


Research Article

A Vehicular Service-Oriented Dynamical Routing Searching Algorithm in Software-Defined BusNet

Lingqiu Zeng,^{1,2} Sheng Cheng,³ Qingwen Han ,^{3,4} and Lei Ye³

¹School of Computer Science, Chongqing University, Chongqing 400044, China

²Key Laboratory of Advanced Manufacture Technology for Automobile Parts (Chongqing University of Technology), Ministry of Education, Chongqing 400054, China

³School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China

⁴Research and Development Center of Transport Industry of Self-Driving Technology, Chongqing 400044, China

Correspondence should be addressed to Qingwen Han; hqw@cqu.edu.cn

Received 22 June 2020; Revised 17 August 2020; Accepted 10 September 2020; Published 22 September 2020

Academic Editor: Konglin Zhu

Copyright © 2020 Lingqiu Zeng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of the vehicular network, BusNet-based service is of great interest to both bus companies and public users. How to provide service and ensure service quality is considered as key points for the promotion of such kinds of applications. In this paper, a system architecture, which is designed based on multiaccess edge computing (MEC) architecture and software-defined network (SDN), is proposed. Moreover, considering the problem of the dynamicity of the network, a new routing path optimization algorithm, SPSO (serial particle swarm optimization), is proposed, while a dynamic weight matrix is used to improve routing performance. The comprehensive simulation results demonstrate that the proposed method can well adapt to the dynamic change of network topology and improve communication performance significantly.

1. Introduction

With recent advances in wireless communication technologies, vehicular ad hoc networks (VANETs) have received much attention in both industries and academia for enabling emerging Intelligent Transportation Systems (ITSs), such as autonomous intersection control [1], cooperative vehicular positioning [2], and advanced information sharing [3].

Besides basic safety message [4] that received considerable attention from the public, the abundant services that can improve user experience are gradually becoming a commercial hot spot too.

Buses, equipped with powerful communication and storage capability, are considered as a powerful service provider, which could provide commercial messages according to passenger's request. Hence, some bus companies believe that information service could bring benefits for public transport operators. Some previous research utilized buses as routing relays to improve network

connectivity [5, 6]. A few studies considered the bus ad hoc network (BusNet) as the delay/disruption-tolerant network (DTN), and different routing strategies based on the historical driving traces have been proposed [7]. The geocast in BusNet has also been widely investigated [8–10], where the mobility statistics from the historical traces are exploited for designing routing paths. However, none of the above studies have considered the delay-constrained routing in the BusNet.

Moreover, both academia and industry believe that a software-defined network (SDN) [11] can conveniently be applied to support the dynamic nature and dense deployment scenarios of future VANET functions and applications while reducing the operation cost [12].

However, the only problem is how to guarantee the quality of service (QoS). BusNet can provide information service via a routing path between information source and destination nodes. Therefore, the corresponding QoS problem could be transformed into a routing optimization

problem, which includes two key points, routing strategy and optimization algorithm.

Due to the dynamic nature of vehicle nodes, VANETs have highly dynamic topology. Researchers are more inclined towards adopting an opportunistic routing strategy [13], which utilizes a set of candidate nodes and discovers the shortest path according to index metric. Literature [14] employed three communication performance parameters—packet loss, delivery delay, and path accessibility—as the index for routing path selection but did not discuss the weight feature of the selected index. Hence, in this paper, we use the index metric of literature [14] and focus on the weight factor optimization.

The other left question is the optimization algorithm. Researchers believe that search algorithms, such as genetic algorithm [15], neighborhood search algorithm [16], and particle swarm optimization (PSO) [17], could be used in dynamic routing path selection. In PSO searching progress, each particle's moving direction is influenced by its inertness, local best-known position, and the group's best-known position. As the local best-known position is computed according to the historical data of this specific particle in the past few cycles, the group's best-known position is selected from the whole group historical data in the past few cycles. Based on this searching mechanism, PSO is able to solve the optimization problem lightly by calculating particle position and velocity, which represent the optimization result and iteration process, respectively. Since the different combinations of weights of VANET key factors can be regarded as the position of several particles, we believe that PSO strongly matches the demand of our application and will perform well in routing path discovery progress.

To deal with the time efficiency problem of PSO searching, in this paper, we proposed a new algorithm, serial particle swarm optimization (SPSO), to realize routing path optimization in SDN.

Given the above problems, the main contributions of this work are outlined as follows:

- (i) We propose a software-defined BusNet consisting of 4 layers: user layer, bus layer, roadside layer, and control layer.
- (ii) Based on the presented architecture, a data fusion scheme in the SDN framework is deployed.
- (iii) Two algorithms are employed to solve the routing selecting problem. Firstly, we use a modified Bellman-Ford algorithm to select the shortest path. Then, an intellectual algorithm called SPSO is applied to adapt to the network dynamicity while selecting the route.
- (iv) We conduct a simulation for these two algorithms to enhance the quality of service and discuss the relevant improvement of using the SPSO algorithm to reduce the packet delivery delay and packet loss.

The rest of this paper is organized as follows. We review the related work in Section 2. Section 3 describes the proposed system architecture. Section 4 formulates the problem

of route selecting and presents the calculation of factors we focused on. Routing selecting problem is proposed in Section 5. In Section 6, we propose the SPSO algorithm based on PSO to solve the optimization problem. In Section 7, simulation and evaluation of the results are given. The conclusion and future work are discussed in Section 8.

2. Related Work

In VANETs, a lot of effort has been devoted to the designing of routing protocols for the aim of enhancing the QoS. Eiza et al. [18] proposed a SAMQ (Situation Aware Multiconstrained QoS) routing algorithm to compute feasible routes and select the best route subject to multiple QoS constraints. The SA levels and ACS (Ant Colony System) mechanism are utilized at intermediate nodes to respond to the link breakage without interruption, and they not only replace the broken link but also consider the QoS requirements of data transmission. Togou et al. [19] developed a distributed geographic source routing scheme named SCRPs (Stable CDS-based Routing Protocol), which built the stable backbones over the road segment by considering vehicles' speed and spatial distribution and connected them through the dynamically elected bridge nodes at intersections. Nevertheless, their research ignores the constriction of communication bandwidth and the dynamicity of the network.

Besides, routing design with buses based on SDN in the city network has also been excavated to solve the problem of the lack of calculation capacity and space sparsity of vehicle distribution. Wang et al. [14] designed a BusNet, using buses as service providers, to forward response packets and introduce SDN into BusNet, believing that the SDN controller is fitted for the role of routing path decision-maker. They employed three communication performance parameters—packet loss, delivery delay, and path accessibility—as the index for routing path selection, but did not discuss the weight feature of the selected index. Although the architecture proved to be a success, the algorithm to find a feasible route did not fit the scenario of high dynamical characteristics. Jiang and Du [20] designed a VANET architecture that integrated buses, RSUs, and TCC (Traffic Control Center) for providing data services. Each vehicle needs to register with a nearby high-tier node (i.e., RSU or bus), which is connected to TCC. The proposed registration technology reduces the number of switches in the routing path by utilizing the updated registration message to locate destination vehicles and selecting the proper registration nodes as relays. This research focused on the bandwidth, yet it failed to take the dynamicity into consideration, which is the core problem of VANET.

Furthermore, many research works have studied the application of intellectual algorithm in routing strategies. Shirmarz and Ghaffari [21] propose a greedy flow routing algorithm on SDN to enhance the QoS. They consider a series of network variables including connectivity, bandwidth, delay, jitter, and packet loss of links matrices. After selecting all paths between source and destination nodes,

they use the greedy algorithm to try to select the best path in every step. Oh et al. [22] propose the RT-SDN for priority order and adaptive routing that can guarantee the required bandwidth. Zhao et al. [23] propose a joint optimization of flow entry aggregation and routing selection in SDN, but this solution only focuses on delay. Fei et al. [24] work on virtual network function (VNF) to provide flexible services, and they use an efficient online learning method to predict the service chain demand and propose two algorithms for new instance assignment and service chain rerouting. Feng et al. [25] present the design of distributed online solutions that drive flow processing and routing decisions, along with the associated allocation of cloud and network resources. In [26], the authors' goal is to determine the placement of VNFs, the routing of service flows, and the associated allocation of cloud and network resources that satisfy client demands with minimum cost. Bagci and Tekalp [27] propose a heuristic group-constrained SP procedure that aims for a fair allocation of resources among a group of requests with the same service level while maximizing the total NSP (Network Service Provider) revenue. All the above studies have done a good job on the route decision; however, the problem is that their solution cannot be applied to a moving model especially when the cars or the users of which own huge mobility.

Distinguished from the above studies, our work focuses on the dynamicity of BusNet and addresses the problem of routing selecting, thereby enhancing the quality of service. We propose an architecture that can conveniently compute the attributes of each link and then construct a network graph at the controller. In addition, the architecture can save considerable bandwidth and use buses to solve the space sparsity problem. What is more, based on the graph on the controller, an intellectual algorithm, SPSO, is proposed to handle the problem of dynamicity. As a result, the users are able to get a high-quality service according to the suitable communication route.

3. System Architecture

In this section, we shall introduce a basic system architecture that provides BusNet-based service. Meanwhile, a corresponding message dissemination progress should be explained.

3.1. System Framework. Here, we employ a BusNet-oriented SDN architecture, which includes four layers—namely, user layer, bus layer, roadside layer, and controller layer—to provide information service, as shown in Figure 1.

- (1) User layer: the user layer includes social vehicles equipped with LTE-V2X OBUs. Each vehicle could be considered as a user, who has information service requirements.
- (2) Bus layer: the buses act as information collectors and service providers. Here, we assume that all buses are equipped with LTE-V2X OBUs and have sufficient storage room to store the collected information.

- (3) Roadside layer: it includes a roadside unit (RSU) and the cellular base station (BS) and performs information aggregation and redundant information filtering.
- (4) Controller layer: controller acts as a global manager, which executes routing path optimization procedure.

3.2. Message Dissemination Progress. The whole service providing process includes two steps of message dissemination progress; one is service information collection, while the other is service request/response.

3.2.1. Service Information Collection. Sequence diagram of service information collection is shown in Figure 2.

As service subscribers, cars are not involved in this procedure. Bus node generates the Neighborhood Message (NMSG) packet, which includes state information of the bus and its neighboring bus, such as Bus ID, cruising velocity, heading, and GPS position, and broadcasts NMSGs periodically. Meanwhile, the bus node collects service information, such as discount posters, shopping mall opening, 24-hour store position, etc. RSU/BS collects NMSGs, filters redundant information, and generates a message packet NMSG_AggreTag, which is uploaded to the controller. Note that, here, RSU/BS can be geographically organized in several service areas, while each one is governed by an SDN controller. According to NMSG_AggreTag, the SDN controller generates a service map, which includes network status and GPS information of bus nodes.

3.2.2. Service Request/Response. Sequence diagram of request/response is shown in Figure 3.

If a vehicle user has a service requirement, it should send a request packet to the controller. According to the request packet, the controller searches for an information source and will employ a routing path optimization algorithm to select the most suitable routes from the bus route set. Then the controller generates a Link Tag message and sends it to bus nodes. The source bus node sends service information to the target bus node, which provides service to the vehicle user. At last, the vehicle user node should feed back corresponding performance features, such as service delay and packet loss ratio, to the controller.

3.2.3. Data Filtering. To improve the efficiency of BusNet service, a data filtering process, whose pseudocode is shown in Algorithm 1, is done by RSUs/BSSs. Corresponding variables and functions definitions are given in Table 1.

4. Problem Formulation

This section is to abstract the problem in reality into mathematical problems. We will give the problem statement and explain the selected index in detail.

The following assumptions are made:

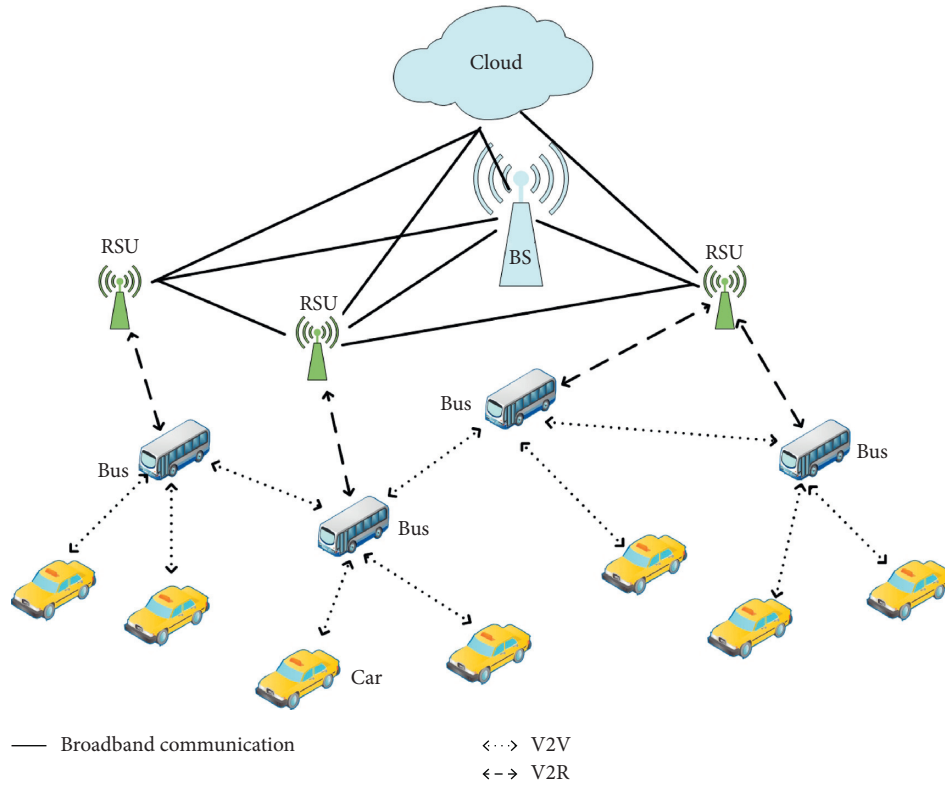


FIGURE 1: Architecture of the software-defined network.

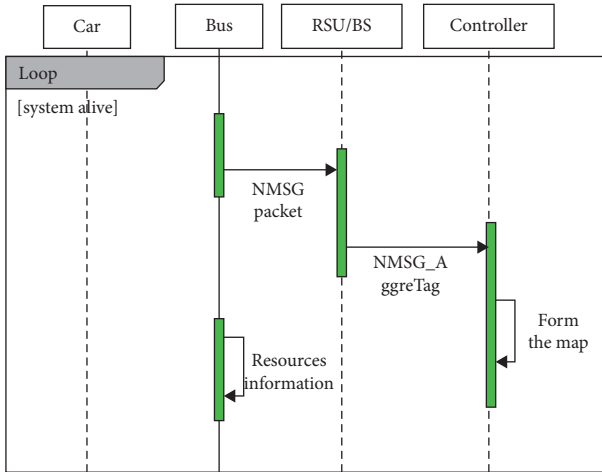


FIGURE 2: Sequence diagram of service information collection.

- (1) BusNet fully covers the object area, which is divided into several nonoverlapping service areas
- (2) Each service area is covered by RSUs/BSs placed in fixed locations
- (3) Each service area can be controlled by one or several SDN controllers
- (4) User nodes and bus nodes broadcast message periodically via LTE-V2X OBU
- (5) The SDN controller could be implemented as 1-to- N associated with vehicle node via RSUs/BSs relay

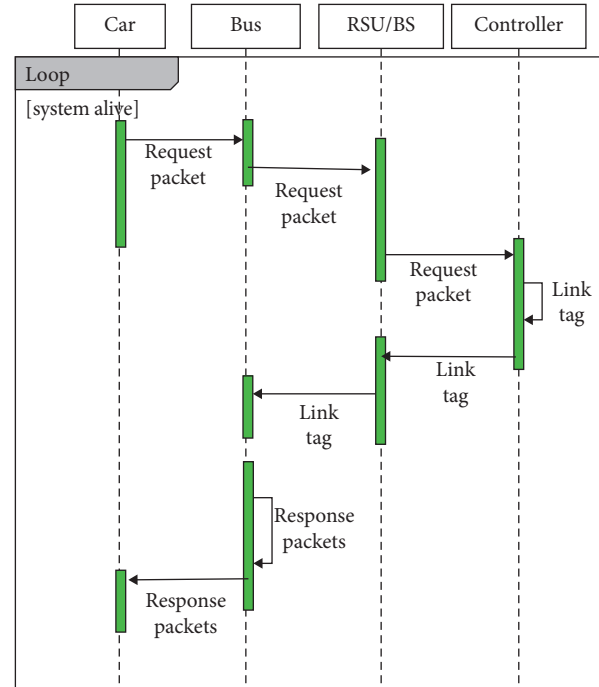


FIGURE 3: Sequence diagram of request/response.

BusNet is abstracted as a directed graph $G(B, E)$, where B and E are the sets of bus nodes and edges. We assume that there are n paths between information source bus node and service subscriber bus node; then, the set of all candidate

```

input: thisMessage, lastMessages
output: messages
(1) newMsgFlag  $\leftarrow$  true
(2) For  $i=0 \rightarrow$  lastMessages.size ()-1 do
(3)   if lastMessages [i].srcId = thisMessage.srcId then
(4)     newMsgFlag  $\leftarrow$  false
(5)     if allSame (lastMessages [i], thisMessage) then
(6)       throwMessage (thisMessage)
(7)     else
(8)       updateMessage (thisMessage)
(9)       newMsgFlag  $\leftarrow$  true
(10)    end if
(11)  end if
(12) end for
(13) if newMsgFlag = true then
(14)  addMessage (thisMessage)
(15) end if
(16) return lastMessages

```

ALGORITHM 1: Data filtering.

TABLE 1: Explanation of variables and functions in data filtering.

Variable/function	Explanation
this Message	Message of current searching period.
last Messages	Messages of last searching period.
new Msg Flag	A flag message expresses that current message is sent by a new bus node.
all Same (p_1, p_2)	Content of packet p_1 is the same as that of packet p_2 .
throw Message (p)	Discard current message.
update Message (p)	Update message p in database.
add Message (p)	Add message p to database.

routing paths can be denoted by $P = \{p_1, p_2, \dots, p_n\}$. As mentioned earlier, in this paper, we employ three communication performance factors—packet loss, delivery delay, and path accessibility—as the index for routing path selection. Assuming that a routing path includes m edges, we defined routing path comprehensive training evaluation index as follows:

$$C_{pi} = \sum_{k=1}^m C_{pi}^k, \quad (1)$$

where C_{pi} is the comprehensive training evaluation index of routing path i and k represents the k th pair of bus nodes of this path. C_{pi}^k is the cost of every pair of bus nodes and is defined as follows:

$$C_{pi}^k = w_p \times \text{Ploss}_i^k + w_d \times \text{delay}_i^k + w_a \times \text{acc}_i^k, \quad (2)$$

where Ploss_i^k , delay_i^k , and acc_i^k are packet loss, delivery delay, and path accessibility of routing path i for k th bus nodes pair. w_p , w_d , and w_a are corresponding weight factors of all the above three factors, and the combination of which implies different focus on network attributes.

Then those three factors can be denoted as

$$\begin{aligned} \text{pLoss}_i^k &= 2 \times g \left(1 - \frac{\text{NR}_i^k}{\text{NS}_i^k} \right) - 1, \\ \text{delay}_i^k &= 2 \times g \left(\frac{\sum_{j=1}^{\text{NR}} \text{delay}_i^k(j)}{\text{NR}_i^k} \right) - 1, \\ \text{acc}_i^k &= 2 \times [1 - g(L_i^k)], \end{aligned} \quad (3)$$

where

$$\begin{aligned} L_E^2 &= [(x_{Bs} - x_{Be}) - (v_{Bs} \sin \theta_{Bs} - v_{Be} \sin \theta_{Be})]^2 \\ &\quad + [(y_{Bs} - y_{Be}) - (v_{Bs} \cos \theta_{Bs} - v_{Be} \cos \theta_{Be})]^2, \\ g(z) &= \left(\frac{1}{1 + e^{-z}} \right). \end{aligned} \quad (4)$$

For one edge with two bus nodes, NS is the sent packet number, and NR is the received packet number. Delay(j) is the dissemination delay of a specific packet, and L is the length of this edge after one second. Bs and Be are the start and end bus nodes of this edge, and x_{Bs} , y_{Bs} , v_{Bs} , and θ_{Bs} represent the horizontal location, vertical location, velocity, and moving direction of Bs, respectively. Then, the set of routing path comprehensive training evaluations' index is denoted by $C_p = \{C_{p1}, C_{p2}, \dots, C_{pn}\}$, and routing selecting problem is formulated as follows:

$$\min_{p_i \in P} (C_{pi}). \quad (5)$$

On the other hand, after selecting a route through training evaluations, we need to use another index called generalization evaluation to evaluate the performance of this specific path in real communication.

We define generalization evaluation as follows:

$$G_{pi} = \left(\frac{\text{loss}_i + \text{delay}_i}{2} \right), \quad (6)$$

here, G_{pi} represents the generalization evaluation, and loss_i and delay_i represent the statistical result of communication loss and delay, respectively.

Then, the optimization algorithm could find an appropriate combination of weight factors, which can help to select an appropriate path with a minimal G_{pi} .

5. Routing Selecting Problem Solution through Modified Bellman–Ford Algorithm

In this section, we propose a modified Bellman–Ford algorithm to select the path with minimal training evaluation. Given a precise request, we can construct the routing graph with specific start bus nodes and destination nodes.

When receiving a request, the controller can construct a routing graph, and, after analyzing the request, it will recognize the request bus node Be and the resource holder bus

node Bs. To find the shortest route, the controller has to compute the length of each edge, C_p^k . Recall that the nodes pair of this edge has three parameters: delay, packet loss, and path accessibility. Then, the length of an edge can be calculated via formula (1) after choosing a specific combination of weight of these three factors.

The steps of the modified Bellman–Ford algorithm are described as follows:

- (1) Initialize: find the id s and e of the resource holder node and resource subscriber node ($s, e \in [1, m]$). Set $D[i] = -1$, where $D[i]$ represents the distance of B_s and B_i . Calculate each $C_p^k \in C_p$ according to formula (1) and the weight matrix of those three factors including packet loss, delivery delay, and path accessibility of routing path, but notice that the following constraints need to be satisfied: (a) Delay^k is smaller than TD (threshold of delay); (b) Ploss^k is smaller than TP (threshold of packet delivery loss); (c) acc^k is smaller than TA (threshold of path accessibility).
- (2) Optimize: traverse each edge, and set $D[i + 1] = \min\{D[i + 1], D[i] + C_p^i\}$.
- (3) Repeat step 2 $N - 1$ times, where N is the number of bus nodes.
- (4) After finding the candidate path, check if the total delay of this route is smaller than the request live time. This should guarantee that the selected path is valid for the specific request.

6. Optimization Problem Solution Based on SPSO Algorithm

6.1. SPSO Algorithm. Particle swarm optimization (PSO) is a parallel population-based computation technique, which was motivated by the organism's behavior such as schooling of fish and flocking of birds. In general, the PSO based route searching method gets optimal weights of the routing path matrix through multiple iterations and could be used to generate a quasi-static route. However, in BusNet, current routing selection is not exactly the same as the former times. That said, the selected routing path is nonpersistent, and it is changing with the network communication status or with time. Then, for a new service request, a lot of time has to be wasted to finish another round of training. As a result, PSO's iteration optimization method could not be used directly in BusNet.

However, the optimal routing path between the source bus node and destination bus node is relatively stable. Then, in this paper, a serial PSO algorithm (SPSO), which finds an optimal routing path according to recent searching results, incorporating the training process into every service request, is proposed. To be clearer, failing to use the past training result, PSO does several rounds of training for one request to find the best combination of weights. On the contrary, SPSO, taking advantage of the former training result, does training only once for each request. We can say that the request-response process is exactly our training process. A set of particles, each of which is denoted by a variable combination of weights, is used for training to avoid trapping in a local optimum.

Here, we assume that each routing path searching process is triggered by a request message sent by a user node. A particle vector is defined as

$$V = [V_1, V_2, \dots, V_k],$$

$$(V_i = [W_{pi}, W_{di}, W_{ai}]), \quad (7)$$

where k is the particle's number. V_i is the position of the i th particle, representing the weight matrix, our optimization targets. The moving trend of particles is denoted as

$$\Delta\alpha_i^t = 0.3 \times \Delta\alpha_i^{t-1} + 0.3 \times (p\text{Best} - V_i^{t-1}) + 0.3 \times (g\text{Best} - V_i^{t-1}), \quad (8)$$

where $\Delta\alpha_i$ is the moving direction of particle i , and t is the time moment of current searching progress, while $t - 1$ is that of last searching progress. $p\text{Best}_i$ is the best position of particle i during last N searching progress. $g\text{Best}$ is the best position of all particles during last N searching progress. The position updating process is expressed as

$$V_i^t = V_i^{t-1} + \Delta\alpha_i^t. \quad (9)$$

The process of implementing the SPSO is described in Algorithm 2, while corresponding variables and functions definitions are explained in Table 2.

6.2. Algorithm Complexity Analysis. The number of particles, K , will affect the update process of each particle, and the number of bus nodes and edges, N and E , will affect the shortest path algorithm complexity. Generally, the most time-consuming part should be the iterating process of the optimization algorithm. Nevertheless, the SPSO algorithm puts iteration into every request, which guarantees that the particles will update only once while the algorithm runs. In other words, the complexity of the iterative process is $O(K)$. As a result, the most time-consuming task is the shortest path algorithm, and the complexity is $O(NE)$. Then, the complexity of the SPSO algorithm is $O(NE + K)$. Considering that the number of particles is usually far less than the number of bus nodes, we can regard the complexity as $O(NE)$.

7. Simulation Results Analysis

All simulations are done based on the NS3 platform, a professional communication simulation platform. The performance of the proposed SPSO algorithm is evaluated by delivery delay and packet loss:

- (1) Delivery delay: average time interval at which a service message is sent from the source bus node to the user node
- (2) Packet loss: number of packets received incorrectly from source bus node to the total number of data packets

7.1. Simulation Setup. Here, we define a simulation scenario, whose area is 10 km^2 , and assume that bus nodes are distributed uniformly in the objective area. Note that, here,

```

input: request, velocity, position
output: path
(1) if velocity == null || position == null then
(2)   initializeParticle ()
(3) end if
(4)  $p [i] \leftarrow$  getPresentParticle ()
(5) path  $\leftarrow$  getMinTrainingEvaluationPath (request,  $p [i]$ )
(6) [ $G_{pi}$  loss delay]  $\leftarrow$  getGeneralizationEvaluation (path)
(7) update_pBest ( $G_{pi}$ )
(8) update_gBest ( $G_{pi}$ )
(9) for  $i = 0 \rightarrow k - 1$  do
(10)   velocity [ $i$ ]  $\leftarrow$  updateVelocity ( $p [i]$ )
(11)   position [ $i$ ]  $\leftarrow$  updatePosition ( $p [i]$ )
(12) end for
(13) return path

```

ALGORITHM 2: SPSO.

TABLE 2: Explanation of variables and functions in SPSO algorithm.

Variable/function	Explanation
request	Request message includes the node id of request node and request content.
velocity	Particle's velocity set.
position	Particle's position set, which corresponds to weight matrix.
path	Generated routing path of specific request.
initialize Particle	Particle initialization.
get Present Particle	Get particle of current request.
get Min Training Evaluation Path (request, $p [i]$)	Select the path with minimal training evaluation based on this request and current weights of index.
get Generalization Evaluation	Get routing path generalization evaluations index.
update_pBest (G_{pi})	Update p Best according to current evaluation results.
update_gBest (G_{pi})	Update g Best according to current evaluation results.
update Velocity ($p [i]$)	Update particle velocity according to (5).
update Position ($p [i]$)	Update particle position according to (6).

RSU and BS play the same role in BusNet; hence, our simulation scenario includes only RSU, and then we assume the following:

- (1) RSUs communicate with vehicle nodes (including bus node and user node) via LTE-V2X
- (2) RSUs communicate with the SDN controller via Ethernet

Detail parameters are listed in Table 3.

7.2. Simulation Results. Simulation results are shown in Figures 4 and 5.

As shown in Figures 4 and 5, both delivery delay and packet loss increase with the increase of bus node number. Compared with literature's routing algorithm RDH, which simply sets the weight of three factors as equal, the proposed SPSO algorithm can reduce both factors significantly. The reason is that SPSO owns the ability to judge the combination of the weights and then adapt them to the current network status. Specially, we can notice that the more the number of bus nodes is, the more obvious the performance

improvement will be. That is because as the requests get more frequent, the controller is more likely to learn more about the status in the current communication environment and can modify the combination of the weights timely, thereby performing better. To be much clearer, SPSO is a serial algorithm which means that it can only handle the requests one by one, and that is why we put 10 particles here. When the requests pour in, each particle can iterate several times in a short time. As a result, the combinations of weights are more likely to catch up with the change of network status.

The optimization progress of the case with 240 bus nodes is shown in Figures 6 and 7.

As mentioned earlier, the SPSO algorithm fulfills routing path optimization based on recent searching results. The result is coming from the requests from all the cars or users. Since the algorithm focuses on the global network status, it is useless to follow a specific user and analyze it. According to Figures 6 and 7, we can derive two implications:

- (1) When the number of searching times is less than 30, both delivery delay and packet loss curve of SPSO coincide with that of the routing algorithm in [14].

TABLE 3: Parameters.

Variable/Function	Explanation
Bus number	60, 120, 240
Bus direction	0–360°
RSU number	3, 6, 12
Controller number	1
Cars number	300, 400, 600
V2V communication range	200 m
Service area	10 km ²
Bus distribution	Uniform distribution
Velocity of buses	5–10 m/s
Request packets lifetime	3 s
Simulation time	80 s
Scheduling period	1 s
Particle number	10
Relaynodes number	[3, 15]

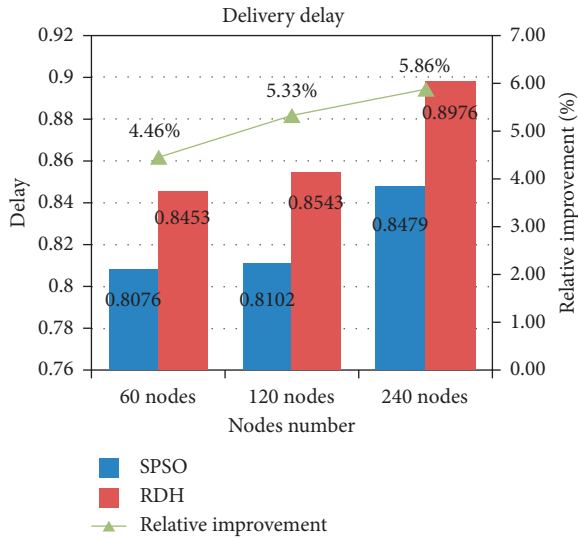


FIGURE 4: Delivery delay.

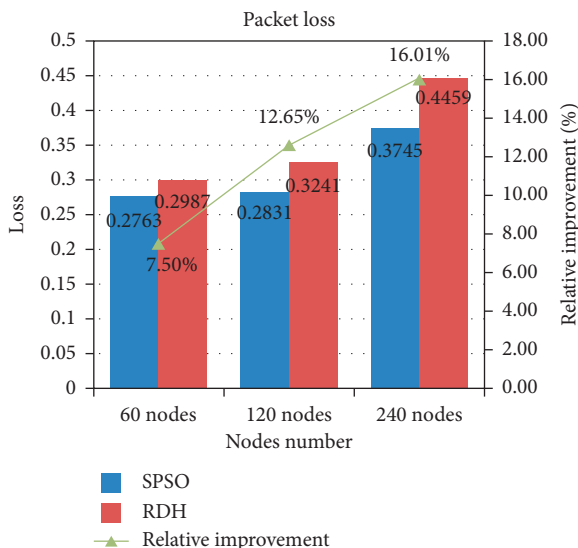


FIGURE 5: Packet loss.

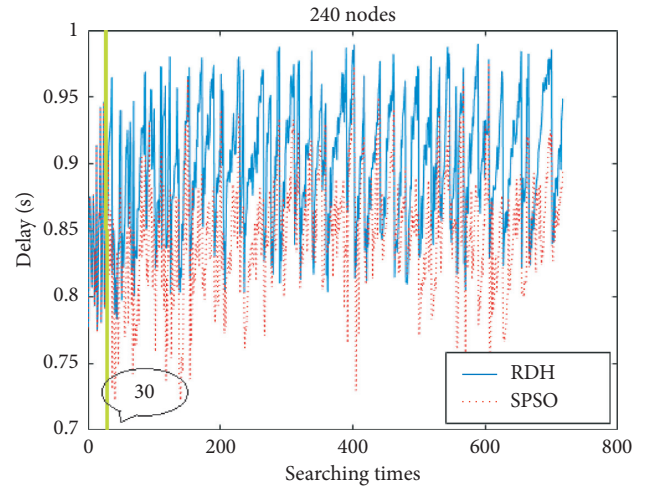


FIGURE 6: Delivery delay.

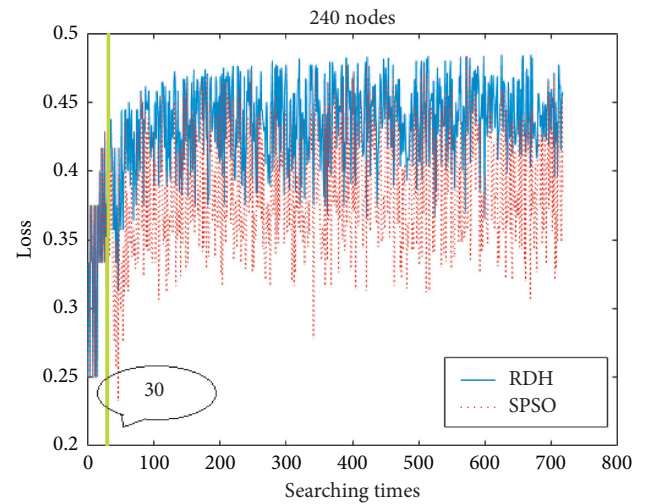


FIGURE 7: Packet loss.

That said, the sample size of recent search records is not enough to support the optimization process and the weights factors remain the same as the initial values.

- (2) When the number of searching times is large than 30, the delivery delay and packet loss values of SPOS are obviously lower than that of the routing algorithm in [14]. That being said, the positions of the particles, also named the combinations of weights, are dynamically and persistently changed with network status.

Hence, we believe that the proposed SPSO algorithm will converge fast and find optimal weight factors, thus improving user experience.

8. Conclusion and Future Work

There is no doubt that routing efficiency is an important topic for most of the wireless networks. As mentioned in Section 3, the particularity of BusNet brings a great challenge

to realize routing optimization. Most of mesh network-oriented routing algorithms cannot be applied to BusNet due to its dynamic feature. In this paper, we define a routing path evaluation factor, which employs a weight matrix to adapt to the dynamic change of the network. On the other hand, to meet the service demand and solve the information dissemination problems, such as poor timeliness and low success rate, we propose a serial PSO algorithm, which finds optimal routing path according to recent searching results. Simulation results show that the proposed method could enhance system performance and communication efficiency.

However, this paper just considers the case with only one SDN controller. Hence, in future work, we shall work on multiple-controller case. Moreover, in this paper, the user vehicle number is set as a fixed value implying that no car leaves or enters. In future work, we shall discuss the impact of the changing of the number of user vehicles.

Data Availability

The data used to support the findings of this study have been deposited in the GitHub repository (<https://github.com/qxcs/Data-for-An-Intellectual-Routing-Algorithm-based-on-SDN.git>).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the Key Laboratory of Advanced Manufacture Technology for Automobile Parts (Chongqing University of Technology), Ministry of Education, under Grant no. 2017KLMT04, and the Key Research and Development Projects of Chongqing Special Industries for Technological Innovation and Application Demonstration under Grant no. cstc2018jszx-cyzdX0064.

References

- [1] P. Dai, K. Liu, Q. Zhuge, E. H.-M. Sha, V. C. S. Lee, and S. H. Son, "Quality-of-Experience-Oriented autonomous intersection control in vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1956–1967, 2016.
- [2] K. Liu, H. B. Lim, E. Frazzoli, H. Ji, and V. C. S. Lee, "Improving positioning accuracy using GPS pseudorange measurements for cooperative vehicular localization," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 6, pp. 2544–2556, 2014.
- [3] J. Wang, K. Liu, K. Xiao et al., "Dynamic clustering and cooperative scheduling for vehicle-to-vehicle communication in bidirectional road scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1913–1924, 2018.
- [4] M. Zhang, G. G. M. N. Ali, P. H. J. Chong, B.-C. Seet, and A. Kumar, "A novel hybrid MAC protocol for basic safety message broadcasting in vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–14, 2019.
- [5] J. Alves Junior and E. C. G. Wille, "P-AOMDV: An improved routing protocol for V2V communication based on public transport backbones," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 12, pp. 1653–1663, 2016.
- [6] A. B. Makar, K. E. McMartin, M. Palese, and T. R. Tephly, "Formate assay in body fluids: application in methanol poisoning," *Biochemical Medicine*, vol. 13, no. 2, pp. 117–126, 1975.
- [7] K. Chen and H. Shen, "DTN-FLOW: inter-landmark data flow for high-throughput routing in DTNs," *IEEE/ACM Transactions on Networking*, vol. 23, no. 1, pp. 212–226, 2014.
- [8] F. Zhang, H. Liu, Y.-W. Leung, X. Chu, and B. Jin, "CBS: Community-based bus system as routing backbone for vehicular ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2132–2146, 2016.
- [9] W.-H. Lin, H. Liu, and H. K. Lo, "Guest editorial: big data for driver, vehicle, and system control in ITS," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1663–1665, 2016.
- [10] F. Zhang, B. Jin, Z. Wang, H. Liu, J. Hu, and L. Zhang, "On geocasting over urban bus-based networks by mining trajectories," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1734–1747, 2016.
- [11] M. Abolhasan, J. Lipman, W. Ni, and B. Hagelstein, "Software-defined wireless networking: centralized, distributed, or hybrid?" *IEEE Network*, vol. 29, no. 4, pp. 32–38, 2015.
- [12] A. A. Khan, M. Abolhasan, and W. Ni, "5G next generation VANETs using SDN and fog computing framework," in *Proceedings of the 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6, Las Vegas, NV, USA, January 2018.
- [13] X. Zhang, X. Cao, L. Yan, and D. K. Sung, "A street-centric opportunistic routing protocol based on link correlation for urban VANETs," *IEEE Transactions on Mobile Computing*, vol. 15, no. 7, pp. 1586–1599, 2015.
- [14] J. Wang, K. Liu, K. Xiao, X. Wang, Q. Han, and V. C. S. Lee, "Delay-constrained routing via heterogeneous vehicular communications in software defined BusNet," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5957–5970, 2019.
- [15] C. W. Ahn and R. S. Ramakrishna, "A genetic algorithm for shortest path routing problem and the sizing of populations," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 6, pp. 566–579, 2002.
- [16] Y. Marinakis, A. Migdalas, and A. Sifaleras, "A hybrid particle swarm optimization - variable neighborhood search algorithm for constrained shortest path problems," *European Journal of Operational Research*, vol. 261, no. 3, pp. 819–834, 2017.
- [17] Q. Lin, S. Liu, Q. Zhu et al., "Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 32–46, 2016.
- [18] M. H. Eiza, T. Owens, Q. Ni, and Q. Shi, "Situation-aware QoS routing algorithm for vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5520–5535, 2015.
- [19] M. A. Togou, A. Hafid, and L. Khoukhi, "SCRIP: stable CDS-based routing protocol for urban vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1298–1307, 2016.
- [20] X. Jiang and D. H. Du, "Bus-vanet: a bus vehicular network integrated with traffic infrastructure," *IEEE Intelligent*

- Transportation Systems Magazine*, vol. 7, no. 2, pp. 47–57, 2015.
- [21] A. Shirmarz and A. Ghaffari, “An adaptive greedy flow routing algorithm for performance improvement in software-defined network,” *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 33, no. 1, Article ID e2676, 2020.
 - [22] S. Oh, J. Lee, K. Lee, and I. Shin, *RT-SDN: Adaptive Routing and Priority Ordering for Software-Defined Real-Time Networking*, KAIST, Daejeon, South Korea, 2016.
 - [23] Z. Zhao, B. Wu, J. Xiao, and Z. Hu, *Joint Optimization of Flow Entry Aggregation and Routing Selection in Software Defined Wireless Access Networks*, Springer, Berlin, Germany, 2018.
 - [24] X. Fei, F. Liu, H. Xu, and H. Jin, “Adaptive VNF scaling and flow routing with proactive demand prediction,” in *Proceedings of the IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, Honolulu, HI, USA, April 2018.
 - [25] H. Feng, J. Llorca, A. M. Tulino, and A. F. Molisch, “Optimal dynamic cloud network control,” *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2118–2131, 2018.
 - [26] H. Feng, J. Llorca, A. M. Tulino, D. Raz, and A. F. Molisch, “Approximation algorithms for the NFV service distribution problem,” in *Proceedings of the IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, Atlanta, GA, USA, May 2017.
 - [27] K. T. Bagci and A. M. Tekalp, “Dynamic resource allocation by batch optimization for value-added video services over SDN,” *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3084–3096, 2018.