

Connected-Vehicle Data Exchanges and Positioning Computing Based on the Publish-Subscribe Paradigm

Kachane Sonklin, Charles Wang, Dhammika Jayalath, Yanming Feng

School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, Australia

Email: k.sonklin@qut.edu.au, cc.wang@qut.edu.au, dhammika.jayalath@qut.edu.au, y.feng@qut.edu.au

How to cite this paper: Sonklin, K., Wang, C., Jayalath, D. and Feng, Y.M. (2019) Connected-Vehicle Data Exchanges and Positioning Computing Based on the Publish-Subscribe Paradigm. *Journal of Computer and Communications*, 7, 82-93.
<https://doi.org/10.4236/jcc.2019.710008>

Received: September 9, 2019

Accepted: October 9, 2019

Published: October 16, 2019

Abstract

Connected vehicles for safety and traffic efficient applications require device-to-device connections supporting one-to-many and many-to-many communication, precise absolute and relative positioning and distributed computing. Currently, the 5.9 GHz Dedicated Short Range Communications (DSRC) and 4G-Long-Term Evolution (LTE) are available for connected vehicle services. But both have limitations in reliability or latency over large scale field operational tests and deployment. This paper proposes the device-to-device (D2D) connectivity framework based on publish-subscribe architecture, with Message Queue Telemetry Transport (MQTT) protocol. With the publish-subscribe communication paradigm, road mobile users can exchange data and information in moderate latency and high reliability manner, having the potential to support many Vehicle to Everything (V2X) applications, including vehicle to vehicle (V2V), vehicle to roadside infrastructure (V2I), and vehicle to bicycle (V2B). The D2D data exchanges also facilitate computing for absolute and relative precise real-time kinematic (RTK) positioning. Vehicular experiments were conducted to evaluate the performance of the proposed publish-subscribe MQTT protocols in term of latency and reliability. The latency of data exchanges is measured by One-trip-time (OTT) and the reliability is measured by the packet loss rate (PLR). Our results show that the latency of GNSS raw data exchanges between vehicles through 4G cellular networks at the rate of 10 Hz and the data rates of 10 kbps are less than 300 ms while the reliability is over 96%. Vehicular positioning experiments have also shown that vehicles can exchange raw GNSS data and complete moving-base RTK positioning with the positioning availability of 98%.

Keywords

Connected Vehicle, Publish-Subscribe Paradigm, IoT Protocol

1. Introduction

A connected vehicle includes the different communication devices (embedded or portable) present in the vehicle, that enable in-car connectivity with other devices present in the vehicle and/or enable connection of the vehicle to external devices, networks, applications, and services. Commercial and consumer applications of connected vehicles include everything from fleet management, emergence assistance, traffic safety and efficiency, infotainment, parking assistance, roadside assistance, remote diagnostics, and telematics. Global Navigation Satellite Systems (GNSS)-based positioning and navigation and precise digital maps are essential parts of the connected vehicle services. In recent years, more professionally connected vehicles include the concepts of Vehicle-to-Everything (known as V2X) interactions, such as Vehicle-to-Vehicle (V2V), Vehicle-to-People (V2P) and Vehicle-to-Network (infrastructure) data or message exchanges. In the V2X scenarios, it requires device-to-device connections supporting one-to-many and many-to-many communications. The mobile devices/platforms often provide data streams and images to the servers or other devices instead of information or context-aware messages. The computation or decision making should be completed at the mobile device edge in real-time instead of at a server with a delay. V2X communications requires high timeliness, low round-trip time (RTT) latency, and high user scalability. In support of wireless connectivity among vehicle-based devices, and between fixed roadside devices and vehicle-based devices, the currently deployable technologies define the hardware and services operating from the application layer down to the physical layer. These include the 5.9 GHz Dedicated Short-Range Communications (DSRC)/IEEE Standard for Wireless Access in Vehicular Environments (WAVE), and 4G-Long-Term Evolution (LTE). The developing technologies include LTE-Direct and LTE-Vehicle and 5th generation cellular technology (5G) Ultra-reliable Machine-Type communications (uMTC), which are also hardware dependent.

The Internet-based vehicular connections have been implemented based on LTE mobile broadband services for the commercial and consumer purposes such as vehicle fleet tracking and management and route guidance. These connections are implemented in the application-layer, which is on top of given physical-layer and transport layers. There are a number of advantages of using application-layer solutions for vehicular connections. Firstly, it can get benefit from widely available mobile devices for implementations, for instance, smart-phone devices. Secondly, it is a software solution, which can be easily implemented and easily to updated, on top to hardware. Thirdly, the solution can be easily migrated from the current 4G LTE MBB services to the future 5G extreme MBB services. Overall, it has potential to support many connected vehicle applications. However, the existing connected vehicle for fleet management is mainly two-way connection between vehicle and a server and not for data exchanges between vehicles. For V2X applications, the main concern is about the perfor-

mance of the application layer connections in terms of latency, reliability and scalability for connected vehicle applications. The consequent question is to what degree the application layer solution can support different V2X applications. Are there more effective application protocols for better performance? For instance, many Internet based connections for vehicles are based on the HTTP client-server architecture. The client-server architecture is a centralized model, in which clients' requests are sent to the server to receive the information. Such client-server architecture performs well when data is stored in a central server, such as a file and a database server, or contents are accessible from third parties through the server. With the client-server model, a single server can accommodate various services to many client requests and the client can send requests to multiple servers through a network.

The client-server architecture consists of three major components including hardware, software, and communication middleware, which can communicate with each other. Each component of the client-server comprises: 1) hardware is related to the client and the server that client usually requests the services to the server, which is the computer that provides the services and the responses to every client requests; 2) software is used to response the requirement of users (clients); 3) communication middle-ware is used to transmit information among the client and the server across a network [1]. However, the client-server architecture has some disadvantages. The server side will take more overheads to serve thousands of simultaneous client requests, due to the Request/Response communication. **Figure 1** shows the centralized structure. Every HTTP request from the client will have to create a new TCP connection, and the client must wait until the responses to the previous HTTP requests have been completed [2]. These overheads lead to the reduced performance of the system, such as high latency. In addition, the client-server architecture has a scalability problem in a large-scale network. Normally, more servers are introduced to meet the demand of a large number of clients. Because of these problems, the client-server architecture is not considered suitable for supporting Device-to-Device (D2D) applications in terms of scalability and timeliness.

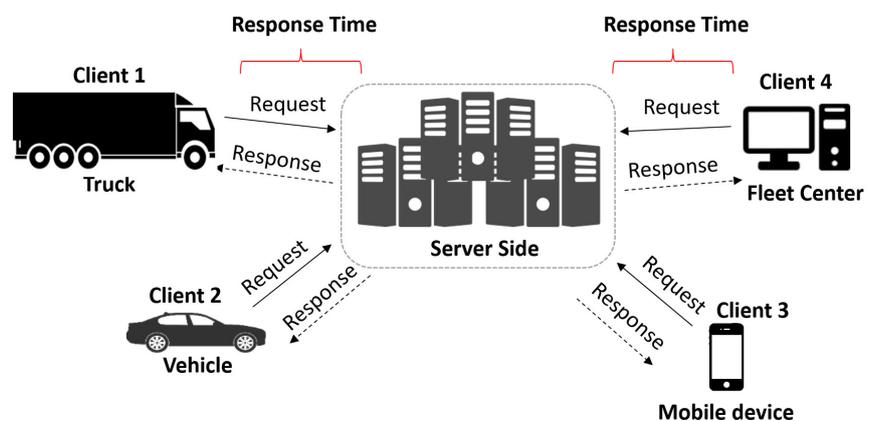


Figure 1. Centralized client-server for mobile device connections.

To overcome the limitations of the client-server architecture and difficulties in the current connected vehicle applications, we extend the current two-way communications to unidirectional communications to support time critical Device-to-Device (D2D) data exchange based on Internet connections. Specifically, the D2D connection is to introduce the publish-subscribe communication paradigm in the application layer to support the proposed D2D data exchanges with the mobile broadband services (MBB) under the current 4G-LTE networks and the future 5G extreme Mobile Broadband (eMBB) services. In the following, the D2D data exchanges based on the publish-subscribe paradigm is outlined, followed by several use cases and requirements for connected device communications. In Section 3, we will discuss about the D2D GNSS data exchanges and positioning performance with MQTT protocols and the experimental results are presented to demonstrate the performance in the vehicle connection case. Section 4 is the summary of the findings of the paper.

2. Device-to-Device Data Exchanges Based on the Publish-Subscribe Paradigm

2.1. Device-to-Device Vehicle Use Case

The term “Device-to-Device (D2D)” refers to direct communication between two mobile devices. D2D was initially proposed as a new paradigm in cellular networks. LTE-Direct is an innovative D2D technology that enables mobile devices and applications to passively discover and interact with the world around them in a privacy sensitive and power efficient manner [3]. LTE-Direct creates new proximity service opportunities for the entire mobile industry in social networking, venue services, loyalty services, local advertising, and much more. One of its main benefits is the ultra-low latency in communication due to a shorter signal traversal path through the nearest base station or access nodes. Along the same line, LTE-Vehicle enables Vehicle-to Everything (V2X) communication and is considered to be one of the optimal choices for effective connected vehicles. On the other hand, the 5G uMTC services provide ultra-reliable and low latency communication for high demanding applications. V2X is one of most important applications of uMTC services in the future. LTE-Direct, LTE-Vehicle and 5G uMTC services will certainly enhance the connected vehicle applications and empower new use cases. However, in the application-layer D2D connections, devices are IP-based, generally including smartphones, computers, mobile devices on-board vehicles, roadside devices, network servers and Internet of Things end-devices. All devices have location sensors. For the connected vehicle applications, D2D connections can mean vehicle-to-vehicle, vehicle-to-infrastructure, vehicle-to-network and vehicle-to-people. More specifically, the D2D connections can include, but not to limit to the following use cases:

- **V2V and V2I safety message and location data exchanges for cooperative awareness:** The data are required to be transmitted at the rate between 1 and 10 Hz. The payload of each message ranges from 60 to 1500 Bytes [4], af-

ected by the road and traffic conditions. The message may also include GNSS raw measurements in the standards Radio Technical Commission for Maritime Services (RTCM) and Basic Safety Messages (BSM) in the SAEJ2735 standards [5]. While the BSM message size (part 1) is less than 60 Bytes, the message size for RTCM data alone can range from 300 to 1200 Bytes for 1 to 4 GNSS constellations.

- **V2N and V2I traffic data exchanges for traffic efficiency:** This transmission does not have strict delay or reliability requirements, since there is no need for prompt action at the vehicle side. Each vehicle updates the Traffic Management server (uplink) every few seconds with location, status and road information, which are required for the more efficient route selection. The payload of this type of message is up to 1500 Bytes [4]. The response from the Traffic Management servers (downlink) includes digital map title updates, which may be in the size of a few MBytes, but the transmission is event-driven and is not time critical.
- **V2P and P2V data exchanges for vulnerable road users (VRUs):** This use case is similar to the cooperative awareness category (*i.e.*, latency and reliability requirements) with the difference in that the destination device is a user equipment (e.g., smartphone), where the needed information is less, e.g., payload sizes from 60 to 120 Bytes [4]. However, RTCM data exchanges between vehicles and VRUs can also be added to enable more precise relative positioning.
- **Location-based vehicle and personal tracking:** This offers a tracking platform within a group of people and vehicles in the same company without a centralized Location-based services (LBS) server. In other words, a mobile app can be designed to track family members. The data exchange rate of 1 Hz is sufficiently frequent and the message size should be about 60 to 120 Bytes.
- **Device to Server GNSS raw data collections and Server to Device distribution in RTCM formats:** This is similar to the V2V RTCM data exchanges, but the data are collected or transmitted at the frequency of 1 Hz or lower. If the payload of each message includes all raw measurements from multiple constellations, the message size can be 1200 to 1500 Bytes.

Overall, in the above D2D use cases, the messages of 60 to 1500 Bytes are required to be transmitted at the frequency of up to 10 Hz. While the LTE Direct and 5G and uMTC technologies may support all use cases of D2D use cases in years of the future, many D2D use cases may also be deployed right now through the application protocols based on the publish-subscribe communication paradigm. Therefore, in the following subsections, we introduce the publish-subscribe communication paradigm, and propose the application-layer D2D framework, which can be implemented with the current Internet connections.

2.2. Publish-Subscribe Paradigm for D2D Connections

The publish-subscribe paradigm enables unidirectional communication from a publisher to one or more subscribers. In software architecture, publish-subscribe

is a messaging pattern where senders of messages, are called publishers, and receivers are called subscribers. Publishers do not program the messages to be sent directly to specific subscribers, but categorize published messages into classes, in which subscribers express interest in one or more classes and receive messages that are of their interest. Publishers and subscribers communicate for the interested messages without knowledge of subscribers or publishers. The Publish-subscribe model enables event-driven architectures and asynchronous event notifications, while improving performance, reliability and scalability. **Figure 2** illustrates the publish-subscribe model with three publishers and three subscribers for the commonly interested messages of mobile phone, vehicles and road information. Vehicle users declare their interests in both roadside data and vehicles data to the publisher-subscriber server. When the server has new data available in that topic, the server platform pushes the data or messages to interested vehicle subscribers. This process will repeat for all publishers and subscribers. The publish-subscribe model is well suited for many D2D connection deployments as a device can be both a publisher and subscriber and can communicate with each other through the publisher-subscriber server. D2D connection can benefit from the loose coupling between communication endpoints, low latency and better scalability. This is because the model can leverage parallelism and multi-cast capabilities of the underlying transport network. It supports point-to-multi-point and multi-point-to-multi-point communications. In other words, one device can choose to receive data from multiple devices or data sources. Although a server is needed for exchange of data, it can be set close to data sources and many can be used in the distributed manner.

In the publish-subscribe model, subscribers typically receive only a subset of the total messages published. The process of selecting messages for reception and processing is called filtering. Three publish-subscribe schemes, including topic-based, content-based and type-based, have been researched to identify the events of interest [6]. How these schemes may be implemented for various V2X applications is an important issue, deserving dedicated research attentions. With various publish-subscribe schemes, flexibility of contents and dynamic contents are offered in order to exchange data between a large numbers of entities without establishing any contracts or knowing each other [7]. However, how to implement a filtering scheme to effectively support a particular D2D use case is a new research problem that requires specific attention in due course.

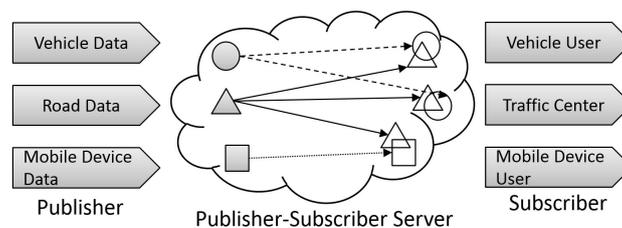


Figure 2. Illustration of the publish-subscribe model with three publishers and three subscribers.

3. D2D GNSS Data Exchanges and Positioning Performance with MQTT Protocols

3.1. Message Queue Telemetry Transport (MQTT) Protocol

MQTT is a message centric wire protocol designed for Machine-to-Machine (M2M) communications that provides lightweight, simple implementation, open standard, reliability, and efficiency with regards to processor, memory, and network resources. The publish-subscribe communication model is used to transfer the telemetry-data in the messages format from publisher (device), along restricted environments and unreliable networks, to brokers across TCP/IP. Recently, the two main versions of MQTT are mentioned as follows: 1) MQTT version 3.1 has been developed to transmit data over Transmission Control Protocol/Internet Protocol (TCP/IP) and it is defined as the basic transport and network service and 2) MQTT-SN which has been developed for transmitting data through User Datagram Protocol (UDP) over low-bandwidth wireless communication networks [8] [9]. For the D2D data exchange use cases, QoS 1 is the default mode of data transfer. With QoS 1 mode, a message is sent at least once, and it will be retransmitted until a broker receives acknowledgement from subscriber by using PUBACK. There have been several MQTT server products in use. According to the benchmark evaluation about MQTT scalability, JoramMQ and Mosquitto have shown desirable performance in terms of basic measurements (latency, CPU and message rate) [10]. In our evaluation for the V2V RTCM data exchange and relative positioning in the next subsection, we choose the widely-used open source Mosquitto [11]. **Figure 3** illustrates the basic model of MQTT. Traffic, road and POI data publishers publish messages with specifying the topic names to a central broker (server). When the broker receives the published messages from the publishers, it then transmits (publishes) these messages to the other subscribers based on their subscription topics.

3.2. V2V RTCM Data Exchange OTT, PLR and Relative Positioning Results

To verify the concept of the D2D publish-subscribe model, we perform the experiments with the MQTT protocol in device-to-device data exchange. This experiment is to compare the RTCM data exchange RTT and PLR between vehicle

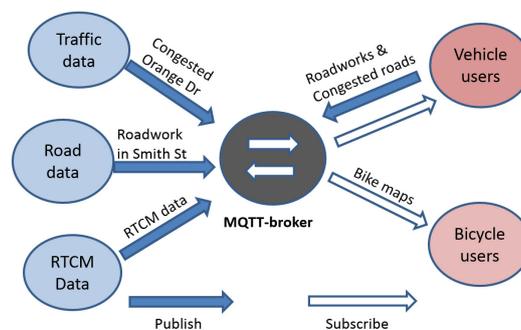


Figure 3. The publish-subscribe model of MQTT.

and base station and between two vehicles, and their Real Time Kinematic positioning performance. The experiment and results are presented in the following subsection.

This experiment demonstrates V2V RTCM data exchanges and RTK positioning performance. One vehicle was used to host two U-Blox ZED-F9P receivers/laptops. **Figure 4** shows the antennas on the roof of the testing vehicles and two laptops on the back seats of the vehicle. Both laptops access the RTCM data separately from the Geoscience Australian data server for the nearby reference station CLEV at 1 Hz, while both receivers/laptops collect GNSS data at 10 Hz. The experiment implements two cases for each vehicle to obtain relative position states. In the first case, Laptop A publishes the RTCM data at 10 Hz using MQTT protocols and Laptop B subscribes the data and runs RTK software to generate RTK solutions for both receivers. This means that one vehicle can directly determine the position states relative to another. In the second case, each Laptop runs RTK software to generate own position states and publishes the NMEA data to the other Laptop. In both cases, the time delays will occur due to the RTK computations and V2V data transmission. While the GNSS states update at the standard 10 Hz GNSS time tags, relative positions can only be available after receiving/obtaining the position data from the other vehicle.

To determine the data exchange latency One-Trip Time (OTT) from one vehicle/Laptop to another, a timestamp is added to the data message with the same GNSS time tag obtained from own receiver/laptop and received the other receiver/laptop. For any time epoch t , we denote GNSS time tag as $GT(t)$, the timestamp of the data message of $GT(t)$ from receiver A/Laptop A as TS_a , and the timestamp of the data message of $GT(t)$ from the receiver B/Laptop B as TS_b . The time offsets between GNSS time and the Laptop time are given as follows:

$$\Delta t_a = TS_a - GT(t) \quad (1)$$

$$\Delta t_b = TS_b - GT(t) \quad (2)$$

When the Laptop A's data message of $GT(t)$ arrives in Laptop B, there is a timestamp TS_{ab} . The latency of the data message from the Laptop A is given as follows:

$$dt_{ab} = TS_{ab} - TS_b \quad (3)$$

The offsets Δt_a and Δt_b are not necessary constant over a period of time due to delays from application layers to hardware clock in the user end and computing delays. **Table 1** summarizes the mean and standard deviations of the above time

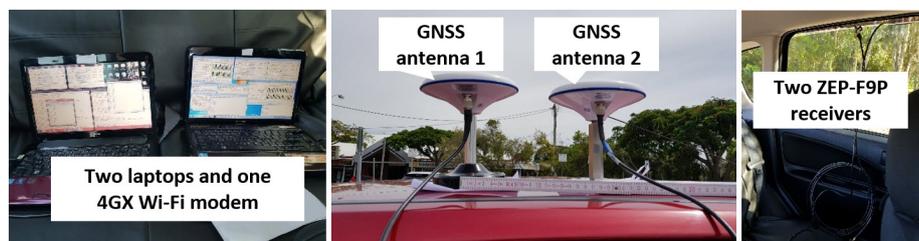


Figure 4. Vehicle experimental setup with GNSS antennas and receivers.

offsets and delays and PLR for the delays over 300 ms. The results show that the mean OTT latency of RTCM data exchanges from one vehicle to another is less than 200 ms, and PLR for the delays over 300 ms is 4% and delays beyond one second are less than 1%. It also is observed that the variation of time offsets and delays are notably large due to the use of the Laptop-receiver connection settings for the experiments. This indicates the importance of time synchronization between two mobile terminals as the vehicle states must be predicted from the last observed update to the current time update for making collision avoidance decisions. Referring to the work [12], with GNSS-1pps chipset, the mobile terminal clock can be synchronized to GNSS time to the precision of microsecond level. The purpose of the data exchange between vehicles is to determine the relative solutions. With the raw data from the Laptop A, the Laptop B have two sets of relative position data computed from stationary-base (SB) RTK and moving-based (MB) RTK processing.

Figure 5 illustrates the OTT for RTCM and NMEA data between two receivers. It can be stated that the average time offset of RTCM data is slightly greater than the NMEA data.

Figure 6 illustrates the distance and height difference between two antennas (two receivers). It shows that both SB and MB RTK solution availability. Because the antennas were set at 20 cm apart and in the same height, the outages are identifiable through the distance and height difference. As summarized in **Table 2**, the RTK results shown centimeter accuracy and availability after removing the outages specified as the difference beyond ± 10 centimeters in this context. MB

Table 1. OTT, Delayed messages and PLR obtained with the exchange of data between receivers/laptops.

	FREQ ^a	Data message obtained/received	Average time offset/delay (ms)	Standard deviation (ms)	Delayed message rate beyond 300 ms	Total message transmitted	PLR ^b (>300 ms)	PLR (>1000 ms)
Δt_a	10 Hz	NMEA	7710.00	32.66	NA	NA	NA ^c	NA
Δt_b	10 Hz	NMEA	7960.00	43.04	NA	NA	NA	NA
dt_{ab}	10 Hz	RTCM	192.72	101.99	839	18,992	4.00%	0.29%
dt_{ba}	10 Hz	NMEA	123.98	62.45	191	18,992	1.01%	0.02%

a. FREQ = Frequency. b. PLR (Packet loss rate) >300, >1000 = The delay message is greater than 300 and 1000 milliseconds. c. NA = Not applicable.

Table 2. Centimeter accuracy and availability after removing the outages specified as the difference beyond ± 10 centimeters.

	Stationary Base (SB)			Moving Base (MB)		
	Mean (cm)	STD (cm)	Availability	Mean (cm)	STD (cm)	Availability
Up-difference	0.640	1.381	91.12%	1.120	1.976	98.01%
Distance	-0.027	1.969	96.99%	0.164	0.689	98.38%

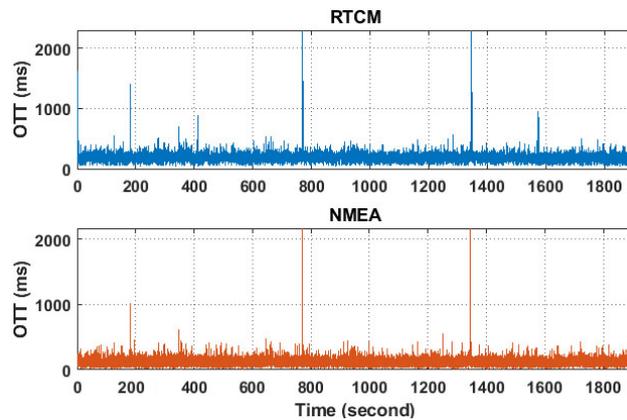


Figure 5. Illustration of the OTT for RTCM and NMEA data between two laptops.

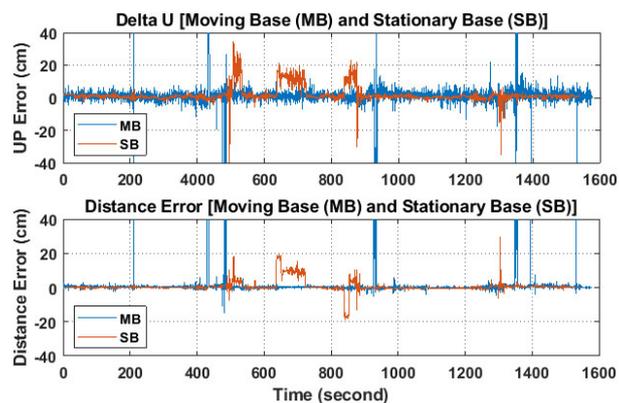


Figure 6. Illustration and comparison of distance and height difference between two antennas, showing RTK solution accuracy and availability.

RTK results in high availability or shorter solution outages, showing some advantage of moving base RTK. Detection and identification of RTK solution outages are the integrity determination issue, which is important for connected vehicle safety applications [13], but being considered out of the scope of this work.

4. Conclusions

Internet-based vehicle connections can address many V2X applications. But the standard client-server architecture is tightly coupling with space, structure and time constraints and has limitations in support of Device to Device (D2D) applications in terms of latency, reliability and scalability performance. These are important requirements for high-demanding connected vehicle applications, such as Vehicle-to-Everything (V2X) applications. Currently, the dedicated short-range communication (DSRC) and 4G-LTE are two widely used candidate schemes for connected vehicle applications. However, the recent 4G-LTE experimental results have shown that the average RTTs are 300 to 400 ms for the vehicle speeds from 60 to 120 Km/h. DSRC latency can be well within the lowest requirement of 100 ms, but their PLRs are significantly degraded when the inter-vehicle distances are over 200 m. In this contribution, a D2D data exchange

framework based on publish-subscribe architecture has been proposed. Publishers and subscribers communicate data of interest or messages without knowledge of subscribers or publishers. Publish-subscribe model enables event-driven architectures and asynchronous event notifications, while improving performance, reliability and scalability. The publish-subscribe model can support many D2D deployments as a device can be both a publisher and subscriber, and can communicate with each other through a publisher-subscriber server. The paper also introduced the well-established publish-subscribe application protocol: Message Queue Telemetry Transport (MQTT) for implementation of the proposed D2D based on the publish-subscribe model.

To demonstrate the performance of D2D GNSS RTCM data exchanges and RTK positioning performance, we have used two separate sets of GNSS receivers on the same testing vehicle. This setting allows for the assessment of relative positioning performance with constant antenna distance and constant (or zero) height difference with relative RTK results. The results have shown the mean OTT latency of RTCM data exchanges from one vehicle to another is less than 200 ms, and PLR for the delays over 300 ms is 4%. It is also observed that latency measurements have an uncertainty of tens of milliseconds due to the cross layer hardware connections and computing time variations. Relative positioning results have shown both SB and MB RTK solution accuracy of centimeters and availability of over 98% in the testing routes in a residential area. Results also indicate the needs for detection and identifications of RTK solution outages which should be excluded or bridged for connected vehicle safety applications.

Conflicts of Interest

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

References

- [1] Barkai, D. (2000) An Introduction to Peer-to-Peer Computing. *Intel Developer update Magazine*, 1-7.
- [2] deSaxc, H., Oprescu, I. and Chen, Y. (2015) Is HTTP/2 Really Faster than HTTP/1.1? 2015 *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs)*, 293-299. <https://doi.org/10.1109/infcomw.2015.7179400>
- [3] Wang, Y., Wei, L., Vasilakos, A.V. and Jin, Q. (2017) Device-to-Device Based Mobile Social Networking in Proximity (MSNP) on Smartphones: Framework, Challenges and Prototype. *Future Generation Computer Systems*, **74**, 241-253. <https://doi.org/10.1016/j.future.2015.10.020>
- [4] Boban, M., Kousaridas, A., Manolakis, K., Eichinger, J. and Xu, W. (2017) Use Cases, Requirements, and Design Considerations for 5G V2X. preprint arXiv:1712.01754.
- [5] Kenney, J.B. (2011) Dedicated Short-Range Communications (DSRC) Standards in the United States. *Proceedings of the IEEE*, **99**, 1162-1182. <https://doi.org/10.1109/jproc.2011.2132790>
- [6] Eugster, P.T., Felber, P.A., Guerraoui, R. and Kermarrec, A.M. (2003) The Many

- Faces of Publish/Subscribe. *ACM Comput. Surv.*, **35**, 114-131.
<https://doi.org/10.1145/857076.857078>
- [7] Ion, M., Russello, G. and Crispo, B. (2010) Supporting Publication and Subscription Confidentiality in Pub/Sub Networks. In: Jajodia, S. and Zhou, J., Eds., *Security and Privacy in Communication Networks*, Heidelberg: Springer, Berlin, 272-289.
https://doi.org/10.1007/978-3-642-16161-2_16
- [8] Stanford-Clark, A.J. and Wightwick, G.R. (2010) The Application of Publish/Subscribe Messaging to Environmental, Monitoring, and Control Systems. *IBM J. Res. Dev.*, **54**, 396-402. <https://doi.org/10.1147/jrd.2010.2050982>
- [9] Pereira, C. and Aguiar, A. (2014) Towards Efficient Mobile M2M Communications: Survey and Open Challenges. *Sensors*, **14**, 19582-19608.
<https://doi.org/10.3390/s141019582>
- [10] Scalagent. (2015) Benchmark of MQTT Servers.
http://www.scalagent.com/IMG/pdf/Benchmark_MQTT_servers-v1-1.pdf
- [11] Light, R.A. (2017) Mosquitto: Server and Client Implementation of the MQTT Protocol. *The Journal of Open Source Software*, **2**. <https://doi.org/10.21105/joss.00265>
- [12] Hasan, K.F., Feng, Y. and Tian, Y. (2018) GNSS Time Synchronization in Vehicular Ad-hoc Networks: Benefits and Feasibility. *IEEE Transportation Intelligent Systems*, **19**, 3915-3924. <https://doi.org/10.1109/tits.2017.2789291>
- [13] Feng, Y., Wang, C. and Karl, C. (2018) Determination of Required Positioning Integrity Parameters for Design of Vehicle Safety Applications. *Proceedings of the 2018 International Technical Meeting of the Institute of Navigation*, 129-141.
<https://doi.org/10.33012/2018.15556>