

Research Article

Software-Defined Collaborative Offloading for Heterogeneous Vehicular Networks

Wei Quan,^{1,2} Kai Wang^{1,2,3}, Yana Liu,¹ Nan Cheng,²
Hongke Zhang,¹ and Xuemin (Sherman) Shen²

¹School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China

²Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada

³School of Computer and Control Engineering, Yantai University, Shandong, China

Correspondence should be addressed to Kai Wang; wangkai.phd@outlook.com

Received 28 December 2017; Accepted 20 March 2018; Published 24 April 2018

Academic Editor: Tao Han

Copyright © 2018 Wei Quan 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.

Vehicle-assisted data offloading is envisioned to significantly alleviate the problem of explosive growth of mobile data traffic. However, due to the high mobility of vehicles and the frequent disruption of communication links, it is very challenging to efficiently optimize collaborative offloading from a group of vehicles. In this paper, we leverage the concept of Software-Defined Networking (SDN) and propose a software-defined collaborative offloading (SDCO) solution for heterogeneous vehicular networks. In particular, SDCO can efficiently manage the offloading nodes and paths based on a centralized offloading controller. The offloading controller is equipped with two specific functions: the hybrid awareness path collaboration (HPC) and the graph-based source collaboration (GSC). HPC is in charge of selecting the suitable paths based on the round-trip time, packet loss rate, and path bandwidth, while GSC optimizes the offloading nodes according to the minimum vertex cover for effective offloading. Simulation results are provided to demonstrate that SDCO can achieve better offloading efficiency compared to the state-of-the-art solutions.

1. Introduction

With the significant advance of automobile technologies, the Internet of vehicles (IoV) is becoming a standard feature in the near future, where vehicles are equipped with advanced communication devices, and exchange data and information through vehicle-to-everything (V2X) communications [1, 2]. Enabled by IoV technologies, a myriad of emerging vehicular applications occur to enrich the transport ecosystem, such as real-time high-definition navigation, multimedia services on wheels, and self-driving system [3, 4]. However, these applications and services require an exponential escalation of mobile data, which should be efficiently supported by the vehicular networks.

Many wireless communications technologies are developed to support the potential vehicular scenarios. For instance, the Dedicated Short-Range Communication (DSRC) enables efficient real-time information exchange among vehicles [5]. Wireless Access for Vehicular Environment (WAVE)

has been applied to the Vehicle-to-Infrastructure (V2I) communications [6]. The Long-Term Evolution (LTE) has the ability to support vehicular communications in a wide range [7, 8]. Not surprisingly, cellular network technologies play a vital role in providing ubiquitous and reliable Internet access for mobile vehicles. However, the LTE network nowadays is straining to meet the current mobile data demand. With the explosive growth of vehicular data, LTE networks are easily overloaded and degrade the experience of both nonvehicular and vehicular users.

Therefore, simply using the single type of access technologies is far from enough to satisfy all the requirements of vehicular services. Researchers have considered solutions to offload traffic from the core network to other wireless networks [9, 10]. This solution can effectively reduce the congestion of transmission paths in the core network. Figure 1 describes three typical offloading scenarios in a heterogeneous vehicular network, where the traffic is offloaded from macro base station via small cell base stations, ad hoc networks

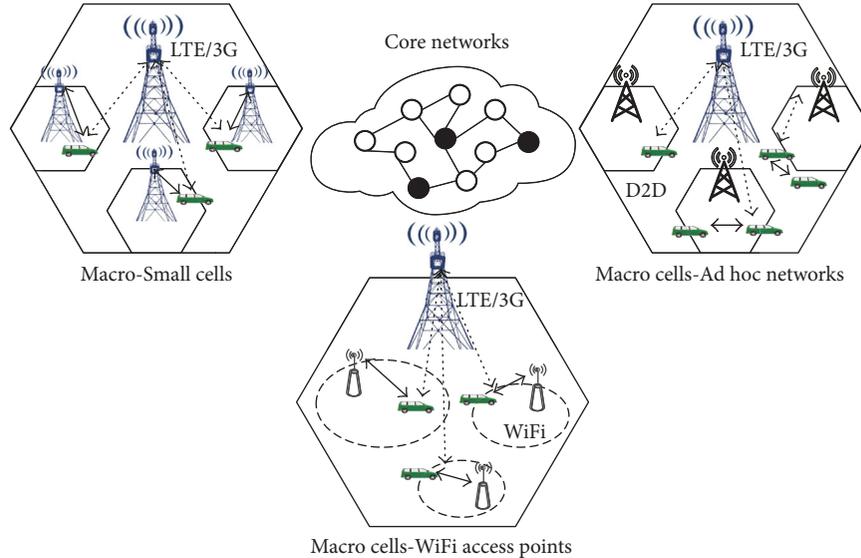


FIGURE 1: Offloading scenarios in heterogeneous vehicular networks.

(no base station), and WiFi access points, respectively. For example, WiFi offloading can alleviate the congestion in cellular networks by delivering data through WiFi access instead of the cellular base station [11]. Statistics show that more than 60 percent of the cellular traffic can be offloaded by WiFi networks.

Recently, the fifth generation (5G) mobile communications system is also designed to integrate networking, computing, and storage resources into one programmable and unified infrastructure. It is not only an evolution of mobile broadband networks but also a collaborative ecosystem of heterogeneous communications technologies. Therefore, the data offloading via various communications technologies has been considered in the 5G deployment. With this feature, 5G has a great potential of providing massive access, high reliability, global coverage, and very low latency for challenging situations like vehicular networks.

However, efficiently offloading the vehicular data is a challenging issue. In the vehicular environment, data offloading is opportunistic due to the limited coverage of heterogeneous wireless technologies and high mobility of vehicles [12]. The connection time between a vehicle and an offloading point (e.g., WiFi access points and small cell base stations) might be very short; therefore the content sources and transmission paths will be highly dynamic. Therefore, with such dynamics, the quality of service (QoS) requirements are difficult to guarantee.

Software-defined networking (SDN) provides a promising and referable approach due to the features of centralized controlling, global decision, and high execution power. Our previous work has investigated a new networking paradigm named Smart Identifier NETWORKING (SINET) and proposed a SINET customized solution for software-defined vehicular networks (SINET-V) [13, 14]. Based on such a SINET-V architecture, in this paper, we propose a software-defined collaborative offloading (SDCO) solution, which employs a

centralized software-defined controller to globally distribute offloading policies to vehicular nodes in heterogeneous vehicular networks. Specifically, the hybrid awareness path collaboration (HPC) and the graph-based source collaboration (GSC) are considered in the offloading controller. HPC is in charge of selecting the suitable paths based on the round-trip time, packet loss rate, and path bandwidth, while GSC optimizes the offloading nodes according to the minimum vertex cover for effective offloading. These specific functions are detailed in the following sections.

The remainder of this paper is organized as follows. In Section 2, we analyze the related works on offloading management. In Section 3, we describe the proposed SDCO solution and the optimization model. In Sections 4 and 5, we detail the specific functions for collaborative offloading: hybrid-awareness path collaboration and graph-based source collaboration. Simulation results validate the performance of SDCO in Section 6. Finally, we conclude the paper and discuss the future works in Section 7.

2. Related Works

Currently, many research efforts have begun towards realizing efficient heterogeneous vehicular communications in a cooperative manner. In our previous work, a collaborative vehicular edge computing framework named as CVEC is introduced to support more scalable vehicular services and applications [15]. Chen et al. presented a novel architecture of software-defined Internet of vehicles (SD-IoV) and developed a centralized vehicular connection management approach to guarantee the QoS [16]. Zheng et al. presented a multilayer and soft-defined HetVNET with a far more flexible configuration capability [17]. Dong et al. analyzed the issues of energy-efficient cluster management in heterogeneous vehicular networks [18]. Zhang et al. proposed a software-defined space-air-ground integrated network architecture

for supporting diverse vehicular services [19]. However, the above newest research mainly concentrated on the general architecture designs for heterogeneous vehicular networks yet failed to analyze the specific offloading mechanisms.

In the aspect of offloading management, Liu et al. proposed an incentive mechanism for computation offloading by using edge computing to deliver computation to the edge of pervasive networks nearby mobile users [20]. Cheng et al. discussed data offloading technologies for big data driven vehicular network [21]. Aijaz et al. made a survey on mobile data offloading from both technical and business views [22]. Zhuo et al. proposed an incentive framework to motivate users to offload cellular traffic with minimal cost [23]. To minimize the incentive cost, sources with large offloading potential and delay tolerance are preferred in their solution. Lee et al. focused on a quantitative study on the performance of mobile data offloading through WiFi access [24]. Singh et al. built a general and tractable model to analyze data offloading in heterogeneous networks [25]. Besides, Cheng et al. used a queueing model to analyze the opportunistic WiFi offloading in vehicular networks [26]. Rebecchi et al. made a complete survey on the data offloading techniques in cellular networks, including classification of mobile data offloading, nondelayed offloading, delayed offloading, assessing mobile data offloading, and open challenges [9]. However, there are few works on how to collaborate among different offloading schemes in heterogeneous network scenarios.

Thanks to these efforts, some progress has been made on understanding the collaborative offloading in vehicular networks. However, there are still several challenges waiting to be overcome due to the intractable characteristics in vehicular environment. This paper leverages the advantages of SDN and proposes a SDN-based collaborative offloading for heterogeneous vehicular networks, which can globally and efficiently distribute offloading policies to vehicular nodes.

3. System Modelling and Problem Formulation

3.1. System Modelling. The typical offloading is to use local communication to offload the opportunistic traffic of the base station. Neighboring peers can communicate with each other without using a network infrastructure and can retrieve the expected content from the peer. It can support to dynamically adjust the access mode for the user equipment and then offloads the data traffic of the macro base station to the small base station.

In SDCO, we employ a centralized software-defined controller to globally distribute offloading policies to vehicular nodes. This design aims to guarantee the transmission reliability by avoiding a large delay in distributed negotiation. It can also improve the overall network throughput and alleviate the problem of mobile data traffic congestion. Figure 2 shows the SDCO system architecture. Based on this architecture, we will focus on how the controller makes optimization for efficient offloading policies.

3.2. Problem Formulation. Typical offloading allows users to relay data through other users within the network coverage, indirectly accessing the base station for data resources. In

addition, the offloading scheme allows UEs with certain QoS requirements, in particular, to guarantee UEs with high-speed mobility to access the macro base station so as to avoid frequent handovers. We transferred this problem as a 0-1 linear programming problem, by which to determine the best source and preferred access mode for each user, for example, to select the base station or relay nodes. The objective is to maximize the amount of access users by offloading some users to the small base station so that the system can serve more UEs. Based on the above considerations, the optimization model is defined as follows:

$$\max_{x,y,z} N = \sum_{k=1}^K \left(\sum_{j=1}^J y_{jk} + \sum_{m=1}^M z_{mk} \right), \quad (1)$$

$$\sum_{j=1}^J x_{mj} + \sum_{k=1}^K z_{mk} \leq C_m, \quad (2)$$

$$\sum_{m=1}^M z_{mk} + \sum_{j=1}^J y_{jk} \leq 1, \quad (3)$$

$$\sum_{k=1}^K y_{jk} \leq \sum_{m=1}^M x_{mj} \leq 1, \quad (4)$$

$$x_{mj} \leq \left(1 - \sum_{k=1}^K y_{jk} g_{mk} \right) e_{mj}, \quad \forall j \forall m > 0. \quad (5)$$

x_{mj} denotes the binary variable characterizing the connection status between base station (b_m) and relay node (r_j), 0 for without connection and 1 with connection. y_{jk} denotes the binary variable characterizing the connection status of user d_k and r_j , 0 for without connection and 1 for with connection. z_{mk} denotes the binary variable characterizing the connection state between d_k and b_m , 0 for without connection and 1 for with connection. In formula (1), $\sum_{k=1}^K (\sum_{m=0}^M z_{mk})$ indicates the number of users connected to the network through the relay node. Formulas (2)–(5) are constraints. Wherein, formula (2) indicates that the maximum number of users and relay nodes which the base station can connect with is C_m , that is, the capacity of the base station. Formulas (3) and (4) indicate that each user node can only be connected with one base station or relay node at most at any time.

If a user node needs to connect to a base station through a relay node, the user must be out of coverage of the base station, so it is limited by formula (5). g_{mk} indicates whether d_k is within the coverage of b_m , 1 means it is within its range, and 0 means it is not. e_{mj} indicates whether r_j is within the coverage of b_m , 1 means it is within its range, and 0 means it is not.

To resolve this optimization problem, a greedy search can be used to achieve the optimized solution for the offloading problem. However, it might be a time-consuming task to achieve a globally optimized solution, especially for a large-scale scenario. In practical operations, the controller can aware more information, like the dynamic topology, mobility trace, connection relationship, and so on. Therefore, in this

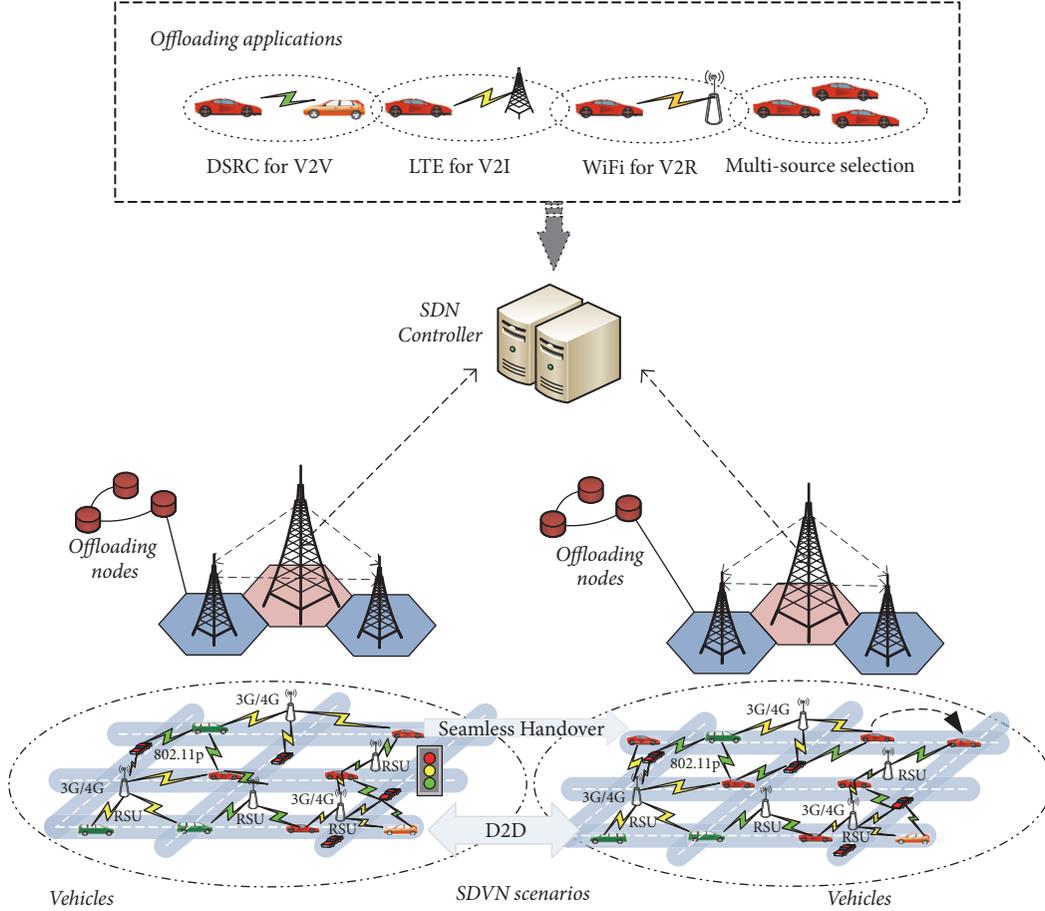


FIGURE 2: The SDCO system architecture.

work, we proposed two mechanisms to assist the controller to make offloading decisions. One is the hybrid awareness path collaboration (HPC) function, and the other one is the graph-based source collaboration (GSC) function. Both of them will be detailed in the following sections.

4. Hybrid Awareness Path Collaboration Function

The controller with global network information will promote the path collaboration more efficiently. SDCO adopts the HPC mechanism to address the path collaboration issue. Different from other existing solutions, the HPC mechanism is based on a new hybrid path metric, termed as Q . This path metric is able to synthetically depict the path status by considering three different parameters: Round-Trip Time (RTT), the packet loss rate (LOSS), and the path bandwidth (BWT). In the initial phase, the controller can obtain the real-time path status Q_i for the path i through sensing and notification from the nodes. Then, the controller will check whether the path status Q_i is bigger than a threshold τ to determine whether to add the path into the path scheduling list (PSL). Besides, the controller will monitor the variation of the status of each path in the PSL and then trigger an update to maintain the list. If the variation exceeds a threshold ω ,

this path will be updated in the PSL. Figure 3 illustrates the process of the HPC mechanism. In the following, we introduce the hybrid awareness path collaboration in detail.

4.1. Path Metric Calculation. In this section, we define a new path metric to synthetically depict the path quality. The metric of each path relies on three link parameters: RTT, LOSS, and BWT.

To be general, both RTT and LOSS need to be considered to evaluate link quality. Based on this, we make a further revision. We believe that there is a close relation between the path bandwidth and the path quality. The path bandwidth reflects the maximum throughput that the link can achieve within a unit time. Therefore, it is necessary to correctly reflect the actual situation of the paths by considering the above factors comprehensively.

$$\text{PATH} = \{P_1, P_2, \dots, P_i, \dots, P_n, n \geq 2\}. \quad (6)$$

In the following, we take PATH as a group of available paths between the vehicular users and the services providers, as shown in formula (6). We define Q_i as the path metric of P_i , as shown in formulas (7), (8), and (9). It is easily to see that the better the path quality is, the bigger the metric value Q_i is.

$$Q_i = \alpha \cdot \frac{1}{\text{RTT}_i} + \beta \cdot \text{BWT}_i + \theta \cdot \frac{1}{\text{LOSS}_i}, \quad (7)$$

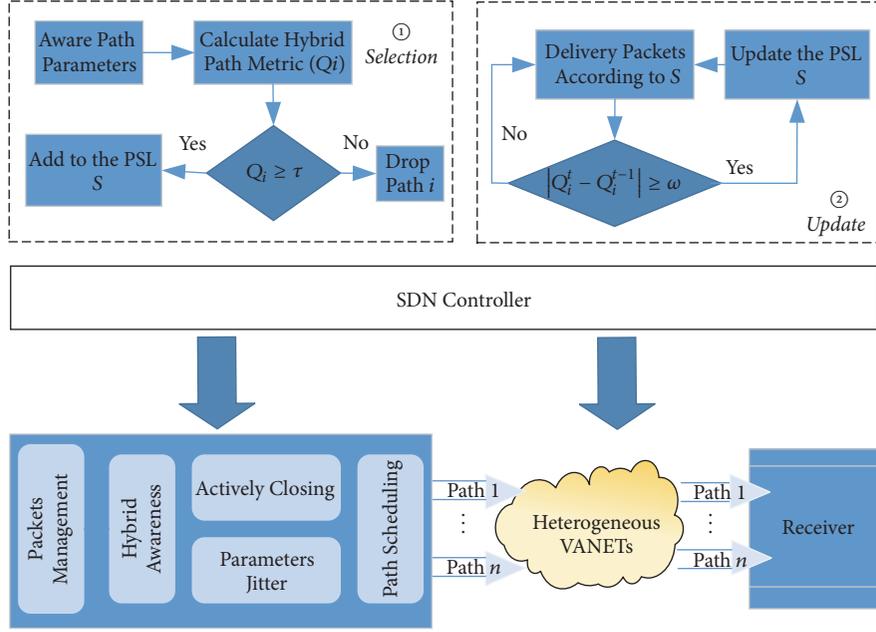


FIGURE 3: The process of the HPC mechanism.

$$\alpha + \beta + \theta = 1, \quad (8)$$

$$0 \leq \alpha, \beta, \theta \leq 1. \quad (9)$$

In the formula (7), RTT_i represents the RTT value of P_i , BWT_i is the path bandwidth of P_i , and $LOSS_i$ is the packet loss rate of P_i . In addition, α , β , and θ are all positive constants less than 1 determined by the weight of each parameter. For example, if the RTT of each path affects the path quality more than the other two parameters, α will be accounted for the largest proportion compared with β and θ . The following gives a method to calculate the value of α , β , and θ in this paper.

$$\widehat{X}_t = \frac{X_t - X_{\min}}{X_{\max} - X_{\min}}, \quad X \in \{RTT, BWT, LOSS\}, \quad (10)$$

$$\varepsilon_X = \frac{\sigma_{\widehat{X}}}{\mu_{\widehat{X}}}. \quad (11)$$

We introduce the definition of fluctuation intensity, which reflects the influence coefficient of a path parameter. If a path parameter has a big fluctuation intensity, it means this path parameter affects the path greatly. In this paper, we calculate the value of α , β , and θ by using the jitter intensity of each path parameter. Firstly, three parameters are normalized by min-max normalization as shown in formula (10). Then, the standard deviation factor ε is used to indicate the fluctuation intensity as shown in formula (11); σ represents the standard deviation of the parameter and μ is defined as the mean value. Finally, formula (12) shows the calculation method of α , β , and θ .

$$\alpha = \frac{\varepsilon_{RTT}}{(\varepsilon_{RTT} + \varepsilon_{BWT} + \varepsilon_{LOSS})},$$

$$\beta = \frac{\varepsilon_{BWT}}{(\varepsilon_{RTT} + \varepsilon_{BWT} + \varepsilon_{LOSS})},$$

$$\theta = \frac{\varepsilon_{LOSS}}{(\varepsilon_{RTT} + \varepsilon_{BWT} + \varepsilon_{LOSS})}. \quad (12)$$

With the above knowledge, the path metric for each path can be obtained by formula (7). Besides, the offloading controller will constantly monitor and update the path metric for each path. If there are packets needed to be sent, available paths will be scheduled according to the path metric. In the following, the detailed scheduling policy will be presented.

4.2. Path Collaboration Policy. The offloading controller will guide each vehicle user to select suitable paths according to the path management. The information includes the sequence of available paths and the path metric of these paths. The quality of path P_i is calculated based on formula (7). After that, available paths list S can be obtained. We take $S = \{S_1, S_2, \dots, S_j, \dots, S_m\}$ as a group of available paths and sort them by the path metric. The Q_i value of S_i is bigger than S_{i+1} . The better the path quality is, the bigger the path metric value is.

In the HPC mechanism, the parameter τ is proposed as the path selection threshold. On the one hand, paths with $Q_i \geq \tau$ can be treated as available paths. In other words, paths with $Q_i \geq \tau$ will be abandoned temporarily to ensure that packets are able to be sent first via the better paths. On the other hand, Q_i^t is dynamic with the time t . When $|Q_i^t - Q_i^{t-1}| \geq \omega$, the path scheduling list S updates, where the parameter ω works as the switching threshold. If the link quality varies slightly ($|Q_i^t - Q_i^{t-1}| < \omega$), path scheduling sequence will not update. This design is able to minimize the number of link switching and ensure the stability of path scheduling.

```

Initialization: Path array  $S = \phi$ ;
getSD() is the function of calculating standard deviation factor;
getC() is the function of calculating weight coefficient  $\alpha, \beta$  and  $\theta$ ;
getPW() is the function of calculating path metric value  $Q_i$ ;
Sort( $\Omega, Q_i$ ) is to sort the members in  $\Omega$  based on  $Q_i$  in ascending.
void HPCFunction(PATH =  $\{P_1, P_2, \dots, P_i, \dots, P_n, n \geq 2\}$ ){
  for each  $P_i \in \text{PATH}$  do
     $\epsilon_{\text{RTT}} = \text{getSD}(P_i, \text{RTT});$ 
     $\epsilon_{\text{BWT}} = \text{getSD}(P_i, \text{BWT});$ 
     $\epsilon_{\text{Loss}} = \text{getSD}(P_i, \text{LOSS});$ 
     $\alpha = \text{getC}(\epsilon_{\text{RTT}});$ 
     $\beta = \text{getC}(\epsilon_{\text{BWT}});$ 
     $\theta = \text{getC}(\epsilon_{\text{Loss}});$ 
     $Q_i = \text{getPW}(\alpha, \beta, \theta);$ 
    if  $Q_i \geq \tau$ ;
      S.append( $P_i$ );
    end
  end
  S = Sort(S,  $Q_i$ );
  for each  $P_i \in S$  do
    if  $|Q_i^i - Q_i^{i-1}| \geq \omega$ ;
      S = Sort(S,  $Q_i$ );
    end
  end
  deliver packets based on the S sequence.
end
}

```

ALGORITHM 1: Hybrid awareness path collaboration.

It should be noted that τ and ω can be obtained by the experience of a number of experiments. Before the data sending, the path scheduling policy will select the paths according to the order of the array S . The first one in the array will be the best path. Here, an efficient lookup method can be used to match the delivery path [27]. The controller can select a best path based on its integral link quality and can avoid the blind roll polling in Round Robin mechanism.

The specific path scheduling is shown in Algorithm 1.

The above HPC function guides the controller to select the optimal paths to balance the traffic load for vehicle users. Afterwards, we will further consider how to determine the number of offloading nodes and select them in an optimal manner.

5. Graph-Based Source Collaboration Function

Efficient offloading relies on a set of suitable offloading sources that can quickly distribute data content to subscribing nodes via an ad hoc network. An intuitive idea is to choose the nodes with high probability of connecting other users as the offloading sources.

We consider the source collaboration to minimize the number of offloading nodes to serve all vehicular nodes in a targeted region. Therefore, source selection strategy can utilize the idea of the minimum vertex cover set in graph theory. It means that we choose the minimum vertex cover set as the offloading sources to connect all the nodes of the network. In order to solve the offloading source selection

problem by using the minimum vertex cover set, we first need to convert the network topology into a bipartite graph. The process of bipartite graph transformation divides the vertices in a vertex set into two sets of disjoint sets. The division is based on the fact that two vertices with side relations do not exist in one set. Therefore, the network topology can be transformed into a bipartite graph.

To formulate this problem, we let $G = (V, E)$ be an undirected graph. The vertex V can be divided into two disjoint subsets (A, B) and each edge (i, j) in the graph is associated with two vertexes i and j belonging to these two different vertex sets (i in A , j in B). Then, the graph G is transferred as a bipartite graph.

Theorem 1. *It is necessary and sufficient to judge that the undirected graph G is a bipartite graph, that G has at least two vertices, and that all loops have an even length.*

Let G be the bipartite graph $\langle X, E, Y \rangle$. X and Y represent two subsets in G . Both X and Y are not empty, so G has at least two vertices. If C is any one of the loops in G , let $C = (v_0, v_1, v_2, \dots, v_i, v_0 = v_i)$, where v_i must appear in either X or Y . For example:

$$\{v_0, v_2, v_4, \dots, v_l = v_0\} \in X, \quad (13)$$

$$\{v_1, v_3, v_5, \dots, v_{l-1} = v_1\} \in Y. \quad (14)$$

Let all loops of G have an even length, and let G be a connected graph. Without loss of generality, if G is not connected, we can discuss the following conditions for each

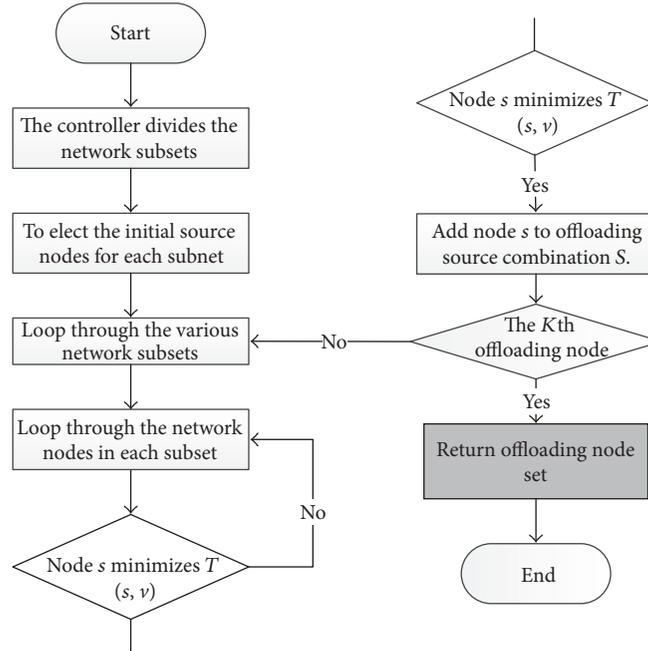


FIGURE 4: Flowchart of source selection.

connected branch of G . Let the vertices of G be V and the set of edges be E , so that X, Y will be constructed and $\langle X, E, Y \rangle = G$. Take $v_0 \in V$ and set $X = \{v \mid v = v_0 \text{ or the path from } v \text{ to } v_0 \text{ have an even length}\}$. Since $Y = V - X$, X is obviously not empty. Now we should prove Y is not empty, and neither end of the two edges is in X or in Y . Since $|V| \geq 2$ and G is a connected graph, v_0 must have adjacent vertices; take v_1 as an example; then $v_1 \in Y$; therefore Y is not empty.

On the other hand, suppose that there are edges (u, v) , $u \in X$, $v \in Y$. Then, the path from v_0 to u has an even length, or $u = v_0$; otherwise, the path from v_0 to v has an even length, or $v = v_0$. In either case, there always is an odd length closed path from v_0 to v_0 which contradicts with the supposition. Therefore, it is impossible to have edges (u, v) to make both u and v be in X .

Based on the above analysis, once we get the knowledge of a certain network topology, we can consider converting the network topology into a bipartite graph.

If there are multiple offloading sources to assist the transmission of data content, the vertex cover problem is to find a vertex cover of minimum size in a given undirected graph. This problem is an NP-complete decision problem. It may be very difficult to find an optimal vertex cover in a graph G . Therefore, we can use a heuristic algorithm to find a near-optimal vertex cover, which is the set of collaborative sources. This process is conducted in a software-defined controller.

Figure 4 introduces the flow chart of source selection, which mainly includes four steps in a controller:

(1) At the starting time $t = 0$, the controller determines the connectivity of each network node according to the collected network node information and divides the entire network into multiple subsets.

(2) According to the subdivision result, the controller selects the initial source randomly in each subset. The initial

source nodes are characterized by the fact that each node in the subset can communicate with the source.

(3) There may be many different initial source nodes according to the former steps. In this case, an optimal offloading source needs to be selected in each subset according to the restricted condition. The optimal offloading source should guarantee the delivery of data from s to v in a short time.

(4) Repeat step (3) until the selected offloading source nodes keep the same as the last iteration, and exit the program. In this case, the total offloading source number should be a fixed value K .

6. Simulation Results

We conduct a group of simulations to verify the feasibility and efficiency of SDCO under the NS-3 open source network simulator. In our experiments, we used the Routes Mobility Model to simulate the vehicular networks. The Routes Mobility Model can generate realistic mobility traces by querying the Google maps directions API. The information, obtained from the Google maps services, allows the simulator to generate realistic mobility traces based on real-world locations and road networks.

We assess the performance of the proposed SDCO solution by comparing with the state-of-the-art solutions, the typical (TYP) offloading mechanism, and the RSRP/RSRQ offloading mechanism [28]. The offloading sources selected are all user nodes, which are endpoint nodes in this experiment. We compare the offloading performance of the three solutions in terms of the offloading source number and the offloading efficiency. The experimental results are shown in Figures 5, 6, and 7.

As shown in Figure 5, the number of offloading sources is obviously reduced compared with the TYP mechanism. The

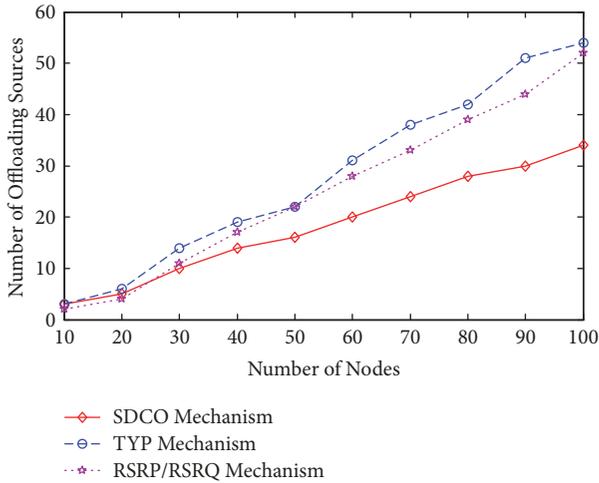


FIGURE 5: Number of offloading sources analysis.

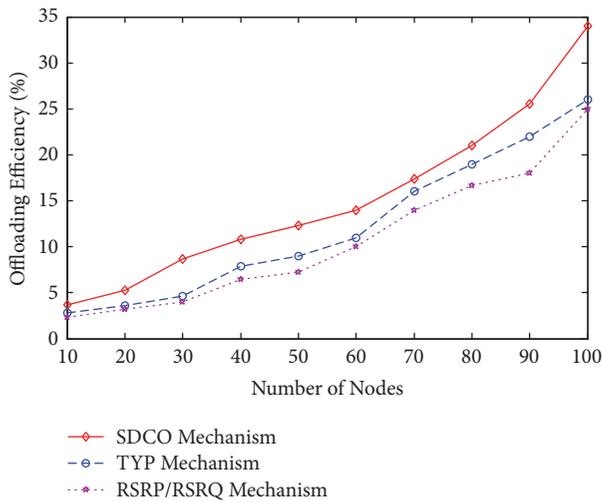


FIGURE 6: Offloading efficiency analysis (10-100).

number of offloading sources for SDCO is always the lowest, and TYP is always the highest. The RSRP/RSRQ mechanism has a moderate performance and it is a bit better than the one of TYP. In this figure, when the number of vehicle nodes is less than or equal to 10, the offloading sources of TYP and SDCO are almost the same. When the number of vehicle nodes increases more than 10, the number of offloading sources in SDCO becomes significantly different from the other two mechanisms. That is because SDCO adopts the GSC method to guarantee the delivery quality with the least offloading sources.

Figures 6 and 7 show the offloading efficiency of the three offloading mechanisms with a different number of vehicle nodes. Offloading efficiency is the ratio of offloaded traffic to the total traffic. In Figure 6, we can see the overall trend that the offloading efficiency rises as the number of vehicle nodes increases in all three offloading mechanisms. When the number of nodes is less than 100, the difference between TYP and RSRP/RSRQ is not obvious, but SDCO is

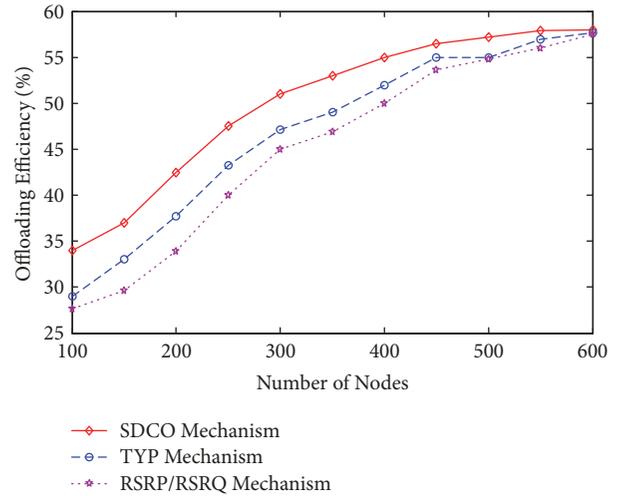


FIGURE 7: Offloading efficiency analysis (100-600).

much better than both of them. When the number of nodes is more than 100, the advantages of SDCO become more prominent. Compared with the RSRP/RSRQ mechanism, SDCO achieves an average improvement by 18% on the offloading efficiency. The reason is that the TYP mechanism transfers more vehicle node requests to the relay nodes (wireless access points) than the RSRP/RSRQ mechanism. The SDCO mechanism can promote the communications between nodes and the communications between the vehicle nodes and the offloading sources.

When the number of nodes adds to 600, the offloading efficiency keeps at 57% in all three mechanisms. Figure 7 shows there is a saturation status for the offloading efficiency, where the three offloading mechanisms have little difference. That is because there are too many nodes, which result in a high density for the network. Therefore, any one of the three solutions can work well for offloading, and the collaborative operations play a small effect on the performance. It is noted that when some traffic is offloaded from the base station to other relay nodes, the base station will have more resource to process the high-priority requests to further improve the experience of users.

7. Conclusion and Future Work

This paper has proposed a software-defined collaborative offloading solution for heterogeneous vehicular networks, named SDCO. Specifically, SDCO adopts a centralized software-defined controller to globally manage offloading policies to vehicular nodes. In addition, the hybrid awareness path collaboration and the graph-based source collaboration are further introduced to optimize the transmission path and offloading sources selection. Simulation results have been provided to demonstrate that SDCO can achieve better improvement in terms of number of offloading sources and offloading efficiency. In the future works, more efficient offloading control functions should be designed based on the collaboration of vehicular nodes, such as the social-based collaboration.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) (nos. 61602030, 61702439, and 91638204), the National Key R&D Program (no. 2016YFE0122900), the Fundamental Research Funds for the Central Universities of China (no. 2016RC036), Shandong Provincial Natural Science Foundation, China (no. ZR2017BF018), the China Scholarship Council, and Natural Sciences and Engineering Research Council (NSERC), Canada.

References

- [1] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, "Heterogeneous vehicular networking: a survey on architecture, challenges, and solutions," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2377–2396, 2015.
- [2] N. Cheng, N. Lu, N. Zhang, X. Zhang, X. S. Shen, and J. W. Mark, "Opportunistic WiFi offloading in vehicular environment: a game-theory approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1944–1955, 2016.
- [3] C. Xu, S. Jia, M. Wang, L. Zhong, H. Zhang, and G.-M. Muntean, "Performance-aware mobile community-based VoD streaming over vehicular Ad Hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 1201–1217, 2015.
- [4] C. Xu, S. Jia, L. Zhong, H. Zhang, and G.-M. Muntean, "Ant-inspired mini-community-based solution for video-on-demand services in wireless mobile networks," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 322–335, 2014.
- [5] H. Zhou, N. Cheng, Q. Yu, X. S. Shen, D. Shan, and F. Bai, "Toward multi-radio vehicular data piping for dynamic DSRC/TVWS spectrum sharing," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 10, pp. 2575–2588, 2016.
- [6] N. Cheng, H. Zhou, L. Lei et al., "Performance analysis of vehicular device-to-device underlay communication," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 5409–5421, 2017.
- [7] H. Zhang, P. Dong, W. Quan, and B. Hu, "Promoting efficient communications for high speed railway using smart collaborative networking," *IEEE Wireless Communications Magazine*, vol. 22, no. 6, pp. 92–97, 2015.
- [8] H. Zhou, N. Cheng, N. Lu et al., "WhiteFi infostation: Engineering vehicular media streaming with geolocation database," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2260–2274, 2016.
- [9] F. Rebecchi, M. Dias de Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, "Data offloading techniques in cellular networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 580–603, 2015.
- [10] G. Mao, Z. Zhang, and B. Anderson, "Cooperative content dissemination and offloading in heterogeneous mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6573–6587, 2016.
- [11] N. Lu, N. Zhang, N. Cheng, X. Shen, J. W. Mark, and F. Bai, "Vehicles meet infrastructure: toward capacity-cost tradeoffs for vehicular access networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1266–1277, 2013.
- [12] Y. Liu, Z. Yang, T. Ning, and H. Wu, "Efficient quality-of-service (QoS) support in mobile opportunistic networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4574–4584, 2014.
- [13] H. Zhang, W. Quan, H.-C. Chao, and C. Qiao, "Smart identifier network: A collaborative architecture for the future internet," *IEEE Network*, vol. 30, no. 3, pp. 46–51, 2016.
- [14] W. Quan, Y. Liu, H. Zhang, and S. Yu, "Enhancing crowd collaborations for software defined vehicular networks," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 80–86, 2017.
- [15] K. Wang, H. Yin, W. Quan, and G. Min, "Enabling collaborative edge computing for software defined vehicular networks," *IEEE Network*, vol. PP, no. 99, pp. 1–6, 2018.
- [16] J. Chen, H. Zhou, N. Zhang et al., "Service-oriented dynamic connection management for software-defined internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2826–2837, 2017.
- [17] K. Zheng, L. Hou, H. Meng, Q. Zheng, N. Lu, and L. Lei, "Soft-defined heterogeneous vehicular network: architecture and challenges," *IEEE Network*, vol. 30, no. 4, pp. 72–80, 2016.
- [18] P. Dong, X. Du, J. Sun, and H. Zhang, "Energy-efficient cluster management in heterogeneous vehicular networks," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 644–649, San Francisco, Calif, USA, April 2016.
- [19] N. Zhang, S. Zhang, P. Yang, O. Alhussain, W. Zhuang, and X. S. Shen, "Software defined space-air-ground integrated vehicular networks: challenges and solutions," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 101–109, 2017.
- [20] Y. Liu, C. Xu, Y. Zhan, Z. Liu, J. Guan, and H. Zhang, "Incentive mechanism for computation offloading using edge computing: a Stackelberg game approach," *Computer Networks*, vol. 129, pp. 399–409, 2017.
- [21] N. Cheng, F. Lyu, J. Chen et al., "Big data driven vehicular networks," *IEEE Network*, vol. PP, no. 99, pp. 1–8, 2018.
- [22] A. Aijaz, H. Aghvami, and M. Amani, "A survey on mobile data offloading: technical and business perspectives," *IEEE Wireless Communications Magazine*, vol. 20, no. 2, pp. 104–112, 2013.
- [23] X. Zhuo, W. Gao, G. Cao, and S. Hua, "An incentive framework for cellular traffic offloading," *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 541–555, 2014.
- [24] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: how much can wifi deliver?" *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 536–550, 2013.
- [25] S. Singh, H. S. Dhillon, and J. G. Andrews, "Offloading in heterogeneous networks: modeling, analysis, and design insights," *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 2484–2497, 2013.
- [26] N. Cheng, N. Lu, N. Zhang, X. S. Shen, and J. W. Mark, "Opportunistic WiFi offloading in vehicular environment: A queuing analysis," in *Proceedings of the 2014 IEEE Global Communications Conference, GLOBECOM 2014*, pp. 211–216, Austin, Tex, USA, December 2014.
- [27] W. Quan, C. Xu, J. Guan, H. Zhang, and L. A. Grieco, "Scalable name lookup with adaptive prefix bloom filter for named data networking," *IEEE Communications Letters*, vol. 18, no. 1, pp. 102–105, 2014.
- [28] L. Al-Kanj, H. V. Poor, and Z. Dawy, "Optimal cellular offloading via device-to-device communication networks with fairness constraints," *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4628–4643, 2014.



Hindawi

Submit your manuscripts at
www.hindawi.com

