

Article

CARA: A Congestion-Aware Routing Algorithm for Wireless Sensor Networks

Jiangyu Yan *  and Bing Qi

School of Electrical and Electronic Engineering, North China Electric Power University, Beijing 102206, China; qbing@ncepu.edu.cn

* Correspondence: yabjy@ncepu.edu.cn

Abstract: Congestion control is one of the key research topics in relation to the routing algorithms of wireless sensor networks (WSNs). In this paper, we propose a congestion-aware routing algorithm (CARA) for unlimited-lifetime wireless sensor networks by integrating the geographic distance and traffic load of sensor nodes. The algorithm takes alleviating congestion as the primary purpose and considers the traffic of the node itself and local network traffic. According to the geographic distance between nodes, CARA defines four decision parameters (node load factor, forward rate, cache remaining rate, and forward average cache remaining rate), selecting the best node as the next-hop through the multi-attribute decision-making method. Compared with the two existing algorithms for congestion control, our simulation results suggest that the CARA algorithm alleviates network congestion and meets reasonable network delay and energy consumption requirements.

Keywords: congestion; wireless sensor networks; traffic load; routing



Citation: Yan, J.; Qi, B. CARA: A Congestion-Aware Routing Algorithm for Wireless Sensor Networks. *Algorithms* **2021**, *14*, 199. <https://doi.org/10.3390/a14070199>

Academic Editors: Marc Sevaux and Frank Werner

Received: 18 May 2021
Accepted: 29 June 2021
Published: 30 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Wireless sensor networks have attracted widespread attention, as technologies such as the Internet of Things, smart grids, and vehicular networks demand underlying data collection [1–4]. Wireless sensor networks (WSNs) consist of tiny nodes with autonomous awareness, processing, and communication capabilities. These tiny sensor nodes connect to each other in an ad hoc manner to sense and monitor their surroundings and devices. As unique wireless self-organizing networks, WSNs have fast and convenient deployment features and complete monitoring capabilities. WSNs are thus usually deployed in hazardous and inaccessible terrain and are used for defense, environmental monitoring, and medical care applications [5–7].

As is the case in traditional wireless self-organizing networks, there is no uniform paradigm between wireless sensor network nodes. Nodes exchange data with the neighbor node through broadcasting and multi-hop communication. Each node uses a short-range wireless signal to route packets through the intermediate node to the sink node [8]. Therefore, routing algorithms have an impact on the performance of WSNs. Sensor nodes have some inherent characteristics and limitations such as Communication range and memory capabilities, which pose a significant challenge to the design of routing algorithms [9]. Some routing algorithms focus on WSNs' node energy limitation characteristics to prolong the network lifetime as the primary goal, such as typical low-energy adaptive clustering hierarchy (LEACH) [10], multi-criteria-based centrality routing protocol (MCRP) [11], a DS evidence theory-based energy balanced routing algorithm for network lifetime enhancement (EBRA-NLE) [12], etc.

However, the data traffic of WSNs involves significant upstream traffic (from the source node to the sink). The data traffic is closely related to the WSNs application and has the characteristics of periodicity or continuity [13]. A large amount of upstream data traffic can overload nodes and exceed their processing power, resulting in increased latency, packet loss, and retransmission. The deterioration of underlying network performance

adversely affects the reliability of monitored applications. In addition to these factors, resources are constrained in WSNs, such as processing power, available memory, and battery power. Resource constraints, node overload, error-prone communication links, and irregular upstream traffic can ultimately lead to network congestion. Congestion significantly reduces network performance and results in increased packet loss.

Moreover, in some industrial applications of wireless sensor networks, such as power transmission line monitoring systems [14], sensor nodes can be powered by power lines or radio frequency waves [2,15], so the above-mentioned energy-efficient routing algorithms cannot achieve the desired results [15]. Meanwhile, when WSNs are used in information acquisition and monitoring systems, the proliferation of sensor-generated data may lead to network congestion due to unexpected events in the perception area, seriously reducing network performance, which affects the quality of service [16]. Therefore, to ensure the quality of information acquisition, it is necessary to design a specialized WSN routing algorithm to reduce network congestion [17].

Based on the above observations, in order to alleviate network congestion and reduce data transmission delay, we propose a congestion-aware routing algorithm (CARA). The CARA algorithm senses the congestion status of nodes and their neighboring areas, and uses the multi-attribute method to make routing decisions. Our main contributions are summarized as follows:

- CARA proposes a node queue model to detect the level of load on the node.
- CARA considers the geographical location relationship between the nodes and sink, the cache state of the node itself, and the network's local congestion state; establishes routing evaluation parameters; and optimizes routing decisions to alleviate congestion.
- CARA optimizes packet transmission paths, bypasses congestion nodes, reduces packet drop rates, and improves network reliability based on multi-attribute decision-making principles.

The rest of the paper is organized as follows. In Section 2, an overview of related research works in the literature on congestion routing is presented. The CARA algorithm is described in detail in Section 3. The relevant simulation results and their analyses are discussed in Section 4. Section 5 concludes the paper.

2. Related Works

Congestion is one of the most common problems in wireless sensor networks (WSNs) due to the computing, storage, transmission, and energy supply constraints of wireless sensor network nodes. In the past, many scholars have researched WSN congestion and have already presented a considerable number of research results. In one study, the media access control (MAC) layer access mechanism of the WSN is analyzed, and a survivability aware channel allocation (SACA) algorithm is developed based on a fuzzy method to prioritize access strategies [18]. The algorithm reduces packet collisions during congestion, improving network throughput and transmission reliability. Furthermore, [19] analyzes network congestion caused by cascading failure among the nodes of WSNs, designs a route-based WSN cascading model, and enhances the congestion resistance of the network. However, these two congestion control algorithms relate to the physical layer of the WSNs and lack adaptability to most WSNs in which the IEEE 802.15.4 standard specification is applied to the physical layer.

The WSN layer feature provides a path from the source node to the destination node, along which packets propagate. It is easier to match the underlying hardware from the network layer to identify and solve congestion problems. As a result, many studies provide solutions to congestion from routing protocols. For example, the decentralized predictive congestion control (DPCC) [20] protocol uses the node's hop-by-hop feedback information and embedded channel estimation algorithm to evaluate channel quality based on queue utilization. As a result, the data flow can adaptively select the route, reducing network congestion to a certain extent. Based on the idea of actively avoiding congestion, a multi-path optimizing routing protocol based on predicting congestion (MOPC) protocol [21]

developed a path satisfaction model. It realizes the optimal route selection based on congestion prediction, remaining energy, and the lowest number of hops. As a result, MOPC improves reliability and real-time performance. According to the overall energy consumption and network lifetime requirements, a scheme of priority of energy congestion relief (PECR) [22] builds an energy priority congestion detection and mitigation mechanism. It takes the residual energy as an essential parameter, periodically detects the cache queue occupancy rate, and attempts to establish a temporary optimal path around the congestion node for congestion control. Thus, PECR effectively extends the network lifetime and alleviates network congestion. The congestion-aware routing (CoAR) protocol [23] is specific to the application of WSNs in low power and lossy networks. It detects congestion based on current queue capacity and observation of historical traffic and selects alternative parent nodes to alleviate congestion, effectively alleviating network congestion in LLN (Low-power and Lossy Networks). However, the above algorithms are designed for wireless sensor networks with limited energy, emphasizing the development of node lifetime through an energy-saving method.

In contrast, the wireless sensor networks studied in this paper have a continuous energy supply and generally do not need to consider energy consumption during routing selection. The optimizing routing algorithm based on congestion control (CCOR) [24] algorithm constructs the network queuing model to detect the congestion degree and selects route probability based on the water flow principle. The traffic-aware and energy-efficient routing (TER) [25] has built a pipe model that considers both spatial relationships and network loads, flowing packets to the convergence node as quickly as possible with minimal energy consumption and avoidance of congestion. The gravitation theory-based routing (GTR) algorithm [26] defines the concepts of “space gravity” and “gravitational pull” so that packets can avoid congestion areas as much as possible in transit and prevent unavoidable congestion. All three algorithms can reduce network congestion to some extent; however, they only consider the load states of nodes or the entire network unilaterally while ignoring the impact of local congestion on the overall network performance and global load.

We propose a congestion-aware routing algorithm (CARA) for active wireless sensor networks based on the above analysis. Considering the node’s load and the network local load, we define the “cache remaining rate” and “forward average cache residual rate” to minimize the network packet loss rate. At the same time, to deliver the data collected by the node to the sink node as soon as possible, the algorithm also puts forward the “forward rate”. Finally, the optimal next-hop node is determined utilizing multiparameter fusion. Simulation results show that the algorithm can significantly alleviate network congestion and reduce network packet loss and network latency under the condition that the sensor node has a continuous energy supply.

3. Congestion-Aware Routing Algorithm

Wireless sensor networks have the advantages of having a large number of nodes, low power consumption and cost requirements, and self-organization and scalability features. Therefore, they can be used at the end of local networks as an effective complement to other networking methods [27]. In an information acquisition system, due to the continuous power supply, the WSN’s sensor node will not die of energy failure. However, it has specific requirements for packet loss and delay, both of which will affect business performance. Therefore, the design of efficient routing algorithms to meet the relevant provisions is critical for WSN applications. In this paper we take network congestion control as the primary purpose, consider reducing network energy consumption and delay, and design an active congestion control routing algorithm.

To simplify the WSN model, we make the following reasonable assumptions:

- Many sensor nodes are randomly placed in a certain area, and all nodes are immovable after deployment;
- All sensor nodes are homogeneous, have routing and data acquisition function;

- A sensor node can estimate the distance to the source node based on the strength of the received signal;
- The cache queue length is null. Therefore, packets enter the cache queue using the mean of the first-in-first-out (FIFO).

3.1. Route Evaluation Parameters

The CARA algorithm proposed in this article is a distributed routing protocol. Each node calculates routing parameters based on existing information when forwarding a data packet and then selects the best neighbor node to receive the data packet according to the steps in Section 3.2. The CARA algorithm defines four routing evaluation parameters that evaluate nodes from the three scales of distance, processing capacity, and congestion. The “forward rate” expresses the distance relationship between nodes and between the nodes and sink. The “load factor” reflects the node’s data packet processing capability. “The cache remaining rate” and “the forward average cache remaining rate” describe the node’s perception of the degree of congestion in itself and its surroundings. These four parameters are described in detail below.

3.1.1. The Forward Rate

Because sensor nodes have a limited range of transmission, most nodes often cannot communicate directly with sink nodes, requiring additional nodes to forward data. Therefore, the routing protocol needs to consider the spatial distance relationships between the forward node, the neighbor node, and the sink node during routing decisions.

If a packet sent to the sink is in node i , node i must send the packet to a forward node to avoid forwarding it back. According to the definition of [28], the forward neighbor set $F(i)$ of node i can be expressed as

$$F(i) = \{j \mid d_{j,sink} \leq d_{i,sink}, d_{i,j} \leq R\} \quad (1)$$

In the definition, the distance from node i to the sink is $d_{i,sink}$; the distance from node j to the sink node is $d_{j,sink}$; the distance from node i to node j is $d_{i,j}$. R is the node’s maximum transmission distance.

To select the next-hop node from the forward neighbor set, we define the forward rate. As shown in Figure 1, j' represents the intersection of the line $isink$ and the arc, the center of which is the sink node, and the radius is $d_{j,sink}$. Therefore, we define the forward distance $d_f(i, j)$ from node i to node j as

$$d_f(i, j) = d_{i,j} = d_{i,sink} - d_{j,sink}, j \in F(i) \quad (2)$$

The forward distance $d_f(i, sink)$ from node i to the sink is

$$d_f(i, sink) = d_{i,sink} \quad (3)$$

Based on the forward distance $d_f(i, j)$ and $d_f(i, sink)$, the forward rate r_f of node i to node j is

$$r_f(i, j) = \frac{d_f(i, j)}{d_f(i, sink)} = \frac{d_{i,sink} - d_{j,sink}}{d_{i,sink}} \quad (4)$$

The forward rate expresses the spatial distance relationship between node i , node j , and the sink. The higher the forward rate of the node, the closer the packet is to the sink node after forwarding to the next-hop node. In the free space model, the node prefers to select the next-hop node with a long single-hop transmission distance to reduce the number of hops on the path and the transmission delay.

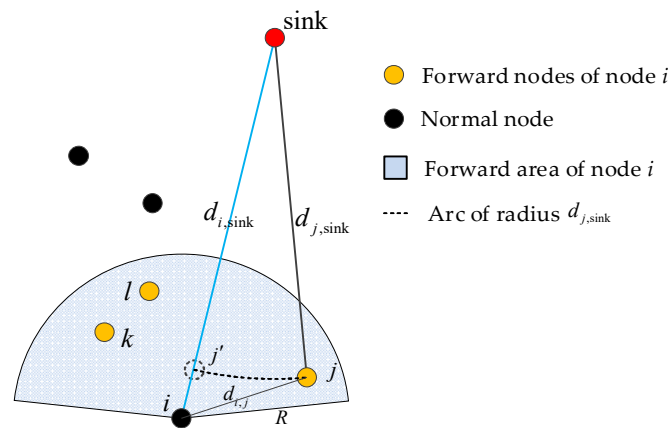


Figure 1. The spatial relationships between the nodes.

The forward rate reflects the spatial relationship between the nodes and causes packets to flow directly to the sink, reducing network latency and energy consumption. Therefore, if the routing selection only considers the forward rate, the node should transfer packets to the forward node with the highest forward rate.

3.1.2. The Node Load Factor

Because the wireless sensor network characteristics include many-to-one transmission, multi-hop forwarding, and centralized data acquisition, they can easily cause network congestion. Congestion can lead to many packets being dropped and increase data transmission delays, seriously limiting the transmission performance of the network. In this study, we accurately judge the node's congestion by establishing a network queuing model to solve network congestion.

Wireless sensor network nodes perform the functions of data acquisition and data forwarding. Because sensor nodes are usually inexpensive, they mostly use microcontrollers as processing units, and their limited storage capacity makes cache queue process packets using the FIFO order. Packets entering the queue include packets generated by the node's acquisition data and packets forwarded by the neighbor node. The cache queue model for node j is shown in Figure 2.

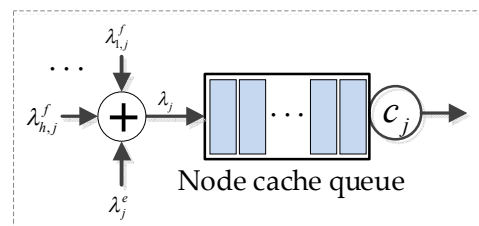


Figure 2. Node j queue model.

The packet generating rate of node j is λ_j^e . The packet receiving rate from node j 's neighbor node i is $\lambda_{i,j}^f$. The packet arrival rate of node j can then be expressed as:

$$\lambda_j = \lambda_j^e + \sum_{\substack{i=1 \\ i \neq j}} \lambda_{i,j}^f \quad (5)$$

When monitoring class sensor network applications, nodes process packets only to select the next hop for them, according to predetermined routing rules. When a node's packet arrival rate is close to its forwarding capacity, the node cache queue will be nearly

full. If other nodes simultaneously forward data to it, packets can be lost because they are unacceptable. In this case, the node also becomes a congestion node. Therefore, we introduce the node load factor to evaluate the node's ability to route and forward packets in the queue. At time t , node j 's load factor L_j^t is:

$$L_j^t = \frac{c_j^t}{\lambda_j^t + q_j^t} \quad (6)$$

Here, c_j^t is the packet service rate of node j at time t , i.e., the number of packets that node j can handle in a unit of time. The packet arrival rate for node j is λ_j^t . q_j^t is the node j 's current cache queue size. The node load factor dynamically reflects the processing of the node and its ability to receive packets. For the node with a significant load factor, its data forwarding ability is strong. The packet stay time in the node cache queue is short, and the node can assume more forwarding tasks and accept more packets. The routing protocol should avoid sending more packets to nodes with a small load factor to prevent network congestion and packet loss.

3.1.3. The Cache Remaining Rate

The data transmission process in event-driven WSNs has prominent "many-to-one" characteristics which could cause network congestion, increase the delay, and reduce the transmission success rate during traffic bursts. Because the cache queue occupancy can reflect node load status, the routing protocol should consider this property to avoid the higher cache occupied node. Moreover, it should transfer the traffic load to the idle or lower cache occupied node to realize the fair utilization of wireless network resources. Thus, the remaining cache $r_c(j)$ of the candidate node j is expressed as follows:

$$r_c(j) = \frac{1 - Q_j}{Q_{max}}, j \in F(i) \quad (7)$$

Here, Q_j is the latest cache queue length of node j , and Q_{max} is the cache's max length.

When selecting the next-hop from the forward region of node i , the preferred node has a more significant cache remaining amount than the mean length. For this reason, based on the *logistic* function, we introduce the $f_\delta(x)$ function, which can be expressed as

$$f_\delta(x) = \frac{1}{1 + \left(\frac{1-x}{1-\delta}\right)^2}, x \in [0, 1] \quad (8)$$

Assume x represents the cache remaining amount of node j , δ represents the average cache remaining number of nodes from the forward neighbor set, and $num[F(i)]$ is the number of node j 's forwarding neighbor.

$$\delta = \overline{r_c(j)} = \frac{1}{num[F(i)]} \bullet \sum_{j \in F(i)} r_c(j) \quad (9)$$

As shown in Equation (8), the value of the function $f_\delta(x)$ increases with the increase in x , and the inflection point is the mean of δ . Moreover, the rate of change for function $f_\delta(x)$ near the mean δ will be significantly greater than the rate of change at both endpoints of x . This will make the nodes with $f_\delta(x)$ values greater than the mean more dominant in routing decisions. Combined with the cache remaining amount and function $f_\delta(x)$, the cache rate of the node j is:

$$r_{fc}(j) = f_{r_c(j)}[r_c(j)] \quad (10)$$

The cache rate can effectively represent the node load. Therefore, in the process of packet transmission, the routing protocol tries to select the node with a larger cache

remaining rate to minimize the loss of packets in the congested node to ensure the success rate of data collection.

3.1.4. The Forward Average Cache Remaining Rate

The cache remaining rate of the node itself can directly reflect its level of congestion. However, it is not enough to consider the node itself. When the node located in the congested region receives packets, the next-hop choice becomes particularly difficult. If the optional next hop has a lower cache remaining rate, it can easily cause congestion resulting in many packets being lost. To solve these problems, we use the forward average cache remaining rate to analyze the congestion of a node's forward area.

Suppose node j is the next hop candidate node of node i , according to equation (10) the cache remaining rate of the node j is $r_{fc}(j)$. When $num[F(j)]$ represents the number of forwarding nodes of node j , the forward average cache remaining rate $r_{fave}(j)$ of node j is

$$r_{fave}(j) = \frac{1}{num[F(j)]} \bullet \sum_{t \in F(j)} r_{fc}(t) \quad (11)$$

The forward average cache remaining rate of a node reflects the load of its forward transmission area. Thus, during packet transmission, the routing protocol should select as far as possible the node with a large forward average cache remaining rate, which is conducive to the further forwarding of the packet.

3.2. Multiparameter Routing Decision

Sensor node i selects a node from the forward neighbor set as the next-hop node based on a given routing policy. Node i collects routing evaluation parameters for each node of the forward neighbors and establishes the initial judgment matrix D . For the j th candidate node, its k th evaluation parameter is a_{jk} ($1 \leq j \leq m, 1 \leq k \leq n$). Therefore, we set the initial judgment matrix $D = (a_{jk})_{m \times n}$ accordingly.

Since the units of each evaluation indicator are different, the initial judgment matrix D needs to be normalized to eliminate its effects. For benefit parameters, we have

$$r_{jk} = \frac{a_{jk} - a_{min}}{a_{max} - a_{min}} \quad (12)$$

So, we get the normalized judgment matrix $D_N = (r_{jk})_{m \times n}$.
For cost parameters, we have

$$r_{jk} = \frac{a_{max} - a_{jk}}{a_{max} - a_{min}} \quad (13)$$

Different weights represent the importance of varying evaluation parameters to reconcile the routing process's unevenness. Here, we define the weight vector as $W = [w_1, w_2, \dots, w_n]^T$, where

$$w_k = \frac{\sigma_k}{\sum_{k=1}^n \sigma_k}, k = 1, 2, \dots, n \quad (14)$$

In the above equation, σ_k is the standard deviation of the k -th evaluation parameter and w_k is the corresponding weight for each parameter in the formula. This results in a weighted standardized judgment matrix:

$$D_T = D_N \bullet W = (v_{jk})_{m \times n} = (r_{jk} \cdot w_k)_{m \times n} (j = 1, 2, \dots, m, k = 1, 2, \dots, n) \quad (15)$$

The definition of the positive ideal solution in the matrix D_T is $A^+ = [v_1^+, v_2^+, v_3^+]$, wherein $v_k^+ = \max\{v_{jk}\}, j = 1, 2, \dots, m$. V_k^+ indicates that each parameter value has reached the best value in each candidate node. Conversely, the definition of the negative ideal solu-

tion is $A^- = [v_1^-, v_2^-, v_3^-]$, and $v_k^- = \min\{v_{jk}\}, j = 1, 2, \dots, m$. A^- represents a solution in which each parameter value reaches the worst value of each candidate node.

We calculate the Euclidean distance from the candidate node j to the positive A^+ and negative A^- ideal values, as follows:

$$\begin{aligned} S_j^+ &= \sqrt{\sum_{k=1}^n (v_{jk} - v_k^+)^2} \\ S_j^- &= \sqrt{\sum_{k=1}^n (v_{jk} - v_k^-)^2} \end{aligned} \quad (16)$$

Here, S_j^+ indicates the proximity of node j and the positive ideal solution. Similarly, S_j^- represents the proximity of node j and the negative ideal solution. Thus, the routing strategy that chooses the node close to the positive ideal solution and away from the negative ideal solution can improve performance. Obviously, $\min_{1 \leq j \leq m} \{S_j^+\}$ and $\max_{1 \leq j \leq m} \{S_j^-\}$ is the optimal solution.

Thus, in this paper we establish a two-dimensional plane (S^+, S^-) in which each candidate node has its corresponding coordinates. For example, the coordinates of node j are (S_j^+, S_j^-) . Assuming A is an ideal optimal node, its coordinates are $(\min_{1 \leq j \leq m} \{S_j^+\}, \max_{1 \leq j \leq m} \{S_j^-\})$. Each candidate node calculates the Euclidean distance from itself to node A :

$$R_j = \sqrt{[S_j^+ - \min_{1 \leq j \leq m} \{S_j^+\}]^2 + [S_j^- - \max_{1 \leq j \leq m} \{S_j^-\}]^2} \quad (17)$$

R_j represents how close the node is to the ideal optimal node. The routing strategy selects the candidate node with the smallest R_j value as the next hop.

3.3. Route Parameter Updates

The object of this paper is an active wireless sensor network, so nodes need not consider energy consumption. After the node finishes forwarding the data, it will update its network parameters and broadcast its routing parameters to the neighbor node through the HELLO message. Each node maintains a neighbor node information table that records parameters such as the load state of the neighbor node, cache queue status, and node distance. When each neighbor node receives the HELLO message, it will update its node information table to monitor the congestion status of the adjacent area in real-time. Finally, the node calculates the next-hop route for each packet, effectively alleviating the network congestion situation. The HELLO message is a kind of control packet. The size of the control packet is small, so this paper does not consider the traffic generated by the control packet.

3.4. The Complexity Analysis

The CARA algorithm proposed in this article is a distributed one-hop routing algorithm. After a node is deployed, its location remains fixed. Each node calculates its routing evaluation parameters. The evaluation parameter reflects a certain aspect of the node's state, regardless of the number of nodes, so its time complexity is constant. When making routing decisions, the decision node establishes judgment matrix D . The decision matrix D is related to the number N of forwarding nodes. In the routing decision-making process, the time complexity of the normalization is $O(N)$. The weight vector computes in constant time. Both of the ideal values and R_j execute in $O(N)$ time. Thus, the time complexity of the CARA algorithm is $O(N)$. For spatial complexity, the neighbor information table and decision matrix maintained by each node are related to N , which is also $O(N)$. To sum up, the CARA algorithm occupies less memory space, offers faster decision-making speed, and has lower computing and storage requirements.

4. Performance Evaluation

To verify the validity of the proposed CARA algorithm, MATLAB was used to simulate the experiment. The comparison algorithms included CCOR and TER algorithms. The final results are presented as the average of multiple experiments. During the simulation process, the simulation area is 100×100 square meters. Then, a certain number of wireless sensor nodes are randomly deployed in the area. Finally, a node transfers self-collected data to the sink located in the center of the simulation region. To simplify the simulation, we assumed that all nodes produced data at the same rate. The specific simulation parameters are shown in Table 1. During the simulation, we modified some parameter values according to our different needs.

Table 1. The simulation parameter settings.

Parameters	Value
Number of nodes	100~300
Maximum transmission range	30 m
Data packet size	1024 bits
Cache queue length	20~50 packets
Sink coordinate	(50,50)
Data rate	4096 bit/round
Simulation time	400 round

In the performance analysis, we used the following three route equation metrics:

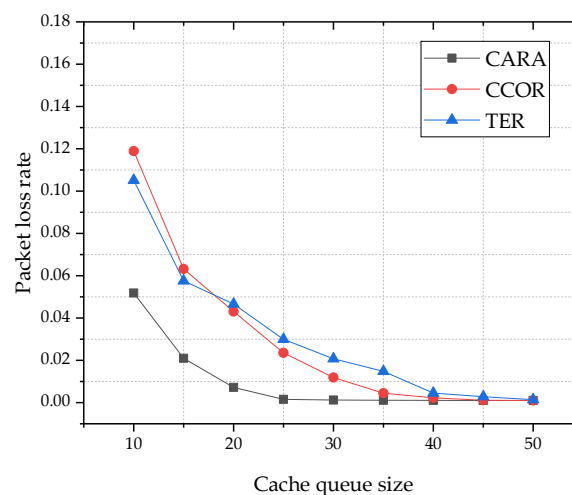
(1) **The Packet Loss Rate.** This metric represents the ratio of the number of lost packets to the total number of packets sent by the source node. In the WSN, the sink node is the final destination of all packets. Therefore, this metric indicates the data delivery reliability.

(2) **The Average Hops.** This metric represents the average number of nodes that the source node experiences transmitting packets to the sink, which can to some extent indicate the real-time performance of the network.

(3) **The Average Energy Consumption.** This metric represents the average energy consumption per round in the whole network. In our simulation, a round means sensor nodes send data packet to the next hop. This metric reflects the energy efficiency of the data transfer.

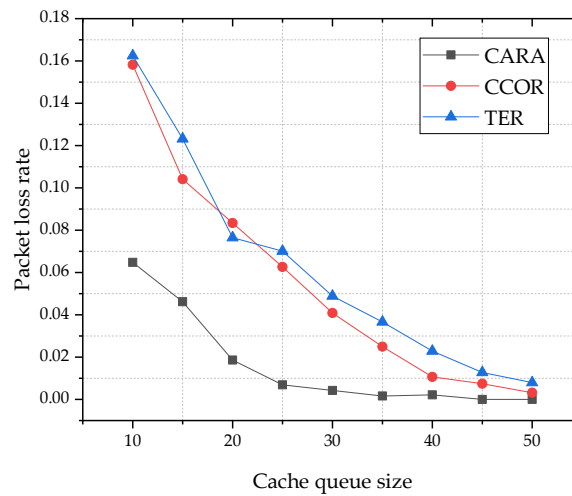
4.1. The Packet Loss Rate

To verify the performance of the proposed CARA algorithm in alleviating network congestion, we simulated the network packet loss rate, which varies with the length of the node cache, as shown in Figure 3.

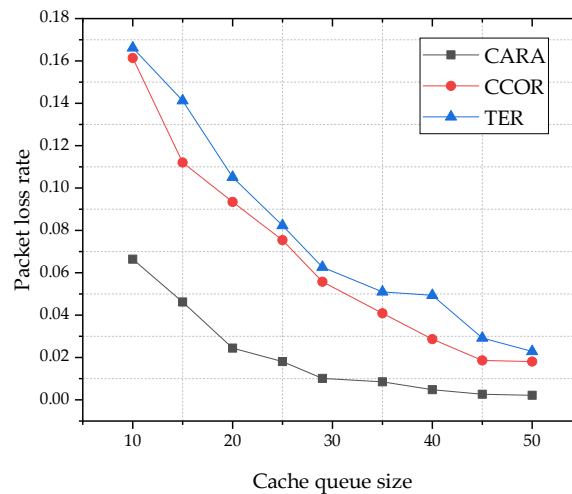


(a) The number of nodes = 100;

Figure 3. Cont.



(b) The number of nodes = 200;



(c) The number of nodes = 300;

Figure 3. The packet loss rate varies with the size of the node cache. (a) The number of nodes = 100; (b) the number of nodes = 200; (c) the number of nodes = 300.

In Figure 3, the numbers of nodes in the three subgraphs are 100, 200, and 300. As the number of network nodes increases, the packet loss rate increases due to the increase in forwarding tasks that the nodes must undertake. However, the packet loss rate of the CARA algorithm at different cache queue lengths and node numbers is lower than the other two algorithms. This is because both the CCOR algorithm and the TER algorithm only make routing selections based on the traffic load of the node itself. The CARA algorithm takes into account both the cache remaining rate and the forward average cache remaining rate. These two parameters can accurately reflect the congestion of nodes and forward regions. Especially in respect of the number of nodes, the ability of our algorithm to avoid congestion nodes and areas becomes more evident.

Therefore, when the number of nodes increases, the comparison algorithms' packet loss rate has a more noticeable increase, but the CARA algorithm's packet loss rate increases slightly. Thus, the CARA algorithm can more effectively balance the network load, and the overall packet loss rate performance is better than the other two algorithms. Furthermore, for the same number of nodes, the difference between the three algorithms gradually decreases with cache queue length. Thus, as the cache queue length increases, the routing algorithms alleviate the node's congestion to a certain extent while the data generation

speed remains the same. However, the CARA algorithm still maintains the lowest packet loss rate, reflecting its excellent congestion control performance.

4.2. The Average Hops

For evaluating routing algorithms, the delay in the network data transmission is also an important evaluation index. Therefore, for node caches of 20 and 30 packets, simulation experiments were performed under different node numbers. Figure 4 shows how the average hops vary with the number of nodes.

The average number of hops of the three algorithms increases as network nodes increase. When the node cache size is 20 packets, both of CARA and CCOR consider the path length and single-hop distance. Hence, the average number of hops of the algorithms is relatively low. The TER algorithm has the highest average number of hops because it attempts to find the shortest single-hop distance in the routing decision process, resulting in a higher average number of hops. To a certain extent, this aggravates congestion, resulting in the dropping of many packets. When the size of the node cache increases to 30 packets, the network drop rate decreases significantly. This is because the single-hop transmission distance element dominates the selection of the next-hop node. The CARA algorithm and CCOR algorithm emphasize the maximum transmission distance of the single-hop and strive to transfer the data to the sink node to the greatest extent. As a result, the average number of hops on routes between the two algorithms decreases compared to the cache size of 20 packets, but the CARA algorithm retains some advantages. In contrast, for the TER algorithm, the average number of hops on the route increases due to the smallest single-hop transmission distance.

4.3. The Average Energy Consumption

Since the research object of this article is a wireless sensor network with adequate energy, the proposed CARA routing algorithm does not evaluate energy consumption when selecting the next-hop node. However, because energy consumption is a critical evaluation index of wireless sensor networks, we compared and analyzed the network's total energy consumption. When the cache queue lengths are set to 20 and 30 packets the average energy consumption is simulated for different node counts, as shown in Figure 5.

As the number of network nodes increases, the average energy consumption of the three algorithms increases. As shown from the Figure 5, the TER algorithm has the highest average energy consumption. Because it has a high average number of hops, it does not have congestion awareness, resulting in higher energy consumption requirements to make its average energy consumption the highest. However, the CARA algorithm and the CCOR algorithm emphasize the maximum transmission distance for a single hop, thus maintaining low average energy consumption. When the number of nodes is less than 200, the CARA algorithm has a slight advantage over the CCOR in both cache queue size settings. This is because the CARA algorithm has a lower average number of hops, resulting in lower energy consumption. However, when the number of nodes exceeds 200 and the network data traffic increases due to the rise in node forwarding tasks, CCOR's energy-based routing algorithm plays an important role, with the average energy consumption of the two being essentially the same.

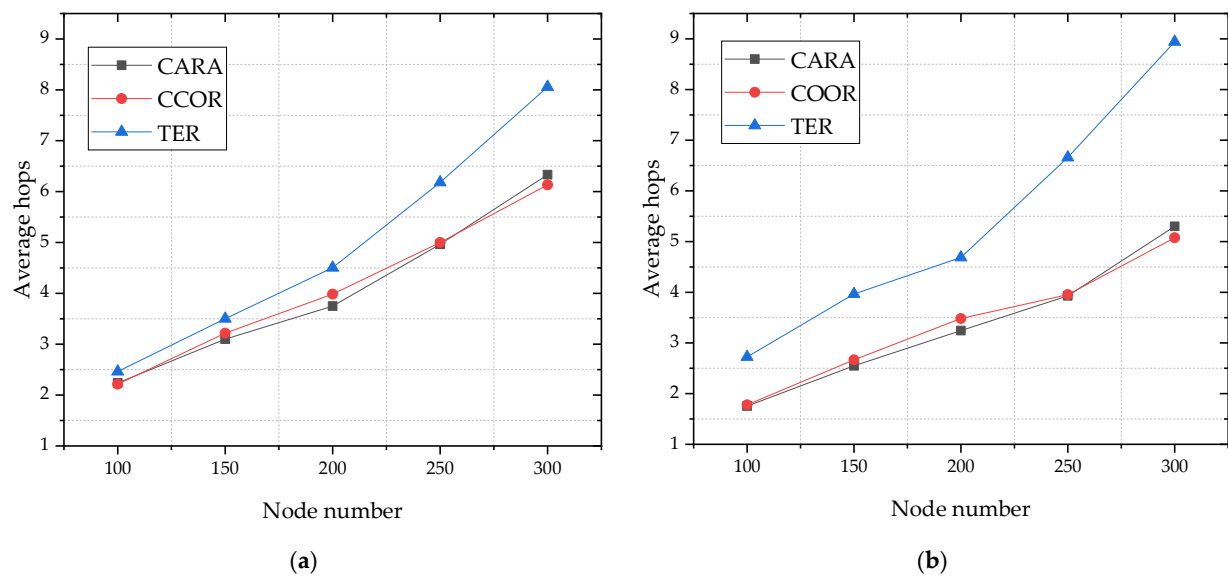


Figure 4. The average hops with the number of nodes. (a) Cache queue lengths = 20; (b) cache queue lengths = 30.

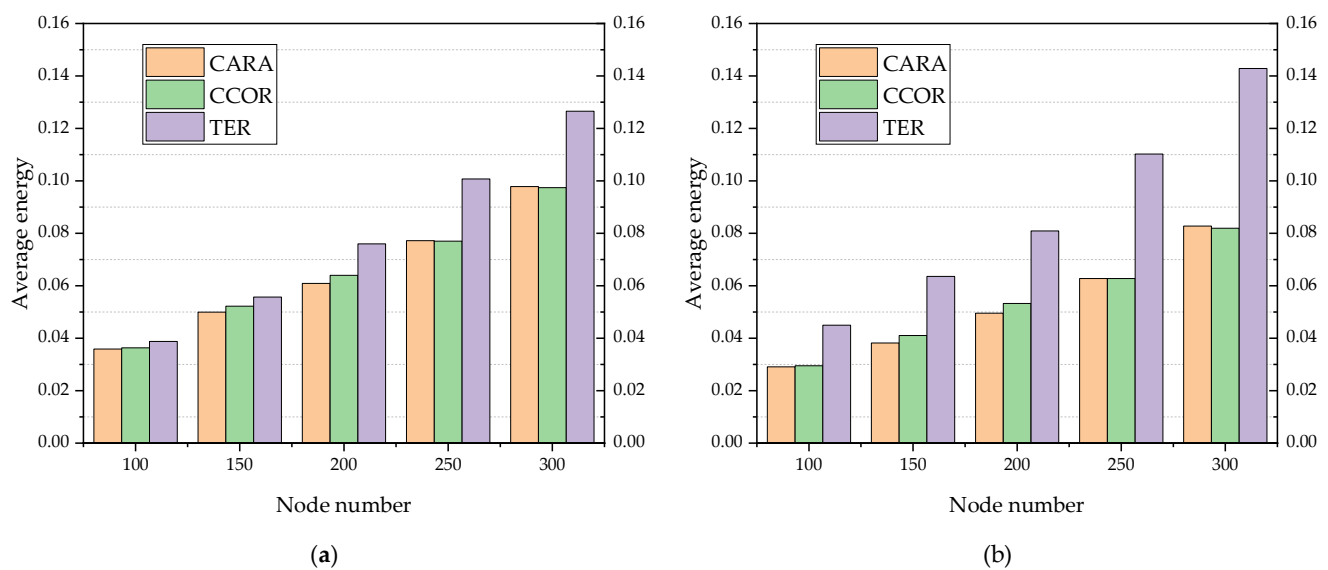


Figure 5. The average energy consumption (mJ) with the number of nodes. (a) Cache queue lengths = 20; (b) cache queue lengths = 30.

5. Conclusions

We have presented an active WSN congestion-aware routing algorithm (CARA) for network congestion control. The algorithm considers both spatial relationship and traffic load and puts forward four route evaluation parameters: forward rate, node load factor, cache remaining rate, and forward average cache remaining rate. The multiparameter fusion method is used to make routing selections. As a result, the CARA algorithm realizes the congestion perception of the sensor network node itself and the nearby area and optimizes network transmission performance. Our simulation results indicate that the CARA algorithm performs better than the TER and CCOR algorithms in terms of congestion control, which significantly reduces the network packet loss rate and results in lower average network energy consumption and delay.

In future research, we will mainly focus on the application of WSNs in the power and energy sectors. For integrated energy management and microgrid management in industrial parks, we plan to build a WSN test validation environment based on the IEEE

802.15.4 protocol. The control packet is an important part of the routing protocol. We will comprehensively analyze the congestion caused by control packets and data packets based on real-world scenarios and physical layer protocols. We also will test and validate our algorithm in real-world scenarios to further analyze and improve the algorithm based on practical data.

Author Contributions: Conceptualization, B.Q. and J.Y.; methodology, B.Q. and J.Y.; software, J.Y.; validation, J.Y.; formal analysis, J.Y.; investigation, J.Y.; resources, B.Q.; data curation, J.Y.; writing original draft preparation, J.Y.; writing—review and editing B.Q.; visualization, J.Y. Both authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 51677065).

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

References

1. Qian, Z.H.; Wang, Y.J. Internet of things-oriented wireless sensor networks review. *J. Electron. Inf. Technol.* **2013**, *35*, 215–227. [\[CrossRef\]](#)
2. Liu, J.X.; Xiong, K.; Fan, P.Y.; Zhong, Z.D. RF Energy Harvesting Wireless Powered Sensor Networks for Smart Cities. *IEEE Access* **2017**, *5*, 9348–9358. [\[CrossRef\]](#)
3. Matta, N.; Ranhim-Amoud, R.; Merghem-Boulahia, L.; Irad, A. A wireless sensor network for substation monitoring and control in the smart grid. In Proceedings of the 2012 IEEE International Conference on Green Computing and Communications, Besancon, France, 20–23 November 2012; pp. 203–209.
4. Liu, Y.; Xiong, N.; Zhao, Y.; Vasilakos, A.V.; Gao, J.; Jia, Y. Multi-layer clustering routing algorithm for wireless vehicular sensor networks. *IET Commun.* **2010**, *4*, 810–816. [\[CrossRef\]](#)
5. Hu, X.M.; Dong, S.F.; Wang, X.D.; Han, Z.X. Research development of application mode of wireless sensor networks in military field. *Transducer Microsyst. Technol.* **2011**, *30*, 1–3. [\[CrossRef\]](#)
6. Song, W.-Z.; Huang, R.; Xu, M.; Shirazi, B.; LaHusen, R. Design and deployment of sensor network for real-time high-fidelity volcano monitoring. *IEEE Trans. Parallel Distrib. Syst.* **2010**, *21*, 1658–1674. [\[CrossRef\]](#)
7. Mathur, A.; Newe, T.; Rao, M.; Elgenaidi, W.; Toal, D. Cluster head election and rotation for medical-based wireless sensor networks. In Proceedings of the 2017 4th International Conference on Control, Decision and Information Technologies (CoDIT), Barcelona, Spain, 5–7 April 2017; pp. 0149–0154.
8. Ahmad, A.; Javaid, N.; Khan, Z.A.; Qasim, U.; Alghamdi, T.A. \$(ACH)^2\$: Routing Scheme to Maximize Lifetime and Throughput of Wireless Sensor Networks. *IEEE Sens. J.* **2014**, *14*, 3516–3532. [\[CrossRef\]](#)
9. Shah, S.; Beferull-Lozano, B. Joint sensor selection and multi-hop routing for distributed estimation in ad-hoc wireless sensor networks. *IEEE Trans. Signal Process.* **2013**, *61*, 6355–6370. [\[CrossRef\]](#)
10. Shen, B.; Zhang, S.-Y.; Zhong, Y.-P. Cluster-based routing protocols for wireless sensor networks. *J. Softw.* **2006**, *17*, 1588–1600. [\[CrossRef\]](#)
11. El Hajji, F.; Leghris, C.; Douzi, K. Adaptive Routing Protocol for Lifetime Maximization in Multi-Constraint Wireless Sensor Networks. *J. Commun. Inf. Netw.* **2018**, *3*, 67–83. [\[CrossRef\]](#)
12. Tang, L.; Lu, Z. DS Evidence Theory-Based Energy Balanced Routing Algorithm for Network Lifetime Enhancement in WSN-Assisted IOT. *Algorithms* **2020**, *13*, 152. [\[CrossRef\]](#)
13. Pandey, D.; Kushwaha, V. An exploratory study of congestion control techniques in Wireless Sensor Networks. *Comput. Commun.* **2020**, *157*, 257–283. [\[CrossRef\]](#)
14. Lu, J.; Zhu, Y.; Xu, Z. A reliable wireless sensor network routing method for power transmission line monitoring. *Power Syst. Technol.* **2017**, *41*, 644–650. [\[CrossRef\]](#)
15. Adu-Manu, K.S.; Adam, N.; Tapparello, C.; Ayatollahi, H.; Heinzelman, W. Energy-Harvesting Wireless Sensor Networks (EH-WSNs): A Review. *ACM Trans. Sens. Netw.* **2018**, *14*, 10. [\[CrossRef\]](#)
16. Wang, C.; Lin, H.; Jiang, H. CANS: Towards congestion-adaptive and small stretch emergency navigation with wireless sensor networks. *IEEE Trans. Mob. Comput.* **2015**, *15*, 1077–1089. [\[CrossRef\]](#)
17. Liu, W.; Cui, Y.; Zhao, Z. Wireless Sensor Network Route Optimization Based on Improved Ant Colony-Genetic. *Int. J. Online Eng.* **2015**, *11*, 4. [\[CrossRef\]](#)
18. Elappila, M.; Chinara, S.; Parhi, D.R. Survivability Aware Channel Allocation in WSN for IoT applications. *Pervasive Mob. Comput.* **2020**, *61*, 101107. [\[CrossRef\]](#)
19. Fu, X.; Yao, H.; Yang, Y. Exploring the invulnerability of wireless sensor networks against cascading failures. *Inf. Sci.* **2019**, *491*, 289–305. [\[CrossRef\]](#)
20. Zawodniok, M.; Jagannathan, S. Predictive congestion control protocol for wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2007**, *6*, 3955–3963. [\[CrossRef\]](#)

21. Hao, X.-C.; Jia, N.; Liu, B. Multi-path optimizing routing protocol based on predicting congestion for wireless sensor network. *Dianzi Yu Xinxu Xuebao J. Electron. Inf. Technol.* **2011**, *33*, 1261–1265. [[CrossRef](#)]
22. Jiang, X.; Qi, J.-D.; Cao, Y.-J.; Zhao, Y.-D. Priority of Energy Congestion Relief Scheme in Wireless Sensors Networks. Available online: https://en.cnki.com.cn/Article_en/CJFDTotal-SJSJ201102011.htm (accessed on 30 June 2021).
23. Bhandari, K.S.; Hosen, A.; Cho, G.H. CoAR: Congestion-Aware Routing Protocol for Low Power and Lossy Networks for IoT Applications. *Sensors* **2018**, *18*, 3838. [[CrossRef](#)] [[PubMed](#)]
24. Ding, W.; Tang, L.; Ji, S. Optimizing routing based on congestion control for wireless sensor networks. *Wirel. Netw.* **2016**, *22*, 915–925. [[CrossRef](#)]
25. Ding, W.; Tang, L.; Feng, S. Traffic-aware and energy-efficient routing algorithm for wireless sensor networks. *Wirel. Pers. Commun.* **2015**, *85*, 2669–2686. [[CrossRef](#)]
26. Tang, L.; Liu, H.; Yan, J. Gravitation Theory Based Routing Algorithm for Active Wireless Sensor Networks. *Wirel. Pers. Commun.* **2017**, *97*, 269–280. [[CrossRef](#)]
27. Liu, Y.; Yin, H.; Wu, T. Design of Wireless Sensor Network Module in Power Consumption Information Collection System Based on IPv6. In Proceedings of the International Conference on Logistics Engineering, Management and Computer Science (LEMCS 2015), Shenyang, China, 29–31 July 2015; pp. 719–721.
28. Zhang, D.; Li, G.; Zheng, K.; Ming, X.; Pan, Z. An Energy-Balanced Routing Method Based on Forward-Aware Factor for Wireless Sensor Networks. *IEEE Trans. Ind. Inform.* **2014**, *10*, 766–773. [[CrossRef](#)]