Real-Time Density Detection in Connected Vehicles: Design and Implementation

Linghe Kong, Guangtao Xue, Kayhan Zara Ghafoor, Rasheed Hussain, Hao Sheng, and Peng Zeng

To speed up density acquisition, the authors propose an RDD system. Leveraging the frequency resource, RDD divides the wireless channel into fine-grained subchannels and detects the neighbors in a parallel manner. They establish a testbed using software defined radios and experimentally validate RDD.

Abstract

Density information plays an important role in intelligent transportation systems for not only traffic control but also information sharing. Existing products have been able to provide coarse-grained density services. For example, Google Maps can illustrate the traffic conditions by different colors via Internet connection. Vehicle-to-vehicle wireless communications can locally acquire the density by information exchange and neighbor counting. However, either the Internet access or one-by-one counting leads to a sub-second-level delay, which cannot satisfy real-time vehicular applications such as autonomous navigation and data dissemination. To speed up density acquisition, we propose an RDD system. Leveraging the frequency resource, RDD divides the wireless channel into fine-grained subchannels and detects the neighbors in a parallel manner. We establish a testbed using software defined radios and experimentally validate RDD. Moreover, to evaluate RDD in high-density scenarios, extensive simulations are conducted based on real collected data. Both the experiment and simulation results demonstrate that RDD achieves 100 ms level density detection, while the state-of-the-art time-domain acceleration method is at the 10 ms level.

Introduction

Smart and connected vehicles [1, 2] are valuable for safe and efficient transportation. To enhance the connectivity, various wireless protocols have been applied into connected vehicles such as WiFi/LTE [3], millimeter-wave [4], and visible light communications [5]. Especially, 5.9 GHz spectrum is licensed to support IEEE 802.11p-based dedicated short-range communication (DSRC) [6]. The scale of connected vehicles is poised to increase dramatically.

Plenty of emerging applications are envisioned for connected vehicles such as traffic control [3], autonomous navigation [4], and satellite-terrestrial data forwarding [7]. Most of these applications rely heavily on real-time density. For example, an autonomous vehicle can "see" line-of-sight vehicles by lidar and camera to avoid crashing. However, non-line-of-sight density is further needed for path planning.

A great number of density detection methods have been investigated in both the academic and industrial communities. For example, drivers are used to checking Google Map to get road density via different colors, in which green presents the clear way and red indicates congestion. However, Internet access causes a second level of delay. Dispensing with the Internet, vehicle-to-vehicle communication systems [8] can get the local density by identifying and counting neighbors through information exchange. However, one-by-one identification may trigger enormous wireless collisions and incomplete detection, especially in high-density and high-speed cases.

Motivated by further accelerating density detection, we propose a novel real-time density detection (RDD) system for connected vehicles by fully exploiting the wireless spectrum. Using DSRC as an example, the core design of RDD is to divide a 10 MHz channel into fine-grained subchannels and detect the number of neighbors in a parallel manner. In RDD, after receiving a request from a detector, every neighbor divides the channel into multiple subchannels and randomly chooses one subchannel to feed back simultaneously. The detector estimates the density by analyzing the overlapped feedback in the frequency domain. Although the concept of RDD sounds straightforward, it is not easy to implement in practice because of three challenges.

It is challenging to determine the number of subchannels without knowledge of density. On one hand, more subchannels can enhance the parallel capability. On the other hand, too many subchannels may lead to inter-subchannel interference and decrease the estimation accuracy. In addition, factors such as frequency offset and communication range also affect the subchannel division. To this end, we initialize a divisor by theoretical computation and then adaptively tune it to approach the optimum.

It is challenging to accurately recognize the feedback because of co-channel interference. When detection is processing, concurrent DSRC transmission in the same channel possibly interferes with RDD feedback. To address this challenge, we design a filter according to the different numbers of subchannels in proposed RDD (fine-grained subchannels) and in standard DSRC. This filter can elaborately separate RDD feedback and DSRC transmission.

Digital Object Identifier: 10.1109/MCOM.2018.1800144

Linghe Kong and Guangtao Xue are with Shanghai Jiao Tong University; Kayhan Zara Ghafoor is with Cihan University-Erbil; Rasheed Hussain is with Innopolis University; Hao Sheng (corresponding author) is with Beihang University; Peng Zeng is with the Chinese Academy of Sciences.

The RDD design is an ingenious combination of several techniques including fine-grained channel division, interference separation, and density estimation. Integrating them effectively is challenging, and has never been examined before. We implement the RDD prototype by building a four-node testbed. The feasibility of the RDD system is preliminarily validated by outdoor experiments.

The major contributions of this work are twofold:

- We study the problem of real-time density detection for connected vehicles. To solve this problem, we design a novel RDD system, which is a local density estimation framework based on vehicle-to-vehicle communications. The core of RDD is fine-grained frequency division to reduce the time cost.
- We implement the RDD system on universal software radio peripherals (USRPs) and establish a four-node testbed. Extensive experiments and simulations are conducted to evaluate its performance. Performance results demonstrate that RDD reduces the estimation duration from the 10 ms level to the 100 μs level with competitive accuracy.

Related Work

We divide existing density acquisition methods into four categories and briefly summarize them in Table 1.

Mathematical models formulate the movement of vehicles and estimate the density by these theoretical models. For example, a Poisson traffic model [9] can present the arrival/departure rates of vehicles, and historical data mining can predict future density. Although the theoretical method is statistically significant and has no communication delay, its results are not accurate enough for practical service.

Targeting the practical, *global density acquisition* has become a commercial service. Google Maps is currently a popular tool to illustrate coarse road density by red, yellow, and green. However, infrastructures and Internet access are required in this category, resulting in a second level of delay.

Independent from the Internet, *local density detection* leverages on-vehicle communication systems to identify neighbors one by one. In most vehicular applications, local density is adequate because faraway density is usually useless for immediate operation. Nevertheless, one-by-one identification such as neighbor discovery [10] may trigger enormous wireless collisions and incomplete detection, especially in high-density and high-speed cases.

Since ID information is not necessary for density detection, *time-domain acceleration* gives up the long ID messages and requires just 1-bit feedback from every neighbor to estimate the cardinality, such as FSA [11], achieving 10 ms level density estimation. Although methods in this category fully exploit the time resource, the frequency resource is totally ignored, where every DSRC channel has 10 MHz bandwidth.

Problem Statement

The concept of local density in connected vehicles and our motivation for studying real-time density detection are introduced in this section.

Category	Representation	Limitation	Time cost
Mathematical model	Poisson traffic model [9]	Inaccurate in practice	N/A
Global density acquisition	Google Map	Internet access	≈ 1 s
Local density detection	Neighbor identification one by one [10]	Collision avoidance	≈ 100 ms
Time-domain acceleration	Frame slotted ALOHA (FSA) [11]	Underuse the bandwidth	≈ 10 ms

Table 1. Existing density detection methods.

Local Density Detection

This work mainly concerns local density detection, where local density is defined as the number of vehicles within a custom range. Compared to global density, local density plays a more important role in emerging vehicular applications, especially in data sharing applications. To quickly detect local density, vehicle-to-vehicle communication such as DSRC is the most appropriate technique, which avoids the time-consuming Internet access.

Then we give three definitions in this problem:

- The *detector* is present because the vehicle needs to know the local density.
- Local density is formulated as $N/\pi R^2$, where N is the number of neighbors and R is the *custom range*. The custom range is set by the detector according to the application requirement. However, a too long R is useless because the faraway density is irrelevant to immediate operation. In this work, we set the range between 0 to 300 m according to field test of DSRC communication range [12]. Moreover, cooperative communication [13] can achieve a longer range in vehicular networks if needed.
- The *neighbors* are the vehicles in the custom range of the detector.

Motivation

Two motivations inspire us to concentrate on the problem of *real-time density detection*.

First, since vehicles are highly dynamic, the functionalities of emerging vehicular applications fundamentally rely on real-time density information. For example, [14] proposes the power control method using real-time density to balance the coverage and communication quality. More applications using real-time density can be found in autonomous navigation [4] and data dissemination [15].

Second, real-time density detection can free more wireless resources. Large amounts of vehicular applications such as transportation safety, advertisement, and entertainment drastically constrain the time resource of DSRC, especially in high-density scenarios. In order to avoid channel saturation and reduce collisions, it is significant to efficiently exploit the wireless channel.

Design of RDD System

The design overview of the RDD system includes two major parts: detector and neighbors. The detector is in charge of the custom range setting and fine-grained subchannel division, while every neighbor randomly chooses one subchannel to

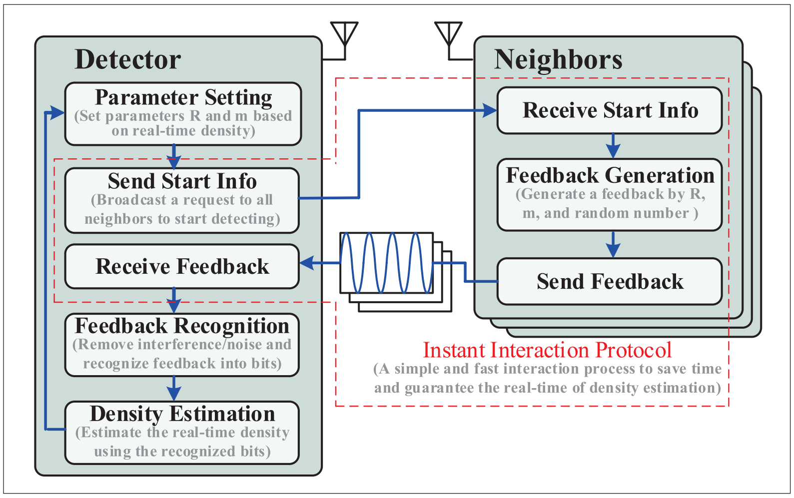

Figure 1. Architecture of the RDD system.

send the shortest beacon as feedback simultaneously. As a result, the detector can recognize the concurrent feedbacks and estimate the local density very quickly. The architecture of the RDD system is illustrated in Fig. 1. Four key components in RDD are described briefly as follows.

Parameter setting determines the custom range and the divisor of frequency division. The custom range is set according to the vehicular applications. The divisor is initialized by the historical density and tuned by the latest density estimation.

Instant Interaction Protocol (IIP) establishes the interaction rules bridging the detector and the neighbors. The interaction procedure consists of five modules (the dashed line area shown in Fig. 1), which generate simple one-return interaction. First, the detector broadcasts its detection message to trigger this procedure. After receiving the message, every neighbor generates feedback by randomly choosing one subchannel and immediately sends this feedback. All neighbors execute the same operation so that their feedback overlaps at the detector.

Feedback recognition translates the feedback into a bitstream by orthogonal frequency-division multiplexing (OFDM), whose essence is to get the bit 0 or 1 from every subchannel by fast Fourier transform (FFT). Particularly, if co-channel interference is observed, a carefully designed filter will separate the feedback from the interference.

Density estimation estimates the number of neighbors based on the recognized bitstream, and thus obtains the local density. If the bitstream is insufficient to accurately estimate, this component will tune the parameter and trigger the detection again.

Benefiting from these components, RDD can significantly reduce the time consumption on local density acquisition compared to the existing methods in Table 1:

- There is only one-return interaction between the detector and neighbors. All neighbors send feedback concurrently without any time consumption in collision avoidance or retransmission.
- The detector just recognizes the number of beacons in fine-grained subchannels instead of decoding information from packets. Hence, the feedback can be the shortest beacon without any meaningful information.
- Besides the communication component IIP, all computing components in RDD are of low computational complexity.

Next, we detail four components one by one.

Parameter Setting

Two parameters, the custom range R and the divisor m, should be set in this component.

The custom range is set according to the requirements of vehicular applications. For example, we suggest a relative long range R = 300 m for traffic control and a short range R = 100 m for data sharing between autonomous vehicles. Furthermore, to know the multi-level local densities, we can execute multiple detections with different ranges such as R = 50, 100, 200, ...

The divisor is the parameter to determine the number of subchannels in the frequency domain. Too few subchannels are inadequate to afford concurrent feedback from large numbers of neighbors, while too many subchannels may incur inter-subchannel interference, resulting in inaccurate density estimation. Hence, it is nontrivial to set this divisor:

- Without the pre-knowledge of real-time neighbors, the divisor is initialized by the average of historical divisors at the same location and same hour in different days. This setting is inspired by the strong spatio-temporal stability of traffic densities.
- When the density estimation component considers the divisor too small or too large, the detection will be triggered again, and the divisor will be double or half.

Note that the number of subchannels cannot be infinite. The setting of the divisor should consider practical factors such as noise, multi-path, and frequency offset.

Instant Interaction Protocol

The objective of IIP is to build the interaction between detector and neighbors while minimizing the time cost. Before detailing the IIP design, we introduce the channel division in standard DSRC and derive the shortest duration of feedback, which are the theoretical foundation of IIP.

Every channel in DSRC [6] is 10 MHz and is divided into 64 subchannels by default. The leftmost six and rightmost six subchannels stay empty (set as 0) for a guard interval. In addition, four subchannels are set as pilots. The other 48 subchannels can carry modulated data. The duration of the minimal transmission unit (i.e., an OFDM symbol) is 8 μs. A DSRC receiver receives a time domain packet and obtains the data in subchannels by an OFDM module.

In IIP design, the duration of every feedback should be minimized, including two parts. One is the necessary duration for recognition, and the other is the additional duration against different propagation delay, as shown in Fig. 2.

The necessary duration should guarantee enough sampling points for signal recognition according to Nyquist's theorem. The additional duration is adopted because the farthest neighbor and the closest neighbor may be at a distance of R, leading to different propagation delay to the detector. To guarantee that the overlapped part of all feedback is larger than the necessary duration, the additional duration should be added.

Based on the above analysis, IIP works as follows:
- The detector sends the detection message with the tuple including the divisor, the given channel, the real-time location of the detector, and the custom range *R*.
- Once receiving the detection message, every neighbor extracts the parameters from the tuple. Then the feedback generation module starts to prepare the feedback including three steps:
  - Feedback or not: If this neighbor is out of the custom range of the detector, feedback is unnecessary; otherwise, feedback is needed.
  - Beacon generation: The channel is divided into *m* subchannels. The feedback is a beacon in a randomly selected subchannel, that is, a baseband sine signal attaching no information, as shown in Fig. 2.
  - Feedback duration: The duration is set as the foundation analysis. All neighbors concurrently send their feedback, which overlaps in the air.
- The detector receives the overlapped feedback as one signal and extracts the necessary part, which contains beacons from all neighbors. The overlapped signal in the necessary part is delivered to the next component for feedback recognition.

Feedback Recognition

The overlapped signal is translated into a bitstream by this feedback recognition component. Similar to the OFDM in standard DSRC, FFT changes the time-domain signal into the frequency domain. Then every subchannel is scanned: an empty subchannelthat has no feedback is recognized as a bit 0, and a non-empty subchannel that has one or more instances of feedback is recognized as bit 1. All recognized bits form an *m*-length bitstream.

The feedback may suffer from co-channel interference. Such interference is attributed to other DSRC transmissions in the same channel. We address this problem by designing a filter that separates the feedback and the DSRC packet based on their different numbers of subchannels.

We introduce our solution for co-channel interference by a typical example. Assume RDD sets a 256-subchannel division and recall that the standard DSRC has 64 subchannels. The detector receives a fused signal combined by RDD feedback and conventional data packets.

Executing FFT on this fused signal, we can observe two obvious peaks at 64 Hz and 256 Hz due to their numbers of subchannels. Using a high pass filter, we can separate them and get the filtered signal, which is nearly the same as the original feedback.

Density Estimation

The main goal of this component is to compute the local density based on the recognized bitstream. Several existing estimators have been studied in time domain acceleration that build the model between the length of stream *m* and the number of entities to be estimated. These estimators can also be applied in our frequency-divided scenarios. For example, the classic UPE [11] estimator can provide fast and accurate estimation of *N* using the 0–1 distribution in the stream. With the number of neighbors *N*, the local density can be calculated.

Furthermore, this component can adaptively tune the divisor. If all bits in the recognized stream are bit 1s, we consider the recent divisor too small to estimate the number of neighbors. Thus, this component will double the divisor and trigger a new detection. On the contrary, if most bits are 0s in the stream, this component will halve the divisor to reduce the recognition error.

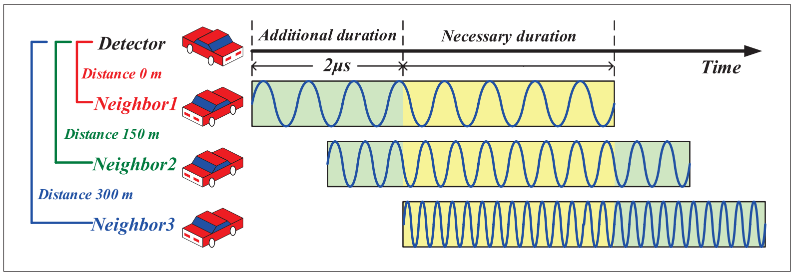

Figure 2. The duration of feedback includes the effective part and the additional part. When the custom range R = 300 m, the additional part is $2R/(3 \times 10^8)$ = 2 μs.

Implementation

We build a four-node testbed and implement RDD in this testbed. Every node consists of a USRP B210, a laptop, a smartphone, and an iRobot, as shown in Fig. 3.

USRP B210: The IIP component is implemented on USRP B210. We select USRP B210 as our hardware because it supports transmission frequency from 70 MHz to 6 GHz, which covers the 5.9 GHz DSRC spectrum. Also, B210 utilizes the fast USB 3.0 port, so the laptop can immediately dump the signals for subsequent computation. 15 dBi antennas are equipped in B210 to enhance the communication range. In addition, we adopt GnuRadio as our development environment and the open project gr-ieee802-11 as the physical layer of IEEE 802.11p. Based on such hardware and software, we develop IIP as introduced above.

Laptop: On one hand, the laptop takes on the major computation tasks. The components, including parameter setting, feedback recognition, and density estimation, are developed and executed on the laptop. On the other hand, the laptop is also a junction that connects the USRP and the smartphone.

Smartphone: Since location information is needed in RDD to make the feedback decision and to compute the additional duration, the smartphone is used to provide real-time GPS information.

iRobot: An iRobot carries all the other devices and moves in a random manner to imitate a mobile vehicle.

Performance Evaluation

Using the testbed, we conduct experiments and real-data-driven simulations to verify the feasibility and evaluate the performance of the proposed RDD system.

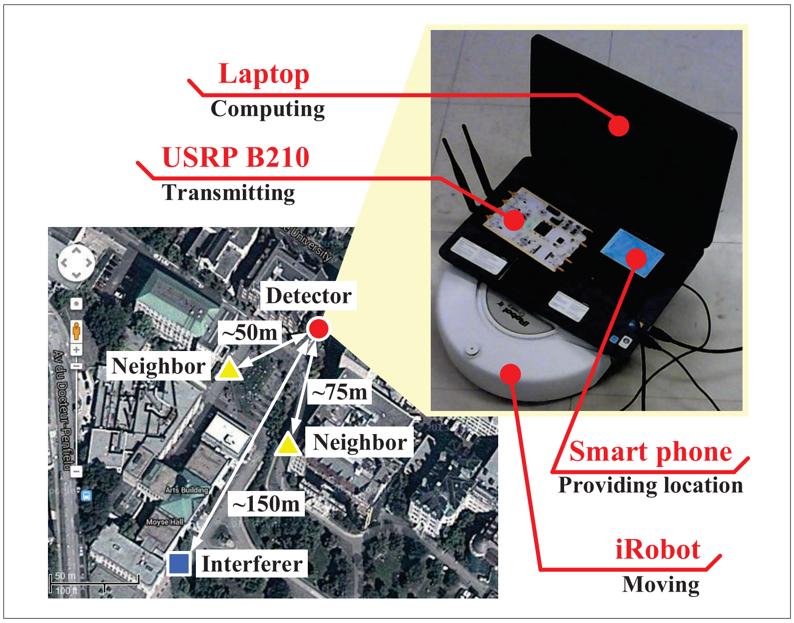

Figure 3. Testbed node and experiment scenario.

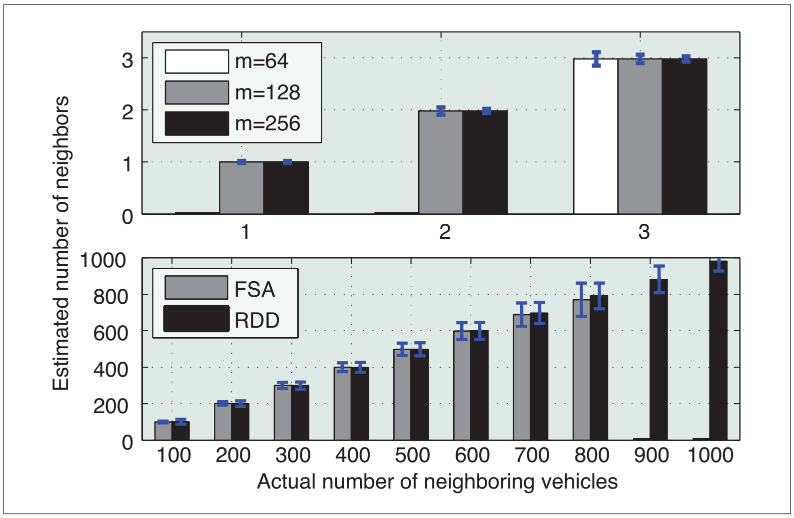

Figure 4. Performance of accuracy in the low-density experiment scenario (top) and high-density simulation scenario (bottom).

EXPERIMENT SETTING

In the experiments, the center frequency of the selected channel is set as 5.9 GHz, which is Channel 180 in DSRC. The bandwidth of this channel is 10 MHz according to the DSRC standard. The data rate is set as 3Mb/s [6]. The frequency stability is 2 ppm provided by USRP B210. The transmission power is set as 10 dBm [12]. In addition, the major parameters are tested in different cases to understand their effects. For example, we test different numbers of subchannels at m = 64, 128, 256, and 512 and different custom ranges at R = 100, 200, and 300 m.

In the four-node testbed, one node acts as the detector, and the other three nodes act as either neighbors or interferers depending on their distance to the detector. All iRobots randomly move in the campus area, so three scenarios can be evaluated, that is, the number of neighbors at N = 1, 2, 3. For instance, a Google Earth snapshot of our experiment is shown in Fig. 3. In this scenario, the custom range is R = 100 m. Two nodes are within the custom range, so they are neighbors. However, one node is outside the 100 m range. This node keeps sending DSRC packets to interfere with the RDD feedback.

All request tuples are logged in the detector. All feedback, interference information, and location information are logged in neighbor/interferer nodes. Totally, more than 500,000 detection messages, 1,000,000 feedback signals, and 100,000 interference packets are logged in our experiments. These logs are our ground truth to assess the RDD results.

EXPERIMENT RESULTS

Feasibility of RDD: We show the estimation accuracy of RDD in Fig. 4 under different settings such as N = 1, 2, 3 and m = 64, 128, 256. From Fig. 4, we find that except for m = 64, the estimated numbers $\tilde{N}$ (Y-axis) are always close to the actual numbers of neighbors N (X-axis), while the standard deviations (error bar) are extremely tiny compared to the actual numbers. *This result demonstrates the feasibility of RDD in low-density scenarios*. Moreover, the exception case happens at m = 64. This exception comes from the interference of DSRC packets, whose number of subchannels is also 64. When both RDD and DSRC adopt 64 subchannels, the filter design fails to separate the DSRC packet and the RDD feedback. Hence, the setting of m = 64 is unavailable in RDD.

SIMULATION SETTINGS

Since our testbed has only four nodes, the experiments of RDD are only conducted in low-density scenarios. In order to evaluate the RDD performance in high-density scenarios, we conduct extensive real-data-driven simulations.

Data synthetic: Based on the feedback and interference packets logged in our experiments, we synthesize the data for simulations. In order to imitate high-density scenarios, we superpose multiple time-domain feedback signals into one, which is equivalent to multiple neighbors responding simultaneously. Consequently, we have various numbers of neighbors N ranging from 1 to 1000.

We comparatively study two methods.

Frame Slotted ALOHA [11]: FSA uses time-divided slots to collect the bitstream and the UPE estimator to estimate the number of neighbors.

Real-Time Density Detection: RDD obtains the bitstream through the frequency-divided subchannels and adopts the UPE estimator for density estimation.

The default settings of RDD include: the number of subchannels is set as m = 256, and the duration of feedback is set as 64 μs. In addition, the custom range is set as R = 300m [12].

SIMULATION RESULTS

Estimation accuracy is one of the most important metrics for density detection because accurate density information can lead to the right decision in vehicular applications. Since RDD adopts 256 subchannels by default, for fairness, feedback in FSA is assigned 256 time slots. In this simulation, we test the cases in which the number of numbers are from N = 100 to 1000. Their average estimation results and corresponding standard deviations are plotted in Fig. 4. First, we observe that RDD achieves accurate estimation in all cases, where its average estimation results are always close to the ground truth. However, FSA cannot estimate when $N > 800$ due to the limitation of the UPE estimator. Second, we observe the interesting variation trend of standard deviations. When N is small, the standard deviations of FSA is a little better than that of RDD. When N is big, RDD is better than FSA. The transition happens at $N = 600$. Combining the average estimation and standard deviation, RDD outperforms FSA in terms of estimation accuracy.

Time cost is the main goal pursued in this work. To evaluate the performance of time cost, we show the comparison results of FSA and RDD in Fig. 5. In this simulation, for FSA, we plot the minimal time costs needed for density estimation, while for RDD, we plot the average time cost with the initial number of subchannels m set as 128. We observe that the increase of FSA is linear because the UPE estimator requires more slots for a larger amount of estimation. In contrast, benefiting from the frequency-divided subchannels, RDD achieves a time-saving estimation, which stays at the 100 ms level with relatively small increase with the number of neighbors. The small increases are generated by the divisor tuning and multiple detections. This result demonstrates that RDD can detect the density within a very short duration, approximating real time.

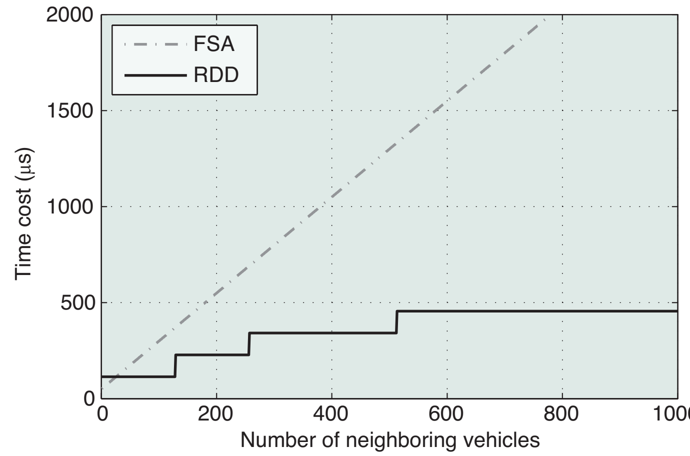

Figure 5. Comparison of the proposed RDD with the existing FSA method on the metric of time cost.

Conclusion

Motivated by real-time traffic control and data forwarding in connected vehicles, we present a novel RDD system to accelerate the local density detection. Different from existing time division concepts, RDD introduces a parallel detection of neighboring vehicles via frequency division, largely reducing the time cost. In addition, RDD is fully distributed, which does not require any roadside infrastructure. We implement the RDD system in USRPs and cover several practical design issues. The RDD system achieves 100-µs-level density detection with high accuracy.

We believe RDD has wider implications for density detection than explored in this work. Many issues need to be further investigated. For example, we will study the local density in different directions, which can benefit smart path planning. Second, based on RDD, we can design more aggressive upper-layer protocols to enhance the time-critical applications in connected vehicles. Moreover, RDD can be extended to other OFDM-based mobile networks such as fifth generation and narrowband Internet of Things.

Acknowledgments

This work is partly supported by the National Key R&D Program of China (No. 2017YFB1002000), NSFC (No. 61672349), Joint Key Project of NSFC (No. U1736207), the Macao Science and Technology Development Fund (No. 138/2016/A3), the Program of Introducing Talents of Discipline to Universities and the Open Fund of the State Key Laboratory of Software Development Environment under grant SKLSDE-2017ZX-09, and the Project of Experimental Verification of the Basic Commonness and Key Technical Standards of the Industrial Internet Network Architecture.

References

[1] S. H. Ahmed *et al.*, "Codie: Controlled Data and Interest Evaluation in Vehicular Named Data Networks," *IEEE Trans. Vehic. Tech.*, vol. 65, no. 6, 2016, pp. 3954–63.

[2] M. N. Mejri and J. Ben-Othman, "GDVAN: A New Greedy Behavior Attack Detection Algorithm for VANETs," *IEEE Trans. Mobile Computing*, vol. 16, no. 3, 2017, pp. 759–71.

[3] Y. Zhang *et al.*, "Sovcan: Safety-Oriented Vehicular Controller Area Network," *IEEE Commun. Mag.*, vol. 55, no. 8, Aug. 2017, pp. 94–99.

[4] L. Kong *et al.*, "Millimeter-Wave Wireless Communications for IoT-Cloud Supported Autonomous Vehicles: Overview, Design, and Challenges," *IEEE Commun. Mag.*, vol. 55, no. 1, Jan. 2017, pp. 62–68.

[5] A. Jovicic, J. Li, and T. Richardson, "Visible Light Communication: Opportunities, Challenges, and the Path to Market," *IEEE Commun. Mag.*, vol. 51, no. 12, Dec. 2013, pp. 26–32.

[6] J. B. Kenney, "Dedicated Short-Range Communications (DSRC) Standards in the United States," *Proc. IEEE*, vol. 99, no. 7, 2011, pp. 1162–82.

[7] Z. Li *et al.*, "Achievable Rate Maximization for Cognitive Hybrid Satellite-Terrestrial Networks with AF-Relays," *IEEE JSAC*, vol. 36, no. 2, 2018, pp. 304–13.

[8] M.Torrent-Moreno *et al.*, "Vehicle-to-Vehicle Communication: Fair Transmit Power Control for Safety-Critical Information," *IEEE Trans. Vehic. Tech.*, vol. 58, no. 7, 2009, pp. 3684–3703.

[9] X. Ma and X. Chen, "Performance Analysis of IEEE 802.11 Broadcast Scheme in Ad Hoc Wireless LANs," *IEEE Trans. Vehic. Tech.*, vol. 57, no. 6, 2008, pp. 3757–68.

[10] K. Wang, X. Mao, and Y. Liu, "Blinddate: A Neighbor Discovery Protocol," *IEEE Trans. Parallel and Distributed Systems*, vol. 26, no. 4, 2015, pp. 949–59.

[11] M. Kodialam and T. Nandagopal, "Fast and Reliable Estimation Schemes in RFID Systems," *Proc. ACM MOBICOM*, 2006.

[12] F. Bai, D. D. Stancil, and H. Krishnan, "Toward Understanding Characteristics of Dedicated Short Range Communications (DSRC) from a Perspective of Vehicular Network Engineers," *Proc. ACM MOBICOM*, 2010.

[13] W. Du *et al.*, "From Rateless to Distanceless: Enabling Sparse Sensor Network Deployment in Large Areas," *IEEE/ACM Trans. Networking*, vol. 22, no. 4, 2016, pp. 2498–2511.

[14] D. B. Rawat *et al.*, "Enhancing VANET Performance by Joint Adaptation of Transmission Power and Contention Window Size," *IEEE Trans. Parallel and Distributed Systems*, vol. 22, no. 9, 2011, pp. 1528–35.

[15] Y. Zhuang *et al.*, "Time and Location-Critical Emergency Message Dissemination for Vehicular Ad-Hoc Networks, *IEEE JSAC*, vol. 29, no. 1, 2011, pp. 187–96.

Biography

Linghe Kong [SM] (linghe.kong@cs.sjtu.edu.cn) is currently a research professor with the Department of Computer Science and Engineering at Shanghai Jiao Tong University. Before that, he was a postdoctoral researcher at Columbia University and McGill University. He received his Ph.D. degree from Shanghai Jiao Tong University in 2012, his Master's degree from Telecom

SudParis in 2007, and his B.E. degree from Xidian University in 2005. His research interests include wireless communication, mobile computing, and the Internet of Things.

Guangtao Xue (gt_xue@sjtu.edu.cn) [M] received his Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China, in 2004. He is a professor in the Department of Computer Science and Engineering at Shanghai Jiao Tong University. His research interests include wireless networks, mobile computing, and big data. He is a member of the IEEE Computer and Communication Societies.

Kayhan Zara Ghafoor (kayhan@ieee.org) is currently a tenure-track associate professor with the Department of Computer Science at Cihan University-Erbil. Before that, he was a visiting researcher at University Technology Malaysia for six months and served as a postdoctoral fellow for two years at Shanghai Jiao Tong University. He received his Ph.D. degree in wireless networks from University Technology Malaysia in 2012. He has published over 55 scientific/research papers in ISI/Scopus indexed international journals and conferences.

Rasheed Hussain (r.hussain@innopolis.ru) is currently working at Secure System and Network Engineering (SNE), Innopolis University, Russia, as an assistant professor and the lead researcher in the field of information security and privacy. At Innopolis University, we have started a new Master's level degree program, SNE, which is a replica of the SNE program at the University of Amsterdam.

Hao Sheng (shenghao@buaa.edu.cn) received his B.S. and Ph.D. degrees from the School of Computer Science and Engineering, Beihang University, China, in 2003 and 2009, respectively. Now he is an associate professor in the School of Computer Science and Engineering, Beihang University. He is working on computer vision, pattern recognition, and intelligent transportation systems. He is the corresponding author of this article.

Peng Zeng (zp@sia.cn) received his B.S. degree in computer science from Shandong University in 1998 and his Ph.D. degree in mechatronic engineering from Shenyang Institute of Automation (SIA), Chinese Academy of Sciences (CAS), in 2005. Currently, he is a professor and Ph.D. supervisor at SIA, CAS. His research interests include industrial communication, smart grids, demand response, and wireless sensor networks. He is a member of IEC TC65 WG16 and a member of the SP100 Standard Committee, ISA.