User Centric Quality of Service Improvement in IoT Networks: A closed Loop Approach

Mutruc Vlad
v.mutruc@student.utwente.nl
University of Twente
Enschede, The Netherlands

ABSTRACT

As the Quality of Service (QoS) requirements in IoT networks continue to rise, the limitations of Enhanced Distributed Channel Access (EDCA) have become increasingly apparent. Network slicing, a powerful solution that addresses these limitations, provides flexibility and QoS diversity in WiFi IoT networks. However, current resource management strategies in network slicing lack the use of end-user QoS satisfaction.

Software-Defined Networking (SDN), which facilitates network slicing, collects KPIs but often neglects user feedback about their experience. This paper proposes an enhancement to the 5GEmPower SDN controller to incorporate this unboserved user feedback, enabling improved slice resource management and QoS.

Our novel framework empowers end-users to share critical parameters such as Modulation and Coding Scheme (MCS), Clear Channel Assessment (CCA), and Transmit Power (TX). The integration of this information can significantly improve the reliability of wireless IoT networks.

The software created during this research is open-source and available on GitLab. [1]

KEYWORDS

User-centric design, QoE, Dynamic AP, WiFi, IoT

1 INTRODUCTION

The diverse range of Quality of Service (QoS) requirements in IoT networks makes it difficult for network managers and administrators to meet QoS in the network. When discussing Quality of Service (QoS) requirements, we are referring to multiple categories of requirements for network resources. Along with these factors, performance, availability, scalability, energy efficiency, and security are also crucial to the functionality of the network. In traditional 802.11 networks, the clients use RSSI (Received Signal Strength Indicator) to establish associations with APs. However, this can result in an uneven and inadequate distribution of resources within the network, particularly when the clients' QoS (Quality of Service) is not taken into account. For instance, in an IoT factory network with a wide array of devices, ranging from temperature sensors to assembly line robots, this approach can result in inefficient resource allocation and a poor user experience. The 'e' amendment in 802.11e attempts to overcome these limitations. [8]

The Enhanced Distributed Channel Access (EDCA) component, introduced in 802.11e, is a widely used approach that categorizes communication flows into Voice, Video, Best Effort, and Background. Based on the assigned category, a device is allowed to access the channel for a period called Transmission Opportunity (TXOP). While this approach has proven effective in small WLANs with limited device types, it lacks flexibility and adaptability in more complex environments, such as large-scale home networks with a variety of IoT devices like smart TVs, home security systems, and personal devices. These limitations can lead to users' QoS requirements not being fulfilled.[3] [Mangold et al.]

To address these challenges, slicing technology, first developed for 5G networks, [13] [5] has been adapted for WiFi networks. This technology logically splits the network into multiple self-contained logical (or virtual) networks called slices, allowing for dynamic resource management. However, despite advancements in slicing technology, achieving fine-grained control over resource allocations remains a challenge. Current scheduling algorithms used in slicing do not take into account the users' QoS satisfaction factors. This means that even though a slice may be allocated for a specific service in a large-scale home network, the actual user experience may not meet expectations if the transmission power or modulation and coding schemes are not optimized for that particular service. Despite the advancements in slicing technology, there remains a challenge in achieving fine-grained and deep control over resource allocations. There were efforts made by researchers in improving resource management of a sliced network. However, they never considered end-user satisfaction based on what actual throughput, packet loss and latency they are achieving [11] [12]. The scheduling algorithms used in slicing do not take into account the users' QoS satisfaction factors. However, considering such information can be beneficial in enhancing users' QoS satisfaction by taking informed decisions on exploiting parameters like modulation and coding schemes (MCS), transmission power, or clear channel assessment (CCA).

In this paper, we present a methodological approach for enabling 802.11 network clients in an IoT environment to dynamically influence resource allocation and management on a 5GEmpower SDN-driven network that utilizes a slicing mechanism. To achieve this, we will assess their real-time experience using network monitoring tools, followed by analyzing and organizing the gathered data. On the receiving end of the controller, we will use Linux applications to store statistical data in a shared database. This data will be made accessible to other applications running on the controller, enabling more informed decisions regarding resource allocation. The proposed method aims to enhance the user experience and improve resource allocation efficiency in IoT networks by transitioning from an open loop approach to a closed loop approach, thereby making the network more user-centric.

Our research makes a significant contribution to the field by addressing the gap in existing research that fails to consider actual quality of service (QoS) satisfaction in resource allocation techniques. This study demonstrates the feasibility of implementing an architecture that enables the incorporation of user feedback into the SDN controller. It emphasizes the importance of including such measurements, highlighting the potential to improve network performance and enhance user satisfaction in Internet of Things (IoT) environments.

2 RELATED WORK

While inspecting the works done on improving the QoS algorithms on top of the 802.11e amendment we identified many successful attempts at queue management where authors managed to elaborate scheduling schemes that would improve the efficiency issues of the HCF (Hybrid Coordination Function) and EDCA introduced in the 'e' amendment. [8] [10][2][Mangold et al.]. The EDCA component attempts to solve the problem of lack of QoS support by introducing a prioritization mechanism based on 4 statically defined categories: Voice, Video, Best Effort, and Background. In an IoT network where there is a wide diversity of types of traffic applications and respective QoS implementing a statically defined mechanism like EDCA did not introduce much improvement in the support of QoS since it has lack flexibility and coverage of a wide range of categories of QoS requirements. Moreover, none of these approaches considers exploiting actual user QoS satisfaction feedback.

With the apparition of WiFi implementation of the slicing mechanism introduced firstly in 5G networks, a whole new world of possibilities appear given the flexibility and dynamic aspect of the new proposed architecture. A multitude of research papers were identified concerning optimizations for Quality of Service (QoS) in IoT WiFi networks. Matías Richard and his colleagues worked on a way towards increasing the QoS via an Airtime-Based Resources Allocation and defining the concept of performance guarantee in a sliced network. [11] [12]. Authors in [3] conducted recent research on SDN-based slice orchestration and MAC management for QoS delivery in IEEE 802.11 networks. The main idea was to influence the creation of resources for each slice and periodical update on those assuring that the QoS requirements of all end-points are fulfilled. In order to dynamically change the resource allocation, their approach uses quantum adaptation via an exponential function algorithm. This algorithm takes the old quantum as input and computes a new quantum based on the allocation and release rates. These rates are initially set to default values and determine the change in the new quantum values. Quantum, being one of the most important principles of slicing, plays a crucial role in the resource allocation mechanism. It determines the value for which each slice has the right to transmit. The slicing methodology also defines different transmission policies guided by the application's QoS like 'No ACK', these changes are logically related to the requirements of the device and are similar to the EDCA approach seen in 802.11 amendment 'e'. They conclude their work by pointing to their improvements in latency with a small cost of throughput. However, they did not consider any user-delivered feedback about QoS satisfaction to drive their decision behaviour. Authors in [6] start from the problems with resource allocation in a diverse IoT network and propose a Lyapunov optimization method with a time-scale separation approach to solve it. They use the Lyapunov mathematical function to quantify the stability of the network over a long time by inspecting queue backlogs. However, they do not exploit any user information related to the QoS they are receiving per time slot which adds some limitations to the approach. A study of optimal resource allocation for an IoT network where devices have strict QoS requirements such as URLLC (Ultra-Reliable and Low-Latency Communications) was conducted in [4] by Isolani. They use mathematical models and probabilistic approaches to describe the behaviour in a typical network and search for appropriate values for the decision variables that dictate the portion of time allocated to each slide on each resource block. This work is characterised by being a novice attempt to exploit the flexibility of airtime allocation considering resource availability and strict KPI (Key Performance Indicator) related to URLLC. This new technology of slicing the network in multiple encapsulated sub-networks having control over the resource management in each of the slices individually is a perfect environment for bringing the optimisation of resource allocation to a maximum. Despite the fact that all of these approaches are undoubtedly innovative and efficient they have one thing in common. We didn't notice in any of the revised papers any attempt to use the clients' QoS satisfaction values in order to address individual communication problems dynamically and improve future satisfaction factors.

Therefore, we propose our intention to study and identify the feasibility and the benefits of a user's experience-aware closed loop strategy of resource allocation within a 5GEmpower SDN-controlled IoT network. Having a clear picture of what the clients are experiencing will help the SDN controller understand that there are issues in the current decisions for resource management that causes some users to lack their QoS requirements being fulfilled and take immediate action by modifying the connection-specific parameters in order to assure better QoS. Upon this logical reasoning, we want to develop an algorithm to measure the QoS satisfaction score and send it to the SDN controller. In addition to it, we would like to elaborate on a protocol to declare and differentiate the QoS requirements for each user.

Having in mind the knowledge we acquired inspecting the existent works and our ideas to make the network design more fine-grained and user-centric we come up with the following contribution to this research paper:

Find a way how QoS satisfaction can be measured by the end-users to make this information available for utilization by the SDN controller in the 5GEmpower framework for improved QoS satisfaction?

This logically splits into the following smaller objectives:

1. Categorize the typical QoS requirements of different types of IoT devices.
2. Measured user satisfaction and sent it to the SDN controller for incorporation into resource management algorithms?
3. Determine a frequency at which the feedback should be shared with the SDN controller to avoid an increase in overhead while improving QoS satisfaction.

In the following sections, we follow with the specifics of our study. The 'Methodology' section outlines the techniques and approaches we utilized in our research, providing a clear understanding of the process. This is followed by the 'Results' section, where we present our findings in a detailed and structured manner. The

'Discussion' section then interprets these results, drawing connections with existing literature and highlighting the implications of our findings. Finally, the 'Conclusion' section describes the achievement of our research, summarizing the key points and talking about potential development directions for the future.

3 NETWORK MODEL AND FRAMEWORK

This research is developed upon the 5GEmpower SDN Networking Platform. 5GEmpower also supports network slicing dedicating a separate module for customized and optimized service delivery for different applications or user groups. Users can create a set of slices and manipulate such parameters as quantum and resource allocation policies for each of the slides. The central point of this platform is the controller, a Python-based program that runs in the background and is responsible for overall network management, resource allocation and policy reinforcement. It also enables user-written apps to involve in network management by manually setting network-related parameters per each slice. The 5GEmpower ships with a showcase collection of demo applications for demonstrating the possibilities and flexibility of the system. The AP for working with 5GEmpower should run the empower-openwrt operating system that includes necessary Kernel packages. Some basic knowledge of Python and the examples provided in the 5GEmpower default package together with an abstract understanding of how the system is working is enough for the purposes of this research project. Another component of the system that makes inter-application communication possible is the shared database InfluxDB. We used it for saving the data received from the users of the network structured in entities called measurements in IfluxDB from where any other application with the intention to improve the resource allocation scheme using the QoS satisfaction factors from users could exploit the available data.

The implementation we developed consists of 2 modules. First, we develop an app within the 5GEmpower framework that will initiate an MQTT broker and a background running task that in case of receiving an MQTT published message on the delegated topic will record the information into the existing influxDB used by the 5GEmpower framework. This data would be accessible to another application that would use it to improve the QoS of the network. The second component we developed is at the client end, a Python script supposed to run all the time that accomplishes the following 2 tasks:

- Inspects the real-time generated capture files calculating the throughput and any other deductible parameters from the information available in the network capture.
- Publishes the calculated QoS ratios to the MQTT topic towards the broker initiated by the app on the SDN controller.

Before we describe the proposed implementation in a higher level of detail it is necessary to understand the way a generic 5GEmpower network that we used for our experiments is functioning at some point of abstraction (see Figure 1).

The central point of this network is the SDN controller where the executable of the framework is installed and running while the network is up. The main purpose of this process is to be able to add flexibility to the network by being able to control the Access Point of the network based on a catalogue of applications that are meant to play a specific role in the managing and monitoring of the network. This controller application is linked to an Influx database, a time series database allowing the apps running to have a common shared memory environment upon which they can make informed decisions. The machine on which the controller is running in our case is a Ubuntu Linux distribution, it is also required that it has a connection to the Access Point of the network.

An access Point is another machine capable of running the OpenWRT environment, in the official documentation of the 5GEmpower project, such an access point is called WTP so we will use these terms interchangeably. For the correct functioning of this WRT, it has to be connected both to the SDN controller and to have an Internet Connection through a local network device such as a router.

In the role of a client, we used a raspberry pi machine with a Linux distribution installed configured to automatically connect to the Wi-Fi network distributed by the AP.

The client being connected to the network is able to benefit from the functionality added by the 5GEmpower framework. In Figure 1 you can also notice an explicative note about the connection of the client to the AP that gives insight into the type of connection they have and abstraction about the resource management technique used. In our setup we initiated 2, in addition to the default '0', slices with the respective identifiers 8 and 24 where 8 and 24 are the DSCP values attached to the packets that correspond to each slice. The resource allocation is chosen as the default one of the framework, the Round Robin.

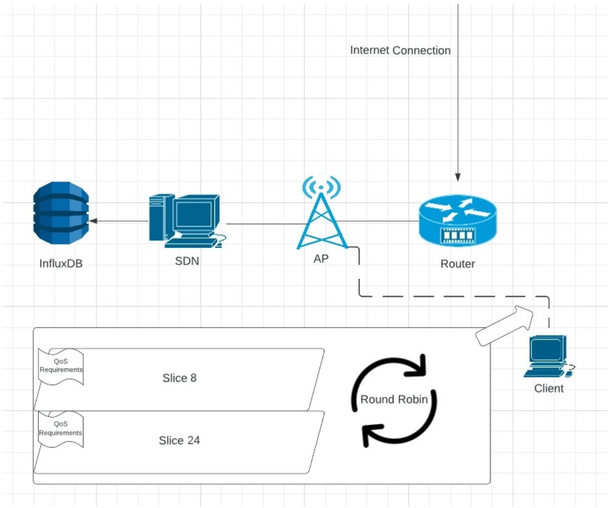

Figure 1: The representative scheme of the 5GEmpower network

The application running at the client side is running parallel with an active packet capturing tool that generates capture files at a fixed time rate synched with the time rate at which the application is ready to digest them calculating the throughput satisfaction rate for each of the file and transferring them to the SDN controller.

$$T_{sat} = \frac{\sum_{b=0}^{EOF} b}{REQS[throughput]_{DSCP}} \tag{1}$$

Equation (1) shows how we calculated the throughput satisfaction factor for each active application with specific QoS requirement.

Table 1: QoS requirements table

MAC	Slice ID	Throughput (bps)	Latency (ms)
AA-AA-AA-AA-AA-AA	8	16000	100
	32	8000	400
	64	6000	1000
BB-BB-BB-BB-BB-BB	8	4000	1000
	32	2000	300
CC-CC-CC-CC-CC-CC	8	32000	200
	32	4000	200
	96	6000	100
	128	6000	100

Table 2: QoS feedback data

MAC	Slice ID	Throughput Satisfaction	Latency Satisfaction
AA-AA-AA-AA-AA-AA	8	0.3	0.5
	32	1.2	1.5
	64	0.2	0.4
BB-BB-BB-BB-BB-BB	8	0.5	0.1
	32	1.1	1.0
CC-CC-CC-CC-CC-CC	8	1.113	2.4
	32	0.56	0.6
	96	0.4	0.3
	128	0.2	0.01

To organize the data, we have developed the following structure for defining QoS requirements at each end node, as well as a similar structure for storing it in the InfluxDB on the controller. Table 1 provides insight into the representation of QoS requirements, while Table 2 illustrates how an example the collected feedback will be stored in the database. These representations are flexible and can be extended to accommodate any additional parameters required by the concrete resource allocation mechanism. However, an important requirement is that the structures of these models must be synchronized between the client and the controller. This is necessary to prevent the transmission of data from the client that the controller does not expect, or vice versa.

On the controller side, we have an app that dispatches a background task for listening to the MQTT messages from the clients and as soon as receives one calls the method on the main thread for digesting and storing this information in the influx db.

The advantage of such a framework like 5GEmpower is that in future is easy to customize the app by for example implementing also a mechanism that would notify other apps when the QoS satisfaction is stored so they can immediately use it.

4 FRAMEWORK EVALUATION

Our research came up with a practical answer to the proposed research question. We achieved a working solution that in an active network collects data about the QoS satisfaction of the user and sends that to the SDN controller for further use. We also met some difficulties related to the speed of computation of the satisfaction factor at the client so in answer to Research Sub Question 3 we need an implementation that would allow us to better control the frequencies of sending.

To evaluate our implementation, we conducted an experiment that utilized the network structure illustrated in Figure 1. We utilized the Linux command line tool, iperf3, to generate the downlink traffic. Iperf3 is a traffic generation tool that enables experimentation with various application parameters. Two separate terminal instances were created on the machine hosting the 5GEmpower runtime. For each instance, traffic targeting the network client was generated by running the iperf3 command. For the first generated traffic stream, a DSCP value of 8 was set, and for the second stream, a DSCP value of 24 was assigned. These values were used to attribute the application their respective available slices slices, 8 and 24. In our experiment slices 8 and 24 represent 2 distinct application running on the network, the number of application that we can hanlde is limited only by the technical limitations of the 5gEmpower framework and available for use DSCP values on the network protocols used.

While maintaining the continuous stream of traffic, the application within the controller framework was launched and began listening for incoming MQTT messages. Additionally, the script responsible for capturing packets on the client was initiated. This script divides the problem into two distinctive phases: capturing packets for a fixed amount of time and reading the captured packets from a file. It then sums up the size of all the packets coming on each of the streams and sends the results in a structured data frame to the controller via the MQTT protocol. The script also appends its own MAC address to the frame to make it distinguishable at the controller.

In Figure 3, you can observe the results of the above-mentioned experiment, where each color represents a separate slice. The y-axis on the graph represents client satisfaction with throughput for each of the two slices.

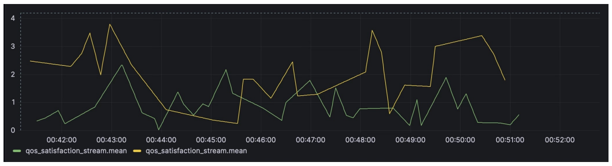

Figure 2: Throughput satisfaction over time

The application throughput satisfaction is continuously monitored and can be viewed through Grafana dashboards (Figure 2) to track application performance. They can also be used by resource management applications within the 5gEmpower SDN controller for a improved and more informed resource allocation decisions.

5 FUTURE WORK

Having explored the potential of our approach in the domain of machine learning and its application in possible improvements of QoS requirements fulfill in the network, we recognize that there are still a lot of possibilities yet to be discovered. Some future directions of development based on this research can be revealed such as:

Extension of the range of values to be included in the feedback provided to the controller. Currently, our implementation focuses on sending the throughput satisfaction rate to the controller. However, it is possible to expand the range of values to include other metrics such as MCS, TX Power, and Clear Assessment Threshold. This would undoubtedly enhance the effectiveness of the closed-loop system by enabling it to make better decisions based on more comprehensive insights into the state of its users. Implementing a mechanism to calculate and append the user's current MCS value would assist the SDN in selecting the most appropriate and suitable MCS. This, in combination with further development in enforcing a client to use a specific MCS, would have a positive impact on the controller's awareness and orchestration authority status on the network. Similarly, TX power feedback will enable the network to optimize the transmission power levels of devices, thereby enhancing coverage and signal strength as well as achieve better energy efficiency. CCA feedback adds a level of protection against congested channels, enabling more efficient allocation of resources from the available spectrum. Sharing these parameters with the central authority of the network will enable the network to adapt to changes in environmental conditions, interference levels, and unfulfilled user demands. There are already efforts being made to develop parameter adjustment based on a machine learning model. Incorporating user feedback into the algorithm will exponentially increase the QoS in network and user satisfactions.

Recently, Machine learning has been greatly employed to solve different network problems and research is being focused on developing distributed learning solutions to improve scalability, specifically in IoT networks. Federated learning, which is class of distributed learning, is a decentralized machine-learning approach that enables training models locally on end devices while only model updates are being exchanged between the central coordinator and end devices. Since data privacy is also a concern in this type of data learning, with the proposed approach we can define from the beginning a set of rules that would regulate the type of data used for FL. Using the clients of a 5GEmpower network for this type of distributed learning would ease the pressure put on the central agency and improve resource management algorithms given that they are based on a model trained on the data actually experienced by the users and not assumptions. However, federated learning requires parameters of trained ML models to be shared with centralized controller running on the SDN controller. This parameter sharing can be efficiently done through our developed MQTT based framework which is currently being used for QOS satisfaction sharing. By addressing the limitations of open-loop systems and proposing a closed-loop approach, this research provides valuable insights and options for improving QoS in distributed networks. So concluding this, the implementation of FL can lead to improved performance, privacy-preserving training, and better utilization of network resources [7]

The lightweight of the developed framework was achieved by using a protocol like MQTT we can also extend it by making it two-way communication with the controller. Such an approach to the future development of the framework will add such improvements like real-Time protocol adjustments synchronisation making it possible to enforce changes to the QoS of individual clients from the controller. It can also help network administrators to develop more robust and dynamic control over slice resources in a network slicing enabled wireless IoT network.

6 CONCLUSION

In this study, we have explored the feasibility of transitioning from an open-loop problem to a closed-loop one by implementing an algorithm that calculates the Quality of Service (QoS) satisfaction rates for users of a 5GEmpower network. These rates are then efficiently transmitted to the SDN controller, enabling it to make informed decisions regarding resource allocation. Through this work, we have identified the current limitations of this approach and have suggested several options to address them. The topic under research has great potential, especially given the need for network decentralization. The progress made on this topic mainly addresses the most important issue, which is improving quality of service (QoS) satisfaction for users. This has been achieved through the implementation of a feedback system and has been proven to be feasible. This opens the way to extend this work and actually implement a mechanism that uses the information and knowledge gained to make real improvements in QoS satisfaction.

REFERENCES

[1] (2023). cs20-04 / 5GEmpower qos satisfaction · GitLab.

[2] Charfi, E., Gueguen, C., Chaari, L., Cousin, B., and Kamoun, L. (2017). Dynamic frame aggregation scheduler for multimedia applications in IEEE 802.11n networks. *Transactions on Emerging Telecommunications Technologies*, 28(2):e2942. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/ett.2942.

[3] Isolani, P. H., Cardona, N., Donato, C., Marquez-Barja, J., Granville, L. Z., and Latre, S. (2019). SDN-based Slice Orchestration and MAC Management for QoS delivery in IEEE 802.11 Networks. In *2019 Sixth International Conference on Software Defined Systems (SDS)*, pages 260–265, Rome, Italy. IEEE.

[4] Isolani, P. H., Cardona, N., Donato, C., Perez, G. A., Marquez-Barja, J. M., Granville, L. Z., and Latre, S. (2020). Airtime-Based Resource Allocation Modeling for Network Slicing in IEEE 802.11 RANs. *IEEE Communications Letters*, 24(5):1077–1080.

[5] Iwamura, M. (2015). NGMN View on 5G Architecture. In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pages 1–5. ISSN: 1550-2252.

[6] Kwak, J., Moon, J., Lee, H.-W., and Le, L. B. (2017). Dynamic network slicing and resource allocation for heterogeneous wireless services. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–5. ISSN: 2166-9589.

[7] Li, L., Fan, Y., Tse, M., and Lin, K.-Y. (2020). A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854.

[8] Luo, H. and Shyu, M.-L. (2009). An Optimized Scheduling Scheme to Provide Quality of Service in 802.11e Wireless LAN. In *2009 11th IEEE International Symposium on Multimedia*, pages 651–656.

[Mangold et al.] Mangold, S., Choi, S., May, P., Klein, O., Hiertz, G., and Stibor, L. IEEE 802.11e Wireless LAN for Quality of Service.

[10] Pang, W. L., Chieng, D., and Ahmad, N. N. (2013). Adaptive Priority Sliding Admission Control and Scheduling Scheme for DCF and EDCA WLANs. *Wireless Personal Communications*, 70(1):295–321.

[11] Richart, M., Baliosian, J., Serrat, J., and Gorricho, J.-L. (2020). Resource Allocation and Management Techniques for Network Slicing in WiFi Networks. In *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6. ISSN: 2374-9709.

[12] Richart, M., Baliosian, J., Serrat, J., Gorricho, J.-L., and Agüero, R. (2019). Slicing in WiFi Networks Through Airtime-Based Resource Allocation. *Journal of Network and Systems Management*, 27(3):784–814.

[13] Zhang, S. (2019). An Overview of Network Slicing for 5G. *IEEE Wireless Communications*, 26(3):111–117. Conference Name: IEEE Wireless Communications.