International Journal for Quality Research 15(1) 171–188
ISSN 1800-6450

Muhamed Begović [1]
Samir Čaušević
Elma Avdagić-Golub

Article info:
Received 09.01.2020
Accepted 04.05.2020

UDC – 005.6
DOI – 10.24874/IJQR15.01-10

QoS MANAGEMENT IN SOFTWARE DEFINED NETWORKS FOR IoT ENVIRONMENT: AN OVERVIEW

Abstract: *Software Defined Networks (SDN) offer a new architecture and different approach to the management of traffic and network resources. Unlike traditional networks, SDN separates control and data planes and enables centralized network management. Forwarding devices are programmable, network statistics are monitored and behavior adjusted to current traffic conditions. SDN is an answer for meeting the QoS (Quality of Service) demands of environments such as IoT (Internet of Things). This paper provides an overview of contemporary approaches in optimizing such complex scenery, and present-day mechanisms for balanced and efficient use of network capacities, ensuring QoS through traffic-aware routing and dynamically adaptive rules. Key new ideas were identified in optimizing the management of the SDN-IoT network. Ultimately, room for future research has been identified and the need to create a new comprehensive model tailored to the specificities of the SDN-IoT environment has been addressed.*

Keywords: *Software Defined Networks; IoT; QoS routing; Network management and control.*

1. Introduction

Modern telecommunications systems have undergone serious changes in the last two decades. The number of Internet users is steadily increasing, leading to an intensive expansion of the network, and in particular an increased number of terminal access devices, most of which are mobile and IoT (Internet of Things) devices. The amount of traffic being transmitted over the network has reached unexpected levels and is already exceeding zettabytes on an annual basis, according to Cisco reports (Cisco, 2019). The type of content being transferred has also changed significantly. From the plain text, through hypertext, audio-rich content, to high and ultra-high quality videos consumed daily by modern-day users. The same report predicts that by 2022, M2M (Machine to Machine) devices will make up more than 50% of globally networked devices, or exceed 14.6 billion. Network management and optimization of available processor, memory, and network resources in these conditions have become a very demanding task.

Network management with traditional architecture is demanding for several reasons. In the first place, it is very difficult to define what is considered the proper behavior of the network. Interactions between multiple routing protocols can produce uncertainty. Furthermore, each autonomous system on the Internet is independently configured, so that interaction between different policies of these autonomous systems can lead to unwanted

[1] Corresponding author: Begović Muhamed
Email: muhamed.begovic@fsk.unsa.ba

behavior. The network configuration in a classical network environment is distributed, with each device configured at a lower level and in a manner specific to the equipment manufacturer.

The typical implementation of IP routers and the architecture of classic networks burdens lower-level devices making them doubly responsible - for control and data plane tasks. This makes them complex, difficult to manage, and their distributed nature makes them extremely vulnerable to configuration errors. Network operators are forced to manually configure devices from different vendors located in networks to suit different applications. They often have to use limited tools, such as CLI (Command Line Interface), and some scripting tools, to translate higher-level configuration policies into a language understood by lower-level devices. Management and optimization of the network in such conditions become very complex, especially for extremely large networks with a large number of devices. Due to the distribution and closeness of the control plane, innovations are difficult to introduce because manufacturers do not allow the software to be modified on their devices, and waiting for fast adaptive changes is unacceptable in modern conditions.

One of the basic problems with classic network architectures is the dynamic nature of network applications. This implies that performance requirements, in terms of QoS, can change over time. Many applications work in different network environments, and data is transmitted over wires and wirelessly across a wide variety of devices. For applications to function effectively, the network substrate must be flexible enough and able to dynamically adapt to frequent and specific changes in the requirements of the applications and their environment. Achieving adequate control and manageability of network traffic in modern communications means implementing mechanisms that will monitor network traffic, collect statistics and perform analysis that will provide insight into the current as well as predict future conditions, and intelligently respond to changes in the structure and quantity of network traffic (Causevic & Begovic, 2019). On top of that, all of this has to be done in near-to-real-time.

Software-Defined Networks (SDNs) are a new paradigm that has been in the special focus of research in the last few years (Xia et al., 2015), (Masoudi & Ghaffari, 2016), (Kreutz et al., 2015). SDNs provide much greater and logically centralized control of the network. They are suitable for dynamic network environments with frequent and unpredictable changes, and for managing and optimizing routing in networks with a large number of devices and large amounts of traffic generated. SDN has offered a new network architecture that tries to overcome the limitations of traditional networks by separating the data and control planes so that devices at lower layers (switches and routers) are freed from jobs that require the use of network logic. They are relieved of the tasks of the control layer and follow only the instructions they receive from the controller above them and operate the data layer.

An increased number of devices means a more complex network and more difficult network control with an increased number of different QoS requirements. Maintaining network scalability in the IoT environment of a traditional distributed network control in which each device is managed individually and separately is no longer sustainable. Network control must be flexible and responsive to frequent and rapid changes to network conditions and major changes in terms of size and type of content that will inevitably occur in the future. Meeting a large number of different application requirements while maintaining the required QoS level for different applications is not possible without network automation and the application of centralized control that SDN synergy with IoT applications can provide.

This paper focuses on a review of current research that treats the application aspects of SDN as the basis for a new principle of network management and control. Paper deals with problems and limitations of present approaches and mechanisms for providing the desired level of service and network control by applying the SDN paradigm in specific environments, such as IoT and its applications. Special attention is devoted to determining problems in optimal paths decision making for load balancing using knowledge of network topology, and the problem of minimizing the assignment of new tasks to devices on both the control and data layers by a dynamic and adaptive offloading strategy (a strategy of load balancing on devices) with a focus on the environment for IoT applications.

The motivation for this paper was found in the absence of review papers that look at contemporary approaches to solving the problem of implementing SDN control and QoS management in an IoT environment. The contribution of this work is reflected in the overview of state-of-the-art research from the most recognized scientific databases in this field. The paper provides a new perspective and an appropriate systematization of papers addressing the issue of ensuring an adequate level of QoS for IoT applications by integrating the concept of SDN network control.

2. IoT environment specifics

IoT is not so new as a concept and has been around for several decades since Mark Wiser and the idea of "ubiquitous computing". The idea was first introduced back in 1988 and later described in the famous paper "The Computer for the Twenty-First Century" (Weiser, 1999). It predicts that computers would be embedded in everyday objects, that people would simultaneously interact with hundreds of computers through wireless communication. The concept was that over time, the device would sink into the reality of our everyday lives and work in the background without us even thinking about their existence. Today, IoT is one of the major focuses of ICT researchers and industry (Čolaković & Hadžialić, 2018; Bajic et al., 2020; Brous et al., 2020). Despite the great effort of leading institutions and academy, there is no precise definition of IoT. However, as a common understanding, IoT represents objects that are connected to the Internet, usually equipped with sensors, and that can exchange information and take action often without human intervention.

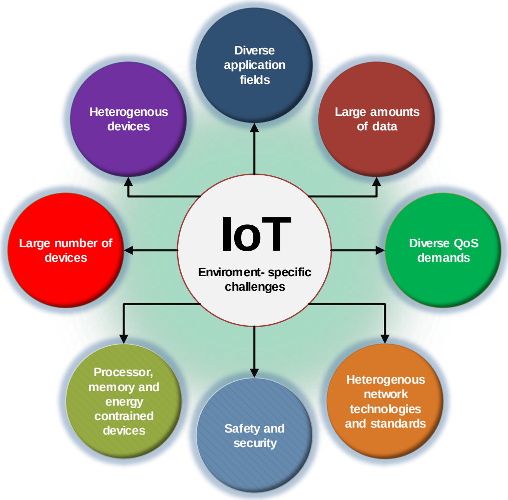

Figure 1. IoT environment-specific challenges

Although the idea of IoT has been around for a long time, the full implementation of IoT concepts requires a certain level of development of hardware and software as a prerequisite. Besides, the communications networks need to be capable to integrate these devices and successfully transmit the information they generate and receive, while providing the required level of service.

The IoT system is particularly demanding for control and management due to the diversity of applications, implementation areas, devices, and different architectures of network access (see Figure 1.). IoT applications are very diverse today and include smart cities (Zanella et al., 2014), health systems (Al-Hamadi & Chen, 2017), energy systems (Bedi et al., 2018), agriculture (Khanna & Kaur, 2019), and industrial IoT (Ghobakhloo, 2020), just to name a few. The variety of IoT deployment also implies very large differences in the requirements that applications will place before the communications network. The current model based on the internet best-effort service will not be able to meet such a variety of QoS requirements (I. Awan et al., 2014)(Ahmad et al., 2020). Application requirements can be very simple data transfer, but also more complex such as special requirements for security, delivery reliability, real-time delivery, etc. Complete industrial production facilities, city energy and transportation systems, essential decision-making processes, and even human safety and lives in the case of some health applications could in the future depend on the reliability of IoT device communication. Communication of IoT devices acting without human intervention also requires special dependability and control. Communication of IoT devices can be based on wired and wireless types of networks, using diverse architecture and standards such as Wi-Fi, WiMax, LR-WPAN, 2-5G mobile communications, Bluetooth, LoRaWAN, RFID, NFC, etc. (Ray, 2018).

When providing network services, additional attention must be paid to the very common limitations on memory and processing capacity of IoT devices. Power-constrained IoT devices are also not uncommon, so it is often a challenge to ensure energy efficiency (Lee et al., 2018). The challenges that IoT will bring in terms of scalability are reflected not only in the number of devices connected but also in the fact that many of them will have continuous sensory activity. Therefore, a large amount of information will be generated, constantly transmitted over the network, and exchanged between interconnected systems. This will certainly represent a great demand for network capacities.

The presented features of the IoT environment indicate the complexity of the tasks that the network will face in fulfilling the huge variety of QoS demands and provide a seamless communication platform. When introducing management and control solutions for an IoT network environment, many factors must be considered, including heterogeneity, security, energy efficiency, scalability, reliability. To deliver all this, the network must be based on a new architecture that is ready to operate with the ability to adapt to different requirements, acting responsively, dynamically, and intelligently responding and coordinating its operations.

3. Contemporary approaches and network management mechanisms in the SDN-IoT environment

Concepts that SDN introduced allowed for direct programmability of the network, simpler introduction of innovations, more detailed routing criteria, implementation of unique policies across different networks and domains, response to real-time changes, etc. (see Figure 2.). Still, harnessing the full potential of SDN networks is not an easy task, especially in specific network environments like IoT. Special attention and comprehensive approach are needed to consider all aspects that are affected by the proposed new solutions, algorithms, mechanisms, architecture organizations, etc.

The introduction of intelligence and automation in managing and controlling network operations has been around for a relatively long time as a concept (Clark et al., 2003). However, the application was still limited due to the complexity that arises as a result of classic network architectures and their distributed nature. Each node has only a partial view and competence in the network, which makes identifying network problems and behavior patterns a very difficult task, and acting following these insights in such an environment is almost impossible to achieve. Particular complexity arises when implementing decisions and actions that need to have a common effect on multiple devices or on multiple network domains at the same time (Mestres et al., 2017). This is why the SDN paradigm is an ideal partner for machine learning concepts, and a natural step in creating the preconditions for developing and managing future networks that will be smart, autonomous and capable of recognizing the problem, turning it into a set of appropriate actions, and delivering instructions to network devices on how to behave for the network to provide the required performance.

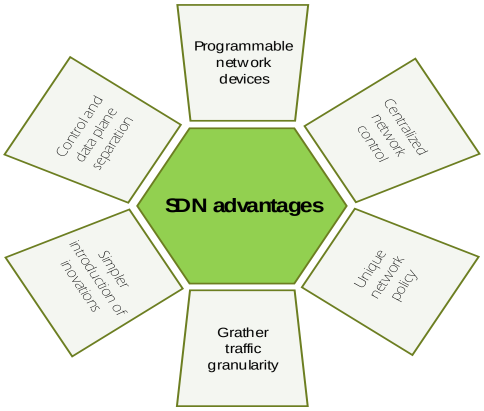

Figure 2. The main advantages of SDN

Applications of the SDN principles have come alive in various network environments, and the capabilities that SDN brings are exploited in combination with other upcoming paradigms. The management and control provided by SDN support the operation of various real-time applications (Yao et al., 2019), (AL-Tam & Correia, 2019), (Hou et al., 2018). SDN is receiving increasing attention in its application for another paradigm called the Internet of Things (IoT) as a relatively new research field (Salman et al., 2018)(Bizanis & Kuipers, 2016)(Tayyaba et al., 2017). Recent studies (see Table 1.) of traffic optimization using SDN-based postulates whose application would affect network environments such as IoT (a large number of devices, heterogeneous networks, wireless sensor networks, variable network traffic, and network conditions...) show that the authors generally pay attention to four basic groups of problems:

A. Offloading or Load Balancing (LB) on links and devices
B. Control Plane Organization (CPO)
C. Rule Placement (RP)
D. QoS Routing (QoSR)

The integration of SDN and IoT can ensure that the current behavior of the network is more properly detected and understood. Centralized, holistic management of the IoT environment might provide more accurate extraction of the network topology and its properties. Knowing the network and current network status, responding to changes in heterogeneous QoS requests by IoT applications can be in near-to-real time. Load balancing is a much more powerful tool with a new dynamic and automated dimension. Rule-based network control, conducted from a separate network control plane, can provide much less load on low-level devices. But, while the potential for integration of these two paradigms is unquestionable, many implementation examples available in the reference literature demonstrate the complexity of the procedure. Below, a review of the state-of-the-art literature on these observed implementation problems is made, and the positive aspects of the approach, as well as their limitations, are identified.

3.1 Achieving load balancing on links and devices

Compliance with the principles of load balancing, both on links and devices in SDN networks is a much more manageable task than in traditional network architectures because it allows the use of adaptive mechanisms that change the forwarding rules according to the current network status (Begović & Bajrić, 2017). A unique network view and access to network statistics give a clearer and more accurate insight into alternative routes and devices, allowing for a balanced load across the network.

Offloading (balancing the load) plays an essential role in the implementation of SDN for the traffic management of IoT applications (Misra & Saha, 2019).

Table 1. Overview of recent SDN load-balancing optimization research

Year	Paper	Treated problem			
		LB	CPO	RP	QoSR
2019	Traffic Load Balancing Using Software Defined Networking (SDN) Controller as Virtualized Network Function (Ejaz et al., 2019)	●	●		
2019	A Novel Approach to Rule Placement in Software-Defined Networks Based on OPTree (Li et al., 2019)			●	
2019	FlowStat: Adaptive Flow-Rule Placement for Per-Flow Statistics in SDN (Bera et al., 2019)			●	●
2019	Detour: Dynamic Task Offloading in Software-Defined Fog for IoT Applications (Misra & Saha, 2019)	●			
2019	Machine Learning Aided Load Balance Routing Scheme Considering Queue Utilization (Yao et al., 2019)	●	●		
2018	Fragmentation-Based Distributed Control System for Software-Defined Wireless Sensor Networks (Kobo et al., 2019)	●	●		
2018	An Efficient Route Management Framework for Load Balance and Overhead Reduction in SDN-Based Data Center Networks (Y. C. Wang & You, 2018)	●			
2018	SDN-Enabled Traffic-Aware Load Balancing for M2M Networks (Chen et al., 2018)	●			
2018	Sway: Traffic-Aware QoS Routing in Software-Defined IoT (Saha et al., 2018)				●
2018	Achieving High Scalability Through Hybrid Switching in Software-Defined Networking (Xu et al., 2018)	●			
2019	Fractional Switch Migration in Multi-Controller Software-Defined Networking (AL-Tam & Correia, 2019)	●	●		
2019	An innovative approach for real-time network traffic classification (Dias et al., 2019)				●
2018	Topology-Preserving Traffic Engineering for Hierarchical Multi-Domain SDN (Hua et al., 2018)		●		●
2018	Load-balancing routing in software defined networks with multiple controllers (H. Wang et al., 2018)	●	●		●
2019	An improved mechanism for flow rule installation in-band SDN (I. I. Awan et al., 2019)			●	
2019	Dynamic Load Balancing of Software-Defined Networking Based on Genetic-Ant Colony Optimization (Xue et al., 2019)	●			
2018	An Optimization Routing Algorithm Based on Segment Routing in Software-Defined Networks (Hou et al., 2018)	●			●
2019	Nature-inspired meta-heuristic algorithms for solving the load balancing problem in the software-defined network (Akbar Neghabi et al., 2019)	●			●
2018	EASM: Efficiency-aware switch migration for balancing controller loads in software-defined networking (Hu et al., 2019)	●	●		
2019	NSAF: An Approach for Ensuring Application-Aware Routing Based on Network QoS of Applications in SDN (Park et al., 2019)				●

The math is simple: the larger the network, the greater the amount of traffic that is generated, so the load balancing on the links is required. Higher traffic means higher demands on data layer devices for forwarding packets from source to destination, and this also brings out the need for load balancing on switching devices. Finally, more forwarding packets on switches mean more requests and more information being sent to the controller. According to the scenario already highlighted, the centralized SDN architecture applied to networks with multiple and heterogeneous IoT devices does not meet user requirements. The connection between the centralization of network control and the load-balancing concept, as well as the research path to solving the raised issues, is shown in Figure 3. For load balancing across networks, different nature-inspired meta-heuristic algorithms can be used. Nevertheless, research (Akbar Neghabi et al., 2019) shows that each of these approaches has its limitations besides the positives and they do not solve the problems of large network loads. Most of them cause communication overhead and have the problem of falling into the local optimum.

Genetic-Ant Colony Optimization (G-ACO) (Xue et al., 2019) is a hybrid algorithm created by combining features of genetic (Genetic Algorithm - GA) and the optimization algorithm of an ant colony (Ant Colony Optimization - ACO). GA is used to achieve fast global search, while ACO is used to achieve the optimal solution. In terms of increasing search speed to find the solution, the goal is met, but the parameters for path selection defined by the fitness function are inadequate and insufficient to determine the actual optimal path through the network.

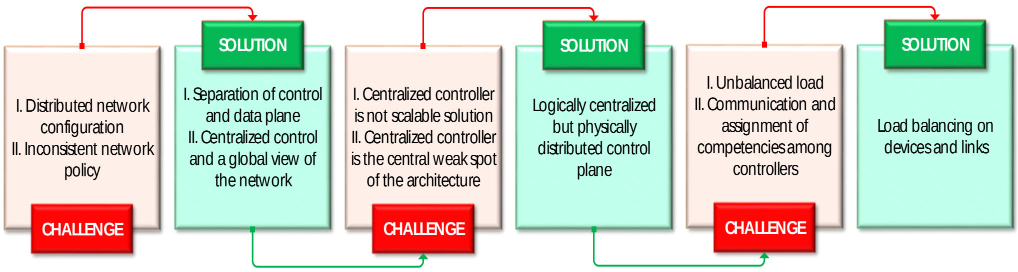

Figure 3. The flow of challenges and research since the emergence of centralized network control concept

Authors (Yao et al., 2019) treat the problems of routing schemes by utilizing load balancing and applying machine learning algorithms in SDN. Conventional routing algorithms such as Bellman-Ford, Link State algorithm, and Dijkstra algorithm have been addressed in terms of the ability to achieve load balancing for routers. These algorithms are time-consuming and do not achieve good performance for this task. The authors suggested how to overcome these problems by using three specific actions: (1) reducing network dimensions using the PCA (Principal Component Analysis) method, (2) predicting queue utilization, and (3) balancing load using DNN (Deep Neural Networks) networks. The architecture considers the use of software defined routers (SDRs), which are classified into local and central routers. The status of queues is monitored on local routers and routed to take care of load balancing. Real-time data is provided to the central router, which hides the DNN module and anticipates queue utilization, and feedback influences the behavior of the local router. PCA method for topology extraction results in an inadequate representation of the network topology, and it causes delay issues. Load balancing routing to select the next hop in the packet path from source to destination is based on queue utilization and realized using DNN networks. This criterion alone is not sufficient to achieve a proper balance. The proposed SDR approach also has a limitation in the case of large networks. The implementation of the central router, in this case, suffers serious load problems, which makes the solution non-scalable.

Authors (Ejaz et al., 2019) addressed the problem of load balancing by introducing a virtual SDN controller. When network traffic exceeds the defined load threshold, a new virtual SDN controller is created with the usage of network resource virtualization (NFV). The concept of primary and secondary virtual controllers is introduced, where the load of the primary controller in case of excessive requests is transferred to the secondary controller, which accepts part of the tasks and processes requests. The second controller is a replica of the primary controller with the same configuration that accepts the part of the traffic burden offloading the primary controller. The downside of the approach is reflected in an insufficient level of QoS parameters because of virtual implementation.

Switches migration is one of the proposed solutions for optimizing the allocation of resources in the organization of a control plane with multiple controllers (AL-Tam & Correia, 2019). This implies that the switch may change the controller to which it is connected, that is, the number of switches controlled by the controller is variable over time. However, in a distributed environment migration of switches is not always an adequate solution and leads to complexity in several dimensions (it is time-consuming, creates additional loads in terms of signaling, reduces the stability of network mapping).

EASM (Hu et al., 2019) (Efficiency Aware Switch Migration) is a strategy that also takes care of reducing unnecessary control traffic, which further burdens the network and increases the cost of migrating Switches. The LDM (Load Difference Matrix) matrix and the Trigger Factor are used to estimate load balancing. EASM-1 is used to detect the maximum load threshold crossing. EASM-2 selects switches for migration considering the load and cost of switch migration at the same time. Finally, the EASM-3 changes the map of switches to controllers. However, EASM has also not proven to be a scalable strategy for environments like IoT with numerous devices. A dynamic strategy for mapping switches to controllers for load-balancing and reducing the total number of changes made to the relevant controllers for switches can be found in (AL-Tam & Correia, 2019). To reduce this number, fractional level migration has been implemented. That is, the possibility of migrating at the level of flows is proposed. A heuristic algorithm is proposed to balance the load on the controllers. Two serious limitations can be noted in the solution described. For an SDN environment with multiple distributed controllers determining the right controllers for taking offloading action can be painstaking. Also, to predict the utilization of controller resources, it is important to predict an absolute load of switches, which will further improve the migration process of switches and reduce the number of new allocations.

L2RM (Low Cost Load Balancing Route Management) (Y. C. Wang & You, 2018) provides a solution for efficient routing management in data centers. The main focus is on achieving load balancing and reduce traffic overhead across the network using SDN, as well as taking care of the rational use of limited memory of switching devices by reducing the number of rules placement. There are two main concerns regarding this approach. First is the error-reporting mechanism which is currently not fast enough, and the second is the lack of large-scale environment performance evaluation.

Detour (Misra&Saha, 2019) is a proposal for the dynamic allocation of tasks and resources in software-defined Fog networks for IoT applications. IoT devices in architecture are connected to fog nodes using multi-hot IoT access points. The SDN controller collects network information through the southbound interface and performs optimal task allocation thanks to the global view of the network at its disposal. The idea is implemented by making decisions for three essential items: whether the task will be performed locally or on a remote device, selecting the ideal fog device, and selecting the optimal path to forward the task to another device. The M/M/1 model is applied to the task queue to select the appropriate application to perform the task after reaching the fog node. The limitation is the time consumed for sending a load balancing inquire to the SDN controller which cannot be neglected for achieving satisfactory QoS performance. Traffic-aware load balancing (TALB) in SDN-assisted IoT networks for Machine to Machine (M2M) communication is the focus in (Chen et al., 2018) for dealing with the bursty traffic and satisfying different QoS requirements. The key of the method lies in updating forwarding rules when the defined load measuring criteria reach previously defined levels which reduced overhead. This approach does not take into consideration the requirements of different types of applications which can be e.g. delay-tolerant and delay-intolerant.

3.2 Effects of control plane organization on network performance

With the rate of increase in the number of network devices and their nature of the continuous sensory activity, the centralized controller cannot handle the number of tasks it has to process. Considering this fact, the organization of a network with multiple controllers has recently been receiving increasing attention, namely the research into the performance of physically decentralized network control in the form of so-called "multi-controller environments" (Hu et al., 2018). A multi-controller environment can be implemented as a hierarchical one (Genge& Haller, 2016) with multiple lower-level controllers managed from a single higher-level central controller or distributed (Phemius et al., 2014) in which there are multiple controllers, but they are all of the same levels and perform tasks independently. In all cases of a decentralized control plane, load balancing on the controllers must be performed to ensure that the resources are used evenly and to prevent situations where the network management is not possible due to device overloading. According to all of the above, it is clear that the organization of the control plane and the balanced use of resources are closely related. Table 2. shows an overview of selected papers systematized according to the SDN architecture layer on which the balancing problem is treated and with emphasis on the type of control plane organization used in the particular case.

Efforts have been made to achieve load balancing in multi-controller networks at the same time on controllers and links (H. Wang et al., 2018) and thus reduce not only the controller response time but also the link utilization rate. It is emphasized that switch migration is an inadequate solution to load balancing on switches since the feature is not supported in OpenFlow specifications.

Two mechanisms are proposed to achieve load balancing in the form of updating the load information on a specific network domain (Area Bound Update) and updating the load information of the controller. However, it is noticeable that the approach has not achieved to influence the reduction of traffic between switching devices. Creating adequate conditions for successful traffic engineering in a multi-domain environment of a hierarchically organized control plane with hiding the details of domain topology as the main goal is a serious challenge (Hua et al., 2018). For this purpose, a control plane organization is proposed in which a local controller is assigned to each domain (geographical area) and the global controller has a global view of all local controllers in different domains. In this kind of organization, the load on the network is not balanced among the controllers. The proposed model does not offer updates to frequent network changes, and the global controller does not resolve failures among local controllers. A distributed SDN control mechanism has been implemented for wireless sensor networks, resulting in a synergy of these two paradigms called Software Defined Wireless Sensor Networks (Kobo et al., 2019). Devices used in these networks are simple, unintelligent, and capable only of forwarding, and all control is on the SDN controller. The benefit of combining wireless sensor networks with SDN is the fact that all energy-intensive operations move from the sensor nodes to the controller. A fragmentation method is used that involves organization with local and global controllers to overcome the problems of a physically centralized control plane. The Global Controller takes a holistic view of local controllers and their actions. Local controllers are located closer to the lower layer devices and control each of their device domains. The conducted research has confirmed that using the fragmentation method for organizing the control plane has improved the general performance and thus has the potential to be applied in a specific environment like industrial IoT. However, the authors also point out the need for an in-depth analysis of the scalability issue.

3.3 Effective rule placement approaches

The problem of forwarding rules placement to data layer devices is also attracting a great deal of attention in recent research (Komajwar & Korkmaz, 2018)(Huang et al., 2015). It is a very challenging problem due to the limited memory space of switches. When it is necessary to set new rules very often, the flow table becomes overloaded, and this can cause problems that reflect on the ability to achieve the desired QoS levels. In addition, when forwarding rules are set based on flow statistics, this also causes heavy burdens for the calculation of new packet forwarding routes, videlicet new rules (Bera et al., 2019).

One solution for placement flow forwarding rules in SDN networks is given in (I. I. Awan et al., 2019). In this proposal, the SDN controller defines the new rules and places them in the flow tables. If the switch receives a packet for which there are no rules in the flow table, the switch communicates directly with the controller. The controller sets the rules for individual switches and gives the best path to process packets. The packet is then transferred to the adjacent switch. In case the delay of the packet on the path from switch A to switch B is less than the time required for the missing rule to treat the packet from the controller to switch B, the packet will be rejected. A mechanism is proposed to check these two times and respond by delaying the packet forwarding on the switch to the precursor of switch B, for the minimum possible time, ensuring that the packet arrives after placing the rule that will be used for its forwarding to switch B. This approach uses keep-alive messages between controllers and switches to avoid additional signaling traffic in the network. However, receiving the aforementioned two types of delays still requires the extra time required for the calculation, especially in larger and more complex networks.

Many earlier approaches to this problem consider devices on the data layer as independent components, but the positional relationship between adjacent devices also has an impact on rule placement and should be considered. In this regard, two innovative strategies have been proposed to set the rules for two possible relations of adjacent devices - serial and parallel link (Li et al., 2019). A new OPTree (Ordered Predicate Tree) representation is proposed for checking whether a rule is contained in an existing rule. Two OPTree algorithms have been created for rule placement and searching.

The presented approach considers the positional relationship for adjacent devices, but the number of rules being set is still large, which also places a heavy burden on the controller. Additionally, in the case of fat-tree or star topologies, the number of rules is increased.FlowStat(Bera et al., 2019) is an adaptive rule placement according to flow-level statistics in the SDN.

Table 2. Overview of SDN load-balancing optimization categorized by layer and control plane organization

Layer	Paper	Network environment	Approach	Control plane organization	Parameters for validation
Data layer	(Yao et al., 2019)	General	Machine learning, Deep Neural Networks (DNN)	H	PLR, Throughput, Latency
	(Hou et al., 2018)	General	MOPSO (Multiple Objective Particle Swarm Optimization) algorithms	C	PLR, Latency, Jitter
	(Y. C. Wang & You, 2018)	Datacenter	ARM (Adaptive Route Modification) mechanism	C	Link load rate
	(Chen et al., 2018)	M2M	TABL (Traffic-aware load balancing) mechanism	C	Latency (service response time)
	(Xue et al., 2019)	General	Hybrid G-ACO (Genetic-Ant Colony Optimization) algorithm	C	PLR, RTT
	(H. Wang et al., 2018),	Large scale network	RDMAR (rounding based multi-area routing)	D	Link load rate, Latency (controller response time)
	(Xu et al., 2018)	Datacenter	Hybrid SDN and classical network architecture, RDSR (rounding-based scalable routing algorithm)	C	Load rate
	(Misra & Saha, 2019)	Fog based IoT	Greedy-heuristic algorithm	C	Latency, energy consumption
Control layer	(AL-Tam & Correia, 2019)	General	CQPP(convex quadratic programming problem), Heuristic algorithm	D	Load rate, Stability of switch-controller mapping
	(Ejaz et al., 2019)	Datacenter	Virtual controller replicating	D	RTT (ping), Throughput, Bandwidth, Latency
	(Kobo et al., 2019)	WSN	Fragmentation of control plane	H	RTT, PLR
	(Hu et al., 2019)	General	Switch migration	C	Latency, Throughput (controller)
	(H. Wang et al., 2018)	Large Scale Network	RDMAR (rounding based multi-area routing)	D	Link load rate, Latency (controller response time)

C - Centralized control plane architecture; D - Distributed control plane architecture; H - Hierarchical control plane architecture; PLR - Packet Loss Rate; RTT - Round Trip Time

The main objectives of the proposed solution are to reduce the memory load of forwarding devices with as few rules as possible while maintaining the visibility of flows and the preservation of flow statistics at the controller level. The maximum-flow-minimum-cost optimization problem is used to select the optimal packet forwarding path. The cost function takes into account the cost of activating the link, using the rules on the switch, and utilizing the link. It is important to emphasize that the controller does not consider the metric for selecting the packet forwarding path, which causes serious threats to QoS enhancement. As some flows that come to be processed on a switching device already have forwarding information in the flow table, the request to the controller will not be in that case. This automatically means that the statistics collected at the funnel level will not be ideal. The authors also point out the need for testing the proposed solution in very large and complex networks.

3.4 QoS Routing (QoSR) and traffic classification

The primary goal of QoS routing is to establish paths through the network to meet the desired level of parameters that make up QoS. All this should be done relying on network information about the availability and workload of individual network resources. In modern network environments, QoS routing selects the optimal paths while taking care of the diverse and rigorous end-to-end requirements of modern dynamic applications and services. To meet the different nature and requirements of applications in a heterogeneous environment, it is also necessary to adequately classify incoming traffic and obtain information that will allow prioritization and appropriate treatment of different flows.

In terms of traffic classification for real-time applications, there is an innovative approach for video streaming applications using the machine learning algorithm Naïve Bayes method (Dias et al., 2019). The primary purpose is to reduce the delay for multimedia applications. Therefore, user settings for traffic classification and delay tolerance for sensitive applications are considered. The classification can differentiate between different video streaming service requirements such as Netflix and Youtube (whether SD, HD, or UltraHD quality is transmitted). It should be noted, however, that the proposal of the Naïve Bayes Traffic Classification Algorithm also means a probabilistic approach, which is not an ideal solution in cases where precise traffic classification is required.

Traffic classification in SDN networks relying on the active participation of the control layer creates problems in large-scale networks. The SDN controller is not designed to handle the load in packet-level processing cases. Therefore, an environment with a large number of devices and the heterogeneous nature of traffic will certainly affect the degradation of controller performance. According to his, there is a classification approach (Hayes et al., 2018) that aims to move the most of classification operations from the control plane to the data plane. It introduces auxiliary data plane devices intending to reduce the controller burden. However, adding new elements to the data layer affects the infrastructure complexity of the network, which is not in the primary spirit of SDN. An additional device also means a trade-off between network performance and classification precision. The introduction of multiple classification data plane devices also causes issues in communication and synchronization in the operation of these devices.

The Network Situation-Aware Framework (NSAF) (Park et al., 2019) is a network routing management framework for applications. This framework calls upon a fact that controller performance, interactions of the controller, and other devices are not enough for achieving flawless network services. For meeting, QoS demands it is necessary to differ the applications as they have distinctive requirements. Parameters such as packet loss, bandwidth, delay, and jitter according to the defined types of IETF (Internet Engineering Task Force) DiffServ specifications have been incorporated for different service classes (application types). The downside in the approach is reflected in the fact that it is not possible to predict changes in the network and control paths when applications are executed. Another application-centric approach for QoSR in IoT environment can be found in Sway (Saha et al., 2018). It differentiates applications according to their requirements regarding packet loss and delay. Treating traffic based on these two parameters alone does not guarantee good performance in environments with many different devices generating diverse traffic.

One of the new algorithms for optimizing routing in SDN networks is based on the principle of Segment Routing (SR) (Hou et al., 2018). SR uses an end-to-end logical path made up of a sequence of segments. Each segment is represented by a midpoint. The key idea is to carry out the routing process using several midpoints, and switching devices in between need only know how to reach those midpoints. This avoids the need to create a large number of rules across all network devices, which saves memory. The segment refers to the instruction, and the node segment contains a unique tag to reach the next switch.

The concept of hybrid switching combines traditional and SDN switching (Xu et al., 2018) to take advantage of both approaches. Flows that are less important (those that are briefly retained on the network or have little impact over the total amount of traffic on the network) are treated by traditional routing to avoid unjustified requests toward controller and the memory space of the switches. Important flows (such as "elephant" flows) are treated using SDN controllers and switches. The primary principles of the proposal are (1) management of the network topology changes, (2) hybrid switching process enabling partial software-defined network management, and (3) integration of network policy requirements. The Dijkstra algorithm is used to implement the routing in this proposal, which results in the longer time it takes to forward packets from the source node to the destination node.

4. Perceived challenges and room for future research

According to available statistics, the number of connected devices online will exceed 75 billion by 2025 and has a steady upward trend (Statista Research Department, 2019). This large number of devices, many of which will have continuous sensor activity, will certainly generate a large amount of data for various applications, which will increase network traffic several times. Higher amounts of traffic also mean more complex network control and a more demanding process of passing data from the point of generation to the desired destination. Using a programmable network capable of dynamically adapting to current conditions and logically centralized traffic control is more needed today than ever.

Profound insight into the available literature shows that there is no comprehensive solution or a proper complete model that takes into account all the above-presented problems in creating an adequate traffic optimization in the idiosyncratic SDN environments like those needed by IoT applications. Presented approaches are based mainly on the partial treatment of problems or they do not take into account all the relevant criteria for achieving the conditions suitable for the successful integrated management and control of the SDN-IoT network.

The incompleteness of the presented approaches is reflected in various disadvantages such as: focusing on only one layer of architecture and neglecting the principle of balancing on all devices and links (AL-Tam & Correia, 2019)(Li et al., 2019)(Hua et al., 2018)(H. Wang et al., 2018)(I. I. Awan et al., 2019), weakness of the physical centralization of the control layer and non-scalable solutions (AL-Tam & Correia, 2019)(Bera et al., 2019)(Hu et al., 2019). Some approaches use inadequate algorithms and criteria when classifying traffic (Saha et al., 2018)(Dias et al., 2019)(Park et al., 2019), selecting optimal packet forwarding paths (Yao et al., 2019)(Hou et al., 2018)(Xu et al., 2018)(Xue et al., 2019), and the process of unburdening network devices and links (Hou et al., 2018)(Misra & Saha, 2019)(Chen et al., 2018).

From all the foregoing, the need to introduce a new model for the SDN controlled IoT environment is evident. This approach should provide mechanisms to determine the optimal paths for balancing the network and device load of the data and control layer using knowledge of network topology. The model should offer a way to minimize the assignment of new tasks to devices (both in the control and data layers) by a dynamic and adaptive strategy with a focus on the environment for IoT applications. The new approach should provide more complete and suitable criteria for the classification and forwarding of network traffic, and appropriate algorithms for the implementation of network operations that will provide the desired level of control and required QoS.

5. Conclusion

SDN paradigm approach enables adaptive offloading for IoT traffic. However, in a high-demanding IoT environment, the SDN network control must be ready to achieve logical centralization with physical distribution. The control plane organization will play a key role in providing the foundation for successful network management. Since control is SDN rule-based, it is very important to ensure a low level of network overhead and minimize the number of new assignments so that network resources can meet the challenge. The finer granularity of routing criteria in SDN with insight into current network statistics makes QoS routing more efficient and adjusted.

Ensuring network optimization for IoT applications undoubtedly requires a new approach. This approach should ensure that the complete network architecture and all key processes are viewed as an interdependent system. This new way of observing network management and control mechanisms in an IoT-like environment should achieve an appropriate level of QoS through offloading on devices and links, on data plane and control plane, while taking adaptive and dynamic actions. In future research, the author of this paper plan to develop and test the new SDN-IoT model according to defined criteria and to improve key performance parameters in the SDN-IoT environment.

References:

Ahmad, E., Alaslani, M., Dogar, F. R., & Shihada, B. (2020). Location-Aware, Context-Driven QoS for IoT Applications. *IEEE Systems Journal*, *14*(1), 232-243. https://doi.org/10.1109/JSYST.2019.2893913

Akbar Neghabi, A., Jafari Navimipour, N., Hosseinzadeh, M., & Rezaee, A. (2019). Nature-inspired meta-heuristic algorithms for solving the load balancing problem in the software-defined network. In *International Journal of Communication Systems* (Vol. 32, Issue 4). John Wiley and Sons Ltd. https://doi.org/10.1002/dac.3875

Al-Hamadi, H., & Chen, I. R. (2017). Trust-Based Decision Making for Health IoT Systems. *IEEE Internet of Things Journal*, *4*(5), 1408-1419. https://doi.org/10.1109/JIOT.2017.2736446

AL-Tam, F., & Correia, N. (2019). Fractional switch migration in multi-controller software-defined networking. *Computer Networks*, *157*, 1-10. https://doi.org/10.1016/j.comnet.2019.04.011

Awan, I. I., Shah, N., Imran, M., Shoaib, M., & Saeed, N. (2019). An improved mechanism for flow rule installation in-band SDN. *Journal of Systems Architecture*, *96*, 1-19. https://doi.org/10.1016/j.sysarc.2019.01.016

Awan, I., Younas, M., & Naveed, W. (2014). Modelling QoS in IoT applications. *Proceedings - 2014 International Conference on Network-Based Information Systems, NBiS 2014*, 99-105. https://doi.org/10.1109/NBiS.2014.97

Bajic, B., Suzic, N., Simeunovic, N., Moraca, S., & Rikalovic, A. (2020). Real-time Data Analytics Edge Computing Application for Industry 4.0: The Mahalanobis-Taguchi Approach. *International Journal of Industrial Engineering and Management, 11*(3), 145-156.

Bedi, G., Venayagamoorthy, G. K., Singh, R., Brooks, R. R., & Wang, K. C. (2018). Review of Internet of Things (IoT) in Electric Power and Energy Systems. In *IEEE Internet of Things Journal* (Vol. 5, Issue 2, pp. 847–870). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/JIOT.2018.2802704

Begović, M., & Bajrić, H. (2017). Solving Management Constraints of Traditional Networks using the Concept of Software Defined Networking. *International Journal of Soft Computing and Engineering (IJSCE)*, *7*(5), 7–12.

Bera, S., Misra, S., & Jamalipour, A. (2019). FlowStat: Adaptive Flow-Rule Placement for Per-Flow Statistics in SDN. *IEEE Journal on Selected Areas in Communications*, *37*(3), 530–539. https://doi.org/10.1109/JSAC.2019.2894239

Bizanis, N., & Kuipers, F. A. (2016). SDN and Virtualization Solutions for the Internet of Things: A Survey. In *IEEE Access* (Vol. 4, pp. 5591–5606). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2016.2607786

Brous, P., Janssen, M., & Herder, P. (2020). The dual effects of the Internet of Things (IoT): A systematic review of the benefits and risks of IoT adoption by organizations. In *International Journal of Information Management* (Vol. 51, p. 101952). Elsevier Ltd. https://doi.org/10.1016/j.ijinfomgt.2019.05.008

Causevic, S., & Begovic, M. (2019). Optimizing Traffic Routing in Different Network Environments Using the Concept of Software-Defined Networks. *Proceedings of the 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 409–414. https://doi.org/10.23919/mipro.2019.8756641

Chen, Y. J., Wang, L. C., Chen, M. C., Huang, P. M., & Chung, P. J. (2018). SDN-Enabled traffic-aware load balancing for M2M networks. *IEEE Internet of Things Journal*, *5*(3), 1797–1806. https://doi.org/10.1109/JIOT.2018.2812718

Cisco. (2019). *White paper, Cisco Visual Networking Index: Forecast and Trends, 2017–2022*. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.pdf

Clark, D. D., Partridge, C., Ramming, J. C., & Wroclawski, J. T. (2003). A knowledge plane for the internet. *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications - SIGCOMM '03*, 3. https://doi.org/10.1145/863955.863957

Čolaković, A., & Hadžialić, M. (2018). Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues. *Computer Networks*, *144*, 17–39. https://doi.org/10.1016/j.comnet.2018.07.017

Dias, K. L., Pongelupe, M. A., Caminhas, W. M., & de Errico, L. (2019). An innovative approach for real-time network traffic classification. *Computer Networks*, *158*, 143–157. https://doi.org/10.1016/j.comnet.2019.04.004

Ejaz, S., Iqbal, Z., Azmat Shah, P., Bukhari, B. H., Ali, A., & Aadil, F. (2019). Traffic Load Balancing Using Software Defined Networking (SDN) Controller as Virtualized Network Function. *IEEE Access*, *7*, 46646–46658. https://doi.org/10.1109/ACCESS.2019.2909356

Genge, B., & Haller, P. (2016). A hierarchical control plane for software-defined networks-based industrial control systems. *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, 73–81. https://doi.org/10.1109/IFIPNetworking.2016.7497208

Ghobakhloo, M. (2020). Industry 4.0, digitization, and opportunities for sustainability. *Journal of Cleaner Production*, *252*, p. 119869. Elsevier Ltd. https://doi.org/10.1016/j.jclepro.2019.119869

Hayes, M., Ng, B., Pekar, A., & Seah, W. K. G. (2018). Scalable Architecture for SDN Traffic Classification. *IEEE Systems Journal*, *12*(4), 3203-3214. https://doi.org/10.1109/JSYST.2017.2690259

Hou, X., Wu, M., & Zhao, M. (2018). An Optimization Routing Algorithm Based on Segment Routing in Software-Defined Networks. *Sensors (Basel, Switzerland)*, *19*(1). https://doi.org/10.3390/s19010049

Hu, T., Guo, Z., Yi, P., Baker, T., & Lan, J. (2018). Multi-controller Based Software-Defined Networking: A Survey. *IEEE Access*, *6*, 15980-15996. https://doi.org/10.1109/ACCESS.2018.2814738

Hu, T., Lan, J., Zhang, J., & Zhao, W. (2019). EASM: Efficiency-aware switch migration for balancing controller loads in software-defined networking. *Peer-to-Peer Networking and Applications*, *12*(2), 452-464. https://doi.org/10.1007/s12083-018-0632-6

Hua, J., Zhao, L., Zhang, S., Liu, Y., Ge, X., & Zhong, S. (2018). Topology-Preserving Traffic Engineering for Hierarchical Multi-Domain SDN. *Computer Networks*, *140*, 62-77. https://doi.org/10.1016/j.comnet.2018.04.011

Huang, H., Guo, S., Li, P., Ye, B., & Stojmenovic, I. (2015). Joint Optimization of Rule Placement and Traffic Engineering for QoS Provisioning in Software Defined Network. *IEEE Transactions on Computers*, *64*(12), 3488-3499. https://doi.org/10.1109/TC.2015.2401031

Khanna, A., & Kaur, S. (2019). Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture. In *Computers and Electronics in Agriculture* (Vol. 157, pp. 218-231). Elsevier B.V. https://doi.org/10.1016/j.compag.2018.12.039

Kobo, H. I., Abu-Mahfouz, A. M., & Hancke, G. P. (2019). Fragmentation-based distributed control system for software-defined wireless sensor networks. *IEEE Transactions on Industrial Informatics*, *15*(2), 901-910. https://doi.org/10.1109/TII.2018.2821129

Komajwar, S., & Korkmaz, T. (2018). Challenges and solutions to consistent data plane update in software defined networks. In *Computer Communications* (Vol. 130, pp. 50-59). Elsevier B.V. https://doi.org/10.1016/j.comcom.2018.08.008

Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, *103*(1), 14-76. https://doi.org/10.1109/JPROC.2014.2371999

Lee, H., Lee, K. J., Kim, H., & Lee, I. (2018). Wireless information and power exchange for energy-constrained device-to-device communications. *IEEE Internet of Things Journal*, *5*(4), 3175-3185. https://doi.org/10.1109/JIOT.2018.2836325

Li, W., Qin, Z., Li, K., Yin, H., & Ou, L. (2019). A Novel Approach to Rule Placement in Software-Defined Networks Based on OPTree. *IEEE Access*, *7*, 8689-8700. https://doi.org/10.1109/ACCESS.2018.2889194

Masoudi, R., & Ghaffari, A. (2016). Software defined networks: A survey. *Journal of Network and Computer Applications*, *67*(October), 1-25. https://doi.org/10.1016/j.jnca.2016.03.016

Mestres, A., Rodriguez-Natal, A., Carner, J., Barlet-Ros, P., Alarcn, E., Sol, M., Munts-Mulero, V., Meyer, D., Barkai, S., Hibbett, M. J., Estrada, G., Ma 'ruf, K., Coras, F., Ermagan, V., Latapie, H., Cassar, C., Evans, J., Maino, F., Walrand, J., … Muntés-Mulero, V. (2017). Knowledge-Defined Networking. *ACM SIGCOMM Computer Communication Review*, *47*(3), 2-10. https://doi.org/10.1145/3138808.3138810

Misra, S., & Saha, N. (2019). Detour: Dynamic Task Offloading in Software-Defined Fog for IoT Applications. *IEEE Journal on Selected Areas in Communications*, *37*(5), 1159-1166. https://doi.org/10.1109/JSAC.2019.2906793

More, T. S. P.-S. and S. from. (n.d.). *Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions)*. https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

Park, J., Hwang, J., & Yeom, K. (2019). NSAF: An Approach for Ensuring Application-Aware Routing Based on Network QoS of Applications in SDN. *Mobile Information Systems*, *2019*. https://doi.org/10.1155/2019/3971598

Phemius, K., Bouet, M., & Leguay, J. (2014). DISCO: Distributed multi-domain SDN controllers. *2014 IEEE Network Operations and Management Symposium (NOMS)*, 1-4. https://doi.org/10.1109/NOMS.2014.6838330

Ray, P. P. (2018). A survey on Internet of Things architectures. In *Journal of King Saud University - Computer and Information Sciences* (Vol. 30, Issue 3, pp. 291–319). King Saud bin Abdulaziz University. https://doi.org/10.1016/j.jksuci.2016.10.003

Saha, N., Bera, S., & Misra, S. (2018). Sway: Traffic-Aware QoS Routing in Software-Defined IoT. *IEEE Transactions on Emerging Topics in Computing*, 1-12. https://doi.org/10.1109/TETC.2018.2847296

Salman, O., Elhajj, I., Chehab, A., & Kayssi, A. (2018). IoT survey: An SDN and fog computing perspective. In *Computer Networks* (Vol. 143, pp. 221-246). Elsevier B.V. https://doi.org/10.1016/j.comnet.2018.07.020

Tayyaba, S. K., Shah, M. A., Khan, O. A., & Ahmed, A. W. (2017). Software defined network (SDN) based internet of things (IoT): A road ahead. *ACM International Conference Proceeding Series, Part F1305*. https://doi.org/10.1145/3102304.3102319

Wang, H., Xu, H., Huang, L., Wang, J., & Yang, X. (2018). Load-balancing routing in software defined networks with multiple controllers. *Computer Networks*, *141*, 82-91. https://doi.org/10.1016/j.comnet.2018.05.012

Wang, Y. C., & You, S. Y. (2018). An Efficient Route Management Framework for Load Balance and Overhead Reduction in SDN-Based Data Center Networks. *IEEE Transactions on Network and Service Management*, *15*(4), 1422-1434. https://doi.org/10.1109/TNSM.2018.2872054

Weiser, M. (1999). The computer for the 21 st century . *ACM SIGMOBILE Mobile Computing and Communications Review*, *3*(3), 3–11. https://doi.org/10.1145/329124.329126

Xia, W., Wen, Y., Foh, C. H., Niyato, D., & Xie, H. (2015). A Survey on Software-Defined Networking. *IEEE Communications Surveys & Tutorials*, *17*(1), 27-51. https://doi.org/10.1109/COMST.2014.2330903

Xu, H., Huang, H., Chen, S., Zhao, G., & Huang, L. (2018). Achieving High Scalability Through Hybrid Switching in Software-Defined Networking. *IEEE/ACM Transactions on Networking*, *26*(1), 618-632. https://doi.org/10.1109/TNET.2018.2789339

Xue, H., Kim, K. T., & Youn, H. Y. (2019). Dynamic load balancing of software-defined networking based on genetic-ant colony optimization. *Sensors (Switzerland)*, *19*(2). https://doi.org/10.3390/s19020311

Yao, H., Yuan, X., Zhang, P., Wang, J., Chunxiao, J., & Guizani, M. (2019). Machine Learning Aided Load Balance Routing Scheme Considering Queue Utilization. *IEEE Transactions on Vehicular Technology*, 1–1. https://doi.org/10.1109/tvt.2019.2921792

Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things Journal*, *1*(1), 22-32. https://doi.org/10.1109/JIOT.2014.2306328

Muhamed Begović
University of Sarajevo, Faculty of Traffic and Communications,
Sarajevo,
Bosnia and Herzegovina
muhamed.begovic@fsk.unsa.ba

Samir Čaušević
University of Sarajevo,
Faculty of Traffic and Communications,
Sarajevo,
Bosnia and Herzegovina
samir.causevic@fsk.unsa.ba

Elma Avdagić-Golub
University of Sarajevo,
Faculty of Traffic and Communications,
Sarajevo,
Bosnia and Herzegovina
elma.avdagic@fsk.unsa.ba