Hindawi
Mobile Information Systems
Volume 2022, Article ID 4988266, 13 pages
https://doi.org/10.1155/2022/4988266

Research Article

DIMDP: A Driving Intention-Based MDP Service Migration Model under MEC/MSCN Architecture

Lei Ye,[1] Kaiwen Ling,[1] Qingwen Han,[1] Yufei Yan,[1] Lingqiu Zeng,[2] Lingfeng Qi,[1] Li Lin,[1] and Junjun Zhang[2]

[1] *School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China*
[2] *College of Computer Science, Chongqing University, Chongqing 400044, China*

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

Received 15 July 2022; Accepted 26 September 2022; Published 11 October 2022

Academic Editor: Konglin Zhu

Service migration is one of the key topics of Internet of Vehicles (IoV). MEC (mobile edge computing), which is carried on the roadside unit (RSU), could serve as a service provider and provide a V2I (vehicle to infrastructure) cooperation service. To solve the high migration rate caused by the vehicle's high mobility feature, MSCN (mobile secondary computing node) framework is defined. To study service migration features under the MSCN framework further, in this paper, road vehicle's motion features, e.g., driving intention and regional traffic condition, are introduced to construct the Markov decision model, which is used to explain the service migration decision procedure, while DIMDP (driving intention-based MDP) is defined. Corresponding cost functions are defined, while the optimal object is given. A two-way road scenario is selected as a typical scenario. NS3 platform is employed to fulfill the simulation process. Simulation results show that the proposed service migration strategy performs well and is intensive to vehicle density change.

1. Introduction

With the rapid growth of vehicles, road safety has become a problem faced by countries all over the world. Vehicle safety applications by means of real-time information exchange between vehicles are techniques for avoiding traffic accidents.

Although cooperative vehicle infrastructure system (CVIS) is considered an effective approach to support vehicle safety applications, it has to meet the requirements of real-time ability, reliability, and service continuity.

For non-local service requests, the traditional centralized cloud computing network has defects in terms of delay and throughput. Therefore, in order to match the needs of real-time driving decision making, scholars have introduced edge computing technologies such as mobile edge computing (MEC) and fog computing (FC) into CVIS. Also, the edge computing server is configured on the side close to the user, deployed at the end of the base station (BS) and roadside unit (RSU). The back-end service information can be calculated and maintained at the edge node, thus providing relatively adequate computing ability [1] and supporting computational-intensive and time-sensitive operations or flexible deployment of applications and services. But for non-local services, MEC needs to maintain a certain service migration rate to reduce service delay. However, the high mobility of vehicle nodes is a negative impact on the efficiency of feedback information and service migration rate. The reason is that the object vehicle may leave the coverage area of the previous MEC, which undertakes computing tasks. To overcome this problem, in our previous work [2], a new concept, mobile secondary computing nodes (MSCNs), was defined, while a three-layer framework is constructed.

Obviously, the relative speed of MSCN and its surrounding vehicles is lower than that of RSU and its surrounding vehicles and partially solves the service migration problem. However, the corresponding migration mechanism is different from that of traditional MEC architecture. Hence, we discussed the service migration problem under MSCN architecture [3] and proposed a MDP (Markov decision process)-based service migration strategy.

However, in [3], the service migration strategy was designed according to the relative distance between vehicles and did not consider the vehicle's moving feature and motion model. Obviously, service migration decision making is influenced by the traveling trace of the vehicle, which is the reflection of driving intention and is affected by regional vehicle density. Since the vehicle prefers to go to places with good traffic conditions, we can estimate the vehicle's intention by considering the vehicle's moving feature and motion model.

Furthermore, the MDP model is considered an effective approach to express service migration progress. By establishing a state space that conforms to the vehicle's moving feature and combining the motion model, a trade-off is made between the cost of migration and the benefits after migration to obtain the optimal service migration strategy.

Therefore, in this paper, driving intention-based MDP service migration strategy under MSCN architecture is proposed. Specifically, our research contributions are as follows:

According to vehicle driving intention and regional traffic condition factors over the dynamic feature of service requirement, a MDP-based service migration method, DIMDP (driving intention-based MDP), is proposed.

A two-way road scenario, which includes four microscopic events, namely, following, turn left, turn right, and U-turn, is selected as the typical scenario.

The rest of the paper is structured as follows. Section 2 presents the related work, and Section 3 discusses the service migration using MDP based on MSCN. The service migration algorithm under MSCN is proposed and applied to traffic for service migration as an example in Section 4. Then, experimental simulation and result analysis are presented in Section 5. Finally, the conclusion is given in Section 6.

2. Related Work

We briefly investigated the existing literature on service migration from the perspective of service delay and trajectory prediction.

Yu et al. [4] prioritized MEC services and proposed a partial dynamic optimization algorithm (PDOA) service migration strategy calculation algorithm, which calculates the migration strategy that minimizes the average delay of long-term service by predicting the mobility of vehicles and achieves good results when the density of vehicles is small. However, the algorithm does not consider the impact of service interruption caused by the long migration time when the vehicle density is high.

Ge et al. [5] proposed the FEE algorithm to calculate the best service migration strategy, which takes into account the current and future time delays in N time slots and the service migration interface between vehicles and changes the focus of the migration strategy by setting up weight coefficients for the delay of each time slot. The algorithm can achieve better results by increasing the number of time slots, which is not obvious when the number of time slots is 5. However, the algorithm does not consider the frequency of service requests, so the algorithm may not achieve better results when services require frequent requests.

Nadembega et al. [6] proposed a mobility-based service migration prediction (MSMP) model, which split user requested service into several portions for service migration by estimating the throughput that the user could receive in advance.

Xu et al. [7] investigated path selection for seamless service migration and proposed a path-selection algorithm to jointly optimize both interests of the network plane and service plane and designed a distance-based filter strategy to eliminate undesired switches in advance to improve the scalability of the proposed algorithm.

Recently, the MDP model is considered an effective approach to express service migration progress. Corresponding researchers considered both user mobility features and MEC-based service requirements to establish decision-making strategies [8]. Moreover, cost function definition methods are proposed [9]. Combined with MDP and cost function definition. Taleb et al. [10] proposed a service migration decision algorithm to verify the effectiveness of Follow Me Cloud and provided guidelines for the method in this paper.

However, the existing MDP-based research has not built a model suitable for the CVIS, which considers the vehicle's moving feature and motion model.

In view of traffic simulation, vehicle motion models are categorized into microscopic [11] and macroscopic [12]. Microscopic models focus on the dynamic features of the separate vehicles, while macroscopic models consider traffic features, such as the traffic density, average velocity, flow, and so on. Hence, in this paper, we introduce density factor into the vehicle intention prediction model, thus constructing a microscopic motion model. On the other hand, to establish the MDP model, we borrow from the idea of the cellular model [13], which has adopted the time-space discretion method, to construct a microscopic motion model.

3. Service Migration under MSCN Framework

According to the percentage of migrated content, service migration can be divided into partial migration and full migration. Different migration methods bring different advantages and disadvantages. Since partial migration involves real-time decisions about which part of the data to migrate, this paper only considers the overall migration of the service.

3.1. MSCN Framework. The MEC-based architecture, MSCN, proposed in our previous work is shown in Figure 1.

Here MSCN framework includes three layers, which are MEC layer, MSCN layer, and general vehicle layer. Experimental verification shows that MSCN can provide reliable RSU-oriented services and significantly improves both communication performance and computing efficiency [2].

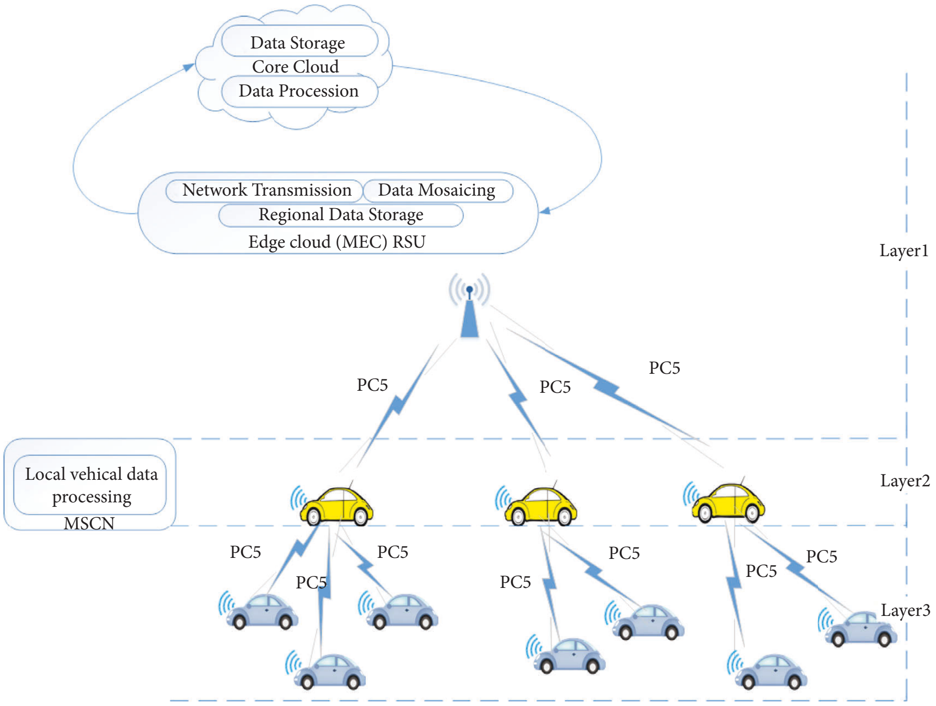


Figure 1: MSCN framework.

3.1.1. MEC Layer. The MEC layer provides computing and storage resource to road vehicle. Meanwhile, the service provider could provide vehicle-oriented service via the MEC interface.

3.1.2. MSCN Layer. Selected vehicle nodes, such as city buses, taxis, and so on, are used to construct the MSCN layer, which could provide local computing resources. MSCN nodes collect BSMs sent by surrounding vehicles, then fulfill information fusion, and upload corresponding results to the MEC layer.

3.1.3. General Vehicle Layer. The general vehicle layer includes all road vehicle nodes, which are equipped with DSRC/LTE-V blocks, and could communicate with each other and RSU directly.

3.2. MDP Model Based on Two-Way Lane. In this paper, we select a two-way lane scenario to discuss the service migration problem.

The problem involves the following assumption conditions:

Object region is full C-V2X coverage, which means that all on-road vehicles could communicate with RSU.

All road vehicles are equipped with a C-V2X module and communicate with RSU via the PC5 interface.

There are enough public traffic vehicles to serve as MSCNs, which communicate with RSU and other on-road vehicles via C-V2X technologies.

RSUs communicate with each other via wired networks, such as fiber backbone and Ethernet.

The relationship between users and service providers is expressed by the distance factor. A service migration event should occur when $d(t) > N$. In this paper, the maximum value of N is set as 10.

MSCN's dwelling time is subject to an exponential distribution with a mean value equal to $1/\mu$.

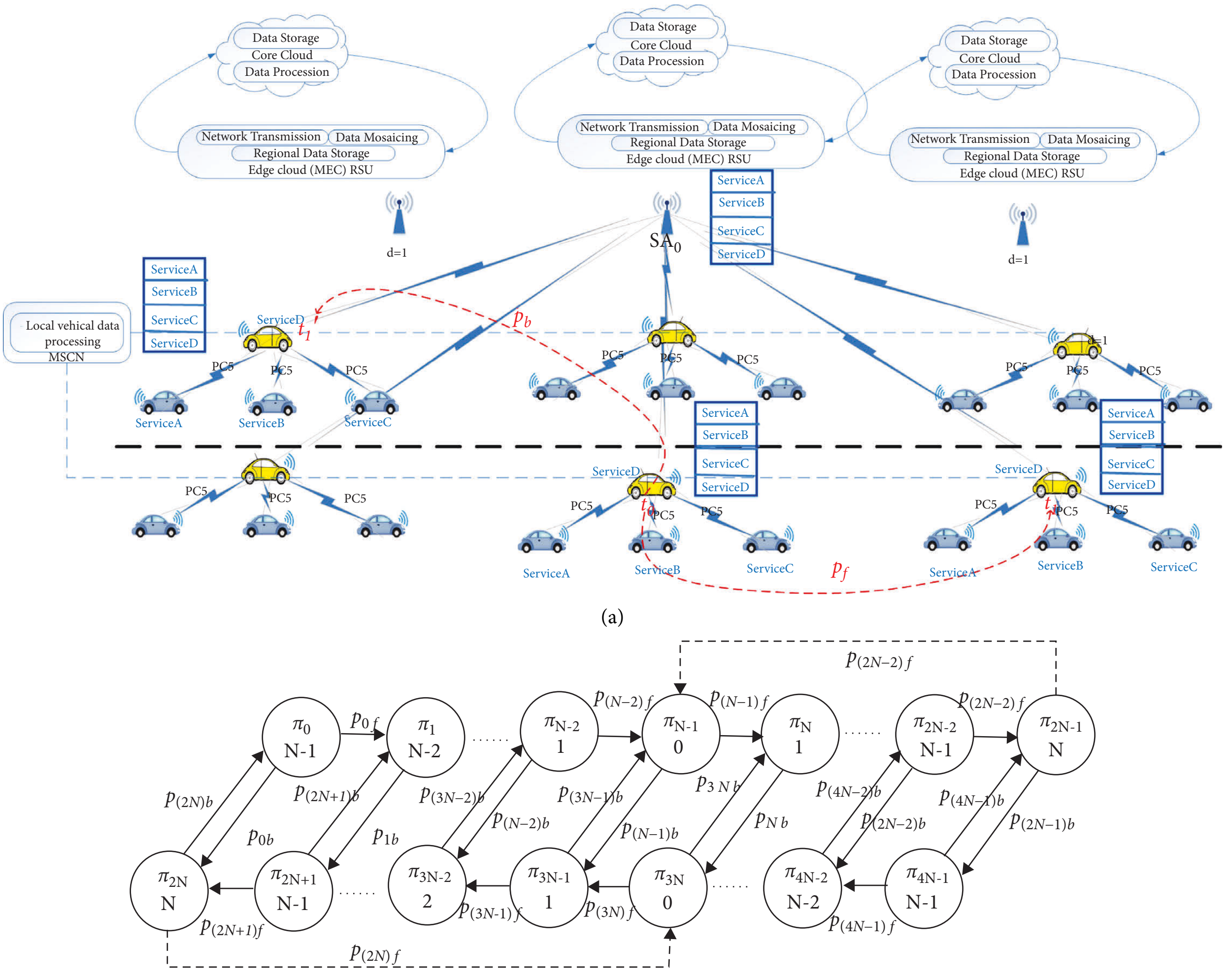
According to the above assumptions, the two-way lane MDP model is constructed. State space of service migration under the MSCN framework is shown in Figure 2(a), while corresponding state transition progress is denoted in Figure 2(b). As shown in Figure 3, service migration progress under MSCN architecture includes two migration levels, is denoted as social vehicle-oriented migration and public vehicle-oriented migration.

Social vehicle-oriented migration: a social vehicle changes its association MSCN (public vehicles), for example, MSCN1⟶MSCN2 as shown in Figure 3(a). It should be noted that social vehicle can only migrate to the MSCN with the same cruising direction.

Public vehicle-oriented migration: a public vehicle (MSCN) changes its association RSU, for example, RSU1⟶RSU2 as shown in Figure 3(b).

As shown in Figure 2(b), for an initial state $\pi_n, n \in [0, 4N-1]$, two inclining migration events are expressed as

$$\begin{cases} \pi_{n+1}, n \in [0, 2N-1], & \text{cruising forward}, \\ \pi_{n+1}, n \in [2N, 4N-1], & \text{cruising backward}. \end{cases} \tag{1}$$



(a)

(b)

Figure 2: MDP model based on two-way lane. (a) State space of service migration. (b) State transition progress.

State transition probability matrix is defined as

$$P = \begin{pmatrix} p_{01} & \cdots & p_{0(4N-1)} \\ \vdots & \ddots & \vdots \\ p_{(4N-1)0} & \cdots & p_{(4N-1)(4N-1)} \end{pmatrix}, \tag{2}$$

where $p_{ij}, i, j \in [0, 4N-1]$ is the transition probability of state i and j. In this paper, we take into account the driving intention of the vehicle to obtain the transition probability in the next section.

Assume that state probability of state π_n is χ_n, and

$$\sum_{n=0}^{4N-1} \chi_n = 1. \tag{3}$$

State transition progress could be denoted as

$$\chi_f = \chi_f \cdot P. \tag{4}$$

Thus, we can obtain the row probability vector χ_n of states at the migration strategy N according to equations (3) and (4).

3.3. Vehicle Motion Features. Obviously, service migration progress is decided by vehicle motion features, e.g., cruising direction, cruising speed, and relative speed. For MSCN architecture, two kinds of relative speed, the relative speed between social vehicles (users) and public vehicles (MSCN) and relative speed between public vehicles (MSCN) and RSU, should be considered.

According to [14], for traffic participants, the probability of the specific event in the time interval $[t, t + \Delta t]$, is defined as

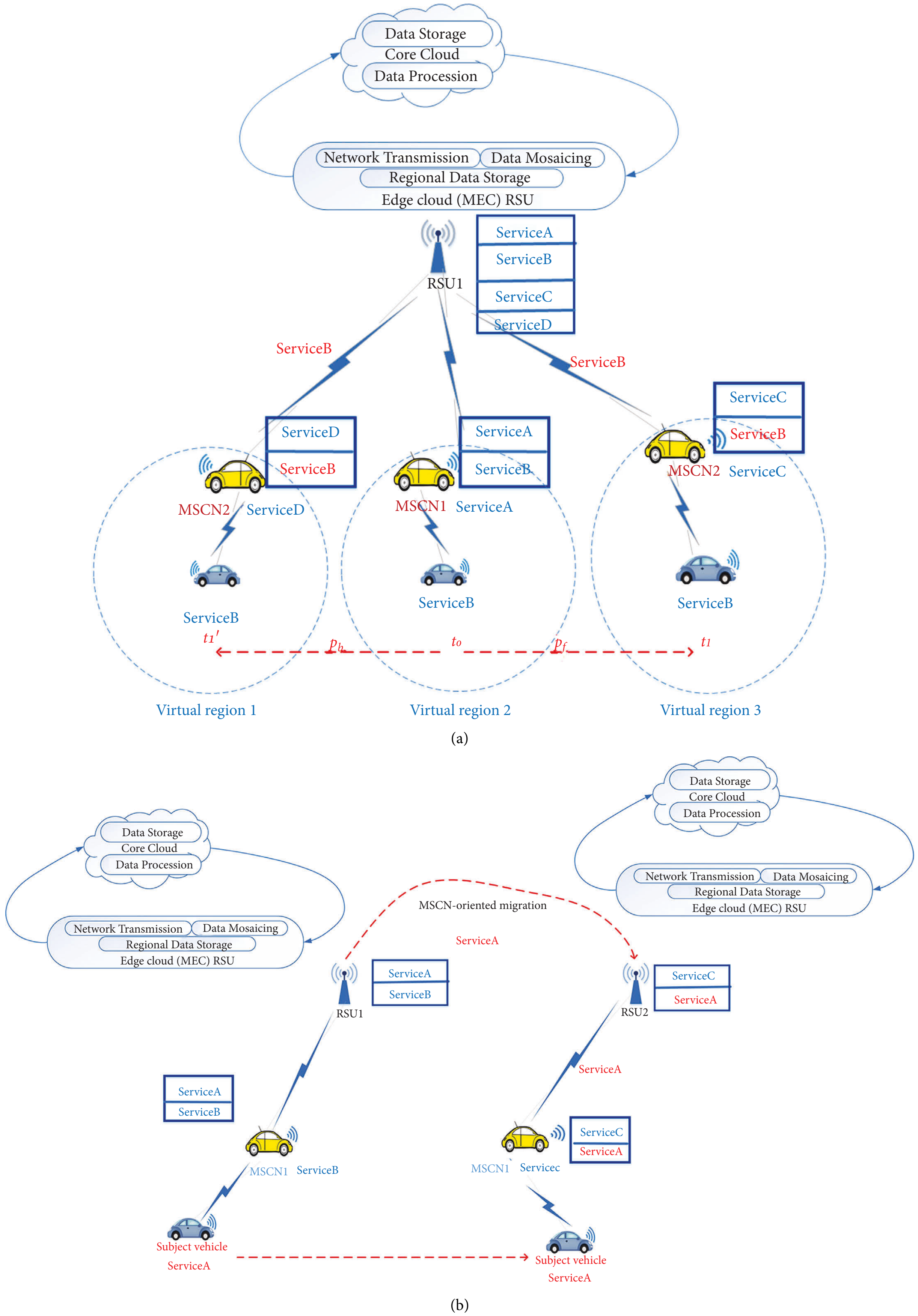
$$P(t; \Delta t) = \tau^{-1}[I(t)]\Delta t, \tag{5}$$

where $\tau^{-1} \in [0, \tau_{\max}^{-1}]$ is the number of events that occur per unit time interval under a certain condition and I is the situation-related risk indicator function.

Although equation (5) is constructed for driving risky evaluation, it could be used to express state transition probability. Hence, we define a situation indicator function I_s, assuming that traffic prefers to go to places with better traffic conditions and regional traffic condition C_{traffic}, to replace risk indicator function I, as shown as follows:

$$I_s \sim (d_I, C_{\text{traffic}}), \tag{6}$$

where d_I is obtained according to path planning information and $C_{\text{traffic}} \in (0, 1]$ is real-time traffic information, defined as



(a)

(b)

Figure 3: Two levels of service migration under MSCN. (a) Social vehicle-oriented migration. (b) Public vehicle-oriented migration.

$$C_{\text{traffic}(t)} = \begin{cases} \max\left(e^{\dfrac{-v_{\mathfrak{R}(t)}\left(\overline{v}_{\text{MSCN}}, 0\right)}{v_{\text{limit}}}}, \dfrac{v_{\mathfrak{R}(t)}\left(\overline{v}, v_{\text{MSCN}}\right)}{v_{\mathfrak{R}(t)}\left(\overline{v}_{\text{MSCN}}, 0\right)} \right), & v_{\mathfrak{R}}\left(\overline{v}_{\text{MSCN}}, 0\right) \neq 0, \\ 1, & \text{else}, \end{cases} \tag{7}$$

where $v_{\text{re}(t)}\left(\overline{v}, \overline{v}_{\text{MSCN}}\right)$ is the relative speed between social vehicles (users) for time t and public vehicles (MSCN), while $v_{\text{re}(t)}\left(\overline{v}_{\text{MSCN}}, 0\right)$ is the relative speed between public vehicles (MSCN) and RSUs for time $t.v_{\text{limit}}$ is the maximum speed limit on road. $\overline{v}$ is the average cruising speed of vehicles in coverage of corresponding MSCN. $\overline{v}_{\text{MSCN}}$ is the average cruising speed of MSCN in coverage of corresponding RSU.

The worse the traffic condition is, the higher the C_{traffic} value should be. Note here that we assume that path change is a result of bad traffic conditions.

$$d_I(t + \Delta t) = d_I(t) * \frac{\exp\left[1 - C_{\text{traffic}}(t + \Delta t)/C_{\text{traffic}}(t)\right]}{\exp\ \left(1 - C_{\text{traffic}}(t)\right)}. \tag{8}$$

In view of service migration, according to vehicle cruising direction, the migration area could be divided into two parts, namely, the forward area and the backward area. As mentioned earlier, here we consider four kinds of events: the following, turn left, turn right, and U-turn. Here we categorize the following, turn left, and turn right as forward area migration inclining events while U-turn as backward area migration inclining events.

Then, we define vehicle states as $\pi_n, n \in [0, M - 1]$, where M is the number of states, and $p_{ij}, i, j \in [0, M - 1]$, is the state transition probability, which is denoted as

$$p_{ij} = P\left(\pi^{\Delta t} = \pi_j \mid \pi = \pi_i\right) \sim \tau^{-1}\left(d_I, C_{\text{traffic}}\right)\Delta t, \tag{9}$$

where π_i is the state of time t and π_j is that of time $(t + \Delta t)$.

Considering vehicle's inclining events, forward or backward, and with the assumption that traffic prefers to go to places with better traffic conditions and regional traffic conditions C_{traffic}, the state transition probability proportional to d_I is defined as follows:

$$\begin{cases} \dfrac{p_{if}}{d_I^f} = \dfrac{p_{ib}}{d_I^b}, \\ p_{if} + p_{ib} = 1, \end{cases} \tag{10}$$

where p_{if} and p_{ib} are occurrence probabilities of forward event state π_f and forward event state π_b.

Then, p_{if} and p_{ib} could be denoted as

$$\begin{cases} p_{if} = \dfrac{d_I^f}{d_I^f + d_I^f}, \\ p_{ib} = \dfrac{d_I^b}{d_I^f + d_I^f}. \end{cases} \tag{11}$$

Substitute (8) into (11); then,

$$\begin{cases} p_{if} = \dfrac{\exp\left[1 - C_{\text{traffic}}^f(t + \Delta t)/C_{\text{traffic}}(t)\right]}{\exp\left[1 - C_{\text{traffic}}^b(t + \Delta t)/C_{\text{traffic}}(t)\right] + \exp\left[1 - C_{\text{traffic}}^f(t + \Delta t)/C_{\text{traffic}}(t)\right]}, \\ p_{ib} = \dfrac{\exp\left[1 - C_{\text{traffic}}^b(t + \Delta t)/C_{\text{traffic}}(t)\right]}{\exp\left[1 - C_{\text{traffic}}^b(t + \Delta t)/C_{\text{traffic}}(t)\right] + \exp\left[1 - C_{\text{traffic}}^f(t + \Delta t)/C_{\text{traffic}}(t)\right]}. \end{cases} \tag{12}$$

4. Service Migration Algorithm and Typical Applications under DIMDP-Based MSCN Architecture

4.1. Migration and Transmission Cost. Obviously, the optimization target of the service migration strategy is to minimize migration and transmission costs, and thus it could improve service performance.

Here we assume that the dwelling time of vehicle node i is calculated as

$$T_{\text{dewll,vehicle}}(i) = \frac{r_{t,rsu}}{v_{\text{vehicle}}(i)}, \tag{13}$$

where $r_{t,rsu}$ is the coverage radius of vehicle i's home RSU and $v_{\text{vehicle}}(i)$ is the cruising speed of vehicle i.

We define two cost functions, namely, transmission cost function $C_{\text{trans}}(d)$ and migration cost function $C_m(N)$, as follows:

$$\begin{cases} C_{\text{trans}}(d) = \dfrac{D_{trans}(d) \cdot r_{t,rsu} \cdot f}{v_i}, \\ \\ C_m(N) = D_m(N) + I_m(N), \end{cases} \tag{14}$$

where $D_{\text{trans}}(d)$ is the message transmission delay between the user vehicle node and service provider, at a distance d, (f) is the packet transmission frequency, and N is the migration distance. Migration progress should occur if $d > N$. $D_m(N)$ is the migration delay function, while $I_m(N)$ is the service interruption function, defined as follows:

$$\begin{cases} D_m(N) = \dfrac{\theta_m}{R} + \dfrac{(\beta+1)\lambda_f}{R}(N+1), \\ \\ I_m(N) = \exp\left(\dfrac{D_m(N)}{k}\right) - 1, \end{cases} \tag{15}$$

where R is the data transfer rate, λ_m is the size of the data that needs to be transmitted during the service request process, λ_f represents the size of data frame structure waiting to be sent in the queue, and β represents the number of the frame structure, which is affected by network congestion.

Message transmission delay is the sum of dissemination delay $D_t(d)$ as shown in (16) and queuing delay $D_q(d)$, as shown as

$$\begin{cases} D_t(d) = \dfrac{\lambda_m}{R} + d_c, \\ \\ D_q(d) = \dfrac{(\beta+1)\lambda_f}{R} d, \end{cases} \tag{16}$$

where d_c, which is a constant, represents the wireless communication delay between the vehicle and the RSU.

Then, we define $D_{\text{trans}}(d)$ as

$$D_{\text{trans}}(d) = D_t(d) + D_q(d) = \frac{\lambda_m}{R} + \frac{(\beta+1)\lambda_f}{R} d + d_c. \tag{17}$$

Average total cost function, where the state probability of state π_{n}, χ_n, is used as a weight to calculate the weighted average of cost, is defined as

$$\begin{aligned} \overline{C_a}(N) = {} & \sum_{d=1}^{N-1} C_{\text{trans}}(d)\left(\chi_{N-1\pm d} + \chi_{3N-1\pm d}\right) \\ & + C_{\text{trans}}(N)\left(\chi_{2N} + \chi_{2N-1}\right) \\ & + C_m(N)\left(\chi_{2N} p_{(2N)f} + \chi_{2N-1} p_{(2N-1)f}\right), \end{aligned} \tag{18}$$

while the optimal objective is to minimize the average total cost as follows:

$$P1\colon S = \arg\min_N \overline{C_a}(n). \tag{19}$$

The pseudocode of optimal strategy is shown in Algorithm 1, while corresponding variables and functions are explained in Table 1. Firstly, the algorithm obtained the state transition probability matrix P according to vehicle driving intention and obtained the row probability vector χ_n according to (3) and (4) in different migration distance N. Then, the average total cost $\overline{C_a}$ is obtained according to (18). Finally, we compare the total cost at each migration distance to obtain the optimal service migration distance corresponding to the minimum average total cost $\overline{C_a}$.

Note here that to avoid the impact of the sudden speed change on service migration performance, in this paper, we use regional average speed as the input value of vehicle speed.

4.2. Service Migration Strategy. The proposed MDP-based service migration strategy under the MSCN framework is shown in Algorithm 2, and the corresponding variables and functions are explained in Table 2. First, the current position of each vehicle is obtained and compared with the position of the previous moment to determine whether the MSCN directly under the vehicle has changed. After that, if the MSCN directly under the vehicle changes, determine whether the RSU directly under the vehicle has changed. Then, if the RSU directly under the vehicle changes, compare the distance from the current distance with the optimal strategy obtained by Algorithm 1. Finally, once the distance to the optimal policy is reached, service migration is performed and P is updated, with the new service provider as the center. This completes a service migration.

According to the 3GPP suggestion, corresponding parameters are set as

$$\begin{cases} \lambda_m = 1200\text{bit}, \\ \theta_m = 3.5\text{Mb}, \\ \lambda_f = 1024\text{bit}, \\ \text{R} = 5\text{Mbps}, \\ f = 10Hz. \end{cases} \tag{20}$$

The relationship between migration distance N and migration cost with different p values is shown in Figure 4. Here regional average speed is set as 20 m/s.

As shown in Figure 4, total cost increases with increasing p, and thus we verify the validity of the cost function. Moreover, the optimal object S, which is defined by (19), varies from 3 to 4.

The relationship between migration distance N and migration cost with different regional average speed values is shown in Figure 5.

As shown in Figure 5, the average total cost increases with increasing regional average speed, which expresses the influence of the vehicle's cruising intention. The value range of optimal object S is [2, 5] and consistent with the result of Figure 5.

4.3. Intersection Control under the MSCN Framework. Intersection control in the MSCN framework leverages RSU collaboration across multiple intersections and decides

Table 1: Explanation for Algorithm 1's variables and functions.

Variable/function	Explanation
pif	Probability set of forward area migration inclining events
λ_m	Service request data size
θ_m	Service migration data size
λ_f	The data size of frame structure
$r_{t,rsu}$	The coverage radius of RSU
v_{vehicle}	Vehicle speed
f	Service request frequency
$C_{\min}$	The minimal average total cost $\overline{C_a}$
S	The optimal migration strategy, a service migration event occurring when $d(t) \geq S$
get_Pmatrix (pif, N)	Obtain the state transition probability matrix P according to pif at migration distance N
get_state (P, N)	Obtain the row probability vector χ_n of states at the migration distance N according to (3) and (-4)
Get_cost (Cset, χ_n, N)	Obtain the average total cost $\overline{C_a}(N)$ according to (18). Cset is $\{\lambda_m, \theta_m, \lambda_f, r_{t,rsu}, v_{\text{vehicle}}, f\}$.

```
Input:
  $p_{if}$, $\lambda_m$, $\theta_m$, $\lambda_f$, $(r)$, $r_{t,rsu}$, $v_{\text{vehicle}}$, $(f)$
Output:
  Optimal migration strategy $S$
$C_{\min} \leftarrow \infty$;
$S \leftarrow 10$;
for $N = 1$ to 10 do
$P \leftarrow$ get_Pmatrix ($p_{if}$, $N$);
$\chi_n \leftarrow$ get_state ($P$, $N$);
$\overline{C_a} \leftarrow =$ get_cost (Cset, $\chi_n$,N);
if $C_{\min} > \overline{C_a}$
  $C_{\min} \leftarrow \overline{C_a}$;
  $S \leftarrow N$;
else
  break;
end
end
Return $S$;
```

Algorithm 1: Optimal strategy.

Table 2: Explanation for Algorithm 2's variables and functions.

Variable/function	Explanation
V	Vehicle set
(xcn, ycn)	Center coordinates of middle cell
$t1$	Last moment
$t2$	Current moment
d	Distance between service requester and service provider
Cset	A set of $\{\lambda_m, \theta_m, \lambda_f, r_{t,rsu}, v_{\text{vehicle}}, f\}$
pif	As explained in Table 1
Vtarget	The vehicle member currently traversed
get_position (V, t)	Get the position of vehicle V at time t
get_cellnum ((xt, yt), (xcn, ycn))	Obtain the cell number of coordinate (xt,yt) by taking coordinate (xcn, ycn) as a reference point
get_MSCN (cellnum)	Obtain the MSCN id corresponding to cell cellnum
get_rsu (MSCNid)	Obtain the RSU id to which MSCNid belongs
mscnMigration	A map < vehicle_id,flag > which represents the vehicle who's id is vehicle_id will perform or not service migration at the MSCN level while the flag is true or false
rsuMigration	A map < vehicle_id,flag > which represents the vehicle who's id is vehicle_id will perform or not service migration at the RSU level while the flag is true or false
setMigration	The service migration set {rsuMigration, mscnMigration}
get_strategy (pif, cset)	Obtain the optimal strategy according to Algorithm 1 with variable pif and variable Cset

```
Input:
  $V$, $(x_{cn}, y_{cn})$, $t1$, $t2$, $d$, $Cset$, $p_{if}$
Output:
  setMigration {rsuMigration, mscnMigration}
for $V_{target}$ each in $V$
  $(x_{t1},y_{t1})$ ← get_position ($V_{target}$,t1);
  $(x_{t2},y_{t2})$ ← get_position ($V_{target}$,t2);
  cellnum1 ← getcell_num ($(x_{t1}, y_{t1})$,$(x_{cn}, y_{cn})$);
  cellnum2 ← getcell_num ($(x_{t2}, y_{t2})$,$(x_{cn}, y_{cn})$);
  MSCNid1 ← get_MSCN (cellnum1);
  MSCNid2 ← get_MSCN (cellnum2);
  rsu1 ← get_rsu (MSCNid1);
  rsu2 ← get_rsu (MSCNid2);
  if (MSCNid1 = = MSCNid2) then
    rsuMigration.insert ($V_{target}$, false);
  else
    mscnMigration.insert ($V_{target}$, true);
end
if (rsu1 = = rsu2) then
  rsuMigration.insert ($V_{target}$, false);
else
  $S$ ← get_strategy ($p_{if}$,Cset);
if ($d > S$) then
  rsuMigration.insert ($V_{target}$, true);
  update $p_{if}$;
else
  rsuMigration.insert ($V_{target}$, false);
end
end
end for
Return setMigration;
```

Algorithm 2: Service migration under MSCN framework.

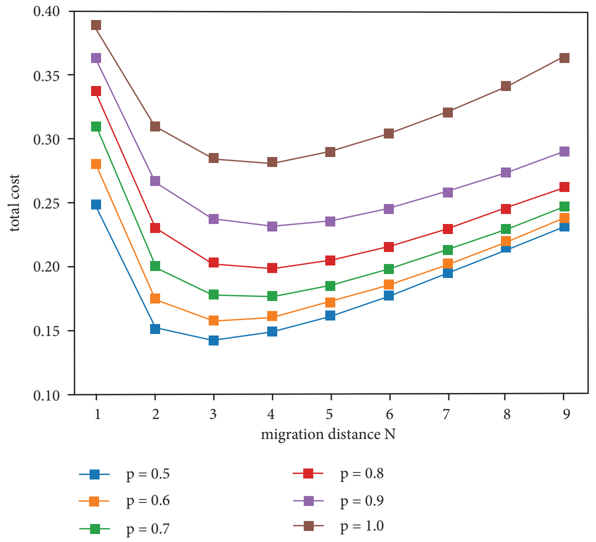


Figure 4: Total cost vs. migration distance N at different p.

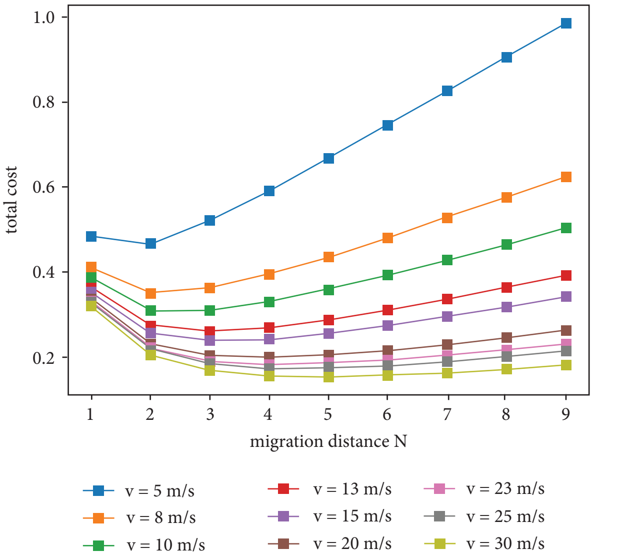


Figure 5: Total cost vs. migration distance (N) at different regional average speed (v).

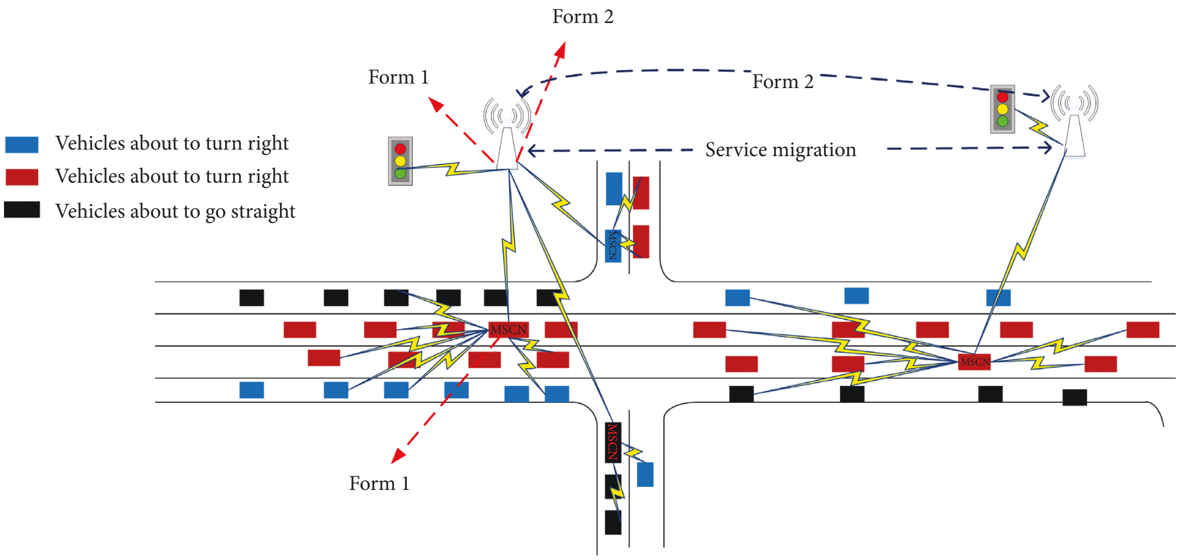

Form 2
Form 1
Form 2
Service migration
🟦 Vehicles about to turn right
🟥 Vehicles about to turn right
⬛ Vehicles about to go straight
MSCN
MSCN
MSCN
MSCN
Form 1

Figure 6: Intersection control under MSCN framework.

whether to perform service migration of applications based on DIMDP.

Intersection traffic flow detection in the MSCN framework requires to collect three sets of data, including

(1) The number of queued vehicles in each lane group (left-turn, straight turn, and right-turn lane group) of the main road at intersections.

(2) The number of vehicles queued on the branch entrance road.

(3) The speed of straight-line vehicles exiting the intersection, which can be obtained by the vehicle's own speed sensor.

The intersection control application in the MSCN framework is shown in Figure 6, in which blue vehicles represent upcoming right-turn vehicles after passing through the intersection, red represents upcoming straight-traffic vehicles, and black represents upcoming left-turn vehicles. RSUs deploying MEC services are set up between upstream and downstream intersections, and the distance between RSUs is determined according to their transmission range. The specific data flow is as follows:

(1) The road vehicles decide whether to perform service migration according to the DIMDP in the MSCN framework.

(2) The road vehicles send application messages (including node id, MSCN id, direction information, vehicle current speed, vehicle current acceleration, and time stamp information) to the MSCN in the region through the information distribution mechanism in the MSCN framework.

(3) The MSCN receives road vehicle information and performs preliminary calculations and statistics to obtain Table 3.

(4) The MSCN decides whether to perform application service migration based on the DIMDP optimization policy.

(5) The MSCN sends Table 3 to the optimal RSU.

(6) Optimal RSU assembles and organizes Table 3 from each cluster and MSCN to obtain Table 4.

(7) The adjacent RSUs exchange Table 4 to generate updating Table 4.

(8) The optimal RSU sends updating Table 4 to the traffic light controller and executes intersection control.

Table 3 shows the preliminary results after MSCN collects all the normal vehicle nodes and its own application information in its region and conducts statistical sorting. Then, the MSCN sends Table 3 to the optimal RSU, which receives Table 3 from each cluster and each MSCN within the cluster, processes, and counts it to form Table 4.

5. Simulation and Result Analysis

5.1. Simulation Setup. Simulation is done based on the NS-3.28 platform. Simulation parameters are listed in Table 5. We excerpted a 600-meter two-way road used in the simulation as a diagram as shown in Figure 7.

Corresponding communication parameters are listed in Table 6.

Note here that V2X information is one kind of timeliness information. Hence, re-transmission procedure is disabled. Moreover, until today, most C-V2X-based modules do not support power adjusting function and use fixed transmit power, valued as 23 dBm.

Three parameters, average backhaul delay (ABL), packet delivery rate (PDR), and effective feedback ratio (EFR), are used to evaluate the performance of the proposed mechanism.

Here effective feedback ratio is defined as the ratio of received effective service message number to uploaded BSM number, while the average backhaul delay is defined as the service response delay, which is the interval between the source node's BSM sending time t_{bsm} and destination node's service response time t_{service}. Considering the influence of BSM dissemination delay, here we employ the

Table 3: MSCN.

Node id	MSCN id	Sending time	Direction	Speed (m/s)	Acce (m/s2)
1	1	10.23.33	Right	20.5	1.01
8	1	10.23.45	Left	21.2	0.32
3	1	10.23.55	Right	24.3	1.51
2	1	10.23.58	Straight	23.5	0.58
. . .	. . .	. . .	. . .	. . .	. . .

Table 4: RSU.

Cluster id	MSCN id	Receiving time	Total right	Total left	Total straight
1	3	10.32.08	8	3	12
1	1	10.32.13	4	6	8
1	2	10.32.45	5	1	5
3	3	10.32.58	4	6	16
. . .	. . .	. . .	. . .	. . .	. . .

Table 5: Simulation parameters.

Parameter	Value
Lane number	2
RSU number	5
MSCN number	15
Distance between RSUs	300m
Lane width	3m
Lane length	1500m
Vehicles' velocity	± 20 m/s– ± 25 m/s

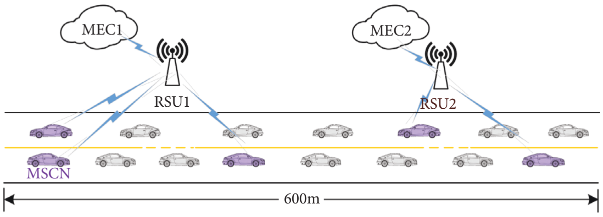


Figure 7: The diagram of the two-way road used in the simulation.

Table 6: Communication simulation parameters.

Parameter	Value
Frequency band	5.9 GHz
Channel bandwidth	10 MHz
Data rate	5 Mbps
Transmit power	23 dBm
Packet size	1200 bit
Packet delivery frequency	10 Hz

average BSM dissemination delay to calculate ABL t_{ABL} as follows:

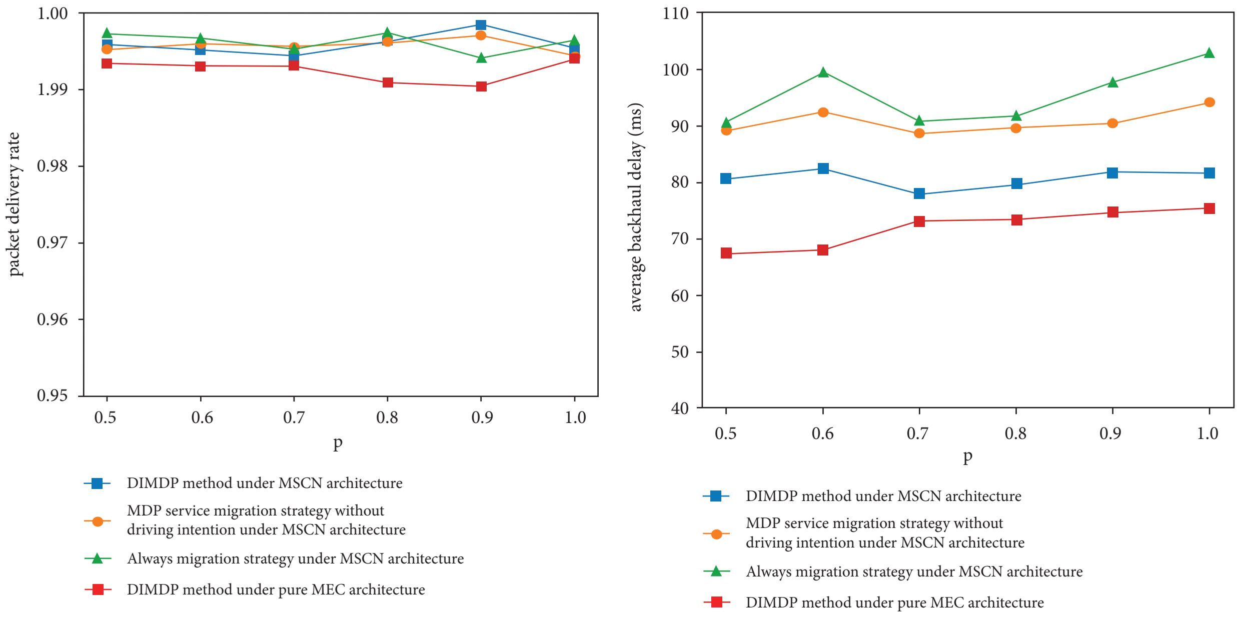
$$t_{\text{ABL}} = \frac{1}{N}\sum_{i=0}^{N} t_{\text{service}} - t_{bsm}, \tag{21}$$

where N is the BSM packet number.

To verify the performance of DIMDP under MSCN architecture, three others are selected as a contrast.

(i) MDP service migration strategy without driving intention under MSCN architecture.

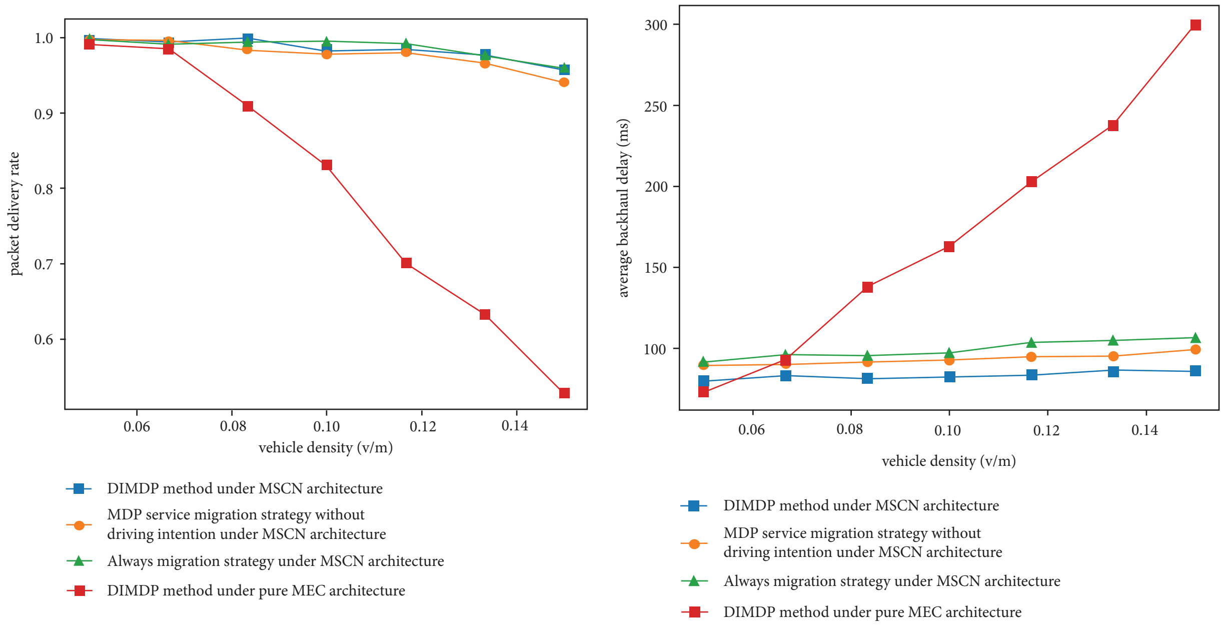
(ii) Always migration strategy under MSCN architecture, whose migration condition is set as $d\,(t) \geq 0$.



(a)

(b)

Figure 8: The impact of p value. (a) Packet delivery rate. (b) Average backhaul delay.



(a)

(b)

Figure 9: The impact of vehicle density. (a) Packet delivery rate. (b) Average backhaul delay.

(iii) DIMDP under pure MEC architecture.

As shown in Figure 8, the p value has no significant effect on packet delivery rate and backhaul delay. The proposed DIMDP method performs better backhaul delay, which means the V2X security applications can respond faster.

As shown in Figure 9, with the increase in road vehicle density, both the ABL and PDR of the proposed DIMDP method under MSCN architecture are nearly equal to a constant and perform better, which means it can still maintain excellent and stable performance in scenes with high vehicle density.

As shown in Figure 10, the higher the vehicle velocity is, the lower the effective feedback ratio should be. Moreover, the effective feedback ratio of the proposed DIMDP method under MSCN architecture is higher than that of other methods.

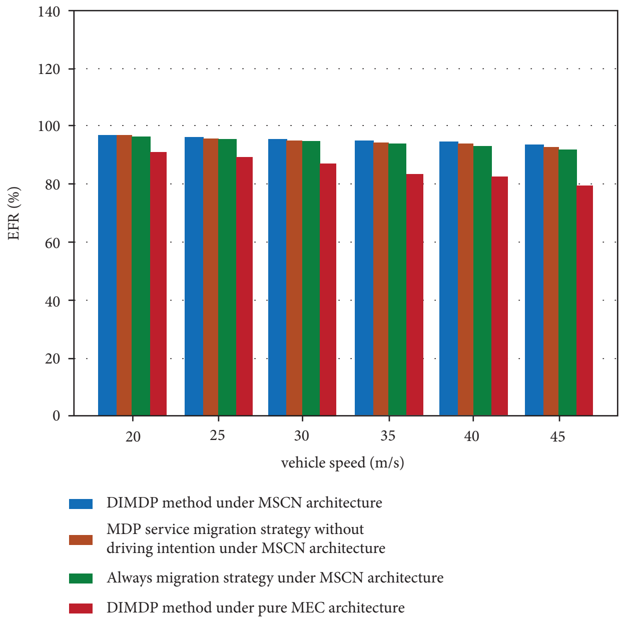


FIGURE 10: EFR vs. vehicle speed of four methods.

6. Conclusion

V2X safety applications have time-sensitive characteristics, and their application performance largely depends on the effectiveness of the information sharing issues. Due to the high mobility of vehicle nodes, service providers, such as MEC servers and cloud servers, could not disseminate the useful messages to vehicle nodes on time. Fortunately, a reasonable calculation framework and effective transmission guarantee measure are useful to solve the above problems.

MSCN framework, which uses public vehicles as mobile service providers, could decrease the service migration rate and maintain service continuity. On the other hand, although the MDP model could be used to express service migration progress, the vehicle motion features should be considered in the construction of the MDP model.

In this paper, we use vehicle driving intention and regional traffic conditions to express the vehicle's mobile tendency. A complex cost function, which considers both transmission factor and migration cost, is defined. Simulation results show that the proposed DIMDP model performs well.

In the future, we shall work on the vehicle motion model, thus further refining our model and improving its performance.

Data Availability

The simulation data supporting the system performance analysis can be obtained from https://github.com/lkw-sssy/DIMDP-A-Driving-intention-based-MDP-service-migration-model-under-MEC-MSCN-architecture.

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 Special Key Project of Chongqing Technology Innovation and Application Development under grant no. csct2021jscx-gksbX0057.

References

[1] D. Grewe, M. Wagner, M. Arumaithurai, I. Psaras, and D. Kutscher, "Information-centric mobile edge computing for connected vehicle environments: challenges and research directions," in *Proceedings of the Workshop on Mobile Edge Communications*, pp. 7–12, Los Angeles, CA, USA, August 2017.

[2] Q. Han, X. Zhanga, J. Zhang et al., "Research on resource scheduling and allocation mechanism of computation and transmission under MEC framework," in *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 437–442, IEEE, Auckland, New Zealand, October 2019.

[3] Q. Han, L. Lin, L. Zeng, J. Zhang, L. Ye, and K. Ling, "Research on service migration and typical applications under MSCN framework," in *Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference*, pp. 1023–1028, IEEE, Indianapolis, IN, USA, October 2021.

[4] X. Yu, M. Guan, M. Liao, and X. Fan, "Pre-migration of vehicle to network services based on priority in mobile edge computing," *IEEE Access*, vol. 7, pp. 3722–3730, 2019.

[5] S. Ge, M. Cheng, and X. Zhou, "Interference aware service migration in vehicular Fog computing," *IEEE Access*, vol. 8, pp. 84272–84281, 2020.

[6] A. Nadembega, A. S. Hafid, and R. Brisebois, "Mobility prediction model-based service migration procedure for follow me cloud to support QoS and QoE," in *Proceedings of the 2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, Kuala Lumpur, Malaysia, May 2016.

[7] J. Xu, X. Ma, A. Zhou, Q. Duan, and S. Wang, "Path selection for seamless service migration in vehicular edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 9040–9049, 2020.

[8] T. Taleb and A. Ksentini, "An analytical model for follow me cloud," in *Proceedings of the 2013 IEEE Global Communications Conference (GLOBECOM)*, pp. 1291–1296, Georgia, GA, USA, December 2013.

[9] A. Ksentini, T. Taleb, and M. Chen, "A Markov decision process-based service migration procedure for follow me cloud," in *Proceedings of the 2014 IEEE International Conference on Communications (ICC)*, pp. 1350–1354, IEEE, Sydney, Australia, June 2014.

[10] T. Taleb, A. Ksentini, and P. A. Frangoudis, "Follow-me cloud: when cloud services follow mobile users," *IEEE Transactions on Cloud Computing*, vol. 7, no. 2, pp. 369–382, 2019.

[11] M.-A. Lèbre, F. Le Mouël, and E. Ménard, "On the importance of real data for microscopic urban vehicular mobility trace," in *Proceedings of the 2015 14th International Conference on ITS Telecommunications*, pp. 22–26, ITST, Copenhagen, Denmark, December 2015.

[12] F. A. Silva, A. Boukerche, T. R. Silva, L. B. Ruiz, and A. A. Loureiro, "A novel macroscopic mobility model for vehicular networks," *Computer Networks*, vol. 79, pp. 188–202, 2015.

[13] O. Biham, A. A. Middleton, and D. Levine, "Self-organization and a dynamical transition in traffic-flow models," *Physical Review A*, vol. 46, no. 10, pp. R6124–R6127, 1992.

[14] J. Eggert and F. Mueller, "A foresighted driver model derived from integral expected risk," in *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 1223–1230, Auckland, New Zealand, October 2019.