2017 31st International Conference on Advanced Information Networking and Applications Workshops

TOMS: TCP Context Migration Scheme for Efficient Data Services in Vehicular Networks

JunSik Jeong[*], Yiwen (Chris) Shen[†], Jaehoon (Paul) Jeong[‡], Tae (Tom) Oh[§], Junghyun Jun[¶] and Sang Hyuk Son[‖]
[*] Department of DMC Engineering, Sungkyunkwan University, Suwon, Republic of Korea
[†] Department of Computer Science & Engineering, Sungkyunkwan University, Suwon, Republic of Korea
[‡] Department of Interaction Science, Sungkyunkwan University, Suwon, Republic of Korea
[§] Department of Information Sciences & Technologies, Rochester Institute of Technology, USA
[¶] Department of Computer Science & Engineering, Indian Institute of Technology at Ropar, India
[‖] Department of Information & Communication Engineering, DGIST, Republic of Korea
Email: {jjs9915, chrisshen, pauljeong}@skku.edu, tom.oh@rit.edu, peterjun@iitrpr.ac.in, son@dgist.ac.kr

***Abstract*—The recent advances in wireless communication techniques have made it possible for fast-moving vehicles to download data from the Internet. For the reliable data upload and download, TCP can be used for vehicular networks. However, TCP requires the connection initialization using three-way handshaking for the data exchange between two end systems over the Internet. Thus, the efficient operation of TCP is important for data services in the vehicular networks. This paper proposes a method of TCP Context Migration Scheme (TOMS) for the enhancement of data services in vehicular networks. TOMS provides vehicles with proactive TCP connection initialization using a moving TCP proxy as a cluster head, which will have the Internet connectivity with a Road-Side Unit (RSU). A cluster member can initiate its TCP connection toward its corresponding TCP end-system (e.g., server and peer) via the TCP proxy within its cluster. The TCP proxy performs the TCP connection set-up for the sake of other cluster member vehicles and acknowledges the received TCP segments toward these vehicles. When the TCP proxy moves out of the communication range of the RSU, it transfers the TCP contexts of other vehicles to another vehicle, which will play the role of a TCP proxy through the proposed TCP context migration scheme. Also, the RSU works as a fixed TCP proxy for handling the acknowledgement of TCP segments and TCP timer handling (e.g., persist timer and keepalive timer) when there happens the disconnection between the moving proxy and the RSU. Thus, it is shown that our TOMS outperforms the legacy TCP in vehicular networks.**

***Index Terms*—Vehicular Networks, Vehicular TCP, Cooperative TCP, IPv6, VANET**

I. INTRODUCTION

As one of the most active research areas these days, the advanced vehicular ad hoc networks (VANETs) [1]–[5] can be used for data exchange in road networks where vehicles as moving networks (MNs) with multiple in-vehicle devices or hosts are inter-connected. Some of these services will want to be connected to the Internet. The vehicles are possible to connection to the Internet through Road-Side Units (RSUs). During car driving, you can enjoy the Internet services only under the communication the coverage of RSUs. Such a fundamental vehicular communication framework is referred to as the Drive-thru Internet [4][6].

By this characteristic, Vehicles and the Internet, which are two most prominent elements of our modern lives, has become ever more important [7]. Moreover, for the support of the VANET in road networks, the dedicated short-range communications (DSRC) has been standardized as IEEE 802.11p (now incorporated into IEEE 802.11-2012), which is an extension of IEEE 802.11a, considering the characteristics of vehicular networks, such as high-speed mobility and network fragmentation. For wireless access in vehicular environments (WAVE), the IEEE has standardized IEEE 1609 family standards, such as IEEE 1609.3 and 1609.4. The IEEE 1609 standards specify IPv6 as the network-layer protocol. With this trend, it is time to enable vehicular networking with IPv6 to let various Internet-based applications run on top of transport-layer protocols, such as TCP, UDP, and SCTP. IPv6 is suitable for a network layer in vehicular networks in that the protocol has abundant address space, autoconfiguration features, and protocol extension ability through extension headers.

Compared with the original wireless local area network (WLAN) scenarios, VANET is a much more challenging task [8] due to the high vehicle mobility, but it has much more predictable information. As reported in [4], the overall connectivity range of an RSU is around 500–600 m, which allows a connection time of 15–18 s to a vehicle moving at the velocity of 120 km/h. In reality, the number of RSUs deployed along the road cannot be enough for providing the ubiquitous coverage due to the high deployment and maintenance cost, particularly in a sparse populated region. Thus, cooperative intervehicle communications is required accordingly as a supplement to extend the coverage of RSUs in vehicular networks. For the fast and reliable data exchange in the vehicular networks, geographically adjacent should be collaborators, not channel competitors of all vehicles [9].

To the best of our knowledge, this paper is the first of TCP-proxy-based data send through TCP Context Migration Scheme (called TOMS). Vehicles can their TCP segments in a delay-tolerant way by using another vehicle close to the communication rage of an RSU as a proxy. Our TOMS allows vehicles to perform proactive TCP connection initialization by

978-1-5090-6231-7/17 $31.00

DOI 10.1109/WAINA.2017.109

using a moving TCP proxy as a cluster head, which reaches or is within the communication range of an RSU. A vehicle as a cluster member can start its TCP connection toward its corresponding TCP end-system (e.g., server and peer) via a moving TCP proxy within its cluster. This TCP proxy performs the TCP connection initialization for the safe of other cluster member vehicles, and also performs the acknowledgement of the received TCP segments toward these vehicles. When the TCP proxy moves out of the communication range of the RSU, the TCP contexts of other vehicles in the TCP proxy migrate to another vehicle to play the role of a moving TCP proxy through our proposed TCP context migration scheme. In addition, the RSU plays the role of a fixed TCP proxy for handling the acknowledgement of TCP segments and TCP timer handling (e.g., persist timer and keepalive timer) before the disconnection between the moving proxy and the RSU will occur. Therefore, our TOMS is a promising TCP enhancement for fast and reliable data services in vehicular networks.

The contributions of this paper are as follows:

- A vehicular network architecture of fast and reliable TCP service.
- A TCP context migration scheme using TCP proxies.

The rest of this paper is organized as follows. Section II summarizes related work relevant to TCP data services in vehicular networks. Section III describes the design of our TCP context migration scheme. Section IV, we discuss research issues for TCP data services. Section V, evaluates the performance of TOMS in terms of the TCP Setup Time. In Section VI, we conclude this paper along with future work.

II. RELATED WORK

VANETs have been extensively studied for the past decade. A vehicle in VANETs is expected to become one of clients for connecting cloud server. For faster transmit, The cooperative vehicular communications represent an effective approach to extend the RSUs coverage and have attracted an extensive research attention. Various cooperative schemes have been proposed in VANETs accordingly, which can be usually divided into two categories [10], i.e., V2I and V2V communications. Our work combines both kinds of cooperation and TCP Connection sharing each vehicular.

Zhou et al. proposed a cooperative Drive-thru Internet scheme called ChainCluster [9]. ChainCluster selects appropriate vehicles to form a linear cluster on the highway. The cluster members then cooperatively download the same content file, with each member retrieving one portion of the file, from the roadside infrastructure. With cluster members consecutively driving through the roadside infrastructure, the download of a single vehicle is virtually extended to that of a tandem of vehicles, which accordingly enhances the probability of successful file download significantly.

Zhu et al. proposed a multiple-vehicle protocol for collaborative data downloading by using network coding (NC) [11]. When multiple vehicles that are approaching each other have a common interest in certain data, they can collaboratively download the data from an RSU to significantly reduce their download time.

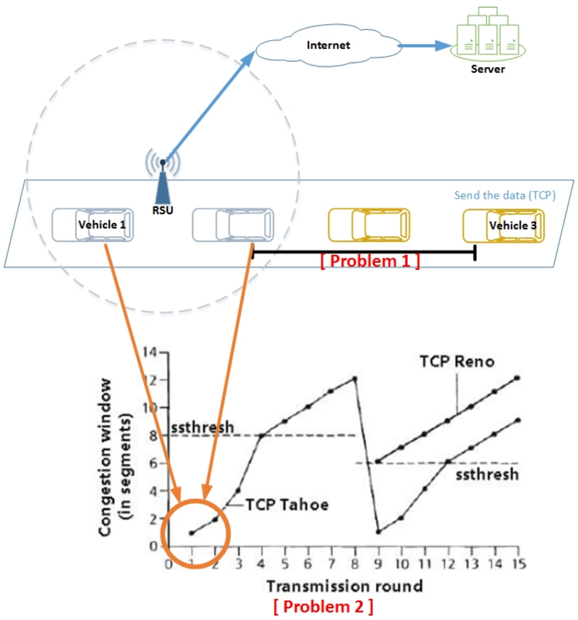


Fig. 1. Problem of TCP in vehicular network

In both aforementioned works [9][11], vehicular cooperation only occurs when vehicles have a common interest to download the same file from the infrastructure, which is different from our work that focuses on TCP pre-setup between the most heading vehicles and cloud server and relay the information to initiator vehicle.

TOMS, however, develop a TCP Context migration scheme for enhancement of data services in vehicular networks, and based on the shared the TCP connection information, TOMS lets vehicles have a quick connection after previous connection for the driving environment.

III. DESIGN OF TCP CONTEXT MIGRATION SCHEME

In this section, we will describe the design of TCP Context Migration Scheme for Enhancement of Data Services in Vehicular Networks. The goal of TOMS is to reduce TCP setup time and cooperative data transmit time between RSU and vehicle. Web sites usually use the TCP protocol for HTTP communication. The dangerous things happen on real roads should be uploaded to the central control center in real time. eq. cloud server. However, there are some problems with putting TCP on VANET at this time.

As shown in Fig. 1, there are two problems. The first problem is the sender must wait until the RSU coverage is reached for sending the packet. Second the problem is always congestion window starts with 1 segment. However, if the channel is clear, there is no need to start TCP on a 1 segment.

To solve this problem, we introduce the network architecture for TCP cooperative

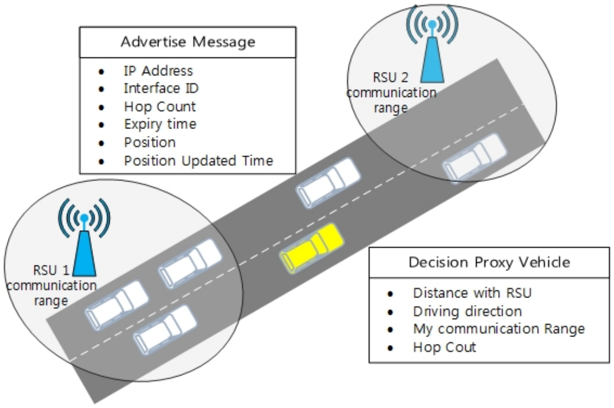


Fig. 2. Option for data transmission

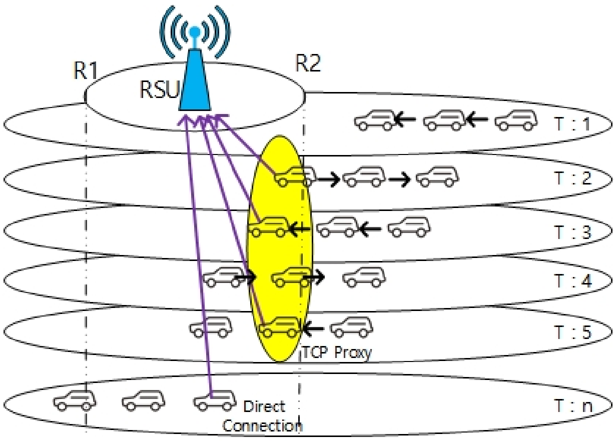


Fig. 3. How to determine the proxy mode vehicle

A. TCP Connection Situation

As shown in Fig. 2, If the yellow vehicle outside the RSU communication range wants to send TCP data. We want to send data to the cloud server through collaborative mode between vehicles. First, the vehicle closest to the RSU in the same direction will be connected to the Internet for the first time. As shown in Fig. 3, it is called the proxy header and is responsible for the TCP connection with the cloud server.

We discuss how to reduce the time of connections and procedures when sending TCP data to the cloud server. TOMS has the following rules:

- *Discovery*: Each vehicle must advertise their vehicles around. This information includes the location, hop count and IP address for reaching a specific host or RSU.
- *Decision Proxy*: In consideration of the vehicle direction, the distance to the RSU and the hop counter, the head vehicle in the same traveling direction is selected as the proxy gateway vehicle
- *TCP Setup*: The vehicle selected as a proxy establishes a TCP connection with the server. And sends the connected connection information to the actual source vehicle.

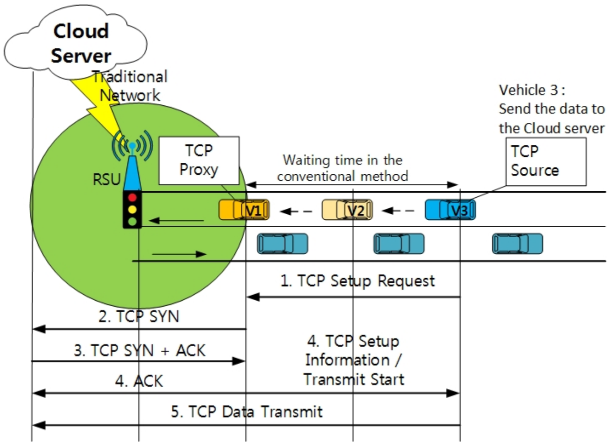


Fig. 4. TCP setup procedure

- *Packet Transmit*: Sends the data of a real vehicle that received the TCP connection information.

B. Cooperative Mode

In this section, describes how to connect to a real cloud servers at each location. Please refer Fig. 3

1) Outside of RSU communication range: Both the initiator vehicle and the proxy gateway vehicle are outside the RSU communication range. The leading vehicles reach RSU communication coverage. Therefore, the initiator vehicle sends the TCP configuration information to the proxy gateway vehicle. The lead vehicle sets itself as a proxy and keeps the packets in a queue.

2) Reaching RSU communication range: The lead vehicle has just arrived in the RSU communication coverage area. If the lead vehicle is set to proxy mode and data for TCP settings is in the queue, the packet is forwarded to the RSU.

3) Exiting RSU communication range: When the vehicle in the proxy mode is out of the RSU communication range, the TCP setting information must be sent to the next vehicle. The next proxy has an RSU and a hop count of 1, and the closest vehicle in the source vehicle is a candidate

4) Inside of RSU communication range: When Communicates directly with the RSU(Hop Count: 1) without the help of another vehicle, they can communicate directly without proxy vehicle.

C. TCP Setup Procedure

As shown in Fig. 4, describes the procedure for requesting access to a website from vehicle 3 to a cloud server.

1) TCP Setup Request: Vehicles at the head of the same direction are selected using the advertised vehicle's information in each vehicle location, speed and routing path information. We do not care about vehicles behind the driving direction. IPv6 addresses are used for inter-vehicle communication. Also, the traditional networks is connected with RSU and wired the network stack should not be changed. We encapsulates the original TCP/IP packet and wraps the IPv6 header for the

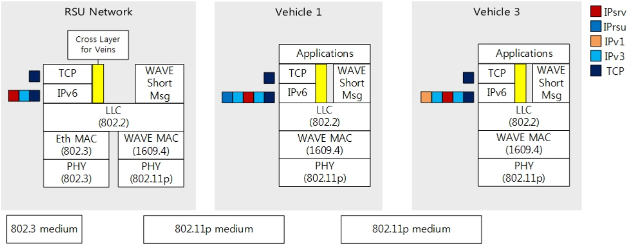


Fig. 5. Network layer architecture

VANET. The IPv6 extension header is used for IPv6 packet transmission between vehicles.

2) TCP SYN: The lead vehicle sends a TCP SYN to the cloud server via the RSU. The RSU manages the TCP SYN packet path from the TABLE's VANET network. Then, the IP Header used in the VANET network is removed and transmitted to the Cloud Server via ethernet.

3) TCP SYN+ACK: When the RSU receives a TCP SYN + ACK over the Ethernet network, it encapsulates the information and attaches a new IPv6 header using the TABLE information stored in the RSU. The leading vehicle (proxy) that receives this information immediately responds with an ACK to the server and simultaneously conveys the connection information to the source vehicle.

4) TCP Data Transmit: After this process, even if the actual source vehicle is outside the RSU communication range, a TCP connection can be established and data can be transmitted.

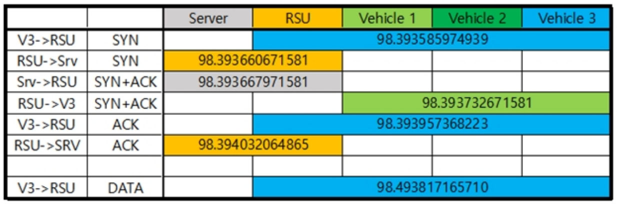
D. Network Architecture

As shown in Fig. 5, We setup LLC over WAVE (1609.4) MAC and installed IPv6 and TCP Stack. RSU has both Ethernet MAC and WAVE MAC due to communicating with the traditional internet outside VANET. As described in section III, for communication in VANET, the actual IPv6 Packet is wrapped once into IPv6 Header for VANET.

IV. RESEARCH ISSUES

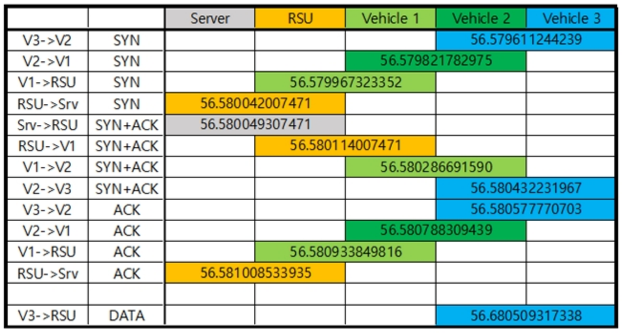
In this section, we introduce some research issues related to TCP Context migration scheme as follows:

- **Security for TOMS.** There is some security issue in the process of retransmitting connection information for TCP Pre-Setup.
- **Head (i.e., Proxy) vehicle determination algorithm.** Currently, the vehicle position is predicted through the DSDV message transmitted from each vehicle. However, generating too much DSDV packets leads to waste of bandwidth.
- **Network layer collapse due to cross layer design.** Overloading occurs due to a lot of processing through TCP, IP, Mac Layer to cooperate TCP data. etc.



		Server	RSU	Vehicle 1	Vehicle 2	Vehicle 3
V3->RSU	SYN		98.393585974939			
RSU->Srv	SYN	98.393660671581				
Srv->RSU	SYN+ACK	98.393667971581				
RSU->V3	SYN+ACK			98.393732671581		
V3->RSU	ACK		98.393957368223			
RSU->SRV	ACK	98.394032064865				
V3->RSU	DATA		98.493817165710			

Fig. 6. TCP Setup Time of traditional wireless network



		Server	RSU	Vehicle 1	Vehicle 2	Vehicle 3
V3->V2	SYN				56.579611244239	
V2->V1	SYN			56.579821782975		
V1->RSU	SYN		56.579967323352			
RSU->Srv	SYN	56.580042007471				
Srv->RSU	SYN+ACK	56.580049307471				
RSU->V1	SYN+ACK		56.580114007471			
V1->V2	SYN+ACK			56.580286691590		
V2->V3	SYN+ACK				56.580432231967	
V3->V2	ACK				56.580577770703	
V2->V1	ACK			56.580788309439		
V1->RSU	ACK		56.580933849816			
RSU->Srv	ACK	56.581008533935				
V3->RSU	DATA				56.680509317338	

Fig. 7. TCP Setup Time of using vehicular network

V. PERFORMANCE EVALUATION

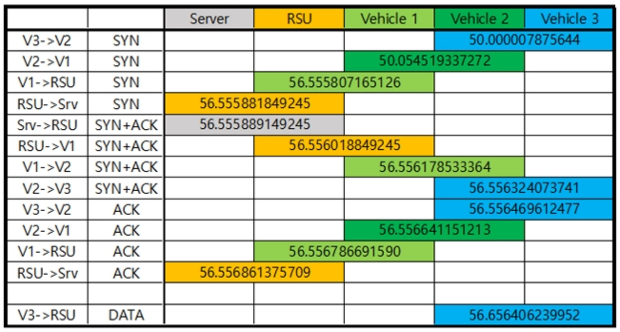
In this section, we evaluate the performance of TOMS in terms of the TCP Setup Time. We compare TOMS with traditional wireless network and vehicular network. In the traditional network approach, communication is possible when the initiator enters the RSU communication range. In the vehicular network approach, it starts packet routing when the lead vehicle enters the RSU communication range. In our proposal, it improves the connection speed by sending the TCP setup information to the lead vehicle. we measure time between vehicle of TCP client and Cloud of TCP server. In the third vehicle, it operates as a TCP client and requests connection to the Cloud Server. The total length of the road is 2 km and the maximum speed of the vehicle is 90 km/h. The vehicle 3 requests a TCP connection at 50 seconds for sending data. The data transmission power is 50.4 mW.

A. Simulation Design

In this section, we present how the simulation is implemented for TOMS. The simulation of TOMS is implemented in Veins which is an open source framework for Inter-Vehicular Communication (IVC) simulation in a data network simulator called OMNeT++, cooperating with a road network simulator called SUMO via Traffic Control Interface (TraCI). This allows for the bi-directionally coupled simulation of road traffic and network traffic. The movement of vehicles in SUMO is reflected in the movement of nodes in OMNeT++ via Veins.

B. The Impact of TCP Setup Time

In this section, we evaluate the performance of TOMS in terms of the TCP setup time. As shown in Fig. 6, It shows data



		Server	RSU	Vehicle 1	Vehicle 2	Vehicle 3
V3->V2	SYN				50.000007875644	
V2->V1	SYN			50.054519337272		
V1->RSU	SYN		56.555807165126			
RSU->Srv	SYN	56.555881849245				
Srv->RSU	SYN+ACK	56.555889149245				
RSU->V1	SYN+ACK		56.556018849245			
V1->V2	SYN+ACK			56.556178533364		
V2->V3	SYN+ACK				56.556324073741	
V3->V2	ACK				56.556469612477	
V2->V1	ACK			56.556641151213		
V1->RSU	ACK		56.556786691590			
RSU->Srv	ACK	56.556861375709				
V3->RSU	DATA				56.656406239952	

Fig. 8. TCP Setup Time of our propose

transmission in a traditional wireless network. The vehicular 3 waits until it enters the RSU communication range and transmits it. The actual TCP setup request occurred in 50 seconds, but the TCP setup is established at 98.494 seconds after moving the vehicle. As shown in Fig. 7, It shows data transmission using vehicular network. In this case, the leading vehicle is connected in 56.681 seconds as a routing function. As shown in Fig. 8, It shows our propose. the TCP setup time is delivered by the lead vehicle. Therefore, it establish only 56.651 seconds.

VI. CONCLUSION

In this paper, we proposed a framework of TCP Context Migration Scheme (TOMS) for efficient data services in vehicular networks. TOMS can reduce TCP setup time and speed up data transmission through cooperative data delivery among vehicles and RSUs. In order to reliably transmit Website data and driving information in real life, TCP Conext Migration can achieve much higher performance. As future work, we will implement the simulation of not only TCP setup procedure, but also the migration of TCP context (e.g., sequence number, acknowledgement number, receiver buffer size, and congestion window size), and also evaluate the performance of our TOMS in a realistic road network setting.

VII. ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2014006438). This research was supported in part by Global Research Laboratory Program (2013K1A1A2A02078326) through NRF, the ICT R&D program of MSIP/IITP (14-824-09-013, Resilient Cyber-Physical Systems Research), and the DGIST Research and Development Program (CPS Global Center) funded by the Ministry of Science, ICT & Future Planning. Note that Jaehoon (Paul) Jeong is the corresponding author.

REFERENCES

[1] V. Naumov and T. R. Gross, “Connectivity-aware routing (car) in vehicular ad-hoc networks,” in *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*. IEEE, 2007, pp. 1919–1927.

[2] Q. Xu, R. Segupta, D. Jiang, and D. Chrysler, “Design and analysis of highway safety communication protocol in 5.9 ghz dedicated short range communication spectrum,” in *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, vol. 4. IEEE, 2003, pp. 2451–2455.

[3] J. Zhao and G. Cao, “Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks,” *IEEE transactions on vehicular technology*, vol. 57, no. 3, pp. 1910–1922, 2008.

[4] J. Ott and D. Kutscher, “Drive-thru internet: Ieee 802.11 b for” automobile” users,” in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1. IEEE, 2004.

[5] J. Eriksson, H. Balakrishnan, and S. Madden, “Cabernet: vehicular content delivery using wifi,” in *Proceedings of the 14th ACM international conference on Mobile computing and networking*. ACM, 2008, pp. 199–210.

[6] N. Lu, N. Zhang, N. Cheng, X. Shen, J. W. Mark, and F. Bai, “Vehicles meet infrastructure: Toward capacity–cost tradeoffs for vehicular access networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1266–1277, 2013.

[7] L. Yang and F.-Y. Wang, “Driving into intelligent spaces with pervasive communications,” *IEEE Intelligent Systems*, vol. 22, no. 1, pp. 12–15, 2007.

[8] T. H. Luan, X. Ling, and X. S. Shen, “Provisioning qos controlled media access in vehicular to infrastructure communications,” *Ad Hoc Networks*, vol. 10, no. 2, pp. 231–242, 2012.

[9] H. Zhou, B. Liu, T. H. Luan, F. Hou, L. Gui, Y. Li, Q. Yu, and X. S. Shen, “Chaincluster: Engineering a cooperative content distribution framework for highway vehicular communications,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2644–2657, 2014.

[10] P. Alexander, D. Haley, and A. Grant, “Cooperative intelligent transport systems: 5.9-ghz field trials,” *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1213–1235, 2011.

[11] W. Zhu, D. Li, and W. Saad, “Multiple vehicles collaborative data download protocol via network coding,” *IEEE Transactions on vehicular technology*, vol. 64, no. 4, pp. 1607–1619, 2015.