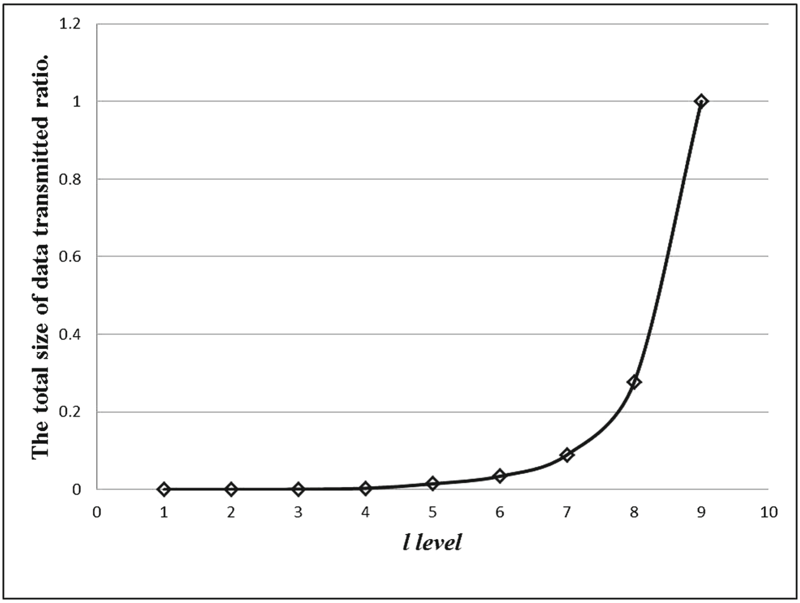


**Fig. 3.** Total size of data transmitted ($b = 5$).

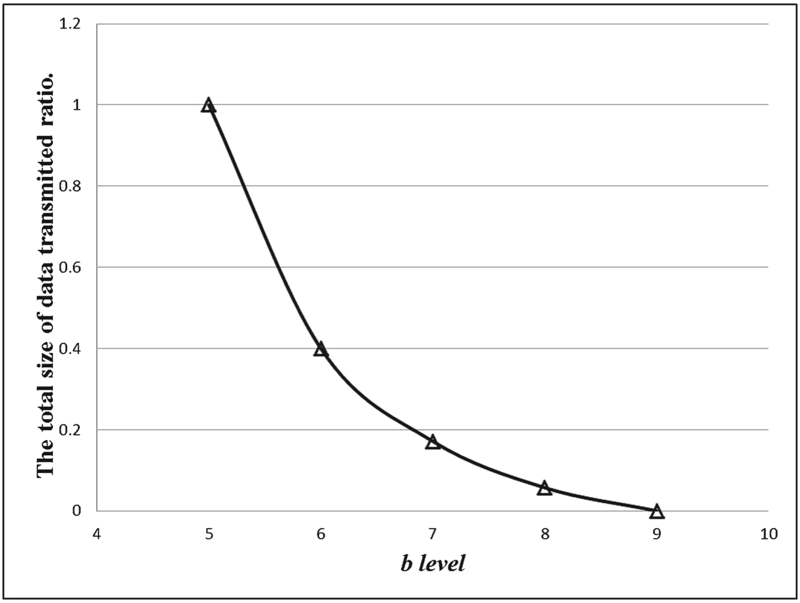


**Fig. 4.** Total size of data transmitted ($l = 5$).

Figure 3 shows the ratio of the total size of data transmitted at the aggregate level $l (1 \leq l \leq 8)$ to the level 8. Here, a broadcast level $b$ is five ($b = 5$). The higher the aggregate level $l$ is, the smaller the total size of data transmitted. Especially, if the aggregate level $l$ is larger than 5, the total size of data transmitted exponentially increases.

Figure 4 shows the ratio of the total size of data transmitted for the broadcast level $b$ ($b \geq l$) where $l = 5$. The higher the broadcast level $b$ is, the more volume of data is transmitted.

## 5 Concluding Remarks

In this paper, we proposed the modified model to efficiently realize the TWTBFC model. Here, one aggregate level $l$ is selected and aggregate nodes at the aggregate level $l$ collect output data of every aggregate node as the aggregate data $AD_l$. A target edge node is one whose actuators are activated for the aggregate data $AD_l$. A fog node whose descendant edge nodes are target ones is a relay node. Only target actuators have to be activated. The aggregate data $AD_l$ has to be delivered to only edge nodes of target actuators. In this paper, only a relay node forwards the aggregate data $AD_l$ to its relay nodes. In the evaluation, we showed the number of messages and energy consumption of nodes to exchange output data and forward aggregate data to descendant nodes can be reduced.

## References


1. Dl360p gen8. www8.hp.com/h20195/v2/getpdf.aspx/c04128242.pdf?ver=2
2. Raspberry pi 3 model b. https://www.raspberrypi.org/products/raspberry-pi-3-model-b/
3. Chida, R., Guo, Y., Oma, R., Nakamura, S., Duolikun, D., Enokido, T., Takizawa, M.: Implementation of fog nodes in the tree-based fog computing (TBFC) model of the IOT. In: Proceedings of the 7th International Conference on Emerging Internet, Data and Web Technologies (EIDWT-2019), pp. 92–102 (2019)
4. Creeger, M.: Cloud computing: an overview. Queue **7**(5), 3–4 (2009)
5. Enokido, T., Ailixier, A., Takizawa, M.: A model for reducing power consumption in peer-to-peer systems. IEEE Syst. J. **4**, 221–229 (2010)
6. Enokido, T., Ailixier, A., Takizawa, M.: Process allocation algorithms for saving power consumption in peer-to-peer systems. IEEE Trans. Ind. Electron. **58**(6), 2097–2105 (2011)
7. Enokido, T., Ailixier, A., Takizawa, M.: An extended simple power consumption model for selecting a server to perform computation type processes in digital ecosystems. IEEE Trans. Ind. Inf. **10**, 1627–1636 (2014)
8. Guo, Y., Oma, R., Nakamura, S., Duolikun, D., Enokido, T., Takizawa, M.: Data and subprocess transmission on the edge node of TWTBFC model. In: The 11th International Conference on Intelligent Networking and Collaborative Systems (INCoS-2019) (2019, Accepted)
9. Guo, Y., Oma, R., Nakamura, S., Duolikun, D., Enokido, T., Takizawa, M.: Evaluation of a two-way tree-based fog computing (TWTBFC) model. In: Proceedings of the 13th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2019), pp. 72–81 (2019)

10. Guo, Y., Oma, R., Nakamura, S., Duolikun, D., Enokido, T., Takizawa, M.: A two-way flow model for fog computing. In: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019), pp. 612–620 (2019)
11. Oma, R., Nakamura, S., Duolikun, D., Enokido, T., Takizawa, M.: An energy-efficient model for fog computing in the internet of things (IoT). Internet Things **1–2**, 14–26 (2018)
12. 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), pp. 99–109 (2018)
13. Oma, R., Nakamura, S., Duolikun, D., Enokido, T., Takizawa, M.: Evaluation of data and subprocess transmission strategies in the tree-based fog computing (TBFC) model. In: Proceedings of the 22nd International Conference on Network-Based Information Systems (NBiS-2019) (2019, Accepted)
14. Oma, R., Nakamura, S., Duolikun, D., Enokido, T., Takizawa, M.: A fault-tolerant tree-based fog computing model. Int. J. Web Grid Serv. (IJWGS) (2019, Accepted)
15. 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)
16. Rahmani, A.M., Liljeberg, P., Preden, J.S., Jantsch, A.: Fog Computing in the Internet of Things. Springer, Heidelberg (2018)