Edge Network Routing Protocol Base on Target Tracking Scenario

Zhongyi Zhang¹ · Weihua Zhao¹ · Ouhan Huang¹ · Gangyong Jia¹ · Youhuizi Li¹ · Songzhu Mei² · Duan Zhao^{3,4}

Accepted: 20 April 2021 / Published online: 8 December 2021
© The Author(s) 2021

Abstract

Edge computing perfectly integrates cloud computing centers and edge-end devices together, but there are not many related researches on how the edge-end node devices work to form an edge network and what the protocols used to implement the communication among nodes in the edge network. Aiming at the problem of coordinated communication among edge nodes in the current edge computing network architecture, this paper proposes an edge network routing and forwarding protocol based on target tracking scenarios. This protocol can meet the dynamic changes of node locations, and the elastic expansion of node scale. Individual node failures will not affect the overall network, and the network ensures efficient real-time with less communication overhead. The experimental results display that the protocol can effectively reduce the communications volume of the edge network, improve the overall efficiency of the network, and set the optimal sampling period, so as to ensure that the network delay is minimized.

Keywords Edge computing · Target tracking protocol · Edge network · Routing and forwarding

1 Introduction

Edge computing is the extension and supplement of cloud computing in the era of big data, especially in the era of

the Internet of Things [1], thus perfectly connecting the traditional mobile edge end with cloud computing data centers. Edge computing refers to a new type of computing model that performs computing at the edge of the network. Its core concept is that “computation should be closer to the source of the data and users”, which greatly promotes the quality of services, including computing, storage, and networking, thus reducing costs [1]. However, the most critical issue in edge computing is how to build a network among mobile edge nodes, and what routing protocol should be applied to forward data. Only by solving these two problems can an edge network be successfully constructed to achieve network communication among nodes.

The emergence of edge networks is partly to compensate for the shortcomings of high latency in cloud computing. Meanwhile, most edge network applications are very sensitive to delays, especially edge networks in target tracking scenarios [2]. However, current researches on routing protocols for edge networks are still based on the Internet of Things, and even some edge nodes use TCP/IP network protocol architecture when communicating. For traditional Internet communications, TCP/IP network protocol architecture has proven to perform very well. However, for new mobile edge networks, the traditional network protocol architecture cannot meet the characteristics of node mobility, and the master-slave model can easily cause some node overload, which is not conducive to network load balancing.

✉ Zhongyi Zhang
makingbug@foxmail.com

Gangyong Jia
gangyong@hdu.edu.cn

Songzhu Mei
sz.mei@nudt.edu.cn

Duan Zhao
Duan.Zhao@cumt.edu.cn

¹ Department of Computer Science, and Key Laboratory of Complex Systems Modeling and Simulation, Hangzhou Dianzi University, Hangzhou, 310018, China

² Science and Technology on Parallel and Distributed Processing Laboratory (PDL), National University of Defense Technology, Changsha, 410073, China

³ State-Province Joint Engineering Lab of Mine Internet Application Technology, Xuzhou, 221008, China

⁴ National and Local Joint Engineering Laboratory of Internet Application Technology on Mine, School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, 221116, China

The wireless sensor network routing protocol is a non-master-slave peer-to-peer routing protocol, and can overcome traditional TCP/IP defects. At the same time, in some scenarios, wireless sensors can move in a small range, which is similar to edge networks. However, the application scenarios of wireless sensors are quite different from the application scenarios of edge networks. Edge nodes are not as sensitive to energy consumption as wireless sensors. Many wireless sensor network routing protocols extend the lifetime of a node at the expense of other network performances, while this cost is unacceptable in edge networks.

For mobile edge networks, especially multi-node collaborative target tracking, based on target tracking scenarios, we urgently hope that a new type of network protocol architecture can meet the characteristics of frequent edge, low latency, and scalability of mobile edge computing nodes. Such a hard requirement poses a challenge to today's edge collaborative communication problems and we also need an edge network routing protocol that can meet the following requirements:

1. The dynamic changes of network nodes in real time. The nodes and goals of the network are changed in real time, and the network must be able to adapt to the dynamic changes of the nodes.
2. The network must support elastic changes in the number of nodes. Nodes in the network may increase or decrease at any time due to various sudden conditions, so the network must remain sufficiently sensitive to such changes.
3. The network must complete the target tracking task with the least amount of communication, which can efficiently apply the network bandwidth and enhance the load capacity of the network.

The main contribution of this paper is to propose a dynamic routing protocol for edge networks on the basis of target tracking scenarios. The routing protocol mainly includes three parts: the construction of the initial network, the routing and forwarding rules for target tracking, and the dynamic construction of the network topology. This routing protocol overcomes the shortcomings, including single node data overload and high communication delay of traditional TCP/IP protocols and wireless sensor networks, so collaboration among edge nodes is completed with the lowest delay, and the network is guaranteed to use less traffic. In order to complete the task, the network also supports the dynamic and flexible expansion of nodes.

The first chapter of this article introduces some problems of existing routing protocols in the edge network environment, and the specific requirements for routing protocols in the edge network environment. The second chapter of this paper introduces several commonly used

routing protocols for mobile edge networks and the related research work of researchers in the industry. The third chapter details the working mechanism of the target tracking routing protocol of the edge network, and analyzes the main factors that affect the network performance. The fifth chapter introduces the relevant experimental platform of this thesis and analyzes and summarizes the experimental results as well.

2 Related works

In recent years, with the improvement of hardware performance and the diversity of people's needs [3–5], mobile edge networks have been widely used [6–9] in disaster tolerance monitoring of complex environments [10, 11], vehicle self-organizing networks [12, 13], collaborative target tracking, and so on. The mobile edge network is a peer-to-peer network architecture and all nodes are completely equal. Each node can either accept requests from other nodes to provide services, or send service requests to other nodes. All nodes are equal in status and can join and leave the network at any time. If any node fails, the function of the overall network could not be affected. Therefore, the network is highly adaptive and indestructible. This section details related works on routing protocols for mobile edge networks.

2.1 Cluster-based topology control

In a cluster network, mobile edge network is divided into clusters. Usually, each cluster consists of several nodes that are close to each other. A cluster head is selected in each cluster, and these cluster heads can form a higher-level network cluster that can be further divided into several clusters. After selecting the cluster head, a higher-level network can be formed, so as to reach the highest-level network. The intra-cluster network is a local planar structure and the nodes and the group capital in the network are dynamically changed. The cluster head is responsible for data forwarding among clusters. This method of clustering the network undoubtedly simplifies the structure of the network, and does not need to maintain a large number of routing tables. Every node only needs to maintain the routing table in the local network. The size of the node is no longer limited, and the network capacity can be expanded by rationally increasing the number of network clusters and the number of network levels. Nodes head is randomly selected, so they have the advantage of strong survivability.

At present, many scholars who study clustering algorithms proposed related routing protocols. Based on fuzzy logic algorithm, Neamatollahi [14] proposed a routing protocol to adjust the clustering radius. This protocol improves

the determination of non-uniform clustering boundaries, and has made progress in network life and energy saving. By scheduling clustering tasks to reduce network energy consumption and extend network life, Neamatollahi [15] put forward distributed energy-saving scheme to cluster wireless sensor networks. Lin [16] et al designed a mobile area-based network architecture. Nodes cooperate with each other to form a dynamic mobile area, which helps broadcast information. Based on information entropy Hoang [17] came up with a clustering scheme for dynamic self-organizing network routing protocols, which uses a cost function to solve the energy and delay problems in the network. Taheri [18] proposed a logically fuzzy clustering routing protocol that increases the stability of the cluster head and the utilization of node energy, and can effectively extend the network life and save the energy of the node. Azharuddin [19] and others put forward a fuzzy logic-aware distributed clustering protocol to dynamically adjust the number of nodes in a cluster to effectively reduce the energy consumption of the nodes in the cluster, so the network can survive longer. Sharma [20] proposed a set-based routing protocol that constructs a tree in the area of the network set to effectively reduce the data transmission delay.

2.2 Routing protocol based on ant colony algorithm

A routing protocol based on the greedy algorithm is a simpler routing and forwarding protocol, and has attracted widespread attention [21] once it was proposed [22].

Based on geographic location, Li [23] et al, proposed a greedy routing protocol algorithm based on geographic location. The ABPP adaptive beacon scheme can dynamically predict the node position and adjust the beacon frequency, which can effectively reduce overhead. Huang [24] introduced an energy algorithm to optimize the traditional GPRS protocol and proposed the EA-GPRS protocol, taking factors such as node energy, node location, and node energy collection into account. Based on the greedy algorithm, Lin [25] proposed an MPGR protocol that effectively predicted the mobility of address locations, and proposed two-hop peripheral forwarding to effectively reduce the effects of network holes. Aboki R [26] introduced a new routing protocol of PLAR, which proposed to use new location services to provide location information for each node, and to improve the GPRS protocol, so as to predict the target's motion information. Veerasamy A [27] researched the application of opportunistic routing in location-assisted routing protocols, and enhanced the location-assisted routing protocols, thus improving the linkability of routing. LIU [28] proposed a greedy anti-void routing (GAR) protocol that uses UDG graph's boundary finding technology to efficiently solve the void problem with increased routing

efficiency. At the same time, a cross navigation mechanism is proposed to obtain the best traversal direction. HSU MT [29] proposed a non-geographical greedy routing protocol that does not require node location information, and sets virtual coordinates to forward data, thus making routing configuration being more flexible. Zhu [30] discovered the impact of multi-level functions on network characteristics through outdoor transmission of information, and proposed a greedy opportunistic routing protocol that is oriented to multiple scenarios. The protocol responds to the effects of multi-level structures by means of probability calculation and greedy forwarding.

3 Edge network target tracking routing protocol

The node characteristics of the edge network are very different from traditional wireless sensor networks. The nodes themselves usually have sufficient energy and their computing and storage capabilities far exceed those of ordinary IoT nodes. Edge nodes also have the characteristics of frequent movement, and frequent changes in network structure also increase energy consumption. Therefore, based on the target tracking scenario, this paper proposes an edge network routing protocol. The protocol mainly solves the problem of cooperative communication among edge nodes, and optimizes the network traffic on the premise of ensuring high real-time network.

3.1 Edge network object tracking routing protocol architecture

Based on the edge network in the context of target tracking, this paper proposes a routing communication protocol. This protocol builds scattered edge nodes into a unified edge network. The nodes work together to complete the target tracking task with the shortest delay. The routing protocol mainly includes the following three parts:

1. The initial state of the edge nodes builds the global topology of the network.
2. After a node finds a target, it plans nodes that tracks the target together and plans the shortest path to send information, and sends the target location information to the nodes that are expected to work together.
3. The edge nodes and targets change in the next cycle, rebuild the global network, plan the path, and send information.

Figure 1 exhibits the overall architecture of the edge network routing protocol based on the target tracking scenario.

Fig. 1 overall architecture of edge network routing protocols

3.2 Building a topological network

The mutual communication among network devices mainly depends on the interconnection of each network device, so as to achieve interconnection with each other. Therefore, the construction of the network topology is very important for the interconnection of network equipment. The edge network routing protocol proposed in this paper is based on a peer-to-peer network. Due to the mobility and disorder of edge nodes, the network architecture also needs to meet the characteristics of dynamic changes and

efficient communication. The followings mainly introduce the specific process of network construction.

3.2.1 Selection of weight

One of the most important parts of an edge network is the network weight. The determination of the weight directly affects the path planning in the network. The delay of data transmission between nodes and the traffic of the entire network are closely related to path planning, so the selection of weights is very important in the entire edge network.

Fig. 2 signal propagation diagram in wireless communication

Figure 2 shows that the wireless communication of the edge node in the target tracking scene is affected by direct electromagnetic signals, reflection, and refraction. This is also a common multipath effect in wireless communication.

The main factor that affects the communication quality is the distance when the communication equipment and the environment are constant. Generally, the communication quality is worse as the distance is longer, but there are some cases where the communication quality on the long distance is better than that on the short distance. This is due to the interference phenomenon of the direct and reflected waves superimposed. The same phase will increase the signal and the communication quality will increase as well. The opposite phase will weaken the signal and the communication quality will deteriorate as well. What ultimately affects the communication quality is the strength of the signal received by the receiving source, also known as the received power. The formula (1) points out that the received power is mainly related to those factors.

$$P_R = P_T G_T G_R \left[\frac{\lambda}{4\pi d} \right]^2 \quad (1)$$

According to the above formula, it can be seen that the strength of the received power is mainly correlated with three factors: 1) the wavelength of the electromagnetic wave, 2) the attributes of the nodes themselves, and 3) the distance among nodes. The factor that affects communication quality mainly is the distanced among nodes, and the received power is inversely proportional to d^2 , so the weight of the network edge is finally determined to be d^2 (Table 1).

3.2.2 Network construction in initial state

According to the calculation rule of the weight, the weight w between the two points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ can be calculated.

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

To build an edge network is to maintain the connection relationship and the weight among nodes. The topology

Table 1 Parameter meaning in the formula

name	description
P_R	Received power
P_T	Transmitter antenna gain
G_T	Transmit power
G_R	Receiver antenna gain
λ	Electromagnetic wave length
d	Communication distance

information of the edge network can be represented by an adjacency matrix M . Where N is the number of edge nodes in the edge network. $w_{i,j}$ represents the weight between node i and node j .

$$M = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,N} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N,1} & w_{N,2} & \cdots & w_{N,N} \end{bmatrix} \quad (3)$$

In the actual network, not all nodes can be connected. According to the formula of $w_{i,j}$ only the nodes that are connected can calculate the weight. The weight among nodes that cannot be connected is specified as $+\infty$. The distance between two nodes exceeds a threshold L meanings that the nodes cannot directly communicate with each other. The specific calculation of $w_{i,j}$ is as formula (4):

$$w_{i,j} = \begin{cases} \frac{1}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}} & (d \leq L) \\ +\infty & (d > L) \end{cases} \quad (4)$$

Threshold L is affected by the sensing range of actual sensors. Different sensors can have different sensing ranges. Only the nodes that can perceive each other can communicate, that is, it is meaningful to calculate the weight of the edge when the distance d between the nodes is not greater than L . According to the actual selection of the sensor, L is selected as 6m in this experiment.

The construction of the edge network essentially maintains the adjacency matrix M , and each weight $w_{i,j}$ in the matrix needs to be calculated based on to the position

Fig. 3 Edge nodes build a global network topology map

of the nodes. Each edge node i stores a copy of the position information of other nodes, so each node can build the global topology of the entire edge network.

$$position[P_1(x_1, y_1), P_2(x_2, y_2), \dots, P_n(x_n, y_n)] \quad (5)$$

Each node stores a copy of the location information of other nodes. Although a part of the storage space and network communication overhead is wasted, each edge node can build a global network. In the process of information transmission, each node can plan the best path to ensure the real-time nature of information transmission.

The network traffic in the network is not only related to the transmission rules, but also closely related to the specific network topology. A global network topology is constructed with the node distribution shown in Fig. 3, and the edge network is constructed based on the above construction rules. The traffic of the entire network node is shown in Fig. 4. When the location information received by each node changes, the newly obtained message is broadcast to neighbors, which is a one-to-many process. Therefore, the amount of communication received by each node in the figure is much larger than the amount of transmission. Carefully observing the ratio of the amount of received information to the amount of information sent by each node, and is basically the number of edges connected by each node, which is basically in line with the expected result: When the nodes in the network obtain the location information of all nodes, the network will converge quickly and will not cause network flooding due to unlimited broadcast information.

3.3 Collaborative tracking algorithm

After the edge nodes successfully build the network topology, according to the position-list and the adjacency matrix M , each node can perceive the network topology

Fig. 4 Network construction traffic

and the position of other nodes. Building an edge network allows edge nodes to work together to complete tasks. In the edge network in the target tracking scenario, the task of the node is to cooperate to complete the tracking target. The followings mainly introduces the problem of coordinator task division in the edge network and the path planning problem in cooperative communication.

In order to complete the task of target tracking, the edge network needs to allocate the entire task to some edge nodes for execution. Given K nodes among them, assign tasks to cooperate to complete target tracking.

The original intention of the edge network is to make up for the shortcomings of long delays in cloud computing networks, and to deal with some delay-sensitive tasks. In order to ensure that the target tracking task is completed in the shortest time, it is necessary to select the $Point_{work}$ node set based on certain rules. The position of the target is goal (x, y) . Based on the shortest distance from the node to the goal, K nodes are selected. In the case of the same node movement speed, the closer the node is to the target, the faster it will catch up with the target, and the shorter the processing task delay.

$$Point_{work} = \min \left(\sqrt{(x - x_i)^2 + (y - y_i)^2} \right) (1 \leq i \leq N) \quad (6)$$

In Fig. 5, the distance among the nodes in the network and the goal is calculated. Sorting the distances to find the three closest nodes to the goal. It can be seen that the three most recent nodes are $\{D, E, F\}$. In this sampling period T , the three nodes are divided to perform the target tracking task: The three nodes in the sector area.

In this process, the task decision-making process is completed directly at the edge network layer. There is no need to upload tasks to the cloud center such as the traditional cloud computing architecture model, which greatly reduces the time consumed by the node task upload process. At the same time, the edge network selection the edge nodes closest to the target can complete the task in the shortest time, which reflects the advantage of low latency of the edge network.

The edge network layer decides the nodes to assign the task, while some nodes assigned to the task may not know the location of the target. As shown in Fig. 5, nodes D and E do not know the location of the target goal. At this time, it is necessary to send the position of the target to nodes including D and E who do not know the position of the target but they are assigned tasks. At this time, it is necessary to send the position of the target to nodes such as D and E who do not know the position of the target but are assigned tasks. In order to ensure that the information is sent as quickly as possible, the forwarding path of the information should be the global shortest path, the sum

Fig. 5 Schematic diagram of network node task division

of $w_{i,j}$ should be minimized, and the smaller $\text{sum}(w_{i,j})$ indicates the higher the line communication quality is, the smaller the network delay. The pseudo code of the algorithm is shown in Algorithm 1.

Algorithm 1 Shortest path algorithm in edge network.

Require: *graph* stored by the neighbor matrix

Ensure: Shortest path *dis* and *path*

```

1: MAX =  $+\infty$ 
2: dis = [MAX]*N
3: path = list(range(N))
4: visit = [False]*N
5: dis[0] = 0
6: for  $i = 1; i < N; i++$  do
7:   curMin = MAX
8:   u = -1
9:   for  $node = 1; node < N; node++$  do
10:    if  $\text{visit}[node] == 0 \&\& \text{dis}[node] < \text{curMin}$ 
then
11:      curMin = dis[node]
12:      u = node
13:    end if
14:  end for
15:  if  $u == -1$  then
16:    return  $\langle \text{dis}, \text{path} \rangle$ 
17:  end if
18:  for each  $node \in \text{graph}[u].\text{keys}()$  do
19:    if  $\text{visit}[node] == \text{False}$  then
20:      dis[node] = min(dis[node], dis[u] + graph[u][node])
21:      path[node] = u
22:    end if
23:  end for
24: end for
25: return  $\langle \text{dis}, \text{path} \rangle$ 

```

3.4 Dynamically build a topology network

The essence of the edge network is the set of edge nodes and the set of edges with weights. The dynamic construction of the topology network is to maintain $\text{position}(nT)$ and $w_{i,j}(nT)$, where these represents position information and edge weight at time nT , respectively.

The process of dynamically constructing the network topology is to maintain the $\text{position}(nT)$ in real time in each period T . To ensure that the communication overhead is maintained at a low level during the dynamic construction of the network, it is only necessary to send the location information of the nodes whose locations have changed to the entire network. Creating a $\text{flag}(nT)$ array to record whether each node's position changes at nT .

In order to rebuild the network topology, broadcast the location information of the node marked 0 in $\text{flag}(nT)$ to other nodes in the network. The pseudo code of the algorithm is shown in Algorithm 2.

Algorithm 2 Broadcast node location information algorithm.

Require: *graph* stored in adjacency matrix, node i

Ensure: *graph* after updating location information

```

1: Q = [i] // Construct queue Q
2: visit = [i] // Construct a set of tags
3: while  $Q \neq \text{null}$  do
4:   cur = Q.popLeft() // Dequeue the current node
5:   update(cur) // Update current node information
6:   for each  $j \in \text{graph}[\text{cur}].\text{neighbor}$  do
7:     if  $j \notin \text{visit}$  then
8:       Q.append(j) // Neighbors enter the queue
9:     end if
10:  end for
11: end while return graph

```

In the process of dynamically constructing the edge network, the sampling interval T has a very important influence on various aspects including network traffic and delay. The tasks completed by the edge network in a cycle T have three parts: broadcast the i -node position of $f_i = 1$; divide the tasks of the network nodes after identifying the target, and plan the shortest path for data forwarding and complete the data forwarding. The traffic $data_{nT}$ comes from the broadcast information of node i and the forwarding of data in the shortest path in a period T .

$$data_{nT} = \sum_1^n data_i | (f_i = 1) + data_{path} \quad (7)$$

Within the same unit time $Time$, for different periods T , $Time/T$ data communications are required, and each communication cost is $data_{nT}$. Therefore, in unit time $Time$, the communication cost of the network is $data_{Time}$.

$$data_{Time} = \frac{Time}{T} \sum_{i=1}^n data_i | (f_i = 1) + data_{path} \quad (8)$$

It can be found from the formula that in the unit time of the network, the communication overhead of the network is related to the network scale, path planning, and the sampling period T of the network. When the sampling period is reduced, the communication overhead in the network is less. If the scale in the network is unchanged, the communication overhead counted in multiple cycles $data_{nT}$ is also approximately the same. The communication overhead of the network is roughly inversely proportional to the sampling period T of the network.

Network latency is the most important performance indicator in edge networks. When the target appears in the edge network, we hope that the network can complete the tracking of the target in the shortest time. The delay in a sampling period T $delay_T$ is the delay of node i broadcasting information and the delay of information transmission on the path. If the target appears in the network at the moment when it has been sampled in this cycle, the target needs to be discovered in the next cycle T , so the network delay $delay_T$ is at most:

$$delay_T = T + delay_{path} + \max(delay_i) \quad (9)$$

$delay_{path}$ refers to the delay in the shortest path planning, and $delay_i$ marks the delay in broadcasting the location information of the i -node. In each sampling period T under the same network scale, the sum of $delay_{path} + \max(delay_i)$ will fluctuate within a fixed range depending on the specific network structure, so the network delay and sampling period T is closely relate. Generally, the larger the value of T , the higher the delay of the network. The delay and the sampling period T are basically linearly correlated with a positive correlation.

4 Experimental platform and result analysis

4.1 The experimental platform

The target recognition part of the edge nodes in this experiment is based on the Cambrian HBoard intelligent terminal platform. HBoard is a set of edge-end artificial intelligence solutions, and it mainly uses Cambrian 1H8 intelligent terminal processor as the artificial intelligence computing module, and ARM7 as the edge-side artificial intelligence edge platform. Meanwhile, it supports all mainstream deep learning frameworks and can be widely used in the field of computer vision. It can quickly complete target recognition with low power consumption and is very suitable for long-term work on the edge. The software platform uses the caffe framework on the basis of Cambrian hardware architecture adaptation version. The caffe framework is the first mainstream industrial-level deep learning tool, specializing in the field of image processing, with the advantages of simple network model definition and being easy to get started.

The target collaborative tracking platform is mainly based on the secondary development of the xrobot platform, which newly adds the optimization of network communication and reduces network traffic without increasing delay. Xrobot is mainly equipped with ROS robot operating system. ROS system can operate the underlying hardware, including radar, odometer, attitude sensor, motor, etc. in a modular form, which is more convenient and fast. At the same time, ROS platform supports a distributed architecture, which is very convenient for the management of multiple ROS nodes, and the management of multiple nodes can be completed by changing only a few configurations. In this paper, xrobot platform is combined with the Cambrian Hboard platform, so as to complete the collaborative work of nodes in the edge network. Figure 6 illustrates Hboard platform and xrobot platform.

4.2 Experimental results and analysis

4.2.1 Impact of cycle on network traffic

The nodes in the edge network have the characteristics of mobility, etc. Frequent movement of the nodes will destroy the topology of the edge network, so the topology of the edge network should also be constructed dynamically. The construction of local network must be completed within each sampling period T , and each construction of the network will have a communication overhead on the network. Therefore the selection of the sampling period will have direct impact on the network communication overhead. Figure 7 gives information that the statistics of the traffic of

Fig. 6 Hboard platform and xrobot platform

the edge network completes the target tracking task in 10s in different cycles.

It is obvious from Fig. 7 that the sampling period has huge impacts on the traffic of the edge network. In the same sampling time, the shorter the sampling period of the edge network is, the greater the communication overhead of the network is. In the same period, the larger the sampling period of the edge network is, the fewer the number of updates, and the communication volume required for network update in each state is basically the same, which is basically equivalent to a breadth-first traversal process and is only related to the node size. Therefore, the longer the period is, the lower the number of network updates per unit time, and the smaller the network traffic is. Therefore, as the period increases, the network traffic consumption decreases. The network communication cost is basically inversely proportional to the edge network update cycle, which is basically consistent with the relationship of formula (8).

Not only is the overall traffic of the entire network affected by the cycle, but also the relationship between the traffic and the cycle of each node in the network is also inversely proportional. The nodes complete the same target tracking task with the same initial position, thus ensuring that they will not be affected by the network topology.

Fig. 7 Relationship between period and traffic

Figure 8 reveals the traffic of each node with periods of 200ms, 400ms, 600ms, and 800ms. Through horizontal comparison, we can see that the traffic of each node in different periods is also inversely proportional to the period, and in different periods. The communication ratio of each node is basically the same, which displays that the impact of the cycle on the network communication is equal to each node in the network, and it will not cause the problem of unbalanced load on the network due to the cycle.

4.2.2 Impact of cycle on the real-time performance of the edge network

The emergence of the edge network is mainly to make up for the shortcomings of the low real-time nature of the cloud computing network architecture. Traditional cloud computing requires a long time cost for task upload and information transmission because the cloud center is far away from the edge device. Real-time is the most important indicator of the edge network. Here we applies response time to measure real-time. Response time refers to the time that is required for a target to appear in the edge network until the target is tracked by the node. Obviously, the shorter the response time is, the better the real-time performance is. The longer the response time is, the worse the real-time performance is. According to the introduction of the previous routing protocol, we can know that the process of the edge network is the longest time consumption, which is the cycle plus the time that is required to build the network and the time that is required for information transmission. The figure below shows the average response time of the edge network under different periods (Fig. 9).

The average response time in formula (9) is related to period T and path transmission delay $delay_{path}$ and broadcast delay $max(delay_i)$. Under the same network scale, $delay_{path} + max(delay_i)$ is basically a fixed value, so the response time is mainly affected by the period T . The response time and T are linearly transformed, which basically meets the relationship of formula (9).

Fig. 8 Traffic volume of each node in different periods

4.2.3 Setting of the edge network cycle

According to the above experimental data, it can be concluded that the cycle has significant impacts on the edge network traffic and the response time of the edge network. Therefore how to set the cycle in an edge network system is the main problem that should be solved in this section. The emergence of edge networks is a network architecture that compensates for the long delay of cloud computing networks. From this perspective, it seems that the cycle should be set as short as possible, so as to ensure that the response time is low, which reflects the characteristics of low latency of edge networks. However, the actual situation is to blindly pursue low latency and set a lower network period, which could lead to an explosive increase in network traffic, thereby increasing the network bandwidth overhead. When the network traffic surges, the network will be congested and the data packets will be lost. Meanwhile, due to network congestion, the data packets resent and packet

loss will waste more time, but they will not achieve the effect of low response time.

For an edge network application, you should evaluate the sensitivity of the application to delay and find the critical delay time; The response time is less than this delay time, so as to meet the application requirements. Otherwise it cannot meet the real-time requirements of the application. In this way, when setting the edge network period, it should ensure that the edge network response time is within the critical delay range as close to the time as possible. This setting can not only meet the requirements of application delay, but also ensure the minimum traffic in the network. For the target tracking edge network in this article, the system delay that can be tolerated is about 500ms. We only need to set the network period to less than 500ms. However, considering the communication overhead of the network, the network period should be set over the system delay 500ms. It should be as small as possible and should be as close as possible to 500ms, so it is set to 450ms in this paper. At present, even if the period is set to the optimal period, it may still fail to guarantee the real-time requirement, because the network traffic at this time has exceeded the actual load capacity; The delay caused by network congestion and packet loss increase. In this case, the application period no longer meet the requirements of the application, and the hardware device of the network node should be replaced with a hardware device with a larger load, which can ensure that the network traffic is less than the load under the optimal period.

Fig. 9 Network delay at different cycles

4.2.4 Comparison of different network scales

The above one horizontally analyzes the impacts of the period on the edge network at the same network scale and how to determine the sampling period in the edge network. This section mainly compares the impact of the increase in the size of the edge network on the communication overhead

Fig. 10 Comparison of the two protocols at different scales

of the network and how much the routing protocol of the edge network optimizes the network traffic compared with traditional broadcast communication method.

According to different network scales in Fig. 10, it is obvious that with the gradual increase of network nodes, the traffic that uses the edge network routing protocol basically increases linearly, while the growth rate of the network traffic that applies the traditional broadcast communication method is much faster than the edge network routing protocol, which also conforms to the mathematical model of the transmission protocol. The edge network routing protocol will first plan the path during data transmission. The path planned by the shortest path algorithm is the guarantee of the minimum network traffic during the communication process, which is basically $O(1)$ complex degree. Consequently, with the increase of the network scale, it basically increases linearly. The broadcast method marks that each node must initiate a broadcast to the surrounding nodes, which is similar to breadth-first traversal. The whole network was traversed once, basically the complexity of $O(N)$, so the network traffic is a power function, which is much larger than the traffic of the edge network routing protocol.

5 Conclusions

In this paper, based on the target tracking scenario, a routing protocol for collaborative communication among nodes in an edge network is proposed. This protocol solves the problems of network construction and routing and forwarding in the process of coordinated communication among edge nodes, which is applicable to application scenarios where nodes move flexibly and network delay is sensitive. The data analysis of the experimental results verifies the relationship between the network sampling period and the traffic and network delay, and gives the selection rules of the optimal sampling period. When

the node size increases, the network traffic volume rises linearly, which is significantly optimized compared with the traditional broadcast method in which the power function increases.

Acknowledgements The work is supported by the National Key Research and Development Program under Grant No. 2019YFC0-118404, the National Natural Science Foundation of China under Grant No. U20A20386, the Zhejiang Key Research and Development Program under Grant No. 2020C01050, the Key Laboratory fund general project under Grant No. 6142110190406, the Zhejiang Natural Science Foundation Project under Grant No. LY19F0200-44. Weihua Zhao is the corresponding author.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Shi W, Sun H, Cao J, Zhang Q, Liu W (2017) Edge computing—an emerging computing model for the internet of everything era. *J Comput Res Dev* 54(5):907–924
- Zhang Q, Zhang Q, Shi W, Zhong H (2017) Enhancing amber alert using collaborative edges: Poster. In: *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, pp 1–2
- Gao H, Kuang L, Yin Y, Guo B, Dou K (2020) Mining consuming behaviors with temporal evolution for personalized recommendation in mobile marketing apps. *Mobile Netw Appl* 25(4):1233–1248
- Yang X, Zhou S, Cao M (2019) An approach to alleviate the sparsity problem of hybrid collaborative filtering based recommendations: The product-attribute perspective from user reviews. *Mobile Netw Appl*
- Jia G, Han G, Jiang J, Liu L, Shu L (2020) Dpam: A demand-based page-level address mappings algorithm in flash memory for smart industrial edge devices. *IEEE Trans Industr Inform* 16(3):1993–2002
- Jia G, Han G, Li A, Du J (2018) Ssl: Smart street lamp based on fog computing for smarter cities. *IEEE Trans Industr Inform* 14(11):4995–5004
- Jia G, Han G, Rao H, Shu L (2018) Edge computing-based intelligent manhole cover management system for smart cities. *IEEE Internet of Things Journal* 5(3):1648–1656
- Jia G, Han G, Du J, Chan S (2019) Pms: Intelligent pollution monitoring system based on the industrial internet of things for a healthier city. *IEEE Netw* 33(5):34–40
- Deng S, Xiang Z, Zhao P, Taheri J, Gao H, Yin J, Zomaya AY (2020) Dynamical resource allocation in edge for trustable internet-of-things systems: A reinforcement learning method. *IEEE Trans Industr Inform* 16(9):6103–6113
- Sun H, Liang X, Shi W (2017) Vu: Video usefulness and its application in large-scale video surveillance systems: An early

- experience. In: Proceedings of the Workshop on Smart Internet of Things, pp 1–6
11. Gao H, Liu C, Li Y, Yang X (2020) V2vr: Reliable hybrid-network-oriented v2v data transmission and routing considering rsus and connectivity probability. *IEEE Trans. Intell. Transp. Syst.*: 1–14
 12. Gao H, Huang W, Duan Y (2021) The cloud-edge-based dynamic reconfiguration to service workflow for mobile ecommerce environments. *ACM Trans Internet Technology (TOIT)*
 13. Yin Y, Cao Z, Xu Y, Gao H, Mai Z (2020) Qos prediction for service recommendation with features learning in mobile edge computing environment. *IEEE Trans Cogn Commun Netw* 6(4):1136–1145
 14. Neamatollahi P, Naghibzadeh M (2018) Distributed unequal clustering algorithm in large-scale wireless sensor networks using fuzzy logic. *J Supercomput* 74(6):2329–2352
 15. Neamatollahi P, Naghibzadeh M, Abrishami S, Yaghmaee M-H (2017) Distributed clustering-task scheduling for wireless sensor networks using dynamic hyper round policy. *IEEE Trans Mob Comput* 17(2):334–347
 16. Lin D, Kang J, Squicciarini A, Wu Y, Gurung S, Tonguz O (2016) Mozo: A moving zone based routing protocol using pure v2v communication in vanets. *IEEE Trans Mob Comput* 16(5):1357–1370
 17. Hoang D, Kumar R, Panda S (2010) Fuzzy c-means clustering protocol for wireless sensor networks. In: *IEEE international symposium on industrial electronics*. IEEE, p 2010
 18. Taheri H, Neamatollahi P, Younis OM, Naghibzadeh S, Yaghmaee MH (2012) An energy-aware distributed clustering protocol in wireless sensor networks using fuzzy logic. *Ad Hoc Netw* 10(7):1469–1481
 19. Azharuddin M, Jana PK (2016) Particle swarm optimization for maximizing lifetime of wireless sensor networks. *Comput Electr Eng* 51:26–42
 20. Sharma S, Puthal D, Jena SK, Zomaya AY, Ranjan R (2017) Rendezvous based routing protocol for wireless sensor networks with mobile sink. *J Supercomput* 73(3):1168–1188
 21. Misra S, Krishna PV, Saritha V (2012) Lacav: an energy-efficient channel assignment mechanism for vehicular ad hoc networks. *J Supercomput* 62(3):1241–1262
 22. Karp B, Kung H-T (2000) Gpsr: Greedy perimeter stateless routing for wireless networks. In: *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp 243–254
 23. Li X, Huang J (2017) Abpp: An adaptive beacon scheme for geographic routing in fanet. In: *2017 18th International conference on parallel and distributed computing, applications and technologies (PDCAT)*. IEEE, pp 293–299
 24. Hua-Liang Z, Jun W, Hai-Bin YU, Peng Z (2011) An routing protocol for energy harvesting wireless sensor networks. *J Chin Comput Sys*
 25. Lin L, Sun Q, Wang S, Yang F (2012) A geographic mobility prediction routing protocol for ad hoc uav network
 26. Aboki R, Shaghaghi E, Akhlaghi P, Noor RM (2014) Predictive location aided routing in mobile ad hoc network. In: *2013 IEEE Malaysia International Conference on Communications (MICC)*
 27. Veerasamy A, Madane SSR (2014) An improved opportunistic location aided routing (iolar) in mobile ad hoc networks. In: *International Conference on Current Trends in Engineering and Technology*
 28. Liu WJ, Feng KT (2009) Greedy routing with anti-void traversal for wireless sensor networks. *IEEE Trans Mob Comput* 8(7):910–922
 29. Hsu MT, Lin YS, Chang YS, Juang TY (2009) Reliable greedy forwarding in obstacle-aware wireless sensor networks. In: *Algorithms & Architectures for Parallel Processing, International Conference, Ica3pp, Taipei, Taiwan*
 30. Zhu L, Li C, Li B, Wang X, Mao G (2015) Geographic routing in multilevel scenarios of vehicular ad hoc networks. *IEEE Trans Veh Technol*: 1–1

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.