Multi-Flow Congestion-Aware Routing in Software-Defined Vehicular Networks

Antonio Di Maio
University of Luxembourg
Luxembourg
antonio.dimaio@uni.lu

Maria Rita Palattella
LIST
Luxembourg
mariarita.palattella@list.lu

Thomas Engel
University of Luxembourg
Luxembourg
thomas.engel@uni.lu

Abstract*—5G-enabled vehicular networks will soon allow their users to exchange safety and non-safety related information over heterogeneous communication interfaces. Routing vehicular data flows over multi-hop Vehicle-to-Vehicle (V2V) communications is one of the hardest challenges in vehicular networking, and it has been tackled in literature by using distributed algorithms. The distributed approach has shown significant inefficiencies in such dynamic vehicular scenarios, mainly due to poor network congestion control. To overcome the complexity of the envisioned architecture, and the inefficiency of distributed routing algorithms, we hereby propose to leverage the coordination capabilities of Software-Defined Networking (SDN) to determine optimal V2V multi-hop paths and to offload traffic from the Vehicle-to-Infrastructure-to-Vehicle (V2I2V) to the V2V communications, using both cellular and Wi-Fi technologies. In order to achieve this goal, we propose Multi-Flow Congestion-Aware Routing (MFCAR), a centralized routing algorithm that relies on graph theory to choose short and uncongested V2V paths. Realistic simulations prove that MFCAR outperforms well-established centralized routing algorithms (e.g. Dijkstra's) in terms of Packet Delivery Ratio (PDR), goodput and average packet delay, up to a five-fold performance gain.

I. Introduction

Nowadays, we are assisting to a rapid deployment of fifth-generation communication technologies (5G) in some of the major cities in the world, such as Bristol, UK (5G-PICTURE[1]) and Barcelona, Spain (5GCAR[2]). One of the major pillars of 5G is to leverage the power of heterogeneous physical interfaces [1] to transfer vehicular information, which will be fundamental to improve driver's and passenger's comfort in future's connected vehicles. In particular, every vehicle will be part of the 5G-powered Internet-of-Vehicles (IoV) by being connected to the core network through a cellular interface and to other vehicles through a Device-to-Device (D2D) interface.

Delivering information over such complex and dynamic 5G networks requires a superior coordination capability [2]. For this reason, the 5G-PPP consortium has deemed the SDN paradigm as an indispensable component of the 5G architecture [3]. We will refer to the novel concept of SDN-enabled 5G vehicular networks as Software-Defined Vehicular Networking (SDVN) [4].

[1]https://www.5g-picture-project.eu/
[2]https://5gcar.eu/

The data generated by in-vehicle sensors can be exchanged between vehicles for a wide variety of applications, both safety and non-safety [5] related. Transferring this large amount of information through the fixed infrastructure can pose big challenges due to its volume, but also because of the strict delay and efficiency requirements of vehicular applications. Therefore, it is essential to devise methods to offload vehicular traffic from the V2I2V communications to the V2V communications. However, source and destination of a data flow are not always in direct communication range, and a V2V multi-hop path must be established. There exist several works that address the problem of distributed routing in vehicular networks, but they present consistent challenges in delivering satisfactory performances in such dynamic scenarios because of inefficient routing path selection.

For these reasons, as the principal contribution in this work, we propose Multi-Flow Congestion-Aware Routing (MFCAR), a centralized routing algorithm for selecting V2V multi-hop flows in SDVNs. The strength of the proposed approach is the possibility of fine-tuning the V2V routes' optimality criteria between route length and route congestion level, in order to match flow's delay and efficiency requirements. As a second contribution, we introduce a graph-theory-based algorithm to approximate network congestion without performing Clear Channel Assessment (CCA).

The remainder of the article is structured as follows: In Section II, related work on centralized unicast routing in vehicular networks is presented. Section III describes the system model. Sections IV and V describe the operation of the proposed algorithms. Sections VI and VII describe the simulation scenarios and analyze the results. Finally, Section VIII concludes the paper and presents future research directions.

II. Related Work

In [6], the authors propose a distributed flow-allocation scheme for multiple paths in a static wireless multihop networks. They formulate the routing problem as an Integer Linear Programming (ILP) optimization problem, trying to maximize the aggregate flow throughput while providing bounded delay. The authors also model intra and inter flow interference at a physical level. Their proposed algorithm outperforms the well-known flow allocation algorithms in literature, such as round-robin, best path only and maximum flow per path. However, this approach is not suitable for dynamic vehicular networks because of the time complexity of solving an ILP problem compared to the typical link lifetime in vehicular networks. Furthermore, additional delay for the convergence of the algorithm and the delivery and collection of routing control information must be added because of the distributed nature of their approach.

The authors in [7] propose a time-slotted SDVN routing strategy over multiple channels, modeled as an ILP problem. The algorithm aims at maximizing delivered packets and minimizing packet delay, while avoiding routing loops and packet collisions. The authors test their algorithm in ideal conditions and claim that realistic simulations are needed. In realistic scenarios, non-slotted media access protocol are used (e.g. CSMA/CA), and schedulers with finite computational power must be considered, in order to understand whether the proposed per-packet scheduling is feasible. When a vehicle needs to initiate a new data flow, the whole optimization problem must be solved again, posing a scalability issue. Allocating flows using graph search algorithms could make the problem tractable again.

In [8], the authors target the offloading of V2I2V communications to V2V paths, using SDN coordination. The proposed algorithm is based on the widest-path search [9], which operates on the network connectivity graph. Considering that the edges' weights are a euclidean estimation of the link residual lifetime, the algorithm will return the most stable path. This algorithm is designed for a single flow: therefore, in case of multi-flow allocation, very stable paths would be severely congested because the algorithm would always select the single most stable path as optimal, and allocate all the traffic on it.

III. System Model

We assume that each vehicle in the SDVN can communicate with other vehicles through an IEEE 802.11p interface and with the core network through a 5G New Radio (NR) cellular interface. The heterogeneity of the envisioned network will allow vehicles to exchange information directly (V2V) and through the fixed infrastructure (V2I2V) at the same time. In the present work, we will focus on analyzing the performance of the V2V/D2D communications only.

We assume that the V2V multi-hop information exchange is coordinated by a hierarchy of Software-Defined Networking Controllers (SDNCs) [10]. The vehicles communicate with the SDNCs at the base of the hierarchy, which are also called edge controllers. Each edge controller is responsible for the vehicles in a defined geographical area, and it will contact higher-level controllers in case of inter-domain routing. The main task of the SDNCs is computing optimal V2V routes and enforcing flow rules upon the selected intermediate relays. Therefore, the routing intelligence does not lie in the vehicle but in the controllers. The route-selection workflow in the proposed system model is represented in Figure 1.

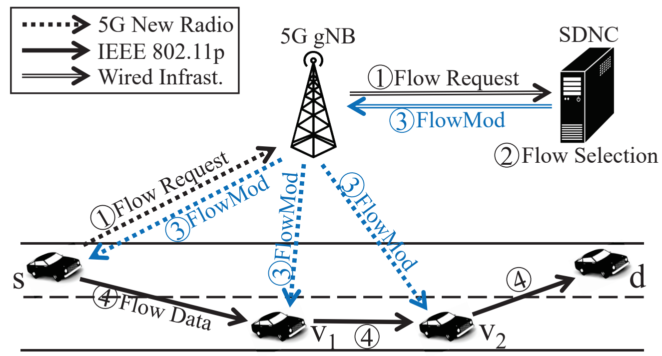


Fig. 1: System model and flow selection workflow. The vehicle asks the SDNC to establish the route (1). Then the SDNC selects it (2) and sends FlowMod messages to the relay nodes (3). Finally, the source vehicle starts transmitting user data over the V2V interface (4).

All the network control messages in our system model are exchanged between vehicles and edge SDNC over the cellular interface, and all the user data are exchanged between vehicles over the V2V interface. In this way, the control and data planes of the network are fully separated, both logically and physically.

A. *Neighbor list and Network connectivity graph*

Vehicles broadcast beacon messages on the Control Channel (CCH) of their IEEE 802.11p interface every T_b seconds, and constantly listen for incoming beacons from other vehicles. Every vehicle collects all the received beacons, timestamps them with the arrival time and creates the list of neighboring vehicles. If a vehicle does not receive a beacon from a neighbor for more than T_e seconds, the neighbor's ID is removed from the neighbor list.

Every T_u seconds, each vehicle sends information about its neighbors to the SDNC, using *delta compression* to save bandwidth. With delta compression we mean that the vehicle does not send the full neighbor list, but only the IDs of vehicles that left and joined the neighbor list in the last T_u seconds. The communication between vehicle and SDNC happens through cellular network.

When the SDNC receives a neighbor list from a vehicle, it uses this information to update the locally-stored *network connectivity graph*. This structure is a directed graph that maintains the information on all the vehicles known by the SDNC and on all the wireless links between them.

B. *SDNC-assisted route selection*

When a vehicle needs to start streaming data to another vehicle, it asks the SDNC, through the cellular network, to allocate a Constant Bit Rate (CBR) data flow of b bps. Followingly, using our proposed MFCAR algorithm, the SDNC selects an "optimal" V2V multi-hop path to deliver the information from source to destination. The definition of path optimality and the operation of the MFCAR routing algorithm will be detailed in Section V. After having selected the relay vehicles that constitute the V2V path, the SDNC modifies the relays' forwarding tables through OpenFlow-like FlowMod messages [11], delivered through cellular network. Followingly, the SDNC records the new flow in the table of active flows, and modifies the network connectivity graph to take into account the impact of the newly allocated flow on the network congestion state. The algorithm for updating the network connectivity graph is presented in Section IV. When the source vehicle terminates the flow, the SDNC removes the impact of that flow from the network connectivity graph.

C. Data plane features

After the SDNC has installed flow rules on the relay nodes of the path, the source vehicle starts sending IEEE 802.11p frames on the Service Channel (SCH). The frames are sent without Automatic Repeat Request (ARQ) mechanism (i.e. the frames are not acknowledged), as this has been considered harmful [12] in vehicular networks.

IV. Network Connecivity Graph Update

After the SDNC has selected a suitable path for the flow, the SDNC must also modify the network connectivity graph to keep track of which vehicles are impacted by the flow allocation in terms of wireless channel congestion and to what extent. The network connectivity graph is updated according to Algorithm 1. We define: p as the newly selected path; v as a relay node $\in p$; $\beta(v) \in [0, +\inf)$ as *business score*, initially set to zero $\forall v \in V$; b as the flow bitrate on the path p, requested by the source s; $\gamma(v) \in (0, 1]$ as *idleness score*, which expresses the expected fitness of the node v to be a relay node; τ as the *time constant*, determined by the physical transmission capacity of the network interfaces (see Appendix A); $N_{\text{out}}(v)$ as the *out-neighbors set*, which contains the vertices connected to v by an outgoing edge from v .

Algorithm 1 Network Connectivity Graph Update

```
for all v ∈ p do
    β(v) = β(v) + b                 ▷ Impact of being a relay
    γ(v) = e^{−β(v)/τ}
    for all x ∈ N_out(v) do
        β(x) = β(x) + b             ▷ Impact of hearing a relay
        γ(x) = e^{−β(x)/τ}
    end for
end for
```

The rationale behind the design of this algorithm is the following: each relay node will occupy the wireless channel to transmit the flow data. Each node in the transmission range of relay nodes (i.e. the nodes belonging to $N_{\text{out}}(v)$) will detect the channel occupied by their transmissions. Therefore, a node x located in the transmission range of several relay nodes will have a high $\beta(x)$ and therefore a small fitness to be selected as relay node for the next flow allocation (i.e. a low $\gamma(x)$).

An example of the operation of Algorithm 1 can be found in Figure 2.

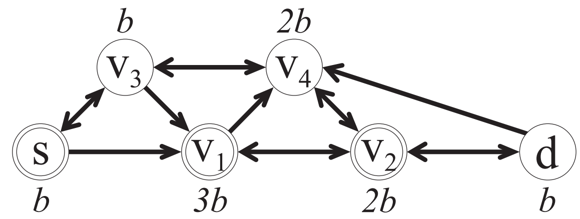


Fig. 2: An example of the Network Connectivity Graph Update algorithm for the path $p = (s, v_1, v_2, d)$ of b bit/s. The relay nodes (s, v_1, and v_2) are indicated with a double circle. Next to each vertex are the values of β. The value of β is increased by b for each relay node and for all the out-neighbors of each relay node.

V. MFCAR Algorithm

The SDNC maintains the network connectivity graph updated and uses it to compute the optimal V2V multi-hop path when requested. This path can be computed using a wide variety of well-known centralized routing algorithms, such as Dijkstra[13], Pollack[9], etc. We hereby present our centralized MFCAR algorithm, based on the uniform-cost search algorithm[14], with the objective function ω defined as:

$$\omega(p) = \alpha H(p) + (1 - \alpha) \left(1 - \prod_{v \in p} \gamma(v) \right) \tag{1}$$

Where $\alpha \in [0, 1]$ is the *congestion insensitivity*, and modifies the algorithm's behavior to prefer either short paths ($\alpha \to 1$) or uncongested paths ($\alpha \to 0$). $H(p)$ is the number of hops of the path p.

The algorithm selects the *source* vertex s as starting point and computes a new value of ω per each neighbor vertex. The algorithm selects the neighbor node with the minimum ω and repeats the process of expansion and selection until it finds the destination node or runs out of vertices to expand. The properties of the MFCAR algorithm are demonstrated in Appendix B. The demonstration of optimality is based on the non-decreasing properties of the addends of the ω function.

The rationale behind the design of the objective function ω is the following: Depending on the QoS specifications required by a flow, the SDNC should be able to select a path that matches those specifications, such as providing lower delay or higher throughput. Therefore, the objective function has been designed as a linear combination between H(p), which is correlated to packet delay, and a function of $\prod_{v \in p} \gamma(v)$, which is correlated to packet delivery ratio. These correlations exist because paths with more relay nodes induce higher packet delay, whereas relay nodes with low idleness score are overloaded and therefore more likely to lose frames due to collisions.

VI. Simulation scenarios

In our performance study, we compare two urban scenarios with Manhattan grids, whose geometrical characteristics are reported in Table I. In these scenarios, vehicles drive along

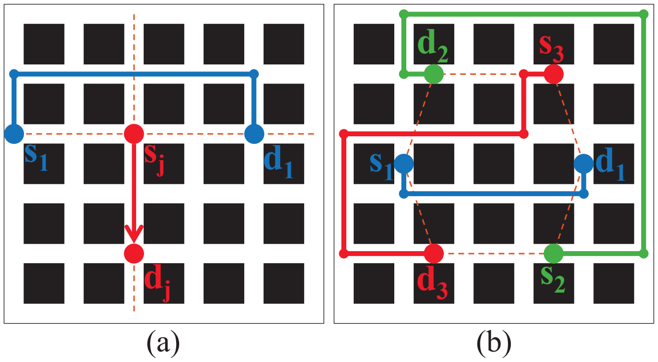


Fig. 3: Cross scenario (a) with one data source. Hexagon scenario (b) with three data sources.

TABLE I: Road Network and Mobility Parameters

Manhattan Grid Scenario	$1\,\mathrm{km} \times 1\,\mathrm{km}$, 6 roads $\times$ 6 roads
Inter-road distance	$200\,\mathrm{m}$
Road width	$6\,\mathrm{m}$ (Two $3\,\mathrm{m}$ lanes)
Road features	No traffic lights, opposite-direction lanes
Buildings size	$180\,\mathrm{m} \times 180\,\mathrm{m}$
Inter-building distance	$20\,\mathrm{m}$
Mobility traces generator	SUMO Netgen, Discretization=$1\,\mathrm{s}$
Mobility features	Random trips, minimum distance $500\,\mathrm{m}$

the shortest path determined between two randomly-picked points on the map, with a speed uniformly chosen between $30\,\mathrm{km/h}$ and $50\,\mathrm{km/h}$. Some vehicles play the role of source nodes, which request the SDNC to allocate a unicast V2V flow towards the target vehicle. All the simulation details about the flows are reported in Table II. The geographical positions of source and destination vehicles are fixed.

The vehicles will send the neighbor list updates to the SDNC every T_u =$100\,\mathrm{ms}$. The SDNC will check for new optimal routes for each active flow every T_c =$100\,\mathrm{ms}$. We compare our proposed MFCAR Algorithm with Dijkstra's Algorithm.

A. Cross Scenario

In this scenario there is one source vehicle s_1, requiring the allocation of a unicast flow to its respective destination vehicle d_1. There is also a *jammer* flow from s_j to d_j, which geometrically lies in between of source and destination vehicles. In Figure 3a we can see that the source and desintation nodes are disposed as to form a cross. This scenario is designed to test the behavior of MFCAR when a flow is disturbed by a source of congestion on its path.

B. Hexagon Scenario

In this scenario there are three source vehicles (s_1, s_2, s_3), each one sending a unicast flow to their respective destination vehicles (d_1, d_2, d_3). In Figure 3b we can see that the source and desintation nodes are disposed on the vertices of an hexagon. This scenario is designed to test the behavior of MFCAR when multiple spatially-distributed flows must be allocated at the same time.

TABLE II: Networking Parameters

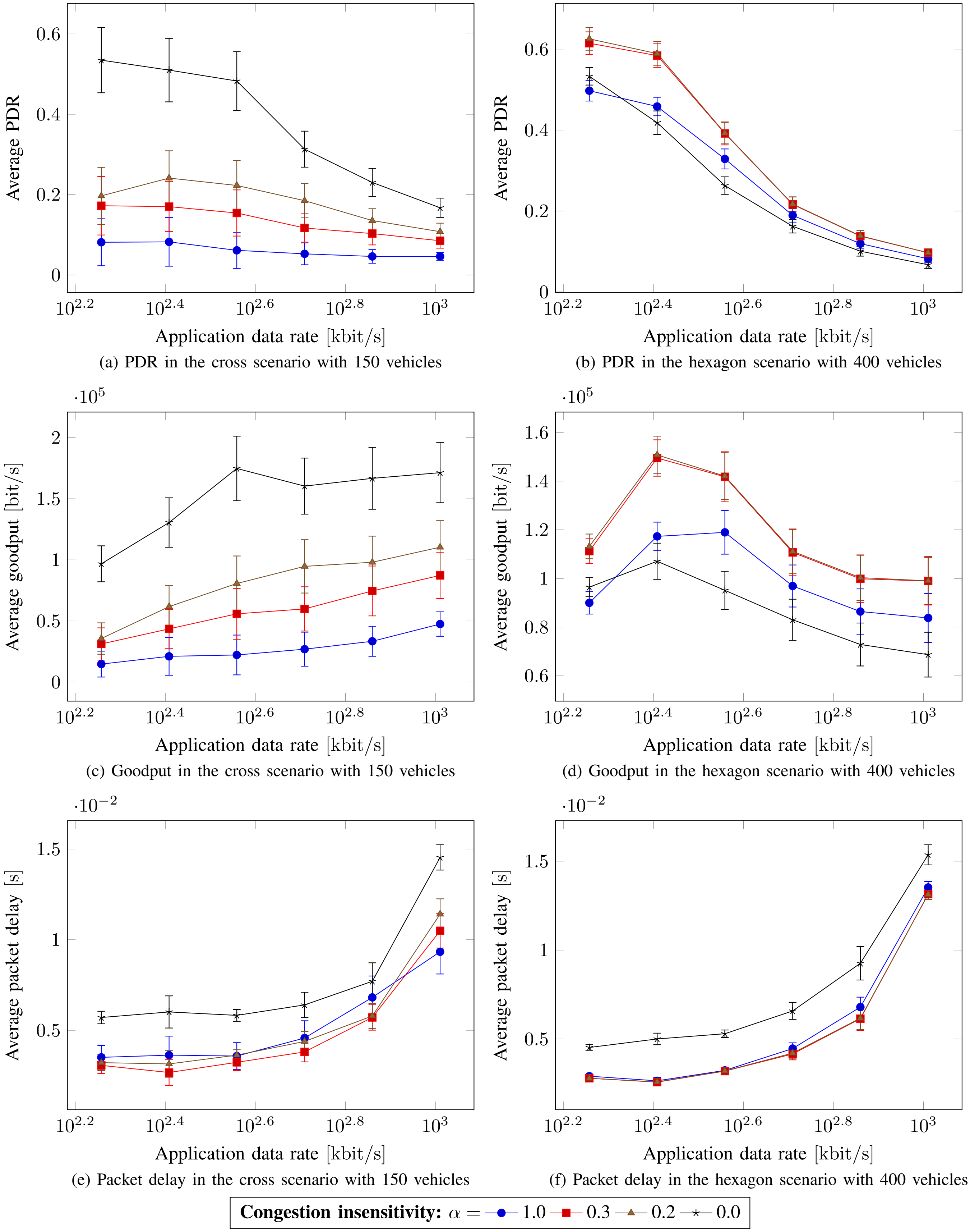
Application Layer	
Beaconing period and lifetime	$T_b = T_e = 100\,\mathrm{ms}$
Application Packet Size	$1024\,\mathrm{bit}$
Application Data Rates (b)	181, 256, 362, 512, 724, 1024 $\mathrm{kbit/s}$
IEEE 802.11p PHY and IEEE 1609.4 MAC Layer	
Transmission Power	$8\,\mathrm{mW}$
Receiver Sensitivity	$-89\,\mathrm{dBm}$
Transmission Capacity	$6\,\mathrm{Mbit/s}$
Signal attenuation model	Two-Ray Interf. Model (Veins[15])
Channels	CCH(178), SCH(174)
Channel Switching Interval	$50\,\mathrm{ms}$
MAC Queue Capacity	8 packets

VII. Performance Evaluation

The performance of the MFCAR algorithm has been evaluated for both cross and hexagon simulation scenarios, in terms of PDR, goodput, and average packet delay. In particular, the PDR is defined as the ratio between the total number of packets received by the flow destination and the total number of packets sent by the flow source. The goodput is defined as the total volume of packets received by the flow destination, divided by the flow duration in seconds. The average packet delay is defined as the average difference between the packet's timestamp at source and desination. In order to strenghten the statistical relevance of the results, the simulations are repeated 64 times, each of them with a different random seed.

In the moderate-density cross scenario (150 vehicles), MFCAR shows the highest performance difference between the optimal value of α and the others. We can observe an improvement of PDR (Figure 4a) and goodput (Figure 4c) as α decreases from 1 to 0. This is justifiable by the fact that Dijkstra's algorithm ($\alpha = 1$) does not consider the congestion status of the network when selecting the optimal route, and forwards the legitimate flow through the jammer flow. However, when α approaches zero, the selected routes must be longer in order to circumvent the jammer flow and therefore generate a higher average packet delay (Figure 4e). The MFCAR performances in the high-density cross scenario (400 vehicles) peak at $\alpha = 0.1$ and offer even higher PDR and goodput than in the moderate-density scenario.

In the hexagon scenario, the highest performance gain between our MFCAR algorithm and Dijkstra's is obtained for high vehicular density (400 vehicles). This is because with such high vehicular density, the network graph is extremely connected and the MFCAR algorithms has a wider range of possible paths from which to choose. This is the ideal scenario in which the MFCAR algorithm can apply its congestion-avoidance properties. We can observe that the optimal PDR (Figure 4b) and goodput (Figure 4d) curves are reached for intermediate values of α. This is because when $\alpha = 1$ the chosen routes overlap and generate congestion (as in the cross scenario). When $\alpha = 0$, the routes become unnecessarily long, increasing not just the average packet delay (Figure 4f), but also packet loss due to frame collision. Therefore, in this scenario, only intermediate values of α can offer paths with an optimal balance between length and congestion level.



(a) PDR in the cross scenario with 150 vehicles

(b) PDR in the hexagon scenario with 400 vehicles

(c) Goodput in the cross scenario with 150 vehicles

(d) Goodput in the hexagon scenario with 400 vehicles

(e) Packet delay in the cross scenario with 150 vehicles

(f) Packet delay in the hexagon scenario with 400 vehicles

Fig. 4: Performance evaluation of the MFCAR algorithm for two vehicular densities in cross and hexagon scenarios. For each scenario and metric, each sample takes the average of the considered metric over all the simulated data flows in the considered scenario.

VIII. Conclusion and future work

In this article, we have proposed and analyzed MFCAR, a SDN-based centralized routing algorithm for vehicular networks. We have shown that MFCAR dramatically outperforms classic centralized routing algorithms (Dijkstra) in terms of PDR, goodput and average delay for V2V multi-hop communications. The reason behind the better performance of our algorithm resides in its design: MFCAR performs a uniform-cost search, guided by a novel objective function that select paths that balance length and congestion level. The network congestion is computed in an innovative way, removing the need to perform CCA by approximating the expected level of congestion after flow allocation.

As future investigation path, it is relevant to compare the fully centralized algorithms analyzed in this article (Dijkstra, MFCAR), with the most studied distributed routing algorithms for mobile networks (e.g. AODV, GPSR, DSDV). This evaluation would lay the foundations of the design of a new hybrid routing algorithms that marshalls part of routing decisions to vehicles in particular cases. Another interesting investigation follow-up is the dynamic and automatic selection of the optimal value of the congestion insensitivity parameter α, according to the particular flow QoS requirements and the current vehicular scenario.

Appendix

A. Definition of time constant τ

Let us assume that the capacity of the physical interface is $C = 6\,\mathrm{Mbit/s}$. It is reasonable to define τ such that $\beta(v) = C \Rightarrow \gamma(v) = 0.98$. This happens only when $C = 4\tau$, therefore $\tau = C/4 = 1.5\,\mathrm{Mbit/s}$.

B. Soundness of the MFCAR Algorithm

Let $G = (V, E)$ be the network topology graph, where $v_i, v_j \in V$ are its vertices and $(v_i, v_j) \in E$ are its edges. Let us define the *out-neighbors set* of the vertex v_i as $N_{\text{out}}(v_i) = \{v_j \in V \,|\, \exists (v_i, v_j) \in E,\, v_i \in V\}$. Let us define the generic path $p_n = (v_0, ..., v_n)$ as a sequence of $n+1$ adjacent vertices $v_i \in V, i \in \{0, ..., |V|\}$, connected by n edges. Let us define $\mathcal{P}$ as the set of all the possible paths for G.

Proposition 1 (Optimality)**.** *The MFCAR algorithm always returns the optimal path $p^\star$ between two vertices s and d in a finite number of steps if a path between s and d exists.*

Proof. From the uniform-cost search theory, it is provable by contradiction that completeness implies optimality. For the MFCAR algorithm, it is enough to prove that the objective function ω is non-decreasing. This is equivalent to proving that $\forall p_n, p_{n+1} \in \mathcal{P}, v_{n+1} \in N_{\text{out}}(v_n) : \omega(p_n) \le \omega(p_{n+1})$.

Let us define $H(p_n) = n$ as the number of edges along the path p_n. Let us define $\gamma(v_i) \in (0, 1]$ as the *idleness score* of the vertex $v_i \in V$. Let us define $W(p_n) = \prod_{i=0}^{n} \gamma(v_i)$ as the product of the idleness scores of the vertices belonging to the path p_n.

We can then write that $\forall p_n, p_{n+1} \in \mathcal{P}, v_{n+1} \in N_{\text{out}}(v_n)$: $W(p_{n+1}) = W(p_n)\gamma(v_{n+1})$, $H(p_{n+1}) = H(p_n)+1$. Because $\gamma(v_i) \in (0, 1]$, it immediately follows that $W(p_{n+1}) \le W(p_n), 1-W(p_{n+1}) \ge 1-W(p_n), H(p_{n+1}) > H(p_n)$. After imposing $\alpha \in [0, 1]$, it follows that $\alpha H(p_n) \le \alpha H(p_{n+1})$ and $(1-\alpha)(1-W(p_n)) \le (1-\alpha)(1-W(p_{n+1}))$. Adding the last two inequalities side by side, we obtain: $\alpha H(p_n) + (1-\alpha)(1-W(p_n)) \le \alpha H(p_{n+1}) + (1-\alpha)(1-W(p_{n+1}))$. We define $\omega(p_n) = \alpha H(p_n) + (1-\alpha)(1-W(p_n))$ as the cost of the path p_n. By applying the definition of the ω function to the previous inequality, we can conclude that $\forall p_n, p_{n+1} \in \mathcal{P}, v_{n+1} \in N_{\text{out}}(v_n) : \omega(p_n) \le \omega(p_{n+1})$. □

Acknowledgment

This work was supported by the FNR CONTACT project, CORE/SWISS/15/IS/10487418.

References

[1] A. Mahmood, W. E. Zhang, and Q. Z. Sheng, "Software-defined heterogeneous vehicular networking: The architectural design and open challenges," *Future Internet*, vol. 11, no. 3, p. 70, 2019.

[2] H. Peng, , X. Shen, and G. Y. Li, "Vehicular communications: A network layer perspective," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1064–1078, Feb 2019.

[3] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally, "5G for vehicular communications," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 111–117, Jan 2018.

[4] I. Yaqoob, I. Ahmad, E. Ahmed, A. Gani, M. Imran, and N. Guizani, "Overcoming the key challenges to establishing vehicular communication: Is SDN the answer?" *IEEE Communications Magazine*, vol. 55, no. 7, pp. 128–134, July 2017.

[5] C. Lai, Y. Chang, H. Chao, M. S. Hossain, and A. Ghoneim, "A buffer-aware QoS streaming approach for SDN-enabled 5G vehicular networks," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 68–73, Aug 2017.

[6] M. Ploumidis, N. Pappas, and A. Traganitis, "Flow allocation for maximum throughput and bounded delay on multiple disjoint paths for random access wireless multihop networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 720–733, Jan 2017.

[7] K. L. Kushan Sudheera, M. Ma, and P. H. J. Chong, "Cooperative data routing and scheduling in software defined vehicular networks," in *2018 IEEE Vehicular Networking Conference (VNC)*, Dec 2018, pp. 1–8.

[8] C. Huang, M. Chiang, D. Dao, W. Su, S. Xu, and H. Zhou, "V2V data offloading for cellular network based on the software defined network (SDN) inside mobile edge computing (MEC) architecture," *IEEE Access*, vol. 6, pp. 17 741–17 755, 2018.

[9] M. Pollack, "The maximum capacity through a network," *Operations Research*, vol. 8, no. 5, pp. 733–736, 1960.

[10] S. Correia, A. Boukerche, and R. I. Meneguette, "An architecture for hierarchical software-defined vehicular networks," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 80–86, July 2017.

[11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008. [Online]. Available: http://doi.acm.org/10.1145/1355734.1355746

[12] F. Klingler, F. Dressler, and C. Sommer, "IEEE 802.11p unicast considered harmful," in *2015 IEEE Vehicular Networking Conference (VNC)*, Dec 2015, pp. 76–83.

[13] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959. [Online]. Available: http://dx.doi.org/10.1007/BF01386390

[14] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009.

[15] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, January 2011.