QoS-Aware Dynamic Flow Management in Software-Defined Data Center Networks

Ayan Mondal[1,*] and Sudip Misra[2]

[1,*] Department of Computer Science
University of Rennes, INRIA, CNRS, IRISA, France
ayan.mondal@irisa.fr
[2] Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur, India
smisra@cse.iitkgp.ac.in

Abstract. In this work, the problem of unbalanced data traffic in the presence of heterogeneous Internet of things (IoT) applications in software-defined data center networks (DCNs) is studied. In the existing literature, the presence of heterogeneous flows and switches and mobile IoT devices in software-defined DCN are not considered. Due to heterogeneity, it becomes an NP-hard problem. To address these issues in polynomial time, we proposed a data traffic management scheme, named FASCES, using the single-leader-multiple-followers Stackelberg game. In FASCES, each controller acts as the leader and the IoT applications act as the followers. Each leader and follower aim to achieve an optimal distribution of flow rules and optimal datarate, respectively. We also evaluate the existence of at least one Stackelberg equilibrium solution in FASCES. Furthermore, we evaluated the performance of FASCES while comparing it with the existing schemes through simulation. We observe that FASCES ensures a 16.67-19.45% increase in network throughput and a 4.34-9.43% reduction in network delay. Additionally, using FASCES, the per-flow delay reduces by 27.78-36.67% while ensuring a 15.37-26.91% increase in the per-flow throughput.

Keywords: IoT, Load Balancing, Software-Defined Networks, Data Center Networks, Heterogeneous Flow, Heterogeneous Switches, Stackelberg game.

1 Introduction

With technological advancement, the Internet of things (IoT) devices are capable of generating a huge amount of data, which are to be stored in data centers or managed by the backbone of the data center networks (DCNs). Additionally, these IoT devices are heterogeneous in terms of computation and memory capacity. As a result, these devices are capable of handling heterogeneous applications in terms of datarate requirements. To handle these heterogeneous applications, we envision using the fat-tree architecture-based DCN, which enables to reduce the single point failure. On the other hand, we consider that software-defined networking (SDN) is one of the promising technology which can enable the fat-tree DCN for balanced data traffic in the presence of heterogeneous applications. In SDN, due to having a centralized overview of the network, the network failures also can be reduced. However, in the existing literature, the heterogeneity of the flows while designing the schemes for software-defined DCNs is not considered.

Software-defined DCN is an integrating architecture of fat-tree DCN, which follows a hierarchical architecture, and SDN. We envision that instead of having a single controller for the overall network, each pod has a dedicated controller in addition to the centralized controller. Thereby, it reduces the load on the centralized controller and also helps in the efficient management of the network. In the existing literature, researchers focused on designing schemes, viz. [15, 28] for the data management and flow rule placement. However, none of these schemes consider the presence of heterogeneous flows in the network. Additionally, few works, viz. [20], focused on managing the heterogeneous flows in SDN. However, these works are capable of providing a local solution and cannot ensure balanced traffic distribution globally. On the other hand, few data transmission schemes, viz. [18], for DCN are proposed by the researchers. However, none of the schemes consider the heterogeneity among the switches and the presence of SDN architecture. Therefore, we argue that we need a design of an efficient data flow management scheme for software-defined DCN while considering the quality of service (QoS) parameters such as per-flow throughput and delay, and overall network throughput and delay.

In this work, we design a QoS-aware flow management scheme, named FASCES, for software-defined DCN to ensure that heterogeneous applications generated by the IoT devices are served efficiently while allocating the network resources dynamically for each application. We use a single-leader-multiple-followers Stackelberg game to design the scheme - FASCES. In FASCES, each controller acts as the leader and decides the flow rule association among the incoming flows and the available switches. On the other hand, the IoT applications are considered to be the followers in FASCES. These followers aim to achieve a high datarate while satisfying a delay bound, which depends on the type of applications. To summarize, the contributions of this chapter are as follows:

(i) We design a dynamic flow management scheme, named FASCES, for software-defined DCN in the presence of mobile IoT devices, while ensuring high QoS in terms of throughput and delay.
(ii) We use a single-leader-multiple-follower Stackelberg game to design the interactions between the IoT applications and the controllers. We also evaluate the existence of the Stackelberg equilibrium for FASCES. Using FASCES, we eventually obtain an optimal distribution of flows in the software-defined DCN and optimal datarate of the IoT applications.
(iii) We evaluate the performance of FASCES in terms of per-flow throughput and delay, and overall network throughput and delay, while comparing with the existing schemes.

The rest of the chapter is organized as follows. In Section 2, we briefly present the related works in the area of resource management in SDN as well DCN, and identified the lacuna in the existing works. The system model and the proposed FASCES scheme are described in Sections 3 and 4, respectively. Thereafter, we analyze the performance of FASCES in Section 5 while comparing with the existing schemes through simulation. Finally, we conclude the chapter while citing a few future directions in Section 6.

2 Related Works

In this section, we survey the related literature on traffic engineering schemes for DCNs and SDNs in detail. The existing literature related to traffic engineering of SD-DCNs is divided into two categories — resource management in SDNs and DCNs.

Resource Management in SDNs: In the existing literature, Bera *et al.* [6] studied different aspects of resource allocation in SDN for IoT. Saha *et al.* [28] proposed a flow-rule aggregation scheme for SDN, while focusing on the problem of over-subscription. The authors used a key-based aggregation policy to reduce the number of flow rules. In another work, Maity *et al.* [15] proposed a tensor-based flow-rule aggregation scheme in SDN. Sadeh *et al.* [26] designed a flow-traffic aware rule placement scheme while reducing the usage of TCAM space. On the other hand, an optimal multipath flow management scheme is proposed by Rottenstreich *et al.* [25] while considering network heterogeneity in terms of network path in SDN. Mondal *et al.* [21] modeled a data traffic management scheme while considering that the data volume associated with the flows is known *a priori.* an SDN-based network storage scheme is proposed by Wang *et al.* [34] in the absence of any physical storage. For reducing the usage of oversubscribed buffer, Li *et al.* [14] suggested not to store the entire packet at the switch but only the packet header. Hayes *et al.* [13] studied the traffic-classification in SDN. In another work, Saha *et al.* [27] proposed a QoS-aware routing scheme for SDN, while maximizing end-to-end delay and considered different types of flows in terms of delay- and loss-sensitivity. Bera *et al.* [5] studied a mobility-aware SDN and attempted to maximize the overall network performance.

Having a centralized overview of the network in SDN, controller can reduce the packet drop and delay while ensuring efficient data traffic management [1]. Tseng *et al.* [32] designed a scheme for ensureing path stability in hybrid SDN. In this work, initially, the paths are calculated distributively and locally while reducing the computational complexity. Thereafter, the paths are re-evaluated centrally to ensure high stability. Misra and Bera [16] proposed a task offloading scheme for an SDN-based fog network. The authors minimized the delay in task offloading and computation while selecting the optimal number of fog nodes. Singh *et al.* [30] proposed a hash-based flow-table to reduce the flow-table lookups. In another work, Aujla *et al.* [4] proposed a traffic flow management scheme in SDN. Moreover, a traffic engineering scheme is proposed by Moradi *et al.* [23] for SDN-based ISP networks in the presence of heterogeneous links and switches. A fair resource allocation scheme is designed by Allybokus *et al.* [3]

in multipath SDN. Sanvito *et al.* [29] also proposed a flow-table reconfiguration scheme, while considering overlapping data flow paths.

Resource Management in DCNs: In existing literature, researchers studied *Fat-tree DCNs* [2], [12]. The different challenges of DCN such as generation, processing, and storage of data is studied by Chen *et al.* [11] in the presence of various applications such as social networks, healthcare, smart grid, and managing enterprises. Similarly, Chakraborty *et al.* designed schemes for provisioning sensor-based services in data center networks while considering economic aspect [7, 9] and resource orchestration [8, 17, 19]. A network selection scheme for multimedia data delivery in ad-hoc networks proposed by Trestian *et al.* [31]. Moreover, the optimal server positioning scheme is proposed by Paul *et al.* [24] while minimizing the maintenance cost.

Synthesis

Based on the study of the existing works, we observe that a few schemes are proposed for data traffic management in SDN as well as DCN. However, the researchers have not considered the presence of heterogeneous applications and switches in the existing literature. Additionally, efficient management of heterogeneous data traffic in software-defined DCN while ensuring optimized QoS in terms of high throughput and low delay is one of the important aspects which needs to be addressed.

3 System Model

We consider an SDN-enabled Fat-Tree architecture of DCN [18]. A general fat-tree architecture is composed of three layers – edge, aggregation, and core layers, which enables reducing the bottleneck in transmission as well as is capable of handling the single point failures. Additionally, we consider a multi-tier SDN, where there is a dedicated SDN controller for each pod at the aggregation layer. The switches at the aggregation layer are connected with the switches at the core layers. We consider that the switches at the core layer are managed by a single controller. Moreover, in this work, we consider that the IoT devices at the edge layer are mobile, and are connected to the switches at the aggregation layer through the access points. The system architecture is depicted in Figure 1. The IoT devices are capable of executing heterogeneous applications having different datarate.

Each application a_n of IoT devices $n \in N$, where N is the set of IoT devices at the edge layer, denotes a separate flow[1] and has a datarate r_a and connected with a switch s. These switches at the aggregation layers, where the set of switches is denoted by S_2, are connected with the set of switches at the core layer, which is

[1] For the rest of the chapter, we use a_n to denote flow or application a generated from IoT device n, synonymously.

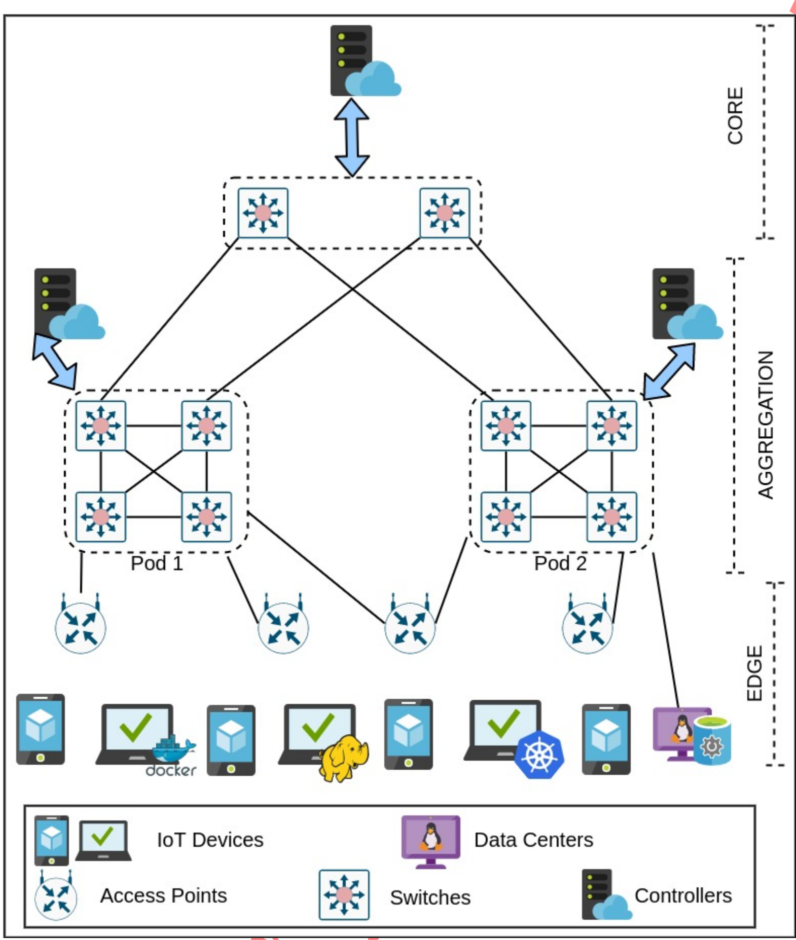

Fig. 1: Schematic Diagram of Software-Defined Fat Tree DCN

denoted by S_1. We consider that each switch at the aggregation and core layers is heterogeneous in terms of bandwidth and TCAM. Furthermore, in addition to the IoT devices, we also consider the presence of servers at the edge layer.

To achieve a high throughput with an optimal delay, we need to ensure a balanced data traffic in the network. Considering that each switch s has a limited capacity of B_s and there are A_s set of flows associated with switch s, where $\forall s \in S_1 \cup S_2$, the following constraint needs to be satisfied:

$$B_s \leq \sum_{a_n \in A_s} r_a \quad (1)$$

On the other hand, each application a_n has a delay threshold d_a^{th}. Hence, while allocating the flows to the switches, the following constraint also needs to be satisfied.

$$d_a^{th} \leq \sum_{s \in P_a} d_a^s \tag{2}$$

where P_a denotes the set of switches associated with the flow a_n; and d_a^s represent the delay at switch s for handling flow a_n. We consider that for handling each flow, the switches follows first-in-first-out (FIFO) scheduling and requires a fixed duration Δ. Hence, for processing a single packet, we get:

$$d_a^s = \sum_{s \in P_a} \sum_{a_n \in A_s} \Delta \tag{3}$$

Additionally, due to limited TCAM, the maximum number of flows can be handled by each switch s is denoted by M_s and must satisfy the following constraint:

$$M_s \geq |A_s| \tag{4}$$

Hence, to serve a high number of flows, the controllers can request to reduce the associated datarate for each application a_n, however, each IoT device n needs to ensure that minimum datarate r_a^{min} is achieved, i.r., $r_a \geq r_a^{min}$.

4 FASCES: QoS-Aware Dynamic Flow Management Scheme

We propose a single-leader-multiple-followers Stackelberg game for studying the interaction between the IoT applications and the controllers in software-defined data center networks. The Stackelberg game is a non-cooperative game that deals with the interaction among the leaders and followers. In FASCES, the controller acts as the leader, and the IoT application act as the followers. In this work, the controllers at the aggregation layer deal with the IoT applications directly. However, the controller at the core layer needs to interact with the controllers at the aggregation layer. Hence, the decision of each leader at the aggregation layer is always influenced by the decision of the controllers at the core layer. In the proposed game, the leaders aim to maximize their utility values while maximizing the bandwidth utilization with optimal delay and maximizing the number of applications served. On the other hand, the followers aim to maximize their utility while obtaining a high datarate with less delay. Therefore, the components of FASCES are as follows:

(i) Each controller act as the leader. The utility of each controller at the aggregation layer is influenced by the decision of the controller at the core layer. The decision of the controllers are executed by the switches, hence the switches are not considered active players in the proposed game.
(ii) Each IoT application acts as the follower and decides the required datarate. The maximum datarate can be achieved by each application depends on the hosted IoT device.
(iii) The IoT applications run for a finite duration which is not known *a priori*.

4.1 Single-Leader-Multiple-Followers Stackelberg Game: The Justification

The proposed system comprising of the fat-tree DCN and the SDN follows a hierarchical architecture. The different entities such as the controllers and IoT applications, are non-cooperative and aim to maximize their payoff values. This results in a 'oligopolistic market' scenario [10]. On the other hand, Stackelberg game is the most suitable game-theoretic approach to model a hierarchical system with non-cooperative players. Hence, we propose to use the single-leader-multiple-followers Stackelberg game for designing the FASCES scheme.

4.2 Game Formulation

To model the game-theoretic interactions in FASCES, we design two utility functions for the controllers and the IoT applications, which are discussed as follows.

Utility Function of Each IoT Application

The utility function $U_a(\cdot)$ of each IoT application signifies the satisfaction of the end-users in data transmission. Each application a_n needs to finalize an optimal datarate r_a^* to ensure that the associated flow rule is active. Considering that each switch $s \in P_a$ handles A_s set of applications, the optimal datarate of flow r_a^* depends on $r_{-\boldsymbol{a}}^*$, where $r_{-\boldsymbol{a}}^* = A_s \setminus \{a_n\}$. This is because the IoT applications are non-cooperative. Therefore, the utility function $U_a(r_a, r_{-\boldsymbol{a}}, A_s, P_a)$ of each IoT application a_n of IoT device n needs to satisfy the following constraints:

(i) Each IoT device aims to achieve the maximum datarate r_a^{max}, where $r_a \leq r_a^{max}$. Therefore, the utility function $U_a(r_a, r_{-\boldsymbol{a}}, A_s, P_a)$ is a non-decreasing function. Mathematically,

$$\frac{\partial U_a(r_a, r_{-\boldsymbol{a}}, A_s, P_a)}{\partial r_a} \geq 0 \tag{5}$$

(ii) The payoff value of $U_a(r_a, r_{-\boldsymbol{a}}, A_s, P_a)$ decreases on increasing the datarate beyond the optimal value. Therefore, in the marginal condition, $U_a(r_a, r_{-\boldsymbol{a}}, A_s, P_a)$ is considered to be a non-increasing function. Mathematically,

$$\frac{\partial^2 U_a(r_a, r_{-\boldsymbol{a}}, A_s, P_a)}{\partial (r_a)^2} < 0 \tag{6}$$

(iii) With the increase in the number of applications, i.e., $|A_s|$ managed by each switch s, the probability of flow rule replacement increases. Hence, the payoff of the utility function $U_a(r_a, r_{-\boldsymbol{a}}, A_s, P_a)$ decreases with the increase in $|A_s|$. Mathematically,

$$\frac{\partial U_a(r_a, r_{-\boldsymbol{a}}, A_s, P_a)}{\partial |A_s|} < 0, \quad \forall s \in P_a \tag{7}$$

Therefore, motivated by the work of Tushar *et al.* [33], in FASCES, the utility function $U_a(r_a, r_{-\boldsymbol{a}}, A_s, P_a)$ of IoT application a_n is represented as follows:

$$U_a(r_a, r_{-\boldsymbol{a}}, A_s, P_a) = r_a^{max} r_a - \left(\frac{r_a^{min}}{r_a^{max}}\right) {r_a}^2 - r_a \frac{\sum_{s\in P_a} |A_s|}{|P_a|} \tag{8}$$

In FASCES, each IoT application aims to maximize the payoff of $U_a(r_a, r_{-\boldsymbol{a}}, A_s, P_a)$, while deciding an optimal datarate. Mathematically,

$$\arg\max_{r_a} U_a(r_a, r_{-\boldsymbol{a}}, A_s, P_a) \tag{9}$$

while satisfying the constraints mentioned in Equations (1) and (2).

Utility Function of Each Controller

The utility function $B_c(r_a, r_{-\boldsymbol{a}}, A_s)$ of each controller c signifies the utilization of the switch capacity B_s. The controllers aim to maximize the the set of applications served as well as maximize the bandwidth allocated to each applications or flows. Therefore, the utility function $B_c(r_a, r_{-\boldsymbol{a}}, A_s)$ of each controller c needs to satisfy the following constraints:

(i) Each controller tries to allocate high bandwidth possible to ensure high utilization of its capacity. Mathematically,

$$\frac{\partial B_c(r_a, r_{-\boldsymbol{a}}, A_s)}{\partial r_a} \geq 0 \tag{10}$$

(ii) The overall objective of the controllers is to accommodate high number of flows, while satisfying the physical limitations of the switches. Mathematically,

$$\frac{\partial B_c(r_a, r_{-\boldsymbol{a}}, A_s, P_a)}{\partial |A_s|} > 0 \tag{11}$$

Therefore, we design the utility function $B_c(r_a, r_{-\boldsymbol{a}}, A_s)$ of each controller c as follows:

$$\arg\max_{\sum r_a, A_s} B_c(r_a, r_{-\boldsymbol{a}}, A_s) \tag{12}$$

Each controller c aims to maximize the payoff of $B_c(r_a, r_{-\boldsymbol{a}}, A_s)$ while satisfying the constraints in Equations (1) and (4).

4.3 Existence of Equilibrium

In this section, we evaluate the existence of the Stackelberg equilibrium, defined in Definition 1, for FASCES in Theorem 1.

Definition 1 *In FASCES, the Stackelberg equilibrium is denoted as a tuple of* $\langle r_a^*, A_s^* \rangle$*, where* r_a^* *and* A_s^* *represent the optimal datarate for application* a_n *and the optimal set of flows associated with switch* $s \in S1 \cup S_2$*. The equilibrium condition also needs to satisfy the following constraints:*

$$U_a(r_a^*, r_{-\boldsymbol{a}}^*, A_s^*, P_a^*) \geq U_a(r_a, r_{-\boldsymbol{a}}^*, A_s^*, P_a^*) \tag{13}$$

$$B_c(r_a^*, r_{-\boldsymbol{a}}^*, A_s^*) \geq B_c(r_a^*, r_{-\boldsymbol{a}}^*, A_s) \tag{14}$$

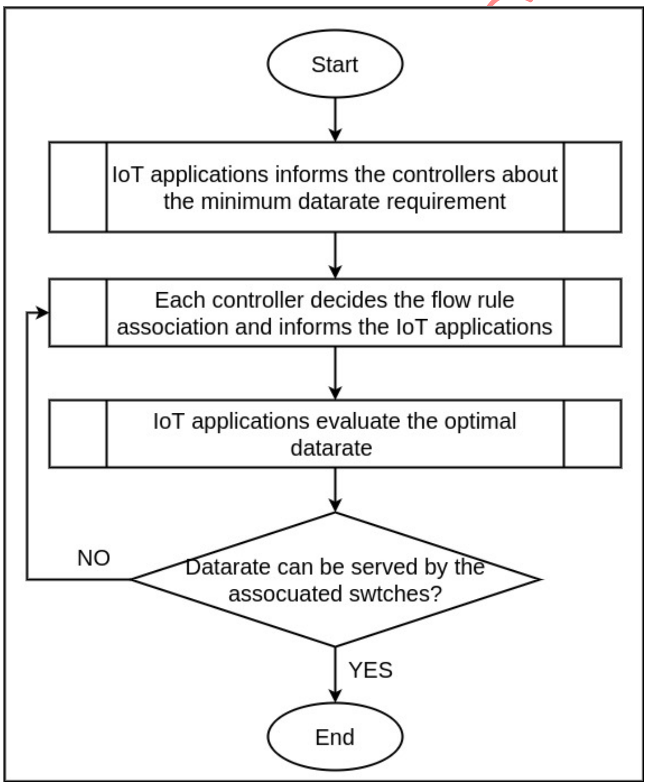

Fig. 2: Workflow of FASCES

Theorem 1 *Given an optimal set of flows* A_s^* *for each switch* $s \in P_a$*, a Stackelberg equilibrium exists for each IoT application* a_n*.*

Proof. The cumulative payoff obtained by the applications A_s associated with switch s is represented as follows:

$$\boldsymbol{U}_s(\cdot) = \sum_{a_n \in A_s} U_a(r_a, r_{-\boldsymbol{a}}, A_s, P_a) \tag{15}$$

By considering the Karush–Kuhn–Tucker (KKT) conditions [22] on $\boldsymbol{U}_s(\cdot)$, we get:

$$L_s = \boldsymbol{U}_s(\cdot) + \lambda_1(B_s - \sum_{a_n \in A_s} r_a) + \sum_{a_n \in A_s} \lambda_2^a(d_a^{th} - \sum_{s \in P_a} d_a^s) \tag{16}$$

where λ_1 and λ_2^a are the Lagrangian multipliers. By taking the derivative of L_s, we obtain the Hessian matrix $\nabla^2 L_s$ is as follows:

$$\nabla^2 L_s = \begin{bmatrix} -\frac{r_1^{min}}{r_1^{max}} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & -\frac{r_a^{min}}{r_a^{max}} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & -\frac{r_{|A_s|}^{min}}{r_{|A_s|}^{max}} \end{bmatrix} \tag{17}$$

We observe that the obtained Hessian matrix is a negative diagonal matrix. Hence, we conclude that there exists at least one Stackelberg equilibrium in FASCES.

4.4 Proposed Workflow

To obtain the optimal distribution of flows, FASCES follows the workflow as shown in Figure 2. Initially, each application informs about their minimum datarate requirement to the controllers. On receiving the set of applications to be served, the controllers allocate the flows optimally among the available switches at aggregation and core layers. Thereafter, the controllers inform the path associated with each flow to the IoT devices, and these devices try to find an optimal value of the datarate can be achieved while interacting with the controllers.

5 Performance Evaluation

In this section, the performance of FASCES is analyzed through simulation with varying the number of heterogeneous IoT applications. We simulated using the MATLAB simulation platform considering a terrain of 10×10 m^2 [18]. The deployment of switches follows a grid patern. On the other hand, IoT devices are deployed randomly and follow random waypoint mobility model [18]. We consider that there are 2 pods at the aggregation layer, where each pod is comprised of 4 switches and 2 switches at the core layer. We considered that the IoT applications generate data in chunk having size 800 Mb, as shown in Table 1. We consider

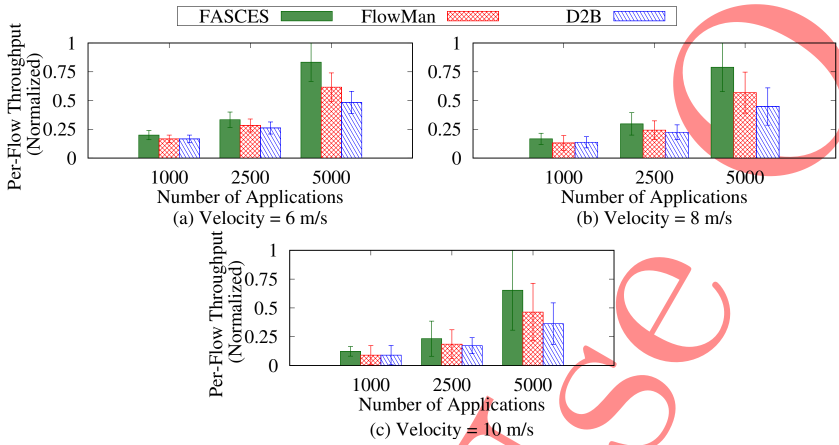

Fig. 3: Per-Flow Throughput

Table 1: Simulation Parameters

Parameter	**Value**
Number of Applications	1000-5000
Maximum Datarate of IoT applications	128-5000 $Kbps$
Velocity of IoT Devices	6-10 m/s
Maximum capacity of Each Switches	5-10 $Gbps$
Chunk of data generated by each IoT Applications	500-2000

Table 2: Maximum Datarate Distribution [18]

Maximum Datarate ($Kbps$)	**IoT Applications (%)**
5000	15
1000	25
1000	25
384	40
128	20

the datarate requirement distribution of IoT applications, as presented in Table 2.

The performance of FASCES is evaluated while comparing two of the existing schemes – data flow management in SDN (FlowMan) [20] and data broadcasting in fat-tree DCN (D2B) [18]. In FlowMan, the authors proposed a Nash bargaining game-based data traffic management scheme for SDN. However, while allocating resources, the authors only considered the flows within one-hop neighbors. In other words, FlowMan is capable of ensuring a local optimum which is

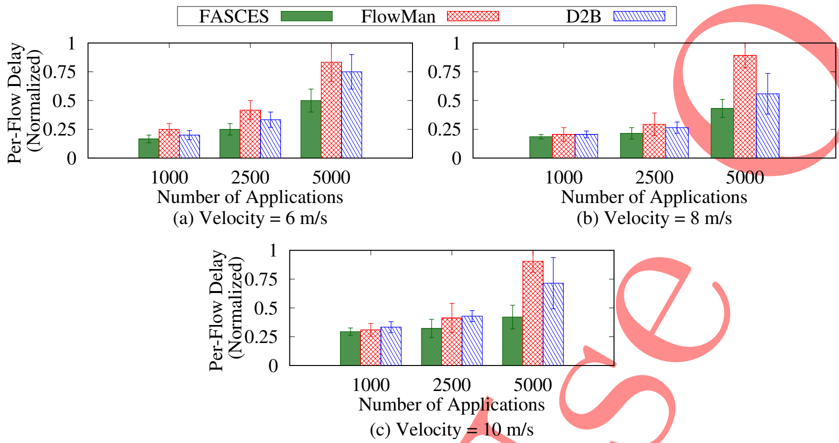

Fig. 4: Per-Flow Delay

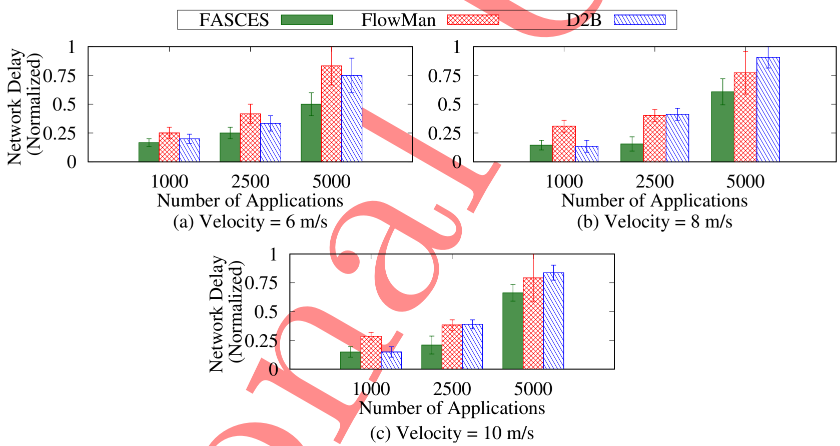

Fig. 5: Network Delay

not sufficient for a network with heterogeneous switches. On the other hand, in D2B, the authors proposed a Stackelberg game-based data broadcasting scheme for DCN. However, in D2B, only a single IoT source is considered. Additionally, the switches are homogeneous and traditional without having any limitation on the set of applications that can be handled by each switch. Using FASCES, we address these lacunae in the existing literature while ensuring balanced data traffic in the network.

We evaluate the performance of FASCES based on the following parameters – (1) per-flow throughput, (2) per-flow delay, (3) network throughput, (4) network delay, and (5) set of serviced IoT applications.

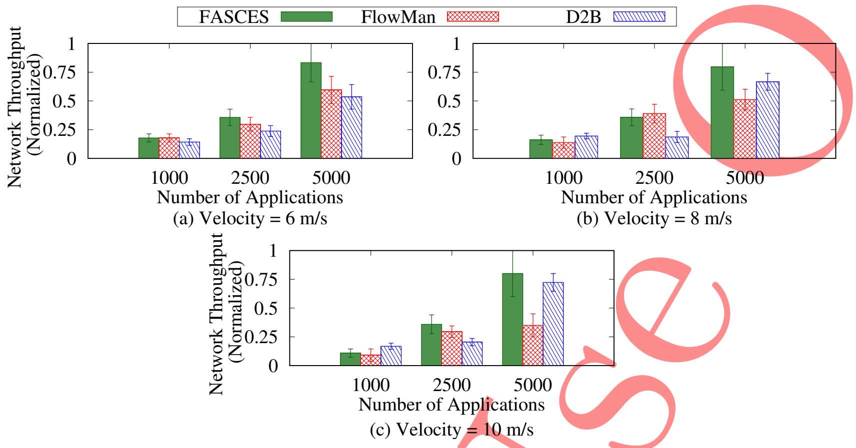

Fig. 6: Network Throughput

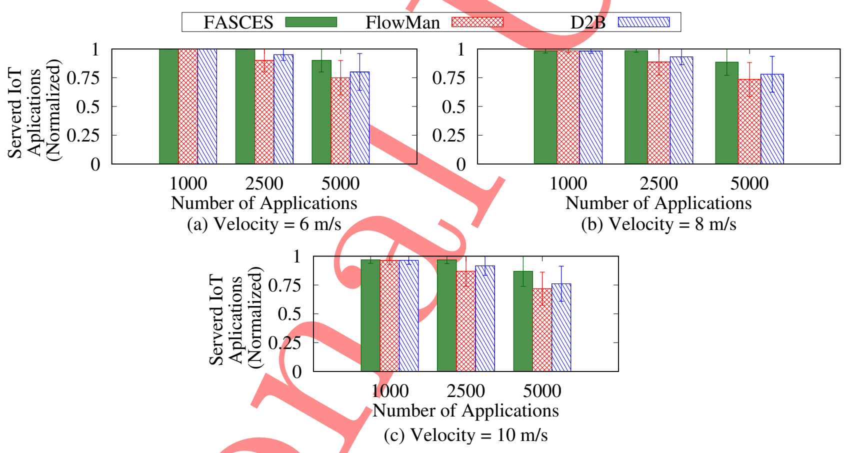

Fig. 7: Serviced IoT Applications

Figure 3 depicts that with the increase in the number of applications, the per-flow throughput increases 15.37-26.91% using FASCES than using FlowMan and D2B. However, with the increase in the velocity of IoT devices, the throughput decreases as the applications need to be associated with new switches very often and a few packets get dropped due to the delay constraint. On the other hand, the delay for each flow also reduces by 27.78-36.67% using FASCES than that of using FlowMan and D2B, as depicted in Figure 4. Additionally, we observe that with the increase in the number of applications, the increase in delay is not significant using FASCES.

Similarly, we observe that the network delay decreases by 16.67-19.45% and the network throughput increases by 16.67-19.45% using FASCES than using

FlowMan and D2B, as depicted in Figures 5 and 6, respectively. This is because the flows are distributed efficiently among the switches at the aggregation and core layers while ensuring efficient utilization of link capacity and TCAM space.

Furthermore, from Figure 7, we observe that FASCES is capable of serving all the applications with efficient data traffic distribution. However, with the increase in the number of applications, FASCES cannot serve 100% application due to physical limitations of the system. Using FASCES, we yield a 4.56-16.67% increase in serviced application than using FlowMan and D2B.

6 Conclusion

In this work, we proposed a data traffic management scheme, named FASCES, and modeled the interaction between the controllers and the SDN switches using a single-leader-multiple-followers Stackelberg game. FASCES is capable of ensuring balanced data traffic in the presence of heterogeneous IoT flows and SDN switches. We observed that FASCES reduces the per-flow delay as well as network delay at least by 27.78% and 16.67%, respectively while ensuring an increase in both per-flow throughput and network throughput. Through simulation, we yield that FASCES ensures efficient flow distribution in software-defined DCN.

In future, this work can be extended while designing data traffic management for the recursive architectures of DCN such as B-Cube and DCell. Additionally, we can also explore this work while considering a multi-tier controller structure for each layer in fat-tree DCN. Moreover, this work also can be extended while considering the link and switch failure in software-defined DCN.

References

1. Agarwal, S., Kodialam, M., Lakshman, T.V.: Traffic Engineering in Software Defined Networks. In: Proc. of IEEE INFOCOM, pp. 1–9 (2013). DOI 10.1109/infcom.2013.6567024
2. Al-Fares, M., Loukissas, A., Vahdat, A.: A Scalable, Commodity Data Center Network Architecture. In: Proceedings of ACM SIGCOMM Conference on Data Communication, pp. 63–74. New York, NY, USA (2008). DOI 10.1145/1402958.1402967. URL `http://doi.acm.org/10.1145/1402958.1402967`
3. Allybokus, Z., Avrachenkov, K., Leguay, J., Maggi, L.: Multi-Path Alpha-Fair Resource Allocation at Scale in Distributed Software-Defined Networks. IEEE J. on Sel. Areas in Comm. **36**(12), 2655–2666 (2018). DOI 10.1109/JSAC.2018.2871293
4. Aujla, G.S., Chaudhary, R., Kumar, N., Kumar, R., Rodrigues, J.J.P.C.: An Ensembled Scheme for QoS-Aware Traffic Flow Management in Software Defined Networks. In: Proc. of IEEE Int. Conf. on Comm., pp. 1–7 (2018). DOI 10.1109/ICC.2018.8422596
5. Bera, S., Misra, S., Obaidat, M.S.: Mobility-Aware Flow-Table Implementation in Software-Defined IoT. In: Proc. of IEEE GLOBECOM, pp. 1–6 (2016). DOI 10.1109/GLOCOM.2016.7841995
6. Bera, S., Misra, S., Vasilakos, A.V.: Software-Defined Networking for Internet of Things: A Survey. IEEE Internet of Things Journal **4**(6), 1994–2008 (2017)

7. Chakraborty, A., Misra, S., Mondal, A.: Qos-aware dynamic cost management scheme for sensors-as-a-service. IEEE Transactions on Services Computing pp. 1–12 (2020). DOI 10.1109/TSC.2020.3011495
8. Chakraborty, A., Misra, S., Mondal, A., Obaidat, M.S.: Sensorch: Qos-aware resource orchestration for provisioning sensors-as-a-service. In: ICC 2020 - 2020 IEEE International Conference on Communications (ICC), pp. 1–6 (2020). DOI 10.1109/ICC40277.2020.9148621
9. Chakraborty, A., Mondal, A., Misra, S.: Cache-enabled sensor-cloud: The economic facet. In: 2018 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6 (2018). DOI 10.1109/WCNC.2018.8377069
10. Chakraborty, A., Mondal, A., Roy, A., Misra, S.: Dynamic trust enforcing pricing scheme for sensors-as-a-service in sensor-cloud infrastructure. IEEE Transactions on Services Computing **Early Access**, 1–12 (2018). DOI 10.1109/TSC.2018.2873763
11. Chen, M., Mao, S., Liu, Y.: Big Data: A Survey. Mobile Networks and Applications **19**(2), 171–209 (2014). DOI 10.1007/s11036-013-0489-0
12. Guo, Z., Yang, Y.: Multicast Fat-Tree Data Center Networks with Bounded Link Oversubscription. In: Proceedings of IEEE INFOCOM, pp. 350–354 (2013). DOI 10.1109/INFCOM.2013.6566793
13. Hayes, M., Ng, B., Pekar, A., Seah, W.K.G.: Scalable Architecture for SDN Traffic Classification. IEEE Syst. J. pp. 1–12 (2017). DOI 10.1109/JSYST.2017.2690259. DOI: 10.1109/JSYST.2017.2690259
14. Li, F., Cao, J., Wang, X., Sun, Y., Pan, T., Liu, X.: Adopting SDN Switch Buffer: Benefits Analysis and Mechanism Design. In: Proc. of IEEE ICDCS, pp. 2171–2176 (2017). DOI 10.1109/ICDCS.2017.255
15. Maity, I., Mondal, A., Misra, S., Mandal, C.: Tensor-Based Rule-Space Management System in SDN. IEEE Systems Journal **13**(4), 3921–3928 (2019)
16. Misra, S., Bera, S.: Soft-VAN: Mobility-Aware Task Offloading in Software-Defined Vehicular Network. IEEE Transactions on Vehicular Technology **69**(2), 2071–2078 (2020)
17. Misra, S., Chakraborty, A.: Qos-aware dispersed dynamic mapping of virtual sensors in sensor-cloud. IEEE Transactions on Services Computing pp. 1–12 (2019). DOI 10.1109/TSC.2019.2917447
18. Misra, S., Mondal, A., Khajjayam, S.: Dynamic big-data broadcast in fat-tree data center networks with mobile iot devices. IEEE Systems Journal **13**(3), 2898–2905 (2019). DOI 10.1109/JSYST.2019.2899754
19. Misra, S., Schober, R., Chakraborty, A.: Race: Qoi-aware strategic resource allocation for provisioning se-aas. IEEE Transactions on Services Computing pp. 1–12 (2020). DOI 10.1109/TSC.2020.3001078
20. Mondal, A., Misra, S.: Flowman: Qos-aware dynamic data flow management in software-defined networks. IEEE Journal on Selected Areas in Communications **38**(7), 1366–1373 (2020). DOI 10.1109/JSAC.2020.2999682
21. Mondal, A., Misra, S., Chakraborty, A.: TROD: Throughput-Optimal Dynamic Data Traffic Management in Software-Defined Networks. In: Proc. of IEEE Globecom Workshops, pp. 1–6 (2018). DOI 10.1109/GLOCOMW.2018.8644398
22. Mondal, A., Misra, S., Obaidat, M.S.: Distributed home energy management system with storage in smart grid using game theory. IEEE Systems Journal **11**(3), 1857–1866 (2017). DOI 10.1109/JSYST.2015.2421941
23. Moradi, M., Zhang, Y., Morley Mao, Z., Manghirmalani, R.: Dragon: Scalable, Flexible, and Efficient Traffic Engineering in Software Defined ISP Networks. IEEE

J.on Sel. Areas in Comm. **36**(12), 2744–2756 (2018). DOI 10.1109/JSAC.2018.2871312
24. Paul, D., Zhong, W.D., Bose, S.K.: Demand Response in Data Centers Through Energy-Efficient Scheduling and Simple Incentivization. IEEE Syst. J. **11**(2), 613–624 (2017). DOI 10.1109/JSYST.2015.2476357
25. Rottenstreich, O., Kanizo, Y., Kaplan, H., Rexford, J.: Accurate Traffic Splitting on SDN Switches. IEEE J. on Sel. Areas in Comm. **36**(10), 2190–2201 (2018). DOI 10.1109/JSAC.2018.2869949
26. Sadeh, Y., Rottenstreich, O., Barkan, A., Kanizo, Y., Kaplan, H.: Optimal Representations of a Traffic Distribution in Switch Memories. In: Proc. of IEEE INFOCOM, pp. 2035–2043 (2019). DOI 10.1109/infocom.2019.8737645
27. Saha, N., Bera, S., Misra, S.: Sway: Traffic-Aware QoS Routing in Software-Defined IoT. IEEE Transactions on Emerging Topics in Computing pp. 1–1 (2018)
28. Saha, N., Misra, S., Bera, S.: QoS-Aware Adaptive Flow-Rule Aggregation in Software-Defined IoT. In: Proceedings of IEEE Global Communications Conference (GLOBECOM), pp. 206–212 (2018)
29. Sanvito, D., Filippini, I., Capone, A., Paris, S., Leguay, J.: Adaptive Robust Traffic Engineering in Software Defined Networks. In: Proc. of IFIP Networking and Workshops, pp. 1–9 (2018). DOI 10.23919/ifipnetworking.2018.8696406
30. Singh, A., Batra, S., Aujla, G.S.S., Kumar, N., Yang, L.T.: BloomStore: Dynamic Bloom Filter-based Secure Rule-Space Management Scheme in SDN. IEEE Transactions on Industrial Informatics pp. 1–11 (2020). DOI: 10.1109/TII.2020.2966708
31. Trestian, R., Ormond, O., Muntean, G.M.: Enhanced Power-Friendly Access Network Selection Strategy for Multimedia Delivery Over Heterogeneous Wireless Networks. IEEE Transactions on Broadcasting **60**(1), 85–101 (2014)
32. Tseng, S.H., Tang, A., Choudhury, G.L., Tse, S.: Routing Stability in Hybrid Software-Defined Networks. IEEE/ACM Trans. on Net. **27**(2), 790–804 (2019). DOI 10.1109/tnet.2019.2900199
33. Tushar, W., Saad, W., Poor, H.V., Smith, D.B.: Economics of electric vehicle charging: A game theoretic approach. IEEE Transactions on Smart Grid **3**(4), 1767–1778 (2012). DOI 10.1109/TSG.2012.2211901
34. Wang, M.H., Chi, P.W., Guo, J.W., Lei, C.L.: SDN storage: A Stream-Based Storage System over Software-Defined Networks. In: Proc. of IEEE INFOCOM Wrksps, pp. 598–599 (2016). DOI 10.1109/INFCOMW.2016.7562146