Mobi-Flow: Mobility-Aware Adaptive Flow-Rule Placement in Software-Defined Access Network

Samaresh Bera, *Student Member, IEEE,* Sudip Misra, *Senior Member, IEEE,* and Mohammad S. Obaidat, *Fellow, IEEE*, *Fellow, SCS*

Abstract—In this paper, we propose a mobility-aware adaptive flow-rule placement scheme, named as *Mobi-Flow*, with an aim to maximize the overall performance in a software-defined access network (SDAN). The proposed scheme consists of two components — *path estimator* and *flow-manager*. The path estimator predicts future locations of end-users present in the network, depending on their history location sets, and delivers the predicted locations to the flow-manager. We use the order-k Markov predictor to predict the next possible locations of the end-users. Based on the predicted locations, the flow-manager determines access points (APs) in the network, which can be associated with the users to fulfill the requirements of the latter. We use the mixed integer linear programming (MILP) approach to determine the *optimal* number of APs, in order to minimize the cost associated with flow-rule placement. Consequently, the flow-manager implements the flow-rules at APs, so that adequate actions for the incoming requests can be taken in an adaptive manner, without querying the controller. We consider a practical scenario of an IoT environment, in which both static and mobile users are present. Therefore, the proposed scheme, *Mobi-Flow*, can be integrated atop the SDN controller to support the emerging concept of SDN-based IoT networking. Through extensive simulations, we show that *Mobi-Flow* is beneficial for minimizing the delay, and number of activated APs, control overhead, energy consumption, and cost in the network, compared to the existing schemes — open shortest path first (OSPF), minimum occupied rule capacity (MRC), *distributed* (non-SDN), and MoRule. Particularly, the proposed scheme is capable of reducing the cost by 39%, 38%, 65%, and 11%, compared to OSPF, MRC, *distributed*, and MoRule, respectively.

Index Terms—Software-Defined Networking, Internet of Things, Wireless Access Network, Rule Placement, Mobility, Markov Predictor.

List of Symbols

$\mathcal{A}$	Set of APs in the network
E_{act}	Energy consumption of AP in active mode
E_{fwrd}	Energy consumption of AP to forward traffic
$\mathcal{A}_{act}$	Set of activated APs in the network
$\mathcal{U}$	Set of users in the network
A_{tr}	AP has traffic to forward
$\mathcal{R}_i^{max}$	Maximum rule capacity of i^{th} AP
$\mathcal{D}_j^{th}$	Allowable delay of a request from user j
$\mathcal{H}$	Movement history set of a user
$\mathcal{S}$	Set of locations of a user in the network
$\mathcal{T}$	Set of arrival time at different locations
$\mathcal{Z}_s$	Duration of stay at location s
$\Phi_{i,j}^t$	Cost at i^{th} AP to serve j^{th} request at time t
L_i	Packet arrival rate at AP i
B_{ij}	Link bandwidth between AP i and user j

1 Introduction

Due to the advent of software-defined networking (SDN) [1], it is getting attention among the networking researchers to support real-time application-specific requirements. In SDN, network-specific control strategies are defined by a centralized controller, while decoupling the *control-plane* and *data-plane* from the forwarding devices. The controller takes real-time control decisions based on the data received from the physical devices (considered as the data-plane). Therefore, the control-plane can be configured according to requirements of end-users, without going into the vendor-specific architecture of the physical devices. Thus, SDN-based solution approaches are among the most promising to configure devices in real-time in order to meet application-specific requirements. Table 1 presents an overview of the benefits of using SDN over traditional networks.

Concurrently, internet of things (IoT) is an emerging technology to digitize everything for the betterment of the connected world [4]. Primary aim of IoT is to connect all network devices together and to control them in a unified manner. As a result, heterogeneous devices present in the network need to be interconnected and configured depending on the requirements. Further, the IoT devices are connected to the backbone network through access networks. The association between the IoT devices and access points (APs) takes place in a distributed manner. However, the traditional networking technologies suffer from vendor-specific configuration of the devices, which, in turn, limits the usability of the devices in an IoT environment. Moreover, business requirements of modern applications require the hardware devices to change their in-built policies over time. This necessitates the hardware devices to be configured according to the application-specific requirements in real-time. Consequently, SDN-based solution approaches are getting interests among researchers to address different

- *S. Bera and S. Misra are with the Computer Science and Engineering Department, Indian Institute of Technology, Kharagpur, 721302, India. E-mail: s.bera.1989@ieee.org, smisra@sit.iitkgp.ernet.in*
- *M. S. Obaidat is with the ECE Department, Nazarbayev University, Astana, Kazakhstan, KASIT, University of Jordan, Amman, Jordan, and University of Science and Technology Beijing (USTB), Beijing, China. Email: m.s.obaidat@ieee.org*

TABLE 1: Advantages of SDN over traditional networks [1]–[3]

Feature	Brief description
Logically centralized control paradigm	Network devices can be controlled in a logically centralized manner through the decouplement of control and data planes from traditional networking devices.
Global network view	Use of application programming interfaces (APIs) enables global-view of the network. This helps to monitor real-time network status in a centralized manner.
Open architecture	Network devices can be programmed according to application-specific requirements, which, in turn, increases the re-usability of networking devices over the traditional approaches.
Optimized data flow	Rule-based data forwarding mechanism enables prioritized data delivery, while improving overall network performance. Further, it enables multipath data forwarding using the concept of flow-splitting across multiple devices.
Adoption of new business policy	Absence of vendor-specific architecture allows network administrators to adopt new business policies and integrate them into the network without changing the physical devices unlike in traditional network.

networking issues (such as data forwarding and network management) to support IoT applications, as it separates out the control plane from traditional hardware devices [2], [3], [5]. Thus, the devices in the network can be configured in real-time according to the application-specific requirements using the concept of SDN. Thus, we consider a software-defined access network (SDAN) in this work to support dynamic requirements of IoT applications.

Typically, an IoT environment consists of both stationary and mobile devices[1], which communicate with access points (APs) to exchange real-time information. In such a scenario, APs maintain a flow-table to exchange information between users and backbone networks. Due to mobility of users and capacity constraints of APs, the flow-table rules need to be optimally managed at the latter, depending on the presence of the devices and their requests. However, the existing SDN-based solution approaches for flow-rule placement either considered the static behavior of the network or mainly focused on backbone networks, in which the impact of network dynamics is very low. Consequently, there is a need to have an optimal flow-rule placement scheme in the SDAN for efficient network management, while considering users' mobility. To address such issues, two solution approaches are feasible to update the flow-rules for packet forwarding — a) reactive – the APs send *packet-in* messages to the controller after receiving new requests, and the controller defines flow-rules; b) proactive – the controller places the flow-rules at APs in a proactive manner based on the end-devices' movement in the network. Due to the mobility of the users, it is expected that flow-rules at the APs are required to be updated frequently, which, in turn, generates *packet-in* messages to the SDN controller on receiving new requests. As a result, both the network overhead and service delay increase in case of the reactive approaches. Therefore, we propose a proactive scheme over the reactive one to minimize the service delay in the network, while holistically minimizing the cost in the network. Detailed contributions of this work are discussed in the subsequent section.

1.1 Contribution

In this paper, we attempt to address these questions: (a) Can we place the flow-rules at APs in an adaptive manner according to the movement of the users in the network? (b) Can we utilize the existing rule-space to deploy flow-rules, while minimizing the number of activated APs in the network? To address these questions, we propose a mobility-aware flow-rule placement scheme for software-defined network, an attempt to maximize the overall performance, while considering end-users' mobility. Therefore, the proposed scheme deploys the flow-rules at the APs in the network in an adaptive manner. It is noteworthy that we address the flow-rule placement problem at software-defined access network (SDAN), while considering incoming traffic from IoT applications. Therefore, the proposed scheme is applicable to SDAN for flow-rule placement in an adaptive manner. In brief, the specific *contributions* in this paper are as follows:

- We propose a mobility-aware adaptive flow-rule placement scheme in SDAN. The problem is challenging because of the presence of heterogeneity and mobility of end-users and their dynamic requirements in an IoT environment, and capacity constraints of the APs.
- We formulate a mixed integer linear problem (MILP) to minimize number of activated APs while considering capacity constraints of the APs and application-specific requirements of users in the network.
- The proposed scheme consists of two components — *path estimator* and *flow-manager* — which are placed at the SDN controller end. The path estimator predicts the future locations of users, and the flow-manager implements the flow-rules at the APs, based on the predicted locations. Order-k Markov predictor [6], [7] is used to predict the future locations.
- We discuss two use-case scenarios to present the suitability and practical applications of the proposed scheme from the perspectives of IoT. It is noteworthy that the proposed scheme does not introduce any client-side changes. Thus, it can be integrated with the existing SDN architecture.
- Extensive simulation results show that the proposed scheme is beneficial to minimize network delay, number of activated APs, control overhead, and associated cost in rule placement, compared to the existing schemes — open shortest path first (OSPF) and minimum occupied rule capacity (MRC) without location prediction (as described in [8]), *distributed* (non-SDN), and MoRule [9].

1. In this paper, the term 'user' and 'end-device' are synonymously used to denote the same entity.

1.2 Organization

The rest of the paper is organized as follows. Section 2 presents the existing state-of-the-art from the aspects of flow-rule placement in SDN-enabled networks. Detailed system model is presented in Section 3. Section 4 describes the solution approach proposed in this paper. Section 5 presents the extensive simulation results to show the effectiveness of the proposed scheme over the existing ones. Sections 6 and 7 discuss the limitations and practical applications of the proposed scheme, respectively. Finally, we conclude the paper in Section 8 with some future research directions.

2 Related Work

There exists a few recent works in the literature, which focused on flow-rule management in SDN. We discuss the existing solution approaches according to their application areas — wired network, access network, and IoT applications.

2.1 Solutions for Traditional Wired Networks

Nguyen et al. [10] surveyed the existing works focusing on rule-placement problem in OpenFlow-enabled networks. The authors first defined the problems and challenges involved in rule-placement in OpenFlow-enabled networks. Secondly, they discussed the existing solution approaches, which are capable of addressing the challenges in rule-placement. Li et al. [9] proposed an optimal rule placement scheme according to devices' association probability to switches, while considering limited TCAM at the switches. On the other hand, an energy-aware routing scheme is proposed by Giroire et al. [11] in an SDN-enabled network. In such a scheme, the SDN controller checks for the available links which are unused in the network. After identifying the unused links, the controller puts those links into the sleep mode to save energy. Similarly, Markiewicz et al. [12] proposed an energy-efficient traffic forwarding scheme for an SDN-enabled network, while considering dynamic traffic in the network. Vawter et al. [13] proposed optimal traffic management policies to minimize unwanted traffic in the network, while maximizing the network performance. The authors developed a test-bed to analyze the network performance of such SDN-enabled network. Huang et al. [14] proposed an optimal rule partition and allocation scheme for backbone network switches. The rules at the switches are handled in an efficient manner, depending on the status of the network. Ma et al. [15] proposed a network function virtualization (NFV) scheme, depending on the dynamic requirements of the network. The proposed traffic-aware middleboxes placement approach is converted to a graph optimization problem. Nguyen et al. [16] proposed a general optimization framework for rule placement based on the OpenFlow protocol. In such a scheme, the under-utilized forwarding devices are utilized first, depending on the number of allocated rules at them. Therefore, the forwarding devices with minimum number of flow-rules occupied are prioritized over the maximum occupied ones. On the other hand, Caria et al. [17] proposed OSPF-based routing strategy in an SDN-enabled network, while leveraging the global view of the network.

2.2 Solutions for Access Network and IoT Applications

Kerpez et al. [18] introduced the concept of software-defined access network (SDAN) to control the access devices in a centralized manner, while utilizing the benefits of SDN and network function virtualization (NFV). The authors examined the proposed concept in different use-case scenarios such as resource allocation, service differentiation, and dynamic spectrum management. Rastegar et al. [19] proposed a rule-caching mechanism at software-defined radio access network to minimize the flow-setup delay. The authors formulated a mixed integer linear program for fair allocation of flow-table space among users in the network. Through simulation results, they showed that the rule-caching mechanism is useful within a base station to reduce the flow-setup delay. Amokrane et al. [8] proposed an energy-efficient routing scheme for campus networks using the concepts of SDN. In such a scheme, the authors considered a practical environment, in which the end-users enter into the service regions of access points and leave them in an unpredictable manner. Consequently, the dynamic behavior of traffic patterns is considered.

Sood et al. [2] discussed different challenges and opportunities of software-defined wireless networking technologies, which have the potential to fulfill the requirements of IoT. The authors mainly focused on the challenges involved in the integration of SDN and IoT from the aspects of security and scalability. Liu et al. [20] proposed an SDN-based IoT architecture for smart urban sensing. The authors claimed that the users in an IoT environment can communicate among themselves and exchange real-time information, in order to have a convenient and comfortable living environment. According to the sensed information from the physical devices, the controller decides the adequate control strategies. Hakiri et al. [21] discussed different issues and challenges which need to be addressed for efficient and scalable IoT communications. The authors discussed different issues such as mobility management, network management, placement of middleware boxes, service provisioning, and interoperability in an IoT environment. On the other hand, Anadiotis et al. [22] proposed an SDN-assisted framework for deploying map-reduce functions to handle big-data in wireless sensor networks (WSNs). The proposed framework can also be applied to an IoT environment, in which multiple WSNs are present to sense the environment, as suggested by Liu et al. [20].

Synthesis: Detailed analysis of the existing works discussed in Sections 2.1 and 2.2 reveals that there is a research lacuna on rule placement policies at SDAN to support IoT applications, in which both the static and mobile devices are present. Due to the presence of mobile devices in the network, frequent flow-rule update for incoming requests is required at the APs. However, the existing solution approaches either considered static behavior of the network or are limited to backbone networks. Therefore, in this paper, we propose mobility-aware adaptive flow-rule management at the SDAN to support IoT applications.

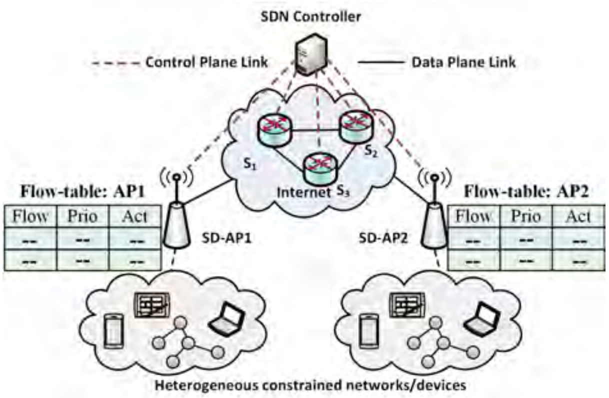

Fig. 1: A schematic view of the SDN architecture

3 System Model

3.1 Architecture

Figure 1 presents a schematic view of an SDN architecture comprising of backbone network, APs, and users/devices present in an IoT environment. We consider that the IoT environment consists of heterogeneous devices (such as sensors, mobile devices, and peripheral devices), which communicate with APs. Additionally, we also consider that the end-devices can be both stationary and mobile in nature, as considered in a practical IoT network. The APs forward data traffic based on the flow-rules decided by a centralized SDN controller, as depicted in Figure 1. Therefore, the flow-table rules at APs are dynamically updated by the controller, depending on application-specific requirements and locations of the users.

The data traffic accessed by the APs is further forwarded through the backbone communication network (i.e., routers and switches). The APs adapt the flow-table rules decided by the controller and take adequate actions for an incoming data traffic from the end-devices. For simplicity, we do not focus on the forwarding issues present in the backbone networks, as network dynamics in the backbone network is very low. Interested readers may refer to the existing works discussed in [23].

3.2 Energy Consumption of APs

The users are located at one-hop distance from the APs, and the latter forwards users' requests to the backbone networks. Thus, the energy consumption model of an AP is presented as follows:

$$E = \begin{cases} E_{act} + E_{fwrd}, & \text{if } A_{act} = True \text{ and } A_{tr} = True \\ E_{act}, & \text{if } A_{act} = True \text{ and } A_{tr} = False \\ 0, & \text{Otherwise} \end{cases} \tag{1}$$

where E_{act} and E_{fwrd} denote the energy consumption of an AP when it is active and has traffic to forward, respectively. A_{act} and A_{tr} denote that the AP is active and has packets for transmission, respectively.

3.3 Programmability of Software-Defined APs

It is possible to control the APs in a centralized manner similar to the SDN-enabled forwarding devices present in wired networks [24]. Therefore, APs forward the traffic in the network according to the flow-rules defined by SDN controller. The APs are also constrained by the rule-capacity. Thus, a fixed number of flow-rules can be inserted at the APs. Consequently, we focus on the rule placement at the APs, which is one of the practical aspects present in SDN.

Typically, access points' association decisions with end-users are made by the latter ones. The APs do not have any centralized control over the association of end-users. However, the association of end-users with APs can be controlled in a centralized manner, while placing an agent at the APs [25]. We consider that the APs are configured with the agent so that the association of end-users can be done based on available energy, residual rule capacity and associated delay at the APs.

It is noteworthy that the SDN controller controls the networking devices in a centralized manner while leveraging the global view of the network. The physical architecture of controller placement can be based on flat or hierarchical or mesh topology [26], [27]. For example, a group of devices are controlled by a controller, and another group of devices are controlled by another controller, and so on. Therefore, an individual controller controls the devices in a semi-distributed manner, while having the information of the entire network. Consequently, we believe that the adoption of SDN in IoT application would not affect the network performance, rather it would help to proliferate the same due to the presence of global knowledge of the network. Recent works (e.g., [2] and [3]) on SDN-based IoT applications also claimed that adoption of SDN would increase the network performance.

3.4 Problem Formulation

As shown in Figure 1, the flow-rules at APs are dynamically changed, depending on the users' positions and requirements. Due to the resource constraint nature of the access points (APs), limited number of flow-rules can be placed [28]. Let us consider a wireless network comprising of multiple APs, which is denoted by the set $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$, and a set of users denoted by $\mathcal{U} = \{U_1, U_2, \ldots, U_n\}$, where $n \in \mathbb{Z}^+$. Let us also consider that the maximum number of rules that can be entered in i^{th} AP due to the capacity constraints is $\mathcal{R}_i^{max}$. Mathematically,

$$\bigcup_{j=1}^{r} A_i(R_j) \leq \mathcal{R}_i^{max} \quad j \in \mathcal{N} \tag{2}$$

where $A_i(R_j)$ denotes the j^{th} flow-rule at i^{th} AP, and r is the current number of flow-rules present at the AP. Therefore, R_j corresponds to the j^{th} flow-rule in an AP.

The objective of the SDN controller is to minimize the number of activated APs, in order to minimize the overall operating cost in the network[2]. Mathematically,

$$\text{Minimize} \quad \sum_{i \in \mathcal{A}} \mathcal{A}^t_{act,i}$$

subject to

$$\mathcal{R}^t_i \leq \mathcal{R}^{max}_i,\ i \in \mathcal{A}, \tag{3}$$
$$r^t_j \geq 1,\ j \in \mathcal{U}, \tag{4}$$
$$r^t_j \in \mathcal{R}^t_i = TRUE,\ \text{if } j \rightarrow i \text{ is } TRUE, \tag{5}$$
$$E^t_i \leq E^{avl}_i,\ i \in \mathcal{A} \tag{6}$$
$$\mathcal{D}^t_j \leq \mathcal{D}^{th}_j \tag{7}$$

where $\mathcal{A}^t_{act,i}$ denotes that an AP, $i \in \mathcal{A}$, is activated at time t. Equation (3) denotes that the total number of flow-rules present at the AP i at time t is always less than or equal to its maximum capacity. Equation (4) represents that the number of flow-rules associated with a user is greater than or equal to 1. Therefore, a flow-rule associated to a user can exist at multiple APs, in order to provide seamless service to the users. It is also important to consider that the flow-rule for a particular request must be inserted into the flow-table of the APs to which the user can be associated. This is confirmed by introducing a constraint, as mentioned in Equation (5). Equation (6) confirms that the required energy to serve all the requests (refer to Equation (1)) must be less than or equal to the available energy, E^{avl}_i, at the associated AP[3]. Finally, Equation (7) denotes that the delay incurred to serve a particular request must be less than or equal to the maximum allowable delay. It is noteworthy that the parameters used to denote the constraints presented in the above optimization problem are the combination of integers and non-integers. Therefore, the above optimization problem is formulated as a mixed integer linear program (MILP).

3.5 Illustrative Example

In this Section, we present an illustrative example of the problem scenario presented in Section 3.4. Figure 2 presents a motivating scenario consisting of three different scenarios. We consider a network consisting of two APs, and few heterogeneous devices such as sensors, smartphones, and PDAs. S1, M1 and M2, and P1 denote sensor node, mobile devices, and personal digital accessories, respectively. Depending on the positions and requirements, flow-rules are defined at the APs, as shown in Figure 2. We consider that both the APs have limited rule-space capacity. We present three scenarios which can occur during flow-rule placement at APs, as depicted in Figure 2. To address such issues, we propose an adaptive flow-rule placement scheme, while considering the movement of the users in the network.

4 Mobility-Aware Flow-Rule Placement

The proposed framework is presented in Figure 3, and it is placed at the controller end. Consequently, the computational complexity is avoided at the APs. The proposed model consists of different components — path estimator, flow manager, database, and flow-table — as presented in Figure 3. The path estimator predicts the future locations of end-devices based on history data (discussed in Section 4.1). Further, based on the predicted locations, the flow manager decides the flow-rules (discussed in Section 4.2), and the rules are placed at the associated APs, in order to provide seamless connectivity.

2. It is noteworthy that the architecture and formulated problem are generalized and can be applied to software-defined access networks.

3. It is possible that some of the APs do not run with traditional power sources. Harvested energy can be used to run such APs. Therefore, we take into account the available energy at the APs before deploying the flow-rules.

4.1 Location Prediction

We use order-k, $O(k)$, Markov predictor [6], [7] to estimate the future locations of end-devices. The model consists of two tuples: $< \mathcal{H}, P >$, where

- $\mathcal{H}$: Set of movement history containing three tuples $< \mathcal{S}, \mathcal{T}, \mathcal{Z} >$, where $\mathcal{S}$ denotes the set of locations of meaningful places visited by a user (i.e., end-device), which is represented as $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$, $n \in \mathcal{N}$. $\mathcal{T}$ is a set of arrival times at the different locations in $\mathcal{S}$, and it is denoted by $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$. $\mathcal{Z}$ denotes the set of durations of stay at all locations in the set S, and it is represented as $\mathcal{Z} = \{z_1, z_2, \ldots, z_n\}$.
- P: Probability of transition from one location to another. Therefore, P_{ij} represents the probability of transition from location s_i to location s_j, $i \neq j$.

It is noteworthy that presented tuples are only for a particular user, and the path estimator predicts the next location of the user according to the tuples. Therefore, this process is repeated for all users present in the network, in order to get predicted locations for all of them.

Assumption 1. *In SDN, the controller periodically checks for the location of the hosts and connections between switches. Therefore, the associated overhead for collecting network statistics is taken care by the OpenFlow protocol [30]. In the proposed work, we adopt such facility, so that it does not add additional control overhead to the system for collecting the network statistics.*

Definition 1. *Location: A user can move from one place (i.e., location) to another, depending on his/her requirements. Each location defines a semantic meaning, for example, home, market, office (work), and playground. Therefore, a set of locations associated with a user is presented as $\mathcal{S}$, as mentioned before.*

Definition 2. *Arrival Time: A user can visit different locations at different time instants. Therefore, in addition to the location set, a set of arrival time is also associated with the user, which is denoted as $\mathcal{T}$.*

Definition 3. *Duration of Stay: Further, the user has different durations of stay at different locations. Consequently, a set of durations of stay is also considered in the movement history set. It is denoted as $\mathcal{Z}$.*

Corollary 1. *The Markov predictor with order k=3 is a finite state predictor having 2^k possible location contexts, where $\mathcal{S}(n-k+1, n) = \{\mathcal{S}_{n-2}, \mathcal{S}_{n-1}, \mathcal{S}_n\}$. The initial context, n=0, does not affect the assymptotic performance of the predictor [31].*

To predict the location and time of the next hand-off, we need to calculate the probability that a hand-off will occur in the next Δt time period, while the current location s and duration of stay z at the location are given. From $\mathcal{H}$, we

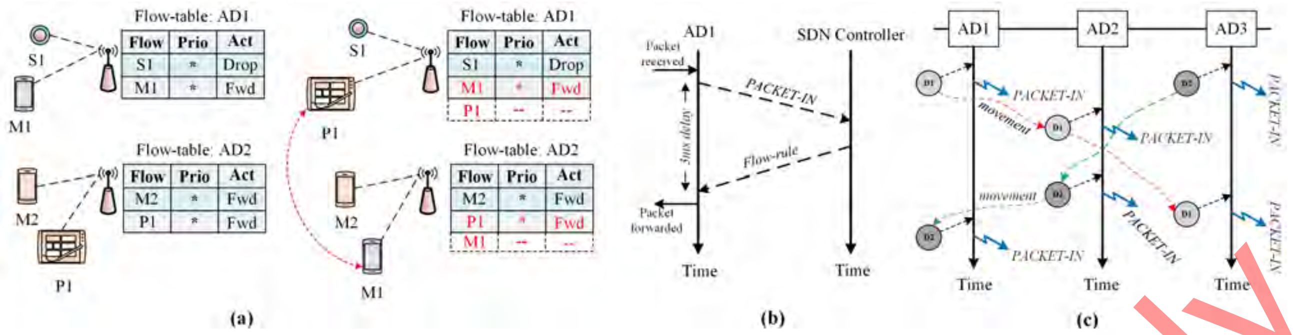

Fig. 2: Motivating example: (a) Scenario 1: APs are capable of handling requests from all devices associated with them. Due to the movement of users, flow-rules at the APs are required to be updated; (b) Scenario 2: On receiving a new packet, AP sends a *packet-in* message to the controller. Accordingly, the controller defines adequate flow-rules, and they are inserted at the APs. Typically, 3ms delay is involved in the flow-rule insertion process [29]. As a result, the packet is queued at the AP for 3ms; (c) Scenario 3: Due to the movement of the users, multiple *packet-in* messages are sent to the controller, which, in turn, increases the control message overhead.

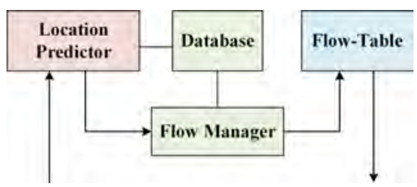

Fig. 3: Proposed controller framework for flow-rule placement

extract the location history set $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$, and from $\mathcal{S}$, the order-k[4] location context $c = \mathcal{S}(n-k+1, n) = \{\mathcal{S}_{n-k+1}, \mathcal{S}_{n-k+2}, \mathcal{S}_n\}$ is calculated. After calculating the location context c, the path estimator searches for the locations whose contexts match with c, in order to find out the duration of stay at each of the locations. Therefore, the path estimator predicts the set of durations of stay Z_s at each possible location s which follows c. Mathematically,

$$Z_s = \{z_i | z_i = t_{i+1} - t_i, \text{ where } \mathcal{S}(i-k+1, i+1) = c_s\} \quad (8)$$

For each Z_s, we calculate the conditional probability $P_s(t \leq z < t + \Delta t)$ that the user will move to location s within Δt time after the current elapsed time t. Consequently, for given context c and elapsed time t, the probability of each user moving to each possible location s within Δt time is calculated as follow:

$$P(s|c,t) = P(s) P_s(t \leq z < t + \Delta t | c, t) \quad (9)$$

where $P(s)$ is the transition probability of every possible next location s, which can be calculated as follows:

$$P(s_{t+1} = s|\mathcal{H}) \approx \hat{P}(s_{t+1} = s|\mathcal{H}) = \frac{N(cs, \mathcal{H})}{N(c, \mathcal{H})} \quad (10)$$

where $N(cs, \mathcal{H})$ denotes the number of occurrences of cs in the history set $\mathcal{H}$. Accordingly, the Markov predictor predicts the most likely location s will be visited at $t+1$ time as follows:

$$s_{t+1} = \underset{s \in \mathcal{S}}{\operatorname{argmax}}(P(s_{t+1} = s)) \quad (11)$$

4. Current k ($k = 3$) instances are considered.

The algorithm for location prediction is presented in Algorithm 1. It may be noted that the presented algorithm is for a single end-device. However, the path estimator estimates the next locations for all the end-devices associated with different APs in the network in a similar manner.

Algorithm 1: Algorithm for location prediction

Input: History set $\mathcal{H}$, current context c
Output: Next predicted location s_{n+1} at time t_{n+1}
Extract the state of location history set $\mathcal{S}$ from $\mathcal{H}$;
Predict $\mathcal{Z}_s$ at possible locations s from Equation (8);
Calculate $P(s|c,t)$ according to Equation (9);
Predict next location s_{t+1} from Equation (11);
Return s_{n+1};

4.1.1 *Justification of Using Markov Predictor*

In an IoT environment, several users are connected to the Internet through different access points (APs), in order to exchange information with backbone networks. Therefore, each of the APs maintains a flow-table based on the users' requests. Moreover, in a practical scenario, the tables need to be updated periodically, while considering users' movement. We prefer a *proactive* approach to update the tables, with an aim to update them in an adaptive manner, as mentioned in Section 1. Consequently, it is required to predict the future locations of the users, while considering their movement patterns.

Several location prediction algorithms exist in the literature [6], [7], [32] to predict the locations of mobile users in wireless networks. Markov predictor [6] is one of the promising approaches to predict the future locations of the mobile users while considering past locations. Our objective is also to predict the next locations to be visited by the users while considering past visited locations by them. Additionally, Song et. al [33] also proved that the Markov predictor is one of the best location predictors, as it takes into account the past locations visited by the user. Therefore, we use the order-k Markov prediction approach to predict the future locations of the users while considering past locations visited by the them.

4.2 Optimal Rule Placement

After predicting the locations of the users, the flow-manager defines the rules at the APs, while considering the constraints presented in Section 3.4. A user may be situated in the vicinity of multiple APs. However, we need to select the *optimal* one in order to meet their requirements, while minimizing the overall cost. First, we present a cost function, Φ, whose properties are as follows.

$$\frac{\partial\Phi(\mathcal{D},E,\mathcal{R})}{\partial\mathcal{D}} \geq 0, \frac{\partial\Phi(\mathcal{D},E,\mathcal{R})}{\partial E} > 0, \text{ and } \frac{\partial\Phi(\mathcal{D},E,\mathcal{R})}{\partial\mathcal{R}} \geq 0 \quad (12)$$

where $\mathcal{D}$, E, and $\mathcal{R}$ denote the delay, required energy, and number of rules present in the AP, respectively. Consequently, we calculate the cost for all APs present in the network, which is described in the subsequent section.

4.2.1 Cost Calculation

In the flow-rule placement in SDN, two important features are important — delay and number of rules present at the device. Additionally, energy consumption is considered as another feature. In IoT environment, we can have mobile APs which do not run with traditional power sources. In such condition, energy consumption should be taken into account for rule placement. Therefore, we consider the energy consumption as one of the features for cost calculation in flow-rule placement. Consequently, we calculate the cost for rule placement at AP based on network delay, energy consumption and load factor (number of rules present) associated with the AP. Mathematically, it is represented as follows:

$$\Phi^t_{i\in\mathcal{A},j\in U} = \frac{\mathcal{D}^t_{i,j}}{\mathcal{D}^{th}_j} + \frac{E^t_{i,j}}{E^t_{i,avl}} + \gamma\frac{\mathcal{R}^t_{i,+j}}{\mathcal{R}^{max}_i} \quad (13)$$

where $\Phi^t_{i\in\mathcal{A},j\in U}$ denotes the cost of an AP, $i \in \mathcal{A}$ after including the request from user $j \in \mathcal{U}$ at time t. $\mathcal{D}^t_{i,j}$ denotes the service delay to serve the request of the user j at the AP i. $E^t_{i,j}$ and $E^t_{i,avl}$ represent the energy consumption to process the request and available energy at the AP at time t, respectively. $\mathcal{R}^t_{i,+j}$ denotes the total number of flow-rules present at the AP after including the request from the user at time t. γ is a binary variable used to decide the load factor at the AP. Mathematically, it is represented as follows:

$$\gamma = \begin{cases} 1, & \text{if } \frac{\mathcal{R}^t_{i,+j}}{\mathcal{R}^{max}_i} \leq 0.8 \\ 2, & \text{Otherwise} \end{cases} \quad (14)$$

We consider the impact of the number of flow-rules for cost calculation. The value of γ increases when it crosses a predefined value to limit the *over-provisioning* at the APs. Therefore, when an AP is full with 80%[5] of its capacity, the cost for flow-rule placement increases.

We follow the standard approach to calculate the network delay involved between a user and an AP, i.e., the combination of processing, queuing, transmission and propagation delays. For simplicity, we consider that the processing and transmission delays remain the same to a user for all APs present in its vicinity. Consequently, we only consider the impact of propagation delay, which is proportional to the distance between the user and AP, and the queuing delay, which depends on the packet arrival rate and link bandwidth. Therefore, Equation (13) is represented as follows:

$$\Phi^t_{i\in\mathcal{A},j\in U} = \beta^t_{i,j}\left(\frac{\lambda^t_{i,j}}{\mathcal{D}^{th}_j} + \frac{E^t_{i,j}}{E^t_{i,avl}} + \gamma\frac{\mathcal{R}^t_{i,+j}}{\mathcal{R}^{max}_i}\right) \quad (15)$$

where $\lambda^t_{i,j}$ is the combination of propagation and queuing delays between the AP i and the user j at time t. Mathematically, $\lambda^t_{i,j} = \frac{L^t_i}{B_{ij}} + \frac{d^t_{i,j}}{s^t_j}$, where L^t_i denotes the packet arrival rate at the AP i, and B_{ij} denotes the link bandwidth between the AP i and the user j at time t. $d^t_{i,j}$ and s^t_j denote the distance between the AP and the user, and propagation speed, respectively, at time t. In Equation (15), $\beta_{i,j}$ is a binary variable used to define whether a particular user j can be associated with a particular AP i, depending on the predicted location as presented in Section 4.1. Mathematically,

$$\beta^t_{i,j} = \begin{cases} 1, & \text{if } j \rightarrow i \text{ is TRUE}, i \in \mathcal{A} \text{ and } j \in \mathcal{U} \\ 0, & \text{Otherwise} \end{cases} \quad (16)$$

Therefore, the objective of the flow-manager is to associate the flow-rules for users to the APs adequately, so that the overall cost is minimized. If $\beta^t_{i,j} = 0$, we cannot associate the user j with the AP i at time t. Hence, the total cost in the network for all users and APs can be represented as follows:

$$\Phi^t_{\forall i\in\mathcal{A},\forall j\in U} = \sum_{i\in\mathcal{A}}\sum_{j\in U}\beta^t_{i,j}\left(\frac{\lambda^t_{i,j}}{\mathcal{D}^{th}_j} + \frac{E^t_{i,j}}{E^t_{i,avl}} + \gamma\frac{\mathcal{R}^t_{i,+j}}{\mathcal{R}^{max}_i}\right) \quad (17)$$

A user may be associated with multiple APs depending on its position. Therefore, we need to select the *optimal* AP, so that the cost is minimized. Therefore, the flow-manager takes the decision to place flow-rule at the APs for a user $j \in U$ according to the decision matrix as follows.

$$\begin{array}{c} \\ \mathcal{D} \\ E \\ \mathcal{R} \end{array} \begin{array}{c} \begin{array}{ccccc} A_1 & . & . & . & A_i \end{array} \\ \left(\begin{array}{ccccc} D^j_1 & . & . & . & D^j_i \\ E^j_{1,avl} & . & . & . & E^j_{i,avl} \\ \mathcal{R}^j_1 & . & . & . & \mathcal{R}^j_i \end{array}\right) \end{array}$$

Therefore, the flow-manager selects the AP to place flow-rules for which the cost (refer to Equation (15)) is minimized. Algorithm 2 presents the algorithm for rule placement at the APs for a user. This algorithm is repeated for all users present in the network. It is noteworthy that the proposed AP selection method for flow-rule placement is the *best-fit* heuristic as finding optimal solution in polynomial time is NP-hard.

Corollary 2. *There exists a minimum value of the cost function $\Phi^t_{i\in\mathcal{A},j\in U}$ in Equation (15), for any AP, $i \in \mathcal{A}$ [34], while considering the application-specific requirements.*

Assumption 2. *The flow-manager takes decisions after gathering information — requirements and predicted location of user, and delay, energy and residual rule capacity of APs — for a particular user during a time interval t.*

5. In this work, we consider that 80% of the capacity is used. However, it can be any percentage depending on the requirements and deployment strategy.

Algorithm 2: Algorithm for optimal rule placement

Input: **Related to AP:** Set of APs $\mathcal{A}$, Set of positions $\mathcal{P}_A$, Set of available energy E, Maximum rule capacity $\mathcal{R}^{max}$.
Related to User: Set of users $\mathcal{U}$, Predicted location set $\mathcal{P}_U$ obtained from Section 4.1.
Output: Set of APs to be activated $\mathcal{A}_{act}$, set of APs to be deactivated $\mathcal{A}_{dact}$.

```
Assign A_act = NULL;
Assign A_dact = NULL;
for j = 1 to |U| do
    for i = 1 to |A| do
        Calculate the cost Φ^t_{i,j} from Equation (15);
    Select the AP, i ∈ A, for which Φ^t_{i,j} is minimum;
    Insert the AP into A_act, i.e., A_act ∪ = A_act;
    Place the flow-rule associated to user j at the AP;
    Increase the occupied rule capacity for the AP;
Insert rest of the APs A \ A_act into A_dact, i.e.,
A_dact ∪ = A_dact;
```

4.3 Complexity Analysis

The computational complexity of the proposed scheme is analyzed in two phases — location prediction and decision making. In location prediction, we use $O(k)$ Markov predictor, which, in turn, depends on the number of input sequences considered for location prediction. Therefore, the time complexity of the Markov predictor is O(n^2), where n is the number of input sequences considered for location prediction. Therefore, the time complexity for location prediction is O(n^2). According to the Algorithm 2, we find that the time complexity for decision making is O($m \times n$), where m is the number of users and n is the number of APs in the network.

The formulated optimization problem is an instance of the d-capacity bin-packing problem, which is a known to be NP-hard [35]. Therefore, we extend the well-known best fit heuristic to provide an approximate solution to the problem. The proposed algorithm takes user-request $j \in \mathcal{U}$ and attempts to place the associated flow-rule in the AP with the minimum cost (refer to Equation (17)), taking into account the mobility and delay-requirements of users, and available energy and rule-capacity of APs, as presented in Algorithm 2. According to [36], the proposed algorithm has a worst-case approximation ratio of 4, i.e., $\leq d + 1$, where $d = 3$ (rule-capacity, energy and delay).

5 Performance Evaluation

We evaluate the performance of the proposed scheme using NS-3 (http://www.nsnam.org). Different simulation parameters are considered, as listed in Table 2. We consider both small- and large-scale scenarios in an IoT environment by varying number of users. It is noteworthy that two different flows may have the same properties in terms of flow-rules. Consequently, total number of flow-rules is always less than or equal to the total number of flows in the network. Further, we use the D-ITG generator [37] to generate IoT traffic flows according to the real-traces presented in [38].

TABLE 2: Simulation parameters

Parameter	Value
Simulation area	500m $\times$ 500m
Number of APs	25
Number of users	200 – 400
Number of packets	8000 – 16000
Speed of users	0–20 m/s
Deployment strategy of APs	Grid-based topology
Deployment strategy of users	Uniform-Random
Mobility model of APs	Constant Position [42]
Mobility model of users	Gauss Markov [42] Random Way-Point [43]
Simulation time	200 min
IoT traffic generation	D-ITG generator [37], [38]

Therefore, multiple services in terms of flows are considered to simulate IoT applications. We limit our discussion on the traffic generation. Interested readers may refer to [37] and [38].

We compare the proposed scheme, *Mobi-Flow*, with the existing schemes — open shortest path first (OSPF) and minimum occupied rule capacity (MRC) algorithms (as described in [8]), *distributed* (non-SDN), and MoRule [9]. In the OSPF algorithm, the routing of an incoming request is done based on the shortest path principle. Accordingly, the flow-rules associated with the users are placed at the APs in the network. Thus, the AP which is at the minimum distance of a user, flow-rules associated with the user are inserted at the former. On the other hand, in MRC, the flow-rule is placed at the AP, which is minimally occupied with the existing flow-rules. Therefore, the flow-rules associated with a user are placed at the AP with low-occupied capacity among all APs in the vicinity of the former. Further, rules are placed in a distributed manner in case of the *distributed* scheme (non-SDN). In the *distributed* scheme, the users are associated to an AP by following three steps — *probing*, *authentication*, and *association* [39]. We adopt this generic approach to simulate the *distributed* scheme. The association between user and AP takes place in distributed manner depending on received signal strength. In case of MoRule, the flow-rules are inserted at the APs according to users' association probability and available rule-space at the AP.

Additionally, the 'confidence interval' is considered to show the deviation in results for multiple runs. We adopt the use of the 95% confidence interval plot [40], i.e., we are confident that the results are within the specified range in 95% cases. We adopt the format of the flow-rule specified in the OpenFlow protocol [41]. Therefore, we follow the standard approach for deploying the flow-rules at the APs.

5.1 Results and Discussion

We use different performance metrics to show the effectiveness of the proposed scheme — prediction accuracy, number of activated APs, energy consumption, delay, control overhead, and cost — for flow-rule placement at the APs. We discuss the results for each of the performance metrics considered in this work, while comparing it with the existing OSPF and MRC algorithms, *distributed* (non-SDN), and MoRule. The results for *number of activated APs*, *energy consumption*, *delay*, *control overhead*, and *cost* are evaluated after predicting the locations of the end-users for the proposed scheme, *Mobi-Flow*. On the other hand, the results for the same are evaluated without location prediction for the existing schemes — OSPF and MRC — referred as OSPF (w/o LP) and MRC (w/o LP), respectively, in the results. The results for MoRule is evaluated after calculating the association probability of users in the network. Results for *distributed* scheme are obtained using standard association mechanism followed in distributed systems [39].

5.1.1 Prediction Accuracy

As discussed in Section 4.1, we predict the locations of users in the network, based on the location history to determine the APs to be associated with the former. Therefore, we evaluate the prediction accuracy of the proposed scheme. Figure 4 presents the prediction accuracy of the proposed scheme using two different mobility models — Gauss Markov and Random Waypoint — while varying the number of users in the network. We see that *Mobi-Flow* predicts the locations of the users in the network with an average accuracy of 70% for all cases, i.e., in 70% cases, the predicted location is accurate. Therefore, the proposed scheme, *Mobi-Flow*, is capable of predicting the locations of the users in the network, in order to deploy the flow-rules in an adaptive manner. The proposed scheme, Mobi-Flow, predicts the locations with an accuracy of 70% although the network is highly dynamic (i.e., the speed of the end-users is highly dynamic — 0 to 20 m/s). In the subsequent sections, we present the results for the proposed scheme, *Mobi-Flow*, from two different aspects — *MobiFlow-act* (i.e., with 100% location prediction accuracy) and *MobiFlow-pre* (with the predicted accuracy) — to show the effects of wrong location prediction of some of the users (approx. 30%).

5.1.2 Number of Activated APs

The main objective of the proposed scheme is to reduce the number of activated APs in a time period, in order to minimize the overall flow-rule placement cost, as mentioned in Section 3.4. Figure 5 shows the number of activated APs using the proposed scheme, *Mobi-Flow*, compared to the OSPF (w/o LP), MRC (w/o LP), *distributed* (non-SDN), and MoRule at different time instants and with different flows in the network. We see that approximately 70% of the total number of APs are activated using the proposed scheme, while considering the capacity of the APs present in the network. On the other hand, using OSPF (w/o LP), MRC (w/o LP), *distributed* (non-SDN), and MoRule, we see that all the APs are always activated at different time periods. Using OSPF (w/o LP), the APs at the minimum distance are always considered. Therefore, all the APs in the network are activated depending on the distance from users. On the other hand, using MRC (w/o LP), the less occupied APs are activated. Therefore, in every iteration, the APs are selected in a round-robin manner within the vicinity of users. In case of MoRule, the flow-rules are placed according to association probability, due to which all the APs are activated. In case of *distributed* (non-SDN), all the APs are always activated as there is no central coordination present among the APs. We also see that using the predicted accuracy, the number APs are activated (refer *MobiFlow-pre*) is slightly higher than the number of activated APs with the 100% prediction accuracy (refer *MobiFlow-act*). However, it is always less than the existing schemes. Therefore, the proposed scheme outperforms OSPF (w/o LP), MRC (w/o LP), *distributed* (non-SDN), and MoRule schemes. Concurrently, with different number of flows in the network, we see that the number of activated APs is always less using *Mobi-Flow*, compared to OSPF (w/o LP), MRC (w/o LP), *distributed* (non-SDN), and MoRule. Additionally, we see that the number of activated APs increases with an increase in the number of users present in the network due to the limited rule capacity and available energy of the APs.

5.1.3 Energy Consumption

Another objective is to implement the flow-rules in an energy-efficient manner, as largely devices in an IoT environment are energy constrained [44]. Therefore, we present the energy consumption for implementing flow-rules at the APs. Figure 6 presents the energy consumption with different number of flows in the network. It is evident that the proposed scheme, *Mobi-Flow*, reduces the energy consumption significantly, compared to OSPF (w/o LP), MRC (w/o LP), *distributed* (non-SDN), and MoRule. As the number of activated APs is minimized using the proposed scheme, *Mobi-Flow* (see Figure 5), total energy consumption in the network is also minimized. On the other hand, in case of OSPF (w/o LP), MRC (w/o LP), *distributed* (non-SDN), and MoRule, all the APs are activated throughout the simulation period. Therefore, the energy consumption in the network increases for the existing schemes. Consequently, it is also evident that the proposed scheme provides better network lifetime compared to the existing schemes.

5.1.4 Delay and Control Overhead

Figure 7 shows the average delay and control overhead in the network with different number of flows in the network. Similar to Figure 6, we see that the average delay experienced by a flow is minimum using the proposed scheme, *Mobi-Flow*, compared to OSPF (w/o LP), MRC (w/o LP), *distributed* (non-SDN), and MoRule. On receiving a new packet, the APs send *packet-in* messages to the SDN controller. Based on the *packet-in* messages, the controller defines the flow-rules, which is eventually placed at the APs. Typically, 3ms delay is involved in this process [29]. In case of *Mobi-Flow*, the SDN controller places the flow-rules in a proactive manner, depending on the predicted locations of the users in the network. Therefore, the proposed scheme is capable of saving the required 3ms time. In contrast, using OSPF and MRC, the packets experienced this 3ms delay, as the flow-rules required to be placed at the APs in real-time. In case of the *distributed* scheme, path calculation is done before forwarding the received information, as there is no central coordination present in distributed approach. Further, in MoRule, the delay is higher compared to the proposed scheme, as some of the requests are served by remote controller. The average delay incurred by the packets with predicted locations of users is more compared to the delay with 100% location prediction accuracy. It is due to the fact that additionally few number of flow-rules are required to be placed at the APs in real-time. However, it is always better than OSPF (w/o LP), MRC (w/o LP), *distributed* (non-SDN), and MoRule. The average delay increases with an

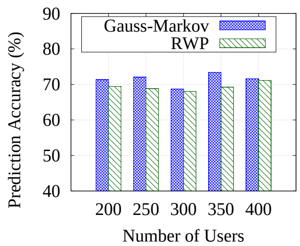

Fig. 4: Prediction accuracy

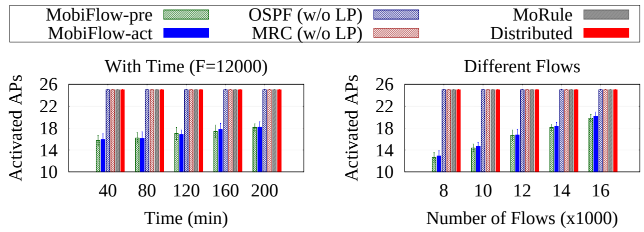

Fig. 5: Number of activated APs in the network

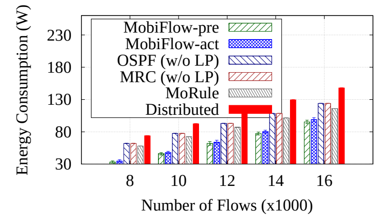

Fig. 6: Energy consumption in the network

increase in the number of flows in the network as the number of packets queued at the APs is high.

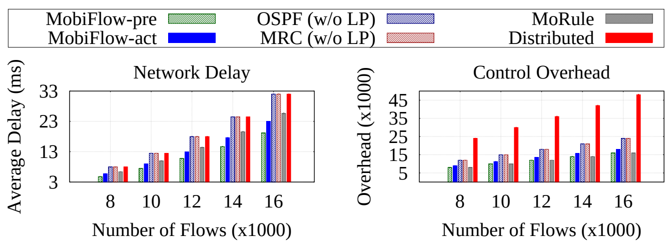

Fig. 7: Average delay and control overhead in the network

We also evaluate the control overhead associated in the rule placement using the proposed scheme (Mobi-Flow), OSPF (w/o LP), MRC (w/o LP), *distributed* (non-SDN), and MoRule, as depicted in Figure 7. We see that the control overhead using Mobi-Flow is higher than the MoRule, as additional messages are exchanged between AP and SDN controller due to wrong location prediction (see Figure 4). However, it is always lower than the other existing schemes — OSPF (w/o LP), MRC (w/o LP), and *distributed* (non-SDN). In case of OSPF (w/o LP) and MoRule (w/o LP), the flow-rule associated to an existing request from a user, who just come in the vicinity of the AP, is not present at the latter. Consequently, the AP sends *packet-in* messages to the SDN controller to serve requests from users, who were not in the vicinity of the AP before. As a result, the control overhead increases using the OSPF (w/o LP) and MoRule (w/o LP) without location prediction. In contrast, APs do not send *packet-in* messages to the SDN controller as the latter places the flow-rules in an adaptive manner, while predicting the locations of the users. Thus, the proposed scheme, MobiFlow, is capable of reducing the control overhead in the network. Further, in the *distributed* (non-SDN) scheme, control messages related to *probing*, *authentication*, and *association* are exchanged between the user and AP before the final association is done, which, in turn, increases the control overhead in *distributed* (non-SDN) approaches.

5.1.5 Cost

Finally, we present the cost associated to flow-rule placement at the APs with different time and flows in the network, as depicted in Figure 8. We see that the overall cost for flow-rule placement is minimized using the proposed scheme, *Mobi-Flow*, compared to OSPF (w/o LP), MRC (w/o LP), *distributed* (non-SDN), and MoRule. Further, we also see that the total cost for flow-rule placement with different number of flows in the network is minimized using the proposed scheme compared to the existing schemes. We see that the flow-rule placement cost increases with an increase in the number of users, as the number of APs required to be activated is high to fulfill the requirements (see Figure 5).

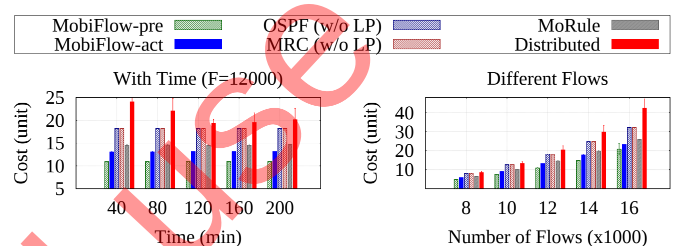

Fig. 8: Flow-rule placement cost in the network

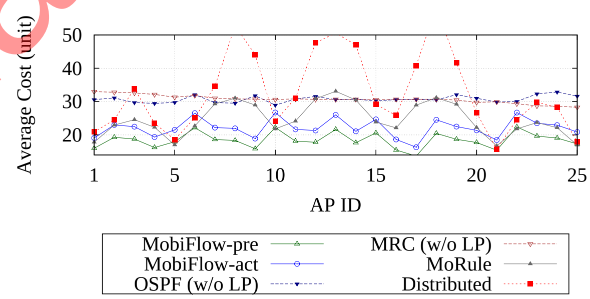

Fig. 9: Flow-rule placement cost at APs with F = 12000

Further, in an IoT environment, the APs can be owned by multiple service providers like the resources are virtualized in a cloud platform. In SDN, the controller can control all the APs in a centralized manner, while the APs are owned by multiple service providers. Therefore, we also present the flow-rule placement cost at different APs using the proposed scheme, while comparing it with the OSPF (w/o LP), MRC (w/o LP), *distributed* (non-SDN), and MoRule. It is evident that the proposed scheme, *Mobi-Flow*, is capable of inserting the flow-rules at the APs in a cost-effective manner compared to the existing schemes.

In summary, we see that the proposed scheme outperforms the existing SDN-based schemes such as OSPF (w/o LP), MRC (w/o LP), and MoRule, and *distributed* (non-SDN) approach. Moreover, it is also evident that the SDN-based approaches yield improved performance compared to the *distributed* (non-SDN) approach, as shown in the results. From the results, it is evident that the proposed scheme can significantly improve the system performance by reducing delay, control overhead, number of activated APs, and cost, while placing the flow-rules in an adaptive manner according to users' movement in the network.

6 Limitations

The proposed scheme is capable of minimizing the energy consumption, delay, and cost for flow-rule placement, as depicted in Figures 6, 7, 8, and 9. Further, it is also capable of minimizing control overhead in certain cases. However, few limitations of the proposed scheme are as follows:

• *Redundant flow-rules*: In the proposed scheme, we place the flow-rules at APs according to the predicted location of the users. However, due to the wrong location prediction (refer to Figure 5), some of the flow-rules are again placed at the suitable APs in real-time. As a result, some of the rules are redundant at multiple APs due to wrong location prediction. These redundant rules are removed from the APs according to the *timeout* defined in OpenFlow protocol.

• *Increased control overhead*: The location prediction involves multiple information to be collected by the SDN controller. Although the information can be collected without adding additional overhead (refer to Assumption 1), due to wrong location prediction, the APs generate few additional packet-in messages for which the control overhead increases compared to MoRule, as depicted in Figure 7.

7 Use-Case: Practical Application

In this section, we briefly discuss two use-cases — software-defined smart grid and healthcare management systems — to realize the practical applications of the proposed scheme. It is noteworthy that the proposed scheme does not introduce any client-side changes. Let focus on the two use-cases in which the proposed scheme can be applied to improve the performance.

• *Software-defined smart grid system*: In smart grid, billions of smart meters (SMs) are deployed at the energy distribution side along with plug-in electric vehicles (PHEVs), which report their energy consumption to the main grid through meter data management systems (MDMS). Further, PHEVs are mobile in nature, which, in turn, establishes a semi-mobile smart grid environment. Typically, the communication between MDMS and users (SMs and PHEVs) are done in wireless mode. Therefore, adequate flow-rules are required to be placed at the MDMS to reduce the cost and network delay, while considering the PHEVs' movement. In such a scenario, the proposed scheme is capable of reducing the cost and delay, while placing the flow-rules at the MDMS in an adaptive manner. It is noteworthy that the MDMSs are considered as APs, and SMs and PHEVs are considered as users in the system.

• *Software-defined healthcare management system*: In healthcare management system, network delay is a primary concern. The proposed scheme is useful to reduce the network delay by placing the flow-rules in an adaptive manner according to the requirements of the users, as depicted in Figure 7, while reducing the overall cost in rule placement. Therefore, prioritized services can also be provided depending on the characteristics of incoming traffic.

8 Conclusion

In this paper, we proposed mobility-aware flow-rule placement scheme, named as *Mobi-Flow*, in SDAN to support IoT applications. The proposed scheme consists of two components — path estimator and flow-manager. The path estimator predicts the locations of users in the network, depending on their history location sets, in order to deploy the flow-rules in an adaptive manner. Based on the predicted locations, the flow-manager determines the optimal number of APs to be activated, so that the total cost for flow-rule placement is reduced, while considering the users' requirements. Through extensive simulations, we see that the proposed scheme, *Mobi-Flow*, is capable of minimizing delay, number of activated APs, control overhead, energy consumption, and cost compared to the existing schemes. Specifically, the cost for flow-rule placement is reduced approximately by 39%, 38%, 65%, and 11% compared to the OSPF (w/o LP), MRC (w/o LP), *distributed* (non-SDN), and MoRule, respectively.

In this work, we considered that the entire network is enabled with SDN. However, in a practical scenario, part of the network may not be SDN enabled. Therefore, we intend to analyze the complexity of flow-rule placement in semi-SDN-enabled network in the future. Additionally, we showed that the proactive approach is useful for flow-rule placement at the APs while predicting the locations of end-users. However, a *hybrid* approach (combination of proactive and reactive) may be useful depending on the network dynamics. Therefore, we also plan to study the impact of the *hybrid* approach for flow-rule placement in an SDN-enabled network. Additionally, emulation-based performance evaluation is also included as a future extension of the work.

Acknowledgments

A preliminary version of this work is published in IEEE GLOBECOM 2016, Washington, DC USA (December 4–8, 2016), DOI: 10.1109/GLOCOM.2016.7841995.

References

[1] B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Comm. Surveys and Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.

[2] K. Sood, S. Yu, and Y. Xiang, "Software-Defined Wireless Networking Opportunities and Challenges for Internet-of-Things: A Review," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 453–463, 2015.

[3] S. Bera, S. Misra, and A. V. Vasilakos, "Software-Defined Networking for Internet of Things: A Survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1994–2008, 2017.

[4] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks (Elsevier)*, vol. 54, no. 15, pp. 2787–2805, 2010.

[5] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, "A Software Defined Networking architecture for the Internet-of-Things," in *Proc. of the IEEE Network Operations and Management Symposium*, Krakow, May 2014, pp. 1–9.

[6] L. Song, U. Deshpande, U. C. Kozat, D. Kotz, and R. Jain, "Predictability of WLAN Mobility and Its Effects on Bandwidth Provisioning," in *Proc. of the IEEE INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–13.

[7] Y. Chon, E. Talipov, H. Shin, and H. Cha, "SmartDC: Mobility Prediction-Based Adaptive Duty Cycling for Everyday Location Monitoring," *IEEE Trans. on Mobile Computing*, vol. 13, no. 3, pp. 512–525, 2014.

[8] A. Amokrane, R. Langar, R. Boutaba, and G. Pujolle, "Flow-Based Management For Energy Efficient Campus Networks," *IEEE Trans. on Network and Service Management*, vol. 12, no. 4, pp. 565–579, 2015.

[9] H. Li, P. Li, and S. Guo, "MoRule: Optimized rule placement for mobile users in SDN-enabled access networks," in *Proc. of the IEEE GLOBECOM*, TX, Dec. 2014, pp. 4953–4958.

[10] X.-N. Nguyen, D. Saucez, C. Barakat, and T. Turletti, "Rules Placement Problem in OpenFlow Networks: A Survey," *IEEE Comm. Surveys & Tutorials*, vol. 18, no. 2, pp. 1273–1286, 2016.

[11] F. Giroire, J. Moulierac, and T. K. Phan, "Optimizing rule placement in software-defined networks for energy-aware routing," in *Proc. of the IEEE GLOBECOM*, TX, Dec. 2014, pp. 2523–2529.

[12] A. Markiewicz, P. N. Tran, and A. Timm-Giel, "Energy consumption optimization for software defined networks considering dynamic traffic," in *Proc. of the IEEE Intl. Conf. on CloudNet*, Luxembourg, Oct. 2014, pp. 155–160.

[13] I. Vawter, D. Pan, and W. Ma, "Emulation Performance Study of Traffic-Aware Policy Enforcement in Software Defined Networks," in *Proc. of the IEEE Intl. Conf. on Mobile Ad Hoc and Sensor Systems*, Philadelphia, PA, Oct. 2014, pp. 775–780.

[14] J.-F. Huang, G.-Y. Chang, C.-F. Wang, and C.-H. Lin, "Heterogeneous Flow Table Distribution in Software-defined Networks," *IEEE Trans. on Emerging Topics in Computing*, vol. 4, no. 2, pp. 252–261, 2016.

[15] W. Ma, C. Medina, and D. Pan, "Traffic-Aware Placement of NFV Middleboxes," in *Proc. of the IEEE GLOBECOM*, San Diego, CA, Dec. 2015, pp. 1–6.

[16] X.-N. Nguyen, D. Saucez, C. Barakat, and T. Turletti, "OFFICER: A general Optimization Framework for OpenFlow Rule Allocation and Endpoint Policy Enforcement," in *Proc. of the IEEE INFOCOM*, Hong Kong, Apr–May 2015, pp. 478–486.

[17] M. Caria, A. Jukan, and M. Hoffmann, "SDN Partitioning: A Centralized Control Plane for Distributed Routing Protocols," *IEEE Trans. on Network and Service Management*, vol. 13, no. 3, pp. 381–392, 2016.

[18] K. J. Kerpez, J. M. Cioffi, G. Ginis, M. Goldburg, S. Galli, and P. Silverman, "Software-Defined Access Networks," *IEEE Comm. Magazine*, vol. 52, no. 9, pp. 152–159, 2014.

[19] S. H. Rastegar, A. Abbasfar, and V. Shah-Mansouri, "On Fair Rule Caching in Software Defined Radio Access Networks," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 460–463, Jun. 2018.

[20] J. Liu, Y. Li, M. Chen, W. Dong, and D. Jin, "Software-defined internet of things for smart urban sensing," *IEEE Comm. Magazine*, vol. 53, no. 9, pp. 55–63, 2015.

[21] A. Hakiri, P. Berthou, A. Gokhale, and S. Abdellatif, "Publish/subscribe-enabled software defined networking for efficient and scalable IoT communications," *IEEE Comm. Magazine*, vol. 53, no. 9, pp. 48–54, 2015.

[22] A.-C. G. Anadiotis, G. Morabito, and S. Palazzo, "An SDN-Assisted Framework for Optimal Deployment of MapReduce Functions in WSNs," *IEEE Trans. on Mobile Computing*, vol. 15, no. 9, pp. 2165–2178, 2016.

[23] F. Idzikowski, L. Chiaraviglio, A. Cianfrani, J. L. Vizcano, M. Polverini, and Y. Ye, "A Survey on Energy-Aware Design and Operation of Core Networks," *IEEE Comm. Surveys & Tutorials*, vol. 18, no. 2, pp. 1453–1499, 2015.

[24] C. Xu, W. Jin, G. Zhao, H. Tianfield, S. Yu, and Y. Qu, "A Novel Multipath-Transmission Supported Software Defined Wireless Network Architecture," *IEEE Access*, vol. 5, pp. 2111–2125, 2017.

[25] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards programmable enterprise WLANS with Odin," in *Proc. of the ACM Workshop on HotSDN*, Helsinki, Finland, Aug. 2012, pp. 115–120.

[26] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, and S. Shenker, "Onix: a distributed control platform for large-scale production networks," in *Proc. of the OSDI*, Oct. 2010, pp. 351–364.

[27] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "ONOS: towards an open, distributed SDN OS," in *Proc. of the ACM HotSDN*, Aug. 2014, pp. 1–6.

[28] C.-T. Chan, P.-C. Wang, S.-C. Hu, C.-L. Lee, and R.-C. Chen, *Information Networking: International Conference, ICOIN 2003.Revised Selected Papers*. Springer, 2003, ch. Scalable Packet Classification for IPv6 by Using Limited TCAMs, pp. 76–85.

[29] K. He, J. Khalid, S. Das, A. Gember-Jacobson, C. Prakash, A. Akella, L. E. Li, and M. Thottan, "Latency in Software Defined Networks: Measurements and Mitigation Techniques," in *Proc. of the ACM SIGMETRICS Intl. Conf. on Measurement and Modeling of Computer Systems*, Oregon, USA, Jun. 2015, pp. 435–436.

[30] C. Yu, C. Lumezanu, Y. Zhang, ishal Singh, G. Jiang, and H. V. Madhyastha, "FlowSense: monitoring network utilization with zero measurement cost," in *Proc. of the Intl. Conf. on Passive and Active Measurement*, Hong-kong, China, Mar. 2013, pp. 31–41.

[31] R. Ahlswede, L. Bumer, N. Cai, H. Aydinian, V. Blinovsky, C. Deppe, and H. Mashurian, Eds., *General Theory of Information Transfer and Combinatorics*. Springer, 2006.

[32] C. Cheng, R. Jain, and E. van den Berg, *Wireless internet handbook*. CRC Press, 2003, ch. Location prediction algorithms for mobile wireless systems, pp. 245–263.

[33] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating location predictors with extensive Wi-Fi mobility data," in *Proc. of the IEEE INFOCOM*, March 2004, pp. 1414–1424.

[34] D. G. Luenberger and Y. Ye, *Linear and nonlinear programming*. Springer, 2008.

[35] W. Leinberger, G. Karypis, and V. Kumar, "Multi-capacity bin packing algorithms with applications to job scheduling under multiple constraints," in *Proc. of Intl. Conf. on Parallel Processing*, Japan, Aug. 1999, pp. 1–9.

[36] M. R. Garey and R. L. Graham, "Bounds for Multiprocessor Scheduling with Resource Constraints," *SIAM J. Comput.*, vol. 4, no. 2, pp. 187–200, 1975.

[37] A. Botta, A. Dainotti, and A. Pescap, "A tool for the generation of realistic network workload for emerging networking scenarios," *Computer Networks (Elsevier)*, vol. 56, no. 15, pp. 3531–3547, 2012.

[38] A. Sivanathan, D. Sherratt, H. H. Gharakheili, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Characterizing and classifying IoT traffic in smart cities and campuses," in *Proc. of the IEEE INFOCOM Workshop*, May 2017, pp. 559–564.

[39] "802.11 Network Security Fundamentals, Cisco Secure Services Client Administrator Guide, Release 5.1," Cisco, Tech. Rep., Accessed on Jul. 31, 2018.

[40] A. Hackshaw, *A Concise Guide to Clinical Trials*. Oxford, UK: BMJ, 2009, ch. Statistical formulae for calculating some 95% confidence intervals.

[41] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. R. andScott Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," in *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, Apr. 2008, pp. 69–74.

[42] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.

[43] G. Jayakumar and G. Ganapathi, "Reference point group mobility and random waypoint models in performance evaluation of MANET routing protocols," *Journal of Comp. Systems, Networks, and Communications*, p. 10, 2008.

[44] R. Buyya and A. V. Dastjerdi, Eds., *Internet of Things: Principles and Paradigms*. Elsevier, 2016.