

Article

Integration of Interdomain Flow-Rule in Software-Defined Networks for Embedded Internet of Things Devices

Sabih Khan Afridi ¹, Saleem Iqbal ², Kashif Naseer Qureshi ³, Saqib Majeed ¹ and Gwanggil Jeon ^{4,5,*}¹ University Institute of Information Technology, UIIT, PMAS-AAUR, Rawalpindi 46000, Pakistan² Department of Computer Science, Allama Iqbal Open University, H-8, Islamabad 44000, Pakistan³ Department of Electronic and Computer Engineering, University of Limerick, V94 T9PX Limerick, Ireland⁴ Department of Embedded Systems Engineering, Incheon National University, Incheon 22012, Republic of Korea⁵ Energy Excellence & Smart City Lab., Incheon National University, Incheon 22012, Republic of Korea

* Correspondence: gjeon@inu.ac.kr

Abstract: Software-defined networking (SDN) is an evolving technology providing proper segregation between the control part and data-forwarding domain of network devices. The expansion of the Internet of Things (IoTs) and embedded mobile devices increases the volume of traffic at the network backbone and causes processing costs in the control plane. This directly affects the Ternary Content Addressable Memory (TCAM) of the switches because insufficient space makes it more challenging to manage the flow-entries. In this situation, providing services to specific users who newly authenticate after the successful handoff from the previous SDN domain is challenging. This paper proposes a method for implanting the users' primary domain's flow-rules in the serving SDN domain. As the TCAM is already suffering from a short space, it is hard to handle the flow-tables of multiple SDN domains in limited TCAM storage. The SDN-based Integration of the Interdomain Flow-rule in the SDN (IIF-SDN) scheme maximizes the proficiency of the switches by effectively storing flow-table and flow-entries. The effectiveness of the proposed scheme is benchmarked with proactive and reactive SDN approaches.

Keywords: software-defined networking (SDN); OpenFlow; flow-table; flow-entry; rule management



Citation: Afridi, S.K.; Iqbal, S.; Qureshi, K.N.; Majeed, S.; Jeon, G. Integration of Interdomain Flow-Rule in Software-Defined Networks for Embedded Internet of Things Devices. *Electronics* **2023**, *12*, 1172. <https://doi.org/10.3390/electronics12051172>

Academic Editor: Martin Reisslein

Received: 4 January 2023

Revised: 19 February 2023

Accepted: 23 February 2023

Published: 28 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The rapid development of online applications, cloud computing, multimedia applications, and big data analytics demands revolutionary steps for flexible, agile, cost-effective, and specific service provisioning related to individuals [1,2]. The concept of virtualization in the network enhances the functionality of the network, such as network function virtualization (NFV) and software-defined networking (SDN) [3,4]. SDN—which is divided into two parts: the control plane and the data plane—is emerging as a key enabler in the networking field, and it aids in packet transmission between network devices and the IoT. The SDN controller remote virtual machine is placed anywhere in the network to manage policy functions, routing-related jobs, and the whole background processes. The SDN itself accomplishes the run-time configuration and programmatic behavior moderation. In SDN, the packet transmission rules are stored and managed in a flow-table inside the switch. Various schemes have been developed for managing flow information for the improvement of the ability of the network [5,6]. Figure 1 demonstrates the flow-table and the flow-entry fields stored inside the flow-table.

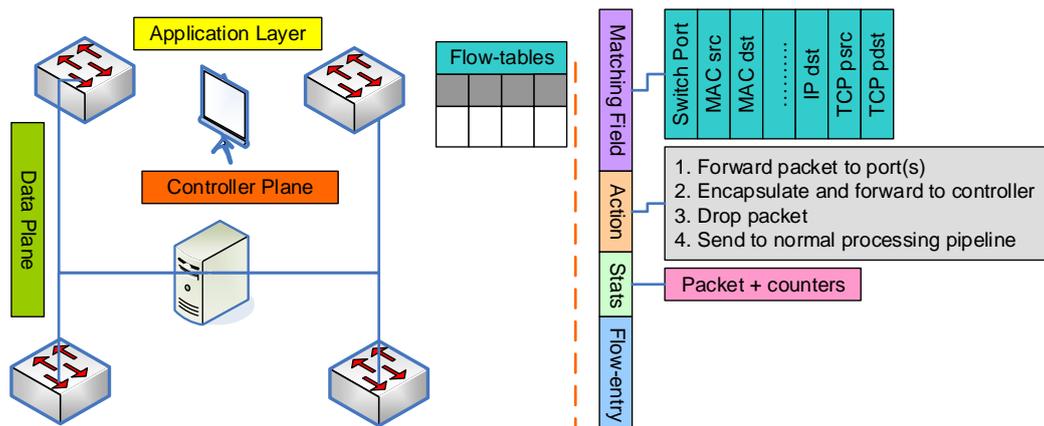


Figure 1. Flow-table and flow-entry structure.

Flow-entry is an element of the flow-table to establish the matching-field, action-field, and stats-field. The matching-field contains packet information, and the action-field holds information about what actions apply when flow-entry matches. The stats-field stores the information about how much time the packet matched [1–3]. The OpenFlow switch contains a flow-table, usually implemented with the technology of TCAM. The flow-table is based on rules, access control lists (ACLs), and the IP route table. The SDN controller randomly checks the switches and updates the flow-table by installing new strategies.

There are two approaches for updating the flow-table in SDN: (i) reactive and (ii) proactive; in the reactive approach, the SDN controller does not install the flow-table when the network operation is launched. Instead, the SDN controller sets up rules in the flow-table as packets arrive at the switch during the network operation. On the other hand, the flow-rules are installed into the flow-table in advance in a proactive approach as a network operation begins. The proactive approach can reduce the overhead in the communication between the controllers and switches [4–6]. Some more degradation issues related to the SDN performance are (i) topology changes, (ii) network reconfiguration, and (iii) re-routing rules creation. The most frequent challenge for the SDN controller is upgrading the switches rapidly and consistently to maintain the quality of service (QoS) [7,8]. Figure 2 shows the taxonomy of SDN performance challenges.

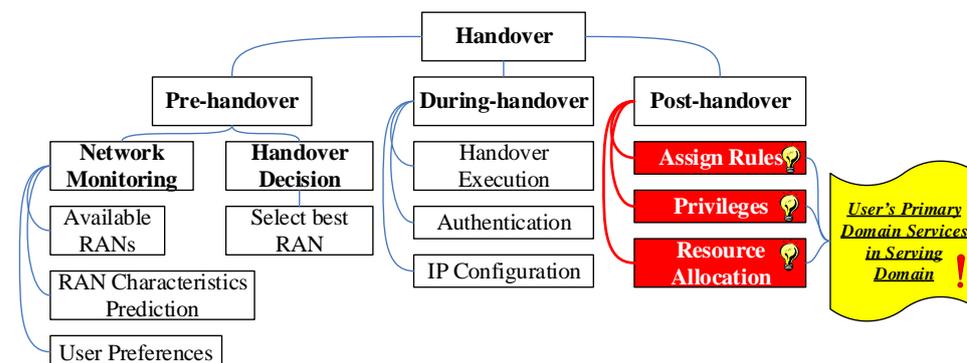


Figure 2. Handover challenges.

The key finding of this research is (i) hybrid flow-rule integration inside the SDN switches, (ii) packet transmission without any anomaly, (iii) hybrid flow-rule execution, (iv) optimized flow entries, and the update process itself. Therefore, the network data can be forwarded generally in the uninterrupted update scenario. Furthermore, the proposed IIF-SDN scheme is applied on account of a rapid response between the controller and switches for services to the continuously connecting users of different SDN domains, which

is highly significant to maintain and incorporate the hybrid flow-rule in TCAM and ACLs of switches.

The rest of the paper is organized as follows: The background knowledge described in Section 2 briefly enlightens the operations in SDN networks and details of the flow-table and flow-entry management. This is followed by explaining the SDN-based IIF scheme and its function in Section 3. The Section 3 subsections describe the complete performance evaluation of IIF according to the defined performance metrics. Finally, Section 5 is the conclusion of the paper.

2. Related Work

The OpenFlow switch in an SDN-based Wi-Fi network is located at the access point, and OpenFlow provides interoperability between the controller and data plane. The centralized control in SDN technologies minimizes many problems, such as handover delay, overload or load balancing issues, and packet losses as compared to conventional Wi-Fi network technologies. The controller takes all decisions for policy implementation, including packet processing, such as the addition, modification, and deletion of entries in the OpenFlow switch's TCAM [9]. In addition, the OpenFlow switches are used as forwarding elements in the SDN-based Wi-Fi network.

SDN is designed and developed for flow-based network applications and has complex matching patterns for packet transmission. TCAM produces more complexity because of its limited storage size for forwarding rules, and the OpenFlow technology has an essential concern regarding its deployment [10–12]. The solution to this problem is to reduce the demand for TCAM and aggregate the required entries by merging the flow-entries, and preserving its original semantics through wildcard rules. For the fast-updating flow-entries, there are more solutions: (i) delegate control back to the switch, (ii) common flow entries, (iii) cluster allocation in the SDN switch. However, the bandwidth can affect the update flow-rule speed between the controller and switches; for this kind of issue, some researchers proposed a solution to delegate the control back to the switch to release the overloaded SDN controller [13–15]. Another solution proposed in [16] is that the intermediate switch does not need any controller for rule installation if sending and receiving switches update the flow-table by parsing information in the packet header. This scheme helps to release the controller's burden and improve the network's speed.

An SDN-based wireless network (SWN) [17] is the scheme proposed for seamless handover, load-balancing, and unified authentication with the cooperation of the SDN controller. The author used software as a bridge between the mobile node and the access point, namely, the software access point (SAP). Although the researchers of this scheme claim good results of throughput of SAP-based handover, experimental tests are not included in this study.

After the successful handoff, it is necessary to allocate resources to the newly registered mobile user. The user frequently complains that the serving domain cannot provide the same services as the primary domain. Therefore, maintaining the QoS of the primary domain in the serving domain is crucial. SDN-based Heterogeneous Inter-Domain Handoff system for IoT (SDN-HIIoT).

The mobility-aware hybrid flow-rule cache scheme [18] chooses whether to install the flow-rule either proactively or reactively for the target-candidate-forwarding nodes based on a comparison between the delay requirement of the incoming flow and the response delay of the controller. Considering the flow-rule cache constraints, the heuristic technique is used to define and resolve the integer linear programming (ILP) issue to determine the ideal number of proactive flow rules.

SDN-HiIoT proposed in [3] is based on the heterogeneous SDN domains handoff system for mobile devices. This methodology is used to exchange the flow rules by using east-bound and west-bound APIs between two SDN controllers. Here, novel technology experiments are used based on the simulation of two SDN RYU controllers and OpenFlow switches.

The proposed solution was invented to provide the primary SDN domain's quality of services (QoS) to those users who register from any SDN domain in the current SDN domain. SDN-HiIoT is established by the "C" controller and all "M" flow-tables. In this proposed technology, the existing SDN domain obtains the primary domain flow-rules for the newly authenticated user for the provision of quality of services (QoS). The equation before the installation of the obtained flow-rules in the current SDN domain looks like $C(\text{current}) = \sum_{i=1}^m M$, where $i = \{1, 2, 3 \dots m\}$; after installation of the primary domain flow-rules, the equation becomes $C(\text{current}) = \sum_{i=1}^{m+k} M$, where k is considered as an obtained flow-rule of the primary domain. Figure 3 shows the flow-rule exchange.

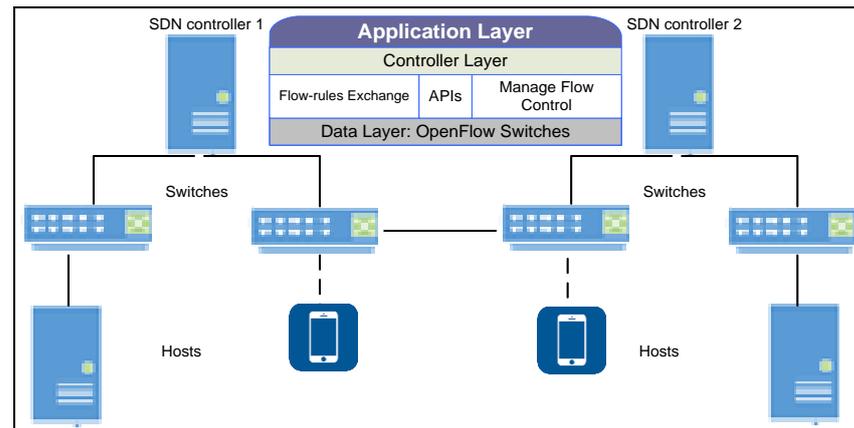


Figure 3. Flow-rule exchanger.

For the management of the flow-table, the Authority Switches scheme is also used to speed up the process of the installation of rules, which requires large-scale modification but is not sure of the OpenFlow protocol specification. An automatic flow-table control technique is proposed in [19] for the OpenFlow switch to solve insufficient flow-table resources. However, this optimization scheme needs to fulfill the requirement of the switch and real network environment. The reduction in the update and time provided some more solutions, for example, caching flow entries, storage structure modification, and some algorithms proposed for the free spacing in TCAM [20–23].

As explained in the above paragraph, the flow-table cannot handle all necessary forwarding rule entries concerning enhancing the efficiency of the flow-table, as illustrated in Figure 4. More studies have been published to manage and control the flow-table rules using the adaptive time-out mechanism [24–28]. Additionally, some other solutions are designed to reduce the flow-table's size by using the aggregation concept. Other solutions focused on splitting the rules and distributing them in the network based on device capacity. Another proposed method is to limit the switch's concurrent forwarding rules and overhead. There are some machine-learning techniques to predict patterns for traffic flow and observe the proper flow being installed in the flow-table of the switch [29–31].

The multi-switching delay is also part of this discussion, which reduces the delay by combing and filtering rules in a global view. The size of the storage of switches varies and is used to constrain flow-entries for the fast updating speed of the flow-table in the entire network.

The table underneath shortly describes the comprehensive research. Table 1 is based, somehow, on the SDN controller placement approach, and the zone represents the area where researchers applied their proposed scheme and obtained the results in the target column of the table.

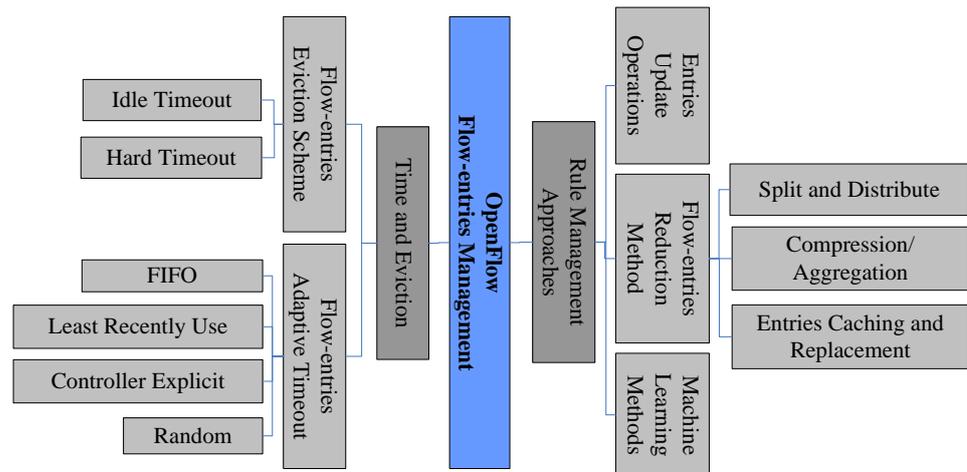


Figure 4. Taxonomy of SDN-switch flow-entry management.

Table 1. Comparison of the improvement of TCAM, flow-table, and flow-entry utilization.

SDN	Controller Placement	Scheme	Target	Zone
[29]	Reactive	Directed acyclic graph used to solve the replacement problem	Under the dependency-constraints replacement problem of TCAM	Distributed ACL
[30]	Reactive	Hidden Markov model using the Quine–McCluskey algorithm	Reduce the flow process time and manage-multiply the flow-table	Flow-table management
[31]	Proactive	“One big switch”	One big switch: using the forwarding element distribute rules	The distributed load balancer and ACL
[28]	Proactive	Split flow-tables into smaller tables and distribute them on the data layer	Minimize the TCAM shortage storage problem by distributing flow-rules	Distributed ACL
[3]	Reactive	SDN-HiIoT	The provision of QoS to mobile users is the same as the primary domain	Heterogeneous SDN domain flow-rule exchange
	Hybrid	IIF-SDN	SDN controllers use proactive, and switches use reactive approaches to reduce lookup efforts and enhance the TCAM functionalities	Fusion of different SDN domain flow-table management

The above work is selected from different journals on SDN-based services and technologies. Researchers have proposed very considerable techniques and methods in various areas of SDN. However, existing schemes do not significantly impact the performance of a single switch and need to resolve the complete problem of update processes. For rapid updating and providing certain services to the specific user, new schemes are required that increase system functionality, provide an optimized solution, and impact the OpenFlow network performance.

3. Proposed Work

This research focuses on making the network dynamic by implanting the different SDN-domain flow rules in the current SDN domain without disturbing the existing users’ flow rules and services. The proposed scheme is developed to provide services to specific users of different SDN-domains. The TCAM was designed for fast lookup and packet matching but is less dense than the RAM and can store fewer bits in the same chip area.

Therefore, it is hard to install multiple SDN-domain flow-rules on a single SDN-domain. We have proposed a scheme for the fast lookup of entries and high-speed flow-rule installations, especially for large-scale networks. It is obligatory to enable the switches to adapt to the new policy and provide quick responses to the newly registered user of different SDN domains. Figure 5 clearly describes how our proposed scheme facilitates the SDN components without adding hardware.

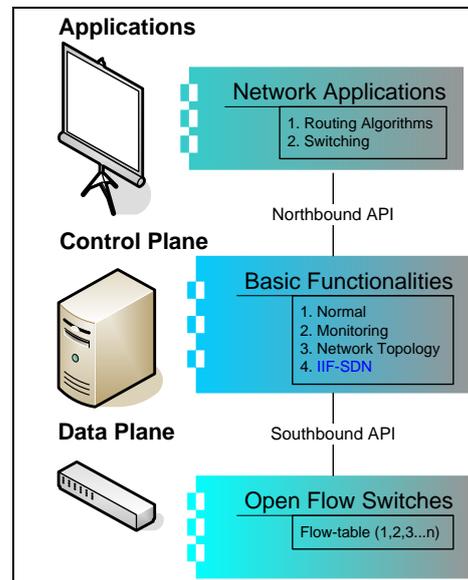


Figure 5. Structure of SDN with the IIF-SDN scheme.

3.1. IIF-SDN Formulation

As already discussed in the introduction of this article, the SDN controller works on two basic concepts—the reactive and proactive approaches—for updating the flow-table in the switch based on the occurrence of certain events in the network. In the case of a reactive approach, the SDN controller does not install any flow-rule in the switch because it waits for the packet influx, then populates the rules accordingly and installs them on the switch [32,33]. On the other hand, the SDN controller installs the flow-rules in advance as the network begins the operation. The selection of the flow-rules has a significant value for enhancing the network performance, especially in large networks, such as data centers. In our proposed IIF-SDN, we have used a proactive approach. In this case, the SDN controller obtained all other SDN domain flow-rules in advance. This technique is used for the enhancement of the efficiency of the network.

In IIF-SDN, the SDN controller stores the flow entries after becoming priority-wise from the other SDN domains; high-priority entries are saved on the top of the list. Then, as the switch is requested for installation in the switch, the controller installs the customized flow-rules list on the switch, considering the priority of each entry). This technique reduces the lookup efforts in ACL and TCAM. Figure 6 shows the IIF-SDN proactive approach on the SDN controller layer.

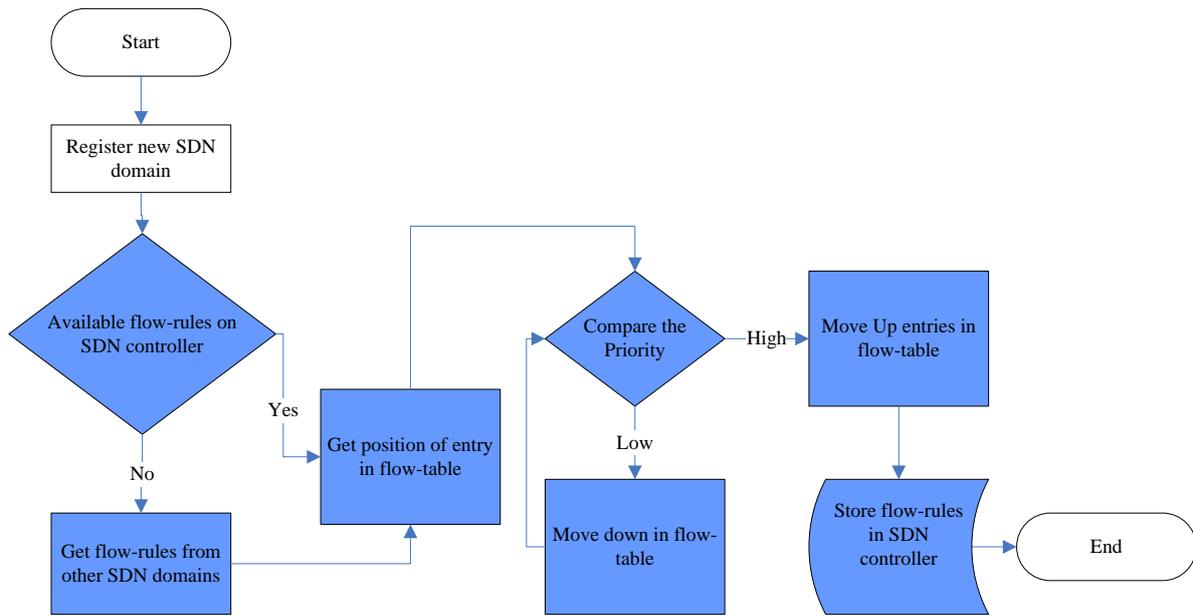


Figure 6. IIF-SDN proactive approach on the SDN controller layer.

3.2. Network Model

The set of SDN domains represented by “ SDN_{dom} ” and all flow-tables of one domain represented by “ β ” are shown in Table 2 and explained in Equations (1) and (2).

$$SDN_{dom} = \sum_{i=1}^n C, \text{ where } i = \{1, 2, 3 \dots n\} \tag{1}$$

$$B = \sum_{j=1}^m frs, \text{ where } j = \{1, 2, 3 \dots m\} \beta \in SDN_{dom} \tag{2}$$

Table 2. System Notations.

Notations	Description
SDN_{dom}	SDN Domain
SDN_C	SDN Controller
SDN_{Sw}	Switches
$SDN_{Sw-TCAM}$	TCAM
B	Flow-table
frs	Flow-rules
μ	Flow-entry
β_c	Controller’s Flow-table
$frs-c$	Controller’s Flow-rules

The new SDN_{dom} is registered in the network of SDN_{doms} ; they share flow-rules with the newly registered SDN_{dom} (as shown in Figure 7). After obtaining frs , the new SDN_C sort “ frs ” on a priority base and organize all entries in the “ β ”, as shown in Equations (3) and (4).

$$\beta_j = frs(1), frs(2), frs(3) \dots frs(m) \tag{3}$$

$$\beta_j = frs(1) \geq frs(2) \geq frs(3) \dots frs(m-1) \geq frs(m) \tag{4}$$

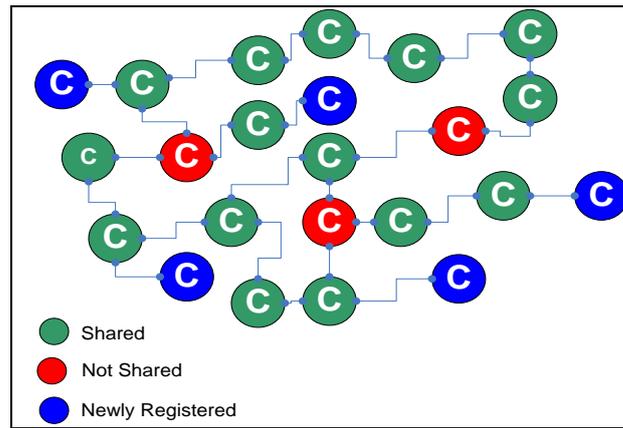


Figure 7. SDN controllers.

After increasing the SDN domains, Equation (1) is extended, as represented below (Equation (5)):

$$SDN_c = \sum_{i=1}^{n+1} C, \text{ where } i = \{C1, C2, C3 \dots Cn, C(n+1), C(n+2), C(n+3) \dots C(n+n)\} \quad (5)$$

Additionally, after increasing the SDN domains, the value of “m” will also increase by “m + 1”; then, Equation (2) of the flow-table is extended, as follows (Equation (6)):

$$B = \{\beta_1, \beta_2, \beta_3 \dots \beta_m, \beta(m+1), \beta(m+2), \beta(m+3) \dots \beta(m+m)\} \quad (6)$$

Initially, calculate the “β” at “S_w” and differentiate the active flow-table β_{act} ∈ SDN_{dom}(B) and inactive flow-tables λ ∈ SDN_{dom}(B) (as shown in Equations (7) and (8)). In this equation, the flow-rule function “f()” has some operation in TCAM for the flow-entries to make them useful for the SDN controller and switches. Equations (9) and (10) make the space by removing “λ” from the SDN_{S_w-TCAM}(B) of the corresponding switch where i, j, and k represent the number of flow-tables, and n, m, and p are the maximum value of the flow-tables.

- Active flow-rules of the SDN_{dom}(B) switch:

$$f(SDN_{S_w-TCAM}(B)) = \sum_{j=1}^m \beta; \text{ where } j = \{1, 2, 3 \dots m\}, \beta \in SDN_{dom}(B) \quad (7)$$

- Inactive flow-rules of the SDN_{dom}(B) switch:

$$f(SDN_{S_w-TCAM}(B)) = \sum_{k=1}^p \lambda; \text{ where } k = \{1, 2, 3 \dots p\}, \beta \in SDN_{dom}(B) \quad (8)$$

For the new rule installation at SDN_{S_w-TCAM}(B), consider Equation (4). β_{act} are the existing rules of SDN_{dom}(B) and β_c are the new rules of the primary domain of the user followed by the SDN_C ∈ SDN_{dom}(B).

$$f(\beta_{act}(B)) = \sum_{j=1}^m \beta - \sum_{k=1}^p \lambda; \text{ where } \beta \in SDN_{dom}(A) \text{ and } \lambda \in SDN_{dom}(B) \quad (9)$$

$$f(SDN_{S_w-TCAM}(B)) = f(\beta_{act}(B)) + \sum_{j=1}^m \beta_c \in SDN_{dom}(A) \quad (10)$$

As discussed above, the “f_{rs}” are acquired from the heterogeneous SDN domains and organized at the controller level. Thus, this scheme is very lightweight for the SDN_{dom} by implanting the flow-rules in advance. The effectiveness and advancement of SDN are shown in the results section of the proposed scheme.

3.3. Approximation Algorithm

Algorithm 1 of the IIF-SDN scheme shows how the newly registered SDN controller will obtain the flow-rules “f_{rs}” from different SDN domains and will store them using a

proactive approach. Furthermore, the IIF-SDN scheme arranges the flow-entries “ μ ” on the controller side, as shown in Algorithm 2. This scheme is designed to reduce the organization of the flow rules “ f_{rs} ” and the searching efforts at the SDN switch level. Here, we take an example between two SDN domains, “ $SDN_{dom}(A)$ ” and “ $SDN_{dom}(B)$ ”, as described in Algorithm 1.

Algorithm 1: For a new Mobile Node Serving SDN Domain

1. *Data: SDN Controller and Switches*
2. *Result: Flow-rules assignment to a single Mobile Node*
3. *begin*
4. *Initialization*
5. $SDN_{dom} = B;$
6. $SDN_C; SDN_{Sw}; SDN_{Sw-TCAM}; f_{rs-c}; // \in SDN_{dom}(B)$
7. $MN \in SDN_{dom}(A)$
8. *for all SDN_{Sw}*
9. *if (MN==New)*
10. *//generate a “prob-request” to SDN_C for installation of f_{rs-c}*
11. *If ($f(SDN_C) == prob-request$)*
12. *execute Algorithms 2 and 3*
13. *else*
14. *assign f_{rs}*
15. *end*

The supporting Algorithm 2 is designed to facilitate Algorithm 1, which provides services to arrange the “ f_{rs} ” priority-wise.

Algorithm 2: Priority-Wise Sorting of Heterogeneous SDN Domains in the Serving SDN Domain

1. *Data: flow-table in current SDN controller*
2. *Result: permutation of flow-entries “ μ ” in flow-table priorities wise such that,*
3. *//create and initialize an array for flow-entries*
4. $arr[\mu_1] \leq arr[\mu_2], arr[\mu_3] \leq \dots arr[\mu_m]$
5. *Procedure $Frs_priority(arr [], m)$*
6. *if ($m \leq 1$) then*
7. *return;*
8. *else*
9. *for $i \leftarrow 1$ to $m-1$ do*
10. *if ($arr[\mu_i] < arr[\mu_m]$) then swap ($arr[\mu_i], arr[\mu_m]$)*
11. *end*
12. *$Frs_priority(arr [], \mu_{m-1})$*
13. *end*

The phenomenon of the switch in SDN is to match the packet with “ μ ” of the “ β ” as the packet arrives; if the packet does not match, the switch requests the controller to install or update the flow-table in the switch. Indeed, the SDN controller switches are homogeneous by nature. Thus, all switches are configured on a reactive approach in the IIF-SDN scheme because of the short storage of TCAM. The main advantage of this scheme is that all hybrid flow-rules “ f_{rs} ” of the SDN domains are available locally, thus can easily deploy on the data layer as required. Figure 8 shows the heterogeneous SDN_{doms} .

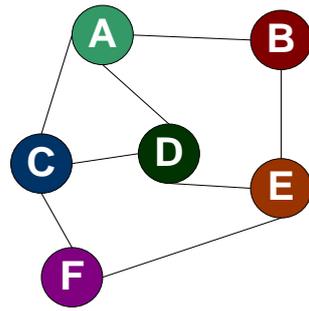


Figure 8. Heterogeneous SDN_{doms} .

The primary SDN_{dom} of MN is “B”, and the serving SDN_{dom} is “A”. The *frs* of $SDN_{dom}(B)$ have already been stored in the $SDN_c \in SDN_{dom}(A)$ controller’s repository under this scheme. The controller installs the *frs* at the corresponding SDN_{Sw} on request of the corresponding SDN_{Sw} . The proposed scheme and its algorithms are very effective and enhance the network capabilities without disturbing the performance of the SDN_c , $SDN_{Sw-TCAM}$, and existing network users.

Algorithm 3: Flow-Rule Integration in TCAM for the New Mobile Node

Data: SDN Controller and Switches

Result: Flow-rules integration in serving SDN domain’s TCAM

begin

1. Initialization
 2. $SDN_c; SDN_{Sw-TCAM}; \beta_c; \lambda; i; k; n;$
 3. for all SDN_{Sw}
 4. do
 5. $\beta_{act(A)}; \lambda[k]; // \in SDN_{Sw-TCAM}(A)$
 6. $\beta_{(B)} [i]; // \in SDN_{c(A)}$
 7. $n = i++;$
 8. $p = k++;$
 9. while $(n \geq p) // n$ is top index of $\beta_c \in SDN_c$ and p is top index of $\lambda \in SDN_{Sw-TCAM} \in SDN_{dom}(A)$
 $f(\beta_{(B)}) + f(\beta_{act(A)}) - \sum_{k=1}^p \lambda(A) + \sum_{i=1}^m \beta_c; // where A and B represent domain A and domain B.$
 10. end
-

Only one thing is added in this scheme that makes it distinguished. It accommodates the user as per its primary domain flow-table. The benefit is to enable the switch to work for all users without any anomaly of the rules. Moreover, the controller can organize the flow-tables of heterogeneous SDN domains.

After obtaining information about a new user’s registration in the network from the switch, the controller distinguishes the user’s domain and sends a request to the primary domains of the user for flow-rules. The user’s primary domain will send the upgraded and organized copy of the flow-rules to the controller. Then, the controller updates/installs the required flow-rules at the corresponding switch as shows in Figure 9.

Algorithm 3, designed for the SDN switches, clearly describes the story of the inner side of the switch and how the IIF-SDN scheme deployed and worked in switches. Under the proposed scheme, Figure 10 is the broader preview of the SDN domains, including hosts, switches, controllers, and other network components.

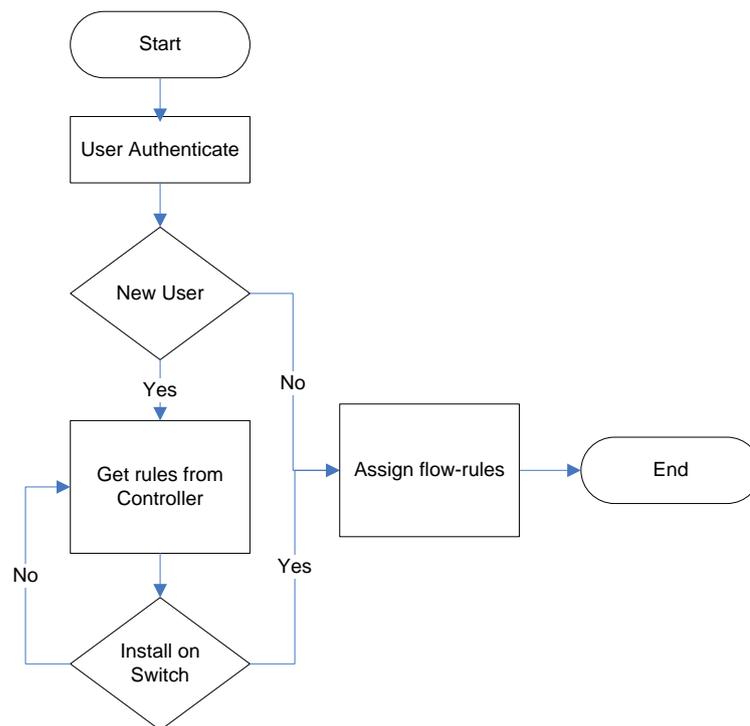


Figure 9. Flow-rule exchange: Reactive approach on the SDN data layer.

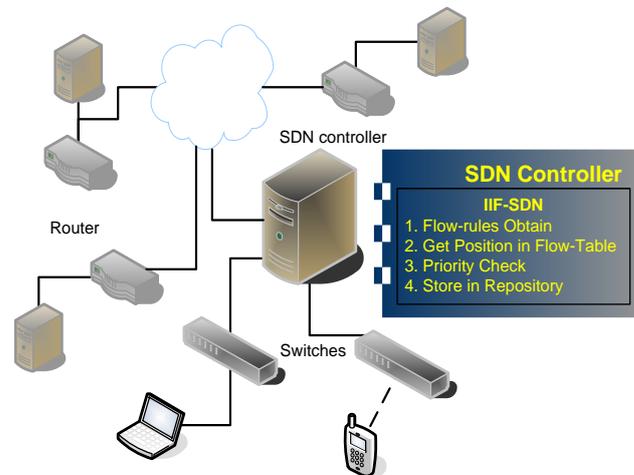


Figure 10. SDN domain (hosts, switches, controller).

3.4. System Model

For the proposed IIF-SDN scheme experiment, software and packages are required to install, for example, processing power and memory size. Underneath, Table 3 describes the environment for the experiments of the proposed scheme.

Table 3. Experimental Environment.

Hardware/Software	Specifications
Processor	3.30 GHz
CPU	i5-4590
RAM	6 GB
OS	Ubuntu 20.04 (64-bit)
Virtual Network	Oracle VM VirtualBox, Mininet, MiniEdit
SDN Controller	RYU
Access Points	OVS (Open Virtual Switch)
OpenFlow Messages	ovs_version 2.13.3
Traffic Generator	V1.3.0
Type of Traffic	IPerf v2.0.5
	TCP and UDP

4. System Implementation and Experimental Demonstrations

The experiment of the IIF-SDN scheme is summarized in Table 4, which is carried out on the small and large system environments separately. A small system experiment is based on two or three nodes only, and an extensive system is based on more than three nodes. Here, the node is considered an SDN domain which shares flow rules. We have separately passed “21,550 and 22,220” flows between the small system and “58,000 and 60,998” in the extensive system under reactive and proactive approaches, as shown in Table 4. We obtained the results of which showed the effectiveness and efficiency of the proposed scheme.

Table 4. Experimental setup.

Scheme	Small System	Large System
Processing Time (ms)	0.28	0.28
Reactive (number of flows)	21,550	58,000
Proactive (number of flows)	22,220	60,998

Numerous flow-tables can be stored in the SDN controller and can be installed on request of the switches in the data layer. The SDN controller communicates with the data layer using the south-bound/north-bound Application Programming Interface (API). A flow-entry is an element of the flow-table that matches and processes the packet. One SDN domain can have many flow-tables. If the SDN domains increase, then the flow-table also increases manifold. In the proposed scheme, we have shared flow-tables of all SDN domains with each SDN domain. Figures 11 and 12 show the SDN domains, flow-table, and flow-table update cost at the SDN controller.

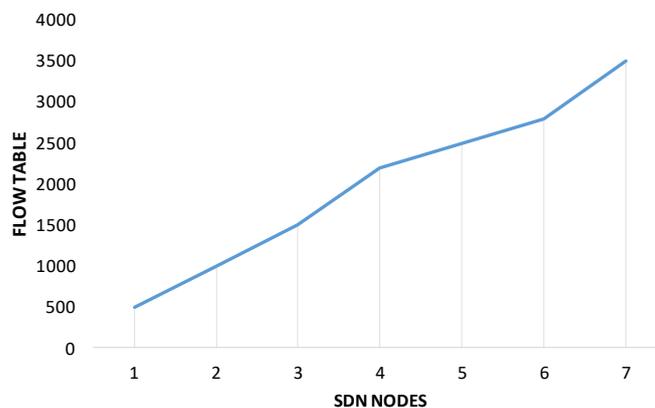


Figure 11. SDN domains and flow-tables.

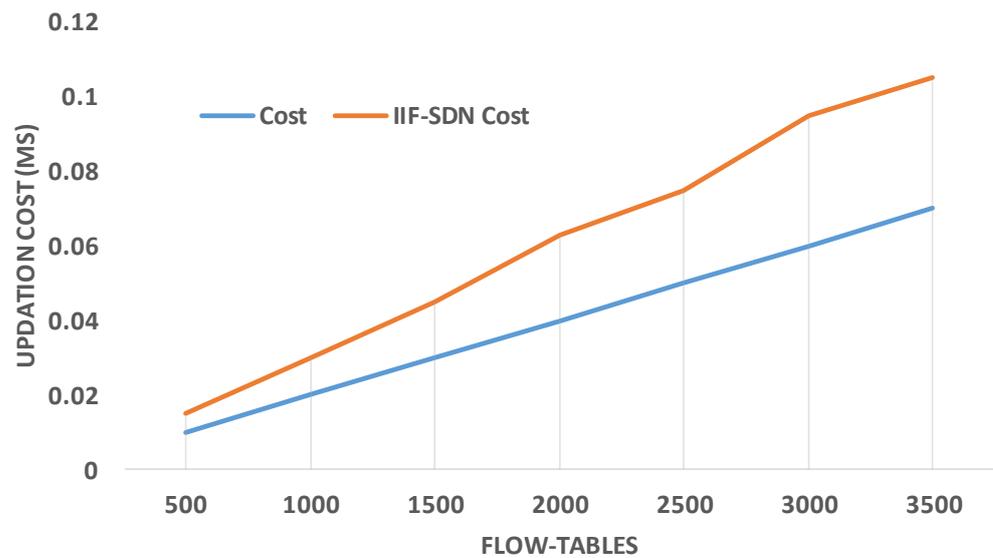


Figure 12. Flow-table update cost at the SDN controller.

We have evaluated the updated cost of the flow-table. The increase in the SDN controller has a direct impact on flow-tables, as shown in Equation (11) below. The graph shows that the IIF-SDN scheme takes more time than the normal SDN domains; this is because we have processed the flow-table priority-wise in descending order on the SDN controller to avoid the switch labor.

$$f(\beta) = \sum_{i=1}^n C^* f(SDN_{Sw-TCAM}); \text{ where } i = \{1, 2, 3, \dots n\} \tag{11}$$

Figures 13 and 14 show the two nodes’ flow-rules and the multiple SDN nodes’ flow-rule exchange. Figures 15 and 16 show the IIF-SDN result with the two and multiple nodes’ flow-rule exchange.

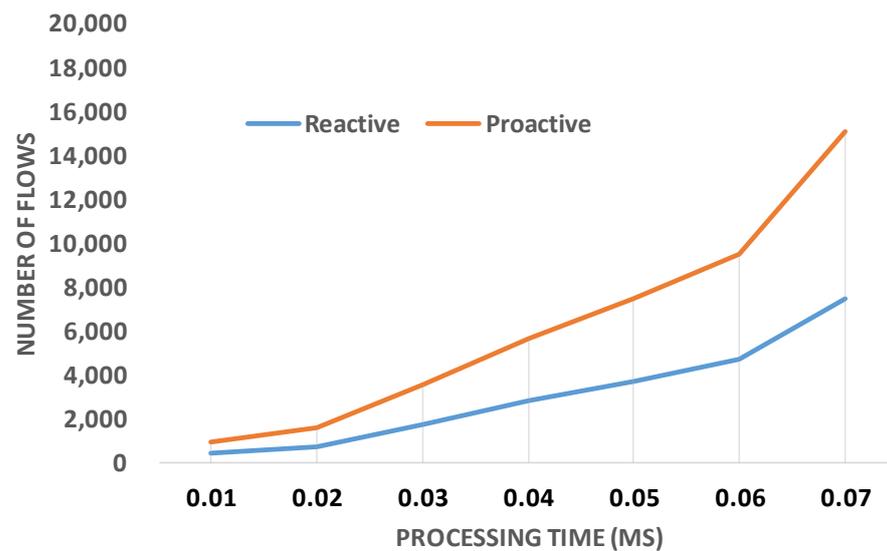


Figure 13. Two nodes’ flow-rule exchange.

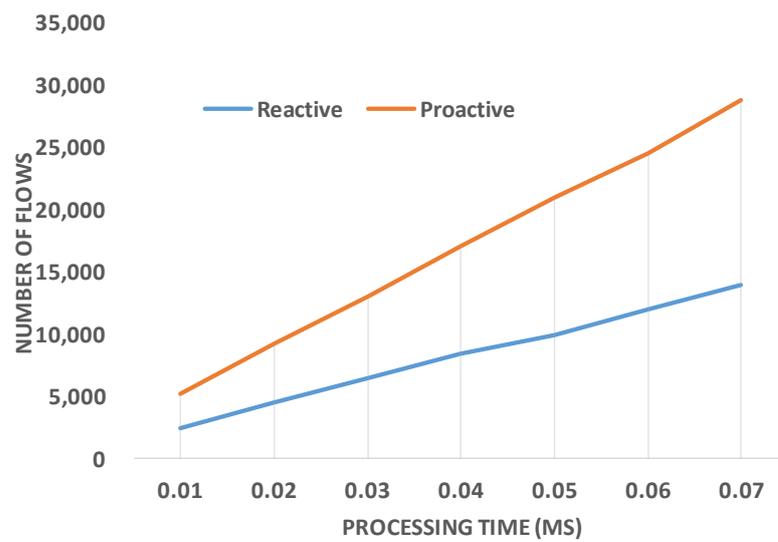


Figure 14. Multiple SDN nodes' flow-rule exchange.

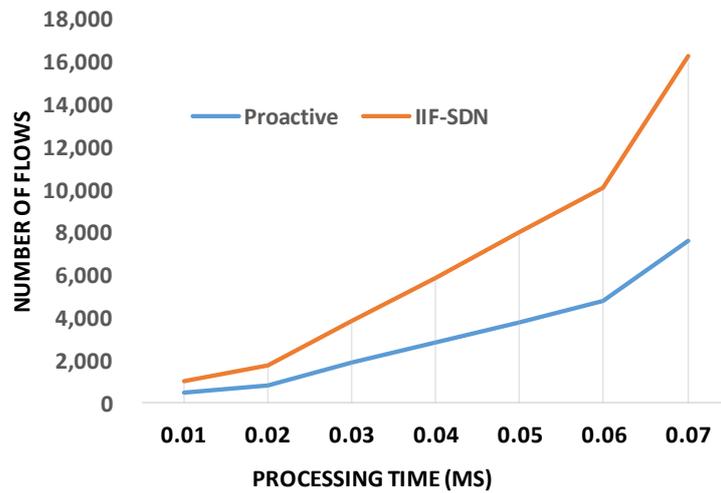


Figure 15. IIF-SDN (two nodes' flow-rule exchange).

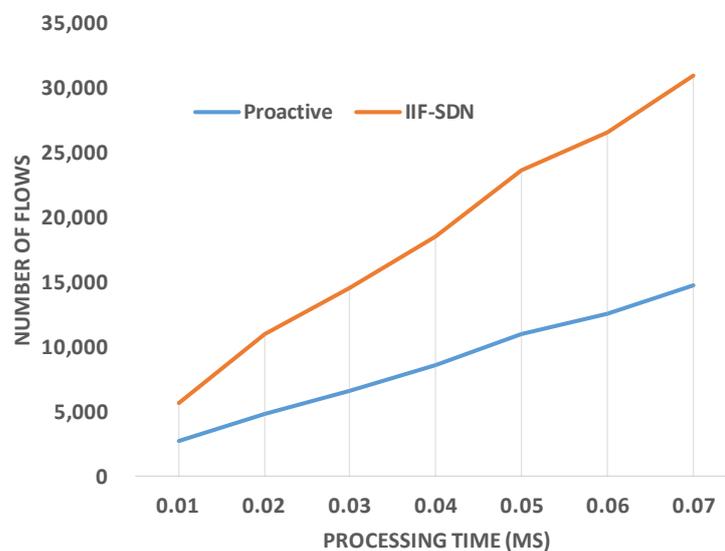


Figure 16. IIF-SDN (multiple nodes' flow-rule exchange).

We analyzed, through an experiment, two SDN domains and multiple SDN domains under reactive and proactive approaches. The result is that after passing several flows, the proactive approach is more efficient than the reactive approach. However, there are better fits for our scenario than the reactive approach because this approach demands the flow-rules when it needs them, and the proactive approach stores them in advance.

We applied the experimental parameters to the IIF-SDN scheme; this scheme was found to be much better than the proactive approach of the SDN domain. The SDN controller worked proactively in our proposed scheme, and the switches used a reactive approach. This hybrid scheme resolved the flow-table management problems of TCAM and ACLs. Figures 17 and 18 show the flow-tables' updating cost and overall performance.

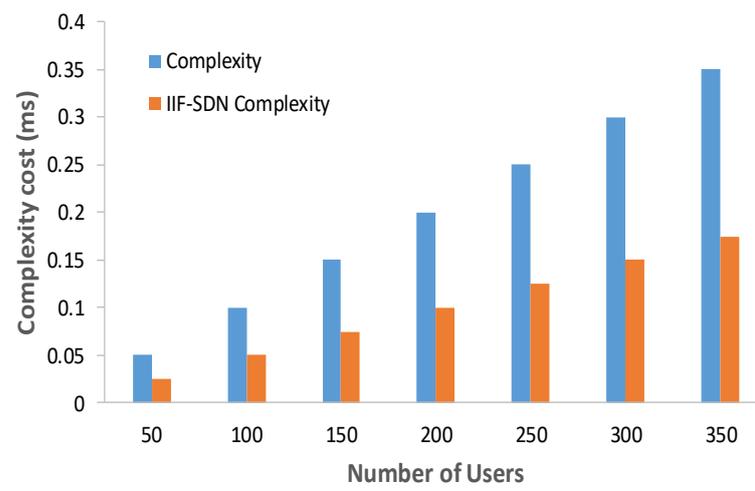


Figure 17. Flow-tables' updating complexity cost.

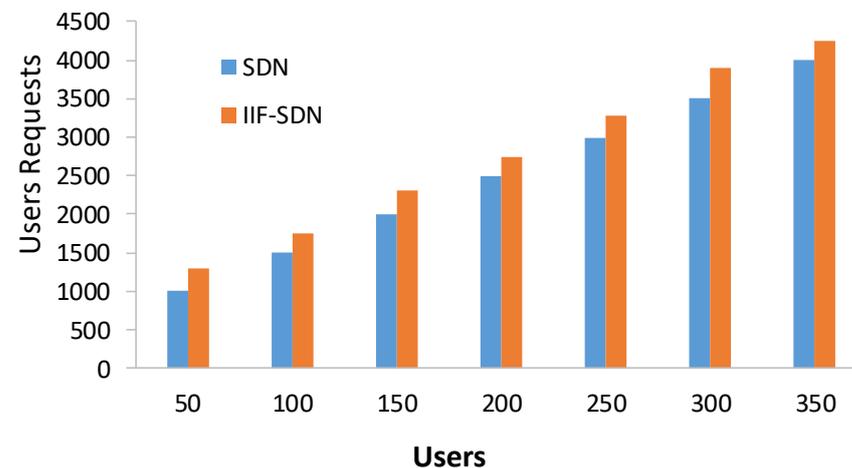


Figure 18. Overall performance.

The above graphs are generated based on simulation results. Initially, communication between all nodes of the proposed topology was established; then, some changes were made in the flow-tables and applied to the nodes of the network. After establishing the connections and some changes between nodes, data were transmitted from one domain to another, and the data transmission stats were achieved.

Once the flow-tables were arranged under the IIF-SDN priority rules, as defined, the complexity of the updating cost remained the least. The overall performance indicated that the results of the proposed scheme are efficient, even if there is an increase in users.

5. Conclusions

It is hard to handle the other SDN domain users' requests for the provision of the same services in comparison to the primary SDN domain in the current domain. As discussed, the TCAM is already overloaded and has a low, dense storage element of the switch. Therefore, it is necessary to reduce the other activities of the switch related to the management of the flow-table and flow-entries. This paper proposed a solution for reducing the lookup activities of TCAM and managing the flow-rules at the switch level. The IIF-SDN scheme is designed to minimize the switch's tasks. A proactive approach is used for the SDN controller to obtain all flow-rules of different SDN domains in advance, and a reactive approach is used for the switches. The proposed scheme is tested on two SDN domains and multiple SDN domains. The scheme is effective and efficient because the switches can obtain all flow-rules locally from the controller. Moreover, the controller arranged all flow-entries priority-wise and saved them in its repository. On the request of the switch, the SDN controller installs flow-rules from its repository for the specific user. The key findings of this paper are the incorporation of the hybrid flow-rule inside the SDN switch, packet transmission without any anomaly, parallel hybrid flow-rule execution, organized flow-entries, and the update process itself. Thus, the network data can be forwarded normally in the continuous update scenario. The proposed scheme is applied because of the quick response between the controller and switches to the service of continuously connecting the users of heterogeneous SDN domains; this is very substantial in maintaining and integrating the hybrid flow-rule in TCAM and ACLs of the switch. In the future, the SDN controller will be assisted by the server that will organize the same flow-tables of different SDN domains on the controller demand of flow-rules for mobile users.

Author Contributions: Conceptualization and Write-up, S.K.A. and S.I., Technical Assistance and paper Improvements, K.N.Q. and G.J.; Validation and Software Assistance, S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. Cui, J.; Jiang, Y.; Song, L.; Zeng, X. Optimizing Flow Entries Update in SDN Switch with Batch Update Mechanism. *Int. J. Innov. Comput. Inf. Control* **2021**, *17*, 1085–1102.
2. Oh, B.-H.; Vural, S.; Wang, N.; Tafazolli, R. Priority-based flow control for dynamic and reliable flow management in SDN. *IEEE Trans. Netw. Serv. Manag.* **2018**, *15*, 1720–1732. [[CrossRef](#)]
3. Khan, S.; Iqbal, S.; Qureshi, K.N.; Ghafoor, K.Z.; Kim, P.; Jeon, G. Survivability of mobile and wireless communication networks by using service oriented Software Defined Network based Heterogeneous Inter-Domain Handoff system. *Comput. Commun.* **2021**, *175*, 177–185. [[CrossRef](#)]
4. Rzepka, M.; Borylo, P.; Lason, A.; Szymanski, A. PARD: Hybrid proactive and reactive method eliminating flow setup latency in SDN. *J. Netw. Syst. Manag.* **2020**, *28*, 1547–1574. [[CrossRef](#)]
5. Li, R.; Pang, Y.; Zhao, J.; Wang, X. A tale of two (flow) tables: Demystifying caching rule in openflow switches. In Proceedings of the 48th International Conference on Parallel Processing, Kyoto, Japan, 5–8 August 2019; pp. 1–10.
6. CN, S. A proactive flow admission and re-routing scheme for load balancing and mitigation of congestion propagation in SDN data plane. *Int. J. Comput. Netw. Commun. (IJCNC)* **2019**, *10*, 117–134.
7. Huang, G.; Youn, H.Y. Proactive eviction of flow entry for SDN based on hidden Markov model. *Front. Comput. Sci.* **2020**, *14*, 1–10. [[CrossRef](#)]
8. Qureshi, K.N.; Hussain, R.; Jeon, G. A distributed software defined networking model to improve the scalability and quality of services for flexible green energy internet for smart grid systems. *Comput. Electr. Eng.* **2020**, *84*, 106634. [[CrossRef](#)]
9. Ying, R.; Jia, W.-K.; Zheng, Y.; Wu, Y. Fast invalid TCP flow removal scheme for improving SDN scalability. In Proceedings of the 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 11–14 January 2019; pp. 1–5.
10. Li, Y.; Zeng, D.; Zheng, L. On communication efficient dataflow computing in software defined networking enabled cloud. *Concurr. Comput. Pract. Exp.* **2021**, *33*, 1–11. [[CrossRef](#)]

11. Yan, B.; Liu, Q.; Shen, J.; Liang, D.; Zhao, B.; Ouyang, L. A survey of low-latency transmission strategies in software defined networking. *Comput. Sci. Rev.* **2021**, *40*, 100386.
12. Rifai, M.; Huin, N.; Caillouet, C.; Giroire, F.; Moulhierac, J.; Pacheco, D.L.; Urvoy-Keller, G. Minnie: An sdn world with few compressed forwarding rules. *Comput. Netw.* **2017**, *121*, 185–207. [[CrossRef](#)]
13. Shi, X.; Li, Y.; Xie, H.; Yang, T.; Zhang, L.; Liu, P.; Zhang, H.; Liang, Z. An OpenFlow-Based Load Balancing Strategy in SDN. *Comput. Mater. Contin.* **2020**, *62*, 385–398. [[CrossRef](#)]
14. Huang, X.; Bian, S.; Shao, Z.; Xu, H. Predictive switch-controller association and control devolution for SDN systems. *IEEE/ACM Trans. Netw.* **2020**, *28*, 2783–2796. [[CrossRef](#)]
15. Huang, V.; Chen, G.; Zhang, P.; Li, H.; Hu, C.; Pan, T.; Fu, Q. A scalable approach to SDN control plane management: High utilization comes with low latency. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 682–695. [[CrossRef](#)]
16. Ramos, R.M.; Martinello, M.; Rothenberg, C.E. Slickflow: Resilient source routing in data center networks unlocked by openflow. In Proceedings of the 38th Annual IEEE Conference on Local Computer Networks, Sydney, Australia, 12–14 October 2013; pp. 606–613.
17. Tao, L.; Xiangming, W.; Zhaoming, L.; Xing, Z.; Yangchun, L.; Biao, Z. SWN: An SDN based framework for carrier grade Wi-Fi networks. *China Commun.* **2016**, *13*, 12–26. [[CrossRef](#)]
18. Kim, Y.; Park, J.; Kyung, Y. Mobility-Aware Hybrid Flow Rule Cache Scheme in Software-Defined Access Networks. *Electronics* **2022**, *11*, 160. [[CrossRef](#)]
19. Xie, L.; Zhao, Z.; Zhou, Y.; Wang, G.; Ying, Q.; Zhang, H. An adaptive scheme for data forwarding in software defined network. In Proceedings of the 2014 Sixth IEEE International Conference on Wireless Communications and Signal Processing (WCSP), Hefei, China, 23–25 October 2014; pp. 1–5.
20. Qiu, K.; Yuan, J.; Zhao, J.; Wang, X.; Secci, S.; Fu, X. Fastrule: Efficient flow entry updates for tcam-based openflow switches. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 484–498. [[CrossRef](#)]
21. Qiu, K.; Yuan, J.; Zhao, J.; Wang, X.; Secci, S.; Fu, X. Fast lookup is not enough: Towards efficient and scalable flow entry updates for TCAM-based OpenFlow switches. In Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Australia, 2–5 July 2018; pp. 918–928.
22. Lu, M.; Deng, W.; Shi, Y. TF-IdleTimeout: Improving efficiency of TCAM in SDN by dynamically adjusting flow entry lifecycle. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 12–14 October 2016; pp. 2681–2686.
23. Marsico, A.; Doriguzzi-Corin, R.; Siracusa, D. An effective swapping mechanism to overcome the memory limitation of SDN devices. In Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, 8–12 May 2017; pp. 247–254.
24. Isyaku, B.; Kamat, M.B.; Bakar, K.A.; Zahid, M.S.M.; Ghaleb, F.A. IHTA: Dynamic idle-hard time-out allocation algorithm based openflow switch. In Proceedings of the 2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 18–19 April 2020; pp. 170–175.
25. Li, X.; Huang, Y. A flow table with two-stage time-out mechanism for sdn switches. In Proceedings of the 2019 IEEE 21st International Conference on High Performance Computing and Communications; (HPCC/SmartCity/DSS), Zhangjiajie, China, 10–12 August 2019; pp. 1804–1809.
26. Zhang, S.Q.; Zhang, Q.; Tizghadam, A.; Park, B.; Bannazadeh, H.; Boutaba, R.; Leon-Garcia, A. TCAM space-efficient routing in a software defined network. *Comput. Netw.* **2017**, *125*, 26–40. [[CrossRef](#)]
27. Kuzniar, M.; Perešini, P.; Kostić, D.; Canini, M. Methodology, measurement and analysis of flow table update characteristics in hardware openflow switches. *Comput. Netw.* **2018**, *136*, 2–236. [[CrossRef](#)]
28. Sheu, J.-P.; Lin, W.-T.; Chang, G.-Y. Efficient TCAM rules distribution algorithms in software-defined networking. *IEEE Trans. Netw. Serv. Manag.* **2018**, *15*, 854–865. [[CrossRef](#)]
29. Ding, Z.; Fan, X.; Yu, J.; Bi, J. Update cost-aware cache replacement for wildcard rules in software-defined networking. In Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC), Natal, Brazil, 25–28 June 2018; pp. 457–463.
30. Wang, C.; Youn, H.Y. Entry aggregation and early match using hidden Markov model of flow table in SDN. *Sensors* **2019**, *19*, 2341. [[CrossRef](#)]
31. Kang, N.; Liu, Z.; Rexford, J.; Walker, D. Optimizing the “one big switch” abstraction in software-defined networks. In Proceedings of the ninth ACM Conference on Emerging Networking Experiments and Technologies, Rome, Italy, 6–9 December 2013; pp. 13–24.
32. Huang, G.; Youn, H.Y. Management of Flow Table of SDN for Proactive Eviction Using Fuzzy Logic. *J. Inf. Sci. Eng.* **2021**, *37*, 4.
33. Wang, J.; Cheng, S.; Fu, X. SDN Programming for Heterogeneous Switches with Flow Table Pipelining. *Sci. Program.* **2018**, *2018*, 2848232. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.