Computers, Materials & Continua
DOI:10.32604/cmc.2021.018625
Article

Tech Science Press

Centralized QoS Routing Model for Delay/Loss Sensitive Flows at the SDN-IoT Infrastructure

Mykola Beshley[1], Natalia Kryvinska[2,*], Halyna Beshley[1], Mykhailo Medvetskyi[1] and Leonard Barolli[3]

[1]Department of Telecommunications, Lviv Polytechnic National University, Lviv, 79013, Ukraine
[2]Department of Information Systems, Faculty of Management, Comenius University in Bratislava, Bratislava, 82005, Slovakia
[3]Department of Information and Communication Engineering, Faculty of Information Engineering, Fukuoka Institute of Technology (FIT), Higashi-Ku, 811-0295, Fukuoka, Japan
[*]Corresponding Author: Natalia Kryvinska. Email: natalia.kryvinska@uniba.sk
Received: 14 March 2021; Accepted: 17 April 2021

Abstract: The rapidly increasing number of Internet of Things (IoT) devices and Quality of Service (QoS) requirements have made the provisioning of network solutions to meet this demand a major research topic. Providing fast and reliable routing paths based on the QoS requirements of IoT devices is very important task for Industry 4.0. The software-defined network is one of the most current interesting research developments, offering an efficient and effective solution for centralized control and network intelligence. A new SDN-IoT paradigm has been proposed to improve network QoS, taking advantage of SDN architecture in IoT networks. At the present time, most publish-subscribe IoT platforms assume the same QoS requirements for all customers. However, in many real-world scenarios of IoT applications, different subscribers may have different E2E delay requirements. Providing reliable differentiated services has become a relevant problem. For this we developed a technique for classifying IoT flows with the individual subscriber recommendation on the importance of certain parameters for particular classes of traffic taken into account. To improve the QoS for mission-critical IoT applications in large-scale SDN-IoT infrastructure, we focused on optimizing routing in the SDN. For this purpose, a centralized routing model based on QoS parameters and IoT priority flow for the SDN was proposed and implemented. We have compared the proposed routing model with the state-of-art deterministic multiconstrained centralized QoS routing model (DMCQR). The developed centralized routing model in comparison with the known DMCQR flow routing achieved better balance of channel resources load due to rational choice of transmission paths for different traffic. And it was possible to reduce up to 3 times the average delay of real time flows service from end to end, for which with the existing DMCQR routing model the permissible delay rates were not met.

Keywords: Internet of things; software defined networking; quality of services; routing; internet of video things

1 Introduction

The trend of the Internet of Things (IoT) is becoming more and more popular now in the Industry 4.0 [1]. Briefly it can be described as follows: an increase in the number of devices that interact not only with users but also with each other [2]. The huge number of devices will enable services from a wide variety of sources such as home appliances, surveillance cameras, monitoring sensors, actuators, displays, vehicles, machines, and so on [3]. The rapid emergence of the IoT takes advantage of the promises of many important new benefits, but at the same time creates some potential problems and issues [4–6]. To support the huge number of connected devices with heterogeneous characteristics, the IoT communication infrastructure needs new architectures and devices that can accommodate the growing IoT traffic with the guarantee of a specific QoS level [7–9].

IoT will allow the development of applications in many different domains, such as home automation, industrial automation, medical aids, traffic management, and many others [10,11]. These applications have a range of QoS requirements, which can be typically categorized as best effort (no QoS), differentiated services (soft QoS) and guaranteed services (hard QoS), especially for a mission-critical IoT applications [12]. The authors [13] considered five important mission-critical IoT applications and characterize them based on several different requirements such as factory automation, process automation, smart grids, intelligent transport systems and professional audio. For example, process automation includes applications for monitoring and diagnostics of industrial elements, and processes such as heating, cooling, mixing, stirring, pumping, etc. Therefore, End-to-End (E2E) delay requirements for such IoT services range from 50 to 100 ms with the available Packet Loss Ratio (PLR) up to 10^{-3}.

The network infrastructure is a critical component of the ecosystem for a mission-critical IoT application. In order to provide strong QoS for IoT applications, it is necessary to provide suitable mechanisms at each layer of the IoT infrastructure, as some factors, such as E2E latency, are very important [14–16]. A delay in any layer can lead to unacceptable QoS for safety critical applications, such as automated driving systems which need constant feedback to maintain control. Such ultimate results are the main reason why strict compliance with QoS requirements is so necessary in the mission-critical IoT. This is also part of why designing for mission-critical IoT platform is so complex. Nowadays, most publish/subscribe IoT platforms suppose that there are equal QoS requirements for all clients. However, in many real-world IoT applications scenarios, different subscribers may have different E2E delay requirements. How to provide reliable differentiated services has become an relevant problem.

In addition, the IoT communication infrastructure has to be energy efficient, cost-effective, flexible and scalable to provide IoT services with different quality of service requirements [17–20]. To develop a more flexible and scalable network, the Software Defined Networking (SDN) paradigm has been proposed in recent years [21–24]. The SDN provides simplified network management by separating the control plane from the data plane [25–27]. Thus, with this programmable and centralized control plane, it is possible to monitor real time network behavior and flexibly manage network traffic, which also makes SDN one of the key technologies in SDN-IoT applications. So, approaches based on the SDN in the context of IoT have recently attracted some attention [28–30]. The SDN-IoT infrastructure is depicted in Fig. 1.

Special attention is paid to the means of routing and distribution of flows under high load conditions to ensure the specified parameters of the quality of service within the network management. In this regard, the development and implementation of SDN solutions require improvements to existing routing protocols and load-balancing mechanisms [31–33]. Analysis of the known routing protocols has a significant disadvantage for they provide the calculation of the shortest path by only one, however composite metrics [34]. Thus, the necessary numerical values of one of the key QoS parameters of a particular flow of a certain client are not guaranteed. On the other hand, these routing protocols generally cannot ensure a universal and satisfactory solution that meets the requirements of heterogeneous large-scale IoT networks [35]. The most common is the flow model of routing with load balancing to minimize the coefficient of maximum channel utilization [36]. The study of this model for SDN showed that load balancing by the criterion of channel utilization ratio does not allow improving the QoS level in all cases. Therefore, it is recommended to change the criterion underlying the optimization of the load balancing process in such a way as to maximize the values of the basic QoS values [37] when addressing the routing and load balancing task in the SDN [38].

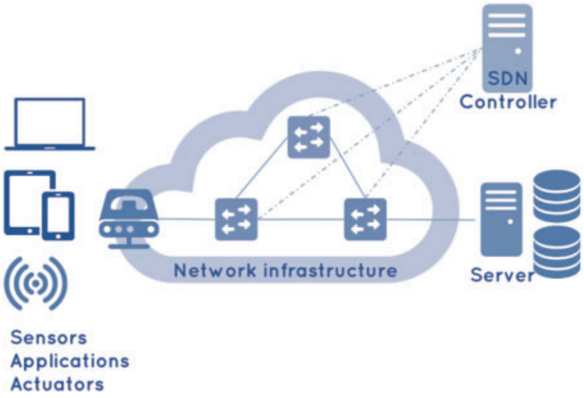

Figure 1: SDN-IoT infrastructure

In order to address the abovementioned challenges, this paper focuses on creating a new cost-effective and low-energy customizable platform based on SDN architecture. It can adapt its configuration to meet the QoS requirements of target IoT applications. Our principal contributions in this paper are summarized as follows:

- We developed a technique for classifying IoT flows with the customers recommendation on the importance of certain parameters for particular classes of traffic taken into account.
- We proposed and realized a centralized routing model based on QoS parameters and IoT priority flow for SDN to enhance QoS for mission-critical IoT applications in a large-scale SDN-IoT infrastructure.
- We have compared the proposed routing model with the state-of-art deterministic multiconstrained centralized QoS routing model (DMCQR)

The paper is organized as follows. Section 2 presents a brief review of the related works. Section 3 introduces the implementation details of our enhanced QoS-aware routing model in the SDN section of the IoT platform and evaluates our model with respect to state of the arts DMCQR. Finally, we conclude in Section 4.

2 Related Work

Routing in SDN based IoT infrastructure has recently become a hot topic of research and has generated so much scientific interest [39–47]. A summary of the related work is shown in Tab. 1.

Table 1: Related work on routing in SDN-IoT infrastructure

Ref.	Description
[39]	Proposed a traffic aware QoS routing model in software defined internet of things (SDIoT) network; greedy approach based on Yen's K's shortest path algorithm to calculate the optimum redirection path in SDN for the QoS requirements of each packet.
[40]	Proposed the application-aware QoS routing algorithm (AQRA) for SDN based IoT networks to ensure multiple QoS requirements of high-priority IoT applications and adapt to the current state of the network for better routing path.
[41]	Developed an SDN infrastructure suitable for the IoT environment, in which the SDN controller uses network computing and genetic algorithm to optimize IoT applications.
[42]	Proposed a deterministic network model using network calculus to compute the optimal paths for each flow through the priority queues in SDN.
[43]	Developed a stochastic model for the E2E delay in SDN switches based on measurements made in Internet-scale experiments performed Mininet.
[44]	Proposed a deep reinforcement learning based QoS-aware secure routing protocol (DQSP) in the SDN-IoT infrastructure. The method can extract knowledge from the requirements of traffic history, interacting with the underlying network environment, and optimize routing policies.
[45]	Proposed the deterministic multiconstrained centralized QoS routing (DMCQR) algorithm based on network calculus in SDN. Experiment results show that the proposed DMCQR algorithm had better performance in terms of effective bandwidth utilization rate, packet loss rate, path load rate and end-to-end delay compared with the performance of the Dijsktra algorithm.
[46]	proposed a deep learning-based intelligent channel assignment algorithm. It can intelligently avoid potential congestion and quickly allocate the appropriate channel in the SDN-IoT infrastructure.
[47]	proposed an approach that can control SDN congestion by dynamically dividing traffic by analyzing the statistics collected by each switch in the network. The traffic partition is done in such a way that when a flow is redirected to another path, the SDN controller checks in advance to see if this action leads to traffic jams on the new path.

To the best of our knowledge, the studies on QoS-aware routing for mission-critical IoT applications in the SDN based IoT platform are still very limited in literature. Most of the existing scientific papers on routing in SDN-IoT still have a number of significant shortcomings, are only theoretical and difficult to implement in practices. Also, when analyzing the scientific papers, we have not found approaches to provide QoS for the mission-critical IoT applications in conditions of high network load with the presence of subscribers to the same IoT applications of topics with different QoS requirements.

3 Enhanced QoS-Aware Routing Model in SDN

3.1 *Centralized Routing Model Based on QoS Parameters and IoT Priority Flow for SDN*

The paper proposes an advanced model of routing that allows balancing the load on the criteria of minimum/maximum network channel load and service quality for each flow in SDN. There is developed a technique for classifying IoT flows with the recommendation on the importance of certain parameters for particular classes of traffic taken into account. Tab. 2 show the requirements for QoS of existing services operating within the Internet of Things [48] (IoT automated emergency call—*A*, Real time Monitoring temperature—*B*, Real time Internet of Video Things (IoVT)—*C*, IoT-Alert (Photo/text/Email)—*D*, Un Real time IoVT (Video on Demand)—*E*, Un Real time IoVT (720p60)—*F*, Un Real time IoVT (1080p60)—*G*). The primary goal of QoS is to provide priority, including dedicated throughput—*C*, delay—*T*, jitter—*J*, packet loss—*P* (required by some real-time and non real-time traffic). We also introduce two additional criteria for the classification of IoT flows the first criterion is the sensitivity to the mixing of packet—P_m and the second criterion is the priority of the IoT subscribers–C_p.

Table 2: Requirements for QoS of certain IoT applications

Flow type	QoS parameters					
	P, %	T, ms	J, ms	C, kbit/s	$\mathrm{P_m}$	$\mathrm{C_p}$
A	<0,01	150	10	64	10^{-2}	2–256
B	<2	100	20	2048	10^{-2}	256–512
C	<1	100	50	4096	10^{-2}	512–768
D	<0,1	100	100	2048	10^{-6}	1280–1536
E	<1	400	500	256–10000	10^{-4}	768–1024
F	<0,1	400	30	2048	10^{-4}	1024–1280
G	0,01	50	50	64	10^{-5}	1

To calculate the relative priority of a flow for a particular path, we defined the relative coefficients for each flow. The relative coefficients are the ratio of the minimum values of the service quality parameters to the current values obtained from the network monitoring.

The relative coefficient of packet loss—*p*, packet delay—*t*, packet jitter—*j*, throughput—*c*, sensitivity to the mixing of packets—p_m and priority of the subscribers—c_p are defined as shown in Eq. (1):

$$p = \frac{P_{\min}}{P}, \quad t = \frac{T_{\min}}{T}, \quad j = \frac{J_{\min}}{J},$$
$$c = \frac{C}{C_{\max}}, \quad p_m = \frac{Pm_{\min}}{P_m}, \quad c_p = \frac{C_p}{Cp_{\max}}. \tag{1}$$

The results of the calculations are shown in Tab. 3.

Table 3: The matrix of relative coefficients

Flow type	QoS parameters				
	p, %	t, ms	j, ms	c, kbit/s	p_m
A	0,1	0,67	1	0,0063	10^{-5}
B	0,012	1	0,5	0,2	10^{-5}
C	0,067	0,1	0,2	0,4	10^{-2}
D	0,1	0,1	0,01	0,2	10^{-2}
E	0,1	0,25	0,02	0,025	10^{-5}
F	0,2	0,2	0,33	1	10^{-5}
G	1	1	0,01	0,0063	1

Tab. 4 is filled with numbers 1–3, which correspond to low, medium, and high significance of the QoS parameters, respectively [49].

Table 4: The matrix of importance of QoS parameters

Flow type	QoS parameters				
	P_r, %	T_r, ms	J_r, ms	C_r, kbit/s	P_{mr}
A	2	3	3	1	3
B	2	3	3	2	3
C	3	2	2	3	3
D	3	1	1	1	2
E	2	2	1	1	2
F	2	2	2	3	2
G	3	2	1	1	3

These parameters can be specified by the operator. Moreover, each subscriber is assigned a priority for each type of traffic. If the priority is not explicitly specified in the service agreement, then the IoT subscribers are assigned the lowest priority of all possible by default.

Thus, the relative priority for each category of IoT applications (services) is determined by Eq. (2).

$$P_{IoT_k} = \frac{\sum_{m=1}^{4} X_{km} Y_{km}}{\sum_{k=1}^{7} \sum_{m=1}^{4} X_{km} Y_{km}} \tag{2}$$

where P_{IoT_k}—the relative priority of the i-th IoT service;

k—IoT service type number;

m—number service quality parameter;

X_{km}—relative priority of parameter m for IoT service k;

Y_{km}—the importance of parameter m for IoT service k.

The final result of this formula is a fraction within the range from zero to one. The greater value, the higher priority of an IoT flow. The formula can be applied to any number of flows and various quality requirements for the IoT service.

Considering above technique of prioritizing IoT flows, we divide them into three categories. The first type flows of should be delivered optimally with respect to the criterion of cost, which considers the following quality of service parameters: delay and jitter. Such flows are very sensitive to delay and jitter in the mission-critical IoT applications, so multi routing is forbidden for them, that is the whole flow can be transmitted one path only. The second type flows are less sensitive to the time parameters of the QoS, and therefore it is allowed to balance such flows. However, when balancing, there is a limit on the minimum sub-flow size. This constrain is essential because the second kind flows are usually TCP flows. They are sensitive to loss and mixing of packets, and as the number of sub-flows increases their size decreases, so the probability of loss and mixing due to multi routing increases. In this case, the delivery of the second types flows is guaranteed with QoS parameters within acceptable limits. The third type flows can be delivered by any routing with no guarantee of quality of service. Therefore, such flows are used to load low-load paths and to balance the load distribution between channels in a network.

When the network is operating in normal mode, without overload or bottlenecks, the monitoring system polls the network devices at a consistently long interval. In particular, the monitoring system polls network interface loads, switch flow table loads, and switch central input buffer loads that identify interface overloads. The monitoring system polls nodes with a certain interval of time and monitors the time parameters of the flows of the first category. In case of average growth in interface load, transmission delay and jitter, the system switches to the state of close monitoring of the specified interface and increases the intensity of delay measurement. In addition, when the interface load level reaches 0.9, the system starts polling the device buffer loads.

The developed monitoring system facilitates the implementation of this approach through adaptive monitoring of the used resources of channels and devices. The search for the optimum path is based on the mathematical theory used to solve the two well-known problems: the Multi-Commodity Flow Problem and the Constrained Shortest Path Problem. The main goal is to find the optimum set of paths in a network for all flows with a minimum total cost. The main constrain is that the total flow rate through a channel cannot exceed channel throughput.

Our model combines both of the abovementioned problems and allows finding the shortest path for each flow, subject to the given constraints. Suppose we have a network of nodes where each channel (link) is characterized by a delay, a loss and a channel throughput. In addition, each path is characterized by the cost calculated as a weighted sum of delays and packet losses. The model assumes that for each type of service/traffic there is a set of flows that can have completely different input and output nodes in the network. The goal is to find a set of optimum network paths for each flow with the minimum cost subjected to certain constraints, in particular such as the maximum delay, packet loss, and available link throughput.

Suppose a network is represented by the graph $G = (V, E)$, where V is the set of nodes, and E is the set of arcs between each pair of nodes. The arcs, that is, the links, are characterized by available channel throughput, delays, packet loss, and the cost of transfer per unit flow. As a result, the cost can be calculated by following Eq. (3):

$$W_{ij} = \theta \cdot d_{ij} + \xi \cdot p_{ij}, \quad \forall (i, j) \in E \tag{3}$$

where θ and ξ—the weighs for delay and loss, respectively.

This formula allows adjusting the cost of the path with respect to either delay or loss of packets for a particular flow. This allows adjusting these settings to meet the quality requirements for each flow. For example, multimedia traffic has restrictions on end-to-end delays, so one can set $\theta = 1$ and $\xi = 0$ to take into account the delay only. Each individual flow k is identified by a relative priority P_{IoT_k}, whose value lies within the range from zero to one. The set of different IoT traffic flows to be transferred over the network is denoted by K. Each flow is characterized by five parameters:

G_{x_k}—the node (host) where the k-th flow enters the network;

G_{y_k}—the node (host) where the k-th flow leaves the network;

f_k—the rate of the k-th flow that must be delivered from the input node to the output one;

$P^k_{\max} \geq 0$—the maximum allowed packet loss for the k-th flow;

$D^k_{\max} \geq 0$—the maximum allowed delay for the k-th flow.

The goal of optimization is to path all flows in the network through the paths with the minimum cost.

Sets:

nodes: $n \in V$;

arcs: $(i, j) \in E$;

channels (links): $(i, j) \in E \cup (j, i) \in E$;

Variables:

$0 \leq x^k_{ij} \leq f_k$—the data volume of the k-th flow passing through the link (i, j).

Parameters:

$b_{ij} \geq 0$—available link throughput(i, j);

$0 \leq \rho \leq 1$—the maximum channel load (link utilization) in the network;

$0 \leq r \leq 1$—relative cost priority over the maximum channel load ρ;

$C_{\max} \geq 0$—the maximum link throughput;

$c_{ij} \geq 0$—channel cost (i,j) calculated as $\theta \cdot d_{ij} + \xi \cdot p_{ij}$;

θ—weight for delay;

ξ—weight for packet loss;

$G_{x_k} \in V$—source of the k-th flow;

$G_{y_k} \in V$—destination of the k-th flow;

f_k—the rate of the k-th flow;

$p_{ij} \geq 0$—actual packet loss in the link (i, j);

$d_{ij} \geq 0$—actual link delay (i, j);

$B^k \geq 0$—the throughput required for the k-th flow.

The routing parameters are controlled based on the relative priority of a flow, which is calculated according to the technique presented above in this section. All values of the relative priority P_{IoT_k} of a flow are within the range from 0 to 1. Let us introduce the parameters $TC_{High\mathrm{P}riority} \in TC$ and $TC_{BestEffort} \in TC$, with the condition $TC_{High\mathrm{P}riority} < TC_{BestEffort}$. The parameter $TC_{High\text{Priority}}$ contains the relative priority which the relative priorities P_{IoT_k} of all other flows are compared with. If the relative priority of a certain flow is greater than $TC_{High\text{Priority}}$, then the flow belongs to the first category and has high priority. This means it cannot be split into sub-flows for load balancing. If the relative priority of the flow is less than $TC_{High\text{Priority}}$, however larger than $TC_{BestEffort}$, then the flow belongs to the second category and has an average priority. This means the flow can be split into sub-flows and choose for them the paths with the quality of service no lower than the shortest available path. Such flows have a limitation that determines the minimum volume of a sub-flow. This allows controlling the levels of redistribution of sub-flows in the network to avoid packet mixing for the TCP flows. The quality of service is not guaranteed for all the flows with the priority lower than $TC_{BestEffort}$. They can be split into sub-flows of arbitrary volume. For these sub-flows, the paths with minimum channel load are selected.

The objective function (4) minimizes cost with respect to the parameters of the quality of service in links, as well as with respect to the coefficient of the minimum/maximum load of links, which depends on the type of relative priority of a flow.

$$F(x) = \rho + r \sum_{(i,j)\in E} \sum_{k\in K} w_{ij} x_{ij}^{k} \rightarrow \min, \tag{4}$$

Constrain in Eq. (5) corresponds to the conservation of the flow

$$\sum_{(i,j)\in E} x_{ij}^{k} - \sum_{(i,j)\in E} x_{ji}^{k} = \begin{cases} f_k, & i = s_k, \\ -f_k, & i = t_k, \ \forall i \in V, \ \forall k \in K\,, \\ 0, & i \neq s_k, \ t_k \end{cases} \tag{5}$$

Constrain in Eq. (6) takes into account the allowed level of flow distribution with respect to its relative priority:

$$x_{ij}^{k} \geq \begin{cases} B_H, & P_{IoT_k} > TC_{High\text{Priority}}, \ \forall\,(i,j) \in E, \\ B_L, & TC_{BestEffort} < P_{IoT_k} < TC_{High\text{Priority}}, \ \forall\,(i,j) \in E. \\ 0, & P_{IoT_k} < TC_{BestEffort}, \ \forall\,(i,j) \in E, \end{cases} \tag{6}$$

where B_L and B_H are the minimum link throughput of sub-flows when balancing the main flow of the second and third category, respectively.

The following two constraints in Eqs. (7) and (8) consider the maximum allowed E2E packet loss and delay, which should not exceed critical values $D_{\max}^{k}$, $P_{\max}^{k}$, for a flow with relative priority k. In the case of routing of the third category, these parameters are given the maximum possible value:

$$\sum_{(i,j)\in E} p_{ij} \leq \begin{cases} P_{\max}^{k}, & x_{ij}^{k} > 0, \ \forall k \in K; \\ 0, & x_{ij}^{k} = 0, \ \forall k \in K; \end{cases} \tag{7}$$

$$\sum_{(i,j)\in E} d_{ij} \leq \begin{cases} D_{\max}^{k}, & x_{ij}^{k} > 0, \ \forall k \in K; \\ 0, & x_{ij}^{k} = 0, \ \forall k \in K; \end{cases} \tag{8}$$

Inequality in Eq. (9) imposes a constrain on the available throughput of each link, taking into account all the flows k through these links. Moreover, this constrain is lower in case of routing flows of the second and the third categories, which causes their transmission in ways with low loading efficiency and with poor quality of service:

$$\sum_{k \in K} x_{ij}^{k} \leq \begin{cases} \rho \cdot C_{\max}, & \forall\, (i,j) \in E,\ k < TC_{High\mathrm{P}riority} \\ b_{ij}, & \forall\, (i,j) \in E \end{cases} \tag{9}$$

The last constrain in Eq. (10) establishes the range of the variable and ensures that the variable acquires values within the flow rate:

$$0 \leq x_{ij}^{k} \leq f_k \quad \forall\, (i,j) \in E, \quad \forall k \in K, \tag{10}$$

The proposed mathematical model allows setting the maximum allowed packet loss and delay for the first category flows. These values are essential for choosing the best path for the QoS criterion. By using weights, one can match the weight of each flow based on its requirements. On the one hand, one can set the maximum allowed loss and delay and, as a result of solving a linear programming problem, obtain a path with the minimum transfer cost that meets the requirements of the QoS parameters. On the other hand, one can change the maximum allowed packet loss and delay, then recalculate the optimum paths to find the path that ensures the required quality.

Let us analyze the structure of the communication network that consists of the N network devices. Numeric values of the metric $W_{(i,j)}$ can be presented as adjacency matrix W:

$$W = \begin{pmatrix} w_{1,1} & \cdots & w_{1,N} \\ \vdots & \ddots & \vdots \\ w_{N,1} & \cdots & w_{N,N} \end{pmatrix}, \tag{11}$$

It is worth mentioning that the matrix of parameters W is not symmetric ($W_{(i,j)} \neq W_{(j,i)}$). Minimization of target function metric $W_{(i,j)}$ that can be under several constraints or limits is called multi criteria optimization. The main difficulty encountered in solving problems of this type is the ambiguity of the optimal solution at the point where one of the criteria reaches its maximum, while the other may be far from not only the maximum, but even from any arbitrarily admissible value. Finding a metric by the "ideal point" method requires that all values have the same dimensionality. To do this, let us model the channel characteristics as follows:

- for the number of lost packets

$$P_{i,j} = min, \quad P_{i,j} = \left(1 - \frac{N_{i,j}}{M_{i,j}}\right), \tag{12}$$

where $N_{i,j}$ is a number of received packets, $M_{i,j}$ is a number of sent packets and $M_{i,j} > 0$

- for the flow delay

$$D_{i,j} = min, \quad D_{i,j} = \left(1 - \frac{d_{\min}}{d_{i,j}}\right), \tag{13}$$

Thus, the metric based on the two parameters can be calculated as:

$$W_{i,j} = \sqrt[2]{\theta (D_{i,j})^2 + \xi (P_{i,j})^2}. \tag{14}$$

where θ and $\xi$are the weight coefficients that change their values in the range between 0 and 1, and their sum must be equal to 1:

Changing the values of weight coefficients in the metric $W_{(i,j)}$, we create an apparatus for controlling the value of a particular parameter when evaluating the channel characteristic from node i to node j.

In terms of mathematics, the "ideal point" is the one that has coordinates that correspond to $D_{\min}, P_{\min}$, under the fixed weight coefficients θ and ξ.

3.2 Experimental Results and Analyses

This part describes the implementation of our novel centralized routing model based on the multiple QoS requirements and IoT priority flow approach described in Section 3.1. The mathematical routing model is implemented as an external module on top of the Floodlight Open SDN Controller. This module is implemented in Java Programming Language. To solve the problem of finding the optimum path in the SDN, we also used AMPL (A Mathematical Programming Language) and the CPLEX optimizer [50].

For our experimental tests, we developed a network topology in the Mininet emulator. This topology composed of 7 Open vSwitches, 1 the Floodlight SDN controller and 6 IoT traffic generators (G1–G6). Link throughput between all nodes for all ports is set equal to 100 Mbps. The experimental SDN testbed is depicted in Fig. 2.

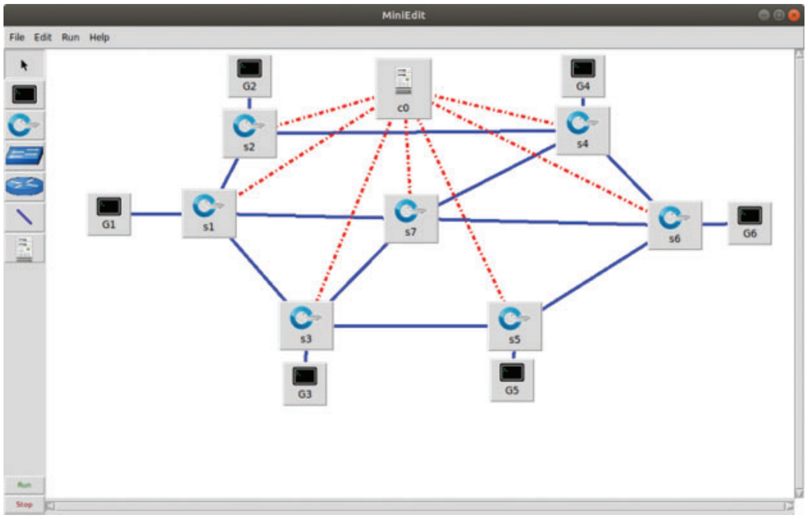

Figure 2: SDN topology considered for experiment

For the first time we tested the average packet delay *vs.* link utilization of the developed SDN switch ports (Fig. 3). For this we also used a traffic generator, iperf, to generate traffic and transfer from host to client. The load was generated from 10 to 100 Mbp in 10 Mbps steps.

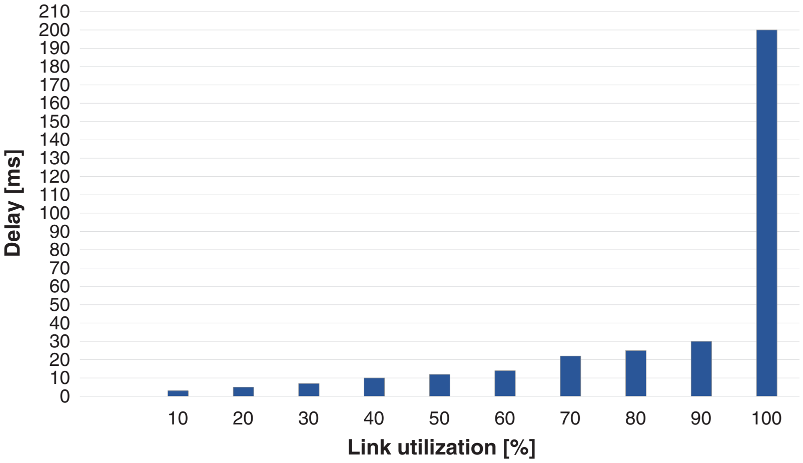

Figure 3: Average packet delay *vs*. link utilization in SDN

During the experiment, IoT traffic was generated in the network using a multi-service traffic generation system proposed in paper [51]. The matrix of requirements C_{ij} for link throughput in Mbps between nodes is given below.

$$Cij = \begin{array}{ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 12,5 & 16,5 & 5 & 9 & 7 \\ 2 & 11 & 0 & 12,5 & 20 & 14 & 13 \\ 3 & 5 & 16,5 & 0 & 21 & 12,6 & 5 \\ 4 & 10 & 14,5 & 12 & 0 & 13 & 11 \\ 5 & 8 & 18 & 6 & 8 & 0 & 15 \\ 6 & 12 & 8 & 9 & 15 & 5 & 0 \end{array} \qquad (15)$$

Node No 7 is intermediate (it represents the aggregation level), so no IoT devices (servers) are connected to it. According to the matrix of requirements, the list of flows for all traffic categories was generated. A set of subscribers and the IoT services they use are generated for the channel (link) throughput of 100 Mbps. The list of flows for 9 subscribers is given in Tab. 5 (C is the link throughput, Mbps; Cp is the priority of IoT subscribers, P_{IoT_k} is the relative priority of the i-th IoT service).

Each subscriber uses a specific set of IoT services. In case an empty cell at the intersection of a column and a row, this service should be considered as the one with no guarantee of quality of service if a subscriber uses it. All other IoT services have throughput requirements and service parameters for a particular user secured by a service agreement between a subscriber and the network operator. As can be seen from Tab. 4, each subscriber is assigned a priority within the appropriate priority range for a particular type of service. This generation is carried out for each pair of servers used to generate subscribers' load.

The next step was to calculate the relative priorities of IoT flows, taking into account the classification of service categories proposed in Section 3.1 of the paper.

For the first category of services (mission-critical IoT applications), which determines real-time flows that are extremely sensitive to fluctuations of time belongs the following ones: IoT automated emergency call (A), monitoring temperature (B), real time IoVT (C).

The results of the calculation of the relative priorities for all the subscribers according to the technique improved in the 3.1 section of this work, are given in Tab. 5.

Table 5: Relative flow priorities based on subscriber's priority and its IoT QoS requirements

IoT services	Parameters	Subscribers								
		1	2	3	4	5	6	7	8	9
IoT automated emergency call (A)	*C*	0,17	0,19	0,16	0,19	0,18	0,18	–	–	–
	Cp	5	10	15	20	25	30	–	–	–
	P_{IoT_k}	0,25	0,22	0,22	0,21	0,21	0,21	–	–	–
Real time Monitoring temperature (B) (Device remote controlling)	*C*	1,25	1,01	1,17	1,25	1,17	1,27	1,11	1,09	1,44
	Cp	260	265	270	275	280	285	290	295	300
	P_{IoT_k}	0,188	0,187	0,186	0,185	0,184	0,183	0,182	0,181	0,18
Real time IoVT (C)	*C*	1,91	2,09	2,13	2,21	2,15	2,17	1,95	2,33	2,42
	Cp	515	520	525	530	535	540	545	550	555
	P_{IoT_k}	0,148	0,147	0,146	0,145	0,144	0,143	0,142	0,141	0,42
IoT-Alert (Photo/text/Email) (D)	*C*	1,59	1,60	1,54	1,80	1,68	1,53	1,90	1,46	1,81
	Cp	770	775	780	785	790	795	800	805	810
	P_{IoT_k}	0,128	0,127	0,126	0,125	0,124	0,123	0,122	0,121	0,12
VoD un real time IoVT (E)	*C*	2,06	2,22	1,71	1,84	2,15	2,13	1,97	2,16	2,14
	P	1025	1030	1035	1040	1045	1050	1055	1060	1065
	P_{IoT_k}	0,108	0,107	0,106	0,105	0,104	0,103	0,102	0,101	0,1
un real time iot video (720p60) (F)	*C*	5,56	5,57	3,99	5,50	5,72	5,72	5,50	–	–
	Cp	1280	1285	1290	1295	1300	1305	1310	–	–
	P_{IoT_k}	0,088	0,087	0,086	0,085	0,084	0,083	0,082	–	–
Un real time iot video (1080p60) (G)	*C*	9,96	9,90	9,48	10,63	10,06	10,49	8,30	12,98	9,82
	Cp	1540	1545	1550	1555	1560	1565	1570	1575	1580
	P_{IoT_k}	0,048	0,047	0,046	0,045	0,044	0,043	0,042	0,041	0,04

We have compared the proposed routing model with the deterministic multiconstrained centralized QoS routing model (DMCQR) presented in work [45].

To verify the effectiveness of the proposed solutions for mission-critical IoT applications at high network loads was used real time IoVT (Internet of Video Things) service, broadcasting video from the server G6. The network is filled with IoT traffic in accordance with the requirements given in the above matrix generated by the IoT multiservice traffic generators G1 to G6 connected to 6 OvS switch.

As a result of the functioning of DMCQR, the network was filled with background multiservice traffic generated by IoT devices. All paths for all requirements are given in Tab. 6.

Table 6: Paths of background IoT traffic according to the DMCQR

Flow	Path	Throughput requirements, Mbps	Flow	Path	Throughput requirements, Mbps
G1 → G2	1-2	12,5	G4 → G1	4-7-1	10
G1 → G3	1-3	16,5	G4 → G2	4-2	14,5
G1 → G4	1-7-4	5	G4 → G3	4-7-3	12
G1 → G5	1-7-6-5	9	**G4 → G5**	**4-6-5**	**13**
G1 → G6	1-7-6	7	**G4 → G6**	**4-6**	**11**
G2 → G1	2-1	11	G5 → G1	5-3-1	8
G2 → G3	2-1-3	12,5	**G5 → G2**	**5-6-4-2**	**18**
G2 → G4	2-4	20	G5 → G3	5-3	6
G2 → G5	**2-4-6-5**	**14**	**G5 → G4**	**5-6-4**	**8**
G2 → G6	2-1-7-6	13	G5 → G6	5-6	15
G3 → G1	3-1	5	G6 → G1	6-7-1	12
G3 → G2	3-1-2	16,5	**G6 → G2**	**6-4-2**	**12**
G3 → G4	3-7-4	21	G6 → G3	6-7-3	9
G3 → G5	3-5	12,6	**G6 → G4**	**6-4**	**15**
G3 → G6	3-7-6	5	G6 → G5	6-5	5

The next step was to generate load with only the real time IoVT service from the G6 IoT traffic generator. As a result of operation of the DMCQR, the paths for IoVT flows are selected as shown in Tab. 7.

Table 7: Paths of real time IoVT according to the DMCQR

Flow	Throughput requirements, Mbps	Path
G6 → G1	19.36	6-7-1
G6 → G2	17.45	6-7-1-2
G6 → G3	9.13	6-5-3
G6 → G4	**9.05**	**6-4**
G6 → G5	1.91	6-5

The network awareness module can use OpenFlow to send HTTP requests in order to obtain real-time link throughput usage. The histogram of link utilization was obtained when using the DMCQR and is depicted in Fig. 4.

As a result of routing with the DMCQR model, unbalanced loading of links occurred in the network. Particular attention should be focused on link 4-6 which has significant delays and a high probability of packet loss due to insufficient link throughput. The links 5–6 and 6–7 are in a state near overload with a high probability of delays for the real time IoVT flow. In addition, there are no alternative transmission paths with sufficient throughput for a 9 Mbps flow from G6 to G4 hosts. Therefore, according to the DMCQR approach, the optimal path for an IoVT flows of 9.05 Mbps from G6 to G4 on the basis of QoS criterion is path 6–4.

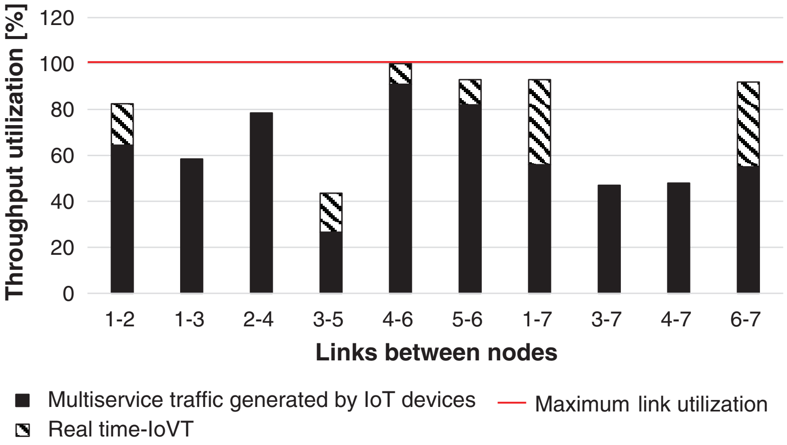

Figure 4: Links utilization in SDN testbed according to the DMCQR model

We have a detailed analysis of the IoT flows leading to overload of link 4–6 according to the DMCQR model. The total load from the background multi-service IoT traffic according to Tabs. 5 and 6 is calculated as follows.

$$f_{(G_2 \to G_5)} = F_3 + B_2 + F_4 + D_2 + D_7 = 3.99 + 1.01 + 5.5 + 1.6 + 1.9 = 14 \;\; [Mbps]$$
$$f_{(G_4 \to G_5)} = F_2 + C_5 + D_5 + A_2 + D_9 + D_2 = 5.57 + 2.15 + 1.68 + 0.19 + 1.81 + 1.6 = 13 \;\; [Mbps]$$
$$f_{(G_4 \to G_6)} = G_7 + A_2 + A_3 + C_6 + A_6 = 8.3 + 0.19 + 0.16 + 2.17 + 0.18 = 11 \;\; [Mbps]$$
$$f_{(G_5 \to G_2)} = G_8 + C_9 + B_8 + B_3 + A_5 = 12.98 + 2.42 + 1.09 + 0.16 + 1.17 + 0.18 = 18 \;\; [Mbps]$$
$$f_{(G_5 \to G_4)} = F_7 + A_3 + B_8 + B_4 = 5.5 + 0.16 + 1.09 + 1.25 = 8 \;\; [Mbps]$$
$$f_{(G_6 \to G_2)} = G_6 + B_3 + A_6 + A_3 = 10.49 + 1.17 + 0.18 + 0.16 = 12 \;\; [Mbps]$$
$$f_{(G_6 \to G_4)} = G_4 + F_3 + A_2 + A_4 = 10.63 + 3.99 + 0.19 + 0.19 = 15 \;\; [Mbps]$$

The load in link 4-6 generated by background IoT traffic and the load in channel 4–6 generated by background IoT traffic and additional real-time IoVT traffic are depicted in the Figs. 5 and 6 respectively.

The total load from the real time IoVT service generated by IoT traffic generator G6 according to the Tabs. 4 and 6 is calculated as follows.

$$f_{(G_6 \to G_1)} = C_1 + C_2 + C_3 + C_4 + C_5 + C_6 + C_7 + C_8$$
$$= 1.91 + 2.09 + 2.13 + 2.21 + 2.15 + 2.17 + 1.95 + 2.33 + 2.42 = 19.36 \;\; [Mbps]$$
$$f_{(G_6 \to G_2)} = C_2 + C_3 + C_4 + C_5 + C_6 + C_7 + C_8$$
$$= 2.09 + 2.13 + 2.21 + 2.15 + 2.17 + 1.95 + 2.33 + 2.42 = 17.45 \;\; [Mbps]$$
$$f_{(G_6 \to G_3)} = C_4 + C_6 + C_8 + C_9 = 2.21 + 2.17 + 2.33 + 2.42 = 9.13 \;\; [Mbps]$$
$$f_{(G_6 \to G_4)} = C_3 + C_6 + C_7 + C_8 = 2.13 + 2.17 + 2.33 + 2.42 = 9.05 \;\; [Mbps]$$
$$f_{(G_6 \to G_5)} = C_1 = 1.91 \;\; [Mbps]$$

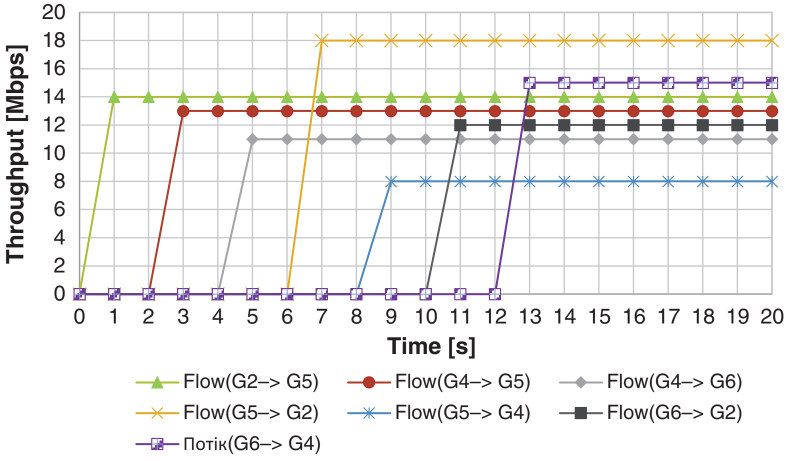

Figure 5: The load in link 4–6 generated by background IoT traffic

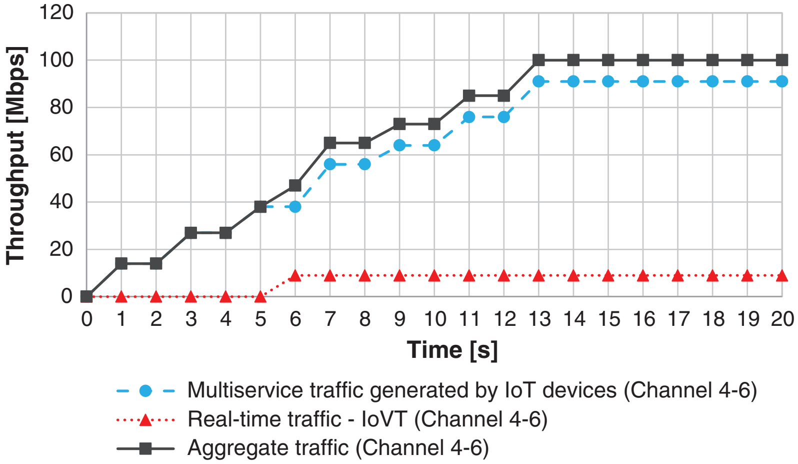

Figure 6: Load in channel 4–6 generated by background IoT traffic and additional real-time IoVT traffic

According to the analysis we can see that the load in link 4–6 is created by multi-service IoT traffic. This multiservice traffic also includes the real time IoT flows (IoT automated emergency call (A), monitoring temperature (B), real time IoVT (C)) for which it is necessary to provide the allowable E2E delay. The E2E delay of IoT flows passing through a link can be defined by Eq. (14).

$$D_{E2E} = \sum_{(i,j)\in E} d_{ij}. \tag{16}$$

We estimated the E2E delay for background multi-service IoT traffic passing through overloaded link 4-6 as follows.

$$D_{E2E(G_2\rightarrow G_5)} = d_{24} + d_{46} + d_{65} = 24.5 + 200 + 35.6 = 260.1 \quad [ms]$$

$D_{E2E(G_4\to G_5)} = d_{46} + d_{65} = 200 + 35.6 = 235.6 \ [ms]$

$D_{E2E(G_4\to G_6)} = d_{46} = 200 \ [ms]$

$D_{E2E(G_5\to G_2)} = d_{56} + d_{64} + d_{42} = 35.6 + 200 + 24.5 = 260.1 \ [ms]$

$D_{E2E(G_5\to G_4)} = d_{56} + d_{64} = 35.6 + 200 = 235.6 \ [ms]$

$D_{E2E(G_6\to G_2)} = d_{64} + d_{42} = 200 + 24.5 = 224.5 \ [ms]$

$D_{E2E(G_6\to G_4)} = d_{64} = 200 \ [ms]$

And we estimated the E2E delay for real time IoVT flows generate from G6 as follows.

$D_{E2E(G_6\to G_1)} = d_{67} + d_{71} = 34.4 + 35.6 = 70 \ [ms]$

$D_{E2E(G_6\to G_2)} = d_{67} + d_{71} + d_{12} = 34.4 + 35.6 + 27 = 97 \ [ms]$

$D_{E2E(G_6\to G_3)} = d_{65} + d_{53} = 35.6 + 11.2 = 46.8 \ [ms]$

$D_{E2E(G_6\to G_4)} = d_{64} = 200 \ [ms]$

$D_{E2E(G_6\to G_5)} = d_{65} = 35.6 \ [ms]$

According to the proposed routing model to prevent overload on channel 4-6, we proposed to use path 4-7-3-5 for non real-time IoT flows $f_{(G_4\to G_5)_{realtime}} = F_2 + D_5 + D_9 + D_2 = 5.57 + 1.68 + 1.81 + 1.6 = 10.66 \ [Mbps]$ and to use path 4-6-5 for real-time flows $f_{(G_4\to G_5)_{unrealtime}} = C_5 + A_2 = 2.15 + 0.19 = 2.34 \ [Mbps]$ generated from host G4 to G5.

The histogram of link utilization was obtained when using the proposed routing model and is depicted in Fig. 7.

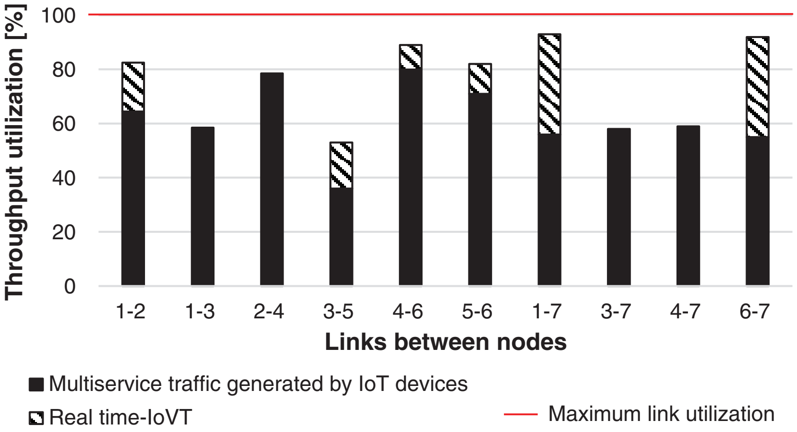

Figure 7: Links utilization in SDN testbed according to the proposed routing model

We estimated the E2E delay of IoT flows passing through overloaded link 4-6 according to the proposed routing model.

The E2E delay for background multi-service IoT traffic is as follows.

$D_{E2E(G_2\to G_5)} = d_{24} + d_{46} + d_{65} = 24.5 + 29.5 + 26 = 80 \ [ms]$

$D_{E2E(G_4\to G_5)} = d_{46} + d_{65} = 29.5 + 26 = 55.5 \ [ms]$

$$D_{E2E(G_4\to G_6)} = d_{46} = 29.5 \ \ [ms]$$

$$D_{E2E(G_5\to G_2)} = d_{56} + d_{64} + d_{42} = 26 + 29.5 + 24.5 = 80 \ \ [ms]$$

$$D_{E2E(G_5\to G_4)} = d_{56} + d_{64} = 26 + 29.5 = 55.5 \ \ [ms]$$

$$D_{E2E(G_6\to G_2)} = d_{64} + d_{42} = 29.5 + 24.5 = 54 \ \ [ms]$$

$$D_{E2E(G_6\to G_4)} = d_{64} = 29.5 \ \ [ms]$$

The E2E delay for real time IoVT flows generate from G6 is as follows.

$$D_{E2E(G_6\to G_1)} = d_{67} + d_{71} = 34.4 + 35.6 = 70 \ \ [ms]$$

$$D_{E2E(G_6\to G_2)} = d_{67} + d_{71} + d_{12} = 34.4 + 35.6 + 27 = 97 \ \ [ms]$$

$$D_{E2E(G_6\to G_3)} = d_{65} + d_{53} = 26 + 13 = 39 \ \ [ms]$$

$$D_{E2E(G_6\to G_4)} = d_{64} = 29.5 \ \ [ms]$$

$$D_{E2E(G_6\to G_5)} = d_{65} = 26 \ \ [ms]$$

Comparison of average E2E delay of the DMCQR model with our proposed routing model is depicted in Fig. 8.

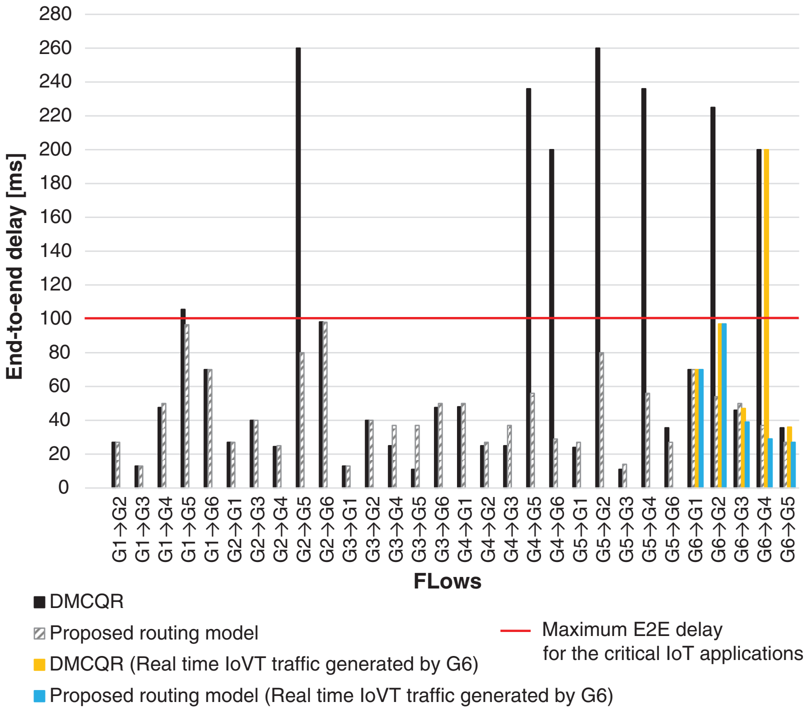

Figure 8: Comparison of average end-to-end delay of the DMCQR model with our proposed routing model

In general, the above results show that the routing model proposed in this paper provides an acceptable E2E QoS guarantee for all mission-critical IoT applications, in contrast to the existing DMCQR routing model that has been developed by authors in paper [41].

4 Conclusions

A software-defined networking paradigm, with simpler hardware and flexible management and monitoring, is the true solution to allow the Internet to converge with the IoT. SDN belongs to the programmable network domain. It separates the data and control planes using simpler hardware devices and centralized software control of the entire network. Our research focuses on guaranteeing QoS applications over an SDN network in the context of the IoT.

To solve this problem, we proposed the centralized routing model based on QoS parameters and IoT subscriber flow priorities for SDN. It enhances QoS for mission-critical IoT applications in large-scale SDN-IoT infrastructure. Experimental results showed that in contrast to the existing DMCQR routing scheme, the proposed model can provide QoS to both delay- and loss-sensitive IoT flows.

The developed centralized routing model in comparison with the known DMCQR flow routing achieved better balance of channel resources load due to rational choice of transmission paths for different traffic. And it reduces up to 3 times the average delay of real time flows from end to end, for which existing DMCQR routing model with the permissible delay rates were not met.

A limitation of the proposed idea is that in practice it is necessary to develop a tool according to which users can report to the controller about the required quality. Such a solution in our next work is proposed to be developed in the form of a personal user account.

In future work, we plan to develop QoE-routing for SDN/IoT networks, which, unlike known, to select the optimal data transmission path to use the adaptive QoE-oriented route metric. This metric is automatically calculated by the centralized SDN network controller based on a mathematical model of correlation QoS/QoE. Such improvement will allow users of IoT services to order the required quality in the form of QoE scores from 1 to 5, where higher value means better quality, and the controller analyzes QoE scores to find the best path.

Funding Statement: This research was supported by the Ukrainian project No. 0120U102201 “Development the methods and unified software-hardware means for the deployment of the energy efficient intent-based multi-purpose information and communication networks” and Comenius University in Bratislava, Faculty of Management.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, “Internet of things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang *et al.,* “A Survey on internet of things: Architecture, enabling technologies, security and privacy, and applications,” *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista and M. Zorzi, “Internet of things for smart cities,” *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[4] I. Lee and K. Lee, “The internet of things (IoT): Applications, investments, and challenges for enterprises,” *Business Horizons*, vol. 58, no. 4, pp. 43–440, 2015.

[5] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, “Internet of things: A Survey on Enabling technologies, protocols, and applications,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[6] L. Xu and G. Mcardle, "Internet of too many things in smart transport: The problem, the side effects and the solution," *IEEE Access*, vol. 6, pp. 62840–62848, 2018.

[7] E. Ahmad, M. Alaslani, F. R. Dogar and B. Shihada, "Location-aware, context-driven QoS for IoT applications," *IEEE Systems Journal*, vol. 14, no. 1, pp. 232–243, 2020.

[8] M. Klymash, M. Beshley and B. Stryhaluk, "System for increasing quality of service of multimedia data in convergent networks," in *Proc. PIC S & T*, Kharkov, Ukraine, pp. 63–66, 2014.

[9] B. K. J. Al-Shammari, N. Al-Aboody and H. S. Al-Raweshidy, "IoT traffic management and integration in the QoS supported network," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 352–370, 2018.

[10] S. Mukherjee and G. P. Biswas, "Networking for IoT and applications using existing communication technology," *Egyptian Informatics Journal*, vol. 19, no. 2, pp. 107–127, 2018.

[11] B. Mohammed and D. Naouel, "An efficient greedy traffic aware routing scheme for internet of vehicles," *Computers, Materials & Continua*, vol. 60, no. 3, pp. 959–972, 2019.

[12] G. Tanganelli, C. Vallati and E. Mingozzi, "Ensuring quality of service in the internet of things," *New Adv. Internet Things. Springer Cham*, vol. 715, pp. 139–163, 2018.

[13] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis *et al.,* "Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 70–78, 2017.

[14] I. Gravalos, P. Makris, K. Christodoulopoulos and E. A. Varvarigos, "Efficient network planning for internet of things with QoS constraints," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3823–3836, 2018.

[15] M. Beshley, N. Kryvinska, M. Seliuchenko, H. Beshley, E. M. Shakshuki *et al.,* "End-to-end QoS "smart queue" management algorithms and traffic prioritization mechanisms for narrow-band internet of things services in 4G/5G networks," *Sensors*, vol. 20, no. 8, pp. 2324–2354, 2020.

[16] R. H. Jhaveri, R. Tan, A. Easwaran and S. V. Ramani, "Managing industrial communication delays with software-defined networking," in *Proc. RTCSA*, Hangzhou, China, pp. 1–11, 2019.

[17] P. K. R. Maddikunta, G. Srivastava, T. R. Gadekallu, N. Deepa and P. Boopathy, "Predictive model for battery life in IoT networks," *IET Intelligent Transport Systems*, vol. 14, no. 11, pp. 1388–1395, 2020.

[18] M. J. Piran, N. H. Tran, D. Y. Suh, J. B. Song, C. S. Hong *et al.,* "QoE-driven channel allocation and handoff management for seamless multimedia in cognitive 5G cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6569–6585, 2017.

[19] S. R. K. Somayaji, M. Alazab, MK. M., A. Bucchiarone, C. L. Chowdhary *et al.,* "A Framework for prediction and storage of battery life in iot devices using DNN and blockchain," in *Proc. GC Wkshps (Virtual)*, pp. 1–6, 2020. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber= 9367520.

[20] Y. Fathy and P. Barnaghi, "Quality-based and energy-efficient data communication for the internet of things networks," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10318–10331, 2019.

[21] H. Tahaei, R. B. Salleh, M. F. Ab Razak, K. Ko and N. B. Anuar, "Cost effective network flow measurement for software defined networks: A distributed controller scenario," *IEEE Access*, vol. 6, pp. 5182–5198, 2018.

[22] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky *et al.,* "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[23] M. Seliuchenko, M. Kyryk, M. Beshley and M. Zhovtonoh, "Automated recovery of server applications for SDN-based internet of things," in *Proc. AICT*, Lviv, Ukraine, pp. 25–29, 2019.

[24] A. Andreas, M. Reisslein and W. Kellerer, "Survey on network vrtualization hypervisors for software defined networking," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 655–685, 2016.

[25] M. Beshley, A. Pryslupskyi, O. Panchenko and H. Beshley, "SDN/Cloud solutions for intent-based networking," in *Proc. AICT*, Lviv, Ukraine, pp. 95–98, 2019.

[26] M. Beshley, M. Seliuchenko, O. Panchenko, O. Zyuzko and I. Kahalo, "Experimental performance analysis of software-defined network switch and controller," in *Proc. TCSET*, Lviv-Slavske, Ukraine, pp. 282–286, 2018.

[27] T. Huang, F. R. Yu, C. Zhang, J. Liu, J. Zhang *et al.,* “A survey on large-scale software defined networking (SDN) testbeds: Approaches and challenges,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 891–917, 2017.

[28] W. Xia, Y. Wen, C. H. Foh, D. Niyato and H. Xie, “A survey on software-defined networking,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 27–51, 2015.

[29] S. Bera, S. Misra and V. Vasilakos, “Software-defined networking for internet of things: A survey,” *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1994–2008, 2017.

[30] S. Shahzadi, F. Ahmad, A. Basharat, M. Alruwaili, S. Alanazi *et al.,* “Machine learning empowered security management and quality of service provision in SDN-NFV environment,” *Computers, Materials & Continua*, vol. 66, no. 3, pp. 2723–2749, 2021.

[31] J. W. Guck, M. Reisslein and W. Kellerer, “Function split between delay-constrained routing and resource allocation for centrally managed QoS in industrial networks,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2050–2061, 2016.

[32] J. Hu, C. Lin and P. Zhang, “Performance evaluation and optimization of hierarchical routing in SDN control plane,” *Chinese Journal of Electronics*, vol. 27, no. 2, pp. 342–350, 2018.

[33] X. Shi, Y. Li, H. Xie, T. Yang, L. Zhang *et al.,* “An openflow-based load balancing strategy in SDN,” *Computers, Materials & Continua*, vol. 62, no. 1, pp. 385–398, 2020.

[34] D. Gopi, S. Cheng and R. Huck, “Comparative analysis of SDN and conventional networks using routing protocols,” in *Proc. CITS*, Dalian, China, pp. 108–112, 2017.

[35] H. Geng, H. Zhang and Y. Zhang, “Efficient routing protection algorithm in large-scale networks,” *Computers, Materials & Continua*, vol. 66, no. 2, pp. 1733–1744, 2021.

[36] H. Babbar, S. Rani, M. Masud, S. Verma, D. Anand *et al.,* “Load balancing algorithm for migrating switches in software-defined vehicular networks,” *Computers, Materials & Continua*, vol. 67, no. 1, pp. 1301–1316, 2021.

[37] O. Panchenko, A. Polishuk, M. Seliuchenko and M. Beshley, “Method for adaptive client-oriented management of quality of service in integrated SDN/Cloud networks,” in *Proc. PIC S&T*, Kharkov, Ukraine, pp. 452–455, 2017.

[38] M. Beshley, M. Seliuchenko, O. Panchenko and A. Polishuk, “Adaptive flow routing model in SDN,” in *Proc. CADSM*, Lviv, Ukraine, pp. 298–302, 2017.

[39] N. Saha, S. Bera and S. Misra, “Sway: Traffic-aware QoS routing in software-defined IoT,” *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 1, pp. 390–401, 2018. https://doi.org/10.1109/TETC.2018.2847296.

[40] G. Deng and K. Wang, “An application-aware QoS routing algorithm for SDN-based IoT networking,” in *2018 IEEE Symp. on Computers and Communications*, Natal, pp. 186–191, 2018.

[41] Z. Qin, G. Denker, C. Giannelli and P. Bellavista, “A Software defined networking architecture for the internet-of-things,” in *Proc. NOMS*, Krakow, Poland, pp. 1–9, 2014.

[42] J. W. Guck and W. Kellerer, “Achieving end-to-end real-time quality of service with software defined networking,” in *Proc. CloudNet*, Luxembourg, pp. 70–76, 2014.

[43] A. Iqbal, U. Javed, S. Saleh, J. Kim, J. S. Alowibdi *et al.,* “Analytical modeling of end-to-end delay in openflow based networks,” *IEEE Access*, vol. 5, pp. 6859–6871, 2017.

[44] X. Guo, H. Lin, Z. Li and M. Peng, “Deep reinforcement learning based QoS-aware secure routing for SDN-IoT,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6242–6251, 2020.

[45] S. Zhu, Z. Sun, Y. Lu, L. Zhang, Y. Wei *et al.,* “Centralized QoS routing using network calculus for SDN-based streaming media networks,” *IEEE Access*, vol. 7, pp. 146566–146576, 2019.

[46] M. Fu and F. Wu, “Investigation of multipath routing algorithms in software defined networking,” in *Proc. ICGI*, Fuzhou, China, pp. 269–273, 2017.

[47] F. Tang, Z. M. Fadlullah, B. Mao and N. Kato, “An Intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-IoT: A Deep Learning Approach,” *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5141–5154, 2018.

[48] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila and T. Taleb, "Survey on multi-access edge computing for internet of things realization," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2961–2991, 2018.

[49] M. Klymash, M. Beshley and V. Koval, "The model of prioritization of services for efficient usage of multiservice network resources," in *Proc. TSCET*, Lviv-Slavske, Ukraine, pp. 320–321, 2012.

[50] H. Pehlivan, "Designing and interpreting a mathematical programming language," *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, vol. 23, no. 6, pp. 1027–1041, 2019.

[51] S. Jun, K. Przystupa, M. Beshley, O. Kochan, H. Beshley *et al.,* "A cost-efficient software based router and traffic generator for simulation and testing of IP network," *Electronics*, vol. 9, no. 1, pp. 40–64, 2020.