BIND: Blockchain-Based Flow-Table Partitioning in Distributed Multi-Tenant Software-Defined Networks

Ayan Mondal and Sudip Misra
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur, India
Email: {ayanmondal, sudipm}@iitkgp.ac.in

***Abstract*—In this paper, we study the problem of flow-table partitioning in distributed multi-tenant software-defined networks (SDNs) having Internet-of-things (IoT) devices. In the existing literature, the optimal usage of the ternary content-addressable memory (TCAM) is studied in the context of data traffic management by introducing the soft flow-table partitioning in the presence of a centralized controller. However, in the presence of distributed multi-tenant controllers, the soft flow-table partitioning may introduce a monopoly among the controllers. Hence, there is a need to design a flow-table partitioning scheme for distributed multi-tenant SDN, while maximizing the network sustainability and throughput. In this work, we propose a utility game-based scheme, named BIND, for dynamic flow-table partitioning. To ensure cooperation among the controllers and to avoid monopoly, we introduce the use of a blockchain among the multi-tenant controllers. Additionally, using BIND, we ensure that each controller gets a fair chance for flow-rule replacement. Moreover, network sustainability is ensured in BIND, while minimizing the flow-rule replacement in the flow-tables and multi-tenant SDN. Through simulation, we observe that using BIND, fairness in flow-rule placement is ensured. Additionally, the network overhead is reduced significantly.**

***Index Terms*—Software-Defined Networks, Internet of Things (IoT), Game Theory, Flow-Table Partitioning, Multi-Tenant, Blockchain.**

I. Introduction

In the presence of Internet of things (IoT) devices, the number of applications in the network increased significantly. Each application generates a set of flows. To handle these flows, we envision that software-defined network (SDN) is one of the promising technologies [1]. SDN separates the control and data planes [2] and introduces two entities — controller and software-defined switch. In SDN, the controller acts as the centralized entity and installs the flow-rules to switches to accommodate the data flows generated by the IoT devices. The network service provider owns the controller and manages the network, accordingly. In this work, we consider that multiple network service providers are present in the network, i.e., multi-tenant SDN, and each network service provider owns a separate controller. Hence, the network resources such the flow-table and network bandwidth need to be distributed among the network service providers. Thereby, flow-table partitioning has proven to be one of the important aspects in multi-tenant SDN.

In the existing literature, researchers proposed a few schemes and architectures for flow-table partitioning in SDN, viz., [3]–[5]. Additionally, the researchers studied data traffic management, viz., [2], [6]–[8], and topology management, viz., [7], [9], [10]. In the existing literature, the flow table partitioning schemes are divided into two types — hard and soft partitioning. In hard flow-table partitioning, the controllers decide to share segments of the flow-tables. Therefore, the flow-rule replacement gets confined to the flow-table resources available to each controller. On the other hand, in soft flow-table partitioning, the controllers ensure soft-bounded resources, but access to other parts of the flow-table is not restricted. Though soft flow-table partitioning enhances the performance of the multi-tenant SDN, it introduces another entity into the SDN — *proxy controller* [4]. Therefore, in soft flow-table partitioning, the controllers send the flow-rule replacement requests to the proxy controller, thereby increasing the overhead in the network and delay in flow-rule replacement. Thereby, the proxy controller acts as the third party controller, that controls the flows in the network. However, we argue that, by removing the proxy controller in multi-tenant SDN, we can enhance the network performance significantly, which necessitates the design of a flow-table partitioning scheme in *distributed* multi-tenant SDN without involving the proxy controller.

In this paper, we introduce a utility game theory-based flow-table partitioning scheme, named BIND, for maximizing the network sustainability and minimizing the network overhead and delay in the distributed multi-tenant SDN. We use utility game to decide the flow-rules to be replaced while ensuring fairness among the controllers. In order to ensure fairness among the multi-tenant controllers, we use a blockchain-based flow-table partitioning. In BIND, we consider that the flow-tables are virtually owned by each controller. Hence, to replace any flow-rule, the controllers need to decide the flow-rule unanimously, in which blockchain plays a key role. In BIND, on receiving a Packet-In message, the controller checks for the free flow-table space, and installs the flow-rule if ternary content-addressable memory (TCAM) is available. Otherwise, it requests other controllers to elect a subset of flow-rules which are eligible to be replaced. Thereafter, the controller selects a flow-rule having the highest payoff in the proposed

utility game-based scheme. In BIND, we also introduce the *flow-priority*, which helps in ensuring network sustainability. In summary, the specific contributions of this paper are as follows:

1) We present the blockchain-based flow-table partitioning scheme, BIND, for minimizing network overhead and delay in distributed multi-tenant SDN.
2) We use a utility game-based flow-table partitioning scheme for distributed multi-tenant SDN, and ensure network sustainability.
3) We propose two algorithms such as flow-rule election (FLE) and flow-rule replacement (FRR). Using the FLE algorithm, each controller evaluates the replacement eligibility factor for each flow-rules and elects at most one flow-rule. Thereafter, using the FRR algorithm, the controller decides which flow-rule to be replaced while ensuring that it does not rise to any monopoly situation.

II. Related Work

In the existing literature, several works focused on the different aspects of SDN. Some of the works, viz. [11]–[13], modeled the performance of SDN, theoretically, whereas some works focus on the data traffic and flow-table management in SDN switches, which are discussed here. Blenk *et al.* [3] provide a survey of the multi-tenant SDN and divide the existing literature into two categories such as hard and soft partitioning of flow-tables. In another work, Caria *et al.* [5] proposed an SDN topology partitioning scheme by introducing SDN border nodes, which are responsible for ensuring connectivity among two controllers. The authors considered that each controller has access to a mutually exclusive set of SDN switches. Lin *et al.* [4] proposed a scheme for multi-tenant SDN while considering the presence of the proxy controller. Meiners *et al.* [8] and Congdon *et al.* [6] proposed to reduce the flow-table usage by reducing the number of flow-rules. On the other hand, Mogul *et al.* [14] and Singh *et al.* [15] studied a hash-based flow-table to reduce the flow-table lookups. Reitblatt *et al.* [16] and Maity *et al.* [17] focused on the flow-table update to reduce the delay incurred for network update. In another work, Katta *et al.* [7] focused on reducing the overhead in rule-space. On the other hand, Jarschel *et al.* [11] proposed a performance evaluation model for OpenFlow hardware based on their measured switching times and obtained expressions for different performance parameters. Metter *et al.* [13] focused on the optimal table occupancy and signaling rate for improved network performance. Aujla *et al.* [18] proposed a traffic flow management scheme in SDN. On the other hand, Mondal *et al.* [19] studied the performance of SDN switches using queuing models and evaluated the optimal buffer size for ensuring minimum packet waiting time.

Synthesis: We infer that there exist a few research works on data traffic management and flow-table partitioning in SDN while utilizing TCAM of the switches, efficiently. However, no scheme is proposed to deal with the problem of soft flow-table partitioning in the presence of multi-tenant SDN, where the controllers are distributed in nature. Additionally, throughput-optimal flow-table partitioning in multi-tenant SDN, while minimizing the flow-rule replacement and maximizing network sustainability, is a pressing issue which needs to be addressed.

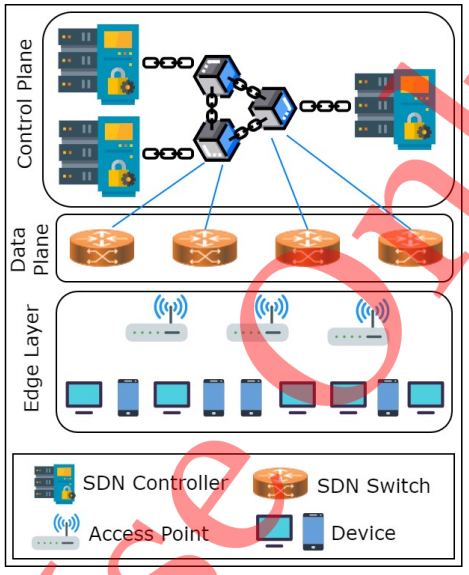

Fig. 1. Schematic Diagram of Multi-Tenant SDN

III. System Model

In this work, we consider a distributed multi-tenant SDN comprising of multiple SDN-controllers and multiple SDN-switches. Each switch $s \in \mathcal{S}$, where $\mathcal{S}$ represents a set of SDN switches, is connected with a set $\mathcal{C}$ of SDN controllers, as shown in Figure 1. Each switch $s \in \mathcal{S}$ has a flow-rule capacity of ρ_s. We consider that the controllers have access to the $\mathcal{S}$ set of switches, and can change the flow-rules of the shared flow-tables as per requirement. The controllers use *permissioned blockchain network* [20] to record and verify the changes in the flow-tables. We consider that, after receiving a Packet-In message, the concerned controller initiates a transaction having flow replacement information. The transactions are digitally signed. The controller generates a block for each transaction and adds it to the blockchain. We consider that each IoT device $n \in \mathcal{N}_c$ registers to controller $c \in \mathcal{C}$ for data transmission, where $\mathcal{N}_c$ denotes a set of IoT devices connected with controller c. Within a duration of Δ, each IoT device n generates $F_n(\Delta)$ set of flows and each controller c receives $F_c(\Delta)$ set of Packet-In messages. Therefore, we get that:

$$F_c(\Delta) \subseteq \bigcup_{n \in \mathcal{N}_c} F_n(\Delta) \tag{1}$$

Additionally, we consider that each controller c sets the priority for each flow-rule based on the type, i.e., mice or elephant flow, and the content of the flow[1], where $\mathbb{P}$ represents

[1]We consider that the controller identifies the content of the flow based on the received meta-data along with Packet-In message.

the set of priorities of the flows. The switches have limited TCAM space, hence, in a dynamic network, the flow-rules may need to be replaced frequently. However, the priority η_f of each flow $f \in \bigcup_c F_c(\cdot)$ ensures the sustainability of the network, as defined in Definition 1. We consider a flow to be an active flow if it has transmitted data within δ duration [1].

***Definition** 1:* We define the sustainability of the network, ζ, as the percentage of flows not blocked by the flows having low priority. In other words, the sustainability of the network varies proportionally with the number active flow with high priority which are not interrupted by any flow with low priority. Mathematically,

$$\zeta = 1 - \frac{\left|\left\{f_i \in \bigcup_c F_c(\cdot) | (x_{f_i} = 1, x_{f_j} = 0, \eta_{f_i} < \eta_{f_j})\right\}\right|}{\left|\left\{f_i \in \bigcup_c F_c(\cdot)\right\}\right|} \quad (2)$$

where x_f is binary variable and denotes the presence of the flow-rule in flow-table corresponding to active flow f. Therefore, we have:

$$x_f = \begin{cases} 1, \text{if rule for active flow } f \text{ is installed in flow-table} \\ 0, \text{otherwise} \end{cases} \quad (3)$$

***Problem Scenario**:* We consider a distributed multi-tenant SDN in the absence of a centralized controller. Multiple controllers share the same flow-space among themselves. We consider that the controller follows the *soft* flow-table partitioning in the proposed SDN. Hence, it may lead to a monopoly situation, where a subset of controllers handle a huge number of flows and modify the flow-rules accordingly. To avoid such a scenario, in this work, we aim to propose a blockchain-based flow-table partitioning scheme for distributed multi-tenant SDNs in the absence of a centralized SDN-controller.

***Assumptions**:* To design the scheme for blockchain-based flow partitioning in distributed multi-tenant SDNs, we consider the following assumptions:

1) There is no centralized SDN-controller to coordinate in multi-tenant SDN.
2) The SDN-controllers do not misbehave and cooperate.
3) The controllers are capable of ensuring the validity of the metadata of Packet-In messages.

IV. BIND: The Proposed Blockchain-Based Flow-Table Partitioning Scheme

To design the interactions among the SDN-switches at the data plane and the SDN-controllers at the control panel, we use a utility game [21]. This is a cooperative game, where the SDN-controllers act cooperatively in sharing the space of the flow-tables while ensuring the sustainability of the network, i.e., the high priority flows are not blocked by the low priority flows. Therefore, in the proposed scheme, BIND, the controllers are the players. On the other hand, the flow-space or the flow-tables are considered to be resources, which need to be shared among the player optimally, while ensuring the sustainability of the network. Moreover, the controllers need to ensure that the optimal network throughput is achieved.

A. Justification for Using Utility Game

For efficient flow-table partitioning, we need to ensure that the monopoly is not present among the controllers. Additionally, the controllers need to ensure network sustainability with optimal throughput. Due to the absence of any centralized coordinator, the controllers decide a subset of flow-rules which can be replaced. Thereafter, they decide which flow-rule to be replaced using cooperation. The utility game plays an important role in the aforementioned process. The actions of the controllers get registered in the blockchain, whenever there is a change in the flow-table. To generate the payoff values of the utility functions, the controllers refer to the previous blocks in the blockchain. Thereby, we argue that the utility game along with the blockchain among the controllers ensures that the controllers get a fair chance to update the flow-table. Additionally, high network sustainability is ensured.

B. Game formulation

To achieve optimal flow-table partitioning, we use utility game. In the proposed blockchain-based flow-table partitioning scheme, named BIND, the controllers act as the players and choose strategies, i.e., flow-rule replacement, for ensuring network sustainability, while ensuring high network throughput. The controllers use the blockchain to record the flow-rule updates as blocks in the multi-tenant SDN. In the absence of a centralized coordinator, the controllers use cooperation to ensure the fair chance in performing the flow-rule update. Thereby, using blockchain and utility game theory, the proposed scheme, BIND, ensures to avoid monopoly among the controllers, which is a significant drawback of the soft flow-table partitioning. In BIND, each block in the blockchain encapsulates the following information:

1) *Controller-Specific Flow-Rule Replacement Counter:* Each controller $c \in \mathcal{C}$ increments the flow-rule replacement counter, $\mathbb{C}_c$, and append the information in the corresponding block.
2) *Total Number of Flow-Rule Replacement:* It represents the total number of flow-rule replacement $\mathbb{R}$ occurred after the last network update. This counter is considered to be a *global* counter.
3) *Tolerable Waiting Time:* The controllers update the tolerable waiting time $\mathbb{T}_f$ for each flow-rule f, when there is a change in the flow-table. The controllers follow an O(K) Markov predictor to update $\mathbb{T}_f, \forall f$ [22], where K is constant.
4) *Elapsed Time:* The controllers update the elapsed time after recent use, t_f, for each flow f, which signifies the information of the least recently used flow-rules.
5) *Flow-Rule Priority:* The priority η_f of the newly installed flow-rule f is considered to be the same as the installed flow-rule priority. The controllers refer to this parameter

to identify the subset of flow-rules that are eligible for replacement in the flow-tables.

C. Replacement Eligibility Factor for Each Flow-Rules

Each controller c calculates the payoff of each flow-rule $f \in F_c(\cdot)$ while considering the two parameters — tolerable waiting time $\mathbb{T}_f$ and elapsed time t_f. Based on these parameters, the controllers calculates the *eligibility* of the flow-rules to be replaced. We define the *replacement-eligibility factor* $\mathbb{E}_f$ for each flow-rule f in terms of probability, as mentioned in Definition 2.

***Definition** 2:* Replacement-eligibility factor $\mathbb{E}_f$ of each flow-rule f as the ratio of the predicted duration before receiving the next packet for flow-rule f and tolerable waiting time (predicted). Therefore, we get:

$$\mathbb{E}_f = \frac{\mathbb{T}_f - t_f}{\mathbb{T}_f} \tag{4}$$

Therefore, we argue that in BIND, the flow-rule f with high replacement-eligibility factor has higher probability to get replaced by the flow-rule associated with the new incoming flow f_n, while considering the following constraint is satisfied:

$$\eta_f < \eta_{f_n} \tag{5}$$

where η_f and η_{f_n} denote the priorities of the flow-rules f and f_n, respectively. Each controller elects a flow-rule having maximum $\mathbb{E}_f$ value which may be replaced.

D. Utility Function of Each Controller

Based on the elected flow-rule, each controller c calculates the payoff of utility function $\mathcal{U}_c(\cdot)$. The utility function $\mathcal{U}_c(\cdot)$ signifies the probability of the flow-rule to replaced associated with controller c. In order to design the utility function $\mathcal{U}_c(\cdot)$, the controllers consider the parameters such as flow-rule replacement counter $\mathbb{C}_c$, the total number of flow-rule replacement $\mathbb{R}$, and replacement-eligibility factor $\mathbb{E}_f$ of the elected flow f. Additionally, the controllers consider the parameter — change in throughput ϕ_f — by replacing the flow-rule f. The utility function $\mathcal{U}_c(\cdot)$ needs to satisfy the following properties:

1) Flow-rules associated with each controller are treated with fairness and monopoly does not occur. Therefore, we consider that:
$$\frac{\partial \mathcal{U}_c(\cdot)}{\partial \mathbb{C}_c} < 0 \tag{6}$$
2) We aim to reduce the number of replacement in the flow-tables. Therefore, we consider that utility function $\mathcal{U}_c(\cdot)$ needs to follows the following constraint:
$$\frac{\partial \mathcal{U}_c(\cdot)}{\partial \mathbb{E}_f} > 0 \tag{7}$$
3) We aim to increase the overall network throughput for ensuring high bandwidth utilization. Therefore, we consider that $\mathcal{U}_c(\cdot)$ has high utility value if the change in the throughput is high.
$$\frac{\partial \mathcal{U}_c(\cdot)}{\partial \mathbb{D}_f} > 0 \tag{8}$$
where $\mathbb{D}_f$ denotes the change in throughput for replacing flow-rule f.

Therefore, we define the utility function $\mathcal{U}_c(\cdot)$ as follows:

$$\mathcal{U}_c(\cdot) = \mathbb{E}_f \mathbb{D}_f \left(1 - \frac{\mathbb{C}_c}{\mathbb{R}}\right) \tag{9}$$

where $f \in F_c$. The controllers aim to maximize the payoff value of the utility function $\mathcal{U}_c(\cdot)$ to ensure network sustainability and throughput.

E. Proposed Algorithms

If flow-rule space is available in the flow-table, the controller that received the Packet-In message installs the flow-rule and generates a block, accordingly. However, in case of no flow-rule space availability, the controllers act cooperatively and decide the flow-rule to be replaced while ensuring the enhanced performance of the network with high network sustainability. To achieve the aforementioned goal in BIND, we propose two algorithms — Flow-Rule Election (FLE) and Flow-Rule Replacement (FRR). In BIND, once a Packet-In message is received by a controller, each controller executes the FLE algorithm (Algorithm 1), distributively. Using the FLE algorithm, the controllers in the multi-tenant SDN elect a subset of flow-rules which are eligible to be replaced. Initially, using the FLE algorithm, each controller selects a subset of flow-rules based on flow-priority, i.e., satisfy the constraint in Equation (5), (refer to Line 2). Using Algorithm 1, each controller aims to elect *at most one* flow-rule having maximum replacement eligibility factor.

Thereafter, the controller, that received the Packet-In message, executes the FRR algorithm (Algorithm 2) and decides if the new flow-rule is to be installed or discarded. Additionally, using the FRR algorithm, the controller decides which flow-rule is to be replaced to install the new flow-rule. Using the FRR algorithm (Lines 4-5), the controller calculates the utility function, i.e., the probability of elected flow-rules to be replaced by the new flow-rule. Accordingly, the controller generates a block and adds it to the blockchain.

Complexity Analysis: In BIND, we take advantage of the distributed architecture of blockchain for designing a scheme for flow-table partitioning in the multi-tenant SDN. As mentioned earlier, the proposed scheme, BIND, has two components — flow-rule election and flow-rule replacement. We observe that the time complexity of Lines 2–5 and 10–12 (Algorithm 1) is $O(|F_c|)$. Therefore, we argue that the time complexity for the FLE algorithm in BIND $O(|F_c|)$ for controller c. Accordingly, we get that, in BIND, the time complexity of the FLE algorithm for the overall network is $O(\max|F_c|)$. On the other hand, we observe that the time complexity of Lines 2-8 (Algorithm 2) is $O(|\mathcal{C}|)$. Therefore, the time complexity of the FRR algorithm is $O(|\mathcal{C}|)$. Hence, the overall complexity of the proposed scheme, BIND, is $O(|\mathcal{C}| + \max|F_c|)$.

Algorithm 1 FLE: Flow-Rule Election in BIND

INPUTS:

F_c ▷ Set of flow-rules maintained by controller c
f_n ▷ New flow-rule for received Packet-In message
$\eta_f, t_f, \mathbb{T}_f \forall f \in F_c$
η_{f_n}

OUTPUTS:

$f_c \in F_c$ ▷ Elected flow-rule by controller c
$\mathbb{E}_{f_c}$ ▷ Replacement eligibility factor of elected flow-rule

PROCEDURE:

$\mathcal{E}_c \leftarrow \{\emptyset\}$
for Each $f \in F_c$ **do**
  if $\eta_f < \eta_{f_n}$ **then**
    Calculate $\mathbb{E}_f$ using Equation (4)
    $\mathcal{E}_c \leftarrow \mathcal{E}_c \bigcup \mathbb{E}_f$
  end if
end for
if $\mathcal{E}_c \neq \{\emptyset\}$ **then**
  Select flow-rule f_c such that $\mathbb{E}_{f_c} \geq \mathbb{E}_f$ where $f \neq f_c$ and $f, f_c \in F_c$
  for Each $f \in F_c / \{f_c\}$ **do**
    Update $\mathbb{T}_f$ using Markov predictor [22]
  end for
  return $\{f_c, \mathbb{E}_{f_c}\}$
else
  return $\{NULL, NULL\}$
end if

Algorithm 2 FRR: Flow-Rule Replacement in BIND

INPUTS:

$f_c, \mathbb{E}_{f_c}, \forall c \in \mathcal{C}$ ▷ Outputs from FLE Algorithm
$\mathbb{C}_c, \forall c \in \mathcal{C}$ ▷ Controller-specific flow-rule replacement counter
$d_{f_c} \forall c \in \mathcal{C}$ ▷ Throughput of the selected flow-rules
f_n ▷ New flow-rule for received Packet-In message
d_{f_c} ▷ Throughput of the new flow-rule f_n
$\mathbb{R}$ ▷ Total number of flow-replacement

OUTPUTS:

x_{f_n} ▷ Presence of new flow-rule in the flow-tables
f^* ▷ Flow-rule replaced by flow-rule f_n

PROCEDURE:

$V \leftarrow \{\emptyset\}$
for Each $c \in \mathcal{C}$ **do**
  if $f_c \neq NULL$ **then**
    $\mathbb{D}_{f_c} \leftarrow d_{f_n} - d_{f_c}$
    Calculate $\mathcal{U}_c(\cdot)$ using Equation (9)
    $V \leftarrow V \bigcup \mathcal{U}_c(\cdot)$
  end if
end for
if $V == \{\emptyset\}$ **then**
  return $\{0, NULL\}$
else
  $f^* \leftarrow f_c$ such that $\mathcal{U}_c(\cdot) \geq \mathcal{U}_{c'}(\cdot)$ where $c \neq c'$ and $c, c' \in \mathcal{C}$
  return $\{1, f^*\}$
end if

Moreover, the space complexity of the FLE algorithm in BIND is $O(|\max F_c|)$, where the space complexity for each controller c is $O(|F_c|)$. On the other hand, the space complexity for the FRR algorithm in BIND is $O(|\mathcal{C}|)$. Therefore, similar to the time complexity, we observe that the space complexity of the proposed scheme, BIND, is $O(|\mathcal{C}| + \max |F_c|)$.

V. Performance Evaluation

We evaluate the performance of the proposed scheme, BIND, with a varying number of flows in multi-tenant SDN.

A. Simulation Parameters

We simulate the proposed scheme while considering the number of SDN switches and the controllers to be 20 and 5, respectively. We varied the number of flows as mentioned in Table I. We assume that each flow are homogeneous with data rate of 0.2 million packets per second ($mpps$) [1].

TABLE I
Simulation Parameters

Parameter	Value
Simulation area	1000 m×1000 m
Number of controllers	5
Number of switches	20
Number of flows	1000, 2000, 3000
Flow priorities	1-5
Data-rate requirement per flow	50-100 kbps
Maximum data-rate per switch	10^3 kbps

B. Benchmarks

The performance of the proposed scheme, BIND, is evaluated by comparing with the two schemes — hard flow-table partitioning (HARD) and soft flow-table partitioning in the presence of proxy controller (PROXY). In HARD, we consider that controllers have non-overlapping and restricted flow-table access. On the other hand, in PROXY, we consider that the controllers use soft flow-table partitioning in the presence of proxy controller.

We evaluated the performance of the proposed scheme, BIND, using the metrics such as network delay, network throughput and network sustainability, as discussed below:

C. Results and discussions

Network Delay: In network delay, we considered the flow-set up delay. From Figure 2(a), we observe that with the increase in the number of flows, the flow setup delay reduces significantly using BIND, due to the distributed nature. However, in HARD and PROXY, the flow setup is a centralized approach. Hence, we observe that, in HARD and PROXY, the delay increases in polynomial curve, whereas the delay using BIND increases linearly.

Network Throughput: We observe that using BIND, the achieved throughput is higher as compared to HARD and PROXY, as shown in Figure 2(b). We argue that, in HARD and PROXY, the flow-rule gets replaced based on the time-stamp only, however in BIND, we ensure that the flow-rules having high throughput get priority in flow-rule replacement.

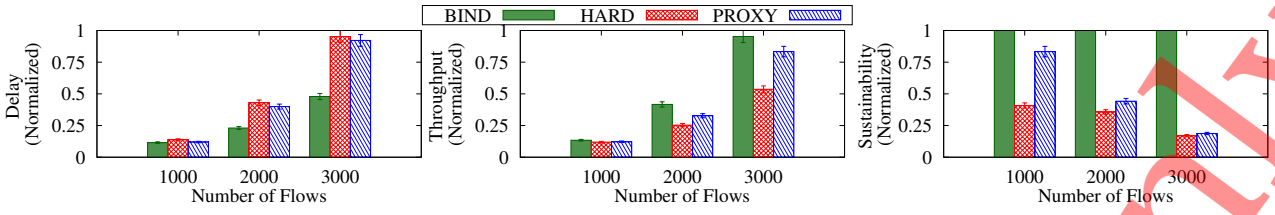

Fig. 2. Comparison of BIND with Other Schemes

Network Sustainability: From Figure 2(c), we observe that BIND ensures 100% network sustainability, whereas using HARD and PROXY, the network sustainability decreases with the increase in the number of flows. In BIND, while evaluating the eligibility factor of the flow-rules, we ensure that the high priority active flows are not replaced by the low priority flows, which results in high network sustainability.

VI. Conclusion

In this paper, we proposed BIND, a blockchain-based flow-table partitioning scheme, named BIND, for distributed multi-tenant SDN. Using blockchain, we ensure that the controllers are synchronized and cooperative in nature. In BIND, we used utility game to propose the distributed algorithm for flow-rule election, where each controller distributively identifies the flow-rules' replacement eligibility factors, and elects a single flow-rule for replacement. Thereafter, we considered a utility game-based centralized algorithm for flow-rule replacement to be performed by the controller receiving the Packet-In message. We observe that BIND ensures fairness in flow-rule replacement for the controllers in a distributed multi-tenant SDN. Through simulation, we observed that the flow setup delay increases linearly using BIND. The proposed scheme, BIND, also ensures high throughput and network sustainability, thereby outperforming the benchmark schemes – HARD and PROXY.

Future extension of this work includes the understanding of flow-table aggregation among the controllers in a distributed multi-tenant SDN. Additionally, this work can be extended while considering that the controllers are non-cooperative in nature.

Acknowledgement

The authors acknowledge the funding support received from INAE (Grant no. INAE/121/AKF, Dt. 13-02-2019), and SERB/IMPRINT-II (Grant no SERB/F/12680/2018-2019;IMP/2018/000451, Dt. 25-03-2019) for executing parts of this work.

References

[1] A. Mondal, S. Misra, and A. Chakraborty, "TROD: Throughput-Optimal Dynamic Data Traffic Management in Software-Defined Networks," in *Proc. of IEEE Globecom Workshops (GC Wkshps)*, Dec 2018, pp. 1–6.

[2] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," in *Proc. of the IEEE*, vol. 103, no. 1, January 2015, pp. 14–76.

[3] A. Blenk, A. Basta, M. Reisslein, and W. Kellerer, "Survey on network virtualization hypervisors for software defined networking," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 655–685, Firstquarter 2016.

[4] Y. Lin, T. Liu, J. Chen, and Y. Lai, "Soft partitioning flow tables for virtual networking in multi-tenant software defined networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 402–415, March 2018.

[5] M. Caria, A. Jukan, and M. Hoffmann, "Sdn partitioning: A centralized control plane for distributed routing protocols," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 381–393, Sep. 2016.

[6] P. T. Congdon, P. Mohapatra, M. Farrens, and V. Akella, "Simultaneously Reducing Latency and Power Consumption in OpenFlow Switches," *IEEE/ACM Trans. on Net.*, vol. 22, no. 3, pp. 1007–1020, June 2014.

[7] N. P. Katta, J. Rexford, and D. Walker, "Incremental Consistent Updates," in *Proc. of ACM SIGCOMM Wrkshp*. New York, NY, USA: ACM, 2013, pp. 49–54.

[8] C. R. Meiners, A. X. Liu, and E. Torng, "Bit Weaving: A Non-Prefix Approach to Compressing Packet Classifiers in TCAMs," *IEEE/ACM Trans. on Net.*, vol. 20, no. 2, pp. 488–500, April 2012.

[9] F. Li, J. Cao, X. Wang, Y. Sun, T. Pan, and X. Liu, "Adopting SDN Switch Buffer: Benefits Analysis and Mechanism Design," in *Proc. of IEEE ICDCS*, Jun 2017, pp. 2171–2176.

[10] S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat, "Soft-WSN: Software-Defined WSN Management System for IoT Applications," *IEEE Syst. J.*, pp. 1–8, 2016, DOI:10.1109/JSYST.2016.2615761.

[11] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia, "Modeling and Performance Evaluation of an OpenFlow Architecture," in *Proc. of Int. Tel. Cong.* International Teletraffic Congress, 2011, pp. 1–7.

[12] A. Mondal, S. Misra, and I. Maity, "Buffer Size Evaluation of OpenFlow Systems in Software-Defined Networks," *IEEE Syst. J.*, pp. 1–8, 2018.

[13] C. Metter, M. Seufert, F. Wamser, T. Zinner, and P. Tran-Gia, "Analytical Model for SDN Signaling Traffic and Flow Table Occupancy and Its Application for Various Types of Traffic," *IEEE Trans. on Net. and Ser. Man.*, vol. 14, no. 3, pp. 603–615, Sep 2017.

[14] J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, A. R. Curtis, and S. Banerjee, "DevoFlow: Cost-effective Flow Management for High Performance Enterprise Networks," in *Proc. of ACM SIGCOMM Wrksp*, New York, NY, USA, 2010, pp. 1–6.

[15] A. Singh, S. Batra, G. S. S. Aujla, N. Kumar, and L. T. Yang, "BloomStore: Dynamic Bloom Filter-based Secure Rule-Space Management Scheme in SDN," *IEEE Transactions on Industrial Informatics*, pp. 1–11, 2020, DOI: 10.1109/TII.2020.2966708.

[16] M. Reitblatt, N. Foster, J. Rexford, C. Schlesinger, and D. Walker, "Abstractions for Network Update," in *Proc. of ACM SIGCOMM*, New York, NY, USA, 2012, pp. 323–334.

[17] I. Maity, A. Mondal, S. Misra, and C. Mandal, "CURE: Consistent Update with Redundancy Reduction in SDN," *IEEE Transactions on Communications*, vol. PP, no. 99, pp. 1–8, 2018.

[18] G. S. Aujla, R. Chaudhary, N. Kumar, R. Kumar, and J. J. P. C. Rodrigues, "An Ensembled Scheme for QoS-Aware Traffic Flow Management in Software Defined Networks," in *Proc. of IEEE Int. Conf. on Comm.*, May 2018, pp. 1–7.

[19] A. Mondal, S. Misra, and I. Maity, "Buffer Size Evaluation of OpenFlow Systems in Software-Defined Networks," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–8, 2018.

[20] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[21] A. Chakraborty, A. Mondal, and S. Misra, "Cache-Enabled Sensor-Cloud: The Economic Facet," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, Apr 2018, pp. 1–6.

[22] D. Joseph and D. Grunwald, "Prefetching using Markov Predictors," *IEEE Transactions on Computers*, vol. 48, no. 2, pp. 121–133, Feb 1999.