A Fuzzy approach for load balancing in Blockchain-based Software Defined Vehicular Networks

Lylia Alouache
ETIS UMR 8051 CNRS lab
CY Cergy Paris University
F-95000 Cergy, France
lylia.alouache@cyu.fr

Tidiane Sylla
COSYS-ERENA Lab
Univ Gustave Eiffel
F-77447 Marne-la-Vallée, France
tidiane.sylla@univ-eiffel.fr

Leo Mendiboure
COSYS-ERENA Lab
Univ Gustave Eiffel
F-77447 Marne-la-Vallée, France
leo.mendiboure@univ-eiffel.fr

Hasnaa Aniss
COSYS-ERENA Lab Univ Gustave Eiffel
F-77447 Marne-la-Vallée, France
hasnaa.aniss@univ-eiffel.fr

***Abstract*—Software Defined Vehicular Networks (SDVN) are increasingly being considered for vehicular communication systems as they could provide both flexibility, programmability and improved performance. The SDVN control plane should be distributed between a set of SDN controllers managing specific geographical areas and network functions (load balancing, firewall, etc.) to ensure a high level of scalability. In this environment, the Blockchain technology could be used to guarantee the coordination of the different SDVN controllers. Therefore, many works have demonstrated the benefits of Blockchain-based SDVN architectures. However, they did not investigate the optimization of the underlying Blockchain network. That is why, in this paper, we propose a high-performance multi-level Blockchain architecture which could enable SDVNs to scale up. We also introduce a Fuzzy logic-based mechanism allowing the real-time reconfiguration of the Blockchain network depending on the actual SDVN requirements: traffic increase, integration of a new controller, etc. We finally demonstrate, through experimentation, the benefits of our approach compared to existing solutions in terms of latency, CPU, throughput and scalability.**

***Index Terms*—Vehicular Networks, SDN, Distributed Systems, Consistency, Blockchain, Scalability, Load Balancing**

I. Introduction

Efficient communication networks will be required to ensure the safety of future automated and connected vehicles [1]: lane change, obstacle detection, intersection management, etc. These Cooperative Intelligent Transport Systems (C-ITS) [2] will use both cellular networks (4G, 5G) and WiFi networks (ITS-G5) to provide a high level of connectivity.

Managing an architecture integrating different radio access technologies for highly mobile nodes will inevitably be complex. That is why Software Defined Network (SDN) [3] is now considered for C-ITS management. This technology, as demonstrated in different papers [4], [5], could be able to meet the high requirements of C-ITS (bandwidth, latency, reliability, etc.) in this challenging environment (connection time, speed, etc.). The concept of Software Defined Vehicular Networks (SDVN) [6] is therefore widely used in the literature.

Unlike traditional network architectures, the SDN architecture relies on the definition of a SDN control plane (SDN controller) managing all the routers and switches (SDN devices). This control plane has a centralized view of the network state and can optimize its operation by adjusting the devices' behaviour. To ensure a high level of scalability (a large number of network devices) this control plane is usually considered to be distributed [7], [8]. In such systems, it corresponds to a set of SDN controllers physically distributed to ensure both availability, fault-tolerance, low latency, low control overhead, and efficient load-balancing. Each controller manages a subset of network devices, located in the same geographical area, and potentially has a specific role: firewall, load balancing, core/edge network management, etc.

A solution increasingly considered to manage these distributed SDVN architectures is Blockchain [9]–[12]. This technology aims to implement efficient distributed systems using cryptographic primitives, peer-to-peer communications, and consensus algorithms and could therefore be useful for distributed SDVN's management [13]. A Blockchain node could be associated with each SDVN controller to ensure the coordination of these controllers (e.g. consistent flow rules) and to secure the SDVN control plan [14].

Many research papers have proposed Blockchain-based SDVN architectures [15]–[22]. However, 1) most of them have considered a global Blockchain network which could not be applied to large-scale SDVNs as shown in [22] 2) none of them addressed the reconfiguration/management of the Blockchain architecture to take into account the SDVN state: increased latency/throughput, integration of a new controller in the control plane, devices mobility management, etc.

To the best of our knowledge, none of the existing solutions enable the deployment and management of large-scale

978-1-6654-6975-3/22/$31.00 ©2022 IEEE

Blockchain-SDVN architectures. That is why, in this paper, we introduce a new multi-level Blockchain architecture to address the limitations of existing Blockchain-based SDVN architectures. We also define a Fuzzy Logic-based load balancing algorithm that could enable the real-time reconfiguration of the underlying Blockchain network. The main contributions of this paper are:

- A review of existing Blockchain-based architectures for SDVN and an identification of their limitations;
- The design of a high-performance multi-level Blockchain architecture to enable both large-scale SDVN deployment and dynamic reconfiguration of the Blockchain network;
- The definition of a Fuzzy Logic-based load balancing algorithm to adaptively reconfigure the Blockchain architecture according to the actual SDVN state;
- The implementation and evaluation of our approach compared to existing solutions (CPU, Throughput, Latency).

The rest of this paper is organized as follows. Section 2 compares state-of-the-art Blockchain-based solutions for SDVNs. Then, Section 3 presents the proposed Blockchain architecture for distributed SDVN management. Thereafter, Section 4 introduces a new Fuzzy Logic-based load balancing algorithm for Blockchain-based SDVN. Finally, in Section 5, performances of our system are compared to existing solutions.

II. Related works

In this section, existing Blockchain-based distributed SDVN architectures are presented (Section 2.A) and their limitations are identified (Section 2.B).

A. State-of-the-art solutions

State-of-the-art Blockchain-based architectures for SDVNs can be classified into two categories:

- Studies that are based on classical Blockchain architectures [15]–[20]: These papers are mainly aimed at ensuring the consistency of the distributed SDVN control plane by leveraging Blockchain technology. Therefore, they have a strong value as they demonstrated the benefits of the Blockchain for both security [16] trust establishment [15] and reliability/coordination [20] in SDVN. They also introduced mechanisms to easily interconnect the Blockchain to external systems and this could be applied to many topics: Federated Learning, Internet of Things, etc. However, these works did not consider the limitations of Blockchain architecture in large-scale deployments [23] (latency, bandwidth) and therefore could not guarantee the management of large-scale SDVNs;
- Studies that have proposed novel Blockchain architectures for SDVN [21], [22]: These papers sought to ensure the scalability of the Blockchain-based SDVN architecture. They introduced the idea of Blockchain architectures broken down into sub-networks (distributed Blockchain architectures) to guarantee local data consistency (in each Blockchain sub-network) and high performance (latency, throughput) as shown in [22]. However, these solutions considered a decomposition into a fixed set of sub-networks without taking into account the potential limitations of such an approach: 1) integration of new SDVN controllers into the Blockchain architecture, 2) unequal load distribution between the different sub-networks and 3) increased communication load (high throughput required). Therefore, these approaches have important limitations in terms of Blockchain-based SDVN management and scalability.

B. Positioning

Existing works have demonstrated the benefits of Blockchain for control plane management and security in distributed SDVN [17], [18]. They have also introduced new Blockchain architectures to deal with SDVN characteristics [21], [22]: high mobility, frequent disconnections, etc. However, we identified two important limitations: 1) existing solutions based on a single Blockchain network could not support large-scale SDVN management/deployment [22] 2) architectures proposed for SDVN do not encompass the idea of adaptability and would not be able to meet the real-time requirements of SDVN: new controller integration, variable traffic, etc. That is why, in this paper, to overcome these limitations, we propose a) a new multi-level reconfigurable Blockchain architecture for SDVN and b) a Fuzzy Logic-based load balancing algorithm to to manage the deployed Blockchain network and guarantee SDVN scaling.

III. A multi-level Blockchain architecture for SDVN

In this section we introduce a new Blockchain architecture for SDVN. This architecture, compared to existing solutions 1) integrates edge Blockchain nodes to simplify the Blockchain network reconfiguration and enable global data exchange 2) includes a new control plan to optimize the Blockchain network operation.

The main role of SDVN controllers is to manage the Radio Access Network (RAN) [22]. Thus, in a distributed SDVN control plane, data exchange must be allowed between neighbouring controllers, controlling specific geographic areas, to manage vehicle mobility and resources allocation. The underlying Blockchain architecture must allow, in real-time, these neighbouring SDN controllers to access the same blocks of data to enable them to coordinate their decision making. Moreover, the SDVN control plane could integrate new SDVN controllers, potentially hosted in vehicles [24], to ensure the scaling up of SDVN. The proposed Blockchain architecture should also guarantee a high level of adaptability.

That is why the characteristics of the Blockchain architecture (cf. Figure 1) we propose are:

- This architecture is composed of a set of Blockchain sub-networks to ensure a high level of scalability (2 sub-networks on the architecture described in Figure 1). Such a distributed architecture ensures low latency exchanges between neighbouring controllers and efficient SDVN management. Each subnet corresponds to a given

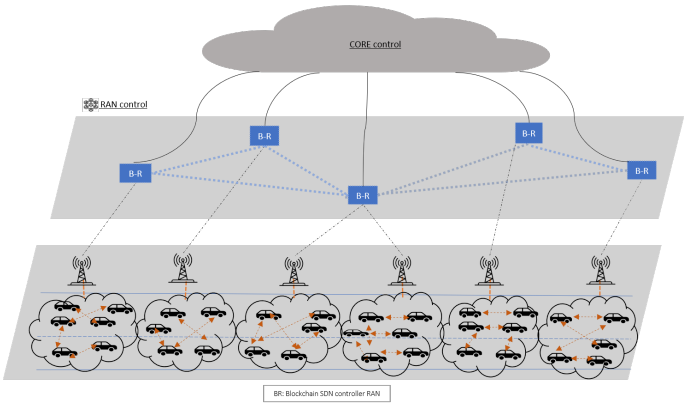

Fig. 1. Proposed architecture for scalable and flexible SDVN

geographical area and a set of SDVN controllers. The Blockchain nodes belonging to these sub-networks will be called BR in Section 4;

- This architecture considers Blockchain nodes at the edge of different sub-networks (cf. the central node in Figure 1). These edge nodes share their updates with all the sub-networks they belong to. This ensures an efficient mobility management of SDVN devices between neighbouring subnets. This also provides an easy way to detach a Blockchain node from a sub-network and integrate it into a new one. Indeed, since it already has the Blockchain ledger of a set of neighbouring networks, it will not have to download them. Two types of Blockchain nodes exist in our architecture: a) those that belong to a single sub-network and b) those that belong to a set of sub-networks. This system guarantees both efficient local decision-making (sub-network decomposition) and sufficient data transmission (edge Blockchain nodes);
- This architecture introduces a CORE Control plane on top of the SDVN architecture. All Blockchain nodes are managed by this CORE plane and report state's information to this plane (cf. Section 4). The integration of this plane will enable the Blockchain-SDVN architecture to be reconfigured when it will be necessary. It is able to 1) build new Blockchain sub-networks 2) integrate new nodes into an existing Blockchain sub-network 3) merge multiple sub-networks 4) move Blockchain nodes from one sub-network to another. The mechanisms for decision-making in this CORE Control plane are described in the next section of this paper.

IV. A Fuzzy algorithm for load balancing in Blockchain-based SDVN

In this section, we define a Fuzzy Logic-based load balancing algorithm for Blockchain-based distributed SDVN.

A. Aims of the algorithm

Our idea is to enable real-time reconfiguration of the Blockchain network used by SDVNs to maximize network performance. The proposed approach, relying on a CORE plane (cf. Section 3), should achieve the following goals:

- Controllers coordination: SDVN controllers may simultaneously try to install conflicting routing/flow rules. This could lead to routing failures. Our solution must therefore guarantee very low latency exchanges between SDVN controllers to limit conflicts and ensure efficient coordination of decision making;
- Flexibility: The SDVN network may evolve over time: integration of new SDVN controllers, variation of the network load, etc. The proposed solution must be able to react to these evolutions and allow an efficient reconfiguration of the underlying Blockchain network by splitting and merging existing sub-networks;
- Security and privacy: The SDVN control scheme must ensure a high level of security to protect the network from malicious entities [22]. The proposed solution must therefore comply with the principle of Blockchain architectures as they aim to offer such a level of security;
- Scalability: Vehicular networks could have to manage large numbers of vehicles in the future. That is why, in the proposed solution, the CORE plane could also be considered as physically distributed (higher scalability). In such a scenario, each CORE controller would manage a defined set of SDVN sub-networks.

B. Fuzzy Inference System

Our approach uses the Fuzzy Logic [25], a well-known Artificial Intelligence technique to make load balancing decisions. The proposed Fuzzy Inference System (FIS) uses Key Performance Indicators (KPIs) as inputs. Then the CORE controller uses this inference model to determine 1) the performance level of the currently implemented Blockchain architecture and 2) the solution that should be used to optimize this performance level (merge, split, load re-balancing).

To be used as inputs of our FIS, the KPIs values (numeric values) are converted into two linguistic terms (fuzzified): *Low* and *High*. *Low* set contains values corresponding to a low ratio of the considered indicator (Latency, Throughput and Number of nodes). *High* set contains values corresponding to a high ratio of a given indicator. Limits of Fuzzy Membership values are not accurately defined and are dependent on the determined objectives [25]. We determined the network state (Underloaded, Satisfying, Overloaded) according to different fuzzy variables and rules and based on the fixed objectives. The considered KPIs (fuzzy variables) are:

1) Latency: It corresponds to the average amount of time required to transmit an information in a Blockchain sub-network (rule emitted by a SDVN controller). This duration must necessarily be lower than the duration of the decision making of an SDVN controller (usually 25ms as proposed in [26], [27]). This duration is measured over a given period (e.g. the last minute) and aims to indicate the current state of the network.
2) Throughput: It corresponds to the number of transactions per second (rate) that can be supported by a

Blockchain network and this is an important for the scaling up of the SDVN architecture. Throughput depends on the Blockchain technology considered. For example, with Hyperledger Fabric (cf. Section 5), the maximum throughput, depending on the deployment environment, can vary between 200 tr/s [28] and 1000 tr/s [29];

3) Number of nodes: The number of nodes that can be deployed in a sub-network is also an important element for scaling. Nodes' number depends on the technology used. As explained in [29], for Hyperledger Fabric, performance could decrease for a number of nodes superior to 16. Moreover, a minimum number of 4 nodes could also be required to ensure security and to take advantage of the Blockchain architecture. In this paper, the number of nodes $NbNodes$ must be in the interval [6,14] to be in the "satisfying" state.

The equations 1, 4 and 7 define fuzzy variables membership functions for latency, throughput and number of nodes respectively. As stated above, the linguistic terms of the considered fuzzy variables are: μ_{Low} and μ_{High}. The membership functions' thresholds are defined for each KPI (Latency: (2) and (3), Throughput (5) and (6), Number of nodes (8) and (9)). As stated previously, we have fixed these membership functions boundaries based on the performances requirement of the SDVN sub-networks. These requirement are well documented in the literature [26]–[29].

1) Latency:

$$AvgL = \frac{MaxLatency(BR_i)}{LThreashold} \tag{1}$$

$AvgL$ corresponds to the highest latency value measured in a sub-network BR_i managed by the CORE Controller. The membership features of $AvgL$ are:

$$\mu_{low}(AvgL_L) = \begin{cases} 1 & 0 \leq AvgL \leq 10\% \\ \frac{50\%-AvgL}{50\%-10\%} & 10\% < AvgL \leq 50\% \\ 0 & 50\% < AvgL \leq 100\% \end{cases} \tag{2}$$

$$\mu_{high}(AvgL_H) = \begin{cases} 0 & 0 \leq AvgL \leq 25\% \\ \frac{AvgL-25\%}{75-25\%} & 0 < AvgL \leq 75\% \\ 1 & 75\% < AvgL \leq 100\% \end{cases} \tag{3}$$

2) Throughput:

$$AvgTh = \frac{AvgTh(BR_i)}{TThreashold} \tag{4}$$

The membership features of $Avgth$ are:

$$\mu_{low}(AvgTh_L) = \begin{cases} 1 & 0 \leq AvgTh \leq 25\% \\ \frac{50\%-AvgTh}{50\%-25\%} & 25\% < AvgTh \leq 50\% \\ 0 & 50\% < AvgTh \leq 100\% \end{cases} \tag{5}$$

$$\mu_{high}(AvgTh_H) = \begin{cases} 0 & 0 \leq AvgTh \leq 25\% \\ \frac{AvgTh-25\%}{80\%-25\%} & 0 < AvgTh \leq 80\% \\ 1 & 80\% < AvgTh \leq 100\% \end{cases} \tag{6}$$

3) Number of nodes:

$$NbNRate = \frac{NbNodes(BR_i)}{NbThreshold} \tag{7}$$

The membership features of $NbNRate$ are:

$$\mu_{low}(NbNRate_L) = \begin{cases} 1 & 0 \leq NbNRate \leq 37\% \\ \frac{87\%-NbNRate}{87\%-37\%} & 37\% < NbNRate \leq 87\% \\ 0 & 87\% < NbNRate \leq 100\% \end{cases} \tag{8}$$

$$\mu_{high}(NbNRate_H) = \begin{cases} 0 & 0 \leq NbNRate \leq 25\% \\ \frac{NbNRate-87\%}{87\%-25\%} & 25\% < NbNRate \leq 87\% \\ 1 & 87\% < NbNRate \leq 100\% \end{cases} \tag{9}$$

Our FIS is based on the Mamdani model. Fuzzy Rules are used to determine BR sub-networks' state. For each rule in the rule table, an appropriate implication must be applied. Each implication is comprised of an antecedent and a consequent [30]. The output of the inference system is also a fuzzy variable. The proposed rules (R1-R8) of our system are described in table I.

TABLE I
STATE RULE'S TABLE

Rule	Latency	Throughput	NBnodes	State
R1	low	low	low	Satisfying
R2	low	low	high	Satisfying
R3	low	high	low	Overloaded
R4	low	high	high	Overloaded
R5	high	low	low	Underloaded
R6	high	low	high	Overloaded
R7	high	high	low	Overloaded
R8	high	high	high	Overloaded

Tules are evaluated at the inference stage of the fuzzy logic system. The result is aggregated then defuzzified to obtain the output State variable. More specifically, the defuzzification process consists of converting the output fuzzy value into a crispy value. There are different defuzzification methods, such as [25]: maxima methods, center of gravity (CoG) method, center of sums (CoS) method, center of largest area (CoA), etc. In view of the fuzzy variables used in our approach, we use the CoG method as defuzzification method. This method is used with Mamdani's Max-Min inference, which has no interpolation effect. Thus, it provides an interpolation proportional to the size of the substantial fuzzy subsets [31]. Consequently, COG method is the most suitable method for our system. Indeed, the CoG can be calculated for our system inputs having Nonsmooth membership functions. The Crispy values are determined using Formula 10. Figure 2 shows the membership functions characterizing the State fuzzy variable.

$$x^* = \frac{\sum_{i=1}^{n} \mu_{A_i}(x_i).x_i}{\sum_{i=1}^{n} \mu_{A_i}(x_i)} \tag{10}$$

The scenario described below illustrates the implementation of our fuzzy system. In this scenario, the following thresholds are considered for the input KPIs: $LThreashold = 25ms$, $TThreashold = 245r/s$ and $NbThreashold = 12nodes$.
A given BR sub-network is characterized by the following indicators: $MaxLatency(BR_i) = 10ms$, $Avgth = 230r/s$ and

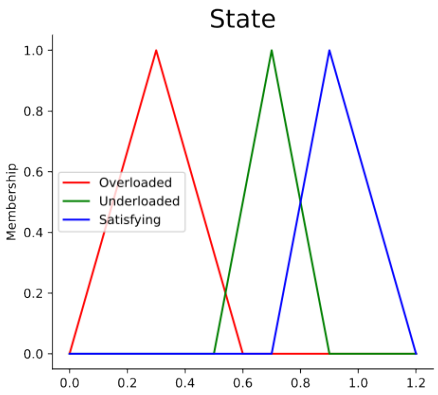

Fig. 2. State membership function

$NbNnode = 11nodes$. In our fuzzy model, the inputs of this scenario are: $AvgL = 0.4$, $Avgth = 0.9387755102040817$ and $NbNRate = 0.9166666666666665$. According to Table I, the rule corresponding to this case is the rule $R4$ and based on the membership functions and the inference model, the output of this use case is shown in Figure 3. The State fuzzy value is Overloaded. The corresponding crispy value x^*_{S1} (calculated by formula 10) is represented by the black projection on the abscissa axis (Figure 3). In Section 4.C we describe how this information is then used.

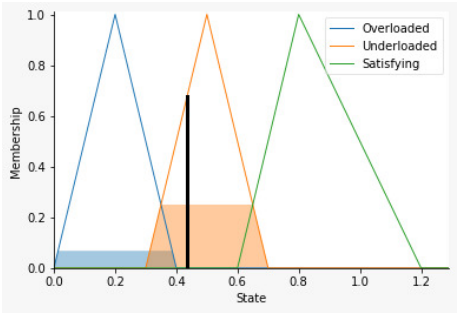

Fig. 3. Overloaded Output State

C. Load balancing algorithm

Our Fuzzy Inference System is used by the CORE controller to determine the best solution optimizing both network operation and network performance level. Using the KPIs defined in section 4.B, different actions are performed (Split, Merge, Load re-balancing) to redefine a high-performance sub-network Blockchain architecture (low latency, high throughput, high scalability). The structure of our Fuzzy based Load Balancing Algorithm for BR-SDVN is shown in Figure 4. It describes the algorithm components and the relationships between them. The main steps required for a CORE controller to manage a set of BR sub-networks are the following:

1) Every period of time τ, the CORE Controller collects KPIs information from subnets (GET requests) using Algorithm 1. This process can also be initiated by the instantiation of a new Blockchain node;
2) The CORE Controller determines and evaluates the fuzzy KPIs (input/output) for each subnet using the proposed Fuzzy Inference System. Sub-networks state is recorded at this stage when it is abnormal. The "Overloaded" State is associated with the "Split" Action and the "Underloaded" State is associated with the "Merge" action) (Algorithm 1);
3) The CORE controller determines a new optimal number of sub-networks (Algorithm 2) using a simple criterion: if the number *n-split* of requested Splits actions ("Overloaded" State) is higher than the number *n-merge* of requested Merge actions ("Underloaded State"), the controller splits (*n-split* - *n-merge*) sub-networks to get to a balanced state (*n-split* equivalent to *n-merge*). Similarly, if *n-merge* is higher than *n-split*, the controller merges (*n-merge* - *n-split*) subnets;
4) The CORE controller finally re-balances load if sub-nets are still over or under loaded. The load is thus sequentially transferred from overloaded sub-networks to underloaded networks (Blockchain nodes relocation).

At this point, the sub-networks managed by the CORE controller reach a new steady state guaranteeing an acceptable latency and sufficient throughput in all sub-networks.

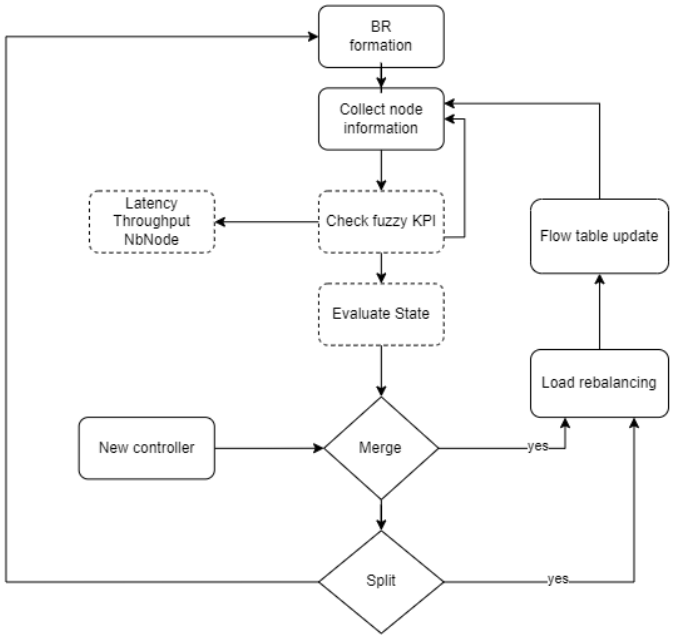

Fig. 4. BR-SDVN controllers management flowchart

V. Evaluation

A. Objectives

In this section, the benefits of the solution (cf. Section 3, 4) that we proposed for Blockchain-based SDVN are assessed. The main improvements compared to existing solutions are 1)

Algorithm 1: BR sub-networks monitoring by CORE node

```
Let τ be a period of time in which CORE node collects
  information about its BR sub-networks;
input : Set of BR sub-networks
output: SplitTable, MergeTable
for Every period of time τ do
    for [i=1;i≤ NumberOfSubnets;i++] do
        Send(State-Request) to BR sub-networks;
        Determine(BR-Abnormal-State);
        if [BR-Abnormal-State == Overloaded] then
            Update(SplitTable);
        else
            Update(MergeTable);
        end
    end
end
```

Algorithm 2: Fuzzy based Load Balancing Algorithm for BR-SDVN

```
input : SplitTable, MergeTable
if [Sizeof(SplitTable)≥Sizeof(MergeTable)] then
    Sort(SplitTable) by decreasing Throughput;
    Sort(MergeTable) by increasing NumberofNodes;
    j=Sizeof(SplitTable)-Sizeof(MergeTable);
    for [i=1;i≤j;i++] do
        Split(SplitTable[i]);
    end
    for [i=1;i≤Sizeof(MergeTable);i++] do
        Offload(SplitTable[j+i],MergeTable[i]);
    end
else
    Sort(MergeTable) by deacreasing Throughput;
    i=1; CumulThput=0;
    while[i≤Sizeof(MergeTable)]∧
      [CumulThput≠Satisfying]
    CumulThput=CumulThput+Thrghput(MergeTable[i]);
    i++;
    Merge(MergeTable) from 1 to i;
    for [j=i+1;i≤Sizeof(MergeTable);j++] do
        Offload(MergeTable[j],MergeTable[j+1]);
    end
end
```

the design of an optimized Blockchain architecture for SDVN (sub-networks + CORE plane) and 2) the definition of a Fuzzy logic-based load balancing algorithm for Blockchain network optimization. We therefore evaluated:

- The benefits of sub-network decomposition for Blockchain-SDVN (Section V.C);
- The benefits of load balancing for Blockchain networks in SDVN (Section V.D).

B. Experimental setup

The experiments conducted in this paper were based on:

- Hyperledger Fabric: a Blockchain platform designed for permissioned Blockchain networks' deployment that is used in industrial and research sectors [32]. This platform allows both data exchange between Blockchain nodes (transactions) and Blockchain sub-networks deployment (Hyperledger Channels) [33] (Version 1.4.2);
- Hyperledger Caliper: a Blockchain benchmark tool used with Fabric to determine the level of performance of pre-defined Blockchain architectures and scenarios both in terms of latency, throughput and CPU [29].

Based on these tools we considered the deployment of a fixed number of Blockchain nodes (8 Hyperledger Fabric Peers deployed in Virtual Machines with 16GB of RAM) and the implementation of a variable number of sub-networks (2 to 8 Hyperledger channels). To simulate exchanges between SDVN controllers (OpenFlow protocol), we generated transactions in the Blockchain network. These transactions would correspond to the different operations enabled with the OpenFlow Protocol (V1.5) [34] to manage flow rules in SDN switches: Packet-In, FlowMod, etc. In this experiment we focused on assessing the performance level of the deployed Blockchain architecture/network structure. Only writing cases (transactions added to the Blockchain network/Rules generated by an SDVN controller) have been considered in this article as this is the main limiting factor of Blockchain architectures as explained in [22].

C. Assessment of sub-network splitting performance

We first compared the performance of an architecture composed of a single network (MAIN) with architectures composed of a variable number of sub-networks (2 to 8): SUB-2, SUB-4, SUB-8. Three different parameters were evaluated, 1) the maximum rate achieved by the system for a variable number of requests per second (100 to 700 r/s), 2) the elapsed time, in milliseconds, between the request and its response (i.e. its addition to the Blockchain ledger) and 3) the average amount of Central Processing Unit (CPU) resources used by each node of the BC network. It can be noted that the number of requests received by each sub-network corresponds to *1/number_of_subnets* of the total number of requests sent.

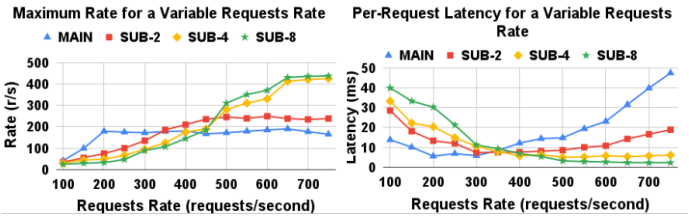

Fig. 5. Assessment of sub-network splitting performance

The results obtained in this evaluation demonstrate the benefits of sub-networks decomposition (cf. Figure 5, Figure 6.A) for Blockchain-based SDVN. When the number of request/second increases, the performance of sub-networked approaches (SUB-2, SUB-4, SUB-8) outperforms the MAIN performance. In particular, regarding the achievable rate, below

350 r/s the MAIN approach guarantees higher performance but above this value the sub-network approaches are more efficient, on average: 2.2x for SUB-8, 1.8x for SUB-4, 1.4x for SUB-2. Moreover, in the considered deployment, even for a high number of requests per second (e.g. 600), the MAIN architecture is unable to handle a number of transactions per second higher than 190. Regarding latency, it can be divided, even for the SUB-2 approach, by half compared to the MAIN approach for a high number of r/s (600 and higher). The SUB-8 approach divides the latency by 10. We can also add that sub-networks decomposition improves CPU usage since the usage level is two to three times higher than the MAIN approach (cf. Figure 6.A).

Performance increase in high load scenarios, for SUB-* approaches, can be explained by the Blockchain's operating scheme. This involves the sequencing of different tasks [32]: endorsement, commit, ordering. With a sub-network structure, these tasks can be parallelized by the different sub-networks. That is why SUB-* approaches can process a higher number of transactions per second. However, SUB-6 and SUB-8 approaches have similar performance levels while CPU usage is increased with the SUB-8 approach due to the management of a larger number of sub-networks. The parallelization capacity seems to reach its limits at this point. We can also add that the underperformance of the sub-network decomposition in low load scenarios is explained by the fixed block size considered for block generation (classic operation of a Blockchain architecture): when the number of r/s is low, block generation is spaced out and latency is consequently increased.

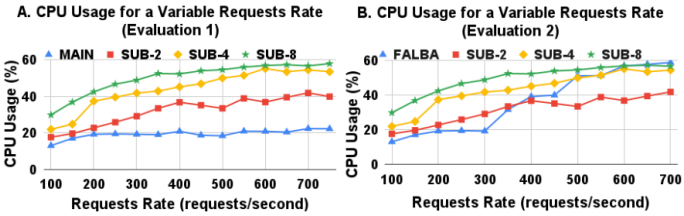

Fig. 6. Assessment of CPU performance for both evaluations

D. Assessment of load balancing performance

We then evaluated the benefits of our Fuzzy bAsed Load Balancing Algorithm (FALBA) (cf. Section 4). Such an algorithm, optimizing the operation of the Blockchain architecture, have not been proposed in the literature yet (cf. Section 2). Therefore, we compared our adaptive solution (FALBA) with fixed Blockchain sub-network architectures (cf. Section 5.C): SUB-2, SUB-4, SUB-8. We considered a variable number of requests per second (100 to 700) and two different cases: a) a uniform requests distribution among the different subnets (as in 5.C) and b) a non-uniform requests distribution among the different subnets. For both of these cases and a variable number of r/s three parameters were evaluated 1) the maximum rate, 2) the per-request latency and 3) the average CPU usage.

As shown in Figure 7, in both cases (Uniform/Non-uniform) FALBA guarantees on average higher performance than SUB-

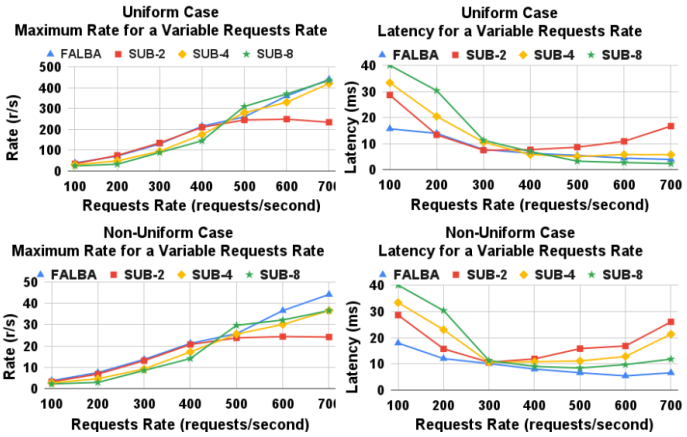

Fig. 7. Assessment of load balancing performance

* approaches both in terms of latency and maximum rate. FALBA selects the most suitable network architecture (eg. for uniform case): SINGLE up to 200 r/s, SUB-2 up to 450 r/s, SUB-4 up to 550 r/s and SUB-8 up to 700 r/s. In the uniform case, FALBA does not systematically provides the maximum throughput: between 450 and 550 r/s the SUB-8 approach performs better while SUB-4 is selected by the FALBA algorithm as SUB-4 guarantees both sufficient rate and low latency. However, FALBA systematically guarantee a latency below 25 ms, an essential criterion as explained in Section 4. None of the other approaches allow this. In the non-uniform case, the benefits of FALBA are higher. The performance level of SUB-* approaches is reduced by the non-uniform requests distribution whereas FALBA rebalances the load among the different subnets. FALBA always provides the highest rate except at 500r/s (SUB-4 approach is selected while SUB-8 is more efficient even in this non-uniform cases). Regarding latency, FALBA is the only approach that guarantees a latency lower than 25ms (as in uniform case). Dynamic network management also optimizes CPU usage (cf. Figure 6.B). These results, corresponding to both uniform and non-uniform cases (similar measurements), show that FALBA maximises performance while minimising the amount of CPU resources required to ensure this level of performance.

By adapting the number of sub-networks to the number of requests per second it is therefore possible to improve the performance of the Blockchain architecture. This is directly related to the results of the first evaluation: the optimal number of sub-networks is directly related to the actual requirements of the Blockchain network (number of requests per second to process, expected latency, block size, etc.). A highly parallelized architecture will be more efficient in high load scenarios but will show limitations when the number of transactions per second is low. Load balancing and node migration also enables a better distribution of requests between subnets and maintains a constant block generation duration. Our approach therefore provides a significant performance improvement.

VI. Conclusions

In this paper we proposed a novel solution for Blockchain-based Software Defined Vehicular Networks. It is based on the definition of a new Blockchain architecture, composed of a set of sub-networks, guaranteeing low latency exchanges between neighbouring SDVN controllers (same geographical area). This architecture also integrates a CORE control plane that could manage the underlying Blockchain architecture. Beyond that we also introduced a Fuzzy-based load balancing algorithm to reconfigure the Blockchain network according to the real-time requirements of the Blockchain-SDVN network: latency, throughput, variable number of controllers, etc.

The experiments carried out in this paper demonstrated the benefits of the proposed architecture and load balancing algorithm in terms of throughput, latency and CPU usage. Our approach could therefore be interesting for studies focusing on distributed SDVN's scalability. As future work we plan to improve the proposed solution by integrating more complex artificial intelligence tools and by defining new mechanisms for SDVN controllers coordination.

References

[1] C. D. Yang, K. Ozbay, and X. Ban, "Developments in connected and automated vehicles," pp. 251–254, 2017.

[2] A. Festag, "Cooperative intelligent transport systems standards in europe," *IEEE communications magazine*, vol. 52, no. 12, pp. 166–172, 2014.

[3] R. Masoudi and A. Ghaffari, "Software defined networks: A survey," *Journal of Network and computer Applications*, vol. 67, pp. 1–25, 2016.

[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, 2017.

[5] T. Mekki, I. Jabri, A. Rachedi, and L. Chaari, "Software-defined networking in vehicular networks: A survey," *Transactions on Emerging Telecommunications Technologies*, p. e4265, 2021.

[6] X. Ge, Z. Li, and S. Li, "5g software defined vehicular networks," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 87–93, 2017.

[7] K. S. K. Liyanage, M. Ma, and P. H. J. Chong, "Controller placement optimization in hierarchical distributed software defined vehicular networks," *Computer Networks*, vol. 135, pp. 226–239, 2018.

[8] A. Alioua, S.-M. Senouci, and S. Moussaoui, "dsdivn: a distributed software-defined networking architecture for infrastructure-less vehicular networks," in *International conference on innovations for community services*. Springer, 2017, pp. 56–67.

[9] P. Li, S. Guo, J. Wu, and Q. Zhao, "Blockrev: Blockchain-enabled multi-controller rule enforcement verification in sdn," *Security and Communication Networks*, vol. 2022, 2022.

[10] M. Azab, R. R. Ergawy, E. M. Ghourab, A. Mokhtar, and M. Rizk, "Towards blockchain-based multi-controller managed switching for trustworthy sdn operation," in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2019, pp. 0991–0998.

[11] H. Yang, Y. Liang, Q. Yao, S. Guo, A. Yu, and J. Zhang, "Blockchain-based secure distributed control for software defined optical networking," *China Communications*, vol. 16, no. 6, pp. 42–54, 2019.

[12] A. Rahman, M. J. Islam, A. Montieri, M. K. Nasir, M. M. Reza, S. S. Band, A. Pescape, M. Hasan, M. Sookhak, and A. Mosavi, "Smartblocksdn: an optimized blockchain-sdn framework for resource management in iot," *IEEE Access*, vol. 9, pp. 28 361–28 376, 2021.

[13] L. Mendiboure, M. A. Chalouf, and F. Krief, "Survey on blockchain-based applications in internet of vehicles," *Computers & Electrical Engineering*, vol. 84, p. 106646, 2020.

[14] T. Sylla, L. Mendiboure, M. A. Chalouf, and F. Krief, "Blockchain-based context-aware authorization management as a service in iot," *Sensors*, vol. 21, no. 22, p. 7656, 2021.

[15] L. Mendiboure, M. A. Chalouf, and F. Krief, "Towards a blockchain-based sd-iov for applications authentication and trust management," in *International Conference on Internet of Vehicles*. Springer, 2018, pp. 265–277.

[16] J. Gao, K. O.-B. O. Agyekum, E. B. Sifah, K. N. Acheampong, Q. Xia, X. Du, M. Guizani, and H. Xia, "A blockchain-sdn-enabled internet of vehicles environment for fog computing and 5g networks," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4278–4291, 2019.

[17] L. Vishwakarma, A. Nahar, and D. Das, "Lbsv: Lightweight blockchain security protocol for secure storage and communication in sdn-enabled iov," *IEEE Transactions on Vehicular Technology*, 2022.

[18] A. Derhab, M. Guerroumi, M. Belaoued, and O. Cheikhrouhou, "Bmc-sdn: blockchain-based multicontroller architecture for secure software-defined networks," *Wireless Communications and Mobile Computing*, vol. 2021, 2021.

[19] J. Luo, Q. Chen, F. R. Yu, and L. Tang, "Blockchain-enabled software-defined industrial internet of things with deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5466–5480, 2020.

[20] G. S. Aujla, M. Singh, A. Bose, N. Kumar, G. Han, and R. Buyya, "Blocksdn: Blockchain-as-a-service for software defined networking in smart city applications," *IEEE Network*, vol. 34, no. 2, pp. 83–91, 2020.

[21] D. Zhang, F. R. Yu, and R. Yang, "Blockchain-based distributed software-defined vehicular networks: A dueling deep q-learning approach," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1086–1100, 2019.

[22] L. Mendiboure, M. A. Chalouf, and F. Krief, "A scalable blockchain-based approach for authentication and access control in software defined vehicular networks," in *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2020, pp. 1–11.

[23] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *Ieee Access*, vol. 8, pp. 16 440–16 455, 2020.

[24] O. S. Al-Heety, Z. Zakaria, M. Ismail, M. M. Shakir, S. Alani, and H. Alsariera, "A comprehensive survey: Benefits, services, recent works, challenges, security, and use cases for sdn-vanet," *IEEE Access*, vol. 8, pp. 91 028–91 047, 2020.

[25] R. R. Yager and L. A. Zadeh, *An introduction to fuzzy logic applications in intelligent systems*. Springer Science & Business Media, 2012, vol. 165.

[26] T. Wang, F. Liu, J. Guo, and H. Xu, "Dynamic sdn controller assignment in data center networks: Stable matching with transfers," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[27] H. Chen and T. Benson, "The case for making tight control plane latency guarantees in sdn switches," in *Proceedings of the Symposium on SDN Research*, 2017, pp. 150–156.

[28] M. Kuzlu, M. Pipattanasomporn, L. Gurses, and S. Rahman, "Performance analysis of a hyperledger fabric blockchain framework: throughput, latency and scalability," in *2019 IEEE international conference on blockchain (Blockchain)*. IEEE, 2019, pp. 536–540.

[29] Q. Nasir, I. A. Qasse, M. Abu Talib, and A. B. Nassif, "Performance analysis of hyperledger fabric platforms," *Security and Communication Networks*, vol. 2018, 2018.

[30] T. Sylla, M. A. Chalouf, F. Krief, and K. Samaké, "Setucom: Secure and trustworthy context management for context-aware security and privacy in the internet of things," *Security and Communication Networks*, vol. 2021, 2021.

[31] V. H. Grisales Palacio *et al.*, "Modélisation et commande floues de type takagi-sugeno appliquées à un bioprocédé de traitement des eaux usées," 2007.

[32] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.

[33] A. Baliga, N. Solanki, S. Verekar, A. Pednekar, P. Kamat, and S. Chatterjee, "Performance characterization of hyperledger fabric," in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 2018, pp. 65–74.

[34] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using openflow: A survey," *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 493–512, 2013.