This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2894002, IEEE Access

Efficient Fault-Tolerant Routing in IoT Wireless Sensor Networks Based on Bipartite-Flow Graph Modeling

Jenn-Wei Lin, Pethuru Raj Chelliah, Meng-Chieh Hsu, and Jia-Xin Hou

Abstract—In Internet of Things (IoT), a wireless sensor network (WSN) is deployed for collecting the interested data of an application field. Sensor nodes in an IoT WSN are usually with heterogeneous property. Some nodes have more power (energy) and additional functionality (e.g. data aggregation) than others. Cluster-based routing is usually used in WSNs for data transmissions due to efficiently routing consideration. Based on cluster-based routing, the cluster heads (CHs) act as the sensed data forwarding role. Once one or more CHs fail, the faulty CHs cannot forward the sensed data of their serving sensor nodes. As a result, the sink node (gateway) has not sufficient sensed data of the IoT application field. This will deeply affect the information processing of IoT application. We utilize the virtual CH formation and flow graph modeling to efficiently tolerate the failures of CHs. First, the available resources of all failure-free CHs are logically organized as a virtual CH to be the common backup of all faulty CHs. Then, the flow graph modeling is used to achieve fault tolerance with the minimum total energy consumption among all failure-free CHs. Finally, we perform extensive experiments to demonstrate the effectiveness of our approach in fault-tolerant routing of IoT WSNs.

Index Terms—Internet of Things (IoT), wireless sensor networks (WSN), cluster based routing, fault tolerance, flow graph algorithm.

1 Introduction

Internet of Things (IoT) applications have been rapidly developed in various fields such as smart home, healthcare, environmental monitoring, industrial control, intelligent transport systems, etc [1]. For an IoT application, a wireless sensor networks (WSN) is usually deployed to collect the sensed data of the IoT application field [2]. Sensor nodes are first scattered in the IoT application field. Then, the sensor nodes periodically send back their sensed data to the sink (gateway) node of the WSN. Next, the sink node further processes the collected data in order to produce useful information for the IoT application. A lot of WSN routing protocols have been proposed to perform the sensed data transmissions from sensor nodes to the sink node. Those routing protocols can be classified to two categories: flat and cluster based routing protocols. Compared to the flat based routing protocols, the cluster based routing protocols have the advantage on the routing energy consumption [3], [4], [5], [6], [7]. In the cluster based routing protocols, all sensor nodes are divided into a number of clusters. For the sensor nodes within the same cluster, one sensor node is selected as the cluster head (CH) and others are the members of the cluster. The CH acts as an important routing role which assists all members in the same cluster to transmit their sensed data to the sink node via a one-hop or multi-hop transmissions. The members do not directly transmit sensed data to the sink node. Due to the routing assistance by the CH, members can save routing energy expenditure.

However, sensor nodes are usually small-size electronic devices equipped with limited power battery. After exhausting the battery energy, sensor nodes will not work. In addition, the hardware and software components of sensor nodes may frequently malfunction, especially when sensor nodes are deployed in the uncontrollable and harsh environments [8]. If a failure occurs in a CH, all serving members of the faulty CH cannot transmit their sensed data to the sink node via the faulty CH. As a result, the sink node has not enough amount of sensed data used for producing the necessary information of the IoT application. For providing reliable IoT applications, the fault tolerance of CHs is very important issue for IoT WSNs.

There have been a number of studies discussing the fault-tolerant issue of the CH for WSNs [9], [10], [11], [12], [13], [14], [15], [16]. In the fault-tolerant WSN literature, the existing approaches are based on two basic methods to tolerate the faulty CH: generation based and join based. In the generation based method, if a CH fails, one of its members will be designated as the new CH to serve other members continuously. The join based method does not generate a new CH. Instead, it makes the members of the faulty CH to be served by one or more existing failure-free CHs. In an IoT WSN, sensor nodes have heterogeneous property [17]. Some sensor nodes have larger amount of energy and additional functionality (e.g. data aggregation) than normal sensor nodes. The above first basic method is not applicable for fault tolerance of the IoT WSN with heterogeneous property. If a new CH is generated from

- *An earlier version of this paper was presented at the 2018 8th International Conference on Engineering and Applied Science Conference.*
- *Jenn-Wei Lin, Meng-Chieh Hsu, and Jia-Xin Houare are with the Department of Computer Science and Information Engineering, Fu Jen Catholic University, New Taipei City 24205, Taiwan. E-mail: jwlin@csie.fju.edu.tw.*
- *Pethuru Raj ChelliahSite is with Reliability Engineering (SRE) Division, Reliance Jio Infocomm. Ltd. (RJIL), Bangalore, India. Email: peterindia@gmail.com.*

Manuscript received xx xx, xx; revised xx xx, xx.

serving members (normal sensor nodes), the new CH has not plentiful energy and data aggregation functionality.

In this paper, we will propose a new approach for dealing with the CH failures of IoT WSNs. The new approach adopts the concept of the join based method. If a CH fails, all its serving members are managed by existing failure-free CHs. However, the join based method does not concern the following problems: 1) Pre-verifying the fault-tolerant capability, 2) Distributing the fault-tolerant load, and 3) Minimizing the total fault-tolerant cost. In the first problem, the fault-tolerant capability is verified for determining whether the IoT application requirement can be sustained or degraded. The second problem is particularly concerned when there is heavy transmission load in each CH of the IoT WSN. If a CH fails, the corresponding transmission load cannot be taken over by a failure-free CH alone. The third problem is for expecting that failure-free CHs do not incur large overhead due to fault tolerance. To cope with the above three problems, we propose two techniques in our new fault-tolerant approach: *virtual CH* and *flow-bipartite graph*. The virtual CH is formed by organizing the available energy of all failure-free CHs. With the virtual CH, we can estimate the average number of sensed data able to be transmitted from each failure-free CH. By verifying the estimated number, we can determine whether the transmission capability of the IoT WSNs can still meet the IoT application requirement. Then, a flow-bipartite graph is modelled for finding an optimal flow based pairs between faulty CHs and failure-free CHs. By following the flow based pairs, a faulty CH can be tolerated by two more failure-free CHs to achieve the fault-tolerant load distribution. In addition, we can also obtain the minimum total energy consumption in the normal and fault-tolerant sensed data transmissions. Finally, we perform extensively simulation experiments to compare the proposed new approach with other approaches in various performance metrics.

The rest of the paper is organized as follows. The WSN model and related fault-tolerate work are supplied in Section 2. Section 3 elaborates the proposed approach for tolerating CH failures in the IoT WSN. Section 4 evaluates the performance of the proposed approach. Finally, Section 5 concludes the manuscript.

2 PRELIMINARIES

This section describes the background materials of this paper. A system model of an IoT WSN is first given. Then, we introduce a well-known cluster based routing protocol: low-energy adaptive clustering hierarchy (LEACH) [18]. This protocol is used to understand that our approach is how to assist a cluster based routing protocol to achieve fault tolerance. Our fault-tolerant approach can be applied to other cluster-based routing protocols, not only LEACH. Next, we review a lot of existing cluster based fault-tolerant approaches.

2.1 System Model

We consider a set of sensor nodes and a sink node deployed in a two-dimensional field, as shown in Fig.1. There are two types of sensor nodes: normal and powerful sensor nodes. Compared to the normal sensor node, the powerful sensor node has more power (energy) and the data aggregation functionality. By dividing sensor nodes into two categories, the WSN field is correspondingly partitioned into a number of sub-fields (clusters). Within each cluster, a powerful sensor node acts as the cluster head (CH) role. The other powerful and normal sensor nodes in the same cluster are the members. The sensed data of a member is through its serving CH to the sink node. After collecting the sensed data of all sensor nodes, the sink node further processes those data to provide useful information for an IoT application. In addition, we also make the WSN of the IoT with the following properties.

- The sensor nodes and sink nodes are deployed at fixed locations.
- With the fixed location, the distance $dist$ between any two sensor nodes or one sensor node and one sink node can be calculated as follows.

$$dist = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

  where (x_1, y_1) is the location of the first node, and (x_2, y_2) is the location of the second node.
- The transmission range of a sensor node can be changed by adjusting the power level of the node [19], [20], [21].
- The failure of a CH can be detected by members of the faulty CH and the sink node. The members know the CH failure due to not receiving the acknowledgment of the sensed data or the beacon from the faulty CH [9]. The sink node can know the CH failure by the absence of the sensed data sent from the faulty CH for a period of time. This failure detection is also made in [12]. The failure of a CH node can be detected by the sink nodes and the serving members. The sink node can know the CH failures by the absence of the sensed data from the faulty CH [12]. The members knows the CH failure due to not receiving the acknowledgment of the sensed data or the bacon from the faulty CH [9].
- The sink node is not a sensor node which is the destination of each sensed data [21]. In cluster-based routing, the energy information of CHs can be appended on the sensed data to be known by the sink node. After collecting the sensed data, the sink node further processes the data to provide desired application information [1]. The sink node can be also regarded as a gateway [2] without the limited amount and size in energy and memory. It can perform complicated computations. Unlike sensor nodes, the sink node has not the limited amount and size in its energy supply and memory, respectively. It can perform complicated computations.

2.2 Cluster Based Routing Protocols

There have been many cluster based routing protocols, which can efficiently transmit sensed data from sensor nodes to the sink node [3], [4], [5], [6], [7]. The basic idea of a cluster based routing protocol is to divide sensor nodes into a number of clusters. In each cluster, a sensor node is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2894002, IEEE Access

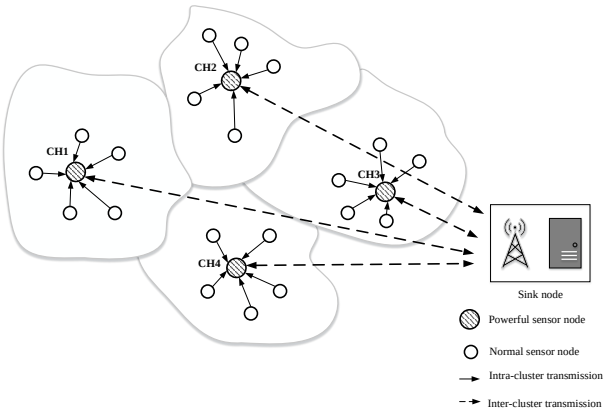


Fig. 1. A Clustered WSN Model.

selected as the cluster head (CH). Then, each CH broadcasts an advertisement message to the rest of sensor nodes to invite them as the cluster members. Each member directly sends the sensed data to its serving CH. Then, the CH aggregates the received data and forwards the sensed data to the sink node directly or indirectly.

The protocol of low energy adaptive clustering hierarchy (LEACH) [18] is the first proposed cluster routing protocol. LEACH is operated based on the round unit. In the beginning of a round, a number of clusters are first formed. Then, each cluster selects a sensor node as the CH hole. The LEACH expects that each sensor node takes the role of CH in different runs. Except CHs, other sensor nodes will select their closest CHs to join the corresponding clusters as members. Then, each CH makes a TDMA (time division multiple access) schedule to arrange the intra-cluster transmissions of its serving members. The length of a round contains a number of TDMA transmission schedules (durations). At the end of a round, clusters are re-established in next new round. The above operations are repeated in the new round. Many variations of LEACH protocol and other types of cluster routing protocols have been proposed in [3], [4], [5], [6], [7]. They have similar operations with LEACH.

2.3 Related Work

In the cluster based routing protocols, the CH acts a very important role in sensed data transmissions. The fault tolerance of CH failure has been discussed in a lot of literature.

In [9], [10], once a failure occurs in a CH, each member of the faulty CH (*failure-affected member*) can detect the failure due to not receiving the acknowledgement message. Each failure-affected member m_i re-selects a backup CH from neighboring clusters by broadcasting a help message within its communication range. If more than one neighboring CH respond, the neighboring CH closest to m_i will be selected as the best backup CH. Similar to [9], [10], the fault-tolerant idea of joining existing CH is also adopted by [11]. In addition to the proximity to the existing CH, the residual energy, distance to the sink node, number of serving members are also considered in the backup CH selection. In the above fault-tolerant approaches, if there are too many failure-affected members, it will introduce the help message explosive problem. Moreover, each failure-affected member may select a different backup CH. The sensed data of such failure-affected members cannot be aggregated together. To cope with the possible message explosive problem and the non-data aggregation problem, the work of [12] suggested two fault-tolerant solutions. One is that the sink node initiates re-clustering of entire WSN. However, this solution is lengthy and involves all sensor nodes. The cheaper solution is to select an existing CH alone to serve the failure-affected members of a faulty CH and aggregate their sensed data. The details of the appropriate backup CH selection is not elaborated in [12].

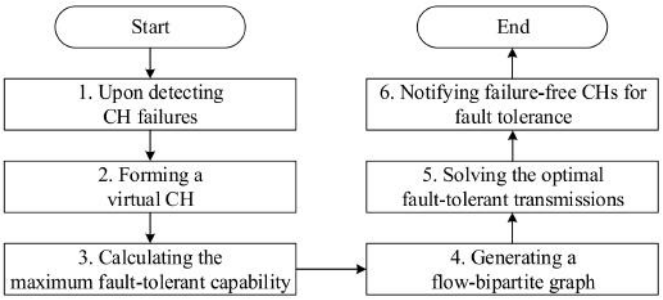


Fig. 2. The basic idea of the proposed approach.

In [13], the proposed fault-tolerant algorithm avoids increasing the workloads of existing CHs. The algorithm generates a new CH from the failure-affected members based on their remaining energy. The node with the largest remaining energy will be designated as the new CH instead of the faulty CH. The similar fault-tolerant method is also applied in [14], [15], [16]. However, [14] has the special characteristic in the two-level cluster formation to reduce transmission burden on CHs. The special work of [15] is on the identification of overlapped nodes. If two sensor nodes has the similar responsible coverage areas, the two nodes are the overlapped nodes with each node. One of the overlapped nodes gets into the sleep mode to remain energy. When a CH fails, the nodes with the sleep mode are waken up to participate the new CH selection. The unique characteristic of [16] uses genetic algorithm to select one or more failure-affected members as the new CH. In the genetic algorithm, the coverage area of a failure-affected member and the residual energy are two important parameters. In the above generation based fault-tolerant approaches, the new CH may be a normal sensor node without plentiful energy and the data aggregation functionality. In such a case, the new CH will quickly run out of its energy to re-select a new CH again.

3 Proposed Fault-Tolerant Approach

In this section, we propose a new fault-tolerant routing approach for IoT WSNs. The approach can enhance the join based fault-tolerant method with the three features mentioned in Section 1: fault-tolerant capability pre-verification, fault-tolerant load distribution, and fault-tolerant cost minimization.

3.1 Basic Idea

To achieve the above three features, we design two techniques in our approach: *virtual CH* and *flow-bipartite graph*.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2894002, IEEE Access

TABLE 1
Summary of Basic Notations

Basic Notation	Description
N_{ff} (N_f)	Number of failure-free (faulty) CHs
D_s	Size of a sensed data frame
E_{tx} (E_{rx})	Transmitting (Receiving) energy consumption
E_{da}	Energy consumption for data aggregation
CH_i^{ff} (CH_i^f)	The ith failure-free (faulty) CH
$\lvert CH_i^{ff}\rvert$ ($\lvert CH_i^f\rvert$)	Number of failure-free (failure-affected) members in the ith CH
$\overline{CH_i^f}$	The set of all failure-affected members of the ith faulty CH
$E_{CH_i}^{ff}$	The amount of available energy of the ith failure-free CH
$Fm(CH_i^f)$	The farthest member of the ith faulty CH

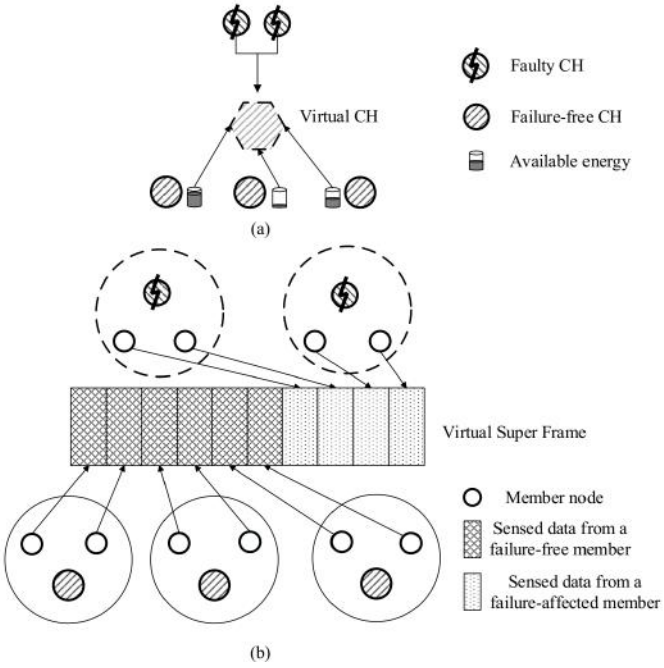


Fig. 3. The organization of the virtual CH and structure of a virtual super frame. (a) Virtual CH. (b) Virtual super frame.

The basic idea of the proposed approach is shown in Fig. 2. When some CHs fail, the failure event can be detected by the sink node (see subsection 2.1). Due to periodically transmitting sensed data between a CH and the sink node, the sink node can also obtain the residual energy information of each CH. Additionally, the sink node has not the limited energy and memory problem (see subsection 2.1). The sink node can perform complicated operations to logically form a virtual CH using the obtained available energy information. Based on the virtual CH, the sink node pre-verify the maximum fault-tolerant capability offered by all failure-free CHs. Then, the sink node models a flow-bipartite graph to represent all fault-tolerant possibilities between faulty CHs and failure-free CHs. Given the calculated maximum fault-tolerant capability to the graph, the sink node determines the optimal fault-tolerant transmission pairs between faulty CHs and failure-free CHs. Finally, the sink node passes the fault-tolerant information to failure-free CHs to take over respective corresponding faulty CHs. The details of the two key techniques will be elaborated in next two subsections.

3.2 Virtual CH

In our approach, we first logically generate a virtual CH on the sink node based on the total available energy of all failure-free CHs, as shown in Fig. 3(a). In Fig. 3(a), the two faulty CHs are tolerated by the virtual CH. The virtual CH is logically constituted by the three failure-free CHs. In addition to failure-affected members, the virtual CH also needs to transmit the sensed data of the sensor nodes originally served by failure-free CHs (the sensed data of failure-free members). We also adopt a virtual super frame structure to represent all the sensed data of the failure-free and failure-affected members in the virtual CH, as shown in Fig. 3(b). In Fig. 3(b), the virtual super frame is constituted by the sensed data of six failure-free members and four failure-affected members.

Note that the virtual CH does not really exist, which is logically formed by failure-free CHs. In this section, we first discuss the fault tolerance based on the virtual VCH and the transmission capability given by the virtual CH. In next section, we will discuss how to re-direct the sensed data transmissions of the virtual CH to be done on failure-free CHs.

Definition 1. *The virtual super frame contains two part: failure-free and fault-tolerant. In these two parts, the sensed data is originally sent from failure-free and failure-affected members and then goes through failure-free CHs, respectively.*

Definition 2. *The transmission capability of the virtual CH is defined as how many virtual super frames able to be transmitted from the virtual CH.*

To calculate the transmission capability of the virtual CH, the following definitions are given, which need to use the notations of Table 1.

Definition 3. *For a super frame, the failure-free energy consumption E_{SF_1} (the energy taken by the failure-free part of the super frame) can be estimated as follows.*

$$E_{SF_1} = \sum_{i=1}^{N_{ff}} |CH_i^{ff}| \times D_s \times E_{rx} + \sum_{i=1}^{N_{ff}} |CH_i^{ff}| \times D_s \times E_{da} + \sum_{i=1}^{N_{ff}} Trans(CH_i^{ff}, D_s) \tag{2}$$

where there are three consumption items in the failure-free energy consumption.

- The first is the energy taken for receiving the sensed data from all the failure-free members. The total number of the failure-free member nodes is $\sum_{i=1}^{N_{ff}} |CH_i^{ff}|$. For convenience, we assume that each member has the same size in its sensed data.
- The second is the energy taken for performing the data aggregation on those received sensed data.
- The last is the energy taken for transmitting the aggregated failure-free part data to the sink node. Most cluster based routing protocols adopts the same-size data aggregation model, such that the size of aggregated sensed data is equal to that of one received sensed data [21], [22]. The size of an aggregated sensed data is also D_s. In addition to the size of the transmitted data, the transmitting energy is also dependent on the distance between a sending node and a receiving node. Each failure-free CH has different distance with the sink node. Here, we use $Trans(CH_i^{ff}, D_s)$ to represent the energy consumption of the failure-free CH CH_i^{ff} for transmitting its aggregated sensed data to the sink node.

Definition 4. *The fault-tolerant energy consumption E_{SF_2} of a super frame can be estimated as follows.*

$$E_{SF_2} = \sum_{i=1}^{N_f} |CH_i^f| \times D_s \times E_{rx} + \sum_{i=1}^{N_f} |CH_i^f| \times D_s \times E_{da} + AveTrans(D_s) \times E_{tx} \times N_f \tag{3}$$

where there are also three consumption items in the fault-tolerant energy consumption similar to Eq. (2). The failure-free CHs instead of the faulty CHs receive the sensed data of failure-affected members, perform the data aggregation on those sensed data, and transmit the aggregated sensed data to the sink node. Here, we do not known which faulty CH is tolerated by which failure-free CH. We use the average transmission energy consumption $AveTrans(D_s)$ to represent the average energy consumption of a failure-free CH for transmitting one aggregated sensed data to the sink node.

$$AveTrans(D_s) = \frac{\sum_{i=1}^{N_f} Transmit(CH_i^f, D_s)}{N_f} \tag{4}$$

Definition 5. *The amount E_{VCH} of available energy of the virtual CH is equal to the sum of the available energy of all failure-free CHs.*

$$E_{VCH} = \sum_{i=1}^{N_{ff}} E_{CH_i} \tag{5}$$

Definition 6. *Based the Definitions 3, 4, and 5, the transmission capability TC_{VCH} of the virtual is*

$$TC_{VCH} = \frac{E_{VCH}}{(E_{SF_1} + E_{SF_2})} \tag{6}$$

where a virtual super frame has the failure-free and fault-tolerant parts. The transmission capability can be derived by E_{VCH} divided by the sum of the failure-free and fault-tolerant energy consumption.

In addition to the above estimated transmission capability, the following two parameters are referred in the verification of the fault-tolerant capability.

- The minimum number D_{IoT} of sensed data from each cluster of a WSN: In a cluster based WSN, the processing data of an IoT application is collected from the clusters of the IoT WSN. If each cluster cannot support enough sensed data within a period of time, the IoT application mission cannot be achieved. For example, in the IoT application of air qualify monitoring, the WSN senses the air data with the (termperature, humidity, CO_2) format from each cluster and sends back to the sink node [23]. In the sink node, the air quality application periodically processes the collected data in order to provide the analysis information of air qualify. If a certain number of air sensed data cannot be collected from a cluster, the analysis outcome of air quality cannot fully represent the actual air quality of the monitoring environment.
- The number R_{IoT} of sensed data from each cluster already received by the sink node. The sink node receives the sensed data from each cluster. After accumulating a required number of sensed data, the IoT application begins to be executed.

Definition 7. *The verification of the fault-tolerant capability of the virtual CH: After some CH fails, the demand number D_d of sensed data from each cluster is first calculated $(D_{IoT} - R_{IoT})$. If TC_{VCH} is large than or equal to D_d, the virtual CH can replace all the faulty CHs to transmit enough sensed data to the sink node for the IoT application processing. Conversely, the current IoT WSN cannot support D_d sensed data from each cluster. In such a case, D_d is degraded to TC_{VCH}.*

3.3 Flow-Bipartite Graph

To achieve the other two features of our approach, we apply for a flow-bipartite graph. The flow-bipartite graph can also make that the sensed data transmissions of the virtual CH are done on failure-free CHs.

Definition 8. *A flow-bipartite graph is(FBG) a graph whose vertices are divided into source and destination sets (U and V) such that every edge connects a vertex in U to one in V. The same amount of input flow f is assigned to each source vertex of U (u_i). Each destination vertex of V (v_j) is attached with a capacity cap_j. Between U and V, each edge is associated with a transmission cost $tc_{(u_i,v_j)}$ and a energy cost $ec_{(u_i,v_j)}$. Unlike the classical network flow graph, FBG has the capacity constrain in each destination vertex, not on each edge.*

FBG is established based on the following steps.

- All faulty and failure-free CHs as the source and destination vertices set U and V, where u_i and v_j are corresponding to a faulty CH $CH_{u_i}^f$ and a failure-free CH $CH_{v_j}^{ff}$, respectively.
- For each source vertex u_i, the amount of input flow f is set to be the demand number D_d of sensed data from each cluster (see Definition 7).
- The capacity cap_{v_j} of each destination vertex v_j is determined by the available energy of the corresponding failure-free CH of v_j, which is denoted as follows.

$$cap_{v_j} = Avail\_energy(CH_{v_j}^{ff}) \tag{7}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2894002, IEEE Access

- To determine an edge between u_i and v_j, we first calculate the largest possible distance between a failure-affected member of $CH^f_{u_i}$ and $CH^{ff}_{v_i}$, as follows.

$$LDist = Dist_{(Fm(CH^f_{u_i}),CH^f_{u_i})} + Dist_{(CH^f_{u_i},CH^{ff}_{v_i})} \tag{8}$$

where *Dist* is the function for calculating the distance between two nodes. $LDist$ is calculated by assuming that the farthest member node of $CH^f_{u_i}$ and CH_{v_j} are located at the opposite directions of $CH^f_{u_i}$. In such a case, the largest possible distance can be estimated. Then, the $LDist$ is compared with the communication range of a sensor node. If the former is larger, it represents that each failure-affected member of CH_{u_i} can transmit its sensed data to CH_{v_j}. Therefore, an edge is established between CH_{u_i} and CH_{v_j}.

- The transmission cost of a flow-bipartite edge (u_i, v_j) is calculated as follows.

$$tc_{(u_i,v_j)} = Dist_{(CH^{ff}_{v_j},sink)} + Dist_{\overline{(CH^f_{u_i},CH^{ff}_{v_j})}} \tag{9}$$

where $tc(u_i, v_j)$ contains two distance factors. The fist factor is the distance between $CH^{ff}_{v_j}$ and sink. If $CH^{ff}_{v_j}$ has the small distance with the sink node, $CH^{ff}_{v_j}$ has the advantage in the transmitting energy consumption with the sink node. The second factor is the average distance of all the failure-affected members of $CH^f_{u_i}$ with $CH^{ff}_{v_j}$. If the average distance is short, the failure-affected members can save energy consumption in their sensed data transmissions to the fault-tolerant CH $CH^{ff}_{v_j}$.

- The fault-tolerant energy cost of a flow-bipartite edge (u_i, v_j) is calculated as follows.

$$ec_{(u_i,v_j)} = E_{(CH^f_i,CH^{ff}_j)} + E_{CH^{ff}_j} \tag{10}$$

where $ec(u_i, v_j)$ also contains two energy factors. The first factor is the energy taken for CH^{ff}_j to assist the transmissions of failure-affected members of CH^f_i to the sink node. The second factor is the energy taken for CH^{ff}_j to transmit the sensed data of itself failure-free members to the sink node. By referring to Eq. (2), $E_{(CH^f_i,CH^{ff}_j)}$ and $E_{CH^{ff}_j}$ can be further denoted as follows.

$$\begin{aligned} E_{(CH^f_i,CH^{ff}_j)} =& |CH^f_i| \times D_s \times E_{rx} + |CH^f_i| \times D_s \times \\ & E_{da} + Trans(CH^{ff}_j, D_s) \end{aligned} \tag{11}$$

$$\begin{aligned} E_{CH^{ff}_j} =& |CH^{ff}_j| \times D_s \times E_{rx} + |CH^{ff}_j| \times D_s \times E_{da} \\ & + Trans(CH^{ff}_j, D_s) \end{aligned} \tag{12}$$

Based on the established FBG, we would like to solve the minimum cost flow (*MCF*) problem on the graph.

Definition 9. *Given an amount of flow to each input node of FBG, what is the total minimum fault-tolerant transmission cost to send all flows from source vertices to destination vertices of FBG as many as possible?*

Based on Definition 8, FBG is a variant of the network flow graph. There are flow transmission contentions on a destination vertex since there is a capacity constraint on the vertex. The well-known MCF solution cannot be applied on the FBG. We use *Integer Linear Programming* (ILP) to obtain the optimal MCF solution of FBG. ILP is a well-known mathematical technique for solving optimal problems, which consists of an objective function, several linear constraints, and an integer solution set [24].

$$\text{Minimize} \sum_{\forall u_i \in U} \sum_{\forall v_j \in V} x(u_i, v_j) \times tc(u_i, v_j) \tag{13}$$

$$\text{subject to } \forall u_i \in U, \sum_{\forall v_j \in V} x(u_i, v_j) = D_d, \tag{14}$$

$$\begin{aligned} \forall v_j \in V, & \sum_{\forall u_i \in U} x(u_i, v_j) \times (E_{(CH^f_{u_i},CH^{ff}_{v_j})} + E_{CH^{ff}_{v_j}}) \\ & \leq cap_{CH^{ff}_{v_j}} \end{aligned} \tag{15}$$

$$\forall u_i \in U, \forall v_j \in V, 0 \leq x(u_i, v_j) \leq D_d \tag{16}$$

In the above ILP model, Eq. (13) is an objective function, which aims to minimize the total transmission cost. $x(u_i, v_j)$ denotes the number of fault-tolerant transmissions is done between the faulty CH_{u_i} and the failure-free CH_{v_j}. Eq. (16) denotes the possible values of $x(u_i, v_j)$ which is between 0 and D_d (the demand number of sensed data, see Definition 7). $tc(u_i, v_j)$ has been defined in Eq. (9). Eq. (14) is the transmission requirement (D_d) of each faulty CH. For a faulty CH, its required sensed data transmissions may be satisfied by more than one failure-free CH. Eq. (15) is the energy constraint of each failure-free CH. For a failure-free CH, it needs to transmit the sensed data of some failure-affected members in addition to itself failure-free members. The energy consumption of those transmissions cannot exceed the own energy capacity For $E_{(CH^f_{u_i},CH^{ff}_{v_j})}$, $E_{CH^{ff}_{v_j}}$), and $cap_{CH^{ff}_{v_j}}$, they have been defined in Eq. (11), Eq. (12), and Eq. (7), respectively We will demonstrate how to apply for the ILP model to get the optimal MCF solution of FBG in next subsection.

After solving MCF problem on FBG, the flow transmission patterns represent which faulty CH will be tolerated by which failure-free CHs to carry a certain number of sensed data to the sink node. In MCF solution, the flow of a source vertex may be split to multiple destination vertex. This means that the required fault-tolerant transmissions of the faulty CH are completed by two or more failure-free CHs. It achieves the feature of fault-tolerant load distribution of the proposed approach. In addition, MCF solution can obtain the total minimum transmission cost. This fulfills the third feature of our approach.

If there are many faulty or failure-free clusters, it will take much time for solving ILP. Therefore, we also propose a heuristic MCF-solving algorithm, as shown in Fig. 4.

The heuristic algorithm first sorts the edges of the flow-bipartite graph in increasing order of the transmission costs as the set O_e. Then, the algorithm runs in iterations. Before iterations, each faulty CH is designated the number D^r_i of required fault-tolerant data transmissions as D_d (see lines 3-5). In each iteration, the edge with the minimum cost is selected from O_e. Based on the selected edge (CH^f_i, CH_jff), we can further get D^r_i and the available energy E^o_j own

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2894002, IEEE Access

Input: A flow-bipartite graph $FBG = (V, E)$ with the sets of faulty and failure-free CHs: CH^f and CH^{ff} as well as the number of demand sensed data from each cluster D_d.
Output: A set CH^{ft} of fault-tolerant transmission tuple $(CH_i^f, CH_j^{ff}, n_{FT})$.
1: $O_e \leftarrow$ Sorting the edges of FBG by their associated transmission costs in increasing order.
2: $CH^{ft} \leftarrow \emptyset$.
3: **for** $i \in V$ **do**
4: $D_i^r \leftarrow D_d$
5: **end for**
6: **while** $O_e \neq \emptyset$ **do**
7: Selecting an edge $(CH_i^f, CH_j ff)$ from O_e with the minimum weight.
8: $E_j^o \leftarrow$ Getting the current energy offered by CH_j^{ff} .
9: $E_i^r \leftarrow$ Calculating the energy required if CH_i^f is tolerated by CH_j^{ff} to transmit D_i^r
10: **if** $E_j^o \geq E_i^r$ **then**
11: $CH_{ft} \leftarrow CH_{ft} \cup (CH_i^f, CH_j ff, D_i^r)$
12: $E_j^o \leftarrow E_j^o - E_i^r$
13: $D_i^r \leftarrow 0$
14: Removing all the edges of FBG with the source vertex = CH_i^f
15: **else**
16: $D_j^o \leftarrow$ Calculating the fault-tolerant data transmissions of CH_i^f offered by CH_j^{ff}
17: $D_i^r \leftarrow D_i^r - D_j^o$
18: $E_j^o \leftarrow 0$
19: $CH_{ft} \leftarrow CH_{ft} \cup (CH_i^f, CH_j ff, D_j^o)$
20: Removing all the edges of FBG with the destination vertex = CH_j^{ff}
21: **end if**
22: **end while**

Fig. 4. The heuristic algorithm for solving MCF problem on FBG.

by CH_j^{ff}. Then, we calculate the energy E_i^r required for transmitting D_i^r sensed data between CH_i^f and $CH_j ff$ (see line 9). By comparing E_i^r and E_j^o, D_i^r and E_j^o are updated. The fault-tolerant transmission tuples are collected in CH_{ft}. In addition, the corresponding edges are also removed from FBG (see lines 10-21). Next, the iteration is repeated until there is no edge on FBG. After ending the iterations, CH_{ft} include the fault-tolerant transmission pairs between faulty CHs and failure-free CHs.

The time complexity of the proposed heuristic algorithm is $O(|E|log|E|)$, where $|E|$ is the number of edges on the modelled flow-bipartite graph. The heuristic algorithm consists of two components: edge sorting and edge selection. The edge sorting takes $O(|E|log|E|)$. In the edge selection, there are $O(|E|)$ iterations. Each iteration takes constant time. Overall, the time complexity of the entire heuristic algorithm is $O(|E|log|E|) + O(|E|) \approx O(|E|log|E|)$

3.4 Implementation

From the above two subsections, we know that the flow-bipartite graph can be used to form a virtual CH among all failure-free CHs. Note that the virtual CH provides fault-tolerant transmissions for all failure-affected members. To further understand the formation of the virtual CH, we present the detailed operations in Fig. 5. The formation process of the virtual CH is initiated after solving MCF problem on the established FBG. From the MCF solution, we can get the fault-tolerant relationships between faulty CHs and fault-free CHs. In addition, it also gives how many failure-affected members should be served by a failure-free

Input: After performing MCF-solving algorithm on FBG, we can obtain a list of flow-pattern triples with the form (u_i, v_j, f_n).
Output: A set of failure-affected members which are re-joined new failure-free CHs.
Sink node part:
1: Delivering a fault-tolerant message containing the flow-pattern list to all failure-free CHs
Each failure-free CH CH_{v_k} part:
2: **if** it receives the fault-tolerant message from the sink node **then**
3: **if** its ID meets one or more flow-pattern triples with $v_j = v_k$ **then**
4: /* CH_{v_k} will tolerate one or more faulty CHs CH_u */
5: CH_{v_k} broadcasts an advertisement message of the members of all its tolerated faulty CHs.
6: **end if**
7: **end if**
8: **if** it receives the join message from the failure-affected member m_k **then**
9: $CH_{u_i} \leftarrow$ Retrieving the ID of the faulty CH from the join message
10: $f_n \leftarrow$ Using (CH_{u_i}, CH_{v_k}) to find the corresponding flow-patter triple (u_i, v_k, f_n) and get the amount of allowed flow f_n
11: **if** more than f_n failure-affected members has re-joined CH_{v_k} **then**
12: Sending a rejection message
13: **else**
14: Sending an acknowledgement
15: **end if**
16: **end if**
Each failure-affected member m_f part:
17: **if** it receives the advertisement message from the failure-free CH_{v_k} **then**
18: Putting CH_k in its potential fault-tolerant CH list
19: **end if**
20: $CH_{u_i}\ leftarrow$ Get the ID of the corresponding faulty CH of m_f
21: **while** not receiving an acknowledge from a failure-free CH **do**
22: Retrieving a CH CH_{v_k} from its potential fault-tolerant CH list
23: Sending a join message attached with Ch_{u_i} to the failure-free CH_{v_k}
24: **end while**

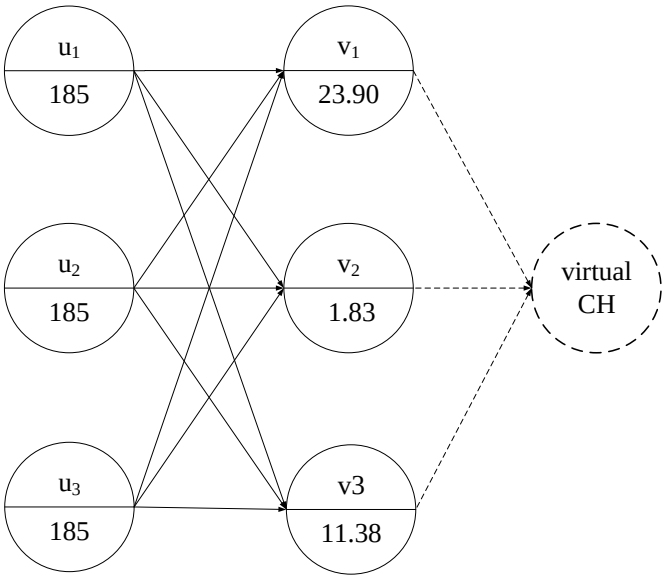
Fig. 5. The implementation of virtual CH.

CH. Next, the sink node delivers the above information attached on a failure-free message to all failure-free CHs. Upon receiving the fault-tolerant message, if a failure-free CH is designated to tolerate one or more faulty CHs, it will broadcast an advertisement message to the corresponding failure-affected members and wait for the join message. If the failure-free CH finds more than allowed failure-affected members, it will reject the incoming join message. For the failure-affected member, it may receive more than one advertisement message from several failure-free CHs. These failure-free CHs are its potential fault-tolerant CHs. It alternatively sends the join message to such failure-free CHs. Once it gets the acknowledgment message, it finds the formal fault-tolerant CH and stops sending the join message.

3.5 An Example

In this subsection, we illustrate an example to clarify how to apply for the flow-bipartite graph (FBG) to achieve fault tolerance. At first, we assume that a WSN of 100 sensor nodes is divided into six clusters. After a period of time, some failures occur in three CHs (u_1, u_2, and u_3). The remaining three CHs are failure-free CHs (v_1, v_2, and v_3) which can tolerate the three faulty CHs. As shown in Fig. 6, FBG has three source and destination vertices. To model

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2894002, IEEE Access



(a)

edge	transmission cost	energy cost
$u_1 \rightarrow v_1$	2.68	0.06
$u_1 \rightarrow v_2$	0.80	0.06
$u_1 \rightarrow v_3$	1.41	0.06
$u_2 \rightarrow v_1$	2.10	0.07
$u_2 \rightarrow v_2$	0.48	0.07
$u_2 \rightarrow v_3$	1.71	0.07
$u_3 \rightarrow v_1$	2.80	0.07
$u_3 \rightarrow v_2$	3.45	0.07
$u_3 \rightarrow v_3$	5.40	0.07

(b)

Fig. 6. An fault-tolerant example using the flow-bipartite graph. (a) graph model (b) edge cost setting

FBG, we need to further set the input flow of each source vertex, the energy capacity of each destination vertex, and the transmission and energy costs of each edge. In Fig. 6, the values of the above parameters on FBG are set by running a simulation experiment to calculate them based on Definition 7 and Eq. (7), Eq. (9), and Eq. (10), respectively. Based on given FBG of Fig. 6, its MCF solution can be obtained by the following ILP model. Note that the ILP model is developed by referring to Eq. (13) − Eq. (16).

$$\begin{aligned} \text{Minimize } & 2.68 \times x(u_1, v_1) + 0.80 \times x(u_1, v_2) + \\ & 1.41 \times x(u_1, v_3) + 2.10 \times x(u_2, v_1) + \\ & 0.47 \times x(u_2, v_2) + 1.71 \times x(u_2, v_3) + \\ & 2.80 \times x(u_3, v_1) + 3.45 \times x(u_3, v_2) + \\ & 5.41 \times x(u_3, v_3) \end{aligned} \quad (17)$$

$$\text{Subject to } \forall i \in \{1,2,3\}, \sum_{\forall j \in \{1,2,3\}} x(u_i, v_j) = 185 \quad (18)$$

$$\begin{aligned} & 0.06 \times x(u_1, v_1) + 0.07 \times x(u_2, v_1) + 0.07 \times x(u_3, v_1) \\ & \leq 23.90 \\ & 0.06 \times x(u_1, v_2) + 0.07 \times x(u_2, v_2) + 0.07 \times x(u_3, v_2) \\ & \leq 1.83 \\ & 0.06 \times x(u_1, v_3) + 0.07 \times x(u_2, v_3) + 0.07 \times x(u_3, v_3) \\ & \leq 11.38 \end{aligned} \quad (19)$$

$$\forall i \in \{1,2,3\}, \forall j \in \{1,2,3\}, 0 \leq x(u_i, v_j) \leq 185 \quad (20)$$

The above ILP model, is solved by IBM ILOG CPLEX Optimizer [27]. We obtain that the total transmission cost is 1144.41. The fault-tolerant transmission pairs between faulty CHs and failure-free CHs are $x(u_1, v_2) = 20$, $x(u_1, v_3) = 165$, $x(u_2, v_1) = 179$, $x(u_2, v_2) = 6$, and, $x(u_3, v_1) = 185$. The results represent that the first faulty CH is tolerated by the second and third failure-free CHs which need to additionally transmit 20 and 165 aggregated sensed data for the first faulty CH, respectively. The second faulty CH is tolerated by the first and second failure-free CHs which additionally transmit 179 and 6 aggregated sensed data for the second faulty CH, respectively. The third faulty CH is only tolerated the first failure-free CH which additionally transmits the required 185 aggregated sensed data.

We also solve the MCF solution of the exemplified FBG using our proposed heuristic algorithm. The algorithm selects the bipartite edges with small transmission costs first. It obtains the total transmission cost: 1149.4. Compared to the ILP solution, the total cost difference is 5.01. The heuristic algorithm also gets the following fault-tolerant transmission pairs: $x(u_1, v_1) = 20$, $x(u_1, v_3) = 165$, $x(u_2, v_1) = 159$, $x(u_2, v_2) = 26$, and, $x(u_3, v_1) = 185$.

4 Performance Evaluation

We used NS2 with incorporation of MIT uAMPS LEACH module [25], [26] to perform simulation experiments for making the performance comparisons among our approach and related approaches. To solve the formulated ILP model of the proposed approach, we adopted IBM ILOG CPLEX Optimizer to obtain the optimal solution [27].

Based on the two well-known basic CH fault-tolerant concepts, the related approaches of Section 2.3 can be further classified into new cluster head generation without data aggregation (NHG_NonDA) [13], new cluster head generation with data aggregation (NHG_DA) [14], [15], [16], joining the closest existing CH with the distributed manner ($JCECHDM$) [9], [10], and joining the existing CH with multiple factor consideration ($JECHMF$) [11]. For our approach, the optimal solution based on ILP and heuristic solution have been presented in Section 3. The two versions of the proposed approach are based on the virtual CH and flow-bipartite graph techniques, which are called as $VCHFBG\_ILP$ and $VCHFBG\_Heuristic$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2894002, IEEE Access

TABLE 2
Energy Parameters

Parameter	Value
Initial energy	10J (powerful sensor) 5J (normal sensor)
E_{elec} (Electronic energy)	50 nJ/bit
E_{amp} (amplifier energy)	0.0013pJ/bit/m^4
E_{fs} (amplifier energy)	10pJ/bit/m^2
E_{fusion} (data aggregation energy)	5nJ/bit/signal

4.1 Simulation Environment

We assume there is a 100 * 100 square field with 1000 sensor nodes randomly deployed in this field. For those sensor nodes, 10% of nodes are powerful sensor nodes. The initial energy of a powerful sensor node is twice that of a normal sensor node. The sink node is located on the coordination (50, 50) of the field. The size of the sensed data is 500 bytes. We refer to the given energy model and energy parameters of [28] to calculate the transmitting and receiving energy consumption of sensor nodes. The energy model is illustrated in Fig. 7. Moreover, the energy parameters used in simulation experiments are summarized in Table 2. The stable election protocol (SEP) [29] is used to perform clustering among heterogeneous sensor nodes at the beginning of each round. In a round, the failure threshold value is set to be 0.2 (0.5). Then, each CH is attached with a random failure occurring probability. If the random probability is less than 0.2 (0.5), the CH is a faulty CH. Here, the two failure thresholds can generate two different numbers of faulty CHs in each round. Compared to 0.2 failure threshold, 0.5 failure threshold has more faulty CHs. Based on the above settings, we perform 50 simulation runs to measure the following metrics:

- Average backup energy consumption : The backup node of a faulty CH may be a serving member of the faulty CH (generation based method) or a failure-free CH (join based method). The backup node will take energy to receive these additionally sensed data from the failure-affected members. Furthermore, the backup node also needs to transmit the additionally sensed data to the sink node. This metric is to measure the average energy consumption taken by a backup node for fault tolerance.
- Average failure-affected energy consumption: The failure-affected members transmit their sensed data to the backup node instead of the faulty CH. This metric is to measure the average energy consumption taken by the failure-affected members of a faulty CH.
- Network lifespan after fault tolerance: The backup nodes instead of faulty CHs perform sensed data transmissions After fault tolerance, the WSN network lifespan is dependent on the minimum and maximum numbers of transmissions offered by the backup nodes. Note that the minimum and maximum numbers denotes the lower-bound and upper-bound of the network lifespan, respectively.

4.2 Simulation Results

Fig. 8 shows the comparisons of the average backup energy consumption at two different failure thresholds. The

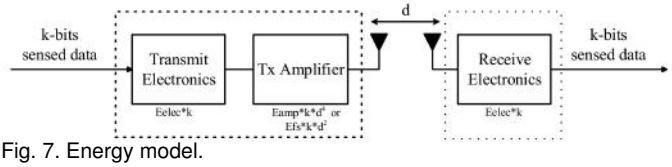
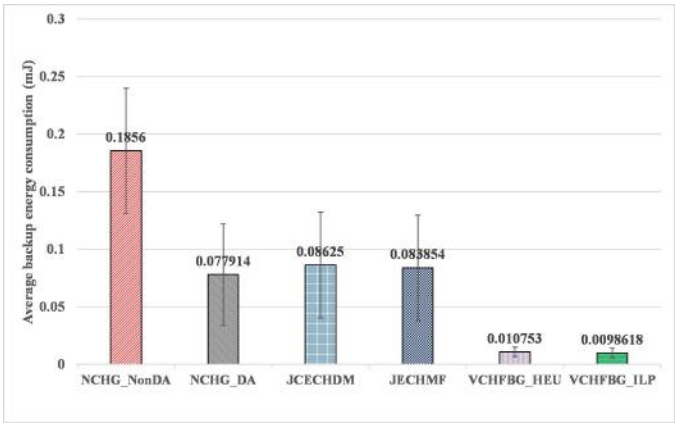
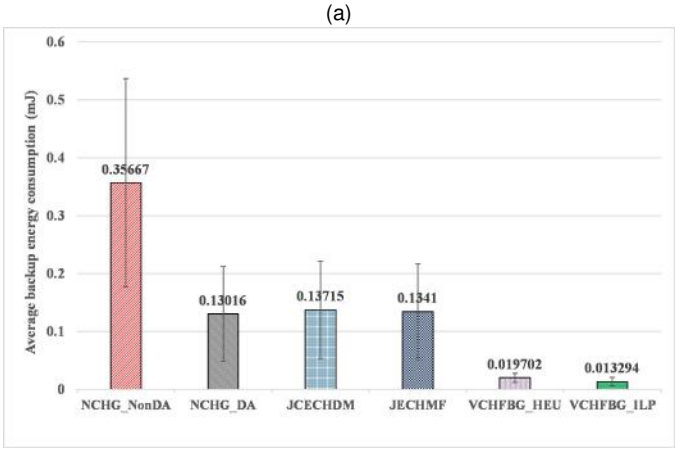


Fig. 7. Energy model.



(a)

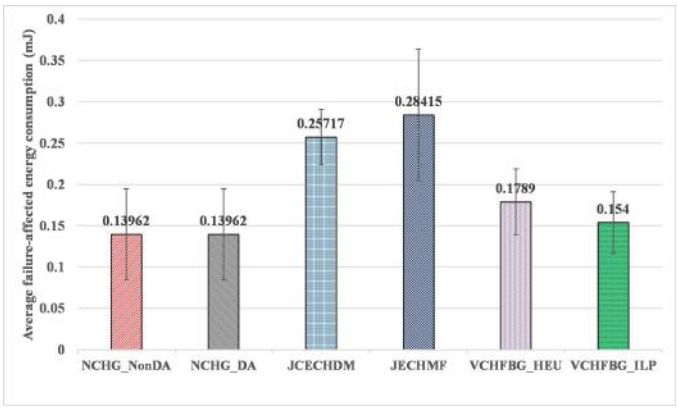


(b)

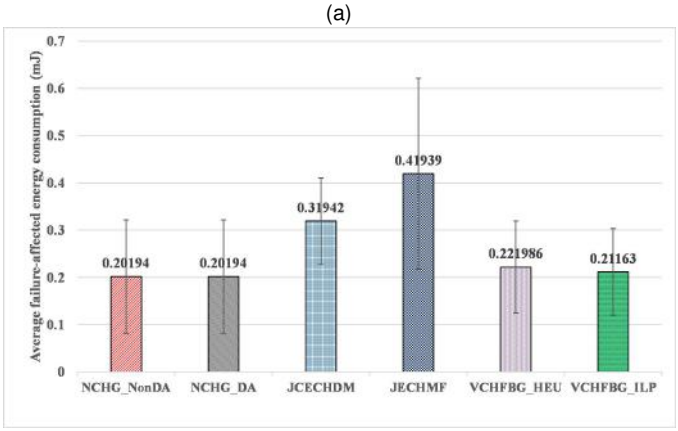
Fig. 8. Average backup energy consumption (1mJ=10^{-3}joule). (a) failure threshold=0.2 (b) failure threshold=0.5

backup node of a faulty CH needs to receive, aggregate, and transmit the sensed data of the corresponding failure-affected members. It meets the expectation that our VCHFBG_Heuristic and VCHFBG_ILP approaches have less backup energy consumption than others. The two approaches can reduce at least 84% of average backup energy consumption than others. In the two approaches, the bipartite-flow graph is first used to model the fault tolerant relationship between faulty CHs and failure-free CHs. Based on the graph, the MCF problem is solved on the graph for selecting failure-frees CHs with smaller transmission cost as fault-tolerant CHs. Due to this feature, fault-tolerant CHs of our approach can take less energy consumption. By comparing our two approaches VCHFBG_Heuristic and VCHFBG_ILP, they have similar values in the average backup energy consumption. This also means the solution of the heuristic algorithm is nealy same as that of the ILP modeling. Among six approaches, NHG_NonDA approach has the largest backup energy consumption. In this approach, the backup node is a normal member. It has not the data aggregation functionality to combine the sensed data of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2894002, IEEE Access

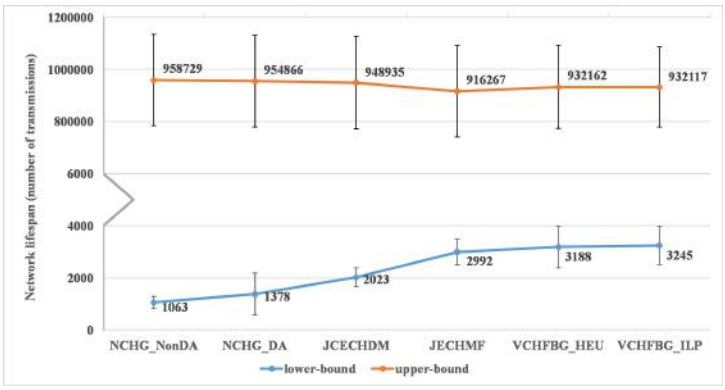


(a)

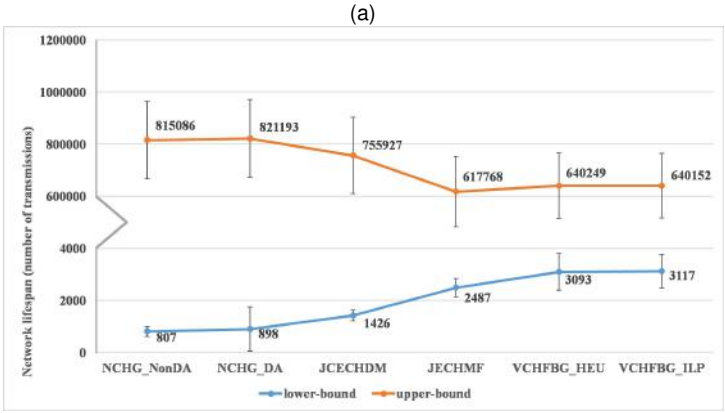


(b)

Fig. 9. Average failure-affected energy consumption (1mJ=10^{-3}joule). (a) failure threshold=0.2 (b) failure threshold=0.5



(a)



(b)

Fig. 10. Network lifetime after fault tolerance. (a) failure threshold=0.2 (b) failure threshold=0.5

multiple failure-affected members. However, if the selected backup node is a powerful member with the data aggregation functionality, the average backup energy consumption is less than JCECHDM and JECHMF. For JCECHDM and JECHMF approaches, they use existing failure-free CHs to tolerate the faulty CHs. Unlike NHG_NonDA and NHG_DA, the backup node is not selected from the cluster of the faulty CH. JCECHDM and JECHMF approaches has larger receiving energy consumption to receive the sensed data of failure-affected members. If the backup nodes of JCECHDM and JECHMF approaches has not shorter distance with the sink node, the two approaches will take larger energy consumption than NHG_DA.

From Fig. 8, we can also observe that the smaller failure threshold has less average backup energy consumption for all approaches. In smaller failure threshold, there are fewer faulty CHs. From the view point of whole WSN, there are also fewer failure-affected members. The backup CHs can take less energy consumption for perform the fault-tolerant transmissions of those members.

For the average failure-affected energy consumption, it is measured in Fig. 9. This metric is strongly dependent on the average distance between the failure-affected members and their common backup CH. In NHG_NonDA and NHG_DA approaches, the backup CH is selected from serving members of the faulty CH. The failure-affected members and the backup CH have smaller distance than other approaches since they are within the same cluster. The two approaches has smaller average failure-affected energy consumption. We can also see that these two approaches have the same values in this metric since they adopt the same backup CH selection method. Unlike the two approaches, the backup node of a faulty CH in other approaches is selected from existing failure-free CHs that are not in the cluster of the faulty CH. In our two approaches, the above concerned distance factor is particularly considered in the edge cost of the modelled bipartite-flow graph. Compared to NHG_NonDA and NHG_DA approaches, our two approaches do not increase more than 0.04mJ (1mJ=10^{-3}joule). However, our approaches have better performance than JCECHDM and JECHMF approaches. In addition, in larger failure threshold, there are more failure-affected members in each faulty cluster. This will reflect larger average failure-affected energy consumption.

Fig. 10 illustrates the lower-bound and upper-bound network lifespan comparisons in terms of the minimum and maximum numbers of transmissions offered by a backup node. As shown in the figure, our two approaches have the larger lower-bound network lifespan. After a period of time, CHs have different residual energy. In the two approaches, the modelled bipartite-flow graph can reflect different fault-tolerant capabilities of failure-free CHs. The more-energy failure-free CH should burden more fault-tolerant loads than the less-energy failure-free CH. In addition, the traffic load of a faulty CH can be distributed to two or more failure-free CHs. With these features, our two approaches can keep all failure-free CHs with higher minimum number of transmissions offered. In our two approaches, the lower-bound network lifespans are larger than other approaches.

By comparing our two approaches, VCHFBG_ILP appraoch has a better lower-bound network lifespan since it is based on ILP modeling to distribute the failure-affected transmissions in an optimal manner. In JECHMF approach, it can also avoid the failure-free CH with heavy traffic or less energy to be the backup CH. However, the traffic load of a faulty CH is taken over by a single failure-free CH. For other approaches, they may put the traffic load of a faulty CH on the failure-free CH with non-healthy status (heavy load or less energy). As a result, failure-free CHs have larger differences in the numbers of their transmissions offered. Therefore, in such approaches, there is a smaller lower-bound network lifespan. The smaller lower-bound network lifespan will cause that some clusters quickly lose data transmission capabilities. Without sufficient sensed data, the information processing of the IoT approach cannot generate precise results.

From Fig. 10, we can also see that the higher failure threshold has a smaller lower-bound network lifespan. In the higher failure threshold, more failure-affected members should be served by a failure-free CH on average. This will degrade the number of transmissions offered to obtain a small lower bound in the network lifespan.

Based on the above simulation results, we can see that our approach has the best performance in the average backup energy consumption and the lower-bound of network lifespan. For the metric of average failure-affected energy consumption, our two approaches hasve higher overhead than NHG_NonDA and NHG_DA approaches. However, the difference is not more than 0.04mJ. All the previous approaches are based on the two basic fault-tolerant methods: generation based and join based (see Section 1). Our two approaches belong to the join based fault-tolerant method. These two basic fault-tolerant methods have different trade-offs in the above three concerned metrics. It is difficult to desing an approach with the best performnace in all the three mtrics.

5 Conclusion

We have investigated the fault-tolerant cluster based routing issue in IoT WSNs. Considering the heterogeneous property for the sensor nodes of IoT WSNs, the powerful sensor node is usually with more energy and the data aggregation functionality than a normal sensor node. The powerful sensor node is also suitable to act the CH role. When a CH fails, another failure-free CH should be used to be the backup CH to continuously serve the transmissions of the failure-affected members. However, faulty CHs may have different transmission loads for their respective failure-affected members. In addition, failure-free CHs also have different residual energy. If a heavy-load faulty CH is tolerated by a less-energy failure-free CH, the original traffic load of the failure-free CH will be severely affected. To efficiently tolerate CH failures, the proposed approach used two techniques: virtual CH and bipartite-flow graph. Based on the virtual CH, we can estimate the maximum fault-tolerant capability offered by failure-free CHs. In addition, the virtual CH also makes failure-free CHs preserve the necessary energy for their original traffic loads. Then, extra resources are utilized to tolerate the faulty CHs. The bipartite-flow graph can model the fault-tolerant possibilities between faulty CHs and failure-free CHs. To solve the MCF problem on the graph, we can obtain the optimal fault-tolerant transmissions pairs with the minimum cost. Moreover, two or more failure-free CHs can commonly tolerate the faulty CH with heavy load. The simulation experiments were performed to compare the proposed approach with other approaches. The simulation results show that the proposed approach has the best performance in the average backup energy consumption and network lifespan.

6 Acknowledgment

This research was supported by the Ministry of Science and Technology, Taiwan, R.O.C, under Grant MOST 105-2221-E-030-004-MY3.

References

[1] J. A. Stankovic, "Research Directions for the Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp.3–9, Feb.2014.

[2] M. Kocakulak, and I. Butun, "An Overview of Wireless Sensor Networks towards Internet of Things," *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pp.1–6, Jan.2017.

[3] V. K. Narottam Chand, and S. Soni "A Survey on Clustering Algorithms for Heterogeneous Wireless Sensor Networks," *Int. J. Advanced Networking and Applications*, vol. 02, pp. 745–754, 2011.

[4] J. N. Al–Karaki, and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, Dec.2004.

[5] S. P. Singh, and S. C. Sharma, "A Survey on Cluster Based Routing Protocols in Wireless Sensor Networks," *Procedia Computer Science.*, vol. 45, pp. 687–695, 2015.

[6] A. Abbasi, and M. Younis, "A Survey on Clustering Algorithms for Wireless Sensor Networks," *Computer Communications.*, vol. 30, no. 14-15, pp. 2826–2841,Oct.2007.

[7] S. S. Kumar, S. MP, and S. DK, "A Survey of Energy-Efficient Hierarchical Cluster-Based Routing in Wireless Sensor Networks," *Advanced Networking and Application: Int. J. of Advanced Networking and Applications*, vol. 02, no. 02, pp. 570–580, 2010.

[8] R. H. Abedi, N. Aslam, and S. Ghani, "Fault tolerance analysis of heterogeneous wireless sensor network," *2011 24th Canadian Conference on Electrical and Computer Engineering(CCECE)*, pp. 175–179, 2011.

[9] Y. Lai, and H. Chen, "Energy-Efficient Fault-Tolerant Mechanism for Clustered Wireless Sensor Networks," *2007 16th International Conference on Computer Communications and Networks*, pp. 272–277, 2007.

[10] Md. Azharuddin, P. Kuila, and P. K. Jana, "A Distributed Fault-Tolerant Clustering Algorithm for Wireless Sensor Networks," *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 997–1002, Aug.2013.

[11] K. Nitesh, M. Azharuddin, and P. K. Jana, "Energy Efficient Fault-Tolerant Clustering Algorithm for Wireless Sensor Networks," *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, pp.234–239, Jan. 2016.

[12] H. Tavakoli, J. Misic, V. B. Misic, and M. Naderi, "Energy-Efficient Cluster-Head Rotation in Beacon–Enabled IEEE 802.15.4 Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3371–3380, Dec.2015.

[13] M. Kaur, and P. Garg, "Improved Distributed Fault Tolerant Clustering Algorithm for Fault Tolerance in WSN," *2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)*, pp.197–201,June, 2017.

[14] C. Sivaraj, P. J. A. Alphonse, and T. N. Janakiraman, "An Energy Efficient Fault Tolerance Nested Clustering Algorithm for Routing in Wireless Sensor Networks," *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, pp.1–6, Nov., 2016.

[15] M. Hezaveh, Z. Shirmohammdi, and N. Rohbani, "A Fault-Tolerant and Energy-Aware Mechanism for Cluster-Based Routing Algorithm of WSNs," *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp.659–664, July, 2015.

[16] K. Rajeswari, and S Neduncheliyan, "Genetic Algorithm Based Fault Tolerant Clustering in Wireless Sensor Network," *IET Communications*, vol. 11, no. 12,pp.1927–1932, Aug., 2017.

[17] A. Engle, and A. Koch "Heterogeneous Wireless Sensor Nodes that Target the Internet of Things," *IEEE Micro*, vol. 36, no. 6, pp.8–15, Dec.2016.

[18] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "Energy Efficient Communication Protocol for Wireless Micro Sensor Networks," *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, vol. 2,pp. 3005-3014, Jan.2000.

[19] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He, "Adaptive Transmission Power Control for Wireless Sensor Networks," *Proceedings of the 4th international conference on Embedded networked sensor systems*, pp. 223–236, 2006.

[20] M. Zhang, M. C. Chan, and A. L. Ananda, "Coverage Protocol for Wireless Sensor Networks Using Distance Estimates, *2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 1–10,2007.

[21] Z. Xu, L. Chen, C. Chen, and X. Guan, "Joint Clustering and Routing Design for Reliable and Efficient Data Collection in Large-Scale Wireless Sensor Networks, *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 520–532, Aug.2016.

[22] V. Mhatre ,and C. Rosenberg, "Design Guidelines for Wireless Sensor Networks: Communication, Clustering and Aggregation," *Ad Hoc Netw.*, vol. 2, no. 1, pp. 45–63, Jan., 2004.

[23] J. Shah,and B. Mishra, "IoT Enabled Environmental Monitoring System for Smart Cities," *2016 International Conference on Internet of Things and Applications (IoTA)*, pp.383–388,Sep., 2016.

[24] S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, *Algorithms*. McGraw-Hill, 2008.

[25] (2018) LEACH Protocol Installation in NS2. [Online]. Available: http://www.nsnam.com/2015/05/leach-protocol-installation-in-ns2-ns.html/

[26] (2018) LEACH Protocol in Linux Forum. [Online]. Available: http://www.linuxquestions.org/questions/linux-newbie-8/leach-in-ns2-35-a-4175499594/

[27] (2018) IBM ILOG CPLEX Optimizer (CPLEX). [Online]. Available: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

[28] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "Application–Specific Protocol Architecture for Wireless Micro Sensor Networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct.2002.

[29] G. Smaragdakis, I. Matta, and A. Bestavros, " SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks ," *Second International Workshop on Sensor and Actor Network Protocols and Applications (SANPA 2004)*, 2004.