Load-Aware and Mobility-Aware Flow Rules Management in Software Defined Vehicular Access Networks

LEO MENDIBOURE[1], **(Student Member, IEEE), MOHAMED AYMEN CHALOUF**[2],
AND FRANCINE KRIEF[3]

[1]CNRS LaBRI Laboratory, University of Bordeaux, 33000 Bordeaux, France
[2]Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA) Laboratory, University of Rennes 1, 22300 Lannion, France
[3]CNRS LaBRI Laboratory, Bordeaux INP, University of Bordeaux, 33000 Bordeaux, France

Corresponding author: Leo Mendiboure (leo.mendiboure@labri.fr)

ABSTRACT To ensure high performance for vehicular access networks, Software Defined Networking (SDN) technology is an efficient solution offering programmability, flexibility and a centralized view of the network. Nevertheless, to guarantee an efficient mobility management, some improvements are needed. In particular, a limitation on the number of exchanges between SDN controller and network devices and a better flow tables' management are required. To achieve that, the ideas of flow rules' pre-deployment and mobility-aware management of flow rules' lifetime have been proposed. However, existing approaches have two major limitations: accuracy and adaptability. Indeed, network load (accuracy) and network devices' feedbacks (adaptability) are not considered, two factors impacting flow rules management. That is why, in this paper, we define a state-based approach using connection/disconnection information reported by network devices to improve adaptability. Moreover, we propose a mobility-aware and load-aware flow rules management to improve accuracy. Finally, we specify a new flow rules' pre-deployment policy based on flow tables occupancy rate, vehicles mobility and SDN control channel load. The evaluations carried out confirm the benefits of the approach, compared to existing solutions, both in terms of recovery delay, control overhead, hard timeout calculation and flow table occupancy.

INDEX TERMS Flow tables, load balancing, mobility, SDN, stateful data plane, vehicular networks.

I. INTRODUCTION

Vehicular communication networks attract the attention of researchers and industrials. In fact, these vehicular networks are an essential component for autonomous vehicles and Cooperative Intelligent Transport Services (C-ITS) [1]. With the emergence of the C-V2X (Cellular-Vehicle to Everything) communication paradigm, associated with the development of a new interface dedicated to vehicular communications (LTE-V2X PC5), cellular networks (LTE-V2X, 5G) could represent the future of Vehicular Access Networks [2]. Indeed, cellular networks guarantee a wide coverage, through an important deployment of Base Stations (BSs), and a sufficient Quality of Service (latency, bandwidth, packet loss) to support vehicular communications.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenzhou Tang.

Nevertheless, in addition to high Quality of Service requirements [3], vehicular communications also have significant constraints: high mobility, variable density, frequent connections/disconnections. Thus, to enable efficient vehicular services (road safety, traffic fluidity, entertainment, etc.), the integration of a new technology has been considered for Vehicular access networks: software-defined networking [4]. This technology proposes to separate the network control and data planes to improve network flexibility and programmability and to provide a centralized view of the state of the network. Different papers, such as [3], [5], have already demonstrated the benefits of this technology in terms of packet loss, latency and bandwidth for Vehicular Access Networks. Thus, Software Defined Vehicular Access Networks (SD-VAN) appear as an interesting solution to meet vehicular services requirements.

The SDN technology relies on two main principles. The first of these pillars is the definition of a new element,

centralized or distributed: the SDN controller. This SDN controller uses information fed back by network devices (topology, delays, available bandwidth, etc.) to perform the main network functions: routing, load balancing, etc. Thus, in the SDN architecture, the SDN controller corresponds to the network control plane. The second pillar is the definition of programmable routing tables (flow tables) at the network data plane level: routers, BSs, etc. These flow tables, managed by the SDN controller, contain the routing rules (flow rules) used to transmit data. To enable a fast access to these flow rules, a specialized type of high-speed memory is used: Ternary Content Addressable Memory (TCAM) [6]. This memory, guarantees extremely low delays but has an important purchase cost. Therefore, to limit capital expense, the size of the SDN flow tables is usually limited.

In vehicular networks, SDN controller's management and flow tables' management could be impacted by the mobility of the terminal devices. First of all, mobility could lead to an overload of the SDN control channel, inducing higher delays and a lower level of responsiveness. Indeed, as the SDN controller manages network functions, every time a user moves from one cell (BS) to another, to maintain communication continuity, a Packet In request is sent to the SDN controller through control channel. Therefore, the high mobility of a large number of connected vehicles could lead to a high load level for this control channel [7]. This is why, mobility should be efficiently managed to limit the number of exchanges between vehicles and SDN controller and to reduce control plane's latency. An associated problem is the management of the network devices' flow tables. These flow tables may also be overloaded [8]. Indeed, as these tables have a limited size, and as new flow rules are deployed by the SDN controller at each new connection, an efficient management and eviction of the rules deployed within flow tables is necessary.

Control channel overload and flow table overload could have the same consequence, a degraded Quality of Experience for users and degraded performances (latency, bandwidth, packet loss) [7]. Therefore, many research work have sought to address these problems. The basic idea behind these solutions is to consider user mobility prediction to determine which flow rules must be deployed [3], [8]–[12]. Thus, the SDN controller can proactively deploy SDN flow rules. This limits the number of Packet In requests sent by the network devices, eliminating the latency associated with reactive flow rules deployment and reducing the control overhead. Moreover, mobility prediction can be used to efficiently manage the lifetime (hard timeout) of the flow rules deployed within network devices. That is why this approach could be used to address the two targeted problems: control channel overload and flow tables occupancy.

Nevertheless, two important limitations can be identified for existing solutions: lack of accuracy and lack of adaptability. The lack of accuracy is related to the fact that user mobility prediction, used to determine inter-cell transitions, is only based on vehicle movement: speed, direction, etc. However, the number of vehicles and the load level of the BSs can also have an impact on these transitions. Indeed, the connection time of a vehicle to a given BS depends on the load level of this BS and the surrounding BSs [13], [14]. The lack of adaptability is related to the fact that flow rules updates are only based on temporal parameters. Depending on the expected mobility, the controller will pre-deploy flow rules with fixed timeouts. Any difference between the estimated transition time and the actual transition time would require an unnecessary intervention of the SDN controller, inducing latency and control overhead. Indeed, the flow rules deployed within network devices would not correspond to the actual state of the network. To address this problem, the connection/disconnection notifications fed back by the BSs could be used to manage inter-cell transitions.

Therefore, in this paper, we present an approach improving control channel management and flow tables management in terms of accuracy and adaptability. To do so, we introduce a solution based on an emerging idea in the SDN environment, a stateful data plane [15]–[17]. We use this stateful data plane to proactively manage user mobility and minimize the number of exchanges between SDN controllers and network devices. In addition, we propose an accurate estimation of the flow rules' hard timeout based on user mobility prediction and BSs' load level prediction. Finally, we define a new policy for flow rules pre-deployment taking into account different parameters: inter-cell transition probability, flow tables' occupancy ratio and SDN controller's load level. The evaluation of the proposed solution demonstrates its benefits compared to existing solutions both in terms of recovery delay, hard timeout calculation, flow table occupancy and control overhead. The contributions of this paper can be summarized to:

- the integration of the idea of Stateful Data Plane in software defined vehicular access networks;
- an optimisation of the flow tables' occupancy ratio taking into account the load level of the BSs and the mobility of the vehicles;
- a reduction of the control channel load using state machines and connection/disconnection information fed back by the BSs,;
- the definition of a flow rule pre-deployment policy taking into account the load level of the SDN controller, the flow table's occupancy ratio and the inter-cell transition probability;
- the implementation and the evaluation of the proposed approach compared to existing solutions.

The rest of this paper is organized as follows. Section 2 compares state-of-the-art solutions improving SDN control plane performances and flow tables management in SD-VAN. Then, Section 3 presents the proposed mechanisms: accurate estimation of the flow rules' hard timeout, definition of state machines enabling mobility management and proposal of a flow rules pre-deployment policy. Finally, in Section 5, the performance of our system is compared to existing solutions in terms of hard timeout

calculation, recovery delay, control overhead and flow table occupancy.

II. RELATED WORK

The issues of control channel management and flow tables management are not specific to vehicular networks. Indeed, whatever the environment considered, the size of the flow tables is limited [6] and the scalability of the SDN control plane must be considered to ensure reliability, availability and a high Quality of Experience [7].

Thus, outside the vehicular environment, many solutions have already been proposed to manage flow rules' hard timeout as efficiently as possible [18]–[21]. Some of these approaches use existing polices such as FIFO (First In First Out) or LRU (Least Recently Used) [18], [19]. Some others combine path selection and flow table management to further improve performances [20], [21]. In particular, these papers propose to select communication paths according to the available bandwidth and the occupancy rate of the switches used by each communication path.

Similarly, different solutions have been introduced to reduce SDN controllers' load level [7], [22]. Some of these approaches, such as [22], are based on an improved SDN control plane, decentralized or distributed. Some others are based on different optimizations of the exchanges between the SDN controllers and network devices [7].

However, in vehicular network, control channel management and flow tables management are impacted by the mobility of the terminal devices. Therefore, vehicles mobility must be considered to define efficient solutions optimizing control plane and data plane performances. As a result, many specific solutions have already been proposed for Vehicular Access Networks.

A first solution [3], [23], [24] is to pre-calculate the flow rules to deploy according to vehicles mobility prediction. With this approach, the response time of the SDN controller is reduced. Indeed, as the flow rules are pre-calculated, when the SDN controller receives a Packet In request from a network device (vehicle, BS), it simply deploys these flow rules, guaranteeing a shorter response time [23]. Moreover, vehicles mobility prediction can be used to estimate the hard timeout of the flow rules deployed by the SDN controller. Indeed, knowing the direction, trajectory and speed of a vehicle, as well as the coverage area of the BSs, it is possible to estimate the average time required by a vehicle to cross a given cell. Thus, this solution also improves flow rules management [3]. However, this approach does not reduce the number of requests sent by the network devices to the SDN controller. Thus, in extreme conditions, the SDN control channel could be overloaded and the SDN controller's response time could be high.

This is why many studies proposed to pre-calculate and pre-deploy flow rules [9], [11], [23], [25]–[27]. Different use cases have been considered: fair flow table entries' repartition between users [25], efficient occupancy ratio of flow tables [9], [23], [26], [27] and flow tables management

in uncovered areas [11]. Nevertheless, pre-deployment is always applied with the same objective: to limit the number of requests sent by the network devices to the SDN controller and to enable an instantaneous mobility management. Indeed, these exchanges between SDN controllers and network devices induce a communication delay and increase the control channel load. With these approaches [9], [11], [23], [25]–[27], flow rules are pre-deployed within network devices. Thus, when a vehicle connects to a new BS, the communication continuity can be ensured without any intervention of the SDN controller, reducing the number of exchanges. Moreover, delays related to flow rules deployment are eliminated as flow rules are pre-deployed. Therefore, by combining flow rules pre-calculation and pre-deployment, control overhead can be reduced and flow table management (hard timeout) can be optimized.

Nevertheless, existing approaches have two major limitations:

- accuracy: the BSs' load level is not considered in the flow rules' hard timeout calculation, resulting in an inaccurate estimation of this duration. Indeed, the handover mechanisms are based on an estimation of the BSs' load level. Thus, depending on this load, inter-cell transitions could be speeded up or slowed down [14], impacting the hard timeout value. That is why, to improve accuracy, and to enable a more effective flow tables' management, BSs' load level must be considered in hard timeout calculation;

- adaptability: transitions between flow rules, corresponding to the predicted transition of a vehicle from one BS to another, are only based on temporal parameters (flow rules' hard timeout), resulting in a non-adaptable system. Indeed, any difference between the estimated transition time (flow rule's hard timeout) and the actual transition time (vehicle disconnection/re-connection) would require an unnecessary intervention of the SDN controller, as rules deployed at the network devices level would not correspond to the actual state of the network. That is why, to limit the control overhead, another parameters must be considered in flow rules' transitions: the connection/disconnection information fed back by the BSs.

In this paper, we specify a solution that addresses both of these challenges: accuracy and adaptability. To achieve that, we propose a flow rules' hard timeout calculation based on vehicle mobility prediction and BSs' load level prediction. This ensures a more accurate estimation of the flow rules' hard timeout and a more efficient flow tables management. Moreover, to manage flow rules transitions, we define a mechanism considering the information fed back by the BSs (connections/disconnections). This approach, based on a stateful data plane, enables an efficient vehicles mobility management and a reduction in the number of exchanges between network devices and SDN controller. Finally, to optimize both flow tables' occupancy ratio and control channel use, we propose an innovative flow rules' pre-deployment
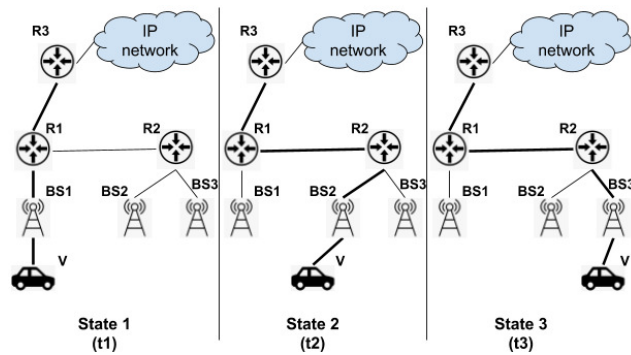
FIGURE 1. Moving vehicle example.

policy based on different parameters: inter-cells' transitions probability, flow tables' occupancy ratio and SDN control channels' load level.

III. PROPOSED LOAD-AWARE AND MOBILITY-AWARE FLOW RULES MANAGEMENT

In this section, we propose a new load-aware and mobility-aware flow rules management. To do so, we first introduce the objective of our solution (cf. section III-A) and the existing elements used to design our approach: a mobility model (cf. section III-B) and a cell load estimator (cf. section III-C). Then, we present the main components of our approach: a load-aware and mobility-aware flow rules' hard time-out calculation (cf. section III-D), a stateful mechanism ensuring adaptive inter-BSs transitions (cf. section III-E) and an innovative policy for flow rules pre-deployment (cf. section III-F).

A. GOAL

5G cellular networks, through Ultra-Reliable Low Latency Communications (URLLC), could be an efficient way to support vehicular applications (road safety, traffic efficiency, entertainment) [28]. To deal with high data rate applications and an increasing number of terminal devices, *small cells* will be deployed for 5G networks [29]. These small cells are characterized by a small radius and, thus, in 5G VAN, handovers (inter-cells transitions) will be frequent. That is why, for 5G SD-VAN, an efficient management of SDN resources (flow tables, control channel) will be required to guarantee high performances: low delays, efficient bandwidth management, etc.

In this 5G SD-VAN environment, considering mobile vehicles connected to Internet (cf. Fig. 1), using a multimedia, remote maintenance or traffic efficiency application, our goal is to:

- use flow tables as efficiently as possible. Each flow table has a maximum number of entries R. For each switch s, at any time t, the number of used entries X must be lower than R. Otherwise, the flow tables would be overloaded. Thus, we must ensure that, at any time t, $X_s < R_s$. This parameter is directly related to the mobility of the

vehicles and the load level of the base stations. Indeed, vehicle mobility and BSs load could lead to an increase in the occupancy level of the flow tables. Thus:

- in section III-D, we propose to define the flow rules lifetime HT as a function of the mobility of the vehicles and the load level of the base stations: HT = f(mobility, load). HT is used to estimate the number of inputs for any switch s at any time t. An accurate estimation of HT would minimize the difference between the actual number of rules deployed at time t and the estimated number of rules;
- in section III-E, we propose to take into account the occupancy rate of the flow tables in the selection of communication paths. Thus, if different paths meet user's requirements (latency, bandwidth, packet loss), the path limiting the occupancy rate (minimum number of entries to be deployed) will be selected;
- in section III-F, we propose to take into account the occupancy rate of the flow tables in the pre-deployment of the flow rules. By adapting the pre-deployment policy to actual network conditions, the number of rules deployed could be minimized;

- use control channel as efficiently as possible. An overload of the control channel could lead to a degradation of performances. Therefore, it is important to ensure that, at any time t, the actual load level of the control channel Lr is below its acceptable load level Lth (Lr < Lth). The control channel load level is directly related to vehicles mobility management and to flow rules pre-deployment. Thus:

- in section III-E, we propose to improve proactive vehicles mobility management. To do so, we introduce a mechanism enabling a dynamic and autonomous management of inter-cell transitions, limiting the number of Packet In requests transmitted by the network devices. This mechanism uses the connection/disconnection information fed back by the BSs;
- in sectionIII-F, we propose to take into account the transition probability and the channel load level in the pre-deployment of the flow rules. The proposed approach ensures that the number of flow rules deployed is at most equal to the number of rules needed in a reactive approach. Thus, this approach enables to optimize performance compared to a reactive approach while guaranteeing an adaptation to the real network conditions.

Considering a set of mobile users and a software-defined mobile access network, our objective is to limit the flow tables' occupancy rate and the control channel's load level while guaranteeing high performances for the users, especially in terms of latency (inter-cell transitions) and bandwidth (path selection).

B. MOBILITY MODEL

Knowing that a given vehicle is currently connected to a BS, BS1 (cf. Fig. 1), the mobility model is used to determine the next inter-cell transition of this vehicle (the next BS it will connect to). Indeed, thanks to this information, it will be possible for the SDN controller to determine at which BSs and routers, flow rules must be pre-deployed.

To compute these transitions between BSs, Hidden Markov Models (HMM) are an effective method commonly used in vehicular networks [30], [31]. With HMM, it is considered that for a given vehicle currently connected to a given BS (Fig 1: State 1), the number of next possible states (i.e. BSs this vehicle could connect to) depends on the observables (information fed back by the BSs). Thus, observables are used to determine HMM states. Moreover, using these observables, this model can also be used to determine the probability of transition from a state X to a state Y: the probability that a vehicle currently connected to a BS X subsequently connects to a BS Y.

Thus, with HMM, we can determine at which BS, corresponding to the possible next state with the highest probability, and which routers, flow rules must be pre-deployed. For example, in Fig. 1, assuming that communication is instantiated at State 1 (i.e. Packet In is sent by BS1), the SDN controller will be able to deploy flow rules at BS1, R1, R3 to manage communication during State 1's duration (t2-t1) but also to pre-deploy flow rules at BS2, R2, R1, R3 to manage communications during State 2's duration (t3-t2). If the inter-cell transition predicted for V corresponds to its actual inter-cell transition, transition from BS1 to BS2 will be managed without any Packet In request from BS2. Therefore, pre-deployment could limit the number of Packet In requests and improve inter-cell transition management.

It can be noted that other approaches could have been considered for mobility prediction [32], [33]: game theory, heuristic, etc. However, Markov models are an efficient method to predict users mobility while considering users' past locations [9]. In addition, this approach enables an accurate prediction of mobility, similar to other powerful approaches such as heuristics [32]. Finally, different existing works, such as [9], [23],are based on Markov models. In the section IV, the use of a same mobility model will enable us to demonstrate that the benefits of the proposed approach come from the considered evolution (state tables, load aware prediction, pre-deployment policy) and not from the mobility model. It can also be noted that the proposed evolution could also be used with another mobility model.

C. CELL LOAD ESTIMATION

If the load level of a BS is too high, communication performances (latency, bandwidth, packet loss) and, consequently, user Quality of Experience could be degraded. This is especially true in 5G cellular networks composed of small cells [29]. Thus, many studies have already proposed handover policies that should enable an efficient load balancing between small cells [13], [14], [29], [34].

Load balancing is primarily based on the definition of an acceptable load threshold for each BS. For example, this threshold can be defined using the Resource Block Utilization Ratio (RBUR). The RBUR is the ratio between the total number of Physical Resource Blocks (PRBs) existing at a cell site and the number of PRBs currently used by the vehicles connected to this BS. This threshold is therefore specific to each BS and must be calculated dynamically.

Then, this threshold, can be used for load balancing between different BSs [13], [14], [34]. For example, considering two BSs, BS1 and BS2, if the load level of BS1 is too high, handover policies may be used to transfer terminal devices (vehicles) from BS1 to BS2. The selection of the terminals devices which should be transferred from BS1 to BS2 is based on different parameters: the number of PRBs allocated to each user, the position of these users (a user can only be transferred if it is connected to both BSs) and the direction of these users (a user will only be transferred from BS1 to BS2 if it moves in the direction of BS2). It should be noted that the estimation of the number of PRBs allocated to each user can be carried out macroscopically (total number of PRBs / number of users) or microscopically (Signal-to-noise ratio for a given user).

Thus, the information reported by the vehicles (position, direction, signal-to-noise ratio) and by the BSs (total PRBs, allocated PRBs) can be used to efficiently balance the load between different small cells, improving network performances (latency, bandwidth, packet loss) and Quality of Experience. These pieces of information can also be used to improve the estimation of the flow rules' hard timeout as demonstrated in the following section.

D. FLOW RULES' HARD TIMEOUT CALCULATION

In mobile networks, the lifetime of flow rules is limited. When a vehicle disconnects from a BS, the corresponding flow rules are no longer necessary and should be removed from the flow table. The automatic deletion of the flow rules can be achieved using the Hard Timeout (HT), the maximum lifetime of a flow rule deployed within a flow table.

Indeed, if the value of the HT corresponds to the actual duration of the communication between a vehicle and a BS, the flow rule will be automatically deleted at the vehicle's disconnection. The value of the HT must be as close as possible to the actual duration of the connection between vehicles and BSs. If the HT value is too short, the BS will send a new Packet In request to the SDN controller. If the HT value is too long, an entry in the BS flow table will be unnecessarily occupied.

In existing solutions [9], [11], [23], [25]–[27], the calculation of this value is only based on vehicles mobility prediction, using the observables fed back from vehicles (position, direction, speed) and BSs (connections/disconnections). These observables are classified according to vehicle speed. A given speed interval (5km/10km/etc.) is associated with a set of observables and an average HT for a given speed is then calculated. At any time t, the current average speed

Vmean of a vehicle connected to a BS *X* can be associated to a speed interval and to an average connection duration *HTx*. *HTx* is thus a function of the mobility (speed) of the vehicles ($HTx \sim f(Vmean)$).

This calculation of *HTx* does not take into account the current load level of the BSs. As shown in Section III-C, handover policies are based on BSs' load level. Thus, the duration of the connections between BSs and vehicles are directly related to the BSs' load level.

This is why we propose to classify the observables according to vehicles speed intervals but also according to the load level of the base stations [35], [36]: low (<20%), medium or high (>70%). More specifically, we considered the load level of the base station to which the vehicle is currently connected and the load level of the base station to which the vehicle should subsequently be connected.

Thus, the estimated connection time between a vehicle *Vx* and a base station *X* becomes a function of the average speed *Vmean* of that vehicle, the load level *Lx* of this base station *X* and the load level *Ly* of the base station *Y* to which that vehicle should then connect:

$$HTx \sim f(Vmean, Lx, Ly)$$

This estimation of the connection duration between a vehicle and a base station can be carried out in real time, based on the information provided by the network devices and the classification previously carried out. It can also be carried out pro-actively. Indeed, at any time t, for a base station *X*, of average load per user Lx_u:

- mobility prediction can be used to determine how many vehicles *N* will be connected to BS *X* at *Tx*;
- mobility prediction can be used to determine the average speed *Vn* of the *N* vehicles connected to BS*X* at *Tx*;
- the total load level *Lx* of BS *X* at *Tx* can be estimated as $Lx = N*Lx\_u$. Note that, if the SDN controller has information concerning the resources used by each user, this total load could be calculated more accurately.

Therefore, the vehicle speed and load level estimated at any time t can be compared with the previously classified data and HT can be calculated.

As it will be demonstrated in Section IV, our solution, taking into account the BSs' load level, enables a more accurate estimation of the HT for the flow rules and greatly improves flow tables' management.

E. STATE MACHINES IMPLEMENTATION

To enable the establishment of a connection between the vehicle and a Cloud service (multimedia, traffic management, etc.), flow rules must be deployed within the BSs flow tables but also within the routers flow tables (cf. Fig. 1: R1, R2, R3).

These flow rules, to ensure data transmission, must correspond to the actual state of the network. For example, in Fig. 1, in State 1, the flow rule deployed at R1 must enable data transmission to BS1. On the contrary, in State 2, the flow rule deployed at R1 must enable data transmission to R2.

The transition between states (inter-cell transitions) are based so far on the use of the HT (cf. section III-D). For example, in Fig. 1, at the end of the HT estimated for State 1, the flow rule within R1 is replaced by a new flow rule: packets are now transmitted to R2 and not BS1.

Nevertheless, as explained in Section III-D, the calculation of the HT is not accurate. Its value does not exactly correspond to the moment of the transition from State 1 to State 2. Therefore, if the replacement of the flow rules in R1's flow table is only based on this HT, this replacement could occur before or after the transition from State 1 to State 2. In these two cases, a Packet In request would be transmitted by the BSs to the SDN controller, increasing the control overhead and inducing latency.

This is why we propose to consider two parameters to manage these inter-states transitions: the value of the HT and the connection/disconnection information fed back by the BSs. Indeed, as demonstrated in [15]–[17], using a stateful data plane in SDN networks, inter-states transitions could be realized using information fed back from switches, in particular, link rupture notifications. Therefore, we applied this idea in SD-VAN to enable an efficient mobility management.

In the considered environment, a state corresponds to the communication between a vehicle and a given BS. For example, in Fig. 1, the communications between vehicle V and BS1, BS2, BS3 correspond to three different states. Assuming that at t1 BS1 sends a new Packet In request to the SDN controller, and that the SDN controller aims to manage V's mobility for a period of time t corresponding to three states (State 1, State 2, State 3), R1's flow table would be updated as follows:

1) a flow rule is deployed within R1's flow table to handle the current communication between V and BS1, with a HT approximately equal to t2-t1;
2) if the pre-deployment policy authorizes it (cf. section III-F), the expected next transition for V is computed (State 2);
3) as R1 is involved in State 1 and State 2, a new state is pre-deployed within the R1's flow table: when the link between V and BS1 will be broken, R1 will transmit data to R2 for a period of time approximately equal to t3-t2;
4) if the pre-deployment policy authorizes it (cf. section III-F), the expected next transition for V is predicted (State 3);
5) for R1, State 2 and State 3 are equal: data is transmitted to R2. Thus, the HT of the flow rule enabling to transmit data during State 2 is lengthened: data can now be transmitted to R2 during State 2 and State 3.

With this solution, mobility would be automatically managed within R1's flow table and the number of Packet In requests sent by the BSs would be reduced.

As shown in Algorithm 1 (Step 3), the proposed approach can be extended to *S* states and *X* network devices. For any State *n* (*n>1*), the SDN controller checks if the network

---

Algorithm 1: Proposed Flow Rules Management

Input: New Packet In from a vehicle V connected to a BS X
Output: Flow Rules to deploy
Variables: dest_BS=X, state_n=1, state_n_start=0, stat_n_HT=0;
while (true);
Step 1: retrieve communication information;
-> *1.1* get current SDN control channel load: C_load;
-> *1.2* compute shortest path between dest_BS and IP_Network: P_n;
-> *1.3* get the corresponding list of network devices: L_n;
Step 2: Get state information;
-> *2.1* compute state starting time state_n_start;
if state_n = 1;
    -> state_n_start = current_time;
else;
    -> state_n_start=state_n-1_start+state_n-1_HT;
-> *2.2* calculation of the cost associated with this pre-deployment: state_n_P;
-> *2.3* compute state duration based on BSs load and mobility prediction: state_n_HT;
Step 3: Compute the flow rules corresponding to this state;
for each *device* in L_n;
    -> retrieve the rule to deploy within *device* flow table: F_d_n;
    -> retrieve occupancy ratio of *device* flow table: Oc_d_n;
    if (state_n>1 **AND** *device* in L_n−1);
        if (F_d_n=F_d_n-1);
            -> update F_d_n-1 with: HT=stat_n-1_HT+stat_n_HT;
        else;
            -> set F_d_n as new state in *device* state machine;
Step 4: Manage flow rules deployment as efficiently as possible;
for each *device* in L_n;
    if (state_n=1);
        -> deploy F_d_n with HT=state_n_HT;
    else
        if (C_load="low");
            if (Oc_d_n="low");
                -> pre-deploy F_d_n with: HT=state_n_HT;
            if (Oc_d_n="high");
                -> pre-deploy F_d_n with: HT=max_diff_between_estimated_and_actual_duration;
        if (C_load="high");
            if (Oc_d_n="low");
                if (state_n_P=="minimize_rules_number");
                    -> pre-deploy F_d_n with: HT=state_n_HT;
Step 5: Determine if next state should be considered;
for each *device* in L_n;
    if (Oc_d_n="high" **AND** state_n+1_P!="minimise_rules_number" **AND** C_load="high");
        -> **return** False ;

---

devices involved in State *n* were also involved in State *n−1*. This information is used to manage flow tables updates and state tables updates. If a network device is involved in State *n* and State *n−1*, a new state is added within its state table (if State *n* and State *n−1* are different for this device) or the HT of the *n−1* flow rule is lengthened (if State *n* and State *n−1* are equal for this device). Otherwise, a new flow rule is defined.

Therefore, state machines can be used to efficiently manage flow rules pre-deployment. This approach enables efficient inter-BSs transitions, limiting the number of Packet In requests, transition delays and control overhead. In addition, thanks to the accurate estimation of the flow rules' HT (Algorithm 1, Step 2), the lifetime of the flow rules can be adapted to the actual network conditions, ensuring a more efficient use of the flow tables' capacity. However, under

extreme conditions, network devices' flow table or the SDN control channel may be overloaded (cf. section II). This is why in the following section we propose an efficient flow rules pre-deployment.

F. EFFICIENT FLOW RULES PRE-DEPLOYMENT

Using BSs' load level prediction and user mobility prediction, network devices' flow tables could be managed more efficiently (cf. section III-D). Moreover, using state tables, flow rules could be efficiently pre-deployed, reducing control overhead and latency (cf. section III-E).

However, pre-deploying a large number of successive states might be irrelevant. For example, in Fig. 1, if the probability that a vehicle currently connected to BS1 then connects to BS2 and then to BS3 is equal to 10%, pre-deploying a rule corresponding to State 3 at t1 could be irrelevant. Indeed, 9 times out of 10, an entry will be unnecessarily occupied in BS3's flow table and a new Packet In request will be sent to the SDN controller. Thus, resources (control channel, flow table) could be used uselessly. If the load level of the control channel or the occupancy ratio of the BS3 flow table were high, this approach could therefore lead to unsafe use of resources. This is why we propose an efficient pre-deployment policy.

This pre-deployment policy (cf. Algorithm 1, Step 4, 5) is based on different parameters: control channel load, flow table occupancy ratio and inter-cell transition probability. The different cases considered are as follows:

- if control channel load and flow table occupancy are low, resources can then be considered as unlimited, the deployment of many successive states can be envisaged;
- if control channel load is low and flow table occupancy is high, a pre-deployment with a shorter HT could be considered. For a given BS, the HT duration could be equal to the maximum measured difference, at this BS, between an estimated connection duration and an actual connection duration (cf. section IV-B1). If the actual transition corresponds to the supposed transition, the SDN controller will be able to extend the HT duration. Otherwise, the corresponding flow rule will be automatically deleted after a shorter duration than the theoretical connection duration. Thus, this approach ensures efficient transitions (when the actual mobility corresponds to the supposed mobility) and reduces the occupancy rate of the flow tables (when the actual mobility does not correspond to the supposed mobility);
- if control channel load is high and flow table occupancy is low, flow rules are pre-deployed only if the total number of flow rules deployed to manage the mobility of a vehicle is minimized thanks to this pre-deployment. Thus, considering that the probability of a transition is equal to *Ptr*, that the pre-deployment of *Npre* rules is required to manage this transition, that at most *Neffect* rules will have to be deployed by the SDN controller

if this transition does not occur, and that the average number of rules to be deployed in response to a *Packet In* is equal to *Mpacket*, rules are pre-deployed only if:

$$Npre + (1 - Ptr)(Neffect + 1) < Mpacket + 1$$

This approach guarantees that the number of messages transiting through the control channel is at most equal to the number of messages transiting in a reactive approach (*Mpacket* + 1). Note that the SDN controller can retrieve the textit*Ptr*, *Npre*, *Neffect*, *Mpacket*, using past observables (*Ptr*, *Neffect*, *Mpacket*) or communication path calculation (*Npre*);

- if control channel load and flow table occupancy are high, a best effort policy is used: flow tables and control channel are only used to manage necessary flow rules. A reactive flow rules deployment is adopted. Flow rules are pre-calculated but are only deployed as a response to a Packet In request, as described in [3]. Therefore, this approach leads to performance degradation (latency) but limits both control channel use and flow tables occupancy.

This pre-deployment policy ensures a better use of available resources. Indeed, with this policy, pre-deployment is adapted to the actual conditions of the network (high control channel load, high flow table occupancy). In the case of a high control channel load, the policy ensures that the number of rules deployed is at most equal to the number of rules needed in a reactive approach. Similarly, in the case of a high flow table occupancy rate, the proposed policy ensures that the flow rules hard timeout corresponds to the minimum necessary duration (transition delays). Finally, in the case of a high flow table occupancy rate and a high control channel load, the proposed policy behaves as the reactive approach. Thus, for flow tables and control channel management, the proposed policy can be seen as an optimal solution compared to reactive approaches.

We can notice that the time complexity of Algorithm 1, combining the different components introduced in section III, is $O(n_s^2 + E + V log V + n_d)$. n_s corresponds to the number of hidden states and $O(n_s^2)$ to the complexity of the decision process of the HMM mobility prediction model. E is the number of edges, V is the number of vertices and $O(+E + V log V)$ is the complexity of the algorithm used to compute the shortest path (Dijkstra's algorithm with Fibonacci heap). n_d is the number of devices in the selected path and $o(n_d)$ is the complexity of the flow rule deployment process.

In addition, it should be noted that the pre-deployment of flow rules can be based on two mechanisms. Flow rules can be pre-deployed in the cache of the network devices, if these devices are equipped with a flow table cache [8]. Flow rules can be deployed after a pre-determined period of time, lower than the HT of the currently deployed flow rules [3]. Whatever the mechanism employed, these flow rules should theoretically be available in the flow table to automatically manage inter-cell transitions.

It can also be noted that 70% can be considered as a high level of occupation of the flow tables, requiring the definition of specific policies aiming to avoid flow tables' overload issues [37].

Finally, it can be noted that, the prediction of vehicle mobility and the pre-deployment of flow rules considered in our approach could lead to a significant workload for the SDN controller. However, many studies are currently focusing on the distribution of the SDN controller [38]. With this architecture, each SDN controller would only have to manage a limited geographical area and a limited number of terminal devices. Thus, the load of each SDN controller could remain acceptable and the pre-deployment of flow rules could improve performance in terms of bandwidth usage and latency. Different research papers, such as [9], [23], have sought to show that the pre-deployment of flow rules can be envisaged in a large-scale mobile network.

IV. EVALUATION

In this section, we demonstrate the benefits of the proposed mechanisms compared to existing solutions (cf. section II):

- an accurate estimation of the flow rules' hard timeout to improve flow tables' occupancy ratio (cf. section IV-B1);
- a Stateful Data Plane to efficiently manage inter-cell transitions (cf. section IV-B2);
- a flow rules' pre-deployment policy to improve both flow tables and control channel management (cf. section IV-B3).

A. EXPERIMENTAL SETUP

For this evaluation, we used different tools:

- Mininet-Wifi: a complete framework enabling to emulate software-defined wireless networks [39];
- SUMO (Simulation of Urban MObility): a road traffic simulator enabling to simulate vehicles' mobility in a given geographical area [40];
- Ryu: a SDN Framework, written in Python, commonly used in research studies [41];
- iPerf: a complete tool enabling to generate traffic and to measure the maximum achievable bandwidth on IP networks [42].

The implementation of the proposed solution as well as the tools/scripts/settings used for this experiment are accessible in a Github repository.[1]

The simulation parameters considered during the experiment are presented in Table 1.

To simulate an environment close to a 5G cellular network, we considered the deployment of small cells, uniformly distributed in a simulation area of 600m by 600m (grid-based distribution), with a coverage radius of 100m and sites located 200m apart [45]. The Least Loaded First mechanism, available in Mininet-Wifi, was used for load balancing between these BSs [44].

[1]Github repository: github.com/lmendiboure/LAMAP/

TABLE 1. Simulation parameters.

Parameter	Value
Simulation Area	600m*600m
Number of BSs	9
BSs Distribution	Grid-based
BS Coverage Radius	100m
Vehicles Number	18-36
Vehicles Speed	8-20m/s
Mobility Model	Manhattan Mobility Model [43]
Mobility Prediction Accuracy	70-80%
Load Distribution	Uniform/Gaussian
Load Balancing Algorithm	Least Loaded First [44]
Number of Concurrent Flows	18-36/63-126
Simulation Duration	3600s
Total Number of Flows	~7000/~25000
Flow Table Size	150 entries

In the defined simulation area a variable number of vehicles were considered (18-36), travelling at variable speed (8-20m/s). The mobility model that we used for this simulation is one of the models usually used in urban areas: the Manhattan Mobility Model [43].

In this environment we compared the proposed approach, Load-Aware and Mobility-Aware with Pre-deployment (LAMAP) (cf. section III), to other existing solutions:

- a Mobility-Aware (MA) solution [3]: in this approach a Markov Model is used to predict mobility and to pre-calculate SDN flow rules. This approach ensures a quicker response from the SDN controller as necessary flow rules are pre-calculated. However, the idea of pre-deployment is not considered;
- a Distributed Mobility-Aware (DMA) solution [46]: in this approach a Markov Model is used to predict mobility and to pre-calculate SDN flow rules. In addition, the load is distributed among different SDN controllers (2 during the experiment). This could further reduce the response time of the SDN controller. However, the idea of pre-deployment is still not considered;
- a Mobility-Aware with Pre-deployment (MAP) solution [9]: in this approach an order-k Markov predictor is used to predict vehicles mobility and to pre-calculate and pre-deploy SDN flow rules. However, in this approach, BSs' load is not considered in flow rules HT calculation and state machines are not used.

For each solution, the same accuracy rate was considered for mobility prediction: 70 to 80% depending on the BSs. This value corresponds to measured performance for Markov Models [9]. Defining the same value ensures that the measured benefits are related to the proposed solution (LAMAP, MA, DMA, MAP) and not to the underlying mobility model.

Moreover two cases were considered to evaluate the performance of the proposed solutions:

- Uniform Load (UL) case: the load is uniformly distributed among the different BSs. A same number of vehicles (2-4) is connected to each BS. Moreover, using Iperf, each vehicle generates a unique data flow;
- Non-uniform Load (NL) case: following a Gaussian distribution, the load is non-uniformly distributed among
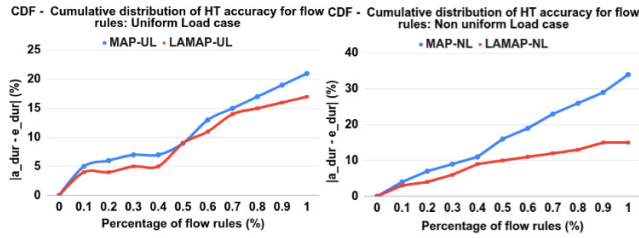
FIGURE 2. Comparison of flow rules HT calculation mechanisms.

the different BSs. A same number of vehicles (2-4) is connected to each Base Station. However, using Iperf, each vehicle generates a variable number of data streams (1-5).

For each of the considered cases (UL, NL), and each of the compared solutions (LAMAP, MA, DMA, MAP), a simulation of one hour was performed.

B. RESULTS

In this section are presented the results of the experiments we conducted: flow rules HT calculation (cf. section IV-B1), inter-cell transitions (cf. section IV-B2) and flow rules pre-deployment policy (cf. section IV-B3).

1) FLOW RULES HARD TIMEOUT CALCULATION

First of all, we compared the proposed flow rules HT calculation (cf. section III-D) mechanism, combining prediction of vehicles mobility and estimation of BSs load level, with existing solutions (MA, DMA, MAP), only based on prediction of vehicles mobility. As the same approach is used by the MA, DMA, MAP solutions, we compared our solution (LAMAP) with the MAP solution, in the uniform load case (LAMAP-UL, MAP-UL) and the non-uniform load case (LAMAP-NL, MAP-NL).

Flow rules HT is used to automatically eliminate unnecessary flow rules from flow tables to prevent flow tables' overload. The smaller the difference between the value of the calculated HT (e_dur) and the actual duration (a_dur) of the communication between the vehicle and the BS, the more efficiently flow tables are used.

Therefore, for all the flow rules deployed during experiments and all the solutions (LAMAP-UL, MAP-UL, LAMAP-NL, MAP-NL), we calculated the percentage of difference between these two values:

$$100*(e\_dur-a\_dur)/a\_dur$$

As it can be seen in Figure 2, representing the Cumulative Distribution Function (CDF) of the flow rules for HT accuracy:

- in the UL case, the performance of the two solutions is similar. Indeed, for our approach and the MAP approach, the average percentage of difference between e_dur and a_dur is the same: 8%. In addition, for about 50% of the flow rules, the MAP approach is more efficient (about 5-25% more accurate) and for about 50% of the flows the LAMAP approach is more efficient (about 10-33% more

accurate). Thus, in the UL case, the proposed approach reduces the maximum percentage of difference between e_dur and a_dur (of 33%) but does not improve the average performance of the system;

- in the NL case, the proposed approach greatly improves the estimation of HT. Indeed, on average, the LAMAP approach guarantees 22% higher performances than the MAP approach. Moreover, in the worst case, our approach improves performances by more than 100% (15% vs 34%). Therefore in the NL case, the proposed approach could guarantee a more reliable calculation of the HT value and a more efficient management of the flow tables.

In the UL case, load balancing is directly related to vehicles mobility. This explains the minor difference between the performance of the MAP approach and our approach. Nevertheless, as 5G Vehicular Access Networks will be based on small cells and as vehicles are mobile, this case seems unrealistic. Thus, in the vehicular environment, the proposed approach could enable an improved flow tables' management.

2) INTER-CELL TRANSITIONS

As a second step, we compared the proposed mechanism for inter-cell transitions (LAMAP), based on state machines, with existing approaches based on pre-calculation (MA, DMA) or pre-calculation and pre-deployment (MAP) in UL (LAMAP-UL, MAP-UL, MA-UL, DMA-UL) and NL (LAMAP-NL, MAP-NL, MA-NL, DMA-NL) cases.

An efficient inter-cell transition mechanism must minimise the duration required to manage vehicle transition from a BS to another (recovery delay) and limit the number of exchanges between SDN controller and network equipment.

Thus, we measured:

- the recovery delay: the time elapsed between the last packet (l_pckt) sent by a vehicle at the BS to which it was previously connected to and the first packet sent (n_pckt) by this vehicle at the BS to which it is currently connected. The value of the recovery delay is therefore equal to the difference between the (n_pckt) and (l_pckt) timestamps;
- the control overhead: the number of Packet In requests sent by the network devices to the SDN controller to manage a given flow. A flow corresponds to a connection between a vehicle and a BS. For each flow, the control overhead corresponds to the sum of Packet In requests sent by the BS to the SDN controller to manage the vehicle communication.

As it can be seen in Figure 3:

- the proposed solution (LAMAP) guarantees shorter recovery delays than the other approaches (MA, DMA, MAP) in any given case (UL, NL): more than 10% (UL)/14% NL on average compared to the MAP approach and more than 90% (UL)/96% (NL) compared to the MA approach. Our solution also reduces the number of Packet In requests sent by the network devices
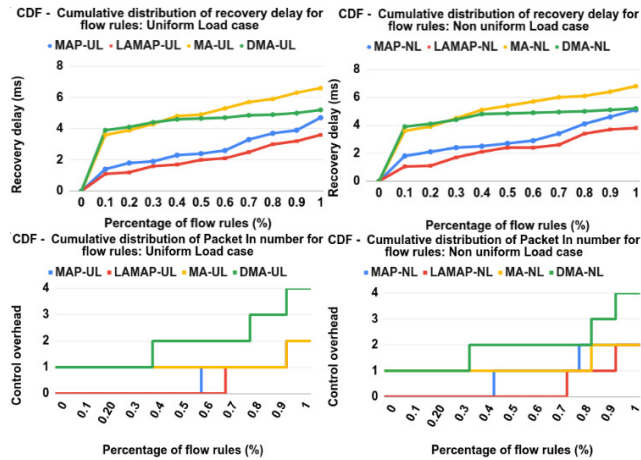
FIGURE 3. Comparison of inter cells transitions mechanisms.

to the SDN controller. For example, compared to the MAP approach, also reducing the number of exchanges, the gains, using our solution, are about 5% in the UL case and more than 30% in the NL case;

- The DMA solution, using multiple SDN controllers, guarantees, on average, lower recovery delays than the MA solution. However, this approach implies a control overhead higher than the MA solution. Indeed, Packet In requests are exchanged between the SDN controllers to reduce recovery delays. Thus, this solution is not the more relevant in our use case. Indeed, the number of Packet In is doubled while recovery delays are only 5-8% lower compared to the MA approach.

Our solution offers significant benefits in terms of inter-cell transitions management. Indeed, the number of Packet In requests sent by the network devices is reduced and the recovery delay is shortened, whatever the case considered (UL, NL). Thus, our approach improves Quality of Service performances both in terms of bandwidth use and latency.

These benefits are due to:

- a more accurate estimation of the flow rules HT: if the calculated HT is too short, a new Packet In request will be sent to the SDN controller (cf. section III-E). Therefore, several Packet In may often be sent to the SDN controller to manage a same flow. With our approach, as the HT calculation is more accurate, the number of useless Packet In requests is reduced;
- state-based transitions: inter-cell transitions only based on HT (MAP) may lead to delays due to inaccuracies in the HT calculation. Similarly, Packet In requests sent to the controller necessarily leads to delays. Thus, the proposed solution, using the connection/disconnection information fed back by BSs to manage inter-cell transitions, guarantees lower recovery delay.

It should be noted that, to compare these approaches, a same accuracy rate was considered for mobility prediction: 70% to 80% depending on the BSs (cf. section IV-A). If a prediction model was able to guarantee a 100% accuracy

for mobility prediction, the measured differences between our solution and existing solutions could be more important for recovery delay. Indeed, as shown in [17], state-based approaches could reduce very significantly delays compared to OpenFlow-based approaches (0ms vs 5-10ms). In addition, the number of Packet In requests would be further reduced compared to reactive approaches.

3) FLOW RULES PRE-DEPLOYMENT POLICY

As a final step, we assessed the benefits of the proposed pre-deployment flow rule policy. To do so, we compared the performances of our approach with this Policy (P-LAMAP) and Without it (W-LAMAP) with the performances of the MAP approach.

The pre-deployment policy is useful in three cases:

- Overloaded control channel (Oc): when the BSs flow tables' occupancy ratio is low and the control channel load level is high;
- Overloaded flow tables (Of): when the flow table occupancy ratio is high and the control channel load level is low. It should be noted that, to simulate a high flow table occupancy ratio, we considered flow tables with a low number of entries (150) pre-filled up to 70% (105 entries occupied by unused flow rules). Although in a real environment the number of entries may be higher (1500, 2500, 5000, 10000 entries), the same problem of flow table overload could occur as the number of terminal devices would be higher;
- Overloaded flow tables an control channel (Ofc): when the BSs flow tables' occupancy is high and the control channel load level is high.

Thus, considering a non-uniform load distribution, we measured the performance of both approaches (P-LAMAP, W-LAMAP, MAP) in these three cases (P-LAMAP-Oc, P-LAMAP-Of, P-LAMAP-Ofc, W-LAMAP-Oc, W-LAMAP-Of, W-LAMAP-Ofc, MAP-Oc, MAP-Of, MAP-Ofc) in terms of:

- average flow table occupancy ratio: average percentage of difference between the total number of entries available in a flow table and the number of entries used, at any moment, by the deployed flow rules;
- recovery delay (cf. section IV-B2);
- control overhead (cf. section IV-B2).

As it can be seen in Figure 4, in the Oc case, the proposed pre-deployment policy offers the expected benefits. Indeed, the number of Packet In requests per flow is reduced with the P-LAMAP-Oc approach compared to the W-LAMAP-Oc and MAP-Oc approaches, improving control channel use. In addition, the P-LAMAP-Oc approach offers performances similar to the MAP-Oc approach in terms of flow table occupancy and higher performances than the W-LAMAP-Oc approach. Finally, the impact of the pre-deployment policy on recovery delay is below 5% on average compared to the W-LAMAP-Oc approach and below 2% on average compared to the W-LAMAP-Oc approach.
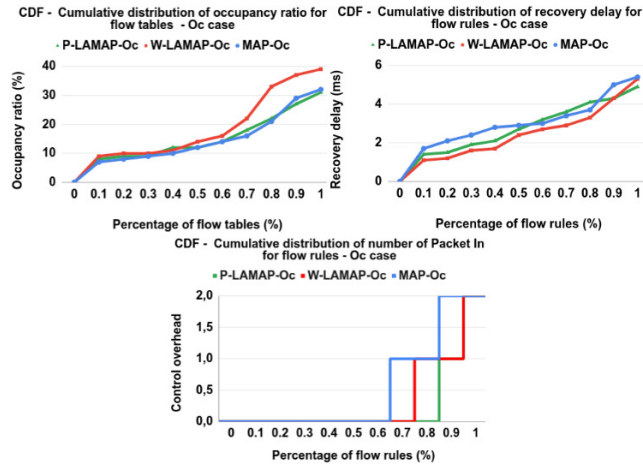
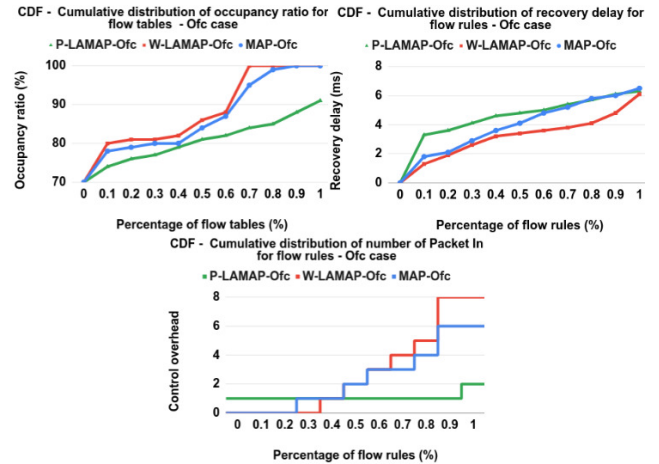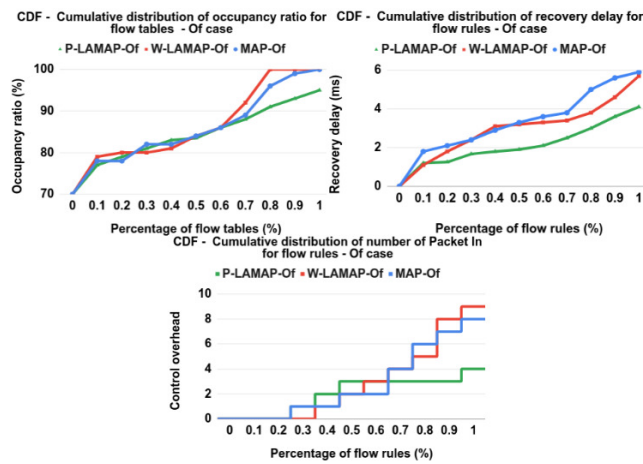**FIGURE 4.** Evaluation of the pre-deployment policy: Oc case.

FIGURE 5. Evaluation of the pre-deployment policy: Of case.

As it can be seen in Figure 5, in the Of case, the proposed pre-deployment policy offers the expected benefits. Although the MAP-Of approach is more efficient than the W-LAMAP-Of approach (10% more efficient), the P-LAMAP-Of approach ensures that none of the switches are overloaded. Thus, it improves flow tables use. The pre-deployment policy also enables to significantly reduce the number of Packet In requests sent per flow and the average recovery delay. Indeed, with the W-LAMAP-Of and MAP-Of approaches, as some flow tables are overloaded (100% full), the number of Packet In requests explodes, increasing delays and control overhead.

As it can be seen in Figure 6, in the Ofc case, the proposed pre-deployment policy offers the expected benefits. The flow tables' occupancy ratio is reduced by 20% on average compared to the W-LAMAP-Ofc approach and by 14% on average compared to the MAP-Ofc approach. Similarly, in the worst case scenario, the number of Packet In requests sent by network devices is divided by 3 compared to the MAP-Ofc approach and by 4 compared to the W-LAMAP-Ofc approach. Thus, both control channel and flow table use are improved. Nevertheless, it should be

FIGURE 6. Evaluation of the pre-deployment policy: Ofc case.

noted that the impact of the pre-deployment policy on recovery delay is high (30% compared to W-LAMAP-Ofc, 15% compared to MAP-Ofc). Indeed, with the P-LAMAP-Ofc approach, to limit flow tables and control channel use, flow rules are reactively deployed, increasing recovery delays.

Thus, the proposed pre-deployment policy enables us to significantly improve the performance of our solution both in the Of, Oc and Ofc cases. Moreover, the P-LAMAP approach provides nearly equivalent (Oc: recovery delay) or even higher (Oc: flow tables' occupancy, Of: recovery delay, control overhead) performance for the other key performance indicators compared to the W-LAMAP approach.

The MAP approach aims to optimally distribute the flow rules and thus offers higher performances than our W-LAMAP approach in terms of flow table occupancy (Of, Ofc, Oc) and control overhead (Ofc). However, our P-LAMAP approach, is designed to reflect the actual state of the network. It therefore guarantees higher performances than the MAP approach both in terms of flow tables occupancy and control overhead in the Of, Oc and Ofc cases.

Therefore, this mechanism is an efficient way to improve flow table management and exchanges between SDN controller and network equipment. This pre-deployment policy could be useful both for our solution, for the MAP solution and for the other approaches proposed so far.

V. CONCLUSION

Software-defined networking could be an efficient way to manage communications in Vehicular Access Networks. Indeed, the SDN technology could improve both mobility management and interoperability thanks to programmability, flexibility and centralization. However, to enable the integration of this technology in mobile networks, two main limitations should be considered: control overhead (exchanges between SDN controller and network devices) and flow tables management.

That is why, in this paper, we introduced an innovative solution to address these two challenges. First of all, to improve flow tables management, we proposed an accurate

calculation of flow rules' hard timeout, based on vehicles' mobility prediction and network devices' load level prediction. Then, to reduce control overhead, we defined a Stateful Data Plane enabling to manage more efficiently vehicles mobility, using information provided by network devices (connections/disconnections). Finally, to improve both flow tables and control overhead management, we defined a flow rules pre-deployment policy, based on control channel load, flow table occupancy ratio and inter-cell transition probability.

The performances evaluation of the proposed approach demonstrates its advantages compared to existing solutions. Indeed, the proposed mechanisms (flow rules hard timeout calculation, Stateful Data Plane, pre-deployment policy) improve performances in terms of hard timeout accuracy, recovery delay, Packet In number and flow tables' occupancy ratio in any given case: uniform load distribution/non-uniform load distribution, overloaded channel/overloaded flow tables.

For future directions, the use of state machines to manage vehicle-to-vehicle communications in software-defined vehicular networks could be considered as this approach ensures low delays. However, because of the instability of the vehicle-to-vehicle communication links, the implementation of such a solution would require a significant evolution of the current proposition. For example, the use of Artificial Intelligence techniques could be considered.

REFERENCES

[1] A. Festag, "Cooperative intelligent transport systems standards in europe," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 166–172, Dec. 2014.

[2] V. Vukadinovic, K. Bakowski, P. Marsch, I. D. Garcia, H. Xu, M. Sybis, P. Sroka, K. Wesolowski, D. Lister, and I. Thibault, "3GPP C-V2X and IEEE 802.11p for vehicle-to-vehicle communications in highway platooning scenarios," *Ad Hoc Netw.*, vol. 74, pp. 17–29, May 2018.

[3] L. Mendiboure, M. A. Chalouf, and F. Krief, "A SDN-based pub/sub middleware for geographic content dissemination in Internet of vehicles," in *Proc. IEEE 90th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2019, pp. 1–6.

[4] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[5] I. Ku, Y. Lu, M. Gerla, R. L. Gomes, F. Ongaro, and E. Cerqueira, "Towards software-defined VANET: Architecture and services," in *Proc. 13th Annu. Medit. Ad Hoc Netw. Workshop (MED-HOC-NET)*, Jun. 2014, pp. 103–110.

[6] X. Wen, B. Yang, Y. Chen, L. E. Li, K. Bu, P. Zheng, Y. Yang, and C. Hu, "RuleTris: Minimizing rule update latency for TCAM-based SDN switches," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2016, pp. 179–188.

[7] M. Karakus and A. Durresi, "A survey: Control plane scalability issues and approaches in software-defined networking (SDN)," *Comput. Netw.*, vol. 112, pp. 279–293, Jan. 2017.

[8] T. Cheng, K. Wang, L.-C. Wang, and C.-W. Lee, "An in-switch rule caching and replacement algorithm in software defined networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[9] S. Bera, S. Misra, and M. S. Obaidat, "Mobi-flow: Mobility-aware adaptive flow-rule placement in software-defined access network," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1831–1842, Aug. 2019.

[10] L. Zhao, Z. Li, J. Li, A. Al-Dubai, G. Min, and A. Y. Zomaya, "A temporal-information-based adaptive routing algorithm for software defined vehicular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[11] A. U. Khan and B. K. Ratha, "Time series prediction QoS routing in software defined vehicular ad-hoc network," in *Proc. Int. Conf. Man Mach. Interfacing (MAMI)*, Dec. 2015, pp. 1–6.

[12] Q. Yuan, R. Chai, and Q. Chen, "End-to-end delay minimization-based joint rule caching and flow forwarding algorithm for SDN," in *Proc. 11th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2019, pp. 1–6.

[13] M. M. Hasan, S. Kwon, and J.-H. Na, "Adaptive mobility load balancing algorithm for LTE small-cell networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2205–2217, Apr. 2018.

[14] Y. Gao and T. Luo, "A load balancing scheme for supporting safety applications in heterogeneous software defined LTE-V networks," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–5.

[15] C. Sun, J. Bi, H. Chen, H. Hu, Z. Zheng, S. Zhu, and C. Wu, "SDPA: Toward a stateful data plane in software-defined networking," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3294–3308, Dec. 2017.

[16] C. Cascone, D. Sanvito, L. Pollini, A. Capone, and B. Sansò, "Fast failure detection and recovery in SDN with stateful data plane," *Int. J. Netw. Manage.*, vol. 27, no. 2, pp. 1–14, 2017.

[17] C. Cascone, L. Pollini, D. Sanvito, and A. Capone, "Traffic management applications for stateful SDN data plane," in *Proc. 4th Eur. Workshop Softw. Defined Netw.*, Sep. 2015, pp. 85–90.

[18] E.-D. Kim, Y. Choi, S.-I. Lee, and H. J. Kim, "Enhanced flow table management scheme with an LRU-based caching algorithm for SDN," *IEEE Access*, vol. 5, pp. 25555–25564, 2017.

[19] A. Panda, S. S. Samal, A. K. Turuk, A. Panda, and V. C. Venkatesh, "Dynamic hard timeout based flow table management in openflow enabled SDN," in *Proc. Int. Conf. Vis. Towards Emerg. Trends Commun. Netw. (ViTECoN)*, Mar. 2019, pp. 1–6.

[20] Z. Guo, R. Liu, Y. Xu, A. Gushchin, A. Walid, and H. J. Chao, "STAR: Preventing flow-table overflow in software-defined networks," *Comput. Netw.*, vol. 125, pp. 15–25, Oct. 2017.

[21] Z. Guo, Y. Xu, R. Liu, A. Gushchin, K.-Y. Chen, A. Walid, and H. J. Chao, "Balancing flow table occupancy and link utilization in software-defined networks," *Future Gener. Comput. Syst.*, vol. 89, pp. 213–223, Dec. 2018.

[22] F. Benamrane, M. Ben Mamoun, and R. Benaini, "An east-west interface for distributed SDN control plane: Implementation and evaluation," *Comput. Electr. Eng.*, vol. 57, pp. 162–175, Jan. 2017.

[23] S. Bera, S. Misra, and M. S. Obaidat, "Mobility-aware flow-table implementation in software-defined IoT," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[24] M. T. Abbas, A. Muhammad, and W.-C. Song, "SD-IoV: SDN enabled routing for Internet of vehicles in road-aware approach," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 3, pp. 1265–1280, Mar. 2020.

[25] S. H. Rastegar, A. Abbasfar, and V. Shah-Mansouri, "On fair rule caching in software defined radio access networks," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 460–463, Jun. 2018.

[26] M. Dong, H. Li, K. Ota, and J. Xiao, "Rule caching in SDN-enabled mobile access networks," *IEEE Netw.*, vol. 29, no. 4, pp. 40–45, Jul. 2015.

[27] H. Li, P. Li, and S. Guo, "MoRule: Optimized rule placement for mobile users in SDN-enabled access networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 4953–4958.

[28] Z. Li, M. A. Uusitalo, H. Shariatmadari, and B. Singh, "5G URLLC: Design challenges and system concepts," in *Proc. 15th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2018, pp. 1–6.

[29] K. M. Addali, S. Y. Bani Melhem, Y. Khamayseh, Z. Zhang, and M. Kadoch, "Dynamic mobility load balancing for 5G small-cell networks based on utility functions," *IEEE Access*, vol. 7, pp. 126998–127011, 2019.

[30] R. Wang, X. Peng, J. Zhang, and K. B. Letaief, "Mobility-aware caching for content-centric wireless networks: Modeling and methodology," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 77–83, Aug. 2016.

[31] H. D. Trinh, N. Bui, J. Widmer, L. Giupponi, and P. Dini, "Analysis and modeling of mobile traffic using real traces," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.

[32] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, "A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1790–1821, 3rd Quart., 2017.

[33] R. Wu, G. Luo, J. Shao, L. Tian, and C. Peng, "Location prediction on trajectory data: A review," *Big Data Mining Anal.*, vol. 1, no. 2, pp. 108–127, 2018.

[34] F. Khan and M. Portmann, "Backhaul, QoS, and channel-aware load balancing optimization in SDN-based LTE networks," in *Proc. 11th Int. Conf. Signal Process. Commun. Syst. (ICSPCS)*, Dec. 2017, pp. 1–10.

[35] A. Safa Sadiq, N. B. Fisal, K. Z. Ghafoor, and J. Lloret, "An adaptive handover prediction scheme for seamless mobility based wireless networks," *Sci. World J.*, vol. 2014, pp. 1–17, Dec. 2014.

[36] S. T. Girma and A. G. Abebe, "Mobility load balancing in cellular system with multicriteria handoff algorithm," *Adv. Fuzzy Syst.*, vol. 2017, pp. 1–8, Jan. 2017.

[37] H. Zhu, H. Fan, X. Luo, and Y. Jin, "Intelligent timeout master: Dynamic timeout for SDN-based data centers," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 734–737.

[38] Y. E. Oktian, S. Lee, H. Lee, and J. Lam, "Distributed SDN controller system: A survey on design choice," *Comput. Netw.*, vol. 121, pp. 100–111, Jul. 2017.

[39] R. R. Fontes, S. Afzal, S. H. Brito, M. A. Santos, and C. E. Rothenberg, "MiMininet-WiFi: Emulating software-defined wireless networks," in *Proc. 11th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2015, pp. 384–389.

[40] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO—Simulation of urban mobility: An overview," in *Proc. 3rd Int. Conf. Adv. Syst. Simulation ThinkMind*, 2011, pp. 1–6.

[41] A. L. Stancu, S. Halunga, A. Vulpe, G. Suciu, O. Fratu, and E. C. Popovici, "A comparison between several software defined networking controllers," in *Proc. 12th Int. Conf. Telecommun. Mod. Satell., Cable Broadcast. Services (TELSIKS)*, Oct. 2015, pp. 223–226.

[42] J. Dugan, "IPERF tutorial," Summer JointTechs, Columbus, OH, USA, Tech. Rep. 2010, 2010, pp. 1–4.

[43] K. Sumathi, V. Seethalakshmi, and M. A. Raja, "Analyzing qos based stable energy aware adhoc routing protocol with different mobility models," *Int. J. Adv. Eng.*, vol. 950, p. 954, Jun. 2016.

[44] H. T. Larasati, R. Hakimi, and T. Juhana, "Extended-LLF: A least loaded first (LLF)-based handover association control for software-defined wireless network," *Int. J. Comput. Eng. Inf. Technol.*, vol. 9, no. 9, p. 203, 2017.

[45] R. Taori and A. Sridharan, "Point-to-multipoint in-band mmwave backhaul for 5G networks," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 195–201, Jan. 2015.

[46] J. Yu, Y. Wang, K. Pei, S. Zhang, and J. Li, "A load balancing mechanism for multiple SDN controllers based on load informing strategy," in *Proc. 18th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Oct. 2016, pp. 1–4.

LEO MENDIBOURE (Student Member, IEEE) received the master's degree in telecommunications engineering from the ENSEIRB-MATMECA Engineering School. He is currently pursuing the Ph.D. degree with the University of Bordeaux, under the supervision of F. Krief and M. A. Chalouf. Since 2018, he has been a member of the CNRS LaBRI Laboratory, "Programming, Networks, and Systems" Team, University of Bordeaux. His main research interests include vehicular networks, and security and data management.

MOHAMED AYMEN CHALOUF received the Ph.D. degree from the University of Bordeaux. He is currently an Associate Professor with the University of Rennes 1 and a member of the Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA) Laboratory. His research has produced a number of publications in international journals and conferences. His research interests include the management of performances and the security of networks.
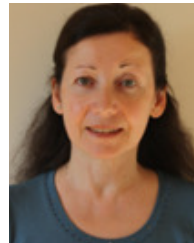
FRANCINE KRIEF is a Professor with Bordeaux INP, the Deputy Director of the CNRS LaBRI Laboratory, and a member of "Programming, Networks, and Systems" Team, University of Bordeaux. Her work on network control and management has led to many publications in journals and at conferences. Her main research interests include the control and management of QoS, and security and power consumption in new communication architectures.

● ● ●