

Received May 25, 2020, accepted June 16, 2020, date of publication June 29, 2020, date of current version July 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3005569

# Service and Resource Aware Flow Management Scheme for an SDN-Based Smart Digital Campus Environment

YOSRA NJAH<sup>ID</sup>, CHUAN PHAM<sup>ID</sup>, (Graduate Student Member, IEEE),  
AND MOHAMED CHERIET<sup>ID</sup>, (Senior Member, IEEE)

École de Technologie Supérieure, Montreal, QC H3C 1K3, Canada

Corresponding author: Yosra Njah (yosra.njah.1@ens.etsmtl.ca)

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant STPGP 506319-17, and in part by the Canada Research Chair, Tier 1, under Grant NSERC 950-229052, held by Prof. Mohamed Cheriet.

**ABSTRACT** Recently, campuses have been embracing smart digital technologies in order to boost the efficiency of education and creativity. Thus, massive heterogeneous flows are generated as a result of multitude simultaneous access from several heterogeneous devices. This is putting pressure on campuses to make better management of their constrained resources and to ensure the required Quality of Service (QoS). In this paper, we propose a multi-flow management scheme over a software-defined smart digital campus network, named Service and Resource Aware Flow Management (SRAFM). Our approach offers a unified fully-programmable architecture, a distributed end-host-based flow characterization plane, and a centralized software-defined optimization model to efficiently manage heterogeneous flows. Network functionalities, including QoS aware routing and resource allocation optimization, are formulated as a mixed-integer linear programming problem. Due to its NP-hard complexity, we propose an approximation algorithm in a decomposed fashion based on Lagrangian Dual Decomposition (LDD) and subgradient methods to find an optimal solution for flow management. We evaluate our scheme from different aspects, including the number of simultaneous heterogeneous flows, QoS provisioning, characterization impacts, and network scalability. Compared to the well-known benchmarks in QoS aware routing and optimization problem, SWAY and LARAC, our simulation results conducted with a large number of flows over a small-scale network show promising performance. The proposed scheme significantly improves the cost reduction by 51% as compared to LARAC, the end-to-end delay by 21% and 34%, the bandwidth availability by 27% and 36%, and the QoS violation by 11% and 29% as compared to SWAY and LARAC, respectively.

**INDEX TERMS** Smart digital campus, Internet of Things (IoT), software-defined networking (SDN), flow characterization, Quality of Service (QoS), distributed rate allocation, and resource optimization.

## I. INTRODUCTION

Campuses are now embracing smart digital technologies (e.g., Industrial Internet, Internet of Things, and Smart Cities) to create intelligent, green, and safe educational environments. Staff and students are empowered with smart services, which boost efficiency for learning, collaborating, creating, and sharing. Many research studies [1]–[5] and industrial companies, such as Cisco [6], Campus Management Corp [7], Deloitte [8], and Ruckus [9] have revolutionized how to design, build and manage smart campus networks to move towards digital education.

The associate editor coordinating the review of this manuscript and approving it for publication was Alessandro Pozzebon.

The smart digital campus, or modern campus, is equipped with thousands of heterogeneous Internet of Thing (IoT) and non-IoT devices that autonomously interact with each other to unleash or use a massive number of services appropriate for smart living and learning applications. With this unlimited development of heterogeneous devices and services, the campus network size keeps scaling up, while massive heterogeneous traffic flows continue to grow inexorably. Accordingly, network resources and bandwidth acquire an unprecedented demand. For example, in [3], Sivanathan *et al.* showed that when there is an activity from IoT devices and non-IoT devices, the campus network load peaks at around 17 Mbps. Meanwhile, the average load is 400 Kbps with IoT devices. Another study produced by

Cisco [10] shows that some types of network applications, like video-based applications supplied by providers such as Youtube, Netflix, and Hulu, will grow at a compound annual growth rate (CAGR) of 31%, while online gaming traffic will have a traffic growth rate of 47%, and traffic including web, email, and data will have a CAGR of 18%.

These services' massive data streams are leading to unprecedented challenges for network administrators in terms of advanced solutions development for network flow management and resource allocation control with minimum cost, especially with the constrained campus network resource problems (e.g., limited link capacity, constrained device with limited CPU, memory, and power resources) [11]–[13]. Furthermore, these heterogeneous services require various specific QoS requirements. For instance, certain mission-critical flows generated during some emergency/urgent periods, such as gas monitors, smoke sensors, and disaster sensors, require transference of data in real-time. Some other applications, such as peer-to-peer file sharing, software updates, and cloud-based file storage systems, are not time sensitive but very bandwidth-hungry applications that might occupy all available bandwidth in the case of inefficient management. Especially, certain types of services, such as online learning and video conferencing, require bandwidth and real-time guarantees to run without degraded performance.

In fact, network services may violate QoS levels because of four main network factors: low bandwidth, high latency, WiFi signal interference, and overloaded constrained device. If all network equipment are working properly, then bandwidth and latency are the two likely reasons [14]. In addition, even though a campus network has been designed with highly adequate resource bandwidth, but without efficient engineering mechanisms for service and resource management, critical flows may compete with all kinds of traffic, including bandwidth-hungry flows; consequently this causes serious flow QoS violations in the network. Moreover, the traditional campus network architecture has limited global state visibility, i.e., it lacks a global view of the available network resources and the overall network architecture, since each router performs a hop-by-hop routing using its coupled control and data planes [15]. This makes traditional network architecture hard and time-consuming to configure devices and manage resources and traffic flows.

Software-defined networking (SDN) has emerged as an efficient network management paradigm to overcome these issues by decoupling the data and control planes. The network control functionalities are further away from network devices and centralized into a logically centralized point, called the control plane (the brain of the network). In other words, network decisions are made by the control plane with a global network view and use different programs in the application plane to optimize network resource management. The forwarding function is performed according to engineering policies programmed and configured by the control plane into network devices that are converted into simple high-speed forwarding elements. This emerging paradigm is particularly

attractive for addressing many network optimization problems, such as dynamic flow control, flexible network resource management, and QoS provisioning [16]–[19].

The above-mentioned issues and the adoption of software-defined based implementation motivate us to focus on the following questions: *How can an efficient fully-programmable SDN-based solution for the smart digital campus be designed? How can the different requirements of each network traffic flow within a set of hundreds, even thousands of flows, be addressed? What are the network policies that proactively automate the management process of a set of heterogeneous flows while fulfilling the required QoS, improving available resources, and minimizing the routing cost?*

To answer these questions, in this paper, we propose a multi-flow management scheme over a software-defined smart digital campus network named Service and Resource Aware Flow Management (SRAFM). The proposed scheme is different from existing works for two reasons. First, it offers a whole solution in terms of architecture, flow characterization, and QoS aware routing and resource optimization to manage the masses of heterogeneous flows generated from thousands of interconnected devices. Thus, over a unified fully-programmable architecture, SRAFM implements a distributed end-host-based flow characterization solution and a centralized software-defined optimization model for service and resource-aware routing problem. Second, in terms of QoS aware routing and resource optimization problem, SRAFM provides an optimal solution for an entire set of simultaneous flows in the network, which is more challenging than most prior schemes that often address this issue with a flow-per-flow strategy. Furthermore, regarding the network in terms of flow-per-flow management cannot guarantee a global optimal allocation in the network; in some special cases, it may reduce the network performance by wrong routing decisions in advance.

The major contributions of this work can be summarized as follows:

- 1) **Design of a smart campus programmable architecture for flexible flow characterization and management:** We discuss the design challenges to deploy a unified fully-programmable architecture that controls wired and wireless software defined-based smart campus networks. Since flow characterization presents a fundamental network functionality to support QoS aware routing and optimization, we provide an overview of the different types of services and their QoS requirements on the basis of delay and bandwidth characterization. Accordingly, we identify the design challenges confronting the characterization of a large number of heterogeneous flows in a fully-programmable architecture, and we propose a distributed end-host-based plane for flow characterization based on a combined approach of device and service identifications. While designing this specific network-level in the architecture, we take into

consideration the end-user's privacy, the processing time, the controller overhead, and the network bandwidth consumption.

- 2) **QoS aware routing and optimization:** SRAFM controls not only delay-sensitive flows, as do prior works, but all smart campus network flows to ensure service QoS requirements, to control bandwidth-hungry services, and to optimize network resource allocation. We propose a centralized optimization framework that manages flows based on an SDN proactive and reactive strategy and optimizes the system cost in terms of joining resource cost and path loss. Since the formulated multi-constraints optimization problem is NP-hard, the Log-det approximation function [20] is used to relax the problem. Then, by using the Lagrangian Dual Decomposition approach, we decompose the relaxed problem into per-flow sub-problems that can be solved simultaneously in a decomposed fashion. Accordingly, the SRAFM routing strategy finds an optimal solution not only for each flow independently but for the whole set of flows while performing coordination between their various requirements and the available resources.
- 3) **Simulation results:** We evaluate the proposed scheme through various simulation aspects, including the number of simultaneous heterogeneous flows, QoS requirements, characterization impacts, and network scalability. We compare our scheme to the SWAY [13] and Lagrange Relaxation based Aggregated Cost (LARAC) [21] algorithms, the well-known benchmarks in QoS aware routing and optimization problem. Our simulation results conducted with a large number of flows over a small-scale network show promising performance. Thus, SRAFM achieves 51% in terms of cost reduction as compared to LARAC. In addition, it improves the end-to-end delay by 21% and 34%, the bandwidth availability by 27% and 36%, and the QoS violation by 11% and 29%, as compared to SWAY and LARAC, respectively. To evaluate the system cost, we consider only LARAC, since SWAY adopts two different cost functions with different metrics in the same network.

The remainder of this paper is organized as follows. Section II reviews related relevant works from the perspective of QoS aware routing and optimization over a software-defined-based environment. Section III discusses the software-defined smart campus network, the unified fully-programmable architecture, and the strategies to deploy flexible multi-flow characterization and optimization solutions. Sections IV and V respectively present the optimization problem and the proposed QoS-aware routing algorithms. The SRAFM operational scenario and the analysis of the results are presented in Sections VI and VII, respectively. Finally, we conclude and discuss future work in Section VIII.

## II. RELATED WORK

In the context of smart digital campus networks, many research studies have been proposed to design and build smart digital campus networks with the appropriate technologies (e.g., traffic profiling, prediction of student attendance, etc.) to move towards digital education [1]–[3], [5]. However, none of these works target service and resource-aware traffic management and optimization. On the other hand, in the context of QoS provisioning and resource optimization over smart networks, the body of literature is vast [22], and it covers different SDN-based networks, such as smart home networks [23], [24] and industrial networks [25], [26]. However, the management of a large number of heterogeneous flows to improve service and network performance over smart digital campus networks is still an open issue with many challenges.

While few research works deal with QoS provisioning and resource optimization over SDN-based smart digital campus networks, industrial companies, such as Cisco [6], Campus Management Corp [7], Ruckus [9], and Huawei [27] have revolutionized how to design, build and manage smart campus networks to move towards digital education. Thus, the remainder of this section reviews some relevant related works from the perspective of QoS-aware routing and optimization problems in SDN-based environments, particularly the most common aspect, i.e., the delay-constrained least-cost (DCLC) routing problem.

In [28] and [29], Egilmez *et al.* propose an optimization model to ensure end-to-end multi-level QoS for video streaming service over SDN-based networks. They treat the base layer of video bit streams as a level-1 QoS flow, while packets of enhancement layers are treated as level-2 QoS or as best-effort flows. The rest of the network traffic is also managed as best-effort flows. They pose optimization QoS routing as a constrained shortest path problem in which delay and packet-loss are considered as QoS requirements, and they use the LARAC scheme [21] as a QoS routing algorithm. In [30], Yu *et al.* propose a QoS routing scheme for video streaming traffic over SDN networks. Similar to the previous works, [30] also treats the base layer and enhancement layer of video bit streams separately as two levels of QoS flows. However, the proposed routing solution is based mainly on the shortest path algorithm to route the base layer packets (level-1 QoS flows), if it meets the delay variation constraints. Otherwise, a QoS routing algorithm is invoked to select the required path. Despite the fact that [28], [29], and [30] cover several critical issues in terms of multi-level QoS, they do not consider different types of network services. In other words, their proposed schemes address problems for a specific service, i.e., video streaming, which might not fit the large number of heterogeneous services in smart networks.

In [31], Guck *et al.* provide a comprehensive survey of QoS routing algorithms in SDN-based networks. They implemented 26 DCLC algorithms and compared their

run-time and cost efficiency within a four-dimensional (4D) evaluation framework. The four dimensions correspond to the type of topology, two forms of scalability of topology, and the tightness of the delay constraint. They conclude with the outperformance of two routing algorithms in the vast majority of the evaluations, namely, LARAC [21] and Search Space Reduction Delay-Cost-Constrained Routing (SSR+DCCR) [32]. All the evaluated algorithms, in [31], are devised to deal with single metric routing schemes (i.e., delay). However, a multi-metric QoS routing optimization should be considered to fulfill heterogeneous service requirements. In other words, the QoS provisioning problem should not take into consideration only time sensitive traffic in which the delay is a highly critical parameter, but also small IoT flows and mission-critical data that need different QoS requirements. Furthermore, bandwidth-hungry flows should be under the control of specific engineering policies, because transferring such type of traffic using only the best-effort mechanism, as in [28], [29] and [30], can create congestion and degrade the performance of critical applications when the same network resources are shared.

In [13], Saha *et al.* propose two different QoS routing strategies to address the issue of heterogeneous flows. One is devised to deal with delay-sensitive flows, and the other is devised to deal with loss-sensitive flows. Both of the deployed algorithms are based on the Yens K-shortest paths algorithm [33], which is included in the comparison performed by [31]. However, because of the different deployed strategies to deal with the two classes of traffic, the authors consider two different cost functions. Thus, they minimize the delay metric for delay-sensitive traffic and the loss metric for loss-sensitive traffic, subject to different constraints. By contrast, we consider that every flow is sensitive to loss, and we propose an optimization problem that minimizes the operational cost of the selected path in addition to the loss-rate, according to the delay and bandwidth sensitivity metrics.

Finally, it is worth mentioning that the literature encloses a multitude of approaches that address the QoS-aware routing problem over the SDN-based environment, using different mechanisms such as machine learning [34], node characterization [35], queue scheduling [25], multi-path selection [36], etc. However, to the best of our knowledge, all these proposed works are based on a per-flow approach, i.e., using the current state of the network, the optimal routing solution is selected independently for each flow, and not for all flows in the system. Though this flow-by-flow technique can reach a fast routing decision, it cannot provide a global optimal solution for the whole set; in some cases, it violates service requirements and network performance due to previous decisions.

### III. A SMART DIGITAL CAMPUS NETWORK

In this work, we propose a multi-flow management scheme called Service and Resource Aware Flow Management (SRAFM) for the smart digital campus network.

This proposed scheme requires being deployed in a flexible programmable architecture. In addition, the large number of services leads us to design a specific network-level in the architecture to analyze the nature of heterogeneous flows before performing traffic routing and resource optimization.

Thus, in this section, we propose the software-defined smart campus architecture, a unified fully-programmable architecture, which manages the heterogeneous wired and wireless network components. Then, we discuss the characteristics of the network services in terms of delay and bandwidth requirements, and we discuss the distributed end-host-based plane for flow characterization.

#### A. SOFTWARE-DEFINED CAMPUS ARCHITECTURE

As depicted in Figure 1, the software-defined campus network is based on a fully-programmable paradigm in which all the network devices in each layer (i.e., access, aggregation, core, and wireless backhaul layers) are controlled and programmed by the centralized control plane [27]. This control plane can include one or multiple controllers to handle the increased management complexity of large-scale wired and wireless networks. Thus, it is possible to control and manage the wired network over the access, aggregation, and core layers with one SDN controller, or to slice network views in a way that each layer is managed via a different SDN controller. On the other hand, to manage the wireless backhaul layer, it is important to note that SDN has been designed for wired networks; but wireless networks have different requirements, and there is not yet a consensus or standard on how to program wireless forwarding elements [16]. The two main challenges of software-defined wireless implementation are the configuration of the wireless forwarding elements by the control plane, and the interaction between the access layer and the programmable wireless backhaul forwarding elements. For these purposes, significant research studies and industrial implementations have been proposed as an extension of the SDN paradigm to incorporate mobile-specific functionalities. For example, in [42], Huawei presents Huawei's agile campus network solution, a fully-programmable architecture that includes an access controller and programmable agile switches enabling unified wired and wireless traffic forwarding. In [43], Nunez *et al.* propose featuring the wireless backhaul forwarding elements and the typical SDN controller with wireless agent extensions that enable the management of packets and forwarding rules in a technology-agnostic manner. In [44], Seppanen *et al.* propose a network abstraction approach by hiding the wireless network from the SDN layer. Thus, instead of controlling the wireless forwarding elements directly with the SDN controller, the whole wireless network is seen as a single SDN switch, controlled like a standard SDN device.

In our work, as shown in Figure 1, for the wireless backhaul layer management, a controller, named Software-Defined Radio (SDR) controller, is proposed to manage the data connections between the radio access elements (e.g., wireless access points), the Wireless Backhaul Forwarding Elements



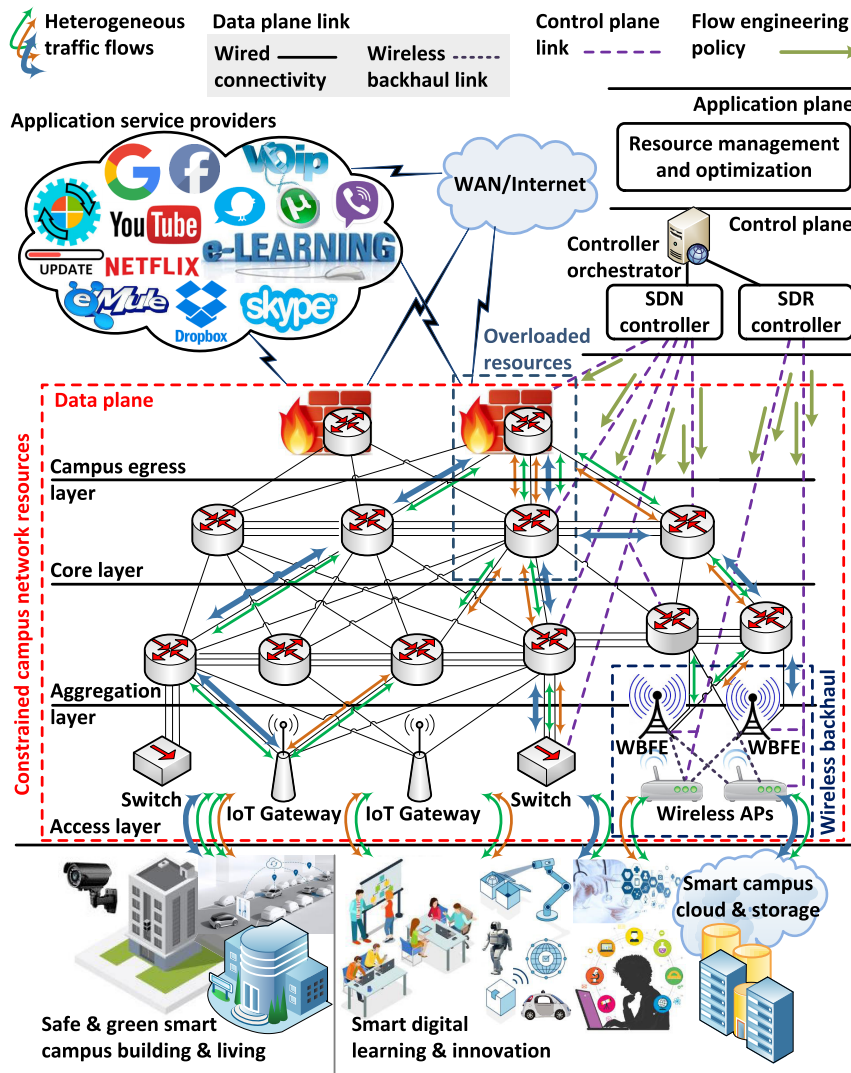


FIGURE 1. Software-defined smart campus network architecture.

(WBFE, such as base stations), and the operators’ SDN enabled devices in the network. All programmability functionalities in the wireless data plane (such as defining forwarding rules and radio resource management) are implemented using the SDR controller. Then, a controller orchestrator is required to ensure the unified interaction of all the heterogeneous network technologies and operators, via the coordination between the different controllers, and the establishment of compatible configurations between the wired and wireless networks.

However, as detailed in [16], where Macedo *et al.* survey SDN, SDR, and network function virtualization (NFV) technologies, achieving unified management of wired and wireless programmable networks is certainly a big challenge, and it is fundamental that these technologies complement each other to develop a highly flexible programmable

network. Thus, in our work, we focus on service management and resource optimization for the software-defined campus network, while assuming that the unified fully-programmable campus architecture is established. In other words, the technical interaction between the different controllers performed by the controller orchestrator and the required features over the wireless forwarding elements to ensure a unified programmable management process are beyond the scope of our current research project. We refer interested readers to [45]–[48] for more details about the design challenges of a unified fully-programmable architecture.

**B. HETEROGENEOUS FLOW CHARACTERIZATION**

**1) SMART DIGITAL CAMPUS NETWORK TRAFFIC**

As shown in Figure 1, over the smart digital campus network, different services related to two main axes, digital learning

**TABLE 1.** Taxonomy of smart digital campus network services generated from IoT and non-IoT devices [37].

Digital learning and innovation services	Priority level	QoS Sensitivity to:	
		Delay	Bandwidth
<b>Video-based communications for online education, administrative, and financial services:</b> e.g., online courses, conferences, training sessions, face-to-face meetings over distance, etc [5][38].	Highly critical	Real-time e.g., < 150 ms	High / medium rate
<b>Smart library, courses, and administration's cloud-based resource transfer and storage:</b> e.g., lectures and assignments, electronic books management, heavyweight software updates, etc [38].	Critical	Non real-time	High rate
<b>Video-based services for students and staff during spare-time:</b> e.g., playing online video games, watching videos from Youtube, Netflix, Hulu, Facebook, etc [24][38][39].	Non critical	Real-time e.g., < 250 ms	High / medium rate
<b>Bulk transfer-based services:</b> e.g., video downloads (for offline viewing), peer-to-peer file sharing, software updates, etc [24][38][39].	Non critical	Non real-time	High rate
Safe & green smart campus building and living	Priority Level	QoS Sensitivity to:	
		Delay	Bandwidth
<b>Video surveillance and security monitoring:</b> e.g., behavior-based emergency services, audio panic alerts, security video surveillance, access control alerts, etc [13][40][41].	Highly critical	Real-time e.g., < 150 ms	High / medium rate
<b>Smart and green campus services:</b> e.g., building automation services, connected lighting, energy and waste management, smart parking, etc [13][40][41].	Critical	Near real-time e.g., < 30 s	Low rate

and smart campus environment, are generated from IoT and non-IoT devices [6]. Table 1 presents a taxonomy and examples of such services.

On the one hand, smart digital learning and innovation services are generated from IoT and non-IoT devices to maximize the potential of learning and research. For example, courses and training are offered by top faculty and leaders from the same university or around the world, and they are accessible to students anywhere and at any time due to innovative learning facilities such as IoT-based classrooms, virtual classes, IoT sensors for note sharing, etc. [7], [38].

On the other hand, smart campus building and living related services are generated mainly from IoT devices to offer a safe, green, and smart living environment. This axis includes certain security services, which report urgent alarm events via messages, high-resolution images, and videos [8]. It also includes various intelligent applications, such as smart access, building automation systems, smart parking, and payment. Besides, it provides services, such as heating adaptation, light-adjustment, and water conservation, which aim to transform the traditional campus environment into a model of a green institution at low cost by reducing energy and carbon footprint [9].

## 2) SERVICES' QoS REQUIREMENTS CHARACTERIZATION

Since the number of services generated from IoT and non-IoT devices is unlimited and network resources are constrained, we define a set of traffic classes with their appropriate QoS requirements and priority levels in the network.

As shown in Table 1, we characterize services' QoS requirements on the basis of delay and bandwidth sensitivities. Thus, we separate network campus services into two main sets: delay and bandwidth sensitive flows. We refer

to these two sets, respectively, as  $F_{ds}$  flows and  $F_{bs}$  flows. Consequently, the whole set of campus network flows  $F$  is represented as follows:

$$F = F_{ds} \cup F_{bs}. \quad (1)$$

The delay-sensitive class includes video-based services and IoT services. All these services are characterized, firstly, by time constraints that have to be deterministically guaranteed. We define, within the  $F_{ds}$  set, three levels of prioritization. The first one (Highly critical/Real-time) includes ultra-high-definition video-based services related, for examples, to security video surveillance, on-line course learning, video conferencing, etc. As shown in Table 1, real-time and high bandwidth are both the main characteristics of these services to ensure an efficient end-to-end delivery without interruptions and packet loss [37]–[39]. All the flows of this level will be managed proactively by the SRAFM scheme since they are regular and frequent in the network. The second level (Critical/Near real-time) includes IoT services that are generated, for examples, by building automation systems, connected lighting, smart parking, etc. As shown in Table 1, these irregular and infrequent services are characterized by near real-time (e.g., tolerable delay of 30 s) and low-rate requirements (i.e., each IoT device exchanges a small amount of data per-flow) [40], [41]. These smart IoT services require a critical priority level in the network, since competing with traditional flows (e.g., bulk transfers) can significantly affect the performance of these low-rate IoT applications [13]. Finally, the third level (Non critical/Real-time) includes non-critical video-based services, which are generated, for instances, by students playing online games and/or watching videos supplied by providers such as Youtube and Netflix [24].

However, the flows of this level require a strict end-to-end delay; for example, an online game requires less than 250 ms to run smoothly [49].

The  $F_{bs}$  set includes services that utilize large amounts of bandwidth and place enormous strain on the network. As shown in Table 1, it includes a significant number of applications, such as peer-to-peer file sharing, large downloads, and software updates, which are very bandwidth-hungry services but not sensitive to delay. This set of flows is also called bulk transfers. The non-controlled management of these bandwidth-hungry applications creates network congestion and leads to performance degradation of the delay-sensitive flows, i.e., the  $F_{ds}$  set. Consequently, these services should be managed through the network with a lower priority compared to the delay-sensitive services. We also categorize this class into two different levels. The first level (Critical/Non real-time) includes mission-critical data such as electronic books management, courses and administration's cloud-based resources transfer and storage, etc. The second level (Non-critical/Non real-time) involves non critical data, such as students' video downloads for offline viewing.

### 3) DISTRIBUTED END-HOST-BASED FLOW CHARACTERIZATION PLANE

Characterizing network flows with the appropriate performance levels and QoS requirements, as shown in Table 1, should be performed before flow management, since this impacts greatly routing decisions and resource allocation optimization. In this section, we discuss the services' massive data stream processing, particularly the challenges of the joint design of multi-flow characterization and management over a fully-programmable architecture, and we propose the distributed end-host-based plane for flow characterization.

With the programmable paradigm, multi-flow processing could be performed within either a centralized approach or a distributed approach. According to the centralized approach, since there is a lack of intelligence in the forwarding plane, thousands of heterogeneous flows are sent by forwarding devices to the centralized controller to be analyzed using flow characterization programs. Then, as illustrated in Figure 2(a), they are mapped to specific engineering policies, which are computed using traffic engineering and management programs. The centralized controller installs these computed rules over the programmable forwarding devices to enable the transfer of the corresponding flows [13], [50]. Nonetheless, with the continuous expansion in the flow number, data rates, and the requirements for detailed analysis, this approach seems to have limited scalability, and it leads to long processing time and heavy overhead that affect the performance of the delay-sensitive flows [50]–[52].

To overcome these problems, the distributed processing approach is proposed through the deployment of multiple processing nodes, such as a cluster of controllers or fog nodes, where each node controls only a specific part of the network's resources and its corresponding heterogeneous

flows [50], [53]. As shown in Figure 2(b), this approach provides data processing as close as possible to the end-devices, which enables reducing the processing time and the overhead as compared to the centralized approach. However, the administrator needs to encounter many issues related to the users' privacy, security, placement of the processing nodes, delay in computing, and energy consumption. Furthermore, being connected to heterogeneous devices, managing the distributed processing nodes, the connections between them, and the heterogeneous networks will be burden unless SDN, SDR, and NFV technologies are applied [16], [53].

In our work, while taking into consideration the above concerns, we separate the flow characterization process from the traffic engineering and resource management process. We propose to perform flow characterization over each end-host in the campus network, based on a combined approach of a device (e.g., ID student laptop) and service (e.g., online gaming) identification, before forwarding flow through the network. We designate this approach as "distributed" since the characterization engine is not centralized over the centralized control plane but distributed over the end-hosts. This approach is developed as well in [54], where a shim layer is introduced over each end-host to detect the specific type of flow in a software-defined inter-data center network. The Differentiated Services Code Point (DSCP) bits are used to mark packets with the specific type of flow [55]. Thus, each non-intelligent forwarding device detects easily and directly the kind of the flow and decides to send it either to the destination using the proactively installed rules or to the centralized controller for path computation.

One of the benefits of characterizing flows by end-hosts is to protect users' privacy. In fact, accessing user data over a processing node (e.g., the SDN controller) and analyzing the corresponding flows may cause discomfort for end-users. Besides, characterizing flows before managing them through non-intelligent programmable devices enables saving network bandwidth and reducing processing time, since network flows are already characterized by sources, and consequently not all flows require the reactive intervention of the controller. Finally, this approach enables reducing controller tasks and improving controller efficiency. Section VI presents more details about the functional description of this approach in the SRAFM scheme over the software-defined smart campus network.

## IV. QoS-AWARE ROUTING PROBLEM AND SRAFM OPTIMIZATION MODEL

In this section, we first outline the adopted network representation and related notations. Then, we present the SRAFM optimization model proposed, in this work, for a software-defined smart campus network.

### A. PREREQUISITE NOTATIONS AND FORMALISMS

Table 2 summarizes the notations used in this work. Let us assume that the network topology is represented by a connected graph  $G = (V, E)$ , where  $V$  is the set

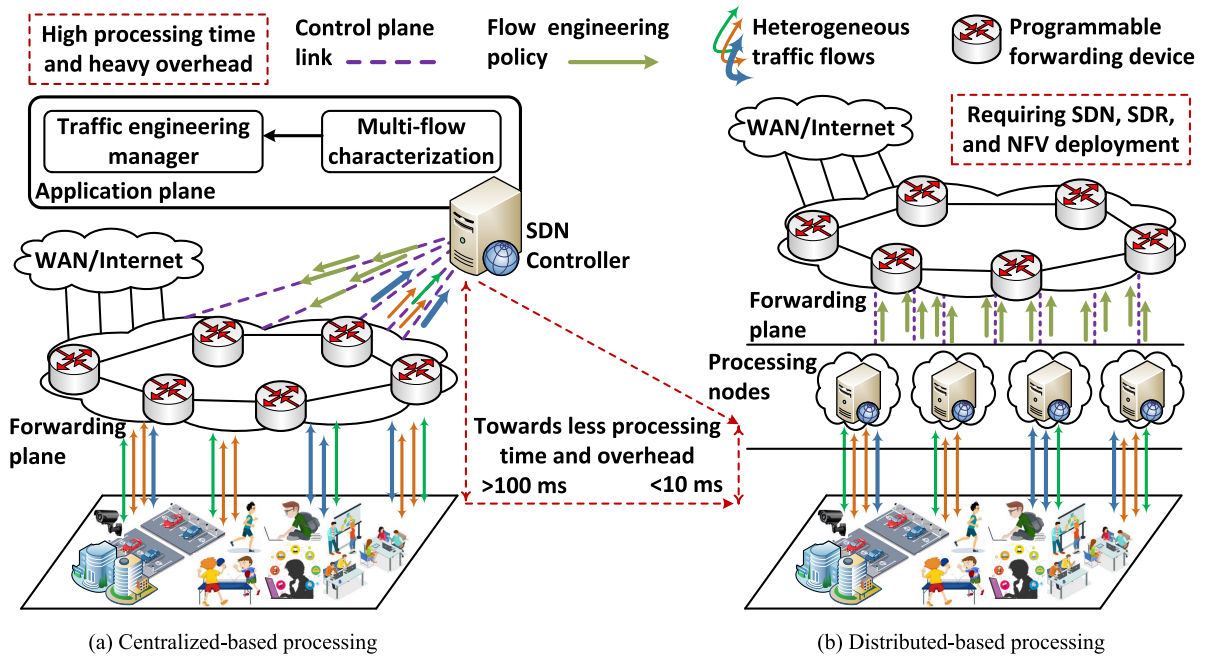


FIGURE 2. Multi-flow processing over programmable architecture based on centralized vs. distributed approaches.

TABLE 2. Summary key notations.

Notation	Description
$F$	Set of all the traffic demands (flows).
$f$	A flow in $F$ .
$P$	Set of paths for routing all flows $F$ .
$P_f$	Set of paths for flow $f$ .
$P_e$	Set of paths going through the link $e$ .
$p$	A path, between source node $s$ and destination node $t$ , that could be used by flow $f$ .
$e$	A directed link $(i, j)$ outgoing from node $i$ and incoming to node $j$ .
$r_{f,p}$	Rate allocation for flow $f$ on path $p$ .
$S_e$	Operational cost of link $e$ .
$S_p$	Operational cost of path $p$ .
$D_e$	Delay of link $e$ .
$D_p$	Delay of path $p$ .
$Q_e$	Packet-loss probability of link $e$ .
$Q_p$	Packet-loss probability of path $p$ .
$C(e)$	Resource capacity of link $e$ .
$C_{res}(e)$	Residual capacity of link $e$ .
$C_p$	Resource capacity of path $p$ .
$D_f^{max}$	Maximum end-to-end delay acceptable by a flow $f$ .
$R_f^{min}$	Minimum required rate by a flow $f$ .
$w, \alpha, \beta$	Constants weights parameters.

of all SDN-enabled devices (nodes), and  $E$  is the set of links. With the following aspects, we formulate the SRAFM optimization model for the QoS aware routing and resource optimization problem.

### 1) THE OBJECTIVE FUNCTION

Assuming that every network service is sensitive to packet loss, we formulate the objective function to optimize the routing decision while considering both aspects: *the operational cost and the packet-loss*. Thus, we aim to minimize the global system cost experienced by a set of simultaneous flows  $F$ , as follows:

$$\min \sum_{f \in F} w_f \sum_{p \in P_f} \alpha S_p \delta(r_{f,p}) + \beta Q_p r_{f,p}. \quad (2)$$

In the remainder of this section, we elaborate on each of the settings used in this objective function. Hence, the path operational cost,  $S_p$ , is calculated by the sum of all the costs of the links belonging to path  $p$ , as shown in (3).

$$S_p = \sum_{e \in p} S_e, \quad \forall p \in P. \quad (3)$$

The end-to-end loss rate probability,  $Q_p$ , on path  $p$  is calculated by the product of the individual packet loss ratios per link of all links belonging to path  $p$  [56], [57], as shown in (4).

$$Q_p = 1 - \prod_{e \in p} (1 - Q_e), \quad \forall p \in P. \quad (4)$$

The  $\delta(r_{f,p})$ , defined in (5), presents an identity function to determine whether or not a flow  $f$  is routed through a path  $p$ .

$$\delta(r_{f,p}) = \begin{cases} 1, & \text{if } r_{f,p} > 0, \\ 0, & \text{if } r_{f,p} = 0. \end{cases} \quad (5)$$

We use the monetary parameters  $\alpha$  and  $\beta \in [0, 1]$  to model a multi-objective function enabling adjustment of the



relative importance of the operational cost and the packet loss, depending on network and traffic characteristics [29], [58]. Besides, we use the weight factor  $w_f \in [0, 1]$  to adjust the priority level of the flows within the same set. Thus, paths with lower latency and higher bandwidth are assigned to high priority services, as detailed in Section III-B.

### 2) SERVICE END-TO-END DELAY CONSTRAINT

Equation (6) defines the end-to-end delay constraint, where the delay of the selected path  $D_p$ , as defined in (7), should be less than or equal to a specified end-to-end delay threshold,  $D_f^{max}$ , required by flow  $f$ .

$$D_p \delta(r_{f,p}) \leq D_f^{max}, \quad \forall f \in F. \quad (6)$$

$$D_p = \sum_{e \in p} D_e, \quad \forall p \in P. \quad (7)$$

### 3) SERVICE RATE AND NETWORK CAPACITY CONSTRAINTS

Equation (8) defines the flow throughput constraint, where the rate of flow  $f$  on selected path  $p$  should be more than or equal to  $R_f^{min}$ , the minimum requirement of flow  $f$ .

$$\sum_{p \in P_f} r_{f,p} \geq R_f^{min}, \quad \forall f \in F. \quad (8)$$

Equation (9) defines the capacity constraint associated with each link  $e \in E$ , where the total rates of all the flows  $f \in F$  going through a specified link  $e$  should not exceed the link's residual capacity.

$$\sum_{f \in F} \sum_{p \in P_e} r_{f,p} \leq C_{res}(e), \quad e \in E. \quad (9)$$

## B. OPTIMIZATION MODEL

The objective of the SRAFM scheme is to select appropriate paths to accommodate a set of characterized network flows while minimizing the total system cost, ensuring the required QoS for each flow, and maximizing the overall network performance. Thus, path selection depends on the flows' QoS requirements and the constrained available network resources.

We formulate the optimization problem as follows:

$$\min \sum_{f \in F} w_f \sum_{p \in P_f} \alpha S_p \delta(r_{f,p}) + \beta Q_p r_{f,p} \quad (10a)$$

$$\text{s.t. } D_p \delta(r_{f,p}) \leq D_f^{max}, \quad \forall f \in F, p \in P, \quad (10b)$$

$$\sum_{p \in P_f} r_{f,p} \geq R_f^{min}, \quad \forall f \in F, \quad (10c)$$

$$\sum_{f \in F} \sum_{p \in P_e} r_{f,p} \leq C_{res}(e), \quad \forall e \in E, \quad (10d)$$

$$\sum_{p \in P_f} \delta(r_{f,p}) = 1, \quad \forall f \in F. \quad (10e)$$

Equation (10a) presents the objective function to be minimized while routing  $F_{ds}$  and  $F_{bs}$  sets of flows. Equations (10b) and (10c) present the delay and bandwidth

## Algorithm 1 Approximation Algorithm for SRAFM

- 1: **Initialization:**  $t = 0, r_{f,p}^0 = 1 - \gamma$ ;
- 2: **Repeat:**
- 3:  $t = t + 1$ ;
- 4:  $\delta^t(r_{f,p}) = \frac{r_{f,p}^t}{r_{f,p}^{t-1} + \gamma}, \forall f \in F, \forall p \in P_f$ , where  $r_{f,p}^{t-1}$  is the solution of the previous iteration, and  $\gamma > 0$  is a small positive constant;
- 5: Replace  $\delta(r_{f,p})$  in (10) by  $\delta^t(r_{f,p})$  to relax the problem;
- 6: Call Algorithm 2 to find  $r_{f,p}^t$ ;
- 7: **Until:**  $|r_{f,p}^t - r_{f,p}^{t-1}| \leq \epsilon_1$ ;
- 8: **Return**  $r_{f,p}^* = r_{f,p}^t$ .

constraints, where  $D_f^{max}$  and  $R_f^{min}$  characterize the QoS requirements of flow  $f$  in terms of delay and rate, respectively. Equation (10d) presents the capacity constraint associated with each link  $e \in E$ . Finally, in (10e), a single path routing decision is assumed for each flow. This assumption leads the controller to configure a small number of forwarding rules and simplifies our system in terms of time complexity since we regard a solution for a set of flows at the same time.

## V. APPROXIMATION AND DECOMPOSITION FRAMEWORK FOR SOLVING SRAFM

In this section, we propose a centralized scheme to be deployed over the SDN application plane. This SRAFM scheme solves the formulated optimization problem. First, since the SRAFM problem is NP-hard, we relax it into a tractable problem. Then, we design a per-flow decomposed algorithm to find the optimal routing solution in the network for a set of characterized flows.

### A. THE APPROXIMATION ALGORITHM

One of the difficulties of the primal problem (10) resides in the integrality condition of the binary variable  $\delta(r_{f,p})$ , which is activated only if a path  $p$  is selected to route flow  $f$ , as shown in (5). Relaxing this problem requires the relaxation of this variable by letting  $\delta(r_{f,p})$  be a real variable in the range of [0,1]. For the SRAFM scheme, we approximate iteratively the original binary value  $\delta(r_{f,p})$  into a real value using (11), a *Log-det* relaxation for approximation function [20], where  $r_{f,p}^{t-1}$  is the rate result of the  $(t - 1)^{th}$  iteration and  $\gamma > 0$  is a small positive constant.

$$\delta^t(r_{f,p}) = \frac{r_{f,p}^t}{r_{f,p}^{t-1} + \gamma}, \quad \forall f \in F, \quad \forall p \in P_f. \quad (11)$$

As shown in Algorithm 1 - line 5, in each iteration  $t$ , the integer-valued function  $\delta(r_{f,p})$  is replaced by the new  $\delta^t(r_{f,p})$  into the NP-hard original optimization problem (10) to formulate the tractable optimization problem, which will calculate the new rate value  $r_{f,p}^t$  by invoking Algorithm 2, in Algorithm 1 - line 6. The relaxed problem is a convex optimization problem, which can guarantee the convergence as proven in [59]. Thus, these processes are repeated until

**Algorithm 2** The Dual Decomposition Algorithm for SRAFM

- 1: **Initialization:**  $\lambda_e^0, l = 0;$
- 2: **Repeat:**
- 3: **Foreach**  $f \in F$  **do:**
- 4: Solving the sub-problem (17) to find the rate allocation variable  $r_{f,p}$ , for each flow;
- 5: **End**
- 6: Perform a subgradient update using (18) to find the dual variables  $\lambda_e^{(l+1)}$ ;
- 7:  $l = l + 1;$
- 8: **Until:**  $|\lambda_e^{(l)} - \lambda_e^{(l-1)}| \leq \epsilon_2;$
- 9: **Return**  $r_{f,p}$  to Algorithm 1 (line 6) as the rate result of the  $l^{th}$  iteration.

the optimal rate value  $r_{f,p}^*$  is achieved upon convergence, i.e.,  $r_{f,p}^{t-1} \approx r_{f,p}^t = r_{f,p}^*$ , with an adequately small  $\epsilon_1$ .

The understanding of the approximation of the modified problem to the original problem upon convergence is proved as follows:

$$\delta^t(r_{f,p}^*) = \frac{r_{f,p}^*}{r_{f,p}^{t-1} + \gamma} \approx \begin{cases} 1, & \text{if } r_{f,p}^* > 0, \\ 0, & \text{if } r_{f,p}^* = 0. \end{cases} \quad (12)$$

Equation (12) shows that, upon convergence,  $\delta^t(r_{f,p}^*)$  of the optimal solution approximately approaches the binary function  $\delta(r_{f,p}^*)$  of the original problem. As a result, the objective function involving  $\delta^t(r_{f,p}^*)$  eventually approximates that of the original problem.

**B. THE DUAL DECOMPOSITION ALGORITHM**

The original problem (10) becomes more tractable by applying the *Log-det* relaxation function. However, its computational complexity is still very high, since this relaxed problem encompasses a wide range of heterogeneous flows with different service requirements. Therefore, to improve the performance of the resolution, we advocate the Lagrangian Dual Decomposition (LDD) approach to find the optimal solution in an efficient decomposed strategy. In other words, we decompose the relaxed optimization problem into *F-optimization* sub-problems, also called *per-flow optimization* sub-problems, that can be solved simultaneously with low complexity. Hence, first, the SRAFM scheme independently solves an optimization sub-problem for each flow within the whole set. Then, it coordinates between all of them and the allocated constrained resources.

For instance, to find the rate solution of the  $t^{th}$  iteration of Algorithm 1, the Lagrangian function of the relaxed primal problem in (10) can be formulated as follows, after decoupling the constraint in (10d):

$$L(r_{f,p}, \lambda_e) = \sum_{f \in F} w_f \left( \sum_{p \in P_f} \alpha S_p \delta^t(r_{f,p}) + \beta Q_p r_{f,p} \right) + \sum_{e \in E} \lambda_e \left( \sum_{f \in F} \sum_{p \in P_e} r_{f,p} - C_{res}(e) \right), \quad (13)$$

where  $\lambda_e \geq 0$  presents the Lagrange multiplier associated with the link capacity. Consequently, the Lagrangian dual function is given by:

$$G(\lambda_e) = \begin{cases} \min L(r_{f,p}, \lambda_e), \\ \text{s.t. (10b), (10c), (10e)}. \end{cases} \quad (14)$$

The dual problem is formulated as follows:

$$\max_{\lambda_e \geq 0} G(\lambda_e). \quad (15)$$

For a fixed dual variable  $\lambda_e$ , (14) is decomposed into *F-optimization* sub-problems, which can be solved with the following objective function:

$$\min_{f \in F} \sum_{p \in P_f} r_{f,p} w_f \left( \frac{\alpha S_p}{r_{f,p}^{t-1} + \gamma} + \beta Q_p \right) + \sum_{e \in E} \sum_{p \in P_e} r_{f,p} \lambda_e, \quad (16)$$

where the constant terms, in (13), can be removed, and the term  $r_{f,p}^{t-1}$  is the rate solution of the previous iteration of Algorithm 1, which is calculated by Algorithm 2. Thus, the whole tractable optimization problem is decomposed into *per-flow optimization* sub-problems and presented as follows:

$$\min_{f \in F} \sum_{p \in P_f} r_{f,p} w_f \left( \frac{\alpha S_p}{r_{f,p}^{t-1} + \gamma} + \beta Q_p \right) + \sum_{e \in E} \sum_{p \in P_e} r_{f,p} \lambda_e \quad (17a)$$

$$\text{s.t. } D_p \delta^t(r_{f,p}) \leq D_f^{max}, \quad \forall p \in P, \quad (17b)$$

$$\sum_{p \in P_f} r_{f,p} \geq R_f^{min}, \quad (17c)$$

$$\sum_{p \in P_f} \delta^t(r_{f,p}) = 1, \quad (17d)$$

$$0 \leq \delta^t(r_{f,p}) \leq 1. \quad (17e)$$

As shown in (17), each sub-problem is intended to select the path that has the minimum cost, which depends on the operational cost, the packet loss, and the rates of all its shared links. The dual variable, which depends on the flow rate variable, presents a penalty to prevent the allocation of flows into congested paths. Hence, flows are spread out into multiple different paths to obtain lower cost while ensuring the delay and bandwidth requirements using constraints (17b) - (17e), which are the remaining constraints of problem (10).

Finally, to solve the dual problem (15), the subgradient projection method is deployed [60]. Thus, the updating rule for the dual variable is formulated as follows:

$$\lambda_e^{(l+1)} = \left[ \lambda_e^{(l)} + \kappa \left( \sum_{f \in F} \sum_{p \in P_e} r_{f,p} - C_{res}(e) \right) \right]^+, \quad \forall e \in E. \quad (18)$$

The  $\lambda_e^{(l+1)}$  reports the evolution of the rate values at each link  $e \in E$ , where  $\kappa$  is a non-negative step-size used to adjust the convergence of the dual decomposition algorithm [63].

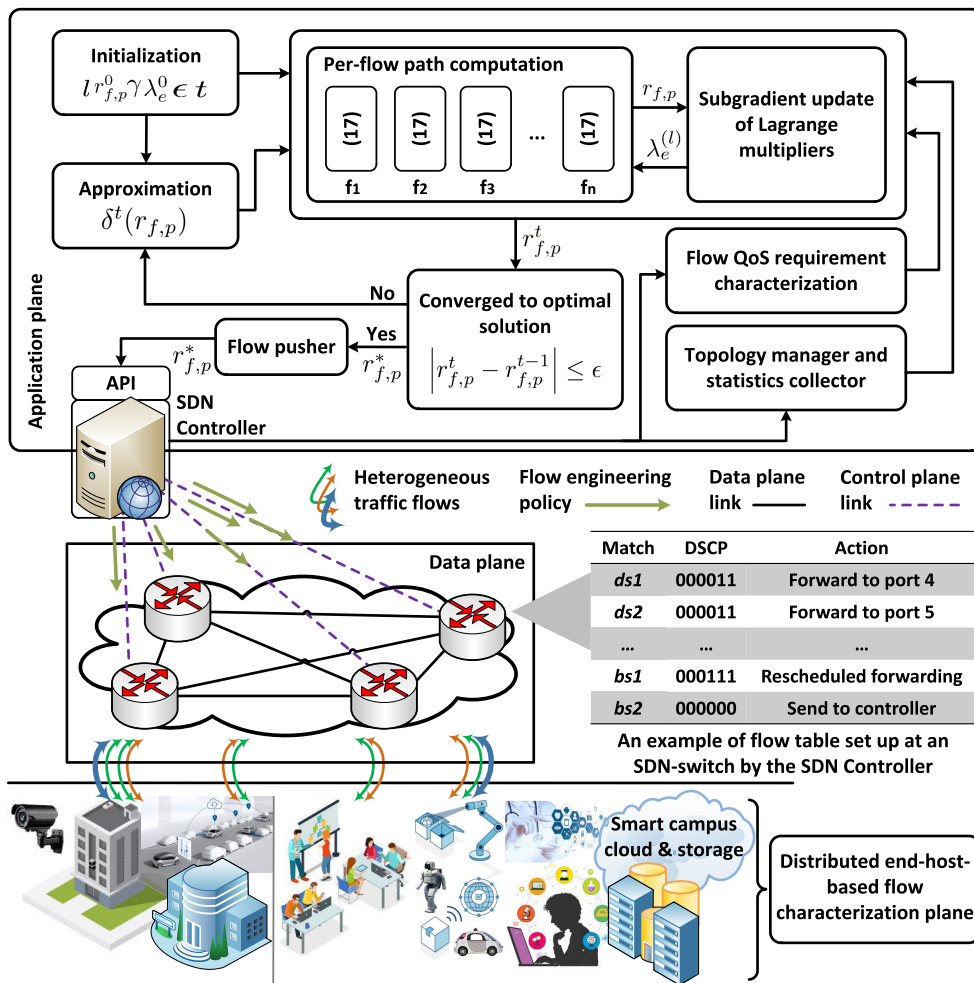


FIGURE 3. SRAFM functional description.

The steps of Algorithm 2 can be summarized as follows. Given the value of  $\delta^t(r_{f,p})$  from Algorithm 1 and the set  $F$  of network flows, at each iteration, Algorithm 2 solves independently the *per-flow optimization* problems to find an optimal path  $p$  for each flow  $f \in F$  (lines 3-5). Then, using the computed value  $r_{f,p}$ , Algorithm 2 (line 6) updates the Lagrangian multiplier  $\lambda_e$  for each link  $e \in E$  using the subgradient method in (18). The algorithm goes in a loop until the change of dual values approximates the stop threshold (line 8). Accordingly, the rate result  $r_{f,p}$  is returned to Algorithm 1 (lines 6-7) as the solution for its  $t^{th}$  iteration (i.e.,  $r_{f,p}^t$ ) to check the convergence to the optimal-rate solution  $r_{f,p}^*$ .

## VI. SRAFM FUNCTIONAL DESCRIPTION

In this section, we discuss the different components included in the whole SRAFM scheme. Figure 3 presents an overview of the functional description of SRAFM.

### A. DISTRIBUTED END-HOST-BASED FLOW CHARACTERIZATION PLANE

When end-users (e.g., students, professors, etc.) connect to the network using their user ID, traffic flows are actively characterized in order to be signaled for the non-intelligent forwarding devices. Flows are characterized on the basis of a combined approach of device-level (e.g., ID professors' devices, ID online-courses' devices, ID students' devices, etc.) and service-level (e.g., video-conferencing, skypeing, online gaming, etc.). Then, to ensure an efficient and easy detection of each flow type by the SDN-forwarding device, the DSCP field (as one of the marking techniques) is used to mark the corresponding packets of each flow with the required priority level. Accordingly, the programmable forwarding element detects the type of each flow smoothly and forwards it either to the controller or to the destination using the corresponding engineering policies (actions), as shown in Figure 3.

As explained in Section III, this distributed end-host-based flow characterization plane enables protecting the privacy of end-users, reducing the processing time required for heterogeneous flow analysis, and reducing controller tasks.

### B. SOFTWARE DEFINED-BASED PROACTIVE AND REACTIVE APPROACH FOR FLOW MANAGEMENT

In our work, the proactive and reactive SDN operational modes are deployed to manage heterogeneous network flows. It is worth mentioning that the existing controllers (e.g., Floodlight, OpenDaylight, NOX, etc.) are by default configured to use only the reactive mode. Meanwhile, the SDN-forwarding devices (e.g., OpenFlow Logical Switch) are by default featured to support both of the SDN modes [64].

With SRAFM, we propose a new controller design that supports proactive and reactive strategies, simultaneously, to manage the regular and irregular network flows, respectively. On the one hand, as explained in Section III-B, the regular and frequent flows of the  $F_{ds}$  set (e.g., courses, video-conferences, and video-surveillance) could be managed within a proactive strategy since the controller is proactively aware of the state of the network and the required resources to ensure the best QoS for these highly critical flows. Hence, their rules are configured in advance in the network, so when they come to the non-intelligent forwarding devices, they are easily identified because of DSCP bits and directly transferred to the destination using a matching between the DSCP bits and the corresponding actions in the flow table, as shown in Figure 3. This SDN operational mode does not require invoking the controller to manage these demands. Consequently, it avoids the time and the amount of overhead needed during the negotiation between the SDN-forwarding devices and the controller to compute paths and configure the network. It also reduces latency and bandwidth consumption of frequent demands, which affect the control plane scalability significantly. On the other hand, the irregular and infrequent IoT flows, the non-critical flows of the  $F_{ds}$  set, and the  $F_{bs}$  flows are managed in the network using the SDN reactive approach. In other words, switching devices need to request the controller, when the first packet of each flow arrives, in order to compute the corresponding routing policy based on the current state of the network, the number of demands (i.e., flows), and the flows' QoS requirements.

To the best of our knowledge, none of the existing works, in the literature, considered both reactive and proactive SDN modes for heterogeneous flow management and resource optimization.

### C. SOFTWARE-DEFINED CONTROLLER MODULES

The flow engineering policy (action) computation process is performed by mainly four SRAFM modules implemented on the SDN application plane. The first module is the topology manager and statistics collector module. It enables maintaining a global view of the network, which is an input to

the path computation modules. It continuously monitors and stores information about all the links and devices currently up in the network. The efficiency of path calculation and resource allocation in the network relies on the accuracy of the network data collected by this function. Hence, many technologies and parameters need to be deployed and configured by the administrator in this functionality to discover and update the entire network state with an efficient strategy, such as the REST-API [65], the Link Layer Discovery Protocol (LLDP) [66], the interval at which network statistics are collected, etc.

The second module is the *flow QoS requirement characterization* module. On the basis of the DSCP bits, this module analyzes the specific flows' QoS requirements and priority levels, as explained in Section III. This analysis is also employed as an input for the *per-flow path computation* module to find the optimal solution and to map the set of flows to the appropriate network engineering policies. Accordingly, the *per-flow path computation* algorithm, which is the third main module in the controller, calculates the optimal solution for the set of characterized flows, based on the QoS requirements, the network topology status, and the *Lagrange multipliers*, as shown in Figure 3, and detailed in Sections IV and V. Finally, when the algorithm converges to the optimal solution, the SDN controller installs the action rules in the forwarding devices over the selected paths using the flow pusher module (the fourth module).

## VII. PERFORMANCE EVALUATION

In this section, we evaluate our SRAFM scheme, first, in terms of running time, system cost, end-to-end delay, average rate allocation, and resource availability while comparing our proposed mechanisms to the state-of-the-art methods. Then, we discuss the percentages of QoS violation and flow rejection. In each of these experiments, we emphasize the importance of performing flow characterization over the distributed end-host-based plane.

### A. SIMULATION SETTINGS AND BENCHMARKS

#### 1) HARDWARE AND SIMULATION SETTINGS

All the experiments were carried out on a PC with an Intel Core i7 CPU @ 3.4 GHz with 16 GB memory. The simulations are performed using the modern programming environment, Julia software [67]–[71], through the mathematical language called Julia for Mathematical Programming (JuMP) [72] and the optimization solver called IPOPT [73]. According to the network topology, we consider Abilene from the SNDlib library, which consists of 12 nodes and 15 links [74]. Then, to evaluate the applicability of our proposed scheme over a large-scale network, we adopt AttMpls topology from the Internet Topology Zoo, which consists of 25 nodes and 57 links [13]. To evaluate the performance of our algorithms under different network conditions, initially, all links' capacities are set to 10 GB and maximum 30% of the links' capacities are randomly



**TABLE 3.** Examples of campus network services and their requirements.

Activity [40] [41] [39] [49]	Delay	Rate
Email and web browsing	$\leq 2 - 5$ s	10 Kbps
Downloading a digital book of 1 MB	–	1.5 Mbps
Online learning	$\leq 150$ ms	2 Mbps
HD-quality video streaming	$\leq 400$ ms	4 Mbps
Skype-group video session (7-10 people)	$\leq 150$ ms	8 Mbps
Game: World of Warcraft	$\leq 250$ ms	50 Kbps
Downloading movies for offline watching during 8 minutes	–	100 Mbps
Motion sensor - IoT service	$\leq 2.6$ s	2.8 - 3.8 Mbps
Voice command and control IoT service	$\leq 0.9$ s	150 Kbps
Video surveillance	$\leq 150$ ms	20 Mbps

consumed. Initially, the delays of links have also been randomly set from 5 to 10 ms and we assume that this value includes the processing, transmission, and queuing delays. We set the operational cost metric randomly according to the load of each link [75], similarly for the delay and the packet-loss metrics [28], during flow allocation.

## 2) BENCHMARK SCHEMES

To evaluate our work, we use LARAC [21] and SWAY [13] algorithms as baselines. The LARAC algorithm is a single metric QoS routing scheme, i.e., a delay-constrained least-cost algorithm. It uses the Lagrange relaxation method to iteratively calculate the best QoS path on the basis of an aggregated concept of cost, which includes the operational cost and the delay [21]. The SWAY scheme is a multi-metric QoS routing approach that considers two different strategies to manage loss and delay-sensitive flows. Both of the SWAY algorithms are based on the Yens K-shortest paths algorithm [33]. Thus, SWAY minimizes the delay metric as a cost function for the delay-sensitive traffic and the loss metric as a second cost function for the loss-sensitive traffic, according to different constraints. In fact, we cannot implement exactly the same algorithms as SWAY since the types of traffic are different. Hence, we adopt only the same strategy of using two different algorithms for delay and bandwidth sensitive flows. Then, according to the type of flow (either  $ds$  or  $bs$ ), SWAY alternates between the two algorithms.

## 3) HETEROGENEOUS FLOWS GENERATION

We randomly generate different smart campus services through the network with different QoS requirements, as detailed in Section III and shown in Tables 1 and 3. We generate 30% of the whole set of heterogeneous flows as IoT services, 50% as delay-sensitive, and 20% as bandwidth-hungry applications. As shown in Table 3, the rate requirements of the generated flows are set in the range of 150 Kbps to 100 Mbps. In addition, to evaluate the end-to-end delay, the thresholds of the heterogeneous  $F_{ds}$  services

are set according to the type of traffic, either interactive applications in the range of 150 to 400 ms or IoT services, for which we use two different services with delay constraints of 2.6 s and 0.9 s, as shown in Table 3. It is noteworthy to mention that the values adopted in our simulation in terms of rate and delay present examples of characteristics for these services. For instance, an online game service could run smoothly with a delay less than 250 ms, as mentioned in Table 3, while an excellent online game (which is not our objective) requires less than 50 ms; also, the more players in an online game, the more data are exchanged between players [39]–[41], [49].

**TABLE 4.** Evaluation of running time (s).

Topology		Abilene			AttMpls		
Number of flows		1000	1500	2000	1000	1500	2000
Algorithms	SRAFM	126	534	1053	302	917	1726
	SWAY	4.2	–	–	–	–	–
	LARAC	5.6	–	–	–	–	–

## B. RESULTS AND DISCUSSION

### 1) SYSTEM RUNNING TIME EVALUATION

Table 4 evaluates the average running time by the number of flows over each topology for each routing strategy. We observe that SRAFM takes more time to select the optimal solution compared to the benchmark algorithms. This difference in terms of required time to achieve the optimal solution is due to the complexity of our algorithm compared to state-of-the-art ones. SRAFM performs coordination between the service quality requirements of the whole set of flows and the available network resources to select the best solution, while LARAC and SWAY select a local best path for each flow until the whole set of flows is achieved. Though these flow-per-flow routing strategies can obtain a fast solution to make a decision for all flows, they cannot reach an optimal routing decision for the whole set, because

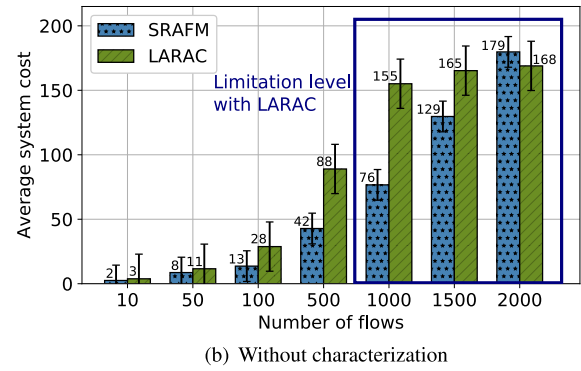
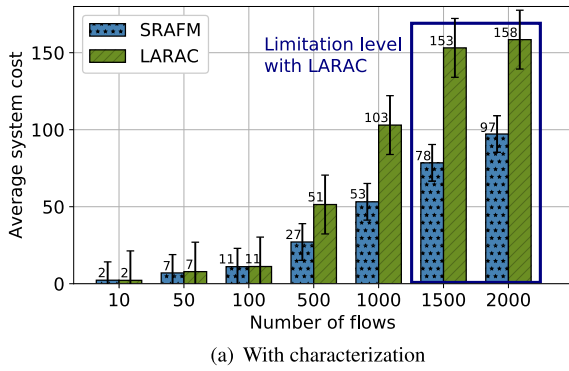


FIGURE 4. System cost evaluation.

of the great number of rejected flows, as shown in Table 4 from the set of 1500 flows with the Abilene topology and the set of 1000 flows with the AttMpls topology.

It is also worth mentioning that solving SRAFM with network topologies as Abilene and AttMpls is not difficult and provides a straightforward solution for a set of simultaneous flows. However, in a real network, the number of paths is extremely large, which makes this problem more complicated. Therefore, the use of parallel computation in multi-core systems and the deployment of the cluster-based distributed controller technology are advised to alleviate this challenge, especially over large-scale networks.

2) SYSTEM COST EVALUATION

In this section, we evaluate the average of the system cost according to the selected solutions. Since the SWAY algorithm takes into account two different cost functions (i.e., it minimizes the delay metric for delay-sensitive flows and the bandwidth utilization metric for bandwidth sensitive flows), we consider only the LARAC algorithm to evaluate the system cost of the SRAFM scheme. Figure 4 depicts the change in the average cost after the allocation of the whole set of flows.

As shown in Figure 4(a) and Figure 4(b), SRAFM shows its outperformance compared to the LARAC algorithm in terms of cost as a result of its distributed rate allocation approach. With the LARAC algorithm, path selection does not depend on the allocation of flow rates, and the cost function increases in accordance with the path operational cost and the delay that are updated after flow allocation has been performed [21]. SRAFM achieves 51% reduction in system cost as compared to LARAC. The system cost with LARAC does not increase compared to SRAFM from 1500 characterized flows and 1000 non-characterized flows, as shown in Figure 4(a) and Figure 4(b), respectively. This is indirectly dependent on the number of QoS violated flows, i.e., reaching the limitation level. In fact, at the limitation level, LARAC rejects all new flows as a result of QoS violation, so its system cost does not increase compared to the SRAFM scheme, which still accepts flows

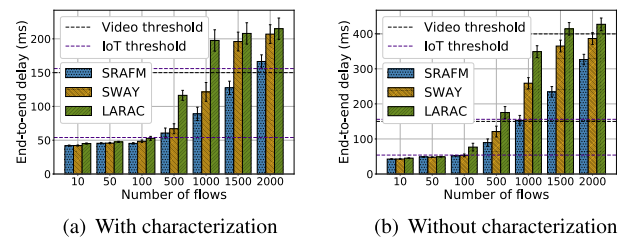


FIGURE 5. End-to-end delay evaluation.

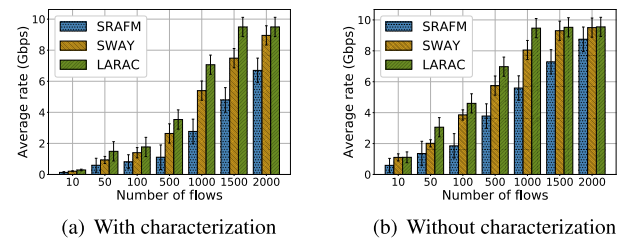


FIGURE 6. Rate allocation evaluation.

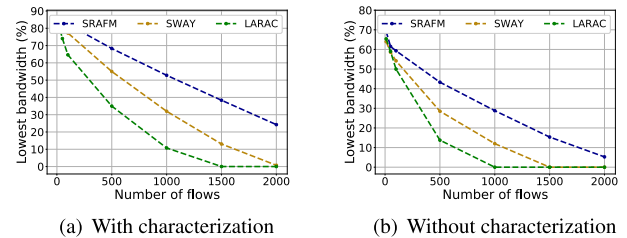


FIGURE 7. Bandwidth availability evaluation.

and consequently its system cost continues to increase. For example, in Figure 4(b) with 2000 non-characterized flows, the system cost of SRAFM is greater than the cost of LARAC because of the flow acceptance by SRAFM, but its rejection by LARAC.

As shown in Figure 4(b), both of the algorithms (i.e., SRAFM and LARAC) reveal the necessity to perform traffic characterization before traffic management, which

reduces the system’s cost by an average of 47% compared to traffic management without the characterization process.

### 3) QoS REQUIREMENT PROVISIONING AND FLOW VIOLATION

Figures 5, 6, and 7 evaluate, respectively, the average of the end-to-end delay, the average of the flow rate allocation, and the lowest bandwidth availability after performing flow allocations over the Abilene topology. From these figures, it is evident that SRAFM obtains the highest network performance as compared to SWAY and LARAC, especially with the increasing number of flows.

SRAFM reduces, on average, the end-to-end delay by 21% and 34% as compared to SWAY and LARAC, respectively. Traffic characterization improves the system end-to-end delay with more than 56%, as compared to the management of the heterogeneous flows without characterization, particularly with a huge set of flows.

Figures 6 and 7 evaluate the average rate allocation and the lowest available bandwidth after performing flows rate allocations in the network. SRAFM reduces on average the rate allocation as a result of the distributed approach by 32% and 48% as compared to SWAY and LARAC, respectively. SRAFM achieves an average improvement of 27% and 36% in terms of available bandwidth as compared to SWAY and LARAC, respectively. Furthermore, with the benchmarks schemes, the lowest available bandwidth achieves 0% with huge sets of flows (i.e., 1500 and 2000 flows), which is not the case with SRAFM. Finally, as shown in Figures 7(a) and 7(b), controlling bandwidth-hungry flows using flow characterization provides a 22% improvement in bandwidth availability and enables postponing network congestion in terms of flow numbers.

This improvement in terms of delay and bandwidth availability depends on the routing strategy of each algorithm. With SRAFM, the rate allocation on each path for each flow depends on the rate requirements of all the flows. In other words, the SRAFM algorithm allocates network flows distributively to network paths that have low bandwidth utilization due to the dual variable, which presents a penalty in the cost minimization function. Furthermore, with SRAFM, the selected solution is related to two constrained

metrics, the delay and bandwidth thresholds. On the other hand, with the current state of the network, the benchmark algorithms perform flow-per-flow allocation for the whole set of flows. Thus, at a limitation level, the new services are rejected because of the paths’ insufficient bandwidth (with the SWAY algorithm) and high latency (with both the SWAY and LARAC algorithms).

Figure 8 evaluates the end-to-end delay, the average of rate allocation, and the lowest bandwidth availability, respectively, after the allocation of characterized flows over the AttMpls topology. From Figures 8(a), 8(b), and 8(c), it is evident that SRAFM obtains the highest network performance compared to SWAY and LARAC, and also compared to the small-scale Abilene topology. SRAFM achieves 47% and 56% reduction in end-to-end delay compared to SWAY and LARAC, respectively. Our algorithm achieves 14% reduction in end-to-end delay with the AttMpls topology compared to the Abilene network, with the huge sets of flows. In terms of rate allocation, with AttMpls, SRAFM reduces the average rate allocation by 62% and 74% compared to SWAY and LARAC, respectively, and by 49% compared to the Abilene network. With AttMpls, SRAFM also improves the lowest available bandwidth by 28% compared to the Abilene network. This improvement in terms of network performance with a large-scale network is related to the distributed rate allocation approach over the whole topology deployed by SRAFM and so for path delay, which is related not only to the number of links but also to the number of flows processed throughout the path. On the other hand, with local-based path selection strategies like SWAY and LARAC, the average rate allocation remains similar compared to the Abilene network, and the end-to-end delay increases because of the large-scale setting.

This SRAFM’s distributed rate allocation approach has an improvement not only on the end-to-end delay and the rate allocation but also on the violated flows. Figure 9 and Table 5 show the outperformance of our proposed scheme in terms of flow violation as compared to the benchmark schemes in the Abilene and AttMpls networks, particularly with a massive number of flows.

As shown in Figure 9(a), with the Abilene topology, SWAY and LARAC achieve the limitation level from 1500 flows; but with the AttMpls topology, the limitation level is achieved

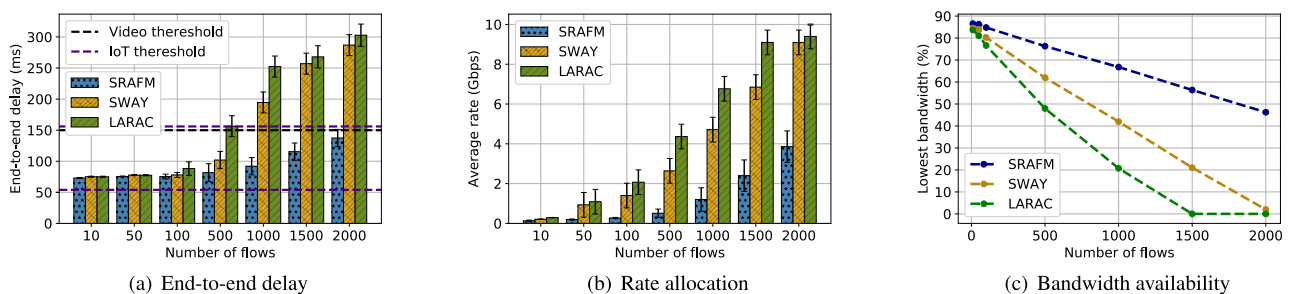


FIGURE 8. Varying number of characterized flows managed over a large-scale network.

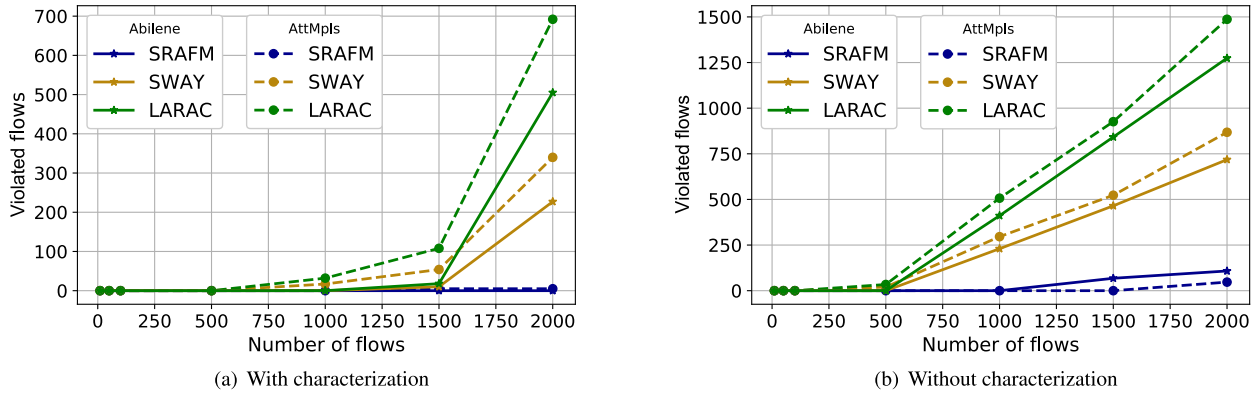


FIGURE 9. QoS violated flows within Abilene and AttMpls topologies.

TABLE 5. Examples of values for QoS violated flows.

Abilene topology		Characterized flows						Non-characterized flows							
		10	50	100	500	1000	1500	2000	10	50	100	500	1000	1500	2000
Algorithms	SRAFM	0	0	0	0	0	0	0	0	0	0	0	0	68	108
	SWAY	0	0	0	0	0	10	227	0	0	0	0	230	465	718
	LARAC	0	0	0	0	0	18	505	0	0	0	0	411	842	1274
AttMpls topology		Characterized flows						Non-characterized flows							
		10	50	100	500	1000	1500	2000	10	50	100	500	1000	1500	2000
Algorithms	SRAFM	0	0	0	0	0	0	0	0	0	0	0	0	0	47
	SWAY	0	0	0	0	17	54	340	0	0	0	20	296	523	868
	LARAC	0	0	0	0	32	108	692	0	0	0	34	507	926	1487

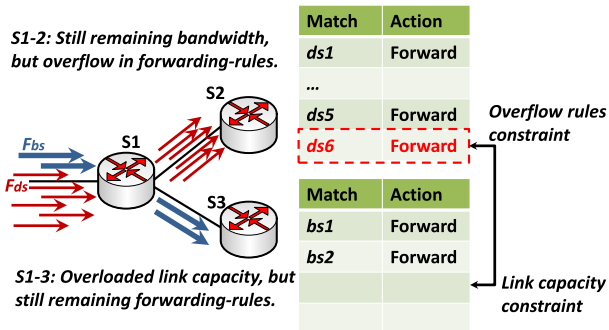


FIGURE 10. Conflict of device forwarding-rules overflow and link capacity overload in a large IoT environment.

from 1000 characterized flows. With 2000 characterized flows in the Abilene topology, 11% and 29% of the whole set are rejected by SWAY and LARAC, respectively. In the AttMpls topology, likewise, 17% and 36% of the whole set are rejected by SWAY and LARAC, respectively. This increase in violated flows in the AttMpls topology compared to Abilene with SWAY and LARAC is due to the local rate allocation strategy and the increase in terms of end-to-end delay. Meanwhile, the distributed rate allocation approach adopted by SRAFM enables an extenuated use of the topology resources, which leads to outperformance in terms of flow violation, with 0%. Figure 9(b) proves

that traffic characterization diminishes significantly the percentage of rejected flows. Without flow characterization, SRAFM achieves the limitation level with a set of 1500 flows over the Abilene topology, and with a set of 2000 flows over the AttMpls topology. SRAFM reduces the percentage of QoS flow violations with 2000 flows by 30% and 58% in the Abilene network compared to SWAY and LARAC, respectively. In AttMpls, SRAFM reduces the percentage of QoS flow violations with 2000 flows by 41% and 71% compared to SWAY and LARAC, respectively.

VIII. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we proposed a multi-flow management scheme for a software-defined smart digital campus network, taking into account the available network resources, the heterogeneous flow types, and their different QoS requirements. We presented a unified fully-programmable architecture, a distributed end-host-based flow characterization plane, a controller design with a proactive and reactive flow management strategy, and a centralized software-defined optimization model to manage the massive heterogeneous flows generated from thousands of interconnected devices.

We considered heterogeneous flows as either delay-sensitive or bandwidth-hungry services, but all as loss-sensitive. We introduced an approximation algorithm to relax the NP-hard proposed optimization problem. Furthermore, a per-flow decomposed optimization algorithm was deployed



using LDD and sub-gradient methods to calculate the optimal routing paths for the whole set of flows on the basis of a distributed rate allocation design. The simulation results showed the outperformance of our proposed scheme in terms of reduction in system cost, end-to-end delay, average rate allocation, and rejected flow percentages.

The proposed SRAFM solution for a smart campus environment could be implemented in a large variety of smart networks, such as hospitals, small and medium-sized enterprises (SMEs), governmental office buildings, etc. It is also possible to implement the SRAFM optimization model over a large IoT environment. However, in such an environment, the limited number of flow rules to be configured on an SDN-forwarding device should be taken into account. In fact, without the rule-constraint, even when the remaining available bandwidth of a specific link is sufficient to accommodate new flow rates, the maximum number of flow rules to be configured on an SDN-forwarding device remains a constraint. Thus, this creates a conflict of device forwarding rules overflow, which produces flow rejection, as shown in Figure 10.

Our future work will focus on the efficiency and scalability of the control plane, on the deployment of the cluster-based distributed controller technology, and on the deployment of the distributed flow characterization solution, since all these issues have significant impacts on network performance with massive data streams.

## REFERENCES

- [1] T. Sutjarittham, H. Habibi Gharakheili, S. S. Kanhere, and V. Sivaraman, "Experiences with IoT and AI in a smart campus for optimizing classroom usage," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7595–7607, Oct. 2019.
- [2] A.-M. Yang, S.-S. Li, C.-H. Ren, H.-X. Liu, Y. Han, and L. Liu, "Situational awareness system in the smart campus," *IEEE Access*, vol. 6, pp. 63976–63986, 2018.
- [3] 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. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPs)*, May 2017, pp. 559–564.
- [4] V. Uskov, A. Pandey, J. P. Bakken, and V. S. Margapuri, "Smart engineering education: The ontology of Internet-of-Things applications," in *Proc. IEEE Global Eng. Edu. Conf. (EDUCON)*, Apr. 2016, pp. 476–481.
- [5] V. L. Uskov, R. J. Howlett, and L. C. Jain, *Smart Education and Smart e-Learning*, vol. 41. Cham, Switzerland: Springer, 2015, doi: 10.1007/978-3-319-19875-0.
- [6] Cisco Systems. Accessed: Jul. 12, 2019. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/industries/education/connected-campus.html>
- [7] Campus Management Corp. Accessed: Jul. 7, 2019. [Online]. Available: <https://www.campusmanagement.com/>
- [8] Deloitte. Accessed: Jul. 11, 2019. [Online]. Available: <https://www2.deloitte.com/us/en/pages/consulting/solutions/next-generation-smart-campus.html>
- [9] Ruckus Wireless. Accessed: Jul. 16, 2019. [Online]. Available: <https://webresources.ruckuswireless.com/pdf/solution-briefs/sb-smartcampus-ebook.pdf>
- [10] Cisco Visual Networking Index, "Cisco visual networking index: Forecast and methodology 2015–2020," Cisco, San Jose, CA, USA, White Paper, 2015. Accessed: Apr. 22, 2019. [Online]. Available: <https://static1.squarespace.com/static/54496f89e4b0ad2be6456bc7/t/57b386f8e4fcb59cf4894907/1471383319341/Cisco+Forecast+2015-2020.pdf>
- [11] M. Ersue, D. Romascanu, J. Schoenwaelder, and U. Herberg, *Management of Networks With Constrained Devices: Problem Statement and Requirements*, document RFC 7547, IETF, 2015.
- [12] C. Bormann, M. Ersue, and A. Keranen, *Terminology for Constrained-Node Networks*. Fremont, CA, USA: Internet Engineering Task Force, 2014, pp. 1721–2070.
- [13] N. Saha, S. Bera, and S. Misra, "Sway: Traffic-aware QoS routing in software-defined IoT," *IEEE Trans. Emerg. Topics Comput.*, early access, Jun. 2018, doi: 10.1109/TETC.2018.2847296.
- [14] J. D. Saunders, C. R. McClure, and L. H. Mandel, "Broadband applications: Categories, requirements, and future frameworks," 1st Monday, 2012, vol. 17, no. 11.
- [15] Z. Wang, *Internet QoS: Architectures and Mechanisms for Quality of Service*. San Mateo, CA, USA: Morgan Kaufmann, 2001.
- [16] D. F. Macedo, D. Guedes, L. F. Vieira, M. A. Vieira, and M. Nogueira, "Programmable networks—From software-defined radio to software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 1102–1125, 2nd Quart., 2015.
- [17] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 27–51, 1st Quart., 2015.
- [18] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.
- [19] Z. Shu, J. Wan, J. Lin, S. Wang, D. Li, S. Rho, and C. Yang, "Traffic engineering in software-defined networking: Measurement and management," *IEEE Access*, vol. 4, pp. 3246–3256, 2016.
- [20] M. Fazel, H. Hindi, and S. P. Boyd, "Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices," in *Proc. Amer. Control Conf.*, vol. 3, 2003, pp. 2156–2162.
- [21] A. Juttner, B. Szviatovski, I. Mees, and Z. Rajko, "Lagrange relaxation based method for the QoS routing problem," in *Proc. IEEE Conf. Comput. Commun. 20th Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, vol. 2, Jun. 2001, pp. 859–868.
- [22] M. Karakus and A. Duresi, "Quality of service (QoS) in software defined networking (SDN): A survey," *J. Netw. Comput. Appl.*, vol. 80, pp. 200–218, Feb. 2017.
- [23] H. H. Gharakheili, V. Sivaraman, A. Vishwanath, L. Exton, J. Matthews, and C. Russell, "SDN APIs and models for two-sided resource management in broadband access networks," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 4, pp. 823–834, Dec. 2016.
- [24] H. Habibi Gharakheili, V. Sivaraman, T. Moors, A. Vishwanath, J. Matthews, and C. Russell, "Enabling fast and slow lanes for content providers using software defined networking," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1373–1385, Jun. 2017.
- [25] J. W. Guck, A. Van Bemten, and W. Kellerer, "DetServ: Network models for real-time QoS provisioning in SDN-based industrial environments," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 4, pp. 1003–1017, Dec. 2017.
- [26] D. Henneke, L. Wisniewski, and J. Jasperneite, "Analysis of realizing a future industrial network by means of software-defined networking (SDN)," in *Proc. IEEE World Conf. Factory Commun. Syst. (WFCS)*, May 2016, pp. 1–4.
- [27] Huawei Technologies, "SDN, a new definition of next-generation campus network," Huawei, Shenzhen, China, White Paper, 2013. Accessed: Dec. 19, 2019. [Online]. Available: [https://www.huawei.com/ilink/en/download/HW\\_274562](https://www.huawei.com/ilink/en/download/HW_274562)
- [28] H. E. Egilmez, S. T. Dane, K. T. Bagci, and A. M. Tekalp, "OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end quality of service over software-defined networks," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Dec. 2012, pp. 1–8.
- [29] H. E. Egilmez, S. Civanlar, and A. M. Tekalp, "An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 710–715, Apr. 2013.
- [30] T.-F. Yu, K. Wang, and Y.-H. Hsu, "Adaptive routing for video streaming with QoS support over SDN networks," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2015, pp. 318–323.
- [31] J. W. Guck, A. Van Bemten, M. Reisslein, and W. Kellerer, "Unicast QoS routing algorithms for SDN: A comprehensive survey and performance evaluation," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 388–415, 1st Quart., 2018.
- [32] L. Guo and I. Matta, "Search space reduction in QoS routing," *Comput. Netw.*, vol. 41, no. 1, pp. 73–88, Jan. 2003.
- [33] *Yens k-Shortest Paths Algorithm*. Accessed: Nov. 19, 2019. [Online]. Available: <https://github.com/guilhermemm/k-shortest-path>

- [34] S.-C. Lin, I. F. Akyildiz, P. Wang, and M. Luo, "QoS-aware adaptive routing in multi-layer hierarchical software defined networks: A reinforcement learning approach," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jun. 2016, pp. 25–33.
- [35] M. Beshley, M. Seliuchenko, O. Panchenko, and A. Polishuk, "Adaptive flow routing model in SDN," in *Proc. 14th Int. Conf. Exper. Designing Appl. CAD Syst. Microelectron. (CADSM)*, 2017, pp. 298–302.
- [36] D. L. C. Dutra, M. Bagaa, T. Taleb, and K. Samdanis, "Ensuring end-to-end QoS based on multi-paths routing using SDN technology," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.
- [37] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Classifying IoT devices in smart environments using network traffic characteristics," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1745–1759, Aug. 2019.
- [38] H. C. Y. Chan and L. Chan, "Smart library and smart campus," *J. Service Sci. Manage.*, vol. 11, no. 06, pp. 543–564, 2018.
- [39] Y. Chen, T. Farley, and N. Ye, "QoS requirements of network applications on the Internet," *Inf. Knowl. Syst. Manage.*, vol. 4, no. 1, pp. 55–76, Mar. 2004.
- [40] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel, A. Puschmann, A. Mitschele-Thiel, M. Muller, T. Elste, and M. Windisch, "Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 70–78, Feb. 2017.
- [41] J. Mocnej, A. Pekar, W. K. Seah, and I. Zolotova, "Network traffic characteristics of the IoT application use cases," School Eng. Comput. Sci., Victoria Univ. Wellington, Wellington, New Zealand, Tech. Rep. ECSTR18-01, 2018. [Online]. Available: [https://ecs.wgtn.ac.nz/foswiki/pub/Main/TechnicalReportSeries/IoT\\_network\\_technologies\\_embfonts.pdf](https://ecs.wgtn.ac.nz/foswiki/pub/Main/TechnicalReportSeries/IoT_network_technologies_embfonts.pdf)
- [42] Huawei Technologies, "Huawei agile campus network solution," Huawei, Shenzhen, China, White Paper, 2015. [Online]. Available: [https://www.karma-group.ru/upload/iblock/496/Huawei%20Agile%20Campus%20Network%20Solution%20Brochure%20\(Detailed%20Version\)%20.pdf](https://www.karma-group.ru/upload/iblock/496/Huawei%20Agile%20Campus%20Network%20Solution%20Brochure%20(Detailed%20Version)%20.pdf)
- [43] J. Nunez-Martinez, J. Baranda, I. Pascual, and J. Mangues-Bafalluy, "WiseHAUL: An SDN-empowered wireless small cell backhaul testbed," in *Proc. IEEE 17th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Coimbra, Portugal, Jun. 2016, pp. 1–3.
- [44] K. Seppänen, J. Kilpi, and T. Suihko, "Integrating WMN based mobile backhaul with SDN control," in *Proc. Int. Internet Things Summit*. Springer, 2014, pp. 222–233.
- [45] P. Shantharama, A. S. Thyagaturu, N. Karakoc, L. Ferrari, M. Reisslein, and A. Scaglione, "LayBack: SDN management of multi-access edge computing (MEC) for network access services and radio resource sharing," *IEEE Access*, vol. 6, pp. 57545–57561, 2018.
- [46] M. Grandi, D. Camps-Mur, A. Betzler, J. J. Aleixendri, and M. Catalan-Cid, "SWAM: SDN-based Wi-Fi small cells with joint access-backhaul and multi-tenant capabilities," in *Proc. IEEE/ACM 26th Int. Symp. Qual. Service (IWQoS)*, Jun. 2018, pp. 1–2.
- [47] R. Santos and A. Kassler, "Small cell wireless backhaul reconfiguration using software-defined networking," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017, pp. 1–6.
- [48] R. Santos and A. Kassler, "A SDN controller architecture for small cell wireless backhaul using a LTE control channel," in *Proc. IEEE 17th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2016, pp. 1–3.
- [49] R. A. Cacheda, D. C. García, A. Cuevas, F. J. G. Castano, J. H. Sánchez, G. Koltsidas, V. Mancuso, J. I. M. Novella, S. Oh, and A. Pantò, "Qos requirements for multimedia services," in *Resource Management in Satellite Networks*. Springer, 2007, pp. 67–94.
- [50] L. Cui, F. R. Yu, and Q. Yan, "When big data meets software-defined networking: SDN for big data and big data for SDN," *IEEE Netw.*, vol. 30, no. 1, pp. 58–65, Jan. 2016.
- [51] M. S. Hadi, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmiringhani, "Big data analytics for wireless and wired network design: A survey," *Comput. Netw.*, vol. 132, pp. 180–199, Feb. 2018.
- [52] T.-Y. Mu, A. Al-Fuqaha, K. Shuaib, F. M. Sallabi, and J. Qadir, "SDN flow entry management using reinforcement learning," *ACM Trans. Auto. Adapt. Syst.*, vol. 13, no. 2, pp. 1–23, Nov. 2018.
- [53] J. Wu, M. Dong, K. Ota, J. Li, W. Yang, and M. Wang, "Fog-computing-enabled cognitive network function virtualization for an information-centric future Internet," *IEEE Commun. Mag.*, vol. 57, no. 7, pp. 48–54, Jul. 2019.
- [54] A. R. Curtis, W. Kim, and P. Yalagandula, "Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 1629–1637.
- [55] F. Baker, J. Polk, and M. Dolly, *A Differentiated Services Code Point (DSCP) for Capacity-Admitted Traffic*, document RFC 5865, Internet Engineering Task Force, 2010.
- [56] A. C. Begen, Y. Altunbasak, O. Ergun, and M. H. Ammar, "Multipath selection for multiple description video streaming over overlay networks," *Signal Process., Image Commun.*, vol. 20, no. 1, pp. 39–60, Jan. 2005.
- [57] D. Jurca and P. Frossard, "Media flow rate allocation in multipath networks," *IEEE Trans. Multimedia*, vol. 9, no. 6, pp. 1227–1240, Oct. 2007.
- [58] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: New insights," *Struct. Multidisciplinary Optim.*, vol. 41, no. 6, pp. 853–862, Jun. 2010.
- [59] Y. Liu, D. Niu, and B. Li, "Delay-optimized video traffic routing in software-defined interdatacenter networks," *IEEE Trans. Multimedia*, vol. 18, no. 5, pp. 865–878, May 2016.
- [60] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1452–1463, Aug. 2006.
- [61] S. Liao, W. Cheng, W. Liu, Z. Yang, and Y. Ding, "Distributed optimization for utility-energy tradeoff in wireless sensor networks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2007, pp. 3190–3194.
- [62] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, p. 48, 2009.
- [63] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.
- [64] A. Mendiola, J. Astorga, E. Jacob, and M. Higuero, "A survey on the contributions of software-defined networking to traffic engineering," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 918–953, 2nd Quart., 2017.
- [65] *Floodlight—How to Collect Switch Statistics (and Compute Bandwidth Utilization)*. Accessed: Aug. 22, 2019. [Online]. Available: <https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/21856267/How+to+Collect+Switch+Statistics+and+Compute+Bandwidth+Utilization>
- [66] S. Krishnan, N. Montavont, E. Njedjou, S. Veerapalli, and A. Yegin, *Link-Layer Event Notifications for Detecting Network Attachments*, document RFC 4957, Internet Engineering Task Force, 2007.
- [67] *High-Performance Dynamic Programming Language for Numerical Computing*. Accessed: Apr. 23, 2020. [Online]. Available: <https://julialang.org/>
- [68] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Rev.*, vol. 59, no. 1, pp. 65–98, Jan. 2017.
- [69] M. Lubin and I. Dunning, "Computing in operations research using Julia," *INFORMS J. Comput.*, vol. 27, no. 2, pp. 238–248, Apr. 2015.
- [70] M. Udell, K. Mohan, D. Zeng, J. Hong, S. Diamond, and S. Boyd, "Convex optimization in Julia," in *Proc. 1st Workshop High Perform. Tech. Comput. Dyn. Lang.*, Nov. 2014, pp. 18–28.
- [71] T. Besard, C. Foket, and B. De Sutter, "Effective extensible programming: Unleashing Julia on GPUs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 4, pp. 827–841, Apr. 2019.
- [72] I. Dunning, J. Huchette, and M. Lubin, "JuMP: A modeling language for mathematical optimization," *SIAM Rev.*, vol. 59, no. 2, pp. 295–320, Jan. 2017.
- [73] *Ipopt Solver in Julia*. Accessed: Apr. 2, 2019. [Online]. Available: <https://github.com/JuliaOpt/Ipopt.jl>
- [74] S. Orłowski, R. Wessälly, M. Pióro, and A. Tomaszewski, "SNDlib 1.0—Survivable network design library," *Netw. Int. J.*, vol. 55, no. 3, pp. 276–286, 2010.
- [75] Cisco-CCNA. *Computer Networking Notes*. Accessed: Mar. 15, 2020. [Online]. Available: <https://www.computernetworkingnotes.com/ccna-study-guide/ospf-metric-cost-calculation-formula-explained.html>



**YOSRA NJAH** received the B.Sc. and M.Sc. degrees in telecommunications from the National Engineering School of Tunis (ENIT), Tunisia, in 2012 and 2013, respectively. She is currently pursuing the Ph.D. degree with the Synchronmedia Laboratory, École de Technologie Supérieure (ÉTS), University of Quebec, Montreal, Canada. She performed her engineering and master of research projects in HEIA-FR, the School of Engineering and Architecture of Fribourg, Switzerland,

where she has worked on the development of an observatory based on honeypot and machine learning techniques for malicious behavior and attacks threats in the DarkNet. Her current research interests include programmable networks, optimization, and data analytics for Internet-of-Things and inter-data center networks.



**CHUAN PHAM** (Graduate Student Member, IEEE) received the B.S. degree from the Ho Chi Minh City University of Transport, in 2004, the master's degree from the Ho Chi Minh City University of Sciences, in 2008, and the Ph.D. degree from Kyung Hee University, in 2017, all in electrical and computer engineering. He was a Postdoctoral Fellow with the Department of Computer Science and Engineering, Kyung Hee University, in August 2017. Since 2018, he has

been a Postdoctoral Fellow with the Synchronmedia Laboratory, École de Technologie Supérieure (ÉTS), University of Quebec, Montreal, Canada. His research interests include applying analytic techniques of optimization and machine learning to network applications in terms of cloud and mobile-edge computing, datacenters, resource allocation for virtual networks, and the Internet of Things.



**MOHAMED CHERIET** (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science from the University of Pierre and Marie Curie (Paris VI), in 1985 and 1988, respectively. Since 1992, he has been a Professor with the Automation Engineering Department, École de Technologie Supérieure (ÉTS), University of Quebec, Montreal, where he was appointed as a Full Professor, in 1998. He is also the Founder and the Director of the Synchronmedia Laboratory, ÉTS,

which targets multimedia communication in telepresence applications. His research interests include cloud computing and network virtualization and softwarisation. He is also an Expert in computational intelligence, pattern recognition, machine learning, and perception. He has published more than 450 technical articles in the field. He is a Fellow of the International Association of Pattern Recognition (IAPR), the Canadian Academy of Engineering (CAE), the Engineering Institute of Canada (EIC), and the Engineers Canada (EC). He was a recipient of the 2016 IEEE J. M. Ham Outstanding Engineering Educator Award and the 2012 Queen Elizabeth II Diamond Jubilee Medal. As the Tier 1 Canada Research Chair on sustainable and smart eco-cloud, he leads the establishment of the first smart university campus in Canada, created as a hub for innovation and productivity at Montreal. He is the Founder and the Former Chair of the IEEE Montreal Chapter of Computational Intelligent Systems (CIS), a Steering Committee Member of the IEEE Green ICT Initiative, and the Chair of the IEEE ICT Emissions Working Group. He serves on the editorial boards of several renowned journals and international conferences.

• • •