The following publication Z. He, D. Zhang, S. Zhu, J. Cao and X. Liu, "SDN Enabled High Performance Multicast in Vehicular Networks," 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), Montreal, QC, Canada, 2016, pp. 1-5 is available at https://doi.org/10.1109/VTCFall.2016.7881215.

SDN Enabled High Performance Multicast in Vehicular Networks

Zongjian He, Daqiang Zhang, Shaomin Zhu
School of Software Engineering
Tongji University
Shanghai, China
E-mail: {hezongjian, dqzhang, kerryzhu}@tongji.edu.cn

Jiannong Cao, Xuefeng Liu
Department of Computing
The Hong Kong Polytechnic University
Hong Kong
E-mail: {csjcao, csxfliu}@comp.polyu.edu.hk

***Abstract*—Software defined network empowers the creation of a flexible network architecture by abstracting flow control from individual devices to the network level. In this paper, we address the challenges in applying SDN to develop high-performance vehicular networks. We present SDVN, a new SDN based vehicular network architecture. It organizes the topology of the vehicular networks and utilizes vehicle trajectory prediction to mitigate the overhead of the SDN control and data plane communication. Moreover, we propose a multicast protocol over SDVN, as multicast is the foundation of many vehicular network applications. The protocol exploits the network topology information provided by SDVN to make far more efficient multicast scheduling decision. The multicast scheduling problem is formulated to minimize the communication cost with bounded delay constraint. A polynomial time approximation algorithm is proposed. We conduct extensive experiments using traffic traces. The evaluation shows that the SDVN based multicast protocol outperforms existing decentralized approaches.**

***Index Terms*—SDN, VANET, multicast, vehicular network**

I. INTRODUCTION

With the advances of wireless communication and embedded technology, vehicle nowadays has become a powerful sensing, computing and communication facility. Vehicles have been connected to the Internet and ambient vehicles using wireless networks. However, challenges remain for the wide deployment of real world vehicular network applications. Foremost, network fragmentation has been caused by the difficulty in integrating heterogeneous wireless technology currently used in vehicle communications, including cellular, Wi-Fi, ZigBee, and DSRC. In addition, due to the mobility of vehicle, the decentralized vehicular ad-hoc networks (VANET) communication protocols are usually vulnerable, especially with a low market penetration of wireless devices [1].

The emerging software defined network (SDN) is a novel paradigm in networking that advocates logically centralized management of network units. SDN and OpenFlow provide new insights and alternative approaches to develop solutions of vehicular networks that can tackle the aforementioned challenges. The separation of data/control plane, and the employment of logically centralized management architecture can not only simplify the network operations but also provide better quality-of-service to vehicular network applications.

SDN has been widely adopted in mobile and wireless networks. OpenRoads [2] is the pioneer work by Stanford University. It enables multicast via heterogeneous networks to improve network throughput. SDWSN [3] presented an SDN based reconfigurable wireless sensor network to update the sensor nodes after they are deployed. Chung et al. [4] proposed an SDN based framework for wireless mesh networks to support mobility management, multi-hop communication and flow based routing in wireless environment. However, they can not be directly applied to vehicular networks scenario because of the heterogeneous nature.

It is non-trivial to directly apply SDN to vehicular networks with a large amount of highly dynamic mobile nodes. First, due to the frequent network topology change of vehicular networks, the control plane is not easy to maintain the global network topology. Since collecting the status of a individual vehicle (e.g., velocity, position, connectivity, and neighborhood) is both cost intensive and time consuming. Next, some existing decentralized protocols for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications may not be able to fully utilize the advantage of SDN. Since the control plane is now logically centralized and a lot of topology information becomes available. Therefore, network protocols may need to be redesigned accordingly to be adaptable.

Previously, we have present SDVN, or Software Defined Vehicular Networks, an SDN enabled new vehicular network architecture [5]. The key insight is to spend small network management overhead to collect the network topology information, which can be used to significantly simplify the network operations. SDVN also utilizes vehicle trajectory prediction to calculate the network topology changes in the near future. Therefore, the frequent vehicle status collection can be mitigated and the SDN management overhead can be notably reduced. Based on SDVN, we propose a multicast protocol named Pretti, or PREdictive Time-dependent mulTIcast. Since multicast is fundamental for many vehicular network applications, such as collision avoidance and real-time traffic aware transportation [6]. The key idea is to make centralized multicast scheduling decision using the global topology provided by SDVN over the heterogeneous vehicular networks.

We summarize the contributions of this research as follows:

1) We design a multicast protocol based on software defined vehicular networks, aiming at reducing the com-

munication delay.

2) We formulate the multicast scheduling problem as an optimization problem using time dependent graph. An approximation-based algorithm is designed.
3) We conduct extensive evaluations with real-world traffic trace using a typical vehicular network application. Results show the feasibility of SDVN and our multicast protocol outperforms existing decentralized approaches.

The rest of this paper is organized as follows: In Section II, the multicast scheduling problem is formulated. Section III proposes solutions for it. In section IV, the solutions are evaluated and the results are explained. Section V concludes this paper.

II. PROBLEM STATEMENT

A. System Model

We exploit SDN in vehicular networks. First of all, we need to make abstraction of the existing vehicular networks to adapt the SDN concepts.

Data Plane: We construct an overlay network, in which all vehicles, RSUs, and base stations are abstracted as SDN switches. We further categorize them with mobile or stationary data plane, and treat them differently when necessary.

Control Plane: The control plane is responsible for making packet forwarding decisions based on the status information of switches, such as connectivity, neighbor information, and etc.

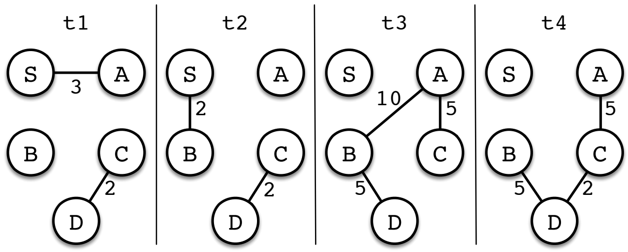
Communication Interface: The control plane and data plane can communicate with each other with a standard interface, which includes some predefined control and notification messages.

In SDVN, to make precise decision, the control plane need to know the status of individual switches at any time. However, the switches are a large amount of highly dynamic vehicles. The overhead of collecting the status in real-time is unfordable. Fortunately, the high frequency status update can be considerably avoided if the control plane can predict the future trajectory of vehicles, since the predicted trajectory can be used as an alternative to replace the real-time information. Researchers have developed many approaches [7] to predict vehicle trajectories, and these solutions can be directly applied to this work. Based on the trajectory prediction, control plane will calculate and update flow table for each switch. By doing this, the overhead of SDN management cost can be significantly reduced.

B. A Motivating Example

With trajectory prediction, the topology changes can be treated as an input of the multicast. But one question still remains: *Given topology prediction, how can we design delay and cost efficient multicast*?

To make the problem clear, we show a running example in Fig. 1. The SDVN is composed of 5 switches $\{S, A, B, C, D\}$, and the topology will change based on the predicted vehicle trajectory within time period $\mathcal{T} = \{t1, t2, t3, t4\}$. As the network is heterogeneous, different interfaces have different communication costs, which are labeled at the edges of the graph. Suppose source node S initiates a multicast to destination $\{C, D\}$. In real applications, the delay is usually more important than the cost, especially in safety related applications. Therefore, the communication delay is designed to be a hard constraint and the total communication cost defines the metric we want to optimize. For this example, the multicast is required to be delivered at the end of $t4$. Then, we need to schedule the packet forwarding in different time period to satisfy the requirement.



	t1	t2	t3	t4	Cost	Delay
S1	S->A	S->B	A->C	B->D	15	4
S2	NA	S->B	B->D	D->C	9	4
S3	S->A	S->B	A->C B->D	NA	15	3

Fig. 1. An example of multicast over SDVN: node S sends the multicast that need to be delivered to $R = \{C, D\}$, cost of each edge is as labeled.

Several possible scheduling policies are applicable. Three of them $\{S1, S2, S3\}$ are given in Fig. 1. We can figure out that $S2$ has the minimum cost among the three policies. If we require the multicast be delivered at the end of $t3$, only $S3$ can satisfy the requirement with a total cost of 15. This example shows that multicast over SDVN is still not trivial. As the network topology is still highly dynamic and the heterogeneity of networks still remains, especially with the consideration of the cost and delay trade-off.

C. Problem Formulation

Now we give rigorous definition of the system model and the multicast problem.

Definition 1: Software Defined Vehicular Network (SDVN): SDVN can be modeled as an overlay graph $G(V, E)$. A vertex $v \in V$ is an abstracted SDN switch. We use $T(v)$ to denote the network interfaces of the vertex, which can be DSRC, Wi-Fi, cellular network, or etc. A vertex may have multiple network interfaces simultaneously (e.g., a vehicle can have both LTE and DSRC installed onboard). An edge $e \in E$ is a wireless connection between two switches. It has two important properties, $D(e)$ and $C(e)$, where $D(e)$ is the communication delay and $C(e)$ is the cost. Without loss of generality, the cost is not necessarily communication cost. It can be either monetary or other limited resources (e.g. spectrum) in the real world. The predicted topology change of

SDVN can be modeled by a time dependent graph [8], which is a key data structure throughout this paper.

Definition 2: Time Dependent Graph (TD-G): A time dependent graph is a graph whose edges change with time. It 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 $||V|| = n$;
- $E \subseteq \{V \times V\}$ 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 *presence function* of a specific edge, where $\mathscr{P}(e) = 1$ represents the edge e is available.

The edges in TD-G can be either *physical* or *virtual*. A physical edge is a real wireless connection. A virtual edge is used to mask the heterogeneity of underlying physical networks, especially for transmission delay. Different physical networks have different one-hop delay, ranging from approximately 10ms for 802.11p to 500ms for cellular network [9]. However, TD-G is partitioned into equal length time periods. During the same time period, different physical networks may have different hop count. Therefore, a virtual edge is composed of multi-hop physical links in real world.

With above definitions, the multicast scheduling problem can be formulated as an optimization problem, as follows:

Given:

1) SDVN and the predicted vehicle trajectory represented by a time dependent graph $\mathcal{G} = (V, E, \mathcal{T}, \mathscr{P})$;
2) $s \in V$ is the multicast source node; $R \subset V$ is a set of multicast receiver nodes.

Objective:

Find a multicast scheduling policy $T_T(V_T, E_T) = (s_1, r_1)^{t1} \cup, \cdots, \cup (s_n, r_n)^{tn}$, such that:

$$\min\left(\sum_{e \in E_T} C(e)\right) \tag{1}$$

Subject To:

1) $(s_1 = s) \wedge (\mathscr{P}(s_k, r_k)^{t_k} = 1) \wedge (t_k \in \mathcal{T}) \wedge (t_1 \leq \cdots \leq t_n) \wedge (\forall_{k \geq 2}, s_k \in \{r_1, \cdots, r_{k-1}\})$;
2) $(V_T \subseteq V) \wedge (E_T \subseteq E) \wedge (R \subseteq V_T)$;
3) $t_n \leq \theta$.

Equation 1 shows that the total cost is the objective function we want to minimize. Constraint 1) reveals the requirements of the scheduling policy. The senders of each time period must have received the multicast before, and the connections we use must be available; Constraint 2) exhibits the solution must be derived from the given TD-G without adding additional vertices or edges, and the multicast must be delivered to all receivers; Constraint 3) is the delay constraint of the multicast. The problem is NP-Hard. We can not include the proof due to the page limit.

III. SOLUTIONS FOR MULTICAST SCHEDULING

A. Path and Shortest Path in TD-G

In this section, we present several essential concepts of TD-G, which the solutions are based on.

Definition 3: Time Dependent Path (TD-P): Node i and j in TD-G 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, TD-P is *directional*. We use $(i \to j)^{\mathcal{T}}$ to denote the time dependent path from i to j in time period $\mathcal{T}$. Given a TD-G, whether there is a time dependent path between two nodes can be determined in polynomial time. 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 $O(t \cdot n \cdot m)$.

Definition 4: Time Dependent Shortest Path (TD-SP): The time dependent shortest path from i to j in TD-G is the path in all time dependent paths from i to j with minimum total cost.

Given a pair of nodes s and d, finding TD-SP is more difficult than determining the existence of a TD-P. 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 1 shows how to find the time dependent shortest path. We use a list of 3-tuples (total cost, time period, 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 $O(t \cdot n)$.

Fig. 2 shows an example of applying the algorithm to a time dependent graph. The lower part of the graph is the list $vdict$ in Algorithm 1. After the execution of the algorithm, actually we can find the time dependent shortest path from A to all other nodes. Say, TD-SP(A, E) is $[(A, C)^{t2}, (C, D)^{t3}, (D, E)^{t4}]$ with total cost 4; TD-SP(A, G) is $[(A, D)^{t1}, (D, E)^{t2}, (E, F)^{t3}, (F, G)^{t4}]$ with total cost 16.

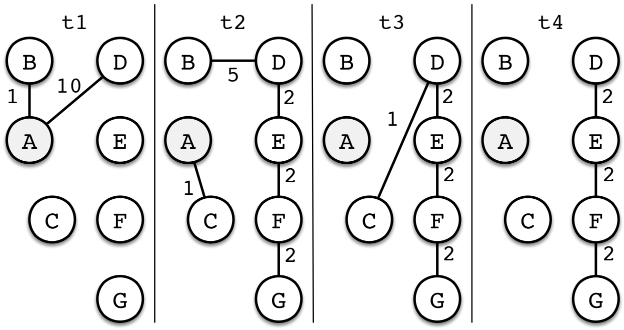
B. The TD-SPT based Approximation

The time dependent shortest path motivates a possible approach to find an approximate solution. The basic idea is to build a time dependent shortest path tree from the multicast source to all the receivers. The definition of time dependent shortest path tree is as follows:

Definition 5: Time Dependent Shortest Path Tree (TD-SPT): Given a tree root s and a set of tree leaves R, a time dependent shortest path tree is such a tree that it connects every node $v \in R$ with time dependent shortest path $(s \to v)^{\mathcal{T}}$.

Algorithm 1 Time dependent shortest path algorithm TD-SP$\{\mathcal{G}, s, d, t_s, t_e\}$, where s is source, d is destination, t_s is starting time and t_e is ending time.

```
vdict[$s$] ← $(0, -1, \text{None})$
for $t \leftarrow t_s, t_e$ do
    for all $v \in V$ do
        if $v \notin$ KEYS(vdict) then
            Continue
        end if
        for all $n \in$ 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)
```



A	0,-1,NA				0
B	1,t1,A				1
C		1,t2,A			1
D	10,t1,A	6,t2,B	2,t3,C		2
E		12,t2,D	8,t3,D	4,t4,D	4
F			14,t3,E	10,t4,E	10
G				16,t4,F	16

Fig. 2. Finding the time dependent shortest path.

We can also use Algorithm 1 to find the solution, since running the algorithm once is sufficient to get the shortest paths from the source to every node if it exists. Take Fig. 1 for example, suppose the time constraint $\theta = 4$, the TD-SPT solution simply merges the shortest path of TD-SP(S, C) and TD-SP(S, D). Hence, the result is $[(S, A)^{t1}, (S, B)^{t2}, (A, C)^{t3}, (B, D)^{t4}]$ with total cost 15. This is a valid solution for the problem. However, it is not the optimal one since the total cost of $S2$ is only 9.

IV. EVALUATION

A. Experiment Design

To make the evaluation convincing, we adopt a typical urban traffic trace dataset collected from the TAPAS-Cologne project. The traffic trace dataset is imported into SUMO traffic simulator to generate trajectories for all vehicles. The trajectories can be regarded as accurate prediction, which can hardly be achieved in the real world. Therefore, to make it practical, we arbitrarily add some prediction errors to the trajectory according to the results of [7]. Vehicle trajectories solely are insufficient to calculate the topology of SDVN. We still need to know the deployment of RSU. In this evaluation, we consider three deployment configurations, i.e., dense (1 km), sparse (5 km), and no. Then, the topology of SDVN can be determined and represented as a TD-G.

We implemented a safety warning application, which sends multicast to relevant vehicles when a traffic accident happens. It requires timely delivery of short messages. The requirements of the application are then converted to other parameters of the algorithms, such as multicast source, destinations and time constraint. Combining with the TD-G previously obtained, we can run the proposed algorithms, which is implemented in Python, and obtain the complexity and the accuracy results of the algorithm. We run the simulation with nodes mobility and multicast scheduling policy using NS-3 simulator. The network performance evaluation results are then obtain.

B. Accuracy of Approximation

Since TD-SPT algorithm is sub-optimal, another important metric is the accuracy of it compared with the optimal solution. This result is depicted in Fig. 3. We use exhaustive search to find the optimal solution, and k times of it as upper bound (TD-SPT is k-approximate). From the figure, we can find out that in practice, the solutions obtained by TD-SPT is much better than the upper bound. Therefore, the solution can be applied in most of the cases unless the cost is a critical metric.

C. Multicast Performance

The performance improvement of applying the proposed multicast in vehicular applications is the most important concern. In our experiment, we compare our SDN based solution with a classical decentralized multicast protocol AODV with multicast, namely MAODV [10]. The evaluation mainly focuses on three metrics: mean latency, successful delivery ratio and delivery cost. Mean latency shows the time consumed from initiating the multicast to all receivers successfully received the multicast. Successful delivery ratio is defined as the percentage of multicast that are successfully received by all receivers. The failure are caused by two reasons: packet loss and time out. Total cost is the cost spent on transmitting the multicast. We evaluated the two approaches with the application and three RSU configurations described previously.

Fig. 4 shows the performance in safety warning application. Approximately 100 vehicles is controlled by SDVN and the number of receivers is set to 5. Fig. 4(a) and (b) show that both the latency and delivery ratio have improved by utilizing

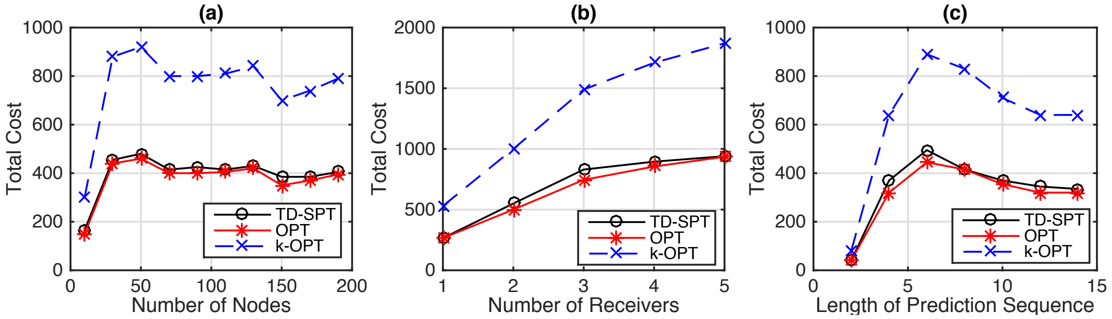


Fig. 3. Accuracy of the approximation algorithm

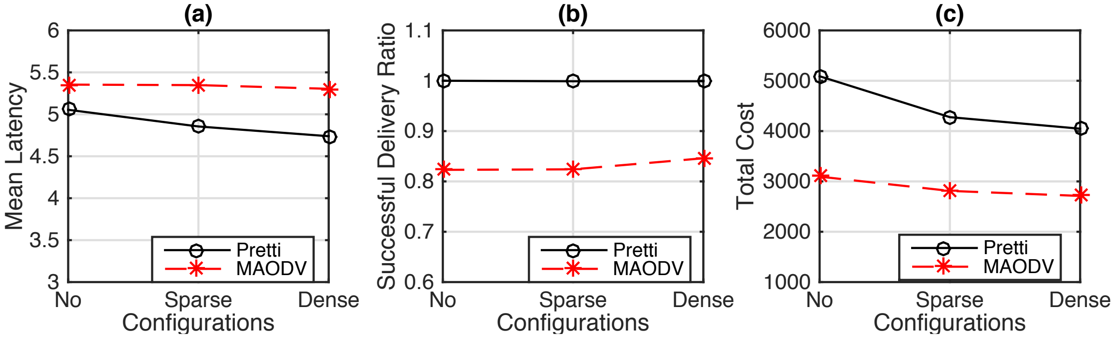


Fig. 4. Comparison of MAODV and Pretti with safety warning

our SDVN based multicast. However, Fig. 4(c) shows Pretti spends more cost than MAODV. This is caused by the switch status update using cellular networks. Since in this application, the amount of data transmitted by multicast is not large and the transmission distance is not far, the SDN management overhead can not be neglected.

V. CONCLUSION

We addressed the problem of building a delay and cost efficient multicast for heterogeneous vehicular networks. We proposed a novel multicast protocol based on software defined vehicular networks to minimize communication cost while guarantee data delivery threshold. Evaluation results showed our solution outperforms decentralized solutions.

ACKNOWLEDGMENT

This work is partially supported by 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). This project is also sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry.

REFERENCES

[1] F. Bai and B. Krishnamachari, “Exploiting the wisdom of the crowd: localized, distributed information-centric VANETs,” *IEEE Communications Magazine*, vol. 48, no. 5, pp. 138–146, May 2010.

[2] K.-K. Yap, M. Kobayashi, R. Sherwood, T.-Y. Huang, M. Chan, N. Handigol, and N. McKeown, “OpenRoads: empowering research in mobile networks,” *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 125–126, 2010.

[3] T. Miyazaki, S. Yamaguchi, K. Kobayashi, J. Kitamichi, S. Guo, T. Tsukahara, and T. Hayashi, “A software defined wireless sensor network,” in *International Conference on Computing, Networking and Communications (ICNC)*, 2014, pp. 847–852.

[4] P. Dely, A. Kassler, and N. Bayer, “Openflow for wireless mesh networks,” in *International Conference on Computer Communications and Networks (ICCCN)*, 2011, pp. 1–6.

[5] Z. He, J. Cao, and X. Liu, “SDVN: Enabling rapid network innovation for heterogeneous vehicular communication,” *IEEE Network Magazine Special Issue on Software Defined Wireless Networks*, 2015.

[6] Z. He, J. Cao, and T. Li, “MICE: A real-time traffic estimation based vehicular path planning solution using vanets,” in *International Conference on Connected Vehicles and Expo (ICCVE)*, 2012, pp. 172–178.

[7] 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.

[8] A. Orda and R. Rom, “Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length,” *Journal of the ACM*, vol. 37, no. 3, pp. 607–625, 1990.

[9] A. Corti, V. Manzoni, S. Savaresi, M. Santucci, and O. Tanna, “A centralized real-time driver assistance system for road safety based on smartphone,” in *Advanced Microsystems for Automotive Applications*, 2012, pp. 221–230.

[10] E. M. Royer and C. E. Perkins, “Multicast operation of the ad-hoc on-demand distance vector routing protocol,” in *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, 1999, pp. 207–218.