

Article

A Distance-Vector-Based Multi-Path Routing Scheme for Static-Node-Assisted Vehicular Networks

Daichi Araki ^{1,†} and Takuya Yoshihiro ^{2,*}

¹ Graduate School of Systems Engineering, Wakayama University, 930 Sakaedani, Wakayama 640-8510, Japan; araki.daichi@g.wakayama-u.jp

² Faculty of Systems Engineering, Wakayama University, 930 Sakaedani, Wakayama 640-8510, Japan

* Correspondence: tac@sys.wakayama-u.ac.jp

† Current address: He is currently with Mitsubishi Electric Corporation.

Received: 14 April 2019; Accepted: 12 June 2019; Published: 14 June 2019



Abstract: Vehicular Ad hoc NETWORKS (VANET) has been well studied for a long time as a means to exchange information among moving vehicles. As vehicular networks do not always have connected paths, vehicular networks can be regarded as a kind of delay-tolerant networks (DTNs) when the density of vehicles is not high enough. In this case, packet delivery ratio degrades significantly so that reliability of networks as an information infrastructure is hardly held. Past studies such as SADV (Static-node Assisted Data dissemination protocol for Vehicular networks) and RDV (Reliable Distance-Vector routing) showed that the assistance of low-cost unwired static nodes located at intersections, which work as routers to provide distance-vector or link-state routing functions, significantly improves the communication performance. However, they still have problems: SADV does not provide high-enough delivery ratio and RDV suffers from traffic concentration on the shortest paths. In this paper, we propose MP-RDV (Multi-Path RDV) by extending RDV with multiple paths utilization to improve performance against both of those problems. In addition, we apply a delay routing metric, which is one of the major metrics in this field, to RDV to compare performance with the traffic-volume metric, which is a built-in metric of RDV. Evaluation results show that MP-RDV achieves high load-balancing performance, larger network capacity, lower delivery delay, and higher fault tolerance against topology changes compared to RDV. As for routing metrics, we showed that the traffic-volume metric is better than the delay one in RDV because delay measurement is less stable against traffic fluctuation.

Keywords: vehicular networks; delay tolerant networks; distance-vector routing

1. Introduction

Vehicles are regarded as a kind of mobile sensor nodes and vehicular ad hoc networks are regarded as a potential infrastructure to help collecting or delivering various data that may enhance our life. Vehicular Ad hoc NETWORKS (VANETs) have been deeply studied to realize the dependable data infrastructure that consists of vehicles by means of multi-hop packet forwarding. Although VANETs are expected to support both metropolitan and rural scenarios, especially rural (or suburban) scenarios in which vehicles are not dense enough have a significant problem on packet delivery ratio. If vehicle density is low, VANET is regarded as a kind of Delay-Tolerant Networks (DTNs), and the past contributions on DTN routing have proved that high packet delivery ratio is hardly achievable with sparsely located nodes; with simple schemes such as the epidemic routing or the spray-and-wait approach, a significant load for duplicated packets is required to improve delivery ratio of packets [1–3].

A large number of studies have been dedicated to overcoming this sparsity problem in VANET [4–11]. From the early stage of VANET, geographic routing is regarded as one of the best-suited routing strategies, and several techniques have been developed based on it. In DTN-based vehicular networks, vehicle selection to pass packets is a key decision to improve packet delivery ratio. Thus, *carry-and-forward* behavior [8] that affords vehicles to have time to choose the next-hop vehicle has been deployed with techniques for selecting good next-hop vehicles such as utilizing vehicles' trajectories (e.g., [9]) or estimating vehicle traffic amount of roads (e.g., [10]). However, although those techniques certainly improve the delivery ratio of packets in sparse scenarios of VANETs, the ratio is still too low in order to support practical applications over VANETs.

The idea to utilize static, but not wire-connected wireless nodes located at intersections is introduced in SADV [12]. In SADV, static nodes are located at each intersection and they work as a link-state router; they retrieve the network topology from the roadmap, measure the packet forwarding delay of each link between two neighboring static nodes, and compute the shortest-path to each destination. At intersections, static nodes receive packets from vehicles and pass them to other vehicles that are likely to carry packets along the shortest paths to the destinations. SADV shows that inexpensive unwired static nodes located at each intersection can significantly improve the packet reachability to their destinations. In addition, by forwarding duplicated copies of a packet to the primary and the secondary next-hops at every relay node, i.e., by using multiple paths, SADV further improved the packet delivery ratio. However, since SADV has no specific mechanism to control the expected ratio on end-to-end packet delivery, there are cases in which packet delivery ratio is not sufficiently high, which is inconvenient in practice as a reliable infrastructure. In addition, since every node always forwards packets to the two next-hops, the overhead of communications is high so that the network capacity is limited.

Yoshihiro et al. proposed a distance-vector-based routing scheme RDV that provides the pre-configured expectation on packet delivery ratio by creating the sufficient number of packet copies at static nodes [13]. Namely, if administrators configure to expect 99% end-to-end packet delivery, RDV provides the 99% expected packet delivery ratio. Since RDV creates the minimum number of packet copies to achieve the preconfigured expectation, RDV requires the lowest overhead of duplicated packets. However, since RDV uses a single shortest path to forward packets to a destination, data traffic easily concentrates on the shortest paths and exceeds the capacity. In addition, since the paths are selected based on vehicle density rather than delivery delay, the forwarding paths provide far larger delay compared to SADV.

Now note that, although RDV has a strong characteristic providing probabilistic expectation on packet delivery ratio, the performance of RDV has not still been optimized. SADV has two major advantages on RDV that may further improve performance of RDV, that is, multi-path utilization and delay routing metric. Thus, in this paper, as an extension of a conference paper [14], we try to introduce those two features into RDV to improve the communication performance. First, we incorporate the multi-path utilization into RDV to relax the concentration of packets on the shortest paths so as to expand the traffic capacity of RDV. Second, we compare two major routing metric criteria, i.e., traffic volume and delivery delay, in RDV routing schemes in order to make better selection of the metrics.

Specifically, in this paper, we propose MP-RDV (Multi-Path RDV) that extends RDV to utilize multiple paths. MP-RDV forks forwarding paths at each intermediate static node and creates duplicated packet copies for each of the paths to improve the load balancing performance as well as the delivery delay. Inheriting the property of RDV, MP-RDV minimizes the total traffic loads to achieve the preconfigured delivery ratio by computing the minimum number of packet copies to forward on each of the forked paths. In addition, we designed a delay routing metric and its computational mechanism that work with RDV to provide an option of delay-based metric utilization in RDV. Since the delivery delay of packets consists of waiting delay on nodes and transportation delay between nodes, those two values are measured separately in cooperation with related nodes and summed up to compute the delay metric values, and the metric values are advertised by the distance-vector messages.

The simulation results show that MP-RDV provides well load-balanced communications so that larger traffic than both RDV and SADV can be forwarded while simultaneously holding the preconfigured expected delivery ratio. In addition, by utilizing multiple paths simultaneously, delay performance is reduced as the same level as SADV. As for comparison of metrics, we found that the built-in traffic-volume metric has totally better performance because the delay metric is found less stable and reliable against fluctuation of vehicular traffic. The results lead to the conclusion that MP-RDV with traffic-volume metric has totally better communication performance than the conventional methods in the literature.

The remainder of this paper is organized as follows. In Section 2, we describe the related work in the literature and clarify the contribution of this study. In Section 3, we introduce the distance-vector routing scheme RDV, which is the base single-path scheme that we extend in this paper. In Section 4, we present the proposed scheme MP-RDV in detail, and introduce the delay routing metric to MP-RDV. Evaluation is done through simulation in Section 5, which examines the performance of MP-RDV in terms of several measurement criteria. Finally, in Section 6, we conclude the work.

2. Related Work

2.1. Vehicular Ad Hoc Networks

Since vehicular networks could be a data infrastructure in both urban and rural areas, multi-hop vehicular ad hoc networks have been deeply studied so far. Once geographic routing has been found to be suitable in vehicular networks, several effective techniques have been developed based on geographic routing. Lochert et al. proposed a DTN routing scheme called GPCR (Greedy Perimeter Coordinator Routing) and showed that the geometric routing is one of the suitable choices in routing strategies in VANET [4]. Lee et al. proposed GPSRJ+ (Greedy Perimeter Stateless Routing Junction+) that improves the road segment selection method based on geographic routing to achieve higher packet delivery ratio [5]. Lochert et al. presented GSR (Geographical Source Routing) that introduces an anchor-based routing, in which Dynamic Source Routing (DSR) based on anchors on map is applied into vehicular networks [6]. Seet et al. proposed an anchor-based routing called A-STAR (Anchor based Street and Traffic Aware Routing) that considers streets on maps and vehicular traffic amount on streets in making routing decisions [7]. Zhao et al. proposed VADD (Vehicle-Assisted Date Delivery) in which they introduced the *carry-and-forward* behavior into geometric routing that holds packets until they find adequate next-hop vehicles that are likely to forward packets to the destinations [8]. Wu et al. proposed MDDV (Mobility-centric Data Dissemination Algorithm for Vehicular Networks) that forwards packets along the trajectory precomputed from static road information [9]. Jerbi et al. proposed GyTAR (Greedy Traffic Aware Routing), which considers dynamic traffic information such as density of vehicles to select delay-optimal outgoing roads at each intersection [10]. Zhang et al. proposed a street-centric routing that introduces the concept of micro topologies to consider the wireless connections among vehicles on each street [11]. However, although those techniques certainly improve the delivery ratio of packets in sparse scenarios of VANETs, the delivery ratio is still too low in order to support practical applications over VANETs.

2.2. Deployment of Static Nodes in Vehicular Networks

Static nodes have been considered as useful devices to support communications of vehicles. A typical style of static node is called RSU (Road-Side Unit), which often provides single-hop communication with vehicles. The well-known example of such systems is DSRC (Dedicated Short Range Communications), which has already been standardized as IEEE1609 families [15], and also have practically deployed all over the world such as ERP (Electric Road Pricing) in Singapore [16] and VICS (Vehicle Information and Communication Systems) in Japan [17].

RSU also works in combination with multi-hop vehicular networks. To enhance the coverage of RSU, Jeong et al. proposed TSF (Trajectory-based Statistical Forwarding) [18] that delivers packets

from RSUs to a target vehicle via multi-hop communications of vehicles. Reversely, Skordylis et al. and Jarupan et al. respectively proposed D-Greedy (Delay-bounded Greedy forwarding) [19] and PROMPT (cross-layer Positionbased coMMunication PROTcol for delay-aware vehicular access networks) [20], which forward packets from vehicles to RSUs via multi-hop communications. Wu et al. regard the problem of packet delivery among vehicles supported by wired RSUs as an optimization problem and solved the problem to provide better performance on packet delivery [21]. O'Driscoll et al. proposed a geographic routing protocol called IEGRP (Infrastructure Enhanced Geographic Routing Protocol) that is enhanced by wired RSUs located at intersections to exchange packets among vehicles [22]. However, they all assume that RSUs have wired connection to networks, which requires high building costs.

On the other hand, Li et al. proposed an approximated algorithm to compute the optimal locations of wired/wireless static nodes in the roadmap that supports delay-bounded propagation of information in VANETs [23]. This study tries to reduce the building cost of RSUs while maximizing the effect on enhancing the performance of vehicular ad hoc networks.

2.3. Enhancing Vehicular Ad Hoc Networks with Unwired Static Nodes

Several studies have been made to enhance vehicular ad hoc networks with unwired static nodes. As is already described, SADV [12] is based on link-state routing in which static nodes compute the shortest paths as a sequence of static nodes and build their own routing tables. Each node creates two duplicated copies of a packet and forwards them to the next-hops corresponding to the primary and secondary shortest paths. SADV has no other packet duplication strategy so that it is vulnerable to unexpected packet loss due to vehicles' movement such as going into narrow streets or parking lots. Note that this effect grows large when static nodes are sparsely located at intersections on the roadmap. Thus, in SADV, static nodes are expected to be installed at a high percentage of intersections.

RDV [13] improved this point by duplicating the required number of packets to suffice the preconfigured expected packet delivery ratio. This not only improves reliability of packet forwarding, but also enables us to make sparse deployment of static nodes located at intersections, i.e., even if we place static nodes at only major intersections between which there is vehicle reach with relatively low probability, packets can be forwarded with a sufficiently high delivery ratio. Sparse deployment reduces deployment cost and makes the routing scheme practical.

ETGR (Efficient Traffic Geographic Routing) [24] is a reactive routing protocol in which the path discovery process using RREQ (Route REQuest) and RREP (Route REPLY) packets is assisted by static nodes. ETGR in the original paper treats real-time communications in dense-vehicle scenarios, and compared the communication performance with geometric routing schemes such as GSR [6] and GyTAR [10]. In the sparse scenario, however, due to the dynamic nature of VANET, this approach suffers from large paths repairing cost as well as a large delay to search for available paths to the destinations.

AQRV (Adaptive Quality-of-service (QoS)-based Routing for VANETs) [25] proposed QoS-based routing in static-node-assisted vehicular networks. AQRV assumes urban scenarios in which static nodes are installed at a high percentage of intersections, and computes the communication quality for each road segment between neighboring static nodes. Then, the static nodes behave as routers that dynamically select routes based on an ACO (Ant-Colony Optimization) algorithm. Although AQRV provides high-quality communications with the help of static nodes in urban scenarios with high vehicle density, it does not provide communication quality in sparse scenarios.

EGSR (Efficient GSR) [26] does not explicitly assume stationary devices at intersections, and instead assumes that vehicle density is high enough to hold required information in cooperation with vehicles around intersections. EGSR measures the quality of links (here, a link means the road segment between two adjacent intersections) based on the ACO approach, and keeps sharing the information around intersections. Every vehicle computes the shortest paths using the Dijkstra's shortest-paths algorithm with the measured link quality as link weight. The weakness of this method, however, appears when it faces sparse cases in which information sharing among vehicles is impossible.

ROSTER (Routing based on SpatioTemporal Encounter Records) [27] utilizes daily trajectory data of vehicles to find good paths with low delay performance in sparse scenarios. By computing the intersection region of trajectories and placing RSUs there, they achieve low delay and high delivery ratio communications in sparse scenarios. Although ROSTER covers majority of communication requirements, they cannot provide full coverage of communication requests in the region because it covers only the major requirements that appear in the trajectory data. In addition, their method highly depends on traffic patterns so that performance could be degraded when traffic pattern changes.

3. RDV: The Base Scheme

3.1. Overview

RDV assumes that static unwired nodes are sparsely placed at major intersections to help forwarding packets among static nodes. In RDV, static nodes behave as routers and vehicles are the carriers of packets. RDV is a routing scheme that maintains routing tables at each static node in the manner of distance-vector routing scheme. When a vehicle comes into the communication range of a static node, the node exchanges control messages and data packets with the vehicle. Packets are delivered from source static nodes to their destination static nodes via several relay static nodes according to the routing table entries. Since static nodes are located to cover a certain area, RDV supports the packet delivery from and to any locations in the area. Note that, although we describe packet delivery between two static nodes, the sources and the destinations of packets in RDV could not be limited to static nodes but moving vehicles. To support this, we can combine several methods with RDV that provide packet delivery between static nodes (RSUs) and vehicles such as D-Greedy [19], PROMPT [20], or TSF [18].

Vehicles as carriers deliver packets between two static nodes with the *store-and-forward* behavior. When vehicles are in the communication range of static nodes, they switch to *intersection mode*. They pass packets to the node to relay them to other vehicles, and receive packets from the node that should be forwarded toward the vehicle's on-going direction. Otherwise, when vehicles are not in the communication range of any static node, they work in *road mode*, in which they pass packets to the vehicles ahead of them whenever being possible. In combination of those two modes, RDV provides efficient packet delivery in both urban and rural scenarios.

One of the novel characteristics of RDV is providing reliable packet delivery in that packets are delivered with the pre-configured expected delivery ratio such as 99%. This is done by the function APD (Adaptive Packet Duplication), which creates several duplicated copies of a packet at every relay node to improve the expected delivery ratio to the value pre-configured by the administrators. Note that APD controls the number of duplicated packets to the minimum as long as the expected delivery ratio exceeds the preconfigured value. This significantly reduces the overhead of packet copies and enhances the capacity of networks.

We define the units in distances specific to RDV as follows. See Figure 1. We define the unit *hop* as the number of static nodes that a path involves. Consequently, since a vehicle carries packets from node A to B without visiting any other static node, the distance from A to B is 1-*hop*. Since a vehicle visits two nodes between A and D, the path is 3-*hop* long. On the other side, we define the unit *carry* as the number of vehicles required to forward packets. Since a packet can be forwarded from node A to D with a single vehicle, the distance from A to D is 1-*carry* even if it takes several-*hops*. Since a packet is not possible to be forwarded from A to E or F with a single vehicle, the packet must be relayed by static nodes. Since a packet is reachable from A to F via two relay nodes, it must be forwarded by at least three vehicles, so the distance from A to F is 3-*carries*.

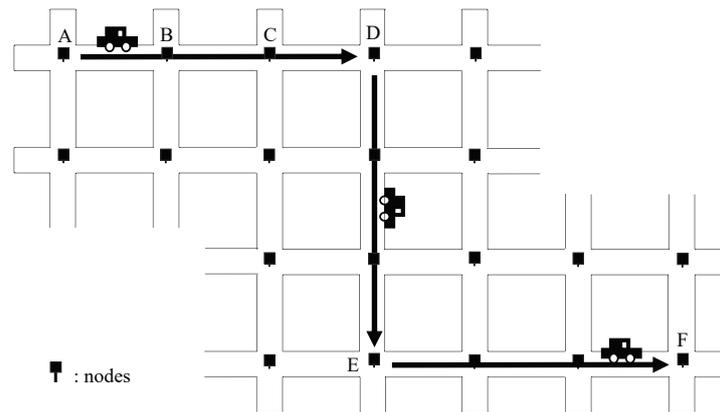


Figure 1. Definitions of Distance Units: The unit *hop* represents the number of static nodes on the path. The unit *carry* represents the number of vehicles required to forward packets on the path.

3.2. Distance-Vector Based Routing Strategy

RDV is designed for vehicular networks to work based on a traditional distance-vector based routing strategy. RDV autonomously detects neighborhoods via control messages to build the shortest-paths so that it does not require map information. Since RDV is independent from maps, RDV is free from maintenance and synchronization cost of map information. Based on the distance-vector routing strategy, nodes in RDV exchange messages periodically to build their routing tables. In the routing table of each static node, a static node in 1-*carry* distance is specified as the *next-relay* for each destination. Since packets are delivered according to routing tables, a packet forwarding path is defined as a chain of static nodes within 1-*carry* distance.

When a node A is going to forward a packet to a 1-*carry* neighbor B, A finds a vehicle that will visit B with high probability, and pass the packet to it. To find a suitable vehicle to pass packets, RDV has a mechanism called SVS (Statistic-based Vehicle Selection), which predicts vehicles that will visit the specified neighbor from the statistical measurements. In SVS, *Hello* messages are sent from each static node to its 1-*hop* and 2-*hop* neighbors by storing a *Hello* message on each vehicle that visits the node. Each node obtains local trajectory information via *Hello* messages. By sharing the information among neighborhoods using *Statistic* messages, every node is ready to compute the best “previous-hop” neighbor node, i.e., the vehicles coming from which will with high probability go to the specified *next-relay* node.

To build a routing table, each node must know the neighbors within 1-*carry* distance. To find 1-*carry* neighbors, RDV uses *Single-carry* messages. Each vehicle distributes the statistical information of SVS with *Single-carry* messages along the roads that the vehicle goes through. As a result, static nodes discover all of their 1-*carry* neighbors, to which packets are delivered with a single vehicle with high probability. The probability of packets to reach a 1-*carry* neighbor is computed as the product of 1-*hop* delivery ratios collected along with the path to the neighbor.

Each node propagates its 1-*carry* neighbors with *Multi-carry* messages in the manner of distance-vector routing. Specifically, a *Multi-carry* message propagates a set of distance-vectors (i.e., basically a set of destinations and the distances for them) to sender’s 1-*carry* nodes. As a result of repeated message propagation, all static nodes finally find the shortest paths for all destinations, and their routing tables have entries for every reachable static node.

With the routing tables computed above, a packet is forwarded to its destination via several intermediate nodes. The next intermediate node for a destination is called *next-relay*, which is retrieved from the routing table at each node. Note that the routing table includes the best *previous-hop* node, the vehicle from which will visit the next-relay with high probability. Thus, when a static node is to forward a packet, it consults with its routing table and find the *previous-hop* node, and passes the

packet to the vehicle that comes from the *previous-hop* node. By repeating this forwarding process at each intermediate node, packets finally reach their destinations.

3.3. Packet Duplication

In VANET, vehicles as carriers of packets move under their drivers' own will so that they might go away or park somewhere without forwarding packets to their next-relay nodes. To achieve reliable packet delivery under this kind of uncertainty, RDV creates several copies for each packet and forwards all of them to its next-relay nodes. Note that RDV creates the copies of a packet dedicated to reaching the next-relay node, i.e., the packet copies destined to its next-relay, and if they fail to reach the next-relay, they will be dropped silently. By forwarding multiple copies to a next-relay, we can raise the packet delivery ratio to a pre-configured level even under a relatively high drop probability for each packet.

The mechanism of packet duplication in RDV to achieve the pre-configured expected packet delivery ratio is called APD. In APD, network administrators previously configure the end-to-end packet delivery ratio that they expect, and RDV generates packet copies at each intermediate node so that the expectation of packet delivery ratio exceeds the preconfigured value such as 99%.

Now let P be the preconfigured expected delivery ratio to suffice, and let δ be the distance in carry of the path to the destination. (See Table 1 for the definitions.) For each link l in the path, we denote the expected packet delivery ratio by E_l . Since the delivery ratio of the path is expressed by the product of those of the included links, P is achieved if each included link l suffices $(E_l)^\delta \geq P$, i.e., the least expected value of E_l is expressed as

$$E_l = P^{\frac{1}{\delta}}. \quad (1)$$

On the other hand, let R_l be the delivery ratio of link l measured by SVS, and D_l the number of duplicated packet copies on link l , respectively. Then, to achieve the delivery ratio E_l , two variables R_l and D_l must suffice the following formula:

$$1 - (1 - R_l)^{D_l} \geq E_l. \quad (2)$$

From this, the smallest value of D_l to achieve E_l is computed as

$$D_l = \text{ceil}\left(\frac{\log(1 - E_l)}{\log(1 - R_l)}\right), \quad (3)$$

where $\text{ceil}(\cdot)$ is the ceiling function. RDV achieves the preconfigured expectation of end-to-end delivery ratio if each of source and intermediate nodes generates D_l copies according to the above Formula (3).

Table 1. Notations on Adaptive Packet Duplication (APD).

Variable	Value
P	Preconfigured expected delivery ratio.
δ	Distance in carry for the destination.
E_l	Expected delivery ratio on link l .
D_l	Number of duplicated copies on link l .
R_l	delivery ratio of link l estimated by SVS.

Figure 2 shows a concise example of APD. The expected delivery ratio that is previously configured by administrators is set as $P = 99\%$, and a packet is to be forwarded from n_1 to n_4 . In this case, n_1 computes the expected delivery ratio E_l for each link $l = (n_1, n_2)$, (n_2, n_3) , and (n_3, n_4) as 99.67% from the above equation. Since the measured delivery ratio of link (n_1, n_2) derived from SVS is $R_{(n_1, n_2)} = 80\%$, the number of duplicated copies of the packet is computed as $D_{(n_1, n_2)} = 4$ from Formula (3). As a result, n_1 forwards four packets to n_2 . Note that E_l is included in the packet and is notified to n_2 . With this information, n_2 computes the number of duplicated copies to be forwarded to n_3 in the same computational process as the case of n_1 .

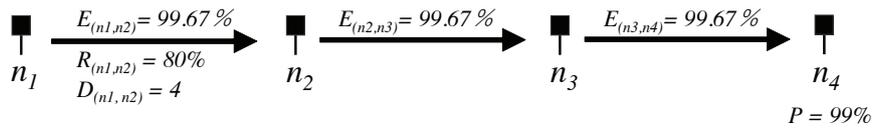


Figure 2. Example of Adaptive Packet Duplication (APD): If the required packet delivery ratio is $P = 99\%$, distance is 3-carries, and the packet delivery ratio from n_1 to n_2 is 80%, n_1 creates four duplicated copies of a packet to suffice the delivery ratio P .

3.4. The Routing Metric

Since RDV is designed to improve delivery ratio, it adopts routing metric by which the paths with large bandwidth, i.e., large vehicle traffic volume, is preferably selected as the packet forwarding paths. To compute the routing metric for each link (n_s, n_r) , RDV computes the expected number of vehicles V that go from n_s to n_r in a unit time using the mechanism of SVS. Then, as customary in the Internet routing to incorporate link bandwidth into the shortest-path computation, RDV defines its routing metric as the inverse of the traffic amount. Namely, the metric $M_l^{(vol)}$ of the link l is defined as

$$M_l^{(vol)} = \frac{1}{V}. \tag{4}$$

Note that this value easily fluctuates with time, which leads instability of the forwarding paths. To prevent the fluctuation, it is strongly recommended to apply a smoothing technique such as taking the moving average over time.

3.5. An Example

An example scenario of packet forwarding in RDV is depicted in Figure 3 with the routing table of n_2 and n_5 shown in Tables 2 and 3, respectively. Note that each entry of RDV’s routing tables consists of a destination, a next-relay node, a previous-hop node, the distance in carry to the destination, the delivery ratio measured by SVS, and the routing metric to the destination. Suppose that a packet is to be forwarded from n_2 to n_7 . First, n_2 finds the entry for destination n_7 and found the corresponding next-relay node is n_5 . Since the previous-hop is n_1 , n_2 passes the copies of the packet if vehicles from n_1 visit n_2 . If APD determines to forward multiple copies of the packet, they are passed to different vehicles coming from n_1 . Note that the entry means that, if the packet is passed to the vehicles coming from n_1 , the probability of the packet to reach the next-relay n_5 is 72%. When n_5 receives one of the packet copies, n_5 actually receives the packet. Thus, even if other copies of the same packet are received thereafter, n_5 silently discards them. Next, n_5 forwards packets to n_7 in the same procedure, i.e., n_5 finds the routing table entry for destination n_7 , passes the packet copies to vehicles coming from n_6 , and finally the destination n_7 receives the packet. RDV forwards packets in this way to achieve high delivery ratio.

Table 2. Routing Table of n_2 in Figure 3.

Dest.	Next,-Relay	Prev.	Dist.	Del. Ratio	Metric
⋮	⋮	⋮	⋮	⋮	⋮
n_7	n_5	n_1	2	72%	0.064
⋮	⋮	⋮	⋮	⋮	⋮

Table 3. Routing Table of n_5 in Figure 3.

Dest.	Next,-Relay	Prev.	Dist.	Del. Ratio	Metric
⋮	⋮	⋮	⋮	⋮	⋮
n_7	n_7	n_6	1	90%	0.028
⋮	⋮	⋮	⋮	⋮	⋮

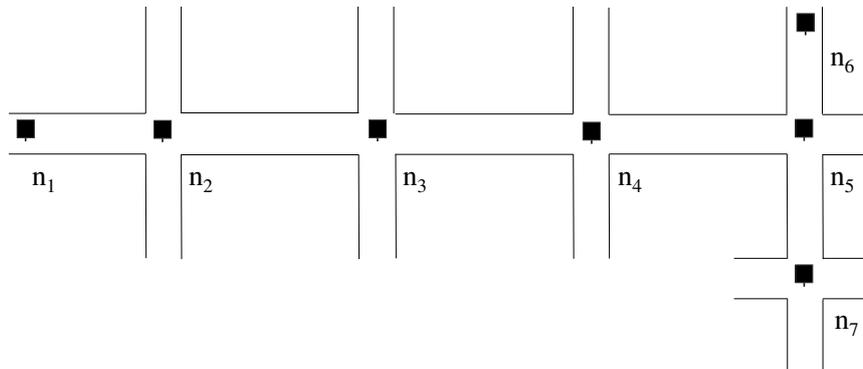


Figure 3. Example Scenario of Packet Forwarding in RDV (Reliable Distance-Vector routing): A packet is to be forwarded from n_2 to n_7 with the routing tables of n_2 and n_5 shown in Tables 2 and 3, respectively.

4. MP-RDV: The Proposed Scheme

4.1. Overview

In this paper, we extend RDV and design MP-RDV in terms of the two drawbacks of RDV. The two drawbacks are the load-balancing and the delivery delay performance described as follows.

On Load Balancing Performance. It is observed in [13] that throughput performance is largely degraded when the volume of data traffic exceeds a certain level. This is due to concentration of data traffic on the shortest-path links. Different from SADV that distributes packet copies into the primary and the secondary next-hops, RDV transmits packet copies into the shortest paths only, which makes the load balancing performance degraded. In MP-RDV, we propose to use multiple paths at each node, i.e., we distribute data traffic into primary and the secondary next-relay nodes. With the multi-path extension to RDV, we show that the load-balancing performance as well as the network capacity significantly improve.

On Delivery Delay Performance. Simulation results of [13] show that the delay performance of RDV is considerably worse than SADV. This is mainly because of the property of the deployed routing metric. That is, SADV uses the routing metric based on delivery delay, whereas RDV's metric is designed based on the volume of vehicle traffic on roads. As a result, in exchange for better delivery ratio, RDV's delay performance is lower than SADV. The countermeasure for this problem is twofold. First, we extend RDV to use multiple paths, which not only improves the load balancing performance but may also improve delivery delay because some of the multiple paths may have smaller delay than the shortest path. Second, we try to make paths selection based on delivery delay. Since delay metric does not work simply in the RDV scheme, we design a mechanism to measure delivery delay of each link in cooperation with related nodes so that the delay metric values are advertised in the distance-vector messages in RDV (as well as MP-RDV).

4.2. Selection of Multiple Paths

The basic strategy for multi-paths selection in MP-RDV is to use the primary and the secondary next-relays simultaneously, i.e., each static node forwards the copies of a packets to both relay nodes. However, a naive deployment of this strategy causes several problems.

The first problem is the so-called routing loop problem, which is shown in Figure 4. In this example, nodes n_3 and n_8 have packets that destines to the right side, e.g., n_{11} . Node n_3 has the primary relay n_5 and the secondary relay n_8 , while n_8 has the primary relay n_{10} and the secondary n_3 . If n_3 and n_8 distribute the packet copies into both relays, packets easily loop between n_3 and n_8 that consume the bandwidth and degrades communication performance significantly.

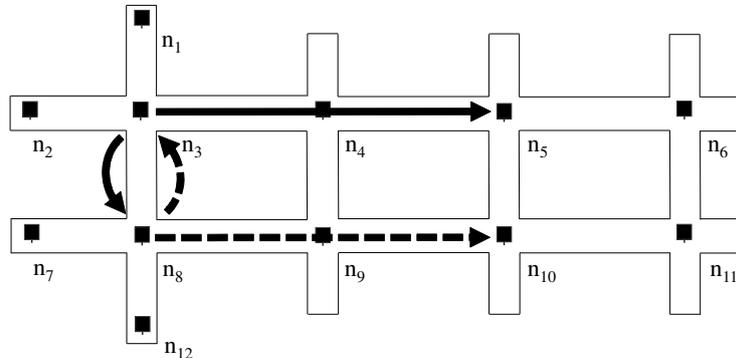


Figure 4. The Routing Loop Problem: When the secondary relay for a destination at n_3 is n_8 and the one at n_8 is n_3 , packets on secondary paths would make routing loops.

We name the second problem “same-street problem,” which is shown in Figure 5. In this case, suppose that node n_3 has packets whose destination lies on the right side, e.g., n_{11} . If the primary relay is n_5 and the secondary is n_4 , all packets are going into the same street, defeating the purpose of load balancing.

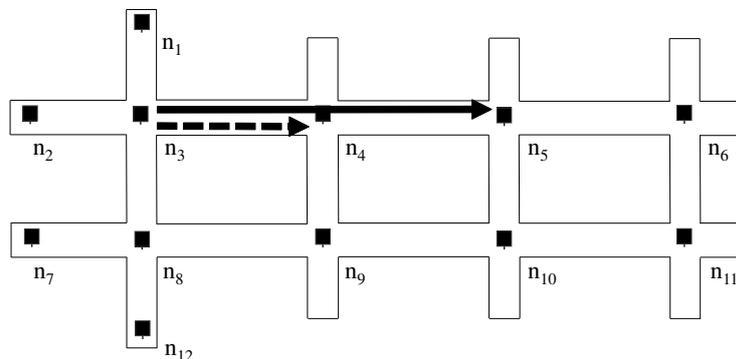


Figure 5. The Same-street Problem: When the primary and secondary relay for a destination at n_3 is n_4 and n_5 , respectively, those paths are along the same street and no load balancing effect is expected.

Unfortunately, both problems easily occur frequently, which makes the multi-path mechanism useless. We must design the method that takes advantage of the merit of multi-path strategy and simultaneously prevents the two harmful defects. To this end, we propose to apply the following two conditions to utilize multiple paths. That is, when a packet is at a static node, only if both of those conditions are met, both the primary and the secondary relays are activated for the packet, and only the primary relay is used otherwise.

- (A) The distance in *carry* of the secondary path is the same as or less than the primary path.
- (B) The previous-hop node of the secondary relay is different from that of the primary relay.

Condition (A) is applied to prevent routing loops in combination with the primary and the secondary relays. If packets are always forwarded to decrease the distance in *carry*, they will surely not create loops. Condition (B) is applied to prevent the same-street problem. Since RDV passes packets to the vehicles that comes from the previous-hop node that has the largest probability to visit

the next-relay nodes, the vehicles from different previous-hop node are expected to go for different directions and visit different next-relays in many cases. One of the exceptions is when two roads are joining into a road, in which vehicles from those two source roads will go to the same direction. Excluding the exceptional cases, this rule solves the same-street problem with high probability.

The flowcharts of RDV and MP-RDV procedures in updating routing tables are shown in Figure 6. In RDV, when a 1-carry or multi-carry message arrives, a node applies the distance-vector next-hop selection algorithm per destination $d \in D$, where D is a set of destinations included in the message. Specifically, for each d , a node compares the distance of old and new routes and updates the routing-table entry with the shorter one. In contrast, MP-RDV maintains the best and the second-best routes in the routing table. Namely, when a new entry for destination d comes with a message, a node compares the new route with two old routes, and updates the routing table entries with the best and the second-best ones. In case of MP-RDV, after this routing-table update, the node tests the second-best route whether it meets the above conditions (A)(B) or not. The second-best route is activated if both of them meet, and is inactivated otherwise.

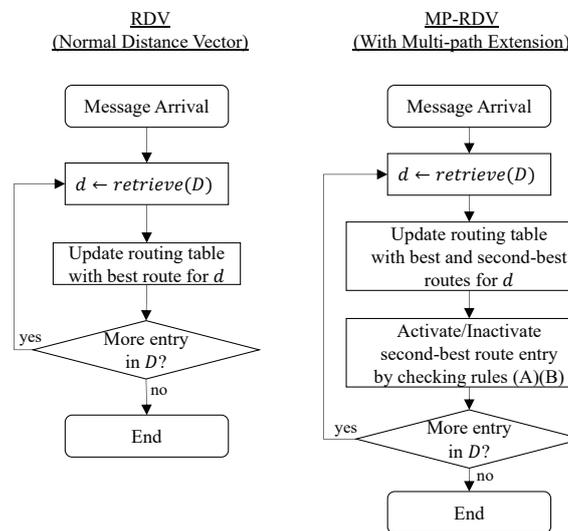


Figure 6. The flowcharts of routing-table updates in RDV and MP-RDV (Multi-Path RDV): MP-RDV maintains both the primary and secondary relays in the routing tables, and makes decisions on activating the secondary relay.

4.3. Extending APD for Multiple Paths

The next problem in designing MP-RDV is how to distribute the duplicated copies of a packet into multiple paths while preserving the expected packet delivery ratio. This is solved by extending the APD mechanism of RDV to fit to the multi-path case.

Similar to the discussion of APD in RDV, let P be the preconfigured end-to-end expected delivery ratio to suffice. A node has two next-relays n_p and n_s , which are the primary and the secondary ones. Let p_p and p_s be the corresponding paths to the destination, and we let the distance in carry of each be δ_p and δ_s , respectively. Since MV-RDV forks delivery paths at every node in the forwarding paths, we further define variables related to the expected delivery ratio. Let P_x be the expected delivery ratio from node x to the destination. Note that $P_x = P$ if x is the source node of the packet.

A node x about to forward a packet first computes the expected delivery ratio for each path to the destination. If x has two paths p_p and p_s to deliver packets, we let the expected delivery ratio for each of them be the same value P'_x . To achieve the expected delivery ratio P_x using those two paths, P'_x must hold the following equation:

$$1 - P_x = (1 - P'_x)^2, \quad (5)$$

which means that the packet loss probability from x to the destination is the same as the product of packet loss probability of two paths. From this, we obtain the following:

$$P'_x = 1 - \sqrt{1 - P_x}. \tag{6}$$

By applying P'_x into the Formulas (1)–(3), we can obtain the number of duplicated packet copies D_l required for each path p_p and p_s to suffice P_x .

An example of APD in MP-RDV is shown in Figure 7. Suppose that a packet is generated at n_1 and destined n_4 . Since $P_1 = P = 99\%$, we obtain the expected delivery ratio for each path from Formula (6) as $P'_1 = 90\%$. From Formula (1), the expected delivery ratio E_l for each link l in p_p and p_s is computed as 96.54%. Thus, from Formula (3), and provided that the delivery ratio estimated by SVS is $R_{(n_1,n_2)} = 80\%$, n_1 forwards $D_{(n_1,n_2)} = 2$ packets for the relay node n_2 . Similarly, for $R_{(n_1,n_5)} = 70\%$, n_1 forwards $D_{(n_1,n_5)} = 3$ packets for n_5 .

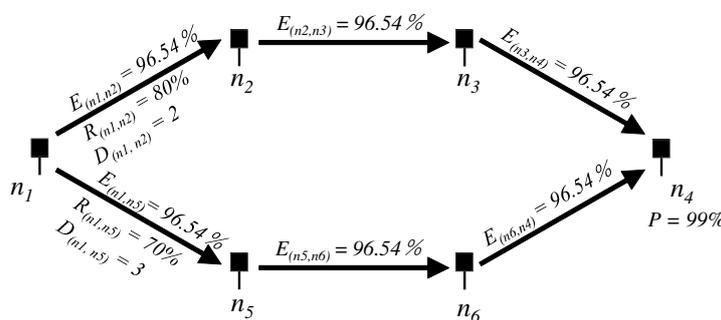


Figure 7. Example of APD in MP-RDV: When two relays are available, nodes such as n_1 expect the same delivery ratio on both paths and computes the number of duplicated packets on each of them.

Note that the intermediate nodes such as n_2 and n_5 again fork paths. The required delivery ratio on n_2 , which is P_2 , is computed by the aggregation of expected delivery ratios in the primary path. Namely, in Figure 7, $P_2 = E_{(n_2,n_3)}E_{(n_3,n_4)} = 93.20\%$. Note that $E_{(n_2,n_3)}$ and $E_{(n_3,n_4)}$ take the same value and the value is notified by the parent node n_1 using the header space of data packets. This is formally written as follows. Let x be the parent node of y and δ_p be the distance in carry from y to the destination. Then, the expected delivery ratio on y is computed as $P_y = \delta_p E_l$, where E_l is notified by x . Once the value P_y is computed, y again distributes packet copies using multiple next-relays in the same process as described above.

4.4. Designing Delay Metrics for MP-RDV

To compare the performance of delivery delay metric with RDV’s built-in traffic-volume-based metric, we design the mechanism to apply the delay metric to MP-RDV. As a result, in MP-RDV, we can deploy either the traffic-volume metric or the delay metric described in this section. Same as the traffic-volume metric, the delay metric values are measured in cooperation with related nodes. The delivery delay on a path is defined as the sum of the transportation delays between nodes and the waiting delays on nodes on the path. The transportation delay is the time required for packets to travel among nodes, and the waiting delay is the time required for packets to wait on nodes until appropriate vehicles to pass the packet come. We designed the mechanism to collect those two sort of values and advertise them so that the delivery delay of any selected path is known in our distance-vector schemes.

Waiting delays are measured by *Hello* messages of RDV. We let $W(n_a, n_b)$ be the average waiting delay on n_b for packets waiting for vehicles coming from n_a . Namely, n_a is the previous-hop node for the packets’ destinations. In this case, node n_b measures $W(n_a, n_b)$.

With Figure 8, we show the process to measure $W(n_a, n_b)$. Let $I_i (i = 1, 2, \dots, k)$ be the time difference of the two sequential vehicles that arrives at n_b from n_a . Within the time interval T ,

the vehicles arriving at n_b is denoted by $c_i (i = 1, 2, \dots, m)$ sequentially, i.e., the first vehicle in T arriving at n_b is c_1 , and the last one is c_m . We assume that the time on which packets arrive at n_b follows a uniform distribution. Then, if a packet arrives at n_b within the period of time T , the probability that it appears in I_i is $\frac{I_i}{T}$, and its average waiting time is $\frac{I_i}{2}$. Thus, the average waiting time $W(n_a, n_b)$ is expressed as follows:

$$W(n_a, n_b) = \sum_{i=1}^k \frac{I_i}{T} \frac{I_i}{2}. \quad (7)$$

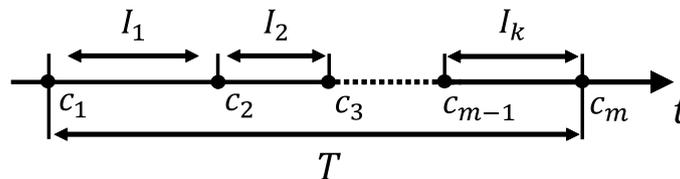


Figure 8. Illustration on measuring waiting delay: The waiting delay at a node is computed from the arriving time of m vehicles within the time period of T .

On the other hand, the transportation delay is measured through polling packets. Note that we define a new type of packets and pass a polling packet to every vehicle visiting the node. Every polling packet is carried to a neighbor node and the node gets to know the transportation delay between neighboring nodes. Note that *Hello* packets are not used for this purpose because the transportation process of RDV allows relaying packets among vehicles. Let n_a be a neighbor of n_b . Then, when a vehicle visits n_a and n_b in this sequence, n_b gets to know the transportation delay, denoted by $D(n_a, n_b)$, between those two nodes n_a and n_b . The average transportation delay $\bar{D}(n_a, n_b)$ in the period of time T is computed from all the delay values collected on n_b in T .

If we assume time synchronization among every node, we can measure both the waiting and the transportation delays easily. However, even if we do not assume time synchronization among nodes, we can measure the waiting delay with clocks on stationary nodes, and the transportation delay with clocks on vehicles. Note that not so much accuracy is required to measure those delay values.

The next step is to sum up those waiting and transportation delays as the routing metric values of links. See Figure 9. We expect a packet forwarded from n_x is relayed by n_b to reach n_d . When a vehicle moves from n_d via n_c to n_b , the measured average delays $\bar{D}(n_b, n_c)$ and $\bar{D}(n_c, n_d)$ are collected to n_b by RDV messages. On the other hand, n_b can compute the waiting delay for the previous node n_a , i.e., $W(n_a, n_b)$. Since the delay on link (n_b, n_d) is the sum of the waiting delay on n_b (i.e., $W(n_a, n_b)$), and transportation delay $\bar{D}(n_b, n_c) + \bar{D}(n_c, n_d)$, the delay metric of link (n_b, n_d) corresponding to the previous-hop node n_a is computed as

$$M^{(dly)}(n_a, n_b, n_d) = W(n_a, n_b) + \bar{D}(n_b, n_c) + \bar{D}(n_c, n_d). \quad (8)$$

This metric value is computed at n_b and advertised to n_x through RDV messages, and then n_x is possible to forward packets to n_d via a relay node n_b . In this way, the delay metrics of links are computed within the framework of RDV and advertised for being used in routing decision.

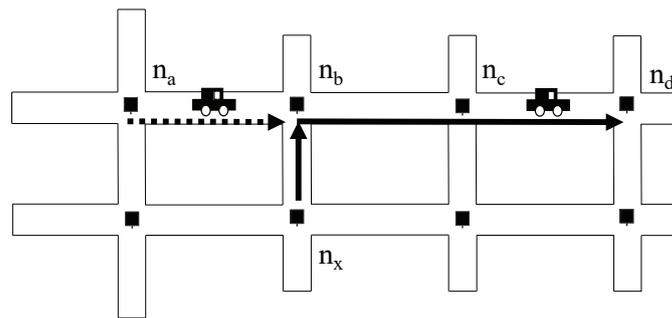


Figure 9. Illustration on computing delay metrics: The delay metric from n_b to n_d is computed as the sum of the waiting delay at n_b and the transportation delays on (n_b, n_c) and (n_c, n_d) .

5. Evaluation

5.1. Methods

We evaluate the performance of MP-RDV through simulation and compared it with RDV and SADV. The main criteria in comparison is the basic communication performance measurement such as delivery ratio and delay under variations of vehicle density and data traffic amount. In addition, as a criterion for DTN routing, we measured the traffic load due to packet copies as well. We designed a scenario with the Kyoto city map in which we simulate the behavior of vehicles and packet delivery. We also measure the fault tolerance of MP-RDV under road failure; we designed a failure scenario in which roads are partially blocked, pretending a traffic accident occurred at the intersection.

We developed our own simulator with C++ language to simulate several routing protocols in relatively sparse vehicular networks. Because we evaluate the performance of MP-RDV in a sparse scenario in which networks behave as a DTN, we do not care about PHY and MAC protocols. Thus, in our simulation, we just assume that a certain amount of packet exchanges is possible when a vehicle meets another vehicle or a static node, as is usually assumed in evaluation of DTN schemes.

We obtained a Kyoto roadmap data set in Open Street Map [28] and removed small streets from it to retrieve major roads as shown in Figure 10. In this map, we have 104 intersections in total, at all of which we place traffic signals and static nodes. As a result, static nodes are placed only at major intersections in the Kyoto city map. To generate vehicle traffic over the map, we used a traffic simulator SUMO [29]. The map includes 20 road ends, at all of which we generate vehicles with a constant time interval. We set the maximum speed of vehicles as 50 km/h. In SUMO, we adopt a probabilistic mobility model; in every intersection, 90% of vehicles go straight, and 5% turn right and left, respectively. If vehicles are not allowed to go straight at an intersection, 50% of them turn left and the rest right. Each vehicle has a buffer (i.e., a transmission queue) that holds 128 data packets. We assume that each static node has a transmission queue for each previous hop (remember that RDV distinguishes previous hops to select vehicles to pass packets), the size of which is 1280 packets each. Nodes and vehicles have the communication range 50 m, i.e., vehicles work as *intersection mode* if they are within the range of a static node, and as *road mode* otherwise.

In RDV and MP-RDV, every node updates its statistics and the routing tables every time the node exchanges messages with a vehicle. This assumption offers the most sensitive updates of the statistics and the routing table, while we can take choices of less frequent updates to reduce the computational load. We set the expected delivery ratio $P = 99\%$. As for other parameters of RDV and MP-RDV, for which detailed descriptions are seen in [13], we set values as follows: We set the capacity of multi-carry messages per vehicle $M_{MC} = 20$ (i.e., a vehicle deletes a multi-carry message randomly when the loaded messages exceed this value). We set the maximum number of hops for a 1-carry node $N_{carry} = 10$ (i.e., 1-carry links must be within 10-hops long). We set the minimum expected packet delivery ratio to be a relay node $T_{SC} = 70\%$ (i.e., 70% delivery ratio is required to establish 1-carry links).

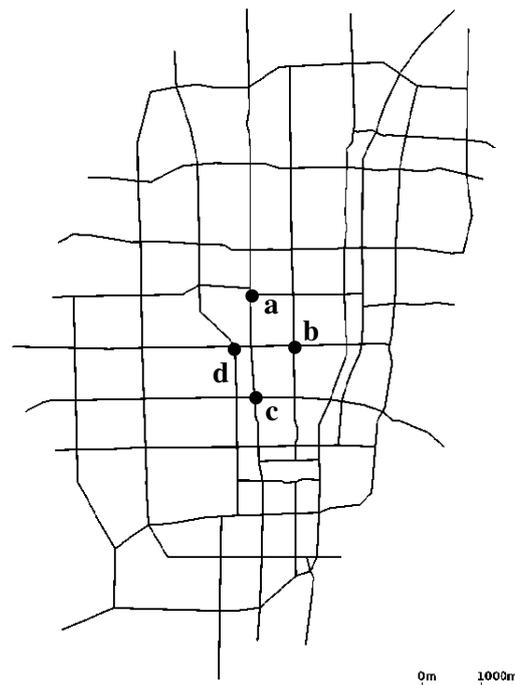


Figure 10. Kyoto City Map used in simulation.

For SADV, as preliminary tests, we measured the average delay of each link, i.e., each pair of neighbor static nodes with 1-hop distance, and used the fixed values in the shortest-path computation. We set TTL as the length of the shortest path plus 4, which allows using at most 4-hop longer paths than the shortest one as detour routes. From the preliminary tests, we found that the performance such as delivery ratio significantly improves if this value is +4 compared to +2 and 0, and that the values larger than +4 does not improve the performance but increasing the load of packet copies.

We transmit data packets for 1000 s after a 3600-s wait to obtain stable routing tables. In the first scenario, in which we measure the communication performance, we generated 1000 packets during the simulation time between randomly-selected static node pairs with a constant time interval. We generate 3–8 vehicles per minute per road-end, which is a relatively sparse traffic state. Note that three vehicles per minute would be sparse enough to demonstrate the case of rural area or night-time scenarios. We take the average of 10 repetitions to show the results.

To consider the uncertainty of vehicle behavior under sparse deployment of static nodes, we make vehicles pretend to leave major roads for entering minor streets or parking lots along the road, and again join in to major roads. Because we do not have static nodes at minor intersections in the sparse scenario, we cannot trace a part of vehicles that escapes into minor streets, etc. Thus, we treat the uncertain behavior of vehicles as a probabilistic phenomenon. Specifically, we probabilistically select vehicles at each section of roads between two static nodes (i.e., at each links), and flush (i.e., delete) all packets and messages held on them. This *flush operation* pretends that a vehicle leaves the road and simultaneously another vehicle joins. We let the probability of flushing at each road section be 0%, 1.0%, and 2.5% to measure the effect of uncertainty of vehicle movement.

On the other hand, in the second scenario in which we measure the fault tolerance, we block the intersections a, b, c, and d shown in Figure 10 to pretend a traffic accident in the intersection. This traffic accident occurs at 500 s after the starting time of data transmission. With this scenario, we observe the degree of packet loss during the time period before converging to the new shortest paths set.

5.2. Results

Figure 11 shows the results on basic communication performance. Figures 11a–c show the performance comparison results under the flush rate 0%. In Figure 11a, we see that delivery ratio is all good and comparable. Note that both RDV and MP-RDV exceed the predefined expected delivery ratio $P = 99\%$. In Figure 11b, we see the trend that delivery delay naturally reduces as vehicle density increases. Note that MP-RDV significantly improves delay compared to RDV, and the value is close to SADV. In Figure 11c, we see that the number of SADV's packet copies is far larger than the others. Note that the difference between RDV and MP-RDV comes from the rounding error of the ceiling function shown in Formula (3). Those results clearly show the characteristics of the three schemes; all achieve a high delivery ratio while SADV requires a large amount of duplication load, RDV expenses delivery delay, and MP-RDV performs the best without degradation in neither of them.

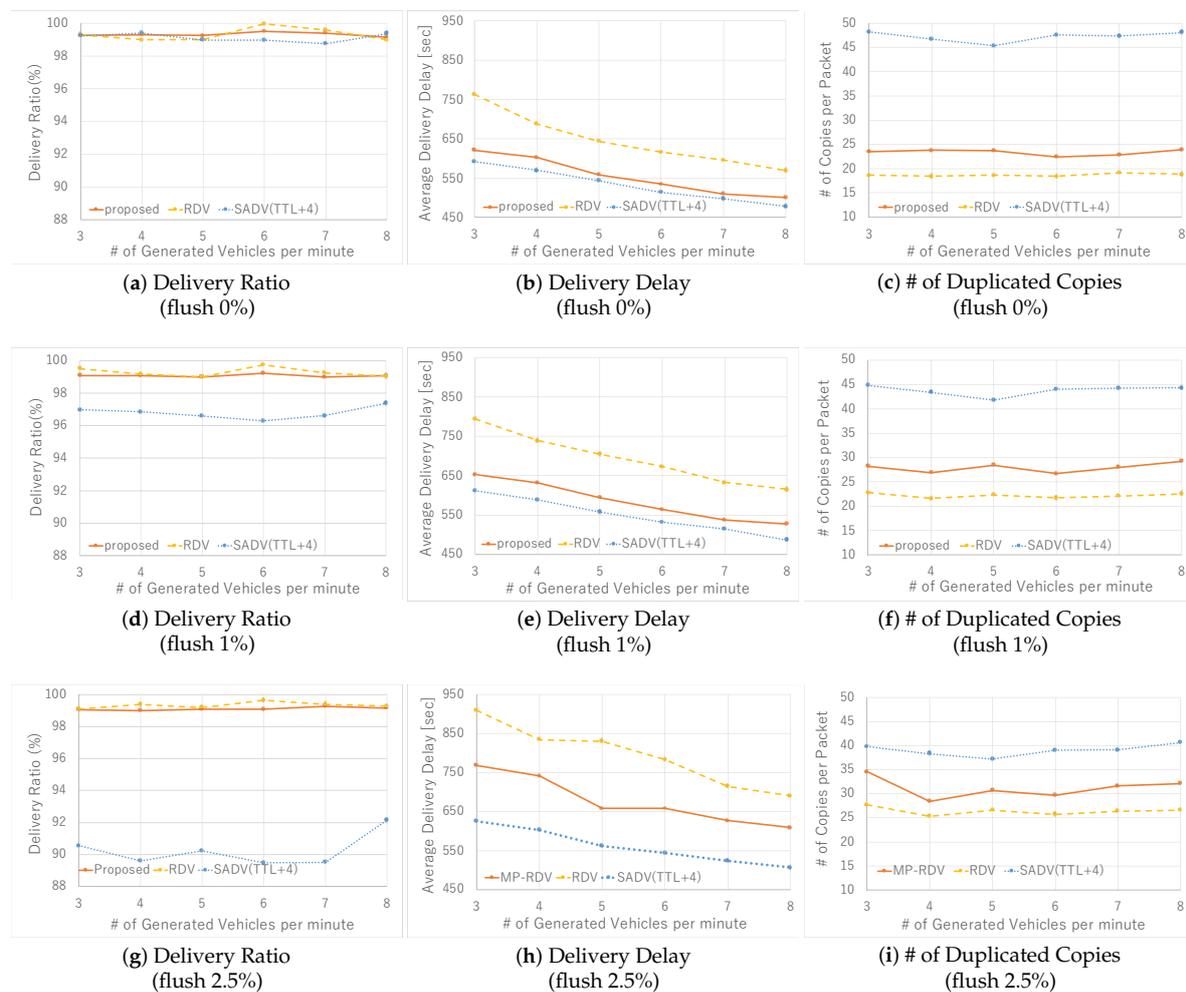


Figure 11. Performance with various vehicle density: delivery ratio, delivery delay, and the number of duplicated copies are measured for each flush rate 0%, 1%, and 2.5%.

As for the effect of flush events, i.e., the effect of uncertainty of vehicle behavior—see Figure 11d–i. The case of flush rate 1% is shown by Figure 11d–f. Although the trend is not so different from the case of 0%, we find that the delivery ratio of SADV is significantly degraded. This means that SADV is so sensitive to flush rate, whereas Adaptive Packet Duplication (APD) deployed in RDV and MP-RDV well works to keep the delivery ratio to 99%. The cases when the flush rate is as high as 2.5% are shown in Figure 11g–i. The same trend as 1% case is still kept, and RDV and MP-RDV still keep 99% delivery ratio. In Figure 11h, we see that the delay of RDV and MP-RDV is increased, meaning that RDV and MP-RDV is sensitive to flush rate on delay performance. In Figure 11i, the number of packet copies for

RDV and MP-RDV increases, which is natural because they require larger number of packet copies when the packet loss ratio goes high. In contrast, SADV’s delivery ratio in Figure 11g is significantly degraded and the load due to packet copies in Figure 11i is reduced as well. This is because many packets are lost early before generating many packet copies, meaning that SADV is very weak against flush events. To summarize the effect of flush rate, we see that RDV and MP-RDV are able to keep the given delivery ratio 99% although delay and duplication load increases to some extent, whereas the performance of SADV significantly degrades due to packet loss.

Performance under variation of traffic load is shown in Figure 12. With flush rate 0%, we see that RDV fails to keep 99% delivery ratio when generating over 30,000 packets, whereas MP-RDV keeps 99% even at 50,000 packets. From this, we clearly conclude that MP-RDV has a larger traffic capacity due to the load balancing function over the multiple paths. In a 1% case, the same trend is seen although MP-RDV fails to keep 99% delivery at the point of 30,000 packets generation. In a 2.5% case, all fail to keep a high delivery ratio, while MP-RDV marks the best performance among them. In summary, the results show that MP-RDV has the largest network capacity among three, and has the highest ability to keep the expected delivery ratio.

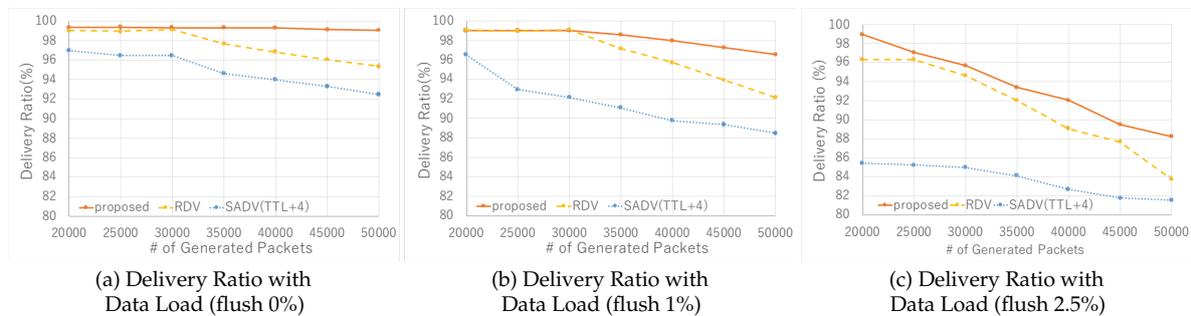


Figure 12. Delivery ratio with various data loads: delivery ratio under variation of data load is shown for each flush rate 0%, 1%, and 2.5%.

Finally, in Figure 13, we show the results on robustness against failure. We see that the performance of MP-RDV and SADV is better than RDV regardless of flush ratio, which is because multi-paths improve the robustness against failure. This means that the robustness of RDV against failure is improved with multi-path functionality.

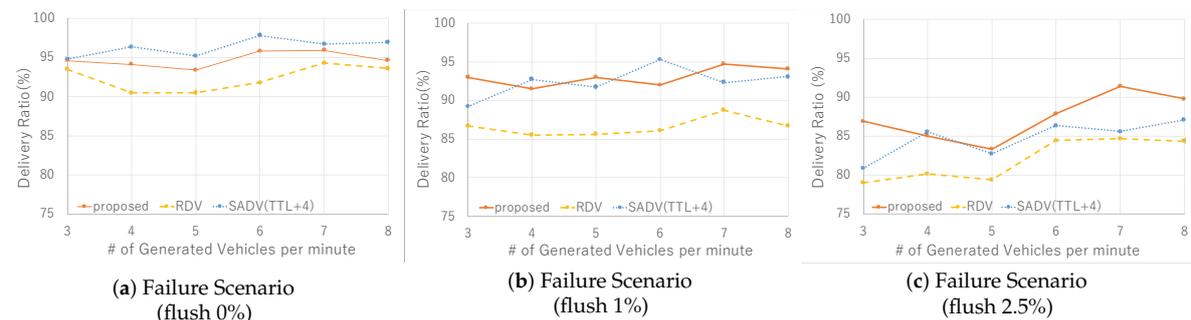


Figure 13. Performance in failure scenario: delivery ratio sensitively degrades as flush rate increases.

5.3. Evaluating Delay Metrics

We compared the performance of two routing metrics, i.e., the traffic volume metric and the delivery delay metric, under the same evaluation procedure as described above. We fix the routing scheme to MP-RDV and see the effects of the routing metrics and flush ratio.

Figure 14 shows the results. First, we focus on the case of 0%, and find that both traffic volume and delivery delay metrics have no difference in all of those three performance criteria. This is because

traffic volume has relatively high correlation with delivery delay so that almost the same set of paths has been selected as a result.

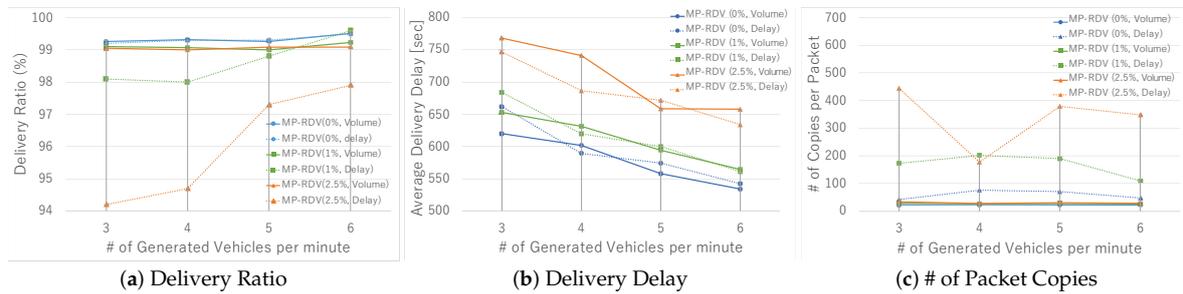


Figure 14. Comparing two routing metrics: delivery ratio, delivery delay, and the number of packet copies are compared with both delay and volume metrics.

Second, when we see the difference in flush ratio, we find that the delay metric significantly reduces its performance as flush ratio increases in delivery ratio shown in Figure 14a, whereas the volume metric remains at 99%. This trend is the same in the load of packet copies shown in Figure 14c. The reason why this occurs is that delay metric has a large fluctuation due to traffic signals and vehicle traffic state. Thus, statistically the delay metric is less stable than the volume metric especially when flush ratio increases, which impacts the performance. When routing metrics fluctuate in a large range, routing paths also change frequently. This causes instability of networks and then the delivery ratio as well as the load of copies get worse as shown in Figure 14a,c.

In Figure 15, we show the performance under variation of generated data rate. The trend seen here is that performance gradually reduces as the data load increases, while performance of delay metric is lower than that of volume metric. This is also caused by the instability of the delay metric. Figure 16 shows the results of failure scenario, in which again the performance of the delay metric is lower than the volume metric.

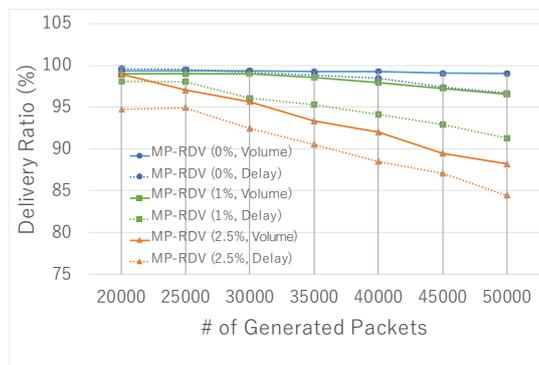


Figure 15. Metric Comparison under Variation on Data Rate.

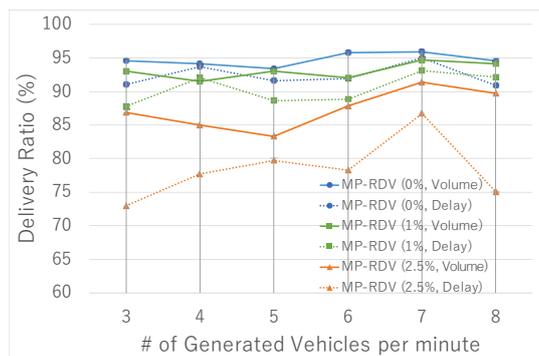


Figure 16. Metric Comparison in Failure Scenario.

6. Conclusions

We proposed two extensions on RDV to further improve the communication performance in static-node-assisted vehicular networks. From comparison with SADV, we extracted two points of potential improvement and designed them as an extension to RDV. The first extension is MP-RDV, a multi-path extension to RDV in which multiple paths are utilized for forwarding each of the packets to avoid load concentration on the shortest paths as well as to improve the delay performance. The second extension is the delay metric that measures the expected delivery delay as the sum of the waiting delay on nodes and the transportation delay between nodes. Our extensions inherit the characteristic of RDV to provide reliable communications that satisfies a pre-configured delivery ratio expectancy with a minimum packet duplication overhead. RDV achieves this in the following mechanisms. First, SVS measures statistics of vehicular movement at each static node located at major intersections to predict the outgoing direction of vehicles, and, second, APD controls the number of duplicated packets to achieve the required expected packet delivery ratio while minimizing the load of duplicated packets. The multi-path selection in MP-RDV cares for selecting two next-relays at each static node to distribute packets into the optimal paths while preventing routing loops. APD is also extended to handle multiple paths such that the load is well balanced among paths while guaranteeing preconfigured expected packet delivery ratio. The delay metric is also designed to work in the framework of RDV as well as MP-RDV. The waiting delay and the transportation delay are measured in cooperation with related nodes, and the measured metrics are advertised through the distance-vector messages in RDV.

Evaluation results show that MP-RDV has better performance in total than the conventional schemes considering the balance of several important measurements. MP-RDV is proven to have well-maintained delivery ratio, lower delay, high network capacity with load-balancing effect, and better fault tolerance performance. This shows that the multi-path strategy is beneficial in RDV schemes. On the other hand, the delay metric has been proven to have reduced performance compared to the traffic-volume metric that was proposed as a built-in metric in RDV. We found that, although those two metrics chose almost the same set of forwarding paths under stable vehicular traffic patterns, the delay metric is statistically unstable under fluctuation on traffic patterns coming from traffic signals, unforeseen driver's behaviors, etc. Finally, we conclude that multi-path strategy with traffic-volume routing metric is the best choice for the RDV scheme.

In considering future work, we would note that this work is a pure theoretic framework that does not use any practically available observable data such as several sensors on vehicles, current location, and current moving directions. Enhancing the proposed framework utilizing those sensor data is one of the most interesting directions for the future.

Author Contributions: D.A. and T.Y. conceived and designed the proposed method MP-RDV. D.A. implemented the simulation software and conducted simulation work. T.Y. directed the analysis and evaluation work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Vahdat, A.; Becker, D. *Epidemic Routing for Partially Connected Ad Hoc Networks*; Duke University Tech Report CS-2000-06; Duke University: Durham, NC, USA, 2000.
2. Yoo, J.; Choi, S.; Kim, C.K. The capacity of epidemic routing in vehicular networks. *IEEE Commun. Lett.* **2009**, *13*, 459–461.
3. Spyropoulos, T.; Psounis, K.; Raghavendra, C.S. Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. In Proceedings of the ACM SIGCOMM Workshop on Delay Tolerant Networks '05, Philadelphia, PA, USA, 26 August 2005; pp. 252–259.
4. Lochert, C.; Mauve, M.; Fusler, H.; Hartenstein, H. Geographic Routing in City Scenarios. *ACM Sigmob. Mob. Comput. Commun. Rev.* **2005**, *9*, 69–72. [[CrossRef](#)]

5. Lee, K.; Haerri, J.; Lee, U.; Gerla, M. Enhanced Perimeter Routing for Geographic Forwarding protocols in Urban Vehicular Scenarios. In Proceedings of the IEEE Globecom Workshops, Washington, DC, USA, 26–30 November 2007; pp. 1–10.
6. Lochert, C.; Hartenstein, H.; Tian, J.; Fussler, H.; Hermann, D.; Mauve, M. A Routing Strategy for Vehicular Ad Hoc Networks in City Environment. In Proceedings of the IEEE Intelligent Vehicles Symposium, Columbus, OH, USA, 9–11 June 2003; pp. 156–161.
7. Seet, B.C.; Liu, G.; Lee, B.S.; Foh, C.H.; Lee, K.J.; Lee, K.K. A-STAR: A Mobile Ad Hoc Routing Strategy for Metropolis Vehicular Communications. In *Networking 2004*; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3042, pp. 989–999.
8. Zhao, J.; Cao, G. VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks. *IEEE Trans. Veh. Technol.* **2008**, *57*, 1910–1922. [[CrossRef](#)]
9. Wu, H.; Fujimoto, R.; Guensler, R.; Hunter, M. MDDV: A Mobility-centric Data Dissemination Algorithm for Vehicular Networks. In Proceedings of the VANET'04, Philadelphia, PA, USA, 1 October 2004; pp. 47–56.
10. Jerbi, M.; Senouci, S.M.; Meraihi, R.; Ghamri-Doudane, Y. An improved Vehicular Ad Hoc Routing Protocol for City Environments. In Proceedings of the IEEE International Conference on Communication (ICC), Glasgow, UK, 24–28 June 2007; pp. 3972–3979.
11. Zhang, X.M.; Chen, K.H.; Cao, X.L.; Sung, D.K. A Street-centric Routing Protocol Based on Micro Topology in Vehicular Ad hoc Networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 5680–5694. [[CrossRef](#)]
12. Ding, Y.; Xiao, L. SADV: Static-Node-Assisted Adaptive Data Dissemination in Vehicular Networks. *IEEE Trans. Veh. Technol.* **2010**, *59*, 2445–2455. [[CrossRef](#)]
13. Yoshihiro, T.; Araki, D.; Sakaguchi, H.; Shibata, N. Providing Reliable Communications over Static-node-assisted Vehicular Networks Using Distance-vector Routing. *Mob. Netw. Appl.* **2018**, *23*, 1376–1393. [[CrossRef](#)]
14. Araki, D.; Yoshihiro, T. A Multi-path Extension to RDV Routing Scheme for Static-node-Assisted Vehicular Networks. In Proceedings of the IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA2018), Krakow, Poland, 16–18 May 2018.
15. Karagiannis, G.; Altintas, O.; Ekici, E.; Heijnen, G.; Jarupan, B.; Lin, K.; Weil, T. Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions. *IEEE Commun. Surv. Tutor.* **2011**, *13*, 584–616. [[CrossRef](#)]
16. Electronic Road Pricing. Available online: https://en.wikipedia.org/wiki/Electronic_Road_Pricing (accessed on 28 February 2019).
17. Fujimoto, A.; Sakai, K.; Hamada, M.O.S.; Handa, S.; Matsumoto, M.; Takahashi, K. Toward Realization of SMARTWAY in Japan. In Proceedings of the 15th World Congress on Intelligent Transportation Systems (ITSWC'08), New York, NY, USA, 16–20 November 2008.
18. Jeong, J.; Guo, S.; Gu, Y.; He, T.; Du, D.H.C. Trajectory-Based Statistical Forwarding for Multihop Infrastructure-to-Vehicle Data Delivery. *IEEE Trans. Mob. Comput.* **2012**, *11*, 1523–1537. [[CrossRef](#)]
19. Skordylis, A.; Trigoni, N. Delay-bounded Routing in Vehicular Ad hoc Networks. In Proceedings of the MobiHoc'08, Hong Kong, China, 26–30 May 2008; pp. 341–350.
20. Jarupan, B.; Ekici, E. PROMPT: A Cross-layer Position-based Communication Protocol for Delay-aware Vehicular Access Networks. *Ad Hoc Netw.* **2010**, *8*, 489–505. [[CrossRef](#)]
21. Wu, Y.; Zhu, Y.; Li, B. Infrastructure-assisted Routing in Vehicular Networks. In Proceedings of the IEEE Infocom'12, Orlando, FL, USA, 25–30 March 2012.
22. O'Driscoll, A.; Pesch, D. Hybrid geo-routing in urban vehicular networks. In Proceedings of the 2013 IEEE Vehicular Networking Conference, Boston, MA, USA, 16–18 December 2013; pp. 63–70.
23. Li, P.; Huang, C.; Liu, Q. Delay Bounded Roadside Unit Placement in Vehicular Ad Hoc Networks. *Int. J. Distrib. Sens. Netw.* **2015**, *2015*, 1–14. [[CrossRef](#)]
24. Xu, L.; Huang, C.; Wang, J.; Zhu, J.; Zhang, L. An Efficient Traffic Geographic Static-node-assisted Routing in VANET. *J. Netw.* **2014**, *9*, 1096–1102. [[CrossRef](#)]
25. Li, G.; Boukhatem, L.; Wu, J. Adaptive Quality-of-Service-Based Routing for Vehicular Ad Hoc Networks With Ant Colony Optimization. *IEEE Trans. Veh. Technol.* **2017**, *66*, 3249–3264. [[CrossRef](#)]
26. Goudarzi, F.; Asgari, H.; Al-Raweshidy, H.S. Traffic-Aware VANET Routing for City Environments—A Protocol Based on Ant Colony Optimization. *IEEE Syst. J.* **2019**, *13*, 571–581. [[CrossRef](#)]

27. Celes, C.; Boukerche, A.; Braga, R.B.; Ramos, H.S.; Andrade, R.M.C.; Loureiro, A.A.F. Exploiting Daily Trajectories for Efficient Routing in Vehicular Ad Hoc Networks. In Proceedings of the International Conference on Communications (ICC2018), Kansas City, MO, USA, 20–24 May 2018.
28. Open Street Map. Available online: <https://www.openstreetmap.org/> (accessed on 28 February 2019).
29. Behrisch, M.; Bieker, L.; Erdmann, J.; Krajzewicz, D. SUMO—Simulation of Urban MObility: An Overview. In Proceedings of the SIMUL2011, Barcelona, Spain, 23–29 October 2011; pp. 23–28.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).