Optimal Agreement Achievement in a Fog Computing Based IoT

Shu-Ching Wang[1], Wei-Shu Hsiung [1], Kuo-Qin Yan[2], Yao-Te Tsai[3]
[1] Department of Information Management, Chaoyang University of Technology, Taiwan
[2] Department of Business Administration, Chaoyang University of Technology, Taiwan
[3] Department of International Business, Feng Chia University, Taiwan
{scwang, s10714902, kqyan}@cyut.edu.tw, yaottsai@fcu.edu.tw

Abstract

Since Fog computing is proposed to enable computing directly at the edge of the network, which can deliver new applications and services especially for the Internet of Things (IoT). In order to provide a high flexible and reliable platform of IoT, an IoT platform that combining Fog computing and Cloud computing is proposed in this study. In an IoT platform, the fault-tolerance is an important research topic. To cope with the influence from faulty components, reaching a common agreement at the presence of faults before performing some special tasks is essential. However, the previous protocols for the agreement problem of distributed computing are not enough for the IoT platform that combining Fog computing and Cloud computing. In this study, the agreement problem is revisited. The new proposed protocol can make all fault-free nodes reach agreement with minimal rounds of message exchanges and tolerate the maximal number of allowable faulty components in the IoT platform that combining Fog computing and Cloud computing.

Keywords: Internet of Things, Fog computing, Cloud computing, Interactive consistency problem, Consensus problem

1 Introduction

The Internet of Things (IoT) paradigm is based on intelligent and self-configuring nodes (things) interconnected in a dynamic and global network infrastructure. The IoT can make many applications, including consumer electronic devices, home appliances, medical devices, cameras, and all types of sensors. This innovation facilitates new interactions among things and humans, and enables the realization of smart cities, infrastructures, and services that enhance the quality of life. It represents the technologies, enabling ubiquitous and pervasive computing scenarios. IoT is generally characterized by real world and small things with limited storage and processing capacity. It is hard to avoid the circumstances of faulty behavior occurred in real world. Sometime, to make all fault-free nodes have a common value is very important. For instance, the initial time and the time stamps should be the same for all fault-free nodes in the system, otherwise, the distributed system may not be worked well. To reach an agreement is a part of reliability issue. To ensure that an IoT environment is reliable, a mechanism is provided in this study to make all fault free nodes reach an agreement and free from the influence of faulty nodes. Since, Cloud computing has virtually unlimited capabilities in terms of storage and processing power, hence has most of the IoT issues at least partially solved. Therefore, the IoT paradigm that combines the two technologies of cloud and Internet can provide current and future Internet [4].

Cloud computing is a great option, because the IoT demand significant compute and storage resources. Unfortunately, the requirements and design space of IoT make Cloud computing unfeasible in numerous scenarios, especially, when the goal is to build a general and multipurpose platform that can serve a wide variety of IoT applications [6]. According to the research by Yannuzzi et al., the main requirements for designing and building a scalable IoT platform include: (1) A platform for IoT must support rapid mobility patterns, even requiring in some cases high throughput on demand for short time periods. (2) A platform for IoT must support systems requiring reliable sensing, analysis, control and actuation, in scenarios subject to poor or unreliable connectivity to the Cloud and/or requiring very low latency. (3) A platform for IoT must be able to manage a large amount of geographically distributed "things" (either physical or virtual), which may produce data that require different levels of real time analytics and data aggregation [24]. Therefore, the recipe for building scalable IoT platforms is the following: (a) add Fog computing; (b) add Cloud computing and smartly combine it with the Fog, and (c) whenever the platform needs to scale either to cover more "things", just add more Fog nodes [24].

The emerging wave of Internet deployment, especially the IoT needs mobility support and geographical distribution, as well as location awareness and low latency. Bonomi et al. argue that Fog

*Corresponding Author: Yao-Te Tsai; E-mail: yaottsai@fcu.edu.tw
DOI: 10.3966/160792642019102006008

computing can meet these requirements [3]. Fog computing enables a new breed of applications and services, and that there is a fruitful interplay between the Cloud and the Fog, particularly when it comes to data management and analytics.

Fog computing is a distributed paradigm that provides Cloud-like services to the network edge. It leverages Cloud and edge resources along with its own infrastructure. In essence, the technology deals with IoT data locally by utilizing clients or edge devices near users to carry out a substantial amount of storage, communication, control, configuration, and management. The approach benefits from edge devices' close proximity to sensors, while leveraging the ondemand scalability of Cloud resources [6].

As network bandwidth and quality outstrip computer performance, various communication and computing technologies previously regarded as being of different domains can now be integrated, such as telecommunication, multimedia, information technology, and construction simulation [18]. Thus, applications associated with network integration have gradually attracted considerable attention. Similarly, IoT facilitated through distributed application over networks has also gained more recognition [12].

As the IoT has greatly encouraged distributed systems design and practiced to support user-oriented service applications [18]. However, distributed systems have grown rapidly in both size and number. In a distributed computing system, nodes allocated to different places or in separate units are connected together so that they collectively may be used to greater advantage. However, the network infrastructure and connectivity in IoT applications are becoming increasingly complex and heterogeneous, opening up many challenges including reliability [17]. In many cases, reaching a common agreement in the presence of faulty components is the central issue of fault-tolerant distributed computing, because many applications require such agreement [12]. For instance, the initial time and the time stamps should be the same or agree on for all nodes in the system, otherwise, the distributed system may not be worked well. Furthermore, many applications of IoT provide the convenience of users. For users, the system must provide better reliability and fluency [18]. Therefore, reliability is one of the most important aspects of IoT. To ensure that an IoT environment is reliable, a mechanism to allow a set of nodes to reach an agreed value is necessary.

In order to provide a high flexible and reliable platform of IoT, an IoT platform that combining Fog computing and Cloud computing (FC-IoT) will be proposed in this study. In the FC-IoT, numerous nodes are interconnected. Achieving agreement on a same value in the FC-IoT even if certain components fail, the protocols are required so that systems can still operate correctly.

Up to now, there have none related studies involving agreement issue in the FC-IoT. It is the first time a protocol is proposed to reach agreement underlying FC-IoT. In this study, the agreement is revisited with the assumption of nodes failure due to malicious faults in FC-IoT. The proposed protocol, FC Agreement Protocol (FCAP) of FC-IoT, can make all fault-free nodes reach agreement with minimal rounds of message exchanges, and tolerate the maximal number of allowable faulty components.

The rest of this paper is organized as follows. Section 2 will serve to introduce the related works. The IoT platform that combining Fog computing and Cloud computing (FC-IoT) is proposed in Section 3. Then, the proposed FC Agreement Protocol (FCAP) of FC-IoT will be brought up and illustrated in detail in Section 4. In Section 5, an example of executing the proposed protocol is given. Section 6 is responsible for proving the complexity of our new protocol. Finally, Section 7 gives conclusions of this research.

2 The Literature Review

Before the agreement problem can be solved, the related works must be discussed first. They are the agreement problems and the failure types of faulty components.

2.1 The Agreement Problems

In an IoT environment, a mechanism to allow a given set of nodes to agree on a common value is necessary for reliable smart city [11, 22]. Such a unanimity problem was called *Byzantine Agreement* (BA) [13]. It requires a number of independent nodes to reach agreement in cases where some of those nodes might be faulty. Namely, the goal of BA is making the fault-free nodes reach a common value.

The BA problem first studied by Lamport et al. is a well-known paradigm for the problem of achieving reliability in a distributed network of nodes. The definitions of the BA problem are [13]:

(1) There are n nodes ($n \geq 4$), of which at most one-third of the total number of nodes could fail without breaking down a workable network.

(2) The nodes communicate with each other through message exchange in a fully connected network.

(3) The message's sender is always identifiable by the receiver.

(4) A node is chosen as a source, and its initial value v_s is transmitted to other nodes for executing the protocol.

(5) The faulty component considered is node only.

Agreement is reached if all fault-free nodes agree on a common value. Based on these assumptions, various protocols for the BA problem have been developed in order to meet the following requirements [13]:

Agreement. All fault-free nodes agree on a common value v.

Validity. If the initial value of the source is v_s and the source is fault-free, then all fault-free nodes shall agree on the value vs; i.e., $v = v_s$.

The *consensus problem* [16] is extended from BA problem. The solutions of consensus problem are defined as protocols, which achieve a consensus and hope to use the minimum number of rounds of message exchanges to achieve the maximum number of allowable faulty capability. In this study, the solution of consensus problem is concerned in the Fog computing layer of FC-IoT. The definition of the problem is to make the fault-free nodes in the Fog computing layer of FC-IoT to reach consensus. Each Fog node of Fog computing layer chooses an initial value to start with, and communicates to each other by exchanging messages. The Fog nodes are referred to make a consensus if it satisfies the following conditions [16]:

Consensus. All fault-free Fog nodes agree on a common value.

Validity. If the initial value of each fault-free Fog node n_i is v_i then all fault-free Fog nodes shall agree on the value v_i.

A closely related sub-problem, the *interactive consistency problem* (IC problem) has been studied extensively [9]. In this study, the solution of IC problem is concerned in the Cloud computing layer of FC-IoT. The definition of IC problem is to make the fault-free Cloud nodes in the Cloud computing layer reach interactive consistency. Each Cloud node chooses an initial value and communications with the others by exchanging messages. There is interactive consistency in that each Cloud node i has its initial value v_i and agrees on a set of common values. Therefore, interactive consistency has been achieved if the following conditions are met [9]:

Consistency. Each fault-free Cloud node agrees on a common vector $V = [v_1, v_2, \ldots, v_n]$.

Validity. If the initial value of fault-free Cloud node i is v_i, then the i-th value in the common vector V should be v_i.

In previous studies, the BA algorithms were designed in traditional network topology [1, 9, 13, 19-21, 23]. Those works reach BA underlying different topologies respectively, including Broadcasting Network (BCN), Fully Connected Network (FCN), Multicasting Network (MCN), Wireless Sensor Network (WSN), and Cloud Computing environment (CC). All those previous protocols are not suitable for FC-IoT due to the difference of the network topology. The IoT environment is an Internet-based development. It is a style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet. Nevertheless, in an IoT environment, the connected topology is not very significant. In this study, the consensus problem is to be solved on the Fog computing layer and IC problem is to be solved on the Cloud computing layer of the proposed IoT platform that combining Fog computing and Cloud computing. And, the proposed protocol, is named FC Agreement Protocol (in short FCAP) that can use a minimum number of message exchanges and can tolerate a maximum number of allowable faulty components to make each fault-free node reach a common agreement in the cases of node failure.

2.2 Failure Types

In a distributed system, the network components may not always work well. A node is said to be fault-free if it follows protocol specifications during the execution of a protocol; otherwise, the node is said to be faulty. The symptoms of node failure can be classified into two categories. There are dormant fault and malicious fault (also called as the Byzantine fault) [13]. The dormant faults of nodes include crashes and omission. A crash fault happens when a node is broken. An omission fault takes place when a node fails to transmit or receive a message on time or at all. On a malicious fault, the behavior of a faulty node is unpredictable and arbitrary. The message transmitted by a malicious faulty node is random or arbitrary. It is the most damaging failure type and causes the worst problem. That is, if the agreement problem can be solved in a malicious fault case, then the agreement problem can also be solved in other failure mode.

Therefore, in this study, malicious faults are investigated, and the means by which fault-free nodes may reach agreement in the FC-IoT platform are explored. In addition, Bousbiba and Klaus also indicate that BA in an asynchronous network is impossible even if only one crash faulty node [3]. Hence, the assumption of underlying FC-IoT is synchronous.

3 The Network Structure

With the advancement and development of various technologies, computing problems become more complicated and larger [18]. A Cloud computing environment allows a user faster operation of Internet applications. The majority of Cloud computing infrastructure consists of reliable services delivered through data centers and built on servers with different levels of virtualization technologies [14]. The services are accessible anywhere that has access to networking infrastructure. Commercial offerings must meet the quality of service requirements of customers, and typically offer service-level agreements [18]. Therefore, a distributed system must be having high stability to handle instances where many users utilize a given environment. In this section, the proposed IoT platform is discussed.

In order to provide a high flexible and reliable platform of IoT, an IoT platform that combining Fog computing and Cloud computing (FC-IoT) is proposed in this study. The topology of FC-IoT is shown in

Figure 1. There are three layers in the FC-IoT: *IoT sensors layer*, *Fog computing layer* and *Cloud computing layer*. The IoT sensors layer is consisted by sensor nodes, which is responsible for sensing the data required by the IoT application. The Fog computing layer is constructed by Fog groups; each Fog group is composed of a large number of Fog nodes, responsible for the processing of specific information and judgments. The Cloud computing layer is made up of many Cloud nodes, which provide Cloud users' services.

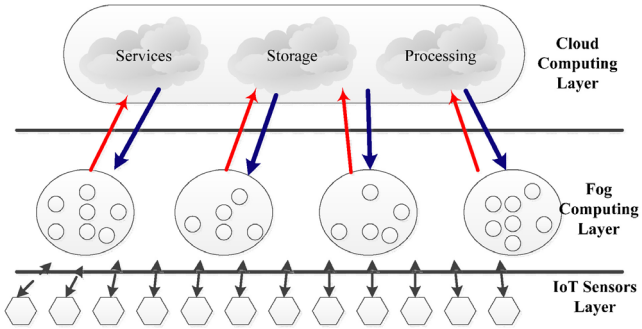


Figure 1. The topology of FC-IoT

In the IoT environment, through the combination of a large number of sensors, various types of sensing data in real life can be collected. These huge sensing data from all over are used, and then a wide range of application services can be provided. For example, FC-IoT proposed in this study can be used in the monitoring system for prevention of earth-rock-flow disaster. At FC-IoT, the sensed data of the sensors in different regions are sent to the corresponding Fog groups in the Fog computing layer, and the data are processed by the Fog nodes in the specific Fog group. The related monitoring information of different regions is collected by each Fog group, and then the collected information is analyzed and judged in each Fog group. Finally, the status of different regions is transmitted to the disaster prevention center at the Cloud computing layer. Figure 2 is an example of the monitoring system for prevention of earth-rock-flow disaster constructed by FC-IoT.

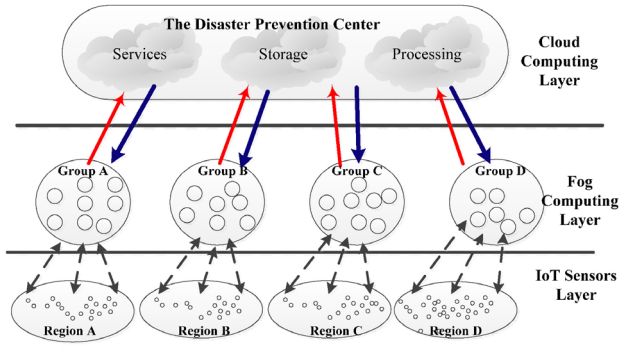


Figure 2. An example of the monitoring system for prevention of earth-rock-flow disaster constructed by FC-IoT

Recently, VANET becomes increasingly popular in many countries. It is an important element of the Intelligent Transportation Systems (ITSs) [15]. Vehicles and roadside equipment can form a VANET and communicate with each other via wireless and multi-hop communications. When the traffic control system of ITS is constructed by IoT, then the connecting vehicles are used to get the data required by the ITS [8]. The VANET environment contains numerous challenges for communication, many of which can be addressed by a clustered network [10]. A cluster-based VANET consists of a set of loosely or tightly connected nodes that work together so that, in many respects, they can be viewed as a single system. For example, the nodes in a cluster at the same traffic intersection can detect the status of traffic is smooth, with lots of traffic or traffic congestion. When the traffic control system is constructed by FC-IoT, then the IoT sensors layer is used to sense the data required by the ITS. The Fog computing layer is used to catch the traffic status of each intersection of road. The Cloud computing layer is used to provide the services of traffic control. An example of the traffic control system constructed by FC-IoT is shown in Figure 3.

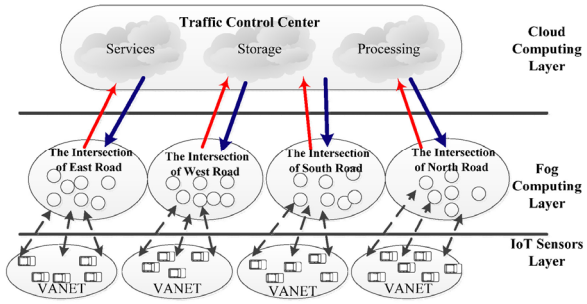


Figure 3. An example of the traffic control system constructed by FC-IoT

In short, the FC-IoT is proposed by Fog computing, where data can be analyzed and processed by devices in the network rather than being centralized in the Cloud computing. By coordinating and managing the computing and storage resources at the edge of the network, more and more connected devices and the emerging needs of IoT can be processed by the Fog computing. When the technological requirements and constraints of the IoT applications are properly fulfilled, it is up to the platform designer to decide whether an endpoint should be served by the Cloud, the Fog, or an adequate combination of the two at any given time during the service lifetime [24]. According to the above features, the Fog computing can be made as an appropriate platform for providing the critical services and applications of IoT, including connected vehicle, smart grid, and wireless sensor and actuator networks, ... and so on [24].

4 The Proposed Protocol

In this study, the agreement problem is discussed in an IoT platform that combining Fog computing and Cloud computing (FC-IoT), there is no delay of nodes or communication media is included in our discussion. Therefore, the nodes executing our new protocol should receive the messages from other nodes within a predictable time period. If the message is not received on time, the message must have been influenced by faulty components.

In this research, FCAP is used to solve the agreement problem in an FC-IoT with malicious fallible nodes. With consideration for efficient agreement, the nodes of the sensors layer in IoT is used to detect the required data of a specific IoT application, the *Consensus* is applied to the Fog nodes of Fog computing layer, and the *Interactive Consistency* is applied to each Cloud node in Cloud computing layer.

In the agreement problem, the number of faulty components can be allowed is determined by the total number of nodes. In Lamport et al.'s protocol [13], the constraints is $n>3f$ where n is the number of nodes and f is the total number of allowable malicious faulty nodes in the distributed system. Therefore, the constraints of the FCAP are shown in follow.

Constraint of IoT sensors layer. $n_{Rj}\geq\lfloor(n_{Fj}-1)/2\rfloor+f_{mRj}$ where n_{Rj} is the number of sensor nodes and f_{mRj} is the total number of allowable malicious faulty sensor nodes in Region R_j of IoT sensor layer. This constraint specifies the number of sensor nodes required in Region R_j of IoT sensor layer.

Constraint of Fog computing layer. $n_{Fj}\geq\lfloor(n_{Fj}-1)/3\rfloor+2f_{mFj}$ where n_{Fj} is the number of Fog nodes and f_{mFj} is the total number of allowable malicious faulty Fog nodes in Fog group F_j of Fog computing layer. This constraint specifies the number of Fog nodes required in Fog group F_j of Fog computing layer.

Constraint of Cloud computing layer. $n_C\geq\lfloor(n_C-1)/3\rfloor+2f_{mC}$ where n_C is the number of Cloud nodes and f_{mC} is the total number of allowable malicious faulty Cloud nodes in Cloud computing layer. This constraint specifies the number of Cloud nodes required in Cloud computing layer.

(Constraint of IoT sensors layer) specifies the number of sensor nodes in IoT sensors layer required; due to the unit of the Region R_j of IoT sensor layer is sensor node, so that an agreement can be achieved if $n_{Rj}\geq\lfloor(n_{Fj}-1)/2\rfloor+f_{mRj}$. (Constraint of Fog computing layer) specifies the number of Fog nodes in Fog group F_j of Fog computing layer required; due to the unit of the Fog group F_j of Fog computing layer is Fog node, so that an agreement can be achieved if $n_{Fj}\geq\lfloor(n_{Fj}-1)/3\rfloor+2f_{mFj}$. (Constraint of Cloud computing layer) specifies the number of Cloud nodes in Cloud computing layer required; due to the unit of the Cloud computing layer is Cloud node, so that an agreement can be achieved if $n_C\geq\lfloor(n_C-1)/3\rfloor+2f_{mC}$.

In this study, FCAP is proposed to solve the agreement problem with fallible nodes underlying an IoT platform that combining Fog computing and Cloud computing (FC-IoT). The proposed protocol FCAP is divided into three parts based on the three layers of FC-IoT. The nodes of IoT sensor layer execute *Sensing and Transmission Process,* the nodes of Fog computing layer execute *Consensus Process*, and the nodes of Cloud computing layer execute *Interactive Consistency Process.*

In *Sensing and Transmission Process*, the sensor node senses the related information for the specific application service in a particular region. Then, the related information for the specific application service is transferred to the corresponding Fog group of Fog computing layer. In *Consensus Process*, the Fog node takes the majority value of the sensing data received from IoT sensors layer firstly, and the majority value is used as the initial value (v_i) of Fog node to execute function *Agreement*. When the Consensus value of each Fog group is gotten, the value is represented as the result of a specific service in a particular region. Finally, the Consensus value is transferred to Cloud computing layer. In *Interactive Consistency Process*, the primary work of nodes in Cloud computing layer is to collect the results of a specific service in different regions, and then the request vector of the interactive consistency can be obtained to provide an integrated service such as the intelligent traffic controller or the monitoring system for prevention of earth-rock-flow disaster. The progression steps of FCAP are shown in Figure 4.

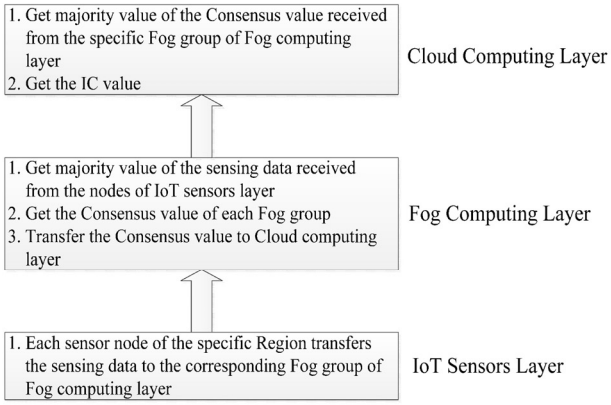


Figure 4. The progression steps of FCAP

FCAP is initiated by the nodes of IoT sensor layer to get the related information for the specific application service. The nodes of Fog computing layer need to execute *Consensus Process* and the nodes of Cloud computing layer need to execute *Interactive Consistency Process*. In the *Consensus Process* and *Interactive Consistency Process*, the function *Agreement* will be called.

There are two phases of function *Agreement*, one is

the *Message Exchange Phase*, and the other is *Decision Making Phase*. The parameters of *Agreement* include σ, v_s, and n_A, where σ is the required rounds, v_s is the initial value and n_A is the number of nodes participating in the agreement. In order for all fault-free nodes to reach agreement, each node must collect enough exchanged messages from all other nodes if they are fault-free. As a result, exchanging the received values helps fault-free nodes to collect enough exchanged messages.

Fischer and Lynch proved that $\lfloor(n\text{-}1)/3\rfloor+1$ is the necessary and sufficient rounds of message exchanges to solve an agreement problem, where n is the number of nodes in the underlying network [9]. Based on the works of Fischer and Lynch, $\lfloor(n\text{-}1)/3\rfloor+1$ rounds of message exchanges are the lower bound for solving the agreement problem [9]. Therefore, the required rounds σ is $\lfloor(n_{Fj}-1)/3\rfloor+1$ when Fog nodes execute the function *Agreement* of *Consensus Process*, where n_{Fj} is the number of nodes in Fog group F_j of Fog computing layer and $n_{Fj}>3$. And, the required rounds σ is $\lfloor(n_C-1)/3\rfloor+1$ when Cloud nodes execute the function *Agreement* of *Interactive Consistency Process*, where n_C is the number of nodes in Cloud computing layer and $n_C>3$.

The received messages of *Message Exchange Phase* are stored in a tree structure called the information-gathering tree (ig-tree), which is similar to that proposed by Bar-Noy et al. [2]. Each fault-free node maintains such an ig-tree during the execution of FCAP. In the first round of *Message Exchange Phase*, node i transmits its initial value to other nodes. However, each receiver node could always identify the sender of a message is assumed. When a fault-free node receives the message sent from the node i, it stores the received value, denoted as *val*(i), at the root of its ig-tree. In the second round, each node transmits root value of its ig-tree to all other nodes. If node *1* sends message *val*(i) to node *2*, then node *2* stores the received message, denoted as *val*(*i1*), in vertex *i1* of its ig-tree. Similarly, if node *2* sends message *val*(*i1*) to node *1*, the received message is named *val*(*i12*) and stored in vertex *i12* of node *1*'s ig-tree in the third round. Generally, message *val*(*i12…n*), stored in the vertex *i12…n* of an ig-tree, implies that the message just received was sent through the node *i*, the node *1*,…, the node *n*; and the node *n* is the latest nodes to pass the message. When a message is transmitted through a node more than once, the name of the node will also be repeated correspondingly. For instance, message *val*(*11*), stored in vertex *11*, and indicates that the message is sent to node *1*, then to node *1* again; therefore name *1* appears twice in vertex name *11*. In summary, the root of ig-tree is always named *i* to denote that the stored message is sent from the node *i* in the first round; and the vertex of an ig-tree is labeled by a list of node names. The node name list contains the names of the nodes through which the stored message was transferred. Figure 5 shows an example of ig-tree. In the Message Exchange Phase of function Agreement, the vertices with repeated node names in each ig-tree will be deleted. Finally, all fault-free nodes use function VOTE to remove the faulty influence from faulty nodes to obtain the common value. Among them, the function VOTE only calculates the non-value "α" of all the vertices of the α-th level of the ig-tree (excluding the last level of the ig-tree), where $1\leq\alpha\leq$σ. Since VOTE(α) is a common value, the impact of faulty nodes will be removed and each fault-free node can reach an agreed value. When the function VOTE is applied to the root of each corresponding ig-tree, and then the common value VOTE(i) is obtained. The proposed protocol FCAP is presented in Figure 6.

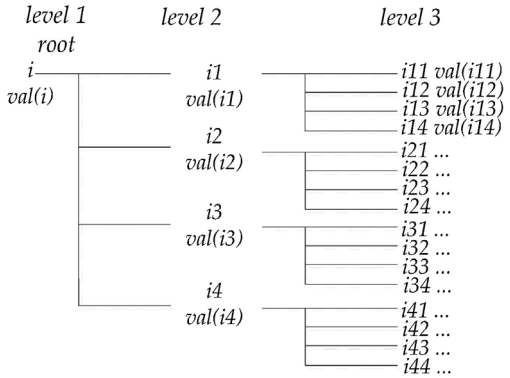


Figure 5. An example of ig-tree

5 An Example of Executing FCAP

Taking the traffic control system constructed by FC-IoT as an example to execute FCAP is presented in Figure 7. In *Sensing and Transmission Process*, each sensor node in the IoT sensor layer senses the traffic status. The sensing data of each node in the Region R_1 of IoT sensor layer is shown in Figure 7(a), and nodes s_{12} and s_{15} are assumed in malicious fault. The sensing traffic statuses of the specific road intersection in Region R_1 are transferred to Fog group F_1 of Fog computing layer.

In *Consensus Process*, each Fog node in Fog group F_1 receives the sensing traffic statuses transferred from sensor nodes in the Region R_1. The received traffic statuses are taken as the majority and the majority value is used as the initial value (v_i) of Fog node in Fog group F_1 when function *Agreement* is executed. Since nodes s_{12} and s_{15} are malicious faulty nodes, it is assumed that the traffic status they transmit is malicious. However, as long as the total number of failed nodes does not exceed half of the total number of nodes in Region R_1, the majority value obtained is still the correct values. Then, the number of rounds required, σ = $\lfloor(n_{Fj}-1)/3\rfloor+1$, is computed and *Agreement*(σ, v_i, n_{Fj}) is executed. The initial value of each node in Fog group F_1 of Fog computing layer is shown in Figure 7(b).

FC Agreement Protocol (FCAP)

The nodes of IoT sensor layer execute *Sensing and Transmission Process*.
The nodes of Fog computing layer execute *Consensus Process*.
The nodes of Cloud computing layer execute *Interactive Consistency Process*.

Sensing and Transmission Process (for the sensor node s_{ij} in the Region R_j of IoT sensor layer, $1 \le i \le n_{Rj}$ where n_{Rj} is the number of sensor nodes in Region R_j of IoT sensor layer)

1. The sensor node s_{ij} senses the related information for the specific application service.
2. The related information for the specific application service is transferred to the corresponding Fog group of Fog computing layer by sensor node s_{ij}.

Consensus Process (for the node f_{ij} in the Fog group F_j of Fog computing layer, $1 \le i \le n_{Fj}$ where n_{Fj} is the number of nodes in Fog group F_j of Fog computing layer and $n_{Fj} > 3$)

1. The node f_{ij} receives the sensing information transferred from sensor nodes in the Region R_j of IoT sensor layer.
2. The received information from sensor nodes in the Region R_j are taken as the majority. In addition, the majority value is used as the initial value (v_i) of f_{ij} when function *Agreement* is executed.
3. Compute the number of rounds required, $\sigma = \lfloor (n_{Fj}-1)/3 \rfloor + 1$. Execute *Agreement*($\sigma$, v_i, n_{Fj}), then the agreement vector of Region R_j is obtained.
4. Take the majority value of the agreement vector, and then the Consensus value is obtained.
5. The Consensus value is transferred to Cloud computing layer.

Interactive Consistency Process (for the node c_i in the Cloud computing layer, $1 \le j \le n_C$, where n_C is the number of nodes in Cloud computing layer and $n_C > 3$)

1. The node c_j receives the Consensus values transferred from nodes in the Fog group F_j of Fog computing layer.
2. The received Consensus values from nodes in the Fog group F_j of Fog Computing layer are taken as the majority. Moreover, the majority value is used as the initial value (v_j) of c_j when function *Agreement* is executed.
3. Compute the number of rounds required, $\sigma = \lfloor (n_C-1)/3 \rfloor + 1$. Execute *Agreement*($\sigma$, v_j, n_C), then the agreement vector is obtained.
4. The obtained vector is IC value.

Agreement(σ, v_s, n_A) (σ is the required rounds, v_s is the initial value and n_A is the number of nodes participating in the agreement)

Message Exchange Phase:

If $r = 1$ then:
1) Each node broadcasts its initial value v_s to other nodes in the same cluster.
2) Each node receives and stores the n_A values sent from n_A nodes of same cluster in the corresponding root of its ig-tree.

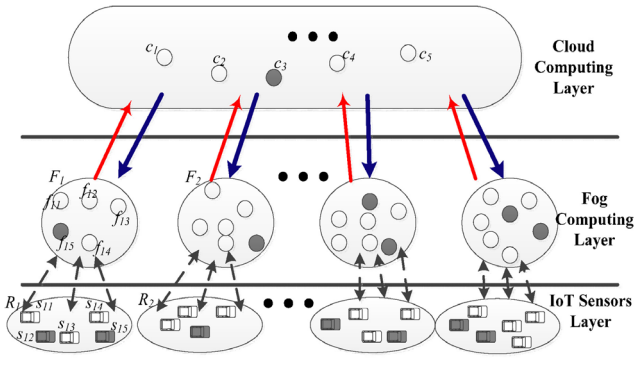
For $1 < r \le \sigma$, do:
1) Each node transmits the values at level $r-1$ in its ig-tree to other nodes in same cluster.
2) Each receiver node stores the received values in the corresponding vertices at level r of its ig-tree.

Decision Making Phase:

Step 1: Reorganize each ig-tree by deleting the vertices with repeated node names.
Step 2: Using function *VOTE* with the root i of each node's ig-tree and obtaining the common value *VOTE*(i).

VOTE(α)= If the α is a leaf
then outputs the value *val*(α); else
If the majority value in the set of {VOTE(αi)|$1 \le i \le n_A$ and vertex αi is a child of vertex α} exists
then outputs the majority value; else
a default value φ is outputted.

Figure 6. Protocol FCAP



IoT Sensor Layer

s_{11}	s_{12}	s_{13}	s_{14}	s_{15}
1	0	1	1	0

Figure 7. (a) The sensing data of each node in the Region R_1 of IoT sensor layer

For this example, two rounds ($\sigma = \lfloor (n_{F1}-1)/3 \rfloor + 1 = \lfloor (5-1)/3 \rfloor + 1 = 2$, where n_{F1} is the number of nodes in Fog group F_1) are required to exchange the messages when *Agreement* is executed. In this example, there are five nodes in Fog group F_1 and Fog node f_{15} is assumed in malicious fault. Figure 7(b) is the initial value of each node in Fog group F_1. During the first round of *Message Exchange Phase*, each node of Fog group F_1 parallel transmits the initial value to all nodes of Fog group F_1 and stores the received n_{F1} (=5) values in the

corresponding root of each ig-tree, as shown in Figure 7(c). In the second round, each node parallel transmits the values in the root of the corresponding ig-tree to other nodes in Fog group F_1 and stores the received values in level 1 of the n_{F1} (=5) corresponding ig-trees. The progression of nodes f_{11} and f_{13} during *Message Exchange Phase i*s shown in Figsure 7(d) and 7(f). Subsequently, in the *Decision Making Phase*, the ig-tree is reorganized by deleting those vertices with repeated node names. The corresponding ig-tree of nodes f_{11} and f_{13} is shown in Figure 7(e) and 7(g). Then, function VOTE is applied on the ig-tree root of each node to take the majority value. The majority value of the agreement vector is taken, and the Consensus value is obtained. The Consensus value of nodes f_{11} and f_{13} is obtained and shown in Figure 7(h). The Consensus value of each Fog group in the Fog computing layer represents the traffic state of each region. Finally, the Consensus value is transferred to the Cloud computing layer.

Fog Computing Layer

Majority(1,*0*,1,1,*0*)=1

f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
1	1	1	1	*0*

$\sigma = \lfloor (n_{Fj}-1)/3 \rfloor + 1 = \lfloor (5-1)/3 \rfloor + 1 = 2$

Execute *Agreement*(σ, v_i, n_{Fj})=(2,1,5)

Figure 7. (b) The initial value of each node in Fog group F_1 of Fog computing layer

f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
1	1	1	1	*0*

$\sigma = \lfloor (n_{Fj}-1)/3 \rfloor + 1 = \lfloor (5-1)/3 \rfloor + 1 = 2$

Figure 7. (c) The ig-tree of each node in Fog group F_1 of Fog computing layer at the first round of *Message Exchange Phase*

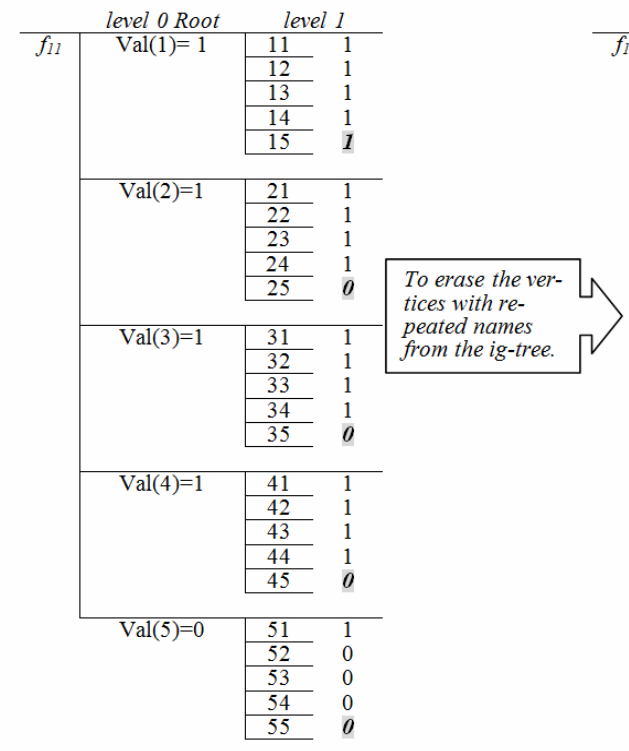


Figure 7. (d) The final ig-tree of f_{11} after the *Message Exchange Phase*

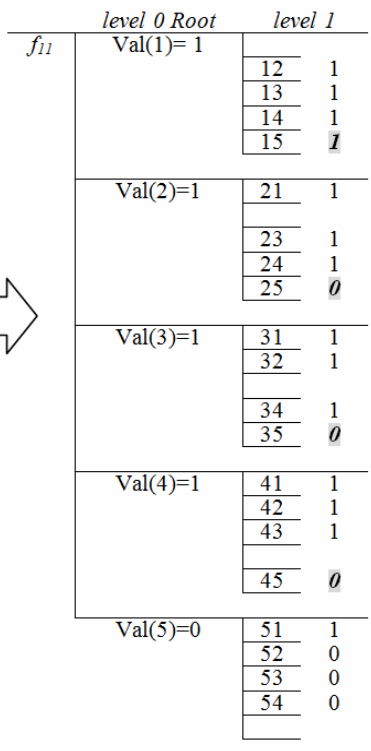


Figure 7. (e) The ig-tree of f_{11} by *Decision Making Phase*

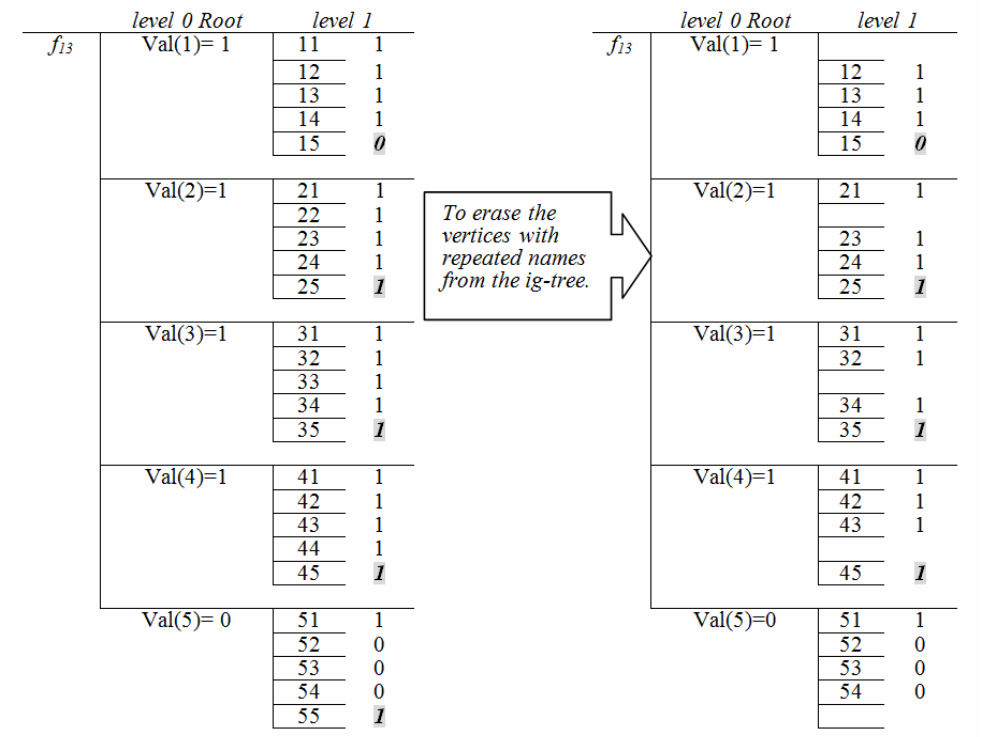


Figure 7. (f) The final mg-tree of f_{13} after the *Message Exchange Phase*

Figure 7. (g) The ig-tree of f_{13} by *Decision Making Phase*

VOTE(1)= majority(*val*(12),*val*(13),*val*(14),*val*(15))=majority(1, 1, 1, 1)=1
VOTE(2)= majority(*val*(21),*val*(23),*val*(24),*val*(25))=majority(1, 1, 1, 0)=1
VOTE(3)= majority(*val*(31),*val*(32),*val*(34),*val*(35))=majority(1, 1, 1, 0)=1
VOTE(4)= majority(*val*(41),*val*(42),*val*(43),*val*(45))=majority(1, 1, 1, 0)=1
VOTE(5)= majority(*val*(51),*val*(52),*val*(53),*val*(54))=majority(1, 0, 0, 0)=0
Consensus value of f_{11} =(1,1,1,1,0)=1
VOTE(1)= majority(*val*(12),*val*(13),*val*(14),*val*(15))=majority(1, 1, 1, 0)=1
VOTE(2)= majority(*val*(21),*val*(23),*val*(24),*val*(25))=majority(1, 1, 1, 1)=1
VOTE(3)= majority(*val*(31),*val*(32),*val*(34),*val*(35))=majority(1, 1, 1, 1)=1
VOTE(4)= majority(*val*(41),*val*(42),*val*(43),*val*(45))=majority(1, 1, 1, 1)=1
VOTE(5)= majority(*val*(51),*val*(52),*val*(53),*val*(54))=majority(1, 0, 0, 0)=0
Consensus value of f_{13} =(1,1,1,1,0)=1

Figure 7. (h) The common value VOTE(*i*) by in *Decision Making Phase* of Fog computing layer

In the *Interactive Consistency Process*, the Cloud node in the Cloud computing layer receives the Consensus value sent from Fog nodes in the Fog group of Fog computing layer. The received Consensus values are taken as the majority. In addition, the majority value is used as the initial value of Cloud node when function *Agreement* is executed. The initial value of each node in Cloud computing layer is shown in Figure 7(i).

For this example, two rounds ($\sigma = \lfloor (n_C-1)/3 \rfloor + 1 + 1 = \lfloor (5-1)/3 \rfloor + 1 = 2$, where n_C is the number of nodes in Cloud computing layer) are required to execute *Agreement*. In this example, there are five nodes in Cloud computing layer and Cloud node c_3 is assumed in malicious fault. Figure 7(i) is the initial value of each node in Cloud computing layer. During the first round of *Message Exchange Phase*, each node of Cloud computing layer parallel transmits the initial value to all nodes of Cloud computing layer and stores the received n_C (=5) values in the corresponding root of each ig-tree, as shown in Figure 7(j). In the second round, each node parallel transmits the values in the root of the corresponding ig-tree to other nodes in Cloud computing layer and stores the received values in level 1 of the n_C (=5) corresponding ig-trees. The progression of nodes c_1 and c_4 during *Message Exchange Phase i*s shown in Figs. 7(k) and 7(m). Subsequently, in the *Decision Making Phase*, the ig-tree is reorganized and the corresponding ig-tree of nodes c_1 and c_4 is shown in Figs. 7(l) and 7(n). Then, function VOTE is applied on the ig-tree root of each node to take the majority value.

Cloud Computing Layer

Majority(1,1,1,1, *0*)=1

c_1	c_2	c_3	c_4	c_5
1	1	*0*	1	1

$\sigma = \lfloor (n_C - 1)/3 \rfloor + 1 = \lfloor (5-1)/3 \rfloor + 1 = 2$

Execute *Agreement*(σ, v_j, n_C)=(2,1,5)

Figure 7. (i) The initial value of each node in Cloud computing layer

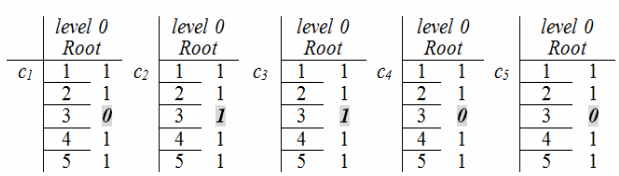


Figure 7. (j) The ig-tree of each node in Cloud computing layer at the first round of *Message Exchange Phase*

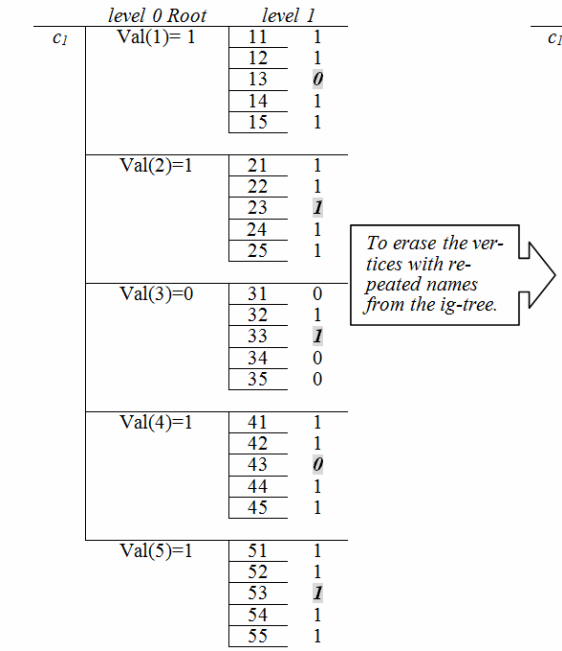


Figure 7. (k) The final ig-tree of c_1 after the *Message Exchange Phase*

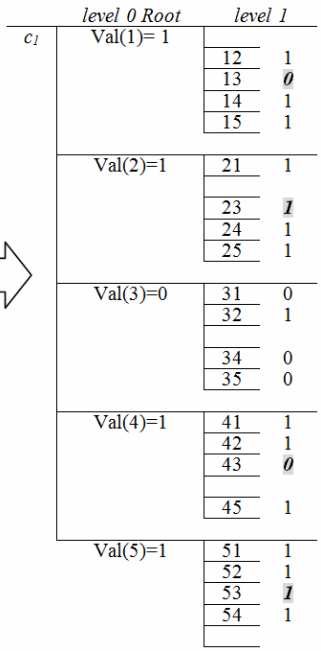


Figure 7. (l) The ig-tree of c_1 by *Decision Making Phase*

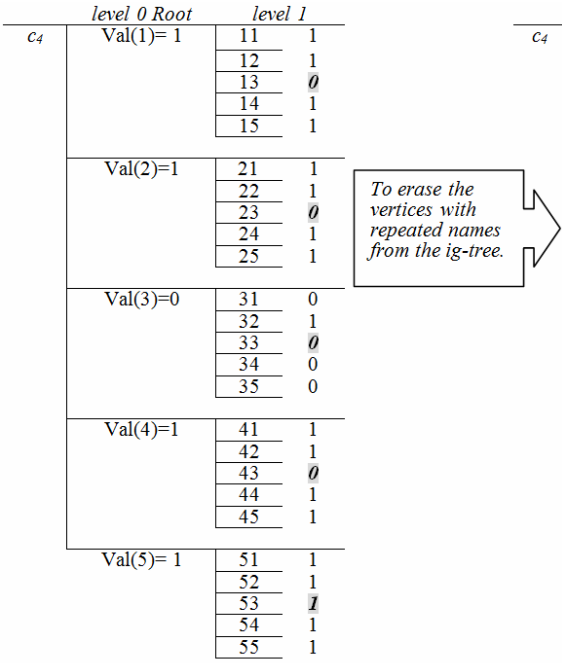


Figure 7. (m) The final mg-tree of c_4 after the *Message Exchange Phase*

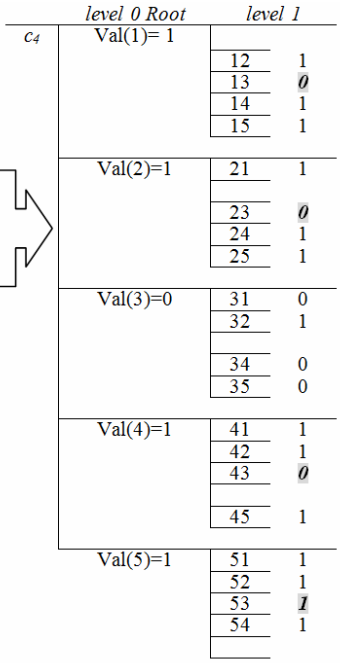


Figure 7. (n) The ig-tree of c_4 by *Decision Making Phase*

The majority value obtained through function *Agreement* is mapped to a traffic status at the specific traffic intersection. The IC value is a vector, and each element in the vector is the majority value obtained through *Agreement* function. Each element is used to present the traffic status of a specific traffic intersection. The IC value of nodes c_1 and c_4 is shown in Figure 7(o). Eventually, the agreement is reached in FC-IoT. Finally, the service of traffic control system can be supported by each Cloud node in Cloud computing layer.

VOTE(1)= majority(*val*(12),*val*(13),*val*(14),*val*(15))=majority(1, 0, 1, 1)=1
VOTE(2)= majority(*val*(21),*val*(23),*val*(24),*val*(25))=majority(1, 1, 1, 1)=1
VOTE(3)= majority(*val*(31),*val*(32),*val*(34),*val*(35))=majority(0, 1, 0, 0)=0
VOTE(4)= majority(*val*(41),*val*(42),*val*(43),*val*(45))=majority(1, 1, 0, 1)=1
VOTE(5)= majority(*val*(51),*val*(52),*val*(53),*val*(54))=majority(1, 1, 1, 1)=1
IC_{c1}=(1,1,0,1, 1)
VOTE(1)= majority(*val*(12),*val*(13),*val*(14),*val*(15))=majority(1, 0, 1, 1)=1
VOTE(2)= majority(*val*(21),*val*(23),*val*(24),*val*(25))=majority(1, 0, 1, 1)=1
VOTE(3)= majority(*val*(31),*val*(32),*val*(34),*val*(35))=majority(0, 1, 0, 0)=0
VOTE(4)= majority(*val*(41),*val*(42),*val*(43),*val*(45))=majority(1, 1, 0, 1)=`
VOTE(5)= majority(*val*(51),*val*(52),*val*(53),*val*(54))=majority(1, 1, 1, 1)=`
IC_{c4}=(1,1,0,1,1)

Figure 7. (o) The common value VOTE(*i*) in *Decision Making g Phase* of Cloud computing layer

Figure 7. The example of executing the FCAP

6 The Complexity of FCAP

The following theorems are used to prove the complexity of FCAP. The complexity of FCAP is evaluated in terms of (1) the minimal number of rounds of message exchanges, and (2) the maximum number of allowable faulty nodes. Theorems 1 and 2 below will show that the optimal solution is reached.

Theorem 1. The number of required rounds of message exchanges by FCAP is the minimum.

Proof. The total number of required rounds of message exchanges by FCAP can be discussed by three layer of FC-IoT.

(1) **IoT sensor layer**. In IoT sensor layer, each sensor passes the received sensing data to Fog computing layer. Therefore, only one round of message exchange is needed.

(2) **Fog computing layer**. Because message passing is required only in the *Message Exchange Phase*, the *Message Exchange Phase* is time consuming. Dolev and Reischuk pointed out that $\lfloor(n-1)/3\rfloor+1$ rounds are the minimum number of rounds to send sufficient messages to achieve agreement in an n-node fallible distributed system [7]. However, in the fallible Fog computing layer, the nodes maybe in malicious fault. In addition, each node in the fallible Fog computing layer must exchange messages with other nodes. Therefore, a constraint on the minimum number of rounds can be applied to the study. In other words, in Fog computing layer, there are n_{Fj} nodes in Fog group F_j of Fog computing layer, FCAP needs $\lfloor(n_{Fj}-1)/3\rfloor+1$ rounds to exchange messages. In an F-groups Fog computing layer, the nodes in each Fog group execute FCAP parallel, where F is the total number of groups in the Fog computing layer of FC-IoT. Therefore, the required rounds of executing FCAP by each node in all Fog groups are depended on the number of nodes in Fog group.

(3) **Cloud computing layer**. As in the discussion of the number of message exchanges required in the Fog computing layer. In the Cloud computing layer, the research of Dolev and Reischuk can still be applied [7]. In Cloud computing layer, there are n_C nodes in Cloud computing layer, FCAP needs $\lfloor(n_C-1)/3\rfloor+1$ rounds to exchange messages.

In short, number of required rounds of message exchanges by FCAP in FC-IoT is the minimum.

Theorem 2. The number of allowable faulty nodes by FCAP is the maximum.

Proof. The total number of allowable faulty nodes by FCAP can be discussed by three layer of FC-IoT.

(1) **IoT sensor layer**. Since the number of faulty nodes in each Region of IoT sensor layer does not exceed half, and the majority value of the Region can be determined. Therefore, T_{FS} be the total number of allowable faulty nodes in IoT sensor layer. $T_{FS}=\sum_{j=1}^{R} f_{mRj}$ where R is the total number of Regions in IoT sensor layer and f_{mRj} is the total number of allowable malicious faulty sensor nodes in Region R_j. In addition, $f_{mRj}\leq\lfloor(n_{Rj}-1)/2\rfloor$ where n_{Rj} is the number of sensor nodes in Region R_j.

(2) **Fog computing layer**. Fischer and Lynch indicate the lower bound for agreement problem for node faults as $f\leq\lfloor(n-1)/3\rfloor$, where f is the total number of allowable malicious faulty nodes and n is the total number of nodes in a distributed computing system [9]. However, the fault status of our assumption is also that nodes are faulty. Therefore, $f\leq\lfloor(n-1)/3\rfloor$ in the study of Fischer and Lynch [9] can be applied to $f_{mFj}\leq\lfloor(n_{Fj}$ -

1)/3⌋ in the Fog computing layer, where f_{mFj} is the total number of allowable malicious faulty Fog nodes in Fog group F_j and n_{Fj} is the number of Fog nodes in Fog group F_j. Then, $T_{FF}= \sum_{j=1}^{F} f_{mFj}$ where F is the total number of Fog groups in the Fog computing layer of FC-IoT, and T_{FF} is the total number of allowable faulty nodes in Fog computing layer.

(3) **Cloud computing layer**. The research result of Fischer and Lynch [9] also can be applied to Cloud computing layer. Therefore, f_{mC} is the total number of allowable faulty nodes in Cloud computing layer, and $f_{mC} \leq \lfloor (n_C\text{-}1)/3 \rfloor$ where n_C is the number of Cloud nodes.

In short, the maximum number of allowable faulty components by FCAP is $T=T_{FS}+T_{FF}+f_{mC} = \sum_{j=1}^{R} f_{mRj} + \sum_{j=1}^{F} f_{mFj} + \lfloor (n_C\text{-}1)/3 \rfloor$. And, T is the maximum number of allowable faulty nodes in FC-IoT.

As a result, FCAP takes the minimum number of rounds and tolerates the maximum number of faulty components to make fault-free nodes reach a common consistency. The optimality of the protocol is proven.

7 Conclusions

The IoT could enable innovations that enhance the quality of life, but it generates unprecedented amounts of data that are difficult for traditional systems, the cloud, and even edge computing to handle. Fog computing is designed to overcome these limitations [6]. Fog computing extends the Cloud Computing paradigm to the edge of the network, thus enabling a new breed of applications and services [3].

While Fog nodes provide localization, therefore enabling low latency and context awareness, the Cloud provides global centralization. Many applications require both Fog localization, and Cloud globalization, particularly for analytics and Big Data. In this study, a high flexible and reliable IoT platform is proposed that combining Fog computing and Cloud computing (FC-IoT). By using FC-IoT, the monitoring system for prevention of earth-rock-flow disaster and the traffic control system can be constructed.

The agreement problem is fundamental in a distributed system, and has been extensively studied. Network topology is an important issue related to consistency. However, FC-IoT is a new concept for distributed systems. It has greatly encouraged distributed system design and practice to support user-oriented services. In this paper, the FCAP protocol is proposed to make all fault-free nodes reach agreement. This protocol can use a minimal number of rounds of message exchanges and tolerate a maximal number of allowable faulty components in a malicious fallible FC-IoT.

Merely considering component faults in the agreement problem is insufficient for the highly reliable distributed system of an IoT environment. A related closely problem is called the Fault Diagnosis Agreement (FDA) problem [5]. The objective of solving the FDA problem is to make each fault-free node detect or locate the common set of faulty components in the distributed system. Therefore, solving the FDA problem for the highly reliable distributed system underlying topology of FC-IoT is included in our future works.

Acknowledgments

This work was supported in part by the Ministry of Science and Technology MOST 107-2221-E-324-005-MY3.

References

[1] O. Babaoglu, D. Rogério, Streets of Byzantium: Network Architectures for Fast Reliable Broadcasts, *IEEE Transactions on Software Engineering*, Vol. 9, pp. 546-554, June, 1985.

[2] A. Bar-Noy, D. Dolev, C. Dwork, H. R. Strong, Shifting Gears: Changing Algorithms on the Fly to Expedite Byzantine Agreement, *Information and Computation*, Vol. 97, No. 2, pp. 205-233, April, 1992.

[3] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog Computing and its Role in the Internet of Things, *The first edition of the MCC Workshop on Mobile Cloud Computing*, Helsinki, Finland, 2012, pp. 13-16.

[4] A. Botta, W. De Donato, V. Persico, A. Pescapé, On the Integration of Cloud Computing and Internet of Things, *The 2014 International Conference on Future Internet of Things and Cloud*, Barcelona, Spain, 2014, pp. 23-30.

[5] O. Bousbiba, E. Klaus, A Fast Byzantine Fault-Tolerant Diagnostic Agreement Protocol for Synchronous Distributed Systems, *The 29th International Conference on Architecture of Computing Systems*, Nuremberg, Germany, 2016, pp. 1-11.

[6] A. V. Dastjerdi, R. Buyya, Fog Computing: Helping the Internet of Things Realize Its Potential, *Computer*, Vol. 49, No. 8, pp. 112-116, August, 2016.

[7] D. Dolev, R. Reischuk, Bounds on Information Exchange for Byzantine Agreement, *Journal of the ACM*, Vol. 32, No. 1, pp. 191-204, January, 1985.

[8] M. Ficco, C. Esposito, Y. Xiang, F. Palmieri, Pseudo-Dynamic Testing of Realistic Edge-Fog Cloud Ecosystems, *IEEE Communications Magazine*, Vol. 55, No. 11, pp. 98-104, November, 2017.

[9] M. J. Fischer, N. A. Lynch, A Lower Bound for The Assure Interactive Consistency, *Information Processing Letters*, Vol. 14, No. 4, pp. 183-186, June, 1982.

[10] Y. L. Hsieh, K. Wang, Dynamic Overlay Multicast for Live Multimedia Streaming in Urban VANETs, *Computer Networks*, Vol. 56, No. 16, pp. 3609-3628, November, 2012.

[11] J. Jin, J. Gubbi, S. Marusic, M. Palaniswami, An Information Framework for Creating a Smart City Through Internet of Things, *IEEE Internet of Things Journal*, Vol. 1, No. 2, pp.

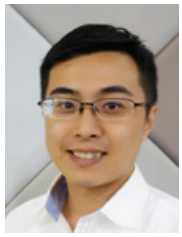
112-121, April, 2014.
[12] P. Kumar, S. K. Gupta, Abstract Model of Fault Tolerance Algorithm in Cloud Computing Communication Networks, *International Journal on Computer Science and Engineering*, Vol. 3, No. 9, pp. 3283-3290, September, 2011.
[13] L. Lamport, R. Shostak, M. Pease, The Byzantine General Problem, *ACM Transactions on Programming Languages and Systems*, Vol. 4, Issue 3, pp. 382-401, July, 1982.
[14] V. Mauch, M. Kunze, M. Hillenbrand, High Performance Cloud Computing, *Future Generation Computer Systems*, Vol. 29, No. 6, pp. 1408-1416, August, 2012.
[15] R. Merzouki, A. K. Samantaray, P. M. Pathak, B. O. Bouamama, *Intelligent Mechatronic Systems: Modeling, Control and Diagnosis*, Springer, London, 2013.
[16] F. J. Meyer, D. K. Pradhan, Consensus with Dual Failure Modes, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 2, No. 2, pp. 214-222, April, 1991.
[17] Y. Mo, L. Xing, W. Guo, S. Cai, Z. Zhang, J. H. Jiang, Reliability Analysis of IoT Networks with Community Structures, *IEEE Transactions on Network Science and Engineering*, 2018, DOI: 10.1109/TNSE.2018.2869167.
[18] D. Puthal, B. P .S. Sahoo, S. Mishra, S. Swain, Cloud Computing Features, Issues, and Challenges: A Big Picture, *2015 International Conference on Computational Intelligence and Networks*, Bhubaneshwar, 2015, pp. 116-123.
[19] H. S. Siu, Y.H. Chin, W.P. Yang, A Note on Consensus on Dual Failure Modes, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 7, No. 3, pp. 225-230, March, 1996.
[20] S. C. Wang, S. S. Wang, K. Q. Yan, Reaching Optimal Interactive Consistency in a Fallible Cloud Computing Environment, *Journal of Information Science and Engineering*, Vol. 34, No. 1, pp. 205-223, January, 2018.
[21] S. C. Wang, K. Q. Yan, C. F. Cheng, Efficient Multicasting Agreement Protocol, *Computer Standards & Interfaces*, Vol. 26, No. 2, pp. 93-111, March, 2004.
[22] A. Whitmore, A. Anurag, D. X. Li, The Internet of Things-A Survey of Topics and Trends, *Information Systems Frontiers*, Vol. 17, No. 2, pp. 261-274, March, 2015.
[23] K. Q. Yan, S. C. Wang, C. S. Peng, S. S. Wang, Optimal Malicious Agreement Protocol for Cluster-based Wireless Sensor Networks, *ScienceAsia*, Vol. 40S, No. 1, pp. 8-15, February, 2014.
[24] M. Yannuzzi, R. Milito, R. Serral-Gracià, D. Montero, M. Nemirovsky, Key Ingredients in an IoT Recipe: Fog Computing, Cloud Computing, and More Fog Computing, *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Network*, Athens, Greece, 2014, pp. 325-329.

Biographies

Shu-Ching Wang is a Professor at the Department of Information Management, Chaoyang University of Technology, Taiwan. Her current research interests include distributed data processing and reliability.

Wei-Shu Hsiung is a Ph.D. student of the Department of Information Management, Chaoyang University of Technology, Taiwan. His current research interests include distributed processing and fault tolerant.

Kuo-Qin Yan is a Professor at the Department of Business Administration, Chaoyang University of Technology, Taiwan. His current research interests include distributed fault tolerant computing and mobile computing.



Yao-Te Tsai is an assistant professor in the Department of International Business, Feng Chia University. His interests include internet of things and data analytics.