

Article

Distributed Reliable and Efficient Transmission Task Assignment for WSNs

Xiaojuan Zhu ¹, Kuan-Ching Li ^{2,*} , Jinwei Zhang ¹ and Shunxiang Zhang ¹

¹ School of Computer Science and Engineering, Anhui University of Science and Technology, Huainan 554, China; xjzhu@aust.edu.cn (X.Z.); jwzhang@aust.edu.cn (J.Z.); sxzhang@aust.edu.cn (S.Z.)

² Dept. of Computer Science and Information Engineering, Providence University, Taichung 43301, Taiwan

* Correspondence: kuancli@pu.edu.tw; Tel.: +886-4-2632-8001

Received: 4 September 2019; Accepted: 10 November 2019; Published: 18 November 2019



Abstract: Task assignment is a crucial problem in wireless sensor networks (WSNs) that may affect the completion quality of sensing tasks. From the perspective of global optimization, a transmission-oriented reliable and energy-efficient task allocation (TRETA) is proposed, which is based on a comprehensive multi-level view of the network and an evaluation model for transmission in WSNs. To deliver better fault tolerance, TRETA dynamically adjusts in event-driven mode. Aiming to solve the reliable and efficient distributed task allocation problem in WSNs, two distributed task assignments for WSNs based on TRETA are proposed. In the former, the sink assigns reliability to all cluster heads according to the reliability requirements, so the cluster head performs local task allocation according to the assigned phase target reliability constraints. Simulation results show the reduction of the communication cost and latency of task allocation compared to centralized task assignments. Like the latter, the global view is obtained by fetching local views from multiple sink nodes, as well as multiple sinks having a consistent comprehensive view for global optimization. The way to respond to local task allocation requirements without the need to communicate with remote nodes overcomes the disadvantages of centralized task allocation in large-scale sensor networks with significant communication overheads and considerable delay, and has better scalability.

Keywords: wireless sensor networks; task assignment; distributed; reliable; energy-efficient

1. Introduction

Task assignment is an essential issue in wireless sensor networks (WSNs). In such multi-sensor systems, task allocation is based on the deadline and the priority of tasks, and different tasks are reasonably assigned to the sensor nodes for best perception performance. However, task allocation schemes that only aim at reducing energy consumption are not sufficient for several applications with high-reliability requirements. That is, whether tasks with high-reliability requirements in WSNs such as measurement, monitoring, and configuration can be completed in a timely manner depends on how the transmission tasks in WSNs can be reliably and efficiently assigned. As is known, energy-savings and high reliability are two conflicting goals, and it is thus challenging to assign high reliability and low energy tasks to appropriate nodes in WSNs.

In recent years, task priority assignment in WSNs has attracted the attention of researchers and satisfactory research results have been achieved. Most existing methods for practical applications are centralized algorithms, as the centralized task assignment method based on a single sink node still has the problem of poor reliability and scalability, so offline execution is selected due to computational complexity. Besides, the communication overhead and delay of task assignment will be significant in large-scale network scenarios or frequent network and dynamic changes. Therefore, it is very urgent to investigate novel distributed, reliable, and efficient task assignment methods for WSNs.

Aiming to solve the reliable and efficient distributed task assignment problem in WSNs, task-aware reliable and efficient task allocation for WSNs is investigated in this paper. The contributions of this paper are summarized as follows:

- 1) A global network view consisting of a multi-level view (including physical topology, routing topology, and task view) is constructed as a conceptual basis for global optimization;
- 2) A reliability evaluation model is established from the perspective of the task, and the constraint factors and objective functions of the task allocation are analyzed;
- 3) We propose the transmission-oriented reliable and energy-efficient task allocation (TRETA)-Cluster, in which the sink assigns reliability to all cluster heads according to the reliability requirements, and the cluster head performs local task assignment according to the assigned phase target reliability constraints. Simulation results show that TRETA-Cluster reduces the communication cost and latency of task assignment compared to centralized task allocation;
- 4) We propose the TRETA-Multi-Sink, where the global view is obtained by fetching local views from multiple sinks. Multiple sinks have a consistent comprehensive view of global optimization. The ways to respond to local task assignment requirements without the need to communicate with remote nodes overcomes the disadvantages of centralized task assignment in large-scale WSNs with significant communication overhead and delay, and have better scalability.

The remainder of this paper is structured as follows. Section 2 introduces the related work, the task allocation problem in WSNs is described in Section 3, and a global network view consisting of a multi-level view is proposed in Section 4. Next, we present a reliable and efficient task allocation algorithm for WSNs in Section 5, including centralized task allocation and two distributed task assignments. The performance is evaluated by theoretical analysis and simulation in Section 6, and finally, concluding remarks and future work are discussed in Section 7.

2. Related Work

Many research results on distributed task assignments for WSNs have been achieved, although most methods aim solely at maximizing the network lifetime [1,2]. Yu Wanli et al. propose an optimal online task assignment algorithm DOOTA (Distributed Optimal On-line Task Allocation) that is suitable for multi-partition scheduling by taking into consideration the energy cost of communication, calculation, sensing, and sleep activities [1]. To solve the problem of task allocation on the Internet of Things-based applications, considering special functions and design features, Khalil Enan et al. proposed a new task group and a virtual object-based framework and then adopted a meta-heuristic solving method [2] by modeling the problem as a single-objective optimization problem with the primary goal of minimizing energy consumption.

In addition to considering the extension of the network lifetime, part of the work also takes into account the quality of service (QoS) requirements [3,4]. Using fuzzy inference systems, Ghebleh Reza et al. proposed a method for discovering multi-criteria resource discovery in the context of distributed systems [4]. The main disadvantage of this is that the computational overhead increases rapidly as more requests and resources are invoked. Several researchers have applied game theory to WSNs in order to solve task allocation problems [5,6]. To solve the selfish behavior of some nodes, some have adopted the cooperative enforcement games strategy [7]. Despite that, the QoS requirements they satisfy are mainly focused on minimizing completion time, load balancing, data sampling rate, and accuracy, among others. Among the requirements to be considered, there is no research work on reliability requirements. Alternatively, some researchers have addressed the problem of distributed task allocation in WSNs from cloud-based architectures [8,9]. Though, due to the distance between the cloud platform and the user, the group perception has higher waiting times.

Based on the self-organizing characteristics of WSNs, Ye et al. divided the implementation techniques of distributed task allocation into two types [10]: one is based on reinforcement learning [11], and another is a collaboration based on nodes—the local interaction between nodes in a collaborative

manner to achieve a self-organizing task assignment. For example, based on the auction method [9,12] and the distributed task allocation based on the negotiation method [13], the two methods can achieve optimal results since they are obtained through negotiation between the two parties, which differs from other methods in that they use only a specific algorithm or a specific set of algorithms to obtain results. It happens that substantial communication overhead cannot be avoided during the negotiation process, so the auction-based and negotiation-based approaches are not suitable for resource-constrained WSNs.

The task assignment based on auction and negotiation has a significant communication load in a typical distributed task allocation due to the frequent interaction between nodes and unsuitable resource-limited WSNs. Moreover, the task allocation based on reinforcement learning also faces the problem of longer convergence time. In research that utilizes intelligent algorithms to solve the optimization goals of task assignment, such as particle swarm optimization, it is easy to fall into the optimal local problem. Wang et al. proposed a globally optimized task allocation algorithm based on an ant colony algorithm that requires a global pheromone matrix to obtain the optimal solution [14]. However, the authors missed mentioning how to construct a three-dimensional (3D) path pheromone storage space and the cost for the construction of this 3D space.

The research status presented above prompted the authors to investigate the distributed reliable task allocation problem in WSNs in order to develop an optimal global solution for reliable and efficient task allocation problems at a minimal cost. As is known, the transmission task is the most crucial issue in WSNs and thus the goal is to minimize the energy consumption of task allocation in WSNs based on the deadline and the reliability of tasks.

Two task allocation strategies under the constraints of reliability and task deadline are proposed in this paper, named TRETA-Cluster and TRETA-Multi-Sink. The term “task” represents the “transmission task” in the subsequent parts of this paper.

3. Description of Reliable and Efficient Task Allocation in WSNs

3.1. Energy Consumption Model

The total energy consumption of a task m_i is the sum of the energy consumed on all nodes that undertook such a task. That is, the energy consumption of a node includes computational and communication energy consumption associated with that task. For calculation purposes, the energy model depicted in [15] is used in this paper, so e_{ij} represents the energy consumed by the task m_i in node N_j . When the node N_j is not selected by the task m_i , $e_{ij} = 0$. Otherwise, e_{ij} is composed of the calculated energy consumption and communication energy consumption of node N_j . The calculated energy consumption of N_j , namely e_{ij}^{comp} , is given by Equation (1):

$$e_{ij}^{comp} = e_j \times t_{ij}^{comp} \quad (1)$$

where e_j represents the average processing energy consumption of the nodes in the network, and t_{ij}^{comp} is the calculation time of task m_i .

The communication energy consumption of a node, namely e_{ij}^{comm} , includes the energy consumption for transmitting and receiving data packets. According to the commonly used communication energy model [6], the energy consumption of transmitting and receiving data of length l_{ij} bit at the distance d is calculated by Equations (2) and (3):

$$e_{ij}^t = (e_{elec} + \varepsilon_{amp} \times d^2) \times l_{ij} \quad (2)$$

$$e_{ij}^r = e_{elec} \times l_{ij} \quad (3)$$

where e_{elec} is the energy consumption of operating the radio model for each bit, and ε_{amp} is the coefficient of the transmit amplifier. The communication energy consumption of the node is calculated by Equation (4):

$$e_{ij}^{comm} = e_{ij}^t + e_{ij}^r \quad (4)$$

Thus, the energy consumption of task m_i in node N_j is then calculated using Equation (5).

$$e_{ij} = e_{ij}^{comp} + e_{ij}^{comm} \quad (5)$$

3.2. Constraints of Task Reliability and Deadline

3.2.1. Task Deadline Constraint

Let P_i , S_i and C_i represent the task period of m_i , the task start time and the task end time (deadline) respectively. The assignment of task m_i thus needs to be satisfied with the deadline constraint:

$$C_i \geq P_i + S_i \quad (6)$$

3.2.2. Task Reliability Constraint

As is known, a WSN consists of a large number of randomly distributed sensor nodes. Performing different tasks in the same network will also show different task reliability due to factors such as network traffic and transmission path; thus, a scheme that matches the task assignment of WSNs with task reliability requirements becomes necessary. Task assignment needs to satisfy Equation (7):

$$R_{WSN} \geq R_s \quad (7)$$

where R_{WSN} represents the transmission reliability of the current WSN, whose value can be obtained from the evaluation model of reliable transmission given in Equations (8) or (10). Also, R_s represents a threshold for the application of WSN transmission reliability.

Reliability is an important indicator to measure the QoS of WSNs. It has been well studied in the past and is mainly regulated by the successful delivery rate of the link packets. Users are more concerned with the quality of the transmission of a task than the quality of the link in many applications [16]. Since a task is contained by many packets, the user is concerned whether the end-to-end event is successfully perceived rather than the successful delivery rate of the individual node's data packets. Therefore, it is necessary to measure the reliability of WSNs from the perspective of transmission and how the tasks in the network are utilized based on the analysis of the granularity.

For two typical topologies in WSNs, mesh (planar) and cluster-based (hierarchical), the cognitive data transmission process of these conventional topologies is analyzed. A task reliability evaluation model is then established based on the transmission path.

(A) Clustered Topology

Clustered topology is widely used in a variety of applications due to its higher energy efficiency and scalability. The heads of different clusters compose the backbone layer of WSNs. When evaluating a transmission task, it is supposed there are L task-related clusters in the sensing area. The reliability of clustered WSNs at time t is:

$$ClusterR_{s_i, sink}(t, i) = \prod_{c=1}^L R_A^c(t, i) \times R_T^c(t, i) \quad (8)$$

where $R_A^c(t, i)$ refers to the reliability of the i -th transmission performed by cluster c . In most clustered WSNs, a cluster is designed with one cluster head, and the transmission between cluster head and members is single-hop [17]. Since the packet delivery rate is within one hop in a cluster, the error

between the average and the actual value is smaller. Therefore, $R_A^c(t, i)$ can be modeled by a k-out-of-n system under the multisource environment.

Let $R_T^c(t, i)$ represent the reliability of the head of cluster c , which successfully sends the collected data of i -th transmission to sink at time t . This stage is considered successful as long as there is at least one path whose packet delivery rate is higher than the transmission threshold. Thus, the head of cluster c which successfully sends the collected data of the i -th transmission to sink at time t can be modeled by a parallel system.

(B) Mesh Topology

In a mesh WSN, there are multiple source nodes randomly located in the perceived area. The sink is the destination in the uplink transmission, as multiple source nodes collect data packets independently yet transmit them to the sink via intermediate nodes. The transmission reliability $MeshR_{s_i, sink}(t, i)$ in mesh WSN is shown in Equation (9), where K is the number of source nodes, and $R_{s_i, sink}(t, i)$ represents the reliability of the transmission task from the source s_i to the sink in the network at time t .

$$MeshR_{s_i, sink}(t, i) = \sum_{s_i=1}^K R_{s_i, sink}(t, i) \quad (9)$$

Let l represent the number of paths in the minimal path sets from the source s_i to sink. At time t , $R_{s_i, sink}^i(t, i)$ is the probability that there exists at least one path whose packet delivery rate is greater than the threshold of i -th transmission in the disjoint minimal path sets (from s_i to sink) is shown in Equation (10).

$$R_{s_i, sink}^i(t, i) = 1 - \prod_{j=1}^l [1 - R_{path(s_i, sink)}^j(t, i)] \quad (10)$$

3.3. Reliable and Efficient Task Allocation Problem Model

WSNs can be represented by graph $G(V, E)$. $M(t)$ represents the set of tasks to be allocated at time t , and $m_i \in M(t)$ represents the i -th task in $M(t)$. In a task allocation for WSNs, the two processes of task mapping and task scheduling are included. First, each task is mapped to the sensor node in the graph $G(V, E)$ that is represented by the function $\varphi(m_i) : M \rightarrow V$. When a task is assigned to multiple nodes, communication task scheduling is performed between the nodes, and the process of path allocation is represented by the function $\phi(m_i) : L \rightarrow E$. Therefore, the reliable and efficient WSNs task allocation problem can be abstracted into the following constraint optimization problems:

Input: $TG(V, E), RT(S, Ps), M(t)$

Output: $\varphi(m_i) : M \rightarrow V, \phi(m_i) : L \rightarrow E$

Satisfying Equation (11):

$$\begin{aligned} \min \quad & Eng\varphi(m_i) = \sum_{\tau \in V} e_{ij}, \quad m_i \in M(t) \\ \text{s.t} \quad & R_{WSN} \geq R_s, \forall \tau \in V \\ & C_i \geq P_i + S_i \end{aligned} \quad (11)$$

where $TG(V, E)$ represents the physical topology graph of the network, V represents the set of sensor nodes in the network, E represents the set of links between the nodes in the network, $RT(S, Ps)$ represents the routing topology of the network, S represents the source node set and Ps represents the path set of all source nodes to the sink in the network. $Eng(\varphi(m_i))$ is the energy consumption of task m_i that is equal to the sum of the energy consumption on all nodes in the network assigned to task m_i . From this, Equation (11) indicates that the goal of the task allocation to WSNs is to minimize energy consumption under the constraints of reliability and deadline. Given that this is a nonlinear mixed-integer programming problem, it can be solved using a heuristic algorithm.

4. Global Network View

The physical topology of a WSN, the routing topology, and the set of tasks to be allocated at time t are known as input data assigned by the task in Equation (11). Additionally, the sensor nodes in a WSN learn local topologies through topology discovery, which are periodically sent to sink nodes to form the global physical topology of the network.

4.1. Route Topology

We proposed the route topology inference (RTI) in [18]. The algorithm framework is shown in Figure 1. Since this method is not limited by the routing protocol adopted by the current network, and only uses the packet tracking hybrid active detection method to construct a transmission path from the source node to the sink, an online global routing topology view for a WSN can be provided and shown.

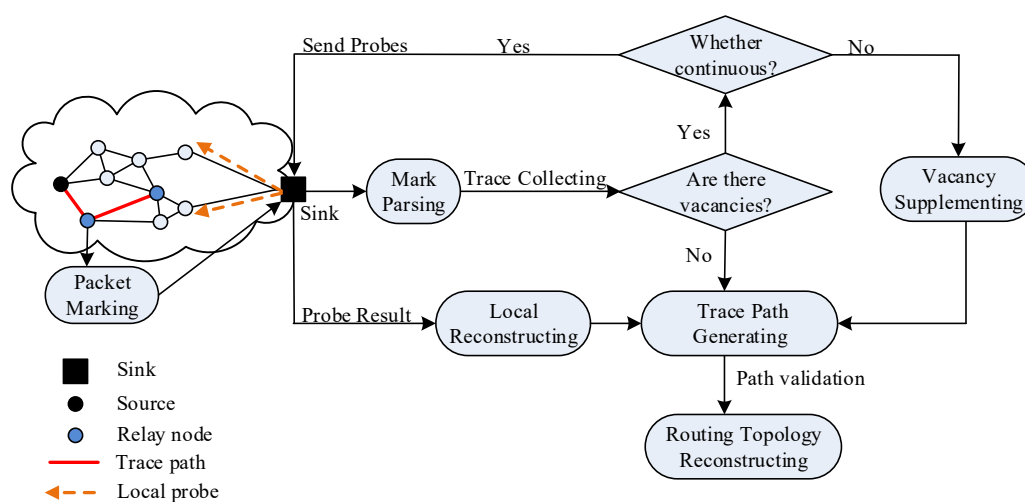


Figure 1. The overall framework of route topology interference (RTI).

The core idea of packet tracing is that, on the transmission path from the source node to sink, the forwarding nodes selectively mark part packets according to the marking rule upon receiving the packets. At the end of the sampling period, the sink can establish trace lists from the source node to the sink by parsing the marked packets. As the sink determines the one-to-one correspondence trace in the trace list according to the packet with the same hop to the source node, it retrieves the trace list from the source node. The trace list returned by mark parsing may be incomplete due to packet loss or insufficiently marked packets. Also, the vacant traces in the trace list can be supplemented by auxiliary inference or active detection [18].

RTI increases the memory load, though it has a significant advantage in the correctness and convergence of reconstruction. Moreover, the relay node does not need to mark all packets, as it only marks the packets based on conditions, and for one path, each packet is marked only by one relay node.

The reconstructed WSN routing topology is shown as a graph, where the nodes on each path and links between the nodes are added to the graph, and the source node identifying a path is added to the link also. If a source node has multiple paths to the sink, the remaining nodes or edges are added to the routing topology based on the tracking path after the first path is added, so the marked routing topology view from the source node to the sink is generated.

4.2. Multi-Level Global View

Once the routing topology is acquired, the task logical topology of different tasks in the current network is further abstracted according to the task mark in the data packet. As shown in Figure 2, the architecture that provides a conceptual basis for global optimization of network management

and the multi-level global view architecture consisting of the physical topology, the routing topology, and the task logical topology is obtained. The routing topology layer can generate different routing topologies for applications according to different node sets in the physical topology layer (i.e., different shadows in Figure 2). Also, links of different thicknesses in the task logic topology reflect the current traffic of the link.

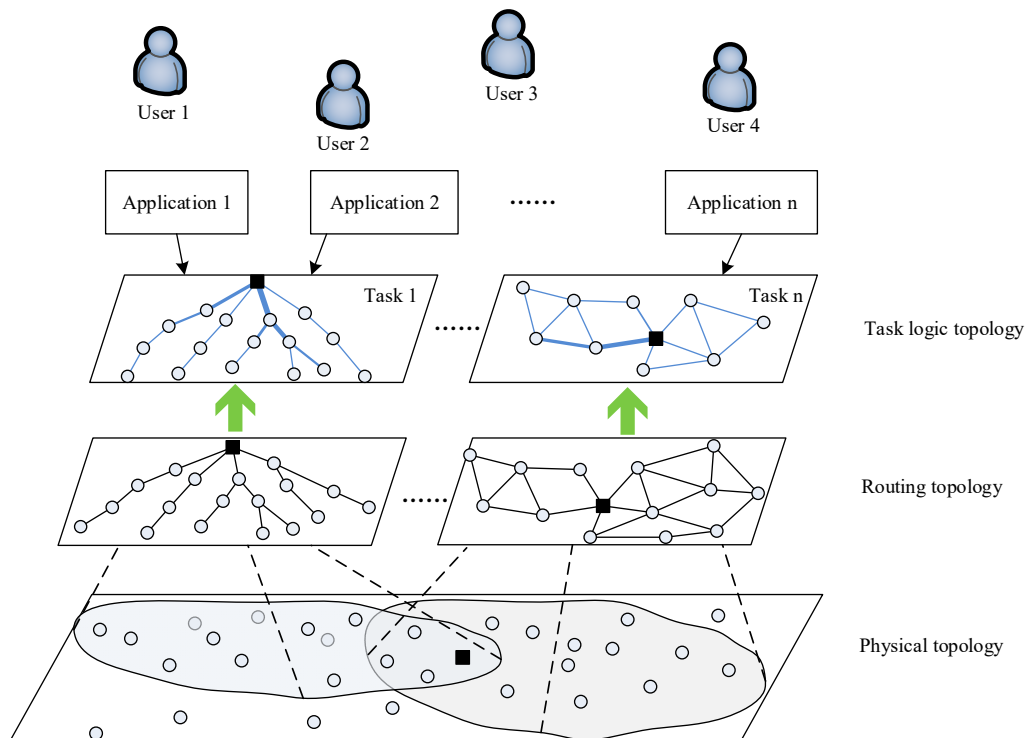


Figure 2. The multi-level global view of a wireless sensor network (WSN).

5. Reliable and Efficient Task Allocation Algorithm for WSNs

5.1. Centralized Task Assignment

As presented before, the aims at this research are to minimize the energy consumption of task allocation under the constraints of reliability and deadlines in the task allocation model problem presented in Section 3, where $TG(V, E)$, $RT(S, Ps)$ and $M(t)$ are the three inputs as presented in Equation (11).

TRETA is performed by the sink based on the given task reliability and deadline constraints through a global view, and its framework consists of five modules, enumerated as: data collection, task reliability evaluation, topology management, event management, and task assignment. The data collection module collects state information of each node, such as node ID (Identity Document), node residual energy information, node neighbor, and packet loss rate information, among others; the node collection module stores network state information such as nodes, paths and task information (transmitted from the application layer) in the corresponding node table, topology information table, state information table, and task information table in the database for extraction by other modules; and the topology information table in the data collection module stores the global physical topology of the network.

To establish a global view of the current network state, a topology management module in the framework of TRETA is designed. In addition to the global physical topology, the topology management module obtains the routing topology of the global network and then constructs a two-level global view.

The event module manages events and triggers other module updates when a driver event occurs. To achieve reliable and efficient task allocation in WSNs, the driver events concerned include topology

change (nodes join or leave), as the reliability of the task is lower than the threshold and new tasks to be allocated.

Based on the modules presented above, a reliable and efficient task allocation algorithm, namely TRETA, is proposed, as depicted in Figure 3.

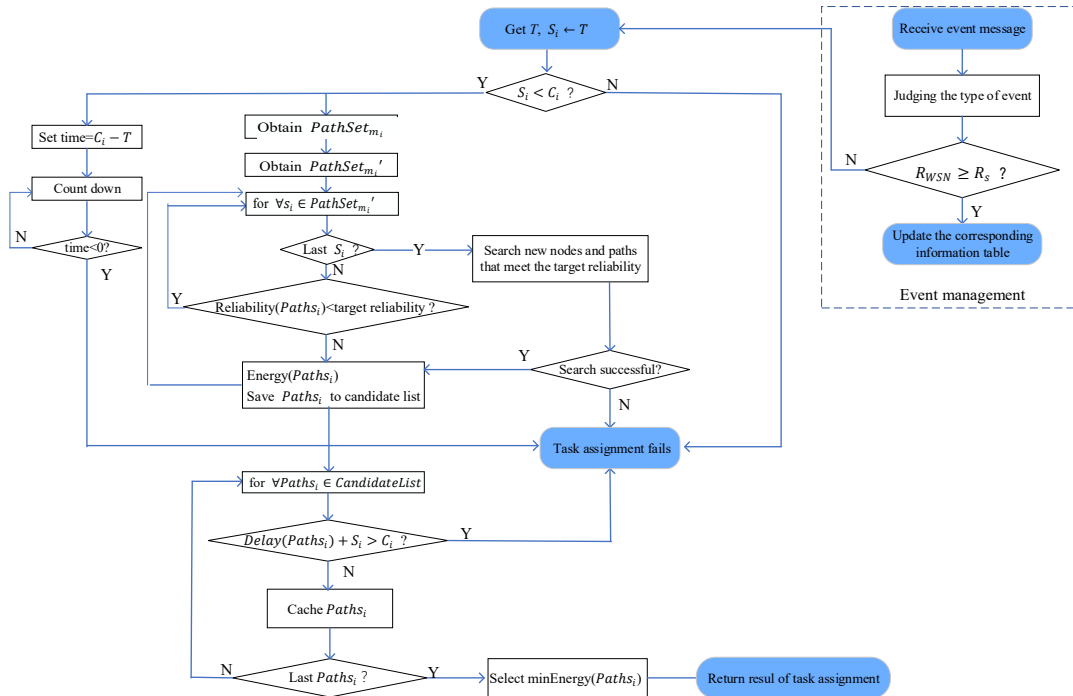


Figure 3. Transmission-oriented reliable and energy-efficient task allocation (TRETA) algorithm.

During the execution of the network and once the defined driving events occur, the event management module triggers the update of the data collection module, and the sink determines whether the reliability of the updated network is less than the target reliability. If yes, the task allocation module is notified to re-assign; otherwise, only the global view corresponding to the event is updated.

5.2. Distributed Task Assignment

Since TRETA performs task assignments from the perspective of global optimization, it has the disadvantage of significant communication overhead and delays in large-scale WSN. Aiming to obtain reliable and efficient distributed task allocation in WSNs, two distributed task assignments for WSNs based on TRETA are proposed.

5.2.1. Distributed Reliable and Efficient Task Allocation in a Hierarchical Topology

A hierarchical topological diagram of a clustered WSN, where the sink is the center of the entire network, is depicted in Figure 4a. The cluster head saves the collected local physical routing topology as well the state information of the member nodes in the local cluster, passing them next to the sink. Then, the sink node can obtain the physical topology of the entire network by merging all the topologies received, so the topology management module receives a two-level global view.

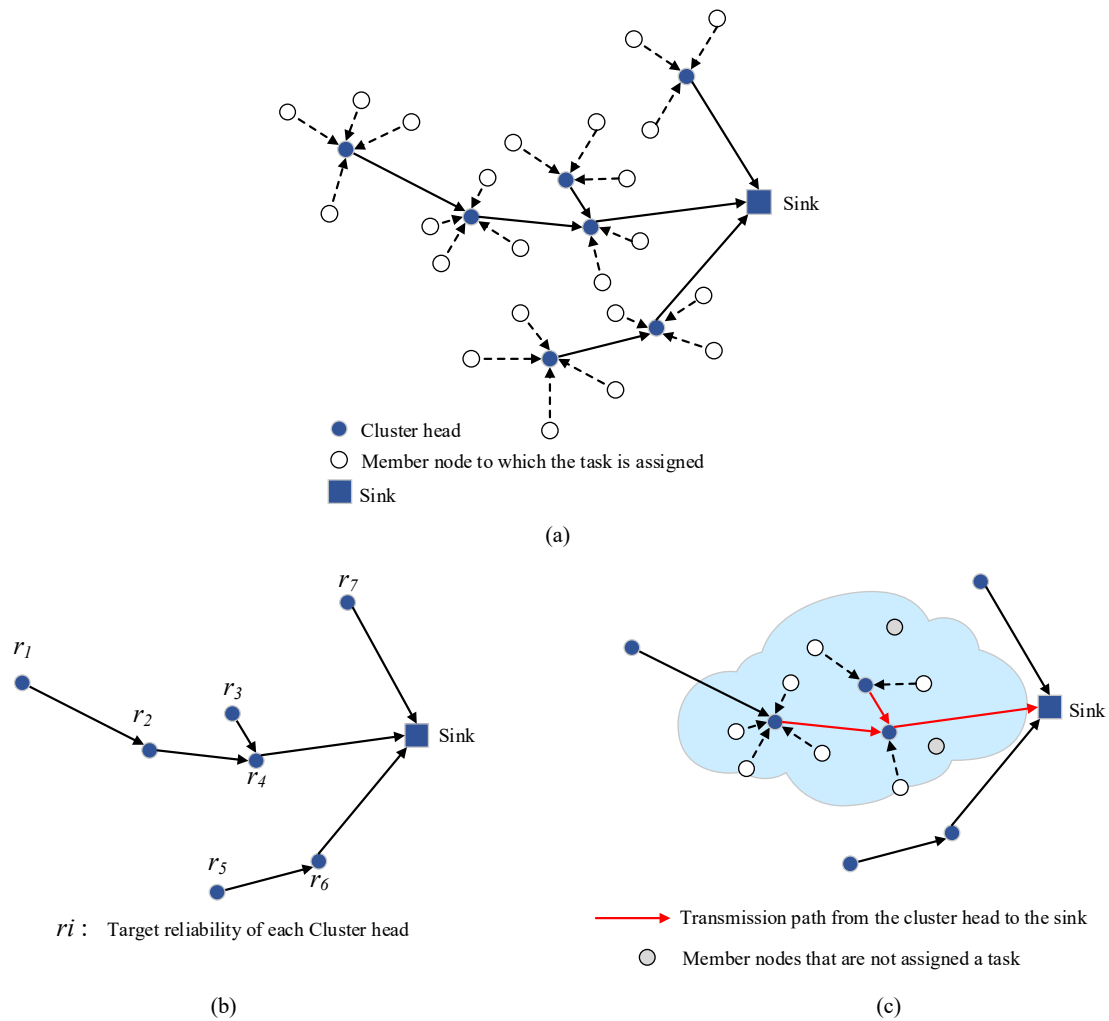


Figure 4. Reliable and efficient distributed task allocation based on clustering: (a) Cluster-based WSNs; (b) Sink assigns reliability to the cluster head; (c) Cluster head performs local task allocation.

Based on the above observations, we present a distributed reliable and efficient task assignment algorithm based on TRETA for clustered WSNs, named TRETA-Cluster, which divides the task allocation into two phases: backbone network and intra-cluster allocation.

- 1) The sink performs the reliability distribution at the backbone network composed of the cluster heads according to the target reliability, where reliability r_i is obtained at each cluster head, as shown in Figure 4b. Combined with the target reliability and the task deadline constraint, the sink calls the TRETA algorithm to select the cluster head with the smallest total energy consumption that satisfies the task deadline and the target reliability constraint, and the transmission path of the cluster head to the sink.
- 2) The cluster head calls the TRETA to select the nodes in the cluster and the intra-cluster paths according to the obtained intra-cluster reliability index and local view to realize the task allocation of the cluster.

If the task allocation is successful, the allocation result is returned; otherwise, the cluster head selects the node that satisfies the target reliability according to the local physical topology. In case the node is found, the allocation result is returned. The task allocation fails and returns if otherwise, as shown in Figure 4c.

As a topology change occurs, the reliability may become lower than the threshold or the cluster head can be interrupted. In this case, the task allocation is re-executed by the sink or the cluster head according to the rule, as seen in Algorithm 1, TRET-Cluster, Lines 17–20.

The reliability allocation [19,20] is to assign the target reliability of a task to the appropriate subsystems, components, and nodes of the system to determine the reliability of each component. Referring to the reliability allocation algorithm in [21], we assign the target reliability of the task to the two stages and use them as the target reliability of each stage (Algorithm 1, TRET-Cluster, Lines 5–6; RA is the reliability distribution function).

Algorithm 1. TRET-Cluster

 Input: $TG(V, E)$, $RT(S, Ps)$, $M(t)$, R_s , C_i

 Output: $\varphi(m_i) : M \rightarrow V$, $\phi(m_i) : L \rightarrow E$

```

1 for  $\forall m_i, m_i \in M(t)$ 
2   get  $T, S_i = T$ 
3   if  $S_i < C_i$ 
4     for  $\forall CH, CH \in RT(S, Ps)$  //CH represents the cluster head
5        $R_{CH} = RA(RT(S, Ps))$ 
6        $R_{CM} = RA(RT(S, Ps))$ 
7     for  $\forall CH \in RT_b(S, Ps)$ 
8       call TRET ( $TG_b(S, Ps), RT_b(S, Ps), R_{CH}$ )
9     for  $\forall$  Cluster  $\in RT(S, Ps)$ 
10      call TRET ( $TG_c(S, Ps), RT_c(S, Ps), R_{CM}$ )
11      if call TRET ( $TG_c(S, Ps), RT_c(S, Ps), R_{CM}$ )  $\neq \emptyset$ 
12        return allocation result
13    get  $T$ 
14    if  $T > C_i$ , return line 1
15    if  $CH$  fault
16      sink update  $TG(V, E)$  and  $RT(S, Ps)$ 
17      if  $R_{WSN} < R_s$ , then
18        call TRET ( $TG_{b'}(V, E), RT_{b'}(S, Ps), R_{CH'}$ )
19        elseif the failed cluster is re-clustered
20        sink update  $TG(V, E)$  and  $RT(S, Ps)$ 
22    if member of  $CH$  change
23      sink update  $TG(V, E)$  and  $RT(S, Ps)$ 
24      if  $R_{WSN} \geq R_s$ , then return
25      else
26        call TRET ( $TG_{c'}(V, E), RT_{c'}(S, Ps), R_{CM}$ )
27        if call TRET ( $TG_{c'}(V, E), RT_{c'}(S, Ps), R_{CM}$ )  $\neq \emptyset$ 
28          return allocation result
29        else return line 1
30    if  $R_{WSN} < R_s$ , then
31      return line 1
32    return
  
```

It can be noted that the local task allocation of the cluster head may select the node participating in the task, and since the path from the cluster head to the sink has been obtained in the first stage, the selection from the node to the path is completed in two phases in a task assignment.

Whenever the nodes in the cluster change (join or leave), the cluster head transmits the topology change to the sink. The sink determines whether the node change affects the reliability of the task. If not, it returns. Otherwise, the cluster head node tries to re-select the node in the cluster after the

change. Additionally, if the intra-cluster target reliability constraint can be met, the cluster head re-performs the local task assignment. Otherwise, the sink performs the task allocation again.

When the sink finds that the task transmission is below the reliability threshold, the task allocation is performed again. However, when the cluster head fails, the cluster will disconnect from other parts of the network and the task assignment policy issued by the sink cannot reach the cluster, bringing challenges to the reliable task allocation of WSNs. The proposed processing scheme follows: the sink updates the two-level global view and determines whether the fault of the cluster has an impact on the target reliability of the current task. If so, the task is reassigned according to the updated two-level global view. Otherwise, the sink only updates the two-level global view that reduces the frequency of the update. After the faulty cluster re-selects the cluster head, the cluster head collects the cluster topology and the sink node updates the global view and re-performs the task assignment.

5.2.2. Distributed Reliable and Efficient Task Allocation in Planar Topology

For a large-scale WSN, perceptual information needs to go through multi-hop communication to the sink. Considering the overhead of establishment, maintenance routing by node, and long-distance multi-hop communication delay in planar topology to ensure the reliability and scalability of WSNs, some researchers have proposed the multi-sink deployment scheme [22–25] or software-defined network (SDN) controller [26].

The deployment strategies of multiple sinks in a planar topology include static and dynamic, random, and scheduled deployment methods. As an illustration, the static and mobile deployment of multiple sinks is shown in Figure 5.

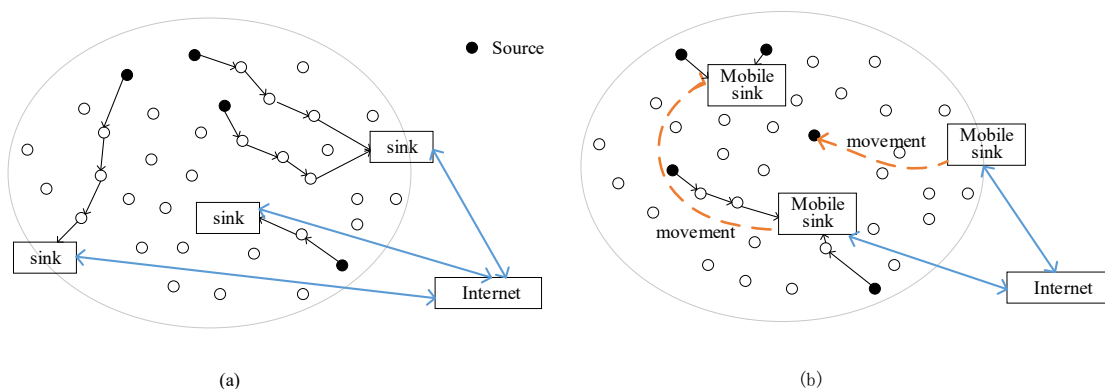


Figure 5. Multi-sink deployed WSNs. (a) Static multi-sink WSNs; (b) mobile multi-sink WSNs.

In a WSN with multiple sinks deployed, this section proposes a distributed task assignment strategy based on multiple sinks. Multiple sinks in this strategy have a global view of the entire network, enabling local requirements to be globally optimized and without the need to communicate with remote nodes, reducing the communication cost and latency of task assignments.

(A) Consistency of Multi-Sink Global View

Due to the geographical distribution and asynchronous operation features, when the sink publishes the global view, the update time of multiple sinks is not synchronized due to network delay or other reasons, resulting in inconsistency in the forwarding or processing of the task data. To share the global view among multiple sinks, it is necessary to consider the consistency problem of the global views between multiple sinks. In a resource-constrained WSN, how to efficiently share global view among multiple sinks to achieve fast and efficient global task allocation is still one of the significant questions.

In Eric Brewer's consistency, availability and partition tolerance (CAP) theory, it has been proved that consistency, availability, and partition tolerance in a distributed system cannot be considered together [27]. Partition occurs easily in WSNs. If strong consistency is guaranteed, the availability of the network cannot be guaranteed at the same time, and the communication cost of realizing the

strong consistency of multiple sink global views in large-scale WSNs is high, the final consistency is chosen to reduce the communication cost of synchronization between nodes.

In an existing final consistency technology, Dynamo, the storage platform of the key-value pattern of Amazon has been paid more attention by many researchers [28,29]. Dynamo proposes an NWR model that guarantees eventual consistency, where N represents the number of copies of data being saved, R represents the number of copies required for each read success, and W represents the number of copies as are necessary for each write success. By setting R and W , when $R + W > N$, it produces a system similar to Quorum.

Quorum is widely used in distributed storage systems [30–33]. It is a set and a subset of all copies C , where two parameters W and R are pre-defined for N copies, and $N = |C|$, $W \geq 1$, $R \leq N$. The quorum set of writing is as follows:

$$S_W = \{Q | Q \subseteq C \wedge |Q| = W\} \quad (12)$$

and the quorum set of reading as:

$$S_R = \{Q | Q \subseteq C \wedge |Q| = R\} \quad (13)$$

If $R + W > N$, there is an intersection between S_W and S_R , so any read operation can return the latest write and guarantee strong consistency. Conversely, if $R + W \leq N$, then the two elements overlap with probability $P_{overlap}$ [29], as in Equation (14):

$$\begin{aligned} P_{overlap} &= 1 - \frac{\binom{N}{W} \binom{N-W}{R}}{\binom{N}{W} \binom{N}{R}} \\ &= 1 - \frac{\binom{N-W}{R}}{\binom{N}{R}} \end{aligned} \quad (14)$$

For a write request, it is first sent to the replica as a coordinator, and then the coordinator propagates the write to all other replicas. After getting at least $W - 1$ responses, the coordinator returns. Read requests are handled in the same way as write requests, although the coordinator should wait for at least $R - 1$ responses. If the read replica set overlaps with the write replica set, the read request can return the most recent value. Otherwise, it will return stale data.

Similarly, this paper proposes a method to ensure the final consistency of multiple global views. In this method, each sink maintains a log in addition to the global view that includes the version of the current global view (new), the version of the previous global view (old), and the update operation from the old version to a new version.

In a multi-sink WSN, if the network status changes, the changed local sink sends an update request to other sinks. If at least $W - 1$ sinks return a response, the update request is considered successful, and the sink submits an update and broadcasts the update to other sinks. Next, the previously responded $W - 1$ sinks update the current global view version, the previous version of the global view, and the update operation from the old version to the new version in the log. As soon as a sink node initiates a task assignment, the global view needs to be read. In addition to reading the locally saved global view, the sink node issues a read request to the other sinks, then waits for $R - 1$ responses, and finally returns the read result.

Since the task allocation has a deadline constraint, to reduce the delay of reading the global view, the algorithm sets $R = 2$ to read the global view of two sinks. Therefore, the complexity of guaranteeing consistency is pushed to the write operation; that is, for the network state update process which is not sensitive to delay, as long as $R + W > N$ is satisfied, the final consistency can be guaranteed. If concurrent partial updates occur in the network at the same time, the concurrency control protocol is used for processing.

(B) Task Assignment Based on Multi-Sink WSNs

In a planar WSN topology where multiple sinks important, these sinks form a multicast group, and the multiple sinks are equal. To simplify the design, it is assumed that multiple sinks can remain fully connected and the sinks do not fail, as each sink collects the local view, including the physical topology and routing topology. The local views are periodically exchanged so that multiple sinks in the multicast group gradually obtain a global view of the network.

The concept of TRETAMulti-Sink follows: after each sink has acquired the global view and there is a task to be allocated, if the current time of the system is less than the deadline of the task to be assigned, the sink closest to the task can be selected to call the TRETAMulti-Sink algorithm for local optimization allocation. Since the sink has a global view and despite a local allocation, the sink can perform a globally optimized task assignment. In the algorithm, multiple sinks select the final consistency when exchanging global views to reduce the communication cost of synchronization between sinks. If the current time of the system is higher than the deadline of the task to be assigned, the current task allocation is interrupted.

Specifically, for the task m_i to be assigned, if the current time of the system is less than the deadline of the task, the nearest sink of task m_i is found through the physical topology information and is represented by $sink_j$, which reads the local and adjacent global view version. To ensure the final consistency of the global view between multiple sinks, the reading number of copies is R according to the NWR model (we let $R = 2$); if the versions are different, it will read the global view with the latest version. Otherwise, it will read the local - global view of $sink_j$. Then, $sink_j$ calls the TRETAMulti-Sink algorithm for local optimization allocation. During this process, if the current time of the system exceeds the task deadline, the task allocation is interrupted.

When the network update occurs, if the scope of the update involves only one sink (represented by $sink_k$), then $sink_k$ multicasts the update request to other sinks. According to the NWR model, the number of copies that need to be written to the update is $W - 1$. W in this paper should satisfy $W > N - 2$.

If the number of sinks in response to the update request is higher than $W - 1$, $sink_k$ submits the view update, and other sinks in the multicast group update the local-global view accordingly. Otherwise, the update of $sink_k$ is deleted to avoid inconsistency with other different sink views.

When the sink finds that the task transmission is lower than the reliability threshold, the task allocation is re-executed. The specific process of Algorithm 2, TRETAMulti-Sink is as follows:

Algorithm 2. TRETA -Multi-sink

```

Input:  $TG_j(V, E)$ ,  $RT_j(S, Ps)$ ,  $M(t)$ ,  $R_s$ ,  $C_i$ 
Output:  $\varphi(m_i) : M \rightarrow V$ ,  $\phi(m_i) : L \rightarrow E$ 
1  $TG(V, E) = \text{Merge}(TG_j(V, E))$ ,  $j = 1, 2, \dots, n$ 
2  $RT(V, E) = \text{Merge}(RT_j(V, E))$ ,  $j = 1, 2, \dots, n$ 
3 for  $\forall m_i, m_i \in M(t)$ 
4   get  $T$ ,  $S_i = T$ 
5   if  $S_i < C_i$ 
6      $sink_j = \{sink | \text{Mindistinct}(m_i, sinks), sink \in TG_j(V, E)\}$ 
7     read  $sink_j.viewID$  and  $sink_{j_n}.viewID$ 
8     if  $sink_j.viewID \neq sink_{j_n}.viewID$ 
9       read view of latest( $sink_j.viewID, sink_{j_n}.viewID$ )
10    else read  $sink_j.RT_j(S, Ps)$ 
11    call TRETA ( $TG_j(V, E)$ ,  $RT_j(S, Ps)$ ,  $M(t)$ ,  $R_s$ )
12    get  $T$ 
13    if  $T > C_i$ , return line 1
14    if  $TG(V, E)$  change happens and  $Num(\text{changed sink}) = 1$ 
15       $sink_k$  record change in cache
16       $sink_k$  multicast update-request to other sinks
17      if  $Num(sink_k^{res}) > W - 1$ 
18         $sink_k$  submit an update
19        update( $sink_k.TG_j(V, E)$ ,  $sink_k.RT_j(S, Ps)$ ,  $sink_k.log$ )
20        update( $sink_k^{res}.TG_j(V, E)$ ,  $sink_k^{res}.RT_j(S, Ps)$ ,  $sink_k^{res}.log$ )
21      else delete  $sink_k.record$ 
22    if  $R_{WSN} < R_s$ , then
23      return line 3
24 return

```

6. Performance Analysis**6.1. Theoretical Analysis**

In this section, the performance of distributed reliable and efficient task allocation is analyzed primarily from the algorithm complexity, communication load, and delay of the task assignment.

6.1.1. Algorithm Complexity

TRETA-Cluster is divided into two phases. In the first phase, the sink performs task allocation for the backbone network composed of the cluster heads and the sink, with the worst-case calculation complexity $O((\alpha N)^3)$, where α represents the ratio of the size of the backbone network to the size of the entire network system. During the second phase, the cluster head is responsible for the task allocation within the cluster. The calculation complexity of this process is $O((\gamma N)^3)$, where γ represents the ratio of the cluster member nodes size to the entire network size. Based on the above process, the calculation complexity of distributed task allocation based on clustering topology is $O(N^3)$.

In TRETA-Multi-Sink, the calculation complexity of multiple sinks in local task allocation is the same as TRETA, although the processing of global view consistency is added to the network update in TRETA-Multi-Sink, which includes the write operation of multiple sinks when global view update occurs and read operation when task assignment occurs. Assuming N is the network size, β is the ratio of the part of the network to be updated, and W is the number of global views that need to be written, the calculation complexity of the write operation is $O(\beta WN)$. Let the number of global views that need

to be read is R ($R = 2$); then, the calculation complexity of the read operation is $O(2N)$. Based on the above process, the calculation complexity of the distributed task allocation algorithm is $O(N^3)$.

The two distributed task allocation algorithms above reduce the size of the network. As noted, the distributed task allocation has the polynomial computation complexity in the background of global optimization.

6.1.2. Communication Load of the Distributed Task Assigned

In the distributed task allocation strategy for the clustered topology, the sink multicasts the allocation policy to the selected cluster heads that are responsible for the task allocation within the cluster. Therefore, the communication load depends on the amount of information in the sink's allocation policy. In this paper, the communication load of TRET-Cluster is measured by the number of packets in the allocation policy by multicast, as shown in Equation (15):

$$CM = Num_{cp} \quad (15)$$

For the multi-sink topology, the communication load includes: (1) multi-casting the global view between the sinks, namely Num_{gv} , where Num_{gv} represents the communication load for transmitting one global view, and a represents the number of times of global views transmitted; (2) ensuring global view consistency between multiple sinks, i.e., the write load WM at update time and the communication load RW at read time, as shown in Equation (16):

$$DM = aNum_{gv} + WM + RW \quad (16)$$

where only the version and update operation of the global view is transferred when the update is written, and performed only when the network is updated, as shown in Equation (17):

$$WM = \begin{cases} VUM \\ 0 \end{cases} \quad (17)$$

where VUM indicates the write load of the version and update operation of the global view when it is updated, and 0 means no update.

When the sink reads the global view, it needs to read two copies of the global view; that is, the global view of the local and that of the nearest sink in the network, as required by the final consistency. If the local version of the global view is newer, there is no need to communicate with other sinks, and thus the communication load of the read is 0. Otherwise, one global view needs to be transmitted. Therefore, the communication load RW when reading is as shown in Equation (18):

$$RM = \begin{cases} 0 \\ Num_{gv} \end{cases} \quad (18)$$

6.1.3. Delay of Distributed Task Assignment

In order to implement the task assignment, two known premises are required: the physical topology of the entire network and the routing topology of the network established in advance. They are considered in the initialization process for the task assignment, so the delay is not considered at initialization.

The task assignment delay for a clustered topology is calculated by Equation (19):

$$RT_{d_c} = RT'_c + MC + CHD \quad (19)$$

where RT_{d_c} represents the delay of task assignment for the clustered topology, RT'_c represents the delay of the sink allocating a task to the subgraph composed of the cluster heads, MC represents the

delay of the multicast performed by the sink, and CHD represents the delay in the allocation of the task by the selected cluster head.

Assuming that each sink has obtained a global view in a multi-sink deployment environment, this stage can be implemented by periodically swapping local views with each other through multiple sinks, requiring it as initialization prior to the task assignment. The delay for distributed task allocation based on multiple sinks is calculated by Equation (20):

$$RT_{d_d} = \overline{RT'_c} + C_{rt}. \quad (20)$$

where RT_{d_d} represents the delay of TRETAMulti-Sink, $\overline{RT'_c}$ represents the average of the delays of the tasks assigned by multiple sinks locally, and C_{rt} indicates the time to ensure the final consistency of the view, as shown in Equation (21):

$$C_{rt} = \begin{cases} W_{rt} \\ R_{rt} \end{cases} \quad (21)$$

where W_{rt} indicates the time of writing of the global view whenever it is updated and R_{rt} indicates the time of reading of the global view when the distributed task is assigned.

6.1.4. Energy Resilience of the Transmission in WSNs

The nodes can go down for various reasons, e.g., the time of life or for the specific protocol used. In order to evaluate the energy resilience of the transmission in WSNs, we quoted an index η [34] as shown in Equation (22), where N_p indicates the number of packets received by the sink after a fixed time, S indicates the number of initial active nodes, and D indicates the number of dead nodes after a fixed time.

$$\eta = \frac{N_p}{S - D} \quad (22)$$

The index gives an indication of how efficient the network is in allowing information to be delivered considering both the number of packets that are running in the network and the number of nodes that are going out over time.

6.2. Simulation

6.2.1. Simulation Design

In order to carry out a reliable and efficient task assignment, two distributed task assignment strategies are proposed, namely TRETACluster and TRETAMulti-Sink. In this section, a simulation is carried out using TOSSIM (TinyOS Simulator). In order to evaluate energy consumption, this paper expands TOSSIM and adds a power consumption model. The performance of the strategies is analyzed and compared in three aspects: different network size, task arrival rate, and network update rate.

In order to compare with the distributed task allocation, simulation of the centralized task allocation TRETACluster is also performed in the same network environment. The performance metrics analyzed include the energy allocated, the delay of successful allocation during the deadline, and the success rate of the task assignment. In addition, the simulation is carried out under the cluster topology and the multi-sink-based topology for the distributed task assignment.

1) Cluster topology

The sensing area is 100 m × 100 m, and the number of sensor nodes is 50 to 300. There is only one sink in the cluster topology, with the sink node located in the center of the sensing area and remaining nodes randomly deployed.

2) Multi-sink-based topology

The deployment strategies of multiple sinks in a multi-sink-based topology include static and dynamic, random, and scheduled deployment methods. In the proposed simulation experiment,

the static uniform deployment method is selected, with the number of sinks set to five, and the sensing area of the square is evenly divided into four sub-areas. One of the five sinks is located in the center of the sensing area, while the remaining four sinks are located in the center of the respective remaining four sub-areas. The five sinks remain fully connected, and the case where the sinks fail is not considered.

The main simulation parameters are shown in Table 1. Three of the abbreviations (e_{elec} , ϵ_{amp} , and e_j) represent the energy consumption of operating the radio model for each bit, the coefficient of the transmit amplifier, and the average calculated the energy consumption of the nodes in the network, respectively. Kenneth et al. mentioned that wireless communication is usually the most energy-consuming process in traditional WSN applications [16]. Specifically, the energy required for a single bit transmission is 1000 times the energy consumed by calculating a single bit in classic 32-bit architecture, and thus the value of e_j is set to 0.05 nJ/b.

Table 1. Main simulation parameters.

Parameter	Value
Bandwidth	250 kbps
e_{elec}	50 nJ/b
ϵ_{amp}	100 pJ/b/m ²
Maximum transmission range of the node	50 m
The initial energy of the node	2 kJ
e_j	0.05 nJ/b

The parameters of the task to be assigned are shown in Table 2, and three aspects are included: task type, transmission parameters, and task environment. Each simulation is executed for 120 minutes and repeated 500 times. For each performance index, we used the Monte Carlo method to obtain simulation results. The specific process is as follows: first, in the TOSSIM simulation environment, Python is used to generate a random number of each task parameter in the value interval based on a Poisson distribution. Secondly, each random variable is directly sampled, and simulation experiments and calculations are performed according to the task assignment strategy, and the optimal strategy of task assignment is obtained. Finally, statistical analysis is performed on the test results to obtain the average value of each evaluation index of the task assignment strategy.

Table 2. Task parameters to be assigned.

Parameter	Value
Transmission direction	Upstream
Task interval	10 s
Task duration	[100 s, 300 s]
Target reliability	90%
Packet size	500 bytes
Number of packets per task	[100, 1000]
Packet loss rate	0–10%
Topology change rate	0–10%

6.2.2. Simulation Results and Analysis

The performance evaluation indicators in this research include the energy consumption of task assignment, the delay of successful allocation within the deadline, and the success rate of the task assignment. The success rate of task assignment represents the ratio of the number of tasks successfully assigned to the total number of tasks to be assigned in a simulation cycle, while the other two indicators are calculated according to Equations (5), (19), and (20). For comparison purposes, the simulation of centralized task allocation in the same network environment is conducted and represented by TRETA.

(A) Energy Consumption for Task Assignment

It is shown in Figure 6, the comparison of energy consumption among the three allocation strategies is proposed: the centralized task allocation TRETA and the distributed task assignment strategies TRETA-Cluster and TRETA-Multi-Sink. As seen in this figure, the energy consumption increases with the increase of node size. Among them, the energy consumption of TRETA is highest, since the global state of the network is converged to a single sink through long-distance multi-hop in large-scale WSNs. On the other hand, the energy consumption of TRETA-Multi-Sink is relatively small, as it is based on the multi-sink being able to collect the local state and merge it into the global view by multicasting. TRETA-Cluster is performed in two stages. When the task cannot meet the target reliability, if the reselected node in the cluster can reach the stage target reliability, the sink does not need to re-allocate the task, so the energy consumption of TRETA-Cluster is less than both TRETA and TRETA-Multi-Sink.

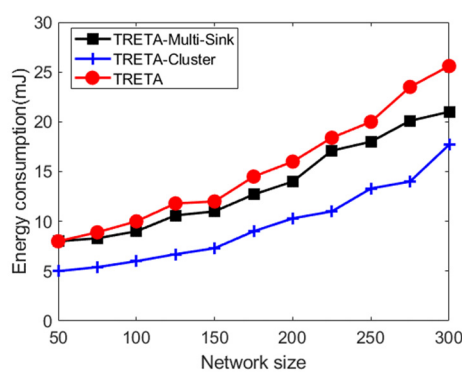


Figure 6. Distributed task assignment energy consumption.

(B) Delay of Successful Assignment within the Deadline

The delay of task assignment in the size of the network from 50 to 300 nodes is analyzed. As can be seen in Figure 7, TRETA has a substantial delay. Due to long-distance multi-hop aggregation to a single sink, achieving the global state of the cluster topology causes considerable delays in large-scale WSNs.

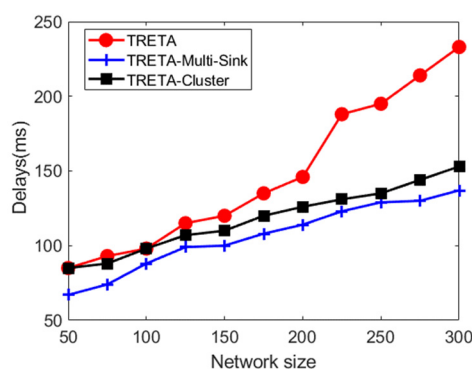


Figure 7. Task allocation delay.

(C) The Success Rate of Task Assignment

The success rate of task assignments under different task arrival rates is analyzed. Such a rate represents the number of tasks waiting to be allocated per second, and the success rate of the task allocation represents the ratio between the number of successfully assigned tasks in one simulation cycle and the total number of tasks to be assigned. As seen in Figure 8, the success rate of the task allocation decreases as the task arrival rate increases, and the rise in the task arrival rate leads to the

increase in link conflict in the network. Since the TRETA-Multi-Sink can be distributed to multiple sinks for local processing according to the task area, its success rate in the task assignment is better than TRET A and TRET A-Cluster. The higher the task arrival rate, the more prominent the advantage.

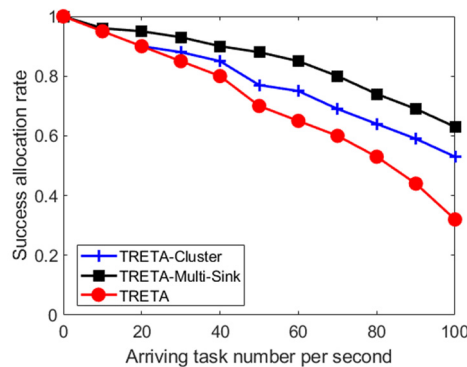


Figure 8. The success rate of task assignment under different task arrival rates.

(D) Energy Consumption and Delay of Task Assignments in the Dynamic Update of the Network

Figure 9 shows the energy consumption of the three proposed strategies for different network update ratios. It is noted that as the network update ratio increases, the energy consumption is significantly increased. Since the network update in TRET A is brought to the only sink through the long-distance multi-hop, the communication load of the network is increased. Moreover, the energy consumption of TRET A is higher than the other two strategies in most cases. With the network update ratio greater than 25%, the energy consumption of TRET A-Multi-Sink increases rapidly, surpassing TRET A and TRET A-Cluster, becoming the highest among the three strategies due to the exchange of global views between multiple sinks and the operation of ensuring global view consistency under a high network update rate.

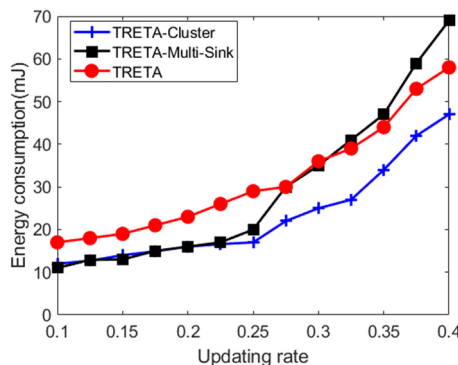


Figure 9. Energy consumption of task assignment under different network update rates.

Figure 10 shows the delays of the three proposed strategies for different network update ratios. As the network update ratio increases, the delays in the three strategies increase. Under the current network simulation environment, analysis shows that the delays of TRET A-Cluster and TRET A-Multi-Sink are smaller than TRET A when the network update ratio is lower than 25%, which is due to the fact that TRET A-Cluster and TRET A-Multi-Sink have a small number of updates in the network, they are processed locally without affecting the reliability of the task target, and a complete network update is not required. Nevertheless, when the network update ratio exceeds 25%, the delay of TRET A-Multi-Sink increases rapidly and becomes higher than TRET A, which is caused by the operation of ensuring global view consistency at such a high update rate.

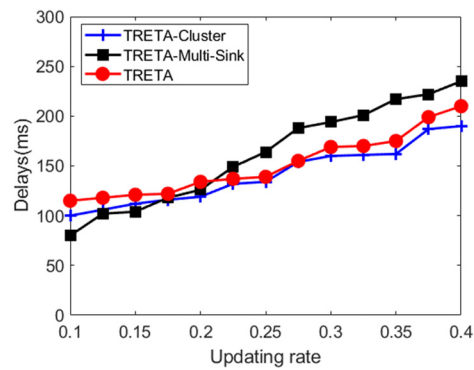


Figure 10. Delay in task assignment under different network update rates.

(E) Energy Resilience of the Transmission for Task Assignments

In order to evaluate the energy resilience of the transmission in WSNs, we quoted an index η in [34]. Since TRET-Cluster is a distributed modification of TRET for clustering topologies, we compared TRET-Cluster and LEACH (Low Energy Adaptive Clustering Hierarchy) with index η . Simulation experiments were carried out under clustering topology. Let the number of initial active nodes be 500. For the same transmission task, we sample and count the number of packets received by sink nodes and the number of dead nodes at different time points. Then, we use Equation (22) to calculate the index η . Figure 11 shows the variation of the index η of the two algorithms over time. It can be seen from Figure 11 that the index η of both methods increases with time. Compared with LEACH, the growth rate of index η of TRET-Cluster is more prominent, since TRET-Cluster adds more redundancy packets than LEACH to meet the target reliability.

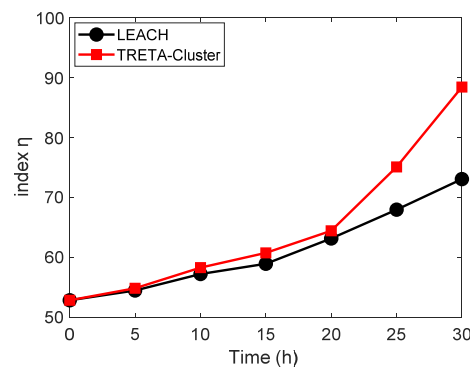


Figure 11. Energy resilience index of TRET-Cluster and LEACH.

7. Concluding Remarks and Future Work

To solve the problems of poor scalability and reliability, significant communication overheads and delays in centralized task assignment for WSNs, we propose two distributed reliable and efficient transmission task allocations for WSNs, namely TRET-Cluster and TRET-Multi-Sink. The performance of these strategies is analyzed and evaluated, then compared with the centralized task assignment TRET.

In TRET-Cluster, the cluster head can perform local task assignment according to the stage target reliability constraint that reduces the communication cost and delay compared with TRET. To summarize, the advantages include: (1) the cluster head can accurately select the number of nodes and the specific node ID in the cluster to complete the task, thereby saving energy; (2) as the topology changes occur in the cluster, the cluster head can select the replacement node according to the stage target reliability which has adaptive features. The advantages of TRET-Multi-Sink are two-fold: (1) the global view is generated by merging the local views of multiple sinks that reduce the delay

in obtaining the global view compared to the centralized acquisition method, and (2) multiple sinks respond to the local task assignment requirements in a globally optimized manner and do not need to communicate with remote nodes. It shows better scalability as well as overcoming the shortcomings of centralized task assignment in large-scale WSNs with significant communication overheads and delays.

From the analysis of both theoretical and simulation results, we show that the proposed distributed task allocation strategies are promising and superior to the centralized task allocation under the same network environment in terms of energy consumption, delay, and success rate.

From the observations of the limitations in distributed strategies when the network update ratio is higher than 25%, as TRET-Multi-Sink no longer has the lead (which is caused by ensuring the consistency of the global view), we target the re-design and improvements of TRET-Multi-Sink as a future direction to improve its efficiency. Another direction of investigation is to apply and adapt the proposed strategies in Cluster of Things (CoT) and Edge environments, where the communication conditions and environments highly yet dynamically vary all the time. Finally, this paper only researches the situation where the global view does not have concurrent updates. The efficiency of the distributed task allocation strategy when the network is concurrently updated will also be included as a future work direction.

Author Contributions: Methodology and investigation, X.Z.; Validation and data curation, J.Z. and S.Z.; investigation, Writing—original draft preparation, X.Z.; Writing—supervision, review, and editing, K.-C.L.

Funding: This work is jointly supported by the National Key Research and Development Program (Grant No. 2018YFC0604404), National Natural Science Foundation of China (Grant No. 51504010) and the Fundamental Research Funds for the Central Universities (Grant No. PA2019GDPK0079).

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

Symbol	Description
s_i	i -th source node
m_i	The task to be assigned, which is the element in the set $M(t)$
e_{ij}	Energy consumed by the task m_i on the node N_j
e_{ij}^{comp}	Calculated energy consumption of N_j
e_j	Average processing energy consumption of the nodes in the network
t_{ij}^{comp}	Calculation time of the task m_i
e_{ij}^{comm}	Communication energy consumption of the node N_j
e_{ij}^t	Energy consumption of transmitting data of length l_{ij} bit at the distance d
e_{elec}	Energy consumption of operating the radio model for each bit
ϵ_{amp}	Coefficient of the transmit amplifier
e_{ij}^r	Energy consumption of receiving data of length l_{ij} bit at the distance d
P_i	Task period of m_i
S_i	Start time of task m_i
C_i	Deadline for task m_i
R_{WSN}	Mission reliability of the network
R_s	Target reliability
$ClusterR_{s_i, sink}(t, i)$	i -th transmission Reliability of clustered WSNs from s_i to the sink at time t
$R_A^c(t, i)$	Reliability of the i -th transmission performed by cluster c .
$R_T^c(t, i)$	Reliability of the head of cluster c which successfully sends the collected data of i -th transmission to sink at time t .
$MeshR_{s_i, sink}(t, i)$	i -th transmission reliability in mesh WSNs from s_i to the sink at time t
$R_{s_i, sink}^i(t, i)$	Probability that there exists at least one path whose packet delivery rate is greater than the threshold of i -th transmission in the disjoint minimal path sets (from s_i to sink)
$R_{path(s_i, sink)}^j(t, i)$	i -th transmission reliability in mesh WSNs of j -th path from s_i to the sink at time t
$TG(V, E)$	Physical topology of the network

$RT(S, Ps)$	Routing topology of the network
$M(t)$	Task set to be assigned at time t
$\varphi(m_i)$	Mapped function of m_i
$Eng(\varphi(m_i))$	Energy consumption of the task m_i
$RT_b(S, Ps)$	Physical topology of the backbone network
$TG_b(S, Ps)$	Routing topology of the backbone network
$TG_c(S, Ps)$	The physical topology of the intra-cluster
$RT_c(S, Ps)$	Routing topology of the intra-cluster
$TG_{b'}(V, E)$	Physical topology after backbone network update
$RT_{b'}(S, Ps)$	Routing topology after backbone network update
$TG_{c'}(V, E)$	Physical topology of the intra-cluster after update
$RT_{c'}(S, Ps)$	Routing topology of the intra-cluster after update
$TG_j(V, E)$	Local physical topology of the j -th sink in a multi-sink topology
$RT_j(S, Ps)$	Local routing topology of the j -th sink in a multi-sink topology
$sink_j$	The closest sink to the task m_i
$sink_{jn}$	Sink adjacent to $sink_j$
$sink_k$	Sink involving topology changes
$viewID$	The version ID of the global view
$PathSet_{m_i}$	Path set of the source node to the sink in the coverage area of the task m_i
$PathSet_{m_i}'$	Path set after deleting the paths with link conflict in $PathSet_{m_i}$
$Paths_i$	Transmission path of each source node s_i to sink in $PathSet_{m_i}'$
R_{s_i}	Reliability of $Paths_i$
E_{s_i}	Energy consumption of $Paths_i$
$path_{R_s}$	Path to meet the target reliability R_s
R_{CH}	Target reliability of cluster head to sink after reliability allocation
$R_{CH'}$	Updated R_{CH}
R_{CM}	Target reliability of cluster head after reliability allocation
$sink_k^{res}$	Sink respond to update request of $sink_k$
$sink_k^{res}.RT_j(S, Ps)$	Routing topology of the j -th sink in $sink_k^{res}$
$sink_k^{res}.TG_j(V, E)$	Physical topology of the j -th sink in $sink_k^{res}$
$sink_k.record$	Update operation record of $sink_k$
$sink_k^{res}.log$	Log of sink which respond to update
S_W, S_R	Quorum set of writing and reading
$P_{overlap}$	Overlapping probability of S_W and S_R
CM	Communication load
Num_{cp}	Number of packets in the allocation policy by multicast,
Num_{gv}	Communication load for transmitting one global view
WM, RW	Writing load, reading load
VUM	Write load of version and update operation of the global view when it is updated
RT_{d_c}	Delay of task assignment for the clustered topology
RT'_c	Delay of the sink allocates a task to the subgraph composed of the cluster heads
MC	Delay of the multicast performed by the sink
CHD	Delay in the allocation of the task by the selected cluster head
RT_{d_d}	Delay of TRET-Multi-Sink
\overline{RT}'_c	Average of the delays of the tasks assigned by multiple sinks locally
C_{rt}	Time to ensure the final consistency of the view
W_{rt}	Time of writing the global view whenever it is updated
R_{rt}	Time of reading the global view when the distributed task is assigned
η	Index for energy resilience of the transmission
N_p	Number of packet received by the sink after fixed time
S	Number of initial active nodes
D	Number of nodes death after a fixed time

References

1. Yu, W.; Huang, Y.; Garcia-Ortiz, A. Distributed optimal on-line task allocation algorithm for wireless sensor networks. *IEEE Sens. J.* **2018**, *18*, 446–458. [[CrossRef](#)]
2. Khalil, E.A.; Ozdemir, S.; Tosun, S. Evolutionary task allocation in internet of things-based application domains. *Future Gener. Comput. Syst.* **2018**, *86*, 121–133. [[CrossRef](#)]
3. Li, W.; Delicato, F.C.; Pires, P.F.; Lee, Y.C.; Zomaya, A.Y.; Miceli, C.; Pirmez, L. Efficient allocation of resources in multiple heterogeneous wireless sensor networks. *J. Parallel Distrib. Comput.* **2014**, *74*, 1775–1788. [[CrossRef](#)]
4. Ghebleh, R.; Ghaffari, A. A multi-criteria method for resource discovery in distributed systems using deductive fuzzy system. *Int. J. Fuzzy Syst.* **2017**, *19*, 1829–1839. [[CrossRef](#)]
5. Alskaf, T.; Zapata, M.G.; Bellalta, B. Game theory for energy efficiency in wireless sensor networks: Latest trends. *J. Netw. Comput. Appl.* **2015**, *54*, 33–61. [[CrossRef](#)]
6. Piloni, V.; Navaratnam, P.; Vural, S.; Atzori, L.; Tafazolli, R. Tan: A distributed algorithm for dynamic task assignment in WSNs. *IEEE Sens. J.* **2014**, *14*, 1266–1279. [[CrossRef](#)]
7. Tang, C.; Li, X.; Wang, Z.; Han, J. Cooperation and distributed optimization for the unreliable wireless game with indirect reciprocity. *Sci. China Inf. Sci.* **2018**, *60*, 110205. [[CrossRef](#)]
8. Duan, X.; Zhao, C.; He, S.; Cheng, P.; Zhang, J. Distributed algorithms to compute Walrasian equilibrium in mobile crowdsensing. *IEEE Trans. Ind. Electron.* **2017**, *64*, 4048–4057. [[CrossRef](#)]
9. Piloni, V.; Atzori, L.; Mallus, M. Dynamic involvement of real-world objects in the IoT: A consensus-based cooperation approach. *Sensors* **2017**, *17*, 484. [[CrossRef](#)]
10. Ye, D.Y.; Zhang, M.J.; Vasilakos, A.V. A survey of self-organization mechanisms in multiagent systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 441–461. [[CrossRef](#)]
11. Khan, M.I.; Xia, K.; Ali, A.; Aslam, N. Energy-aware task scheduling by a true online reinforcement learning in wireless sensor networks. *Int. J. Sens. Netw.* **2017**, *25*, 244–258. [[CrossRef](#)]
12. Nguyen, C.L.; Hoang, D.T.; Wang, P. Data collection and wireless communication in internet of things (IoT) using economic analysis and pricing models: A survey. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2546–2590.
13. de la Hoz, E.; Gimenez-Guzman, J.; Marsa-Maestre, I.; Orden, D. Automated negotiation for resource assignment in wireless surveillance sensor networks. *Sensors* **2015**, *15*, 29547–29568. [[CrossRef](#)] [[PubMed](#)]
14. Wang, L.; Wang, Z.; Hu, S.; Liu, L. Ant colony optimization for task allocation in multi-agent systems. *China Commun.* **2013**, *10*, 125–132. [[CrossRef](#)]
15. Liu, Q.; Yin, X.; Yang, X.; Ma, Y. Reliability evaluation for wireless sensor networks with chain structures using the universal generating function. *Qual. Reliab. Eng. Int.* **2017**, *33*, 2685–2698. [[CrossRef](#)]
16. Cai, J.; Song, X.; Wang, J.; Gu, M. Reliability analysis for a data flow in event-driven wireless sensor networks. *Wirel. Pers. Commun.* **2014**, *78*, 151–169. [[CrossRef](#)]
17. He, Y.; Gu, C.; Han, X.; Cui, J.; Chen, Z. Mission reliability modeling for multi-station manufacturing system based on Quality State Task Network. *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.* **2017**, *231*, 701–715. [[CrossRef](#)]
18. Zhu, X.J.; Lu, Y.; Zhang, J.; Wei, Z. Routing topology inference for wireless sensor networks based on packet tracing and local probing. *IEICE Trans. Commun.* **2019**, *102*, 122–136. [[CrossRef](#)]
19. Xin, Y.W.; Xiao, Y.W.; Narayanaswamy, B. Reliability allocation model and algorithm for phased mission systems with uncertain component parameters based on importance measure. *Reliab. Eng. Syst. Saf.* **2018**, *180*, 266–276.
20. Garg, H.; Sharma, S.P. Multi-objective reliability-redundancy allocation problem using particle swarm optimization. *Comput. Ind. Eng.* **2013**, *64*, 247–255. [[CrossRef](#)]
21. Li, R.; Wang, J.; Liao, H.; Huang, N. A new method for reliability allocation of avionics connected via an airborne network. *J. Netw. Comput. Appl.* **2015**, *48*, 14–21. [[CrossRef](#)]
22. Deng, R.; He, S.; Chen, J. An online algorithm for data collection by multiple sinks in wireless sensor networks. *IEEE Trans. Control Netw. Syst.* **2018**, *5*, 93–104. [[CrossRef](#)]
23. Basagni, S.; Carosi, A.; Petrioli, C.; Phillips, C.A. Coordinated and controlled mobility of multiple sinks for maximizing the lifetime of wireless sensor networks. *Wirel. Netw.* **2011**, *17*, 759–778. [[CrossRef](#)]
24. Lee, E.; Park, S.; Lee, J.; Oh, S.; Kim, S.H. Novel service protocol for supporting remote and mobile users in wireless sensor networks with multiple static sinks. *Wirel. Netw.* **2011**, *17*, 861–875. [[CrossRef](#)]

25. Lanny, S.; Kenneth, N.B.; Cormac, J.S. Planning the deployment of multiple sinks and relays in wireless sensor networks. *J. Heuristics* **2015**, *21*, 197–232.
26. Banerjee, A.; Hussain, D.M.A. SD-EAR: Energy aware routing in software defined wireless sensor networks. *Appl. Sci.* **2018**, *8*, 1013. [[CrossRef](#)]
27. Gilbert, S.; Lynch, N. Brewer’s conjecture and the feasibility of consistent, available, partition-tolerant web services. *Acm Sigact News* **2002**, *33*, 51–59. [[CrossRef](#)]
28. Eventually Consistent-Revisited. Available online: http://www.allthingsdistributed.com/2008/12/eventually_consistent.html (accessed on 4 September 2019).
29. DeCandia, G.; Hastorun, D.; Jampani, M.; Kakulapati, G.; Lakshman, A.; Pilchin, A.; Sivasubramanian, S.; Vosshall, P.; Vogels, W. Dynamo: amazon’s highly available key-value store. In *ACM SIGOPS Operating Systems Review*; ACM: New York, NY, USA, 2007; Volume 41, pp. 205–220.
30. Vashisht, P.; Sharma, A.; Kumar, R. Strategies for replica consistency in data grid—A comprehensive survey. *Concurr. Comput. Pract. Exp.* **2018**, *29*, e3907. [[CrossRef](#)]
31. Wang, X.; Sun, H.; Deng, T. On the tradeoff of availability and consistency for quorum systems in data center networks. *Comput. Netw.* **2015**, *76*, 191–206. [[CrossRef](#)]
32. Raychoudhury, V.; Cao, J.; Wu, W. K-directory community: Reliable service discovery in MANET. *Pervasive Mob. Comput.* **2011**, *7*, 140–158. [[CrossRef](#)]
33. Guerraoui, R.; Vukolic, M. Refined quorum systems. *Distrib. Comput.* **2010**, *23*, 1–42. [[CrossRef](#)]
34. Leccese, F.; Cagnetti, M.; Giarnetti, S.; Petritoli, E.; Luisetto, I.; Tuti, S.; Leccisi, M.; Pecora, A.; Maiolo, L.; Đurović-Pejčev, R.; et al. Comparison between routing protocols for wide archeological site. In *Proceedings of the IEEE International Conference on Metrology for Archaeology and Cultural Heritage*, Cassino, Italy, 22–24 October 2018; ISBN 978-1-5386-52.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).