Cost-efficient Sensory Data Transmission in Heterogeneous Software Defined Vehicular Networks

Zongjian He, *Member, IEEE,* Daqiang Zhang, *Senior Member, IEEE,* and Junbin Liang, *Member, IEEE*

***Abstract*—Sensing and networking have been regarded as key enabling technologies of future smart vehicles. Sensing allows vehicles to be context awareness, while networking empowers context sharing among ambients. Existing vehicular communication solutions mainly rely on homogeneous network, or heterogeneous network via data offloading. However, today's vehicular network implementations are highly heterogeneous. Therefore, conventional homogeneous communication and data offloading may not be able to satisfy the requirement of the emerging vehicular networking applications. In this research, we apply software defined network (SDN) to the heterogeneous vehicular networks to bridge the gaps. With SDN, heterogeneous network resources can be managed with a unified abstraction. Moreover, we propose an SDN-based wireless communication solution, which can schedule different network resources to minimize communication cost. We investigate the problems in both single and multiple hop cases. We also evaluate the proposed approaches using traffic traces. The effectiveness and efficiency are validated by the results.**

***Index Terms*—Handoff, VANET, Software Defined Network, SDN, optimization**

I. Introduction

With the advancement of information and communication technology, more and more things in people's daily lives are becoming smart. Smart phones, smart TVs and smart watches are growing rapidly and have significantly changed people's life style. Meanwhile, vehicles have been considered as the next impactful smart thing that can also potentially exert influence on everyone [1]. To enable a smart vehicle, context awareness through sensors and context sharing via networks are both essential. Today, modern vehicles have already been equipped with more than 100 sensors, and their sensory parameters, ranging from fuel consumptions to engine temperatures, can be obtained via the onboard diagnostic (OBD) interface [2]. Besides onboard sensors, vehicles can also leverage the sensors from smartphones as drivers are very likely to connect their phones to the vehicle while driving. Moreover, researchers have validated that by sharing sensory data among vehicles via vehicular networks can remarkably improve the overall sensing performance [3]. Therefore, vehicular networks are fundamentally important for sensory data transmission and have attracted much attention from both industry and academy in recent years.

Vehicular network aims to connect vehicles to the cloud, roadside units (RSU), and ambient vehicles using wireless communication technologies. Compared with other types of wireless networks, smart vehicle based wireless networks are more complicated due to the heterogeneity. Currently, many vehicles are able to access Internet via cellular networks like GPRS / EDGE / UMTS / LTE. Meanwhile, IEEE has released standard 802.11p [4] for vehicular ad-hoc network (VANET) that can establish vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) connections using dedicated spectrum. In addition, other wireless technologies like IEEE 802.11 ac/ad [5], ZigBee [6], Bluetooth [7], RFID [8], and WiMAX [9] have also been applied as alternative technologies. Device-to-device (D2D) communication is very promising to be included in the future 5G networks [10] [11], which will further increase the heterogeneity of vehicular networks. What is more, it is quite often for a vehicle to equip multiple wireless interfaces onboard simultaneously (e.g., Cellular network and RFID).

Generally speaking, there are two paradigms of vehicular sensory data transmission: single hop and multiple hops. Single hop transmission directly passes the sensory data to ambient vehicles, road side units, or the cloud. Such examples include vehicle speed/location sensing, emission sensing, etc. Multi-hop transmission is usually used in distributed applications, such as distributed traffic speed estimation and crowd sensed data collection [12]. Therefore, both of them should be well treated.

Data transmission over such a heterogeneous network poses a number of challenges that need to be properly considered. First, the heterogeneity of these wireless technologies makes the interconnection and interoperation an intractable task. As it can lead to network fragmentation and inefficiency of network resource utilization. Moreover, different wireless technologies have different performance (e.g., throughput, bandwidth, or latency) and communication cost, and vehicular networking applications also have a variety of requirements for network resources. Arbitrarily selecting one or multiple network interfaces for communication may not be able to satisfy the requirements of an application. Instead, network resources must be allocated carefully. In summary, two problems are caused by the heterogeneity. The first is interoperability, while the second is resource utilization.

Researchers have already proposed many vehicular sensing applications, ranging from driving safety [13] to vehicular

Zongjian He and Daqiang Zhang are with School of Software Engineering, Tongji University, Shanghai, 201804, China. (E-mail: {hezongjian, dqzhang}@tongji.edu.cn)

Junbin Liang are with School of Computer and Electronic Information, Guangxi University, Nanning, Guangxi, 530004, China. (E-mail: liangjb@gxu.edu.cn)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2562699, IEEE Sensors Journal

crowdsourcing [14], and most of them only rely on a single type of wireless network. For example, safety applications usually utilize IEEE 802.11p for V2V communications, and infotainment applications often access the Internet via cellular networks. In order to improve the network performance and reduce the communication cost, mobile data offloading [15] has been employed. In these scenarios, WLAN or other metro-scale networks are generally used as a complementary network to deliver data that are originally transmitted via cellular networks [16] [17]. Additionally, handoff among homogeneous networks [18] and heterogeneous networks [19] [20] have also been proposed to improve the network availability. These researches have remarkably thrived the evolution of vehicular networks. However, imperfections remain for vehicular network to support a wide range of real world applications. Foremost, statically assigned data offloading and handoff are not flexible enough, especially when the vehicles are moving fast and the external environment changes rapidly. Moreover, current approaches also lack of unified abstractions. Consequently, these existing approaches may not be able to fully leverage the underlying network resources.

Software defined networks (SDN) is a novel network paradigm in which the control and data plane are decoupled and network resources are managed by a logically centralized controller. Furthermore, devices from various vendors can communicate with each others via a standardized interface(E.g., OpenFlow). Therefore, it significantly simplifies the network management and offers a programmable and flexible network architecture. In addition, the model of SDN is generic enough to be applied to both wired and wireless networks. Extensive studies have been conducted in applying SDN to wireless networks [21]. With the benefits brought by SDN, we firmly believe that SDN is promising to bridge the gaps between applications' demands and today's limitations in vehicular networks.

In this research, we propose to empower vehicular networks by utilizing SDN as the unified resource manager. We name the paradigm *Software Defined Vehicular Networks*, or *SDVN*. To demonstrate the advantages of SDVN, we design a novel network resource scheduling solution on top of SDVN. In this solution, we perform centralized scheduling of all network resources in the control plane of SDVN. Whenever an application wishes to transmit data, SDVN can adaptively choose the optimal network interfaces from all available candidates. Our algorithm ensures that the communication cost is minimized while the requirements of applications can still be satisfied. Moreover, it applies to both single hop and multi-hop transmission scenarios.

To the best of our knowledge, we are the first to study heterogeneous vehicular communications under software defined networks. The contributions of this paper can be summarized as follows:

- We propose a novel data transmission method for heterogeneous vehicular networks, based on the SDN architecture.
- We present two mathematical models to formally represent the dynamic nature of software defined vehicular networks. One for single hop and the other for multiple hop communication.
- We study the problems of both single and multiple hop unicast under SDVN. The problems are rigorously formulated, and two different solutions are proposed.
- We conduct extensive evaluations using real traffic data, and the results validate our solution.

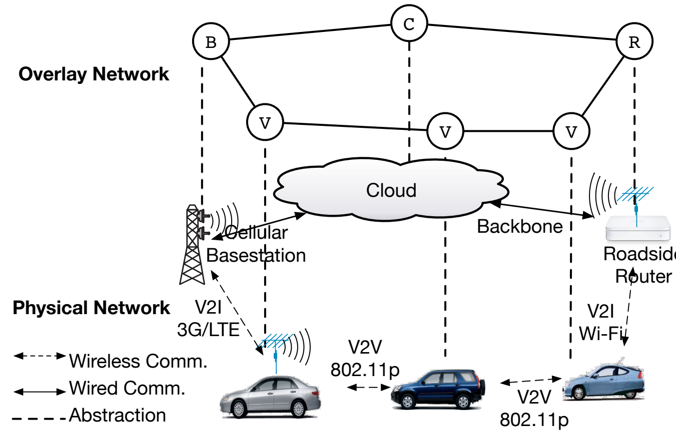


Fig. 1. Heterogeneous vehicular communication

The reminder of this paper is organized as follows: In Section II, we present the system model of SDN based vehicular network, and the mathematical tools that can be used to model SDVN. Section III and Section IV formulate the cost efficient network scheduling problem formally, and describes different solutions for the problem. After that, Section V evaluates the proposed solutions and the insights of the results are explained. Section VI summarizes the related researches on wireless communication in heterogeneous vehicular network, and software defined wireless networks. Finally, Section VII concludes this paper.

II. System Model

A. *Software Defined Vehicular Networks*

In real vehicular networking applications, it is very common that a vehicle can have multiple network interfaces onboard simultaneously. These interfaces can be WiFi, DSRC, UMTS, WiMAX, Bluetooth, and etc. A typical example of such scenario is depicted in Fig. 1. To apply SDN to such a heterogeneous network, the first step is to build a unified abstraction on top of them. In this work, we treat all network devices equally as abstracted nodes in an overlay network, this abstraction is also illustrated in Fig. 1.

To apply SDN, it is also important to decouple the control and data plane, which are originally both implemented by the dedicated onboard hardware. In this research, the control plane is designed to be located at the cloud side. It is responsible for making centralized decisions of all data packets forwarding by installing corresponding flow table entries at data plane elements. The data plane elements then forward data packets according to the flow table. The data plane elements include all vehicles and the corresponding infrastructures, such as

RSUs and base stations. Fig. 2 illustrates the logical system architecture.

However, directly apply SDN to vehicular networks entails several challenges. The key difficulty is how to mitigate the management overhead introduced by the control plane and data plane communication. In traditional wired SDN, data are forwarded through guided media such as optical fibers. Therefore, the topologies seldom change and the overhead for the control plane to maintain the topology information is relatively low. However, in vehicular network, the nodes are composed of highly dynamic vehicles. The cost of tracking the positions and connectivities of all vehicles is unacceptably high.

The connectivity between data plane and control plane is also vital for this architecture. Since not all data plane elements can be directly connected to the control plane with stable wired network connections. In this architecture, both RSUs and the base stations can reach the control plane via wired networks, which is quite similar to the deployment of SDN in wired environment. We call these parts *stationary data plane*. The *mobile data plane*, which is composed of highly dynamic vehicles, should be specially treated. Vehicles can reach the control plane through cellular networks, ambient RSUs, or multi-hop relay, and different wireless media could lead to different level of overhead and reachability. In this research, we adopt a trajectory prediction scheme to reduce the communication overhead and handle the frequent disconnection.

The predictability of trajectory is a unique feature of vehicles [22]. With predictable trajectory, it is possible to estimate the location of a vehicle in the near future. Moreover, we can also predict the future connectivity of different wireless interfaces, given the location of the vehicles, RSUs and the corresponding communication ranges. Therefore, the network availability in the near future can also be regarded as an input in this research. The control plane does not need to monitor the vehicles with a high frequency. Fortunately, researchers have developed both macroscopic and microscopic models for trajectory prediction [23], which can be directly employed in SDVN to reduce the overhead of control plane.

With the aforementioned preliminary knowledges, the software defined vehicular networks can be defined as:

Definition 1: Software Defined Vehicular Network (SDVN): Software defined vehicular network is a vehicular network paradigm whose data forwarding decisions are controlled by the logically centralized control plane. It heavily utilizes vehicle trajectory predictions for network resource management.

B. SDVN Modeling

To precisely describe the SDVN, we need to develop rigorous mathematical models that can reflect the dynamic behavior of the network topology over time.

For single hop communications, given a fixed number of source and destination nodes, the connectivity between two nodes and the bandwidth are not fixed at different time period. Therefore, we can use a matrix to denote the network availability.

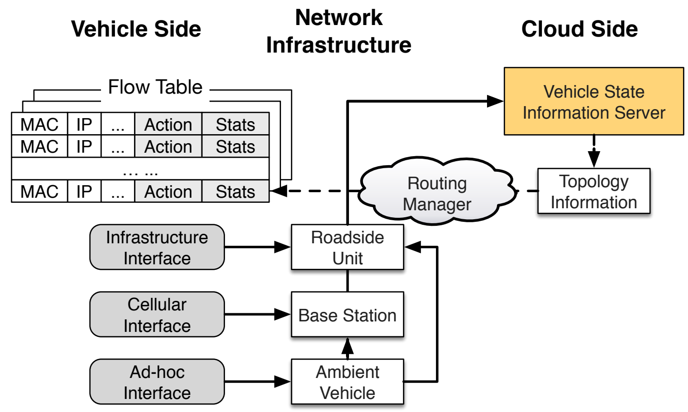


Fig. 2. SDVN enabled network architecture

Definition 2: Modeling of single hop communication in SDVN:

Single hop communication in SDVN can be modeled using a network availability matrix $\mathbb{P}^{(n \times t)}$, where n is the number of network interfaces and t is the number of time periods in trajectory prediction. We use $p_{ij} \in \mathbb{P}^{(n \times t)}$ to denote the bandwidth that network i can provide at time j. If $p_{ij} = 0$, it means the network i is not available at time j.

For multi-hop communication, we must consider the connectivity of all intermediate nodes. In this case, the previous matrix based modeling can hardly be used, because there should be a network availability matrix for every pair of nodes, which result in a high dimensional matrix. Therefore, we need to employ a simpler mathematical model. In this research, we utilize dynamic graph to model multi-hop communications.

Definition 3: Modeling of multi-hop communication in SDVN:

Multi-hop communication in SDVN can be modeled by a *time dependent graph*, whose edges change with time. A time dependent graph can be denoted as a 4-tuple $\mathcal{G} = (V, E, \mathcal{T}, \mathscr{P})$, where

- V is a set of vertices, and the number of vertices is $||V|| = m$;
- $E \subseteq \{V \times V\}$ is a set of connection between vertices;
- $\mathcal{T} = [t_1, \cdots t_t] \in \mathbb{N}^+$ is a set of consecutive equal-length time periods, or *lifetime* of the graph. The period length of a single time period is denoted as l;
- $\mathscr{P} \in E \times \mathcal{T} \to \{0, 1\}$ is the *weighted presence function* of a specific edge, where $\mathscr{P}(e, t) = x$ represents the weight of edge e is x.

In SDVN, the network elements, such as RSU, vehicles, etc can all be represented by vertices in a time dependent graph $\mathcal{G}$. The communication links are modeled as edges in $\mathcal{G}$. The weight in weighted presence function $\mathscr{P}$ can be defined according to different application requirements. It can be network bandwidth, wireless spectrum, or a combination of them. The communication cost of edges can be denoted as $C(e)$. It can be the cost of a physical link, or a combination of multiple links.

The notations and symbols in this paper are summarized in Table I.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2562699, IEEE Sensors Journal

TABLE I
SUMMARY OF NOTATIONS

Symbols	Definition
$\mathbb{P}^{(n\times t)}$	Network availability matrix for single hop communication.
$p_{ij} \in \mathbb{P}^{(n\times t)}$	The bandwidth of that network i can provide at time j.
$\mathbb{C}^{(n)}$	Vector of unit cost.
$c_i \in \mathbb{C}^{(n)}$	Unit cost of network interface i.
$\mathcal{G}$	Time dependent graph. $\mathcal{G}$ is a 4-tuple $(V, E, \mathcal{T}, \mathscr{P})$.
m	$m = \|\|V\|\|$, the number of vertices in the graph.
n	$n = \|\|E\|\|$, the number of edges in the graph.
t	$t = \|\|\mathcal{T}\|\|$, the number of time periods in time dependent graph.
$\mathscr{P}(e,t)$	Weighted presence function.
b	The bandwidth requirement of an application.
$\mathbb{S}^{(n\times t)}$	Handoff schedule matrix for single hop communication.
s, d	Multi-hop routing source and destination nodes.
$C(e)$	Communication cost of edge e.
F^I, F^S	An inseparable or a separable flow.
$BW(F)$	Total bandwidth of a flow, no matter separable or inseparable.
$PF(F)$	The profit of a path.

(a) Network Availability Matrix

	t1	t2	t3	t4	t5
WiFi	0	6	6	0	0
Cellular	5	5	5	5	5
DSRC	2	0	0	0	2

(b) Network Cost Vector

	WiFi	Cellular	DSRC
Cost	5	10	1

(c) Scheduling Solution Example

	t1	t2	t3	t4	t5	Cost	Data
S1	C	C	C	C	C	50	25
S2	D	W	W	C	D	22	21
S3	C	W	W	C	C	40	27
S4	CD	WC	WC	C	CD	62	41

Fig. 3. An instance of the problem and its sample solutions. (a) The predicted bandwidth a wireless interface can provide in the future. (b) The unit cost of a wireless interface. (c) Possible scheduling solutions and the corresponding cost and data volume. C stands for cellular, D for DSRC and W for WiFi.

III. Single Hop Data Transmission

We first discuss the case of single hop data transmission. From previous discussion, we know that the heterogeneity of networks also brings a number of opportunities to transmit similar amount of data with lower network cost. E.g., cellular networks usually cover a wide range of area and can provide satisfactory connection speed via 4G communication. However, the communication cost is relatively higher. In contrast, the speed and coverage range of DSRC is quite limited. But the cost can almost be neglected. Moreover, it is also the common case that a vehicle may choose to use multiple network interfaces simultaneously. E.g., use WiFi and UMTS to transmit / receive data at the same time. This approach can further increase the overall bandwidth.

To ensure the quality of service, applications may have minimum bandwidth requirement. In down-link communications, it can be stream data rate. In up-link, it can be sensor sampling frequency. When scheduling data transmission among heterogeneous networks, the bandwidth requirement must be satisfied. Therefore, in this research, we set the bandwidth requirement as a hard constraint and the overall cost to be the objective function we want to minimized.

A. A Motivating Example

To have a better understanding of the problem we study, Fig. 3 demonstrates a running example of it. Suppose there are three wireless interfaces: WiFi, cellular network and DSRC, and their communication costs per unit time are also given in Fig. 3(b). With the help of trajectory prediction, we can estimate the availability of network, and the bandwidth they can provide in a short period of time in the future, as shown in Fig. 3(a) (0 means the network is unavailable). We also know that the application requires to transmit 25 data volume from time period $t1$ to $t5$.

With these information as given, we can have multiple candidate solutions of how to schedule the network usage. Three of these scheduling policies are listed in Fig. 3(c). The corresponding total communication costs and maximum data volume are also listed in the table. S1 is the simplest solution, there is no handoff and the application uses cellular networks at all time. $S2$ uses all three types of networks, and $S3$ only uses cellular and WiFi. Take $S2$ for example, the total cost is calculated by summing up the unit cost of selected network interfaces: 1 + 5 + 5 + 10 + 1 = 22, and the maximum data volume is calculated by adding the bandwidth together: 2 + 6 + 6 + 5 + 2 = 21. We can see that $S2$ has the minimum cost among the three solutions, but the maximum data capacity does not satisfy the application requirement, which is 25. Therefore, we have to consider $S1$ and $S3$, and we discover that the cost of $S3$ is smaller than $S1$. Hence, in this example, $S3$ is the optimal among the three solutions. Then, the SDVN controller can update the flow table on the vehicle to control the data forwarding according to solution $S2$.

However, if the application requires to transmit a data volume of 40 during that time period, selecting a single interface at a specific time is impossible to satisfy the requirement. Instead, multiple interfaces must be used. Solution $S4$ demonstrates such case: in $t1$ and $t5$, both cellular and DSRC are scheduled for data transmission. Similarly for $t2$ and $t3$, when both WiFi and cellular networks are used. By doing this, the data volume transmitted can reach up to 41, which is slightly higher than the required data volume 40.

B. Problem Formulation

With the preliminary knowledges and the running example, now we can formulate the problem formally, as follows:

Given:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2562699, IEEE Sensors Journal

- n network interfaces and the corresponding cost vector $\mathbb{C}^{(n)}$, $c_i \in \mathbb{C}^{(n)}$ denotes the unit cost of the network i.
- Network availability matrix $\mathbb{P}^{(n\times t)}$. $p_{ij} \in \mathbb{P}^{(n\times t)}$ means the bandwidth network i can provide at time t.
- The bandwidth requirement of an application: b.

Objective:
Find a handoff schedule matrix $\mathbb{S}^{(n\times t)}$, and $s_{ij} \in \mathbb{S}^{(n\times t)}$.

$$s_{ij} = \begin{cases} 1 & \text{network } i \text{ is selected at time } j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The objective is to:

$$\min \sum_{i,j}^{\forall s_{ij}\in\mathbb{S}} c_i s_{ij}, \tag{2}$$

Subject To:
Bandwidth constraint:

$$\sum_{i,j}^{\forall s_{ij}\in\mathbb{S}} s_{ij}p_{ij} \geq b \tag{3}$$

In this problem formulation, Equation 2 is the objective function, which calculates the overall network communication cost for the scheduling. Equation 3 is the only constraint, which calculates the maximum network transmission capacity. The constraint ensures that the data transmission volume requirement by the application are satisfied.

In the rest of this section, we introduce two solutions to the problem. They can be applied to different application scenarios. The exhaustive search can be applied to small-scaled applications to find the optimal solution, while the greedy algorithm is more suitable for large-scaled ones. However, the optimality is not entirely ensured.

C. Exhaustive Search

From the above problem formulation, we can find out that the problem is a typical discrete optimization issue. Intuitively, the simplest solution to the problem is exhaustive search. The basic idea is to exhaustively enumerate all the possible combinations for all networks interfaces in all the time periods. The idea of this algorithm is simple and easy to be implemented. However, the computational complexity is considerably high. Suppose there are n network interfaces and t time slots. The complexity of this algorithm can be calculated by:

$$\sum_{i=0}^{i\leq n} \binom{n}{i} \times t = 2^n \times t \tag{4}$$

From Equation 4, we can see that the computational complexity of the exhaustive search algorithm is $O(2^n t)$, which is not polynomial time solvable. In real world applications, if the number of available network interfaces becomes very large, the computation time consumption may not be affordable. Therefore, this solution is only applicable for very small-scaled network, and acts as a baseline. We have to find an alternative solution for the problem. In this work, we apply a polynomial time greedy approximation scheme.

Algorithm 1 The Greedy Algorithm

1: **procedure** GREEDY($\mathbb{P}$,$\mathbb{C}$,b)
2:     **for all** $s_{ij} \in \mathbb{S}$ **do**
3:         **if** $p_{ij} \neq 0$ **then**
4:             $s_{ij} \leftarrow 1$;
5:         **else**
6:             $s_{ij} \leftarrow 0$;
7:         **end if**
8:     **end for**
9:     **loop**
10:         $(x, y) = \arg\max \frac{c_x}{s_{xy}}$;
11:         **if** $\sum_{i,j}^{\forall s_{ij}\in\mathbb{S}} s_{ij}p_{ij} - c_x \leq b$ **then**
12:             **return** $\mathbb{S}$
13:         **end if**
14:         $s_{xy} \leftarrow 0$;
15:     **end loop**
16: **end procedure**

D. The Greedy Algorithm

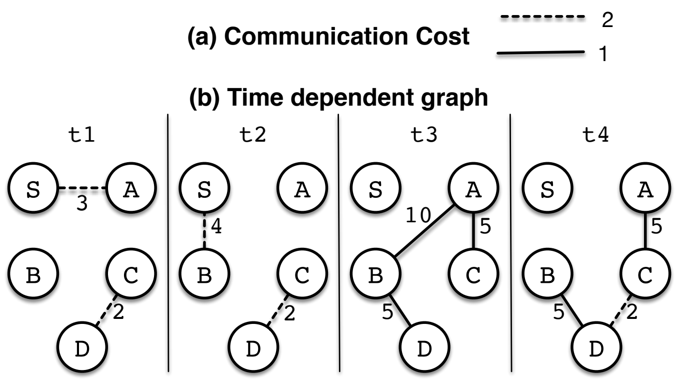
When designing a greedy approximation algorithm, we have to sacrifice the optimality for complexity in many cases. For the first step, we start off with an initial case and iteratively revise the solution according to a certain metric.

In this example, we observe that if all available network interfaces have been selected, the maximum data throughput can be achieved. Meanwhile, the total communication costs is also maximal. If the maximum data throughput can still not satisfy the constraint, we can safely claim that there is no solution for this instance of the problem. In other cases, in order to reduce the cost, we can remove some selected interfaces gradually one by one, until the bandwidth requirement is only slightly more than the given constraint. The observation can be utilized to design a greedy algorithm. To complete the design, another question is which interface to remove in every iteration. Obviously, we have to consider both the unit cost and the bandwidth of the interface. Therefore, we choose to remove the interface with the maximum cost and bandwidth ratio.

The complete algorithm is listed in Algorithm 1. At the very beginning, all the available network interfaces are selected (Line 2 – Line 8). Then, in every iteration, the algorithm finds the interface with the largest cost and bandwidth ratio (Line 10) and remove it (Line 14). If removing the interface will violate the bandwidth constraint, the algorithm will end and return the current selected interfaces (Line 11 – Line 13).

Let us still use the previous example in Fig. 3 to see how the algorithm really works. In this example, the cost and bandwidth ratio of WiFi, DSRC and cellular networks are $\frac{5}{6}$, $\frac{1}{2}$ and 2, respectively. Therefore, the greedy algorithm will try to avoid using cellular networks as much as possible. After all cellular interfaces are removed, it will begin to remove WiFi. However, please be reminded that this is only a simplified case. In real world applications, the cost and bandwidth ratio may be quite complicated and diverse. Even the same network interface may have different ratios in different time period, due to the signal strength ,wireless interferences, and other factors.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2562699, IEEE Sensors Journal



(c) Scheduling Solution Example

	t1	t2	t3	t4	Data	Cost
S1	S,A,3	-	A,C,3	C,D,2	2	5
S2	S,A,3	-	A,B,3	B,D,3	3	4
S3	S,A,3	-	A,C,3	C,D,2	6	8
	-	S,B,4	B,D,4	-		

Fig. 4. An example of multihop data transmission over SDVN. S is source node and D is destination node. (a) The unit communication cost of heterogeneous vehicular networks. (b) The SDWN modeled as a time dependent graph. (c) Some solutions for flow scheduling, the three tuple in each cell indicates the source, destination and bandwidth it consumes.

From Algorithm 1, we can see that the complexity of this algorithm is $O(n^2t^2)$, which is polynomial time. Therefore, the computational resource required by this algorithm is much smaller than the exhaustive search one. However, the optimality can not be fully ensured.

IV. Multi-hop Data Transmission

In many cases, data transmission in vehicular networks involves multi-hop communication. Such examples includes sensory data fusion, and safety warning propagations. Conventionally, multi-hop data transmission protocols in vehicular networks are generally distributed, due to the lack of centralized coordinator. With the help of SDVN, it is now possible to schedule the network forwarding in a centralized manner.

A. A Motivating Example

Fig. 4 depicts an example of multi-hop data transmission over SDWN. Suppose there are two types of wireless interfaces involved in the communication (represented in solid and dashed lines), and their unit communication costs are shown in Fig. 4(a). There are five nodes in the networks in total, and based on the trajectory prediction, their network connectivity and the corresponding bandwidth in different time periods can also be illustrated in Fig. 4(b), which is an instance of time dependent graph. Then, suppose an application wants to send 3 units of data from source node S to destination D. If we only look at the first time period $t1$, it seems to be an impossible task. However, we know that the topology of vehicular networks is highly dynamic. With the trajectory prediction, we can infer the future network connectivity in a short time period. By looking at a longer time period ($t1$ - $t4$), there are multiple possible solutions of how to schedule the network flow.

Three of the sample solutions are listed in Fig. 4(c). In $S1$, packets are sent from S to A with bandwidth 3. So the maximum amount of data transmitted to A is 3. In $t2$, A is disconnected with other nodes, so no operation is conducted. In $t3$, all the three units of data are forwarded from A to C. Notice that in $t4$, the bandwidth between C and D is only 2. Although there are 3 units of data stored at C, only 2 of them can be finally passed to D, due to the bandwidth limitation. Therefore, the total amount of data transmitted for $S1$ is 2, and the cost is the sum of three transmission: 2 + 1 + 2 = 5. Similarly, scheduling according to $S2$ can transmit 3 units of data with cost 4. If application demands to transmit at least 5 units of data, only $S3$ can satisfy the requirement, with a total cost of 8.

From the aforementioned examples, several observations can be found: 1) Solution $S1$ and $S2$ only use a single routing path, and the data flow is *inseparable*. In this case, given an inseparable flow routing path, the amount of data transmitted is bottlenecked by the hop with the minimum bandwidth. 2) To fully utilize the network resources, it is necessary to use multiple path routing simultaneously, just like solution $S3$. We call this scenario as *separable flow*. In this paper, we will discuss both cases. As to implementation concern, separable flow is only supported after OpenFlow 1.1 with group tables. Unfortunately, most of the SDN controllers, including the one we use, POX [24], only support OpenFlow 1.0 standard [25].

In addition, we further assume that the storage capacity of a node is much larger than the network bandwidth, so that the requirement of storing a certain amount of data at a specific node can always be satisfied. This assumption makes sense because the onboard computers of vehicles are usually equipped with a large volume of external storage to store digital maps and multimedia files.

B. Problem Formulation

Before we formally formulate the multi-hop routing problem, we need to formally define separable and inseparable flow, respectively.

Definition 4: Inseparable Flow:

An inseparable flow from node s to node d in a time dependent graph $\mathcal{G} = (V, E, \mathcal{T}, \mathscr{P})$ can be defined as $F^I(V^I, E^I) = (s_1, r_1, w_1)^{t1} \cup, \cdots, \cup (s_n, r_n, w_n)^{tn}$, where s_i and r_i denote the sender and receiver of each time period, w_i is the bandwidth it consumes.

It must satisfy the following constraints:

1) $(s_1 = s) \wedge (r_n = d)$;
2) $(V^I \subseteq V) \wedge (E^I \subseteq E)$;
3) $\mathscr{P}(s_k, r_k, t_k) \geq w_k \geq 1$;
4) $\forall_{k \geq 2}, (s_k \in \{r_1, \cdots, r_{k-1}\}) \wedge w_k \leq \min\{b_1, ..., b_{k-1}\}$.

Constraint 1) reveals the requirements of the source and destination nodes; Constraint 2) exhibits that the flow must be derived from the given time dependent graph without adding additional vertices or edges; Constraint 3) ensures the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2562699, IEEE Sensors Journal

existence of the link, and the bandwidth we use must be no more than that the link could provide; Constraint 4) implies that the senders of each time period must have received the previously forwarded message before, and the amount of data sent must be no more than that they have received.

As we have discussed before, the total bandwidth of an inseparable flow is bottlenecked by the link with minimum bandwidth. Therefore, the bandwidth can be calculated as:

$$BW(F^I) = \min_{\forall e \in E_T} (e). \tag{5}$$

Definition 5: Separable Flow:

A separable flow from node s to node d in a time dependent graph $\mathcal{G} = (V, E, \mathcal{T}, \mathscr{P})$ is a combination of multiple inseparable flows. It can be denoted as $F^S(V^S, E^S) = F_1^I \cup F_2^I ... \cup F_n^I$, which satisfy the following constraints:

1) $\sum_{\forall t \in \mathcal{T}} w_i \leq \mathscr{P}(s_k, r_k, t_k)$;
2) $\sum_{\forall j \geq 2} w_j \leq \sum_{\forall k < j, r_k = s_j} w_k$.

In the definition, Constraint 1) reveals that the amount of data sent must not exceed the total bandwidth that a link could provide; Constraint 2) exhibits that the amount of data sent must be no more than that it has received.

The total bandwidth of a separable flow can be calculated by summing up all the bandwidth of the inseparable flows it contain:

$$BW(F^S) = \sum_{\forall F_i^I \in F^S} BW(F_i^I). \tag{6}$$

With the preliminary knowledges and the running example, now we can formulate the SDVN multi-hop routing scheduling problem formally, as follows:

Given:

1) SDVN and the predicted vehicle trajectory represented by a time dependent graph $\mathcal{G} = (V, E, \mathcal{T}, \mathscr{P})$, the weight of $\mathscr{P}$ represents the bandwidth of the link;
2) $s \in V$ is the multi-hop routing source node; $d \in V$ is the multi-hop routing destination.
3) $C(e)$ is the communication cost of a specific link.

Objective:

Find an inseparable flow F^I or separable flow F^S, such that:

$$\min(\sum_{e \in E_T} C(e)) \tag{7}$$

Subject To:

1) $BW(F) \geq b$.

Equation 7 shows that the total cost is the objective function we want to minimize. The only constraint is the bandwidth of the multi-hop routing required by applications.

C. Solution to Inseparable Flow Routing

We first present the solution to inseparable flow routing problem. Before introducing the detailed design of the solutions, we have to present several essential concepts of time dependent graph, which the solutions are based on.

Algorithm 2 Determine the existence of a time dependent path from s to d in $\mathcal{G}$.

```
function HasPath(𝒢,s,d)
    VisitedNodes ← {s}
    for all t ∈ 𝒯 do
        for all v ∈ VistedNodes do
            for all n ∈ NeighborSet(v) do
                if n ∉ VistedNodes then
                    VistedNodes ← VistedNodes ∪ {n}
                end if
            end for
        end for
    end for
    if d ∈ VisitedNodes then
        return True
    end if
    return False
end function
```

Definition 6: Time Dependent Path (TDP):

Node i and j in time dependent graph are said to be connected, or have a path, if there exists a single hop direct link from i to j, or a sequence of edges $[(v_0, v_1)^{t1}, (v_1, v_2)^{t2}, \cdots, (v_{n-1}, v_n)^{tn}]$, where $v_0 = i$, $v_n = j$, $\mathscr{P}(v_{k-1}, v_k)^{tk} = 1$, $tk \in \mathcal{T}$ and $t1 < t2 < \cdots < tn$.

Differently from paths in static graphs, which satisfy *equivalence relation*, a time dependent path is not an equivalence relation, since it violates both symmetry and transitivity properties. Counter examples can easily be found in Fig. 4. There is a path from S to D, but D can not reach S by any means (asymmetry). D is connected to B and B is connected to S, but there is not path from D to S (intransitivity). Since time dependent path is asymmetry, it is actually *directional*. We use $(i \rightarrow j)^{\mathcal{T}}$ to denote the time dependent path from i to j in time period $\mathcal{T}$.

Given a time dependent graph, the existence of a time dependent path between two nodes can be determined in polynomial time. Algorithm 2 can be used to determine such path. The basic idea is to record all reachable nodes with a set. Then for each time period, check all the neighbors of reachable nodes and add them to the reachable set if they have not been reached yet. The time complexity of the algorithm is $O(t \cdot n \cdot m)$.

Since the definition of path has been significantly changed. Correspondingly, the definition of shortest path between two nodes shall also be redefined.

Definition 7: Time Dependent Shortest Path (TDSP): The time dependent shortest path from i to j in TDG is the path in all time dependent paths from i to j with the minimum total cost.

Given a pair of nodes s and d, finding TDSP is more difficult than determining the existence of a TDP. Since there might be multiple time dependent paths from s to d. Therefore, the basic idea of finding the shortest path is to firstly find the earliest reachable path, and then update if there exists a shorter one. Algorithm 3 shows how to find the time dependent shortest path. We use a list of 3-tuples (total cost, time period,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2562699, IEEE Sensors Journal

Algorithm 3 Time dependent shortest path algorithm TDSP$\{\mathcal{G},\ s,\ d,\ t_s,\ t_e\}$, where s is source, d is destination, t_s is starting time and t_e is ending time.

```
// Store a list of 3-tuples (total cost, time period, parent) for each visited nodes
vdict[s] ← (0, −1, None)
for t ← t_s, t_e do
    for all v ∈ V do
        if v ∉ KEYS(vdict) then
            Continue
        end if
        for all n ∈ NEIGHBOURS(v) do
            cost ← C(v, n)
            // List index -1 is the last item.
            if cost + vdict[v][−1][0] < vdict[n][−1][0] then
                vdict[n] ← vdict[n] ∪ (cost + vdict[v][−1][0], t, v)
            end if
        end for
    end for
end for
return BACKTRACE(d, s)
```

parent) to track the total cost, the data transmission time, and the parent of each transmission. Then, for each time period and each neighbor of reachable node, if the neighbor's cost is greater than the current node's cost plus the edge cost, we will insert the transmission to the tuple list. The shortest path can be obtained by a back trace from the destination to the source. The time complexity of this algorithm is also $O(t \cdot m \cdot n)$, and the space complexity is $O(t \cdot n)$.

D. Solution to Inseparable Flow Routing

With the definition and algorithm of time dependent shortest path, the inseparable flow routing problem can be easily solved. Since the only constraint is the bandwidth of edges.

The key idea is that we firstly traverse the entire time dependent graph to delete all the edges whose bandwidth is less than the constraint b, and then find the time dependent shortest path in the new time dependent graph, with communication cost as the weight. This approach ensures the optimality of the solution. Therefore, the inseparable flow routing problem can be solved in polynomial time.

E. Solution to Separable Flow Routing

Differently from inseparable flow routing problem, which can be solved in polynomial time, the computational complexity of separable flow routing problem is much higher. If the graph is static rather than time dependent, the separable flow routing problem is equivalent to conventional multi-path routing problem, which has been proved to be NP-complete [26] [27]. Now the graph is time dependent, the problem is at least as difficult as multi-path routing. Therefore, the complexity of separable flow routing problem is also NP-complete (Because it is easy to prove that given a solution, the correctness can be verified in polynomial time). Therefore, it is unlikely to have a polynomial time solution, unless P = NP.

Algorithm 4 Greedy Approximation for Separable Flow Routing SFR$\{\mathcal{G},\ s,\ d,\ b\}$, where s is the source, d is the destination, b is the bandwidth constraint.

```
F_a ← ∅
// 1. Calculate the profit of all shortest paths
while HASPATH(G, s, d) do
    F ← TDSP(G, s, d)
    F_a ← F_a ∪ F
    for all (s, r, w)^t ∈ F do
        P(s, r, t) = P(s, r, t) − BW(F)
    end for
end while
// 2. Sort all the paths according to profit
F_a ← SORT(F_a, PR)
// 3. Combine paths with maximum profit
bandwidth ← 0
F_r ← ∅
while bandwidth < b do
    F ← POP(F_a)
    F_r ← F_r ∪ F
    bandwidth ← bandwidth + BW(F)
end while
// 4. Find the min cost path whose bandwidth ≥ b
F_m = argmin_{F_i ∈ F_a ∧ BW(F_i) ≥ b} (C(F_i))
if C(F_r) < C(F_m) then
    return F_r
else
    return F_m
end if
```

Since we have to tolerate a sub-optimal approximation or heuristic algorithm, in this work, we propose to design a polynomial time approximation scheme (PTAS) for this problem. In designing a greedy approximation algorithm, the key idea is to maximize the *profit* of the newly added path at each iteration. The profit of a path is defined as:

Definition 8: Path Profit:

The profit of a path F can be calculated by:

$$PF(F) = \frac{BW(F)}{\sum\limits_{e \in F} C(e)} \tag{8}$$

The complete algorithm is listed in Algorithm 4. This is a polynomial time algorithm. Specifically, the time complexity is upper-bounded by $O(3 \cdot t \cdot n^2 \cdot m)$. Next, we prove the approximation ratio of the proposed algorithm.

Theorem 1: Algorithm 4 is a 2-approximation solution for separable flow routing.

Proof:

We use `OPT` and `Greedy` to denote the optimal solution and our greedy solution, respectively. $\alpha \in (0, 1)$, and $BW(F_r) - \alpha BW(F_m) = b$. From the algorithm, the following equation holds:

$$C(F_r - F_m) \geq C(F_m) \tag{9}$$

Therefore, we can obtain that:

$$
\begin{aligned}
C(\texttt{Greedy}) &= \min(C(F_r), C(F_m)) \\
&\leq 2C(F_r - F_m) \\
&= 2(C(F_r) - C(F_m)) \\
&\leq 2(C(F_r) - \alpha C(F_m)) \\
&\leq 2C(\texttt{OPT})
\end{aligned}
$$

■

To summarize, in real-world applications, according to the capability of SDN controllers, we classify the multi-hop data transmission problem into two categories: separable and inseparable flow routing. The inseparable flow routing problem can be solved in polynomial time. However, the separable flow routing problem is NP-complete. We present a polynomial time approximation scheme (PTAS) and proved the approximation ratio.

V. Evaluation

To evaluate the performance of the proposed solutions, we build an urban traffic monitoring prototype application, which collects surrounding vehicle speed and estimates the overall traffic condition accordingly. To make the evaluation results convincing, we use the crowdsourced taxi traces from Shenzhen, China. It is regulated that every taxi has to install an onboard computer with GPS and cellular network modules, which upload the taxi's latest location information to the data center periodically [28]. There are over 28,000 taxis in total. Everyday, they update approximately 80 million records, which take about 10 GB of storage. The mean update interval is 20.41 seconds, and the standard deviation is 20.54.

Fig. 5 shows the flow chart of the evaluation and Table II lists the configuration of the simulators. The taxi trace contains vehicle distributions in both time and space domain, and can be regarded as the accurate trajectory prediction, which can hardly be achieved in real applications. To be practical, we intentionally add some errors to the trace according to the results of [23]. This ensures the trajectory prediction errors in the real world can also be reflected in the evaluation.

Next, we partition the trace in both time and space domains to obtain the network availability matrix and the time dependent graph, which are inputs of our proposed algorithm. In this evaluation, the change of the data is insignificant in both domains. Therefore, we divide the region into square blocks with edge length from 100 m to 1,000 m, and the time period from 1 min to 50 min. Since the cellular network is always available for all taxis, we use it to communicate with the control plane. For other networks, we calculate the network availability simply according to constant range model.

Finally, we run the proposed algorithms, which are implemented in Python, to obtain the scheduling policies. The policies are simulated using POX SDN controller [24] and NS-3 [29] to obtain the network transmission performance. The entire procedure is also automated using Python script. As we discussed before, due to the limitation of POX and NS-3, we can not simulate the separable flow multi-hop transmission. Therefore, the results are only given by theoretical computation.

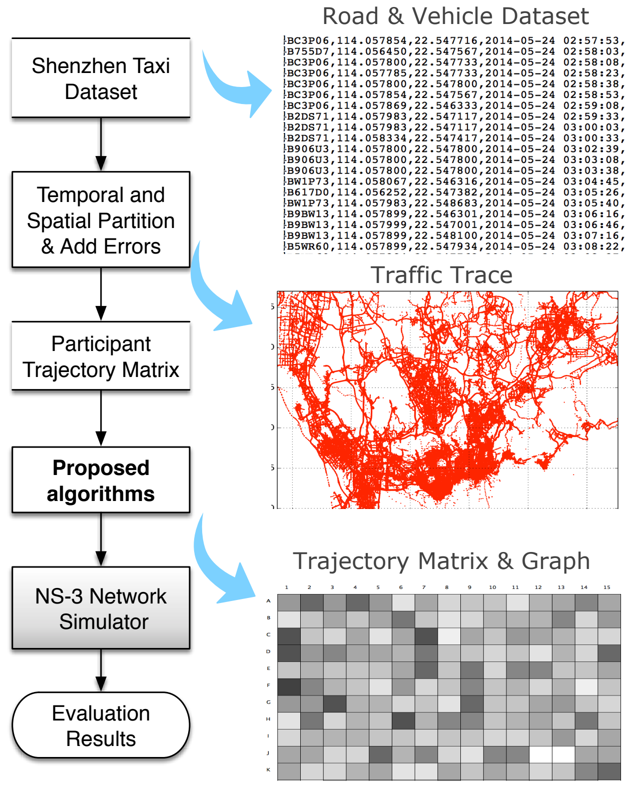


Fig. 5. Flow chart of the evaluation process

TABLE II
EVALUATION CONFIGURATIONS AND PARAMETERS

Parameter	Value
RSU Density	1000m / RSU on avg.
DSRC MAC/PHY Standard	IEEE 802.11p SCH
DSRC Transmission range	250m
Wi-Fi MAC/PHY Standard	IEEE 802.11g
Wi-Fi Transmission range	100m
Propagation delay mode	Constant Speed
MAC type	Adhoc Wifi Mac
Propagation loss model	Range Propagation Loss Model
Cellular Communication Cost	100 / KB
Wi-Fi Communication Cost	10 / KB
DSRC Communication Cost	5 / KB
Bandwidth requirements	50 KB/s

A. *Evaluation of Single Hop Data Transmission*

We examine the computational complexity and optimality of both exhaustive search and greedy algorithms. Fig. 6 illustrates the results. From Fig. 6(a), we can see that the time cost of exhaustive search algorithm increases exponentially, with the increase of wireless interface number n. Differently, the time cost of approximation algorithm increases much slower than the exhaustive search. Therefore, the greedy algorithm is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2562699, IEEE Sensors Journal

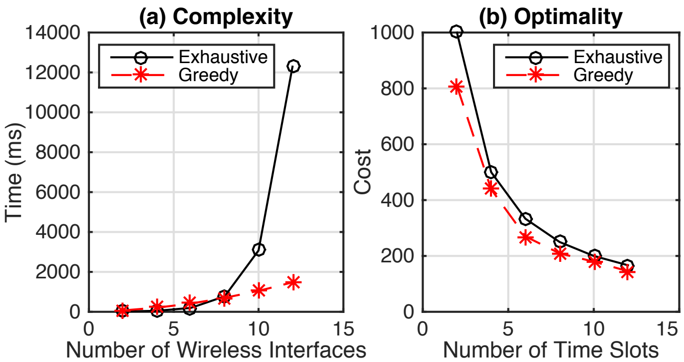


Fig. 6. Complexity v.s. Optimality in single hop data transmission

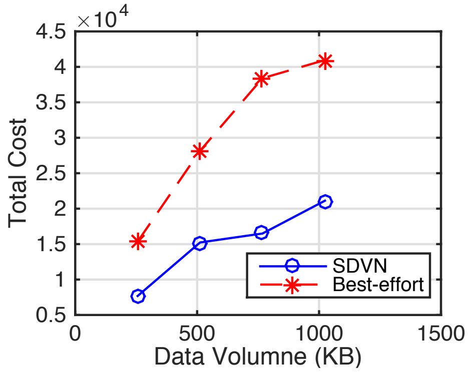


Fig. 7. Total cost comparison between SDVN and the best-effort approach

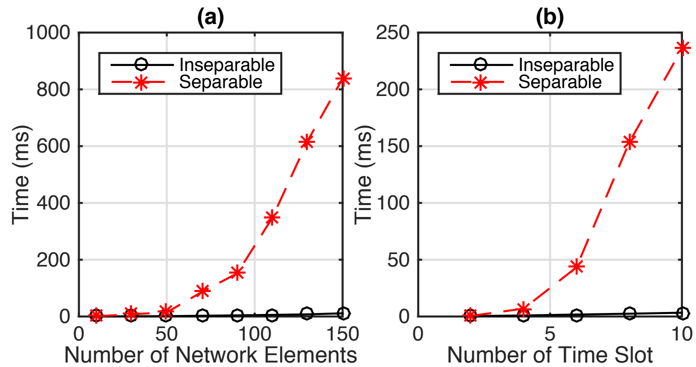


Fig. 8. Computational complexity of separable routing and inseparable routing algorithms. (a) The time increases with the number of network elements. (b) The time increases with the number of time slots.

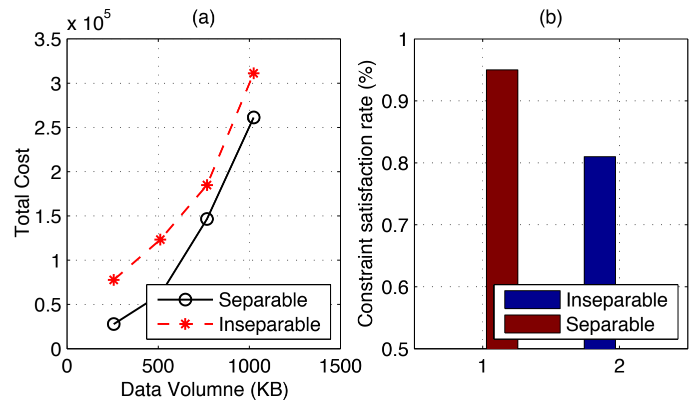


Fig. 9. Performance of separable routing and inseparable routing algorithms: (a) Total cost. (b) Constraint satisfaction ratio

better for large-scaled application. Fig. 6(b) demonstrates the optimality of the two algorithms. Since exhaustive search can always find the optimal solution, this can be regarded as the upper bound of the algorithm. From this figure, we can see that the approximation ratio of the greedy algorithm is quite satisfactory. Normally, it can ensure that the solution found by greedy algorithm is approximately 80% closer to the optimal ones.

For single hop data transmission, we compare the total cost of our solution with the best-effort forwarding, which tries to forward data to the destination whenever there is an available link, while the homogeneous transmission only uses a single network interface to forward data. Fig. 7 illustrates the evaluation results. From this figure, we can find out that the total cost of our SDVN based approach is only approximately 50% of the best-effort one. The main reason is that our scheduling policy can avoid the use of the high cost wireless media. Meanwhile, it can still satisfy the bandwidth requirement of the applications.

B. Evaluation of Multi-hop Data Transmission

We first evaluate the computational complexity of both inseparable and separable flow algorithms. Fig. 8 illustrates the running time of both solutions. From the previous analysis, we know that both algorithms are polynomial time solvable. However, the execution time still have remarkable differences. In both cases, the execution time of inseparable flow routing is much higher than the separable one. The results further validates our theoretical analysis.

Next, we evaluate the total cost of inseparable and separable routing, given the same source, destination, and bandwidth constraint. The total costs are computed theoretically, as depicted in Fig. 9. The separable flow routing always require less cost than the inseparable one. The reason is that the inseparable flow routing always choose a single path that can satisfy the bandwidth constraint. In order to achieve the objective, the single path usually need to “detour” to bypass those links with small bandwidth, which will cause additional data forwarding and add additional cost. Moreover, the separable flow routing can also provider better bandwidth to applications, as shown in Fig. 9(b). The total bandwidth satisfaction rate of separable flow routing is over 10% higher than the inseparable ones. This is also expected because data transmission using multiple paths can always provide better quality of service.

Finally, we select packet delivery ratio as the performance metric of data transmission, and compare our SDVN based approach with two typical protocols widely used in vehicular network: OLSR as structure based protocols, and GPSR as structure free one. We compare the performance using inseparable routing, and the results is depicted in Fig. 10. From this figure, we can see that the overall packet delivery radio

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2562699, IEEE Sensors Journal

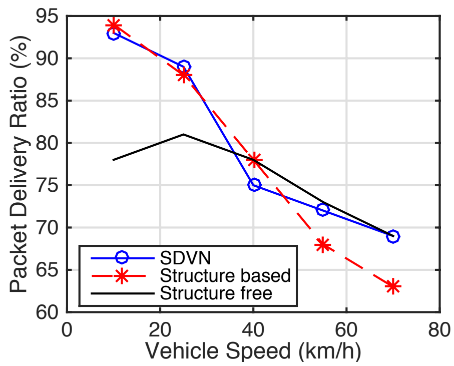


Fig. 10. Packet delivery ratio in vehicular networking applications

decreases with vehicle speed. This is mainly caused by the fact that when vehicle speed goes higher, the density of vehicles is usually also not high. In this case, vehicles may have difficulty in finding next hop neighbors, which will cause packet delivery failure. Despite of this common factor, we can see that with the help of SDN, the overall performance is remarkably improved than any of the two protocols used individually.

VI. Related Work

A. Heterogeneous Vehicular Networks

Heterogeneity is one of the most important features of vehicular networks, especially with the upcoming 5G mobile networks [10], [11]. Problems caused by network heterogeneity should be carefully handled. Currently, delay tolerance, mobile data offloading and vertical handoff are three commonly used technologies in heterogeneous vehicular networks. All of them can potentially improve the performance of the network and reduce the communication cost.

Vehicular networks are usually described as delay tolerant networks [30]. Spyropoulos et al. [31] focus on the routing problem for heterogeneous mobile networks, in which nodes are categorized into different classes according to their characteristics. The major finding is that choosing relay carefully rather than making replica in a best-effort approach can significantly improve the communication performance. Alternatively, GeoDTN+Nav [32] presents a geographical routing scheme for vehicular networks. The key insight is that the vehicle trajectory can be utilized to enhance the routing decision. Similarly, our research also utilize vehicle trajectory for decision making. The main difference is that we use the trajectory of multiple vehicles for global decision making.

Deshpande et al. [17] conduct a head-to-head comparison of the 3G and WiFi performance in vehicular scenario. They discover that two networks are complimentary and it is possible to build a hybrid network for better performance. Make a step further, Hou et al. [16] propose oSCTP, an offloading technique between 3G and WiFi networks. The objective is to reduce the expensive 3G transmission. The offloading issue is formulated as an optimization problem to maximize user benefit. However, data offloading aims at using one technology to augment the other. The application scenarios are quite limited.

Lee et al. [20] developed a connection management and optimal resource allocation algorithm for vertical handoff in vehicular networks. This research also aims at maximizing the battery life of mobile nodes. We argue that in vehicular networks, battery is not a primary concern as there is always an onboard battery to supply the communication devices. Shafiee et al. [19] propose a similar research as us. In this research, they also considered heterogeneous networks in vehicular scenario, and the objective is also to reduce communication cost. However, the main difference is that their approach is fully distributed at vehicle side. In our approach, SDN is utilized for centralized coordination.

B. Multi-path Routing in Vehicular Networks

Multi-path routing protocols have been investigated in vehicular networks and other forms of mobile wireless networks to improve the total bandwidth and increase the reliability. Path discovery is a fundamental step in multi-path routing. Generally speaking, existing multi-path routing protocols can be divided into two categories: proactive and reactive. Proactive protocols discover the routing path according to certain tables, while reactive ones usually conduct path discovery on demand. Due to the mobile nature of vehicular networks, most of existing multi-path routing protocols in vehicular network are reactive ones.

AOMDV [33] is a multi-path version of the AODV, which is a classical reactive protocol for mobile wireless networks. The key idea is to modify the path discovery mechanism to provide loop-free disjoint multi-path between the source and destination nodes. Based on it, researchers have tailored the AOMDV to adapt vehicular networks. S-AOMDV proposed by Chen et al. [34] not only considers hop as metric, but also takes speed into account. Researchers have also presented DSR extensions to support multi-path routing [35].

Besides, to support multi-path routing in vehicular networks, the hop count is usually not the only metric. As we mentioned, S-AOMDV [34] utilize speed as alternative metric. Similarly, JMSR [36] considers geographical information. A road junction based multi-path routing scheme is designed.

Our SDVN based multi-hop data transmission also employ a multi-path scheme. The key difference between our approach and existing ones is that SDVN has a centralized controller, which can be used for centralized coordination of path discovery. Instead, existing approaches usually discover routing paths on demand in a decentralized manner.

C. Heterogeneous Software Defined Wireless Networks

Interoperability is a key issue for heterogeneous wireless networks. Guimaraes et al. [37] studies the handover problem under heterogeneous networks. The proposed architecture combines IEEE standard 802.21 media independent handover (MIH) with OpenFlow. With the enhancement of MIH, SDN controller is aware of the mobility of mobile nodes. Therefore, more smooth handover decisions can be made. Yi's work [38] adopts similar idea with the aforementioned paper. Differently, it highlights the proof of concepts with real world implementation. It also performs experiments with WiFi network.

Besides, unified abstraction of different wireless interfaces is also interesting. U-WN [39] is a uniform network architecture for various wireless standards empowered by SDN. With the help of SDN, uniform protocol stack and global optimization is made possible.

Heterogeneous wireless network enables a series of applications, including data offloading, heterogeneous multi-hop routing and etc. Amani et al. [40] propose an offloading scheme for WiFi and LTE networks. With the help of SDN, the controller can deploy corresponding offloading polices according to real-time network conditions. So that data traffic are delivered either via WiFi or via LTE networks. Qin et al. [41] investigate the network heterogeneity of IoT devices. The paper proposed MINA, a resource matching and service solution specification method to schedule the data flow under a layered SDN architecture. Authors also propose two centralized flow scheduling algorithms to schedule end-to-end data communication, one based on network calculus, the other based on genetic algorithm.

Differently from the aforementioned approaches, in SDVN, the network is not only heterogeneous, but also with mobile node and ad-hoc communication model. Therefore, our architecture and problem are more sophisticated.

VII. Conclusion

We have addressed the problem of data transmission in heterogeneous vehicular networks. We have proposed an SDN based heterogeneous communication approach and have discussed the architecture of SDVN. Moreover, the scheduling problem has been formulated as an optimization problem and two solutions have been proposed. The major finding is that using a small amount of communication cost to collect the network topology information can significantly improve the overall network performance. We have also conducted extensive evaluation with traffic traces.

Acknowledgment

This work is partially sponsored by CCF-Tencent Open Fund, the National Science Foundation of China (Grant No.61472283 and 61103185), the Fok Ying-Tong Education Foundation, China (Grant No. 142006), the Fundamental Research Funds for the Central Universities (Grant No. 2013KJ034 and No. 2100219043).

References

[1] T. RAY, "What kind of computer will you drive?" Barrons, 2014, online; accessed May 2015.

[2] V. Bychkovsky, K. Chen, M. Goraczko, H. Hu, B. Hull, A. Miu, E. Shih, Y. Zhang, H. Balakrishnan, and S. Madden, "Data management in the cartel mobile sensor computing system," in *ACM SIGMOD International Conference on Management of Data*, 2006, pp. 730–732.

[3] S. Kumar, L. Shi, N. Ahmed, S. Gil, D. Katabi, and D. Rus, "Carspeak: A content-centric network for autonomous driving," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 259–270, 2012.

[4] IEEE, "Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments," *IEEE Std 802.11p-2010*, pp. 1–51, 2010.

[5] D. Zhang and C. K. Yeo, "Enabling efficient wifi-based vehicular content distribution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 3, pp. 479–492, March 2013.

[6] H.-M. Tsai, O. Tonguz, C. Saraydar, T. Talty, M. Ames, and A. MacDonald, "Zigbee-based intra-car wireless sensor networks: a case study," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 67–77, 2007.

[7] G. Leen and D. Heffernan, "Vehicles without wires," *Computing and Control Engineering Journal*, vol. 12, pp. 205–211, 2001.

[8] A. Al Masud, M. Mondal, and K. Ahmed, "Vehicular communication system for vehicle safety using rfid," in *Malaysia International Conference on Communications (MICC)*, 2009, pp. 697–702.

[9] C.-M. Chou, C.-Y. Li, W.-M. Chien, and K. chan Lan, "A feasibility study on vehicle-to-infrastructure communication: WiFi vs. WiMAX," in *International Conference on Mobile Data Management (MDM)*, 2009, pp. 397–398.

[10] M. Chen, Y. Zhang, L. Hu, T. Taleb, and Z. Sheng, "Cloud-based wireless network: Virtualized, reconfigurable, smart wireless network to enable 5g technologies," *Mobile Networks and Applications*, vol. 20, no. 6, pp. 704–712, 2015.

[11] M. Chen, Y. Zhang, Y. Li, S. Mao, and V. C. M. Leung, "Emc: Emotion-aware mobile cloud computing in 5g," *IEEE Network*, vol. 29, no. 2, pp. 32–38, 2015.

[12] Y. Zhang, M. Chen, S. Mao, L. Hu, and V. C. M. Leung, "Cap: community activity prediction based on big data analysis," *IEEE Network*, vol. 28, no. 4, pp. 52–57, 2014.

[13] H. Qin, Z. Li, Y. Wang, X. Lu, W. Zhang, and G. Wang, "An integrated network of roadside sensors and vehicles for driving safety: Concept, design and experiments," in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2010, pp. 79–87.

[14] Z. He, J. Cao, and X. Liu, "High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility," in *IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2542–2550.

[15] N. Cheng, N. Lu, N. Zhang, X. S. Shen, and J. W. Mark, "Vehicular wifi offloading: Challenges and solutions," *Vehicular Communications*, vol. 1, no. 1, pp. 13–21, 2014.

[16] X. Hou, P. Deshpande, and S. Das, "Moving bits from 3g to metro-scale wifi for vehicular network access: An integrated transport layer solution," in *IEEE International Conference on Network Protocols (ICNP)*, 2011, pp. 353–362.

[17] P. Deshpande, X. Hou, and S. R. Das, "Performance comparison of 3g and metro-scale wifi for vehicular network access," in *SIGCOMM conference on Internet measurement (IMC)*. ACM, 2010, pp. 301–307.

[18] D. Kwak, J. Mo, and M. Kang, "Investigation of handoffs for ieee 802.11 networks in vehicular environment," in *International Conference on Ubiquitous and Future Networks (ICUFN).*, 2009, pp. 89–94.

[19] K. Shafiee, A. Attar, and V. Leung, "Optimal distributed vertical handoff strategies in vehicular heterogeneous networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 534–544, 2011.

[20] S. Lee, K. Sriram, K. Kim, Y. H. Kim, and N. Golmie, "Vertical handoff decision algorithms for providing optimized performance in heterogeneous wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 865–881, 2009.

[21] C. Chaudet and Y. Haddad, "Wireless software defined networks: Challenges and opportunities," in *IEEE International Conference on Microwaves, Communications, Antennas and Electronics Systems (COMCAS)*, 2013, pp. 1–5.

[22] C. Barrios, Y. Motai, and D. Huston, "Trajectory estimations using smartphones," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7901–7910, 2015.

[23] P. Pathirana, A. Savkin, and S. Jha, "Location estimation and trajectory prediction for cellular networks with mobile base stations," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 6, pp. 1903–1913, 2004.

[24] POX, "Pox: Python-based openflow controller," http://www.noxrepo.org/pox/about-pox/, 2015, online; accessed May 2015.

[25] D. Kreutz, F. M. V. Ramos, P. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *CoRR*, 2014. [Online]. Available: http://arxiv.org/abs/1406.0440

[26] J. Shin, F. Devetak, T. Anjali, and S. Kapoor, "Concurrent multipath routing over bounded paths: Minimizing delay variance," in *IEEE Global Communications Conference (GLOBECOM)*, Dec 2013, pp. 1483–1488.

[27] J. Chen, S.-H. Chan, and V. Li, "Multipath routing for video delivery over bandwidth-limited networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 10, pp. 1920–1932, Dec 2004.

[28] D. Zhang, J. Huang, Y. Li, F. Zhang, C. Xu, and T. He, "Exploring human mobility with multi-source data at extremely large metropolitan

scales," in *International Conference on Mobile Computing and Networking (MobiCom)*, 2014, pp. 201–212.

[29] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and Tools for Network Simulation*, 2010, pp. 15–34.

[30] V. N. G. J. Soares, F. Farahmand, and J. J. P. C. Rodrigues, "A layered architecture for vehicular delay-tolerant networks," in *IEEE Symposium on Computers and Communications. (ISCC 2009)*, 2009, pp. 122–127.

[31] T. Spyropoulos, T. Turletti, and K. Obraczka, "Routing in delay-tolerant networks comprising heterogeneous node populations," *IEEE Transactions on Mobile Computing*, vol. 8, no. 8, pp. 1132–1147, 2009.

[32] P.-C. Cheng, K. C. Lee, M. Gerla, and J. Härri, "GeoDTN+Nav: Geographic dtn routing with navigator prediction for urban vehicular environments," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 61–82, 2009.

[33] M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 3, pp. 92–93, 2002.

[34] Y. Chen, Z. Xiang, W. Jian, and W. Jiang, "An improved aomdv routing protocol for v2v communication," in *IEEE Intelligent Vehicles Symposium (IV)*, 2009, pp. 1115–1120.

[35] S.-J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in *IEEE International Conference on Communications (ICC)*, vol. 10, 2001, pp. 3201–3205.

[36] P. Sermpezis, G. Koltsidas, and F.-N. Pavlidou, "Investigating a junction-based multipath source routing algorithm for vanets," *IEEE Communications letters*, vol. 17, no. 3, pp. 600–603, 2013.

[37] C. Guimaraes, D. Corujo, R. Aguiar, F. Silva, and P. Frosi, "Empowering software defined wireless networks through media independent handover management," in *Global Communications Conference (GLOBECOM)*, 2013, pp. 2204–2209.

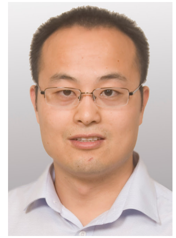
[38] G. Yi and S. Lee, "Fully distributed handover based on SDN in heterogeneous wireless networks," in *International Conference on Ubiquitous Information Management and Communication (ICUIMC)*, 2014.

[39] S. Zhang, C. Kai, and L. Song, "SDN based uniform network architecture for future wireless networks," in *International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2014, pp. 1–5.

[40] M. Amani, T. Mahmoodi, M. Tatipamula, and H. Aghvami, "Programmable policies for data offloading in LTE network," in *International Conference on Communications (ICC)*, 2014, pp. 3154–3159.

[41] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, "A software defined networking architecture for the internet-of-things," in *Network Operations and Management Symposium (NOMS)*, 2014, pp. 1–9.

Zongjian He Zongjian He received his PhD degree in Department of Computing, the Hong Kong Polytechnic University in 2015, and MSc and BEngr degree from Tongji University, Shanghai, China in 2007 and 2004, respectively. He is currently an assistant Professor in School of Software Engineering, Tongji University. His research interests include wireless sensor networks, vehicular networks, participatory sensing applications and mobile computing.



Daqiang Zhang Daqiang Zhang received the B.Sc. degree in management science and the M.Sc. degree in computer science from Anhui University, Heifei, China, in 2003 and 2006, and the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2010. He has published more than 60 papers in major journals and international conferences. As the first author, he has published papers in IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, IEEE WIRELESS COMMUNICATIONS, IEEE NETWORK, IEEE SYSTEMS JOURNAL, IEEE ICPP, IEEE ICC, and IEEE WCNC. His research includes mobile computing, distributed computing, and wireless sensor networks. Currently, he is an Associate Professor with the School of Software Engineering, Tongji University, Shanghai.

Junbin Liang Junbin Liang received his BSc and MSc degrees in Computer Science from the Guangxi University in 2000 and 2005, respectively. He received his PhD degree in Central South University of China in 2010. Currently, he is a professor in Guangxi University. His research interests include mobile ad hoc networks and wireless sensor networks.