2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops) | 978-1-6654-5381-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/PERCOMWORKSHOPS56833.2023.10150306

Autonomous Network Slicing and Resource Management for Diverse QoS in IoT Networks

Sri Harsh Amur, Kamran Zia, Alessandro Chiumento and Paul Havinga
Pervasive Systems Group, EEMCS University of Twente, Enschede Netherlands
s.h.amur@student.utwente.nl, [k.zia, a.chiumento, p.j.m.havinga]@utwente.nl

***Abstract*—The diversity of QoS requirements in modern IoT networks is expanding thus requiring more flexible and autonomous solutions. Network Slicing is an efficient technology that can enable efficient and flexible management of QoS in IoT networks by dividing network resources into small chinks and assigning them to the slices. However, proposed solutions in literature require predefined slices which require prior knowledge of network traffic thus limiting application of these solutions. To address this limitation, we have developed a dynamic and autonomous network slicing solution in a WiFi based IoT network using 5GEmpower SDN controller that creates slices depending on user traffic flows and priorities. Moreover, it autonomously estimates slice throughput requirements to subsequently allocate network resources to meet their QoS requirements. Our system is able to differentiate between multiple QoS requirements and provides framework for flexible prioritization policies for better QoS in IoT networks.**

***Index Terms*—Quality of Service Slicing, 5GEmpower SDN, QoS in IoT Networks, QoS Management, Slice Requirements Estimation, Traffic Rules Abstraction**

I. INTRODUCTION

The number of devices connected to the Internet has been increasing rapidly. IoT devices have seen tremendous growth in the last few years [1] which led to an increase in the volume of heterogeneous traffic patterns [2]. Many new IoT uses cases have emerged in various industries such as healthcare and industry 4.0 [3] [4]. These use cases have introduced diverse QoS requirements in terms of throughput, latency and packet loss ratios. The success of IoT use cases depends heavily on the connectivity and QoS requirement. Failure to do so results in bad user experience and failure of the use case. Meeting diverse QoS requirements in wired networks is relatively easy because of reliable link bandwidths and latencies however, it becomes a challenge in wireless networks where the environment is dynamic and unpredictable due to interferences from multiple sources.

In WiFi networks, QoS is implemented with the help of Enhanced Distributed Channel Access (EDCA) Access Categories (AC) which employ layer two parameters like Contention Window, Arbitrary Inter frame Space Number (AIFSN) and Transmission Opportunity (TXoP) to provide different throughput to different types of application and traffic flows [5]. There are 4 AC namely Voice, Video, Best Effort and Background to provide QoS in network however, these access categories have fixed structure [6] and lacks flexibility to dynamically manage QoS traffic flows for maximizing user satisfactions [7]. Moreover, the best effort and background traffic flows are starved in presence of video and voice traffic which are treated with high priority [8]. The large amount of video and voice traffic also leads to more collisions in the network thus reducing network throughput [9]. On top of it, WiFi QoS framework cannot differentiate between more than 4 types of QoS requirements and cannot cope up with rising diversity in QoS requirements in IoT networks. Therefore, there is a need to implement a QoS framework that can differentiate between larger number of QoS requirements and dynamically manage network resources for large number of QoS flows in the network.

To address the limitation, Network Slicing is an efficient technology that creates QoS slices in WiFi depending on IoT sensors and application's QoS requirements. The network resources (airtime) can then be dynamically assigned to the slices to meet their QoS needs. In our proposed approach, network (QoS) slices are created based on DSCP values of IP packets and packets are subsequently served in their own slices within the AP. Each slice gets a part of airtime in the wireless medium depending on its throughput requirements which are estimated based on average packet size and packet arrival rate in each of the slice. DSCP values a good reflection of the application QoS requirements and can be used to create, manage and configure slices for meeting application QoS requirements. More airtime is allocated to high-priority slices so that they would get more time to transmit their packets compared to low priority and delay tolerant slices. To prioritize one user over another, we have added a traffic rule abstraction mechanism to alter their DSCP values to different priority. Such a mechanism is not present in standard WiFi QoS framework. In order to develop and test our proposed framework, we have employed a Software Defined Networking (SDN) controller called 5GEmpower [10] that collects traffic flows statistics from APs in the network to create and manage QoS slices. The framework is developed in a real world test bed using PC Engines APU2 board with OpenWRT 19.07 OS. The main contributions of our paper are as follows:

- *Flow Monitoring*: A slice flow monitoring framework has been developed to autonomously estimate slice throughput requirements in real time. This estimation can then be used for slice resource management and QoS assurance in the network.
- *Autonomous QoS Slicing*: An autonomous QoS slicing framework has been developed that create / deletes slices

dynamically, based on QoS flows present in the network, to efficiently manage QoS requirements and enhance user satisfaction.
- *Traffic Rules Abstractions*: Traffic Rule abstraction has been developed to reassign DSCP values to the traffic flows to prioritize users over one another and to maximize resource utilization.

II. RELATED WORK

Slicing is a key architecture of 5G and has been actively researched for the past few years. However, most of the research has been done about network slicing or Infrastructure Slicing where the concept of network virtualization is applied over a common network infrastructure and different slices are assigned to different tenants to support multiple tenancy. Most of the research work belongs to slice scheduling [11] where a scheduling algorithm is used to assign resources to different slices. Although this is an important topic which still faces many challenges, this paper remains limited to QoS slicing based on the classification of network traffic's QoS requirements.

Authors in [12] have controlled QoS assurances in WiFi based network through Proportional Time Deficit Round Robin (PT-DRR) queuing where a modified version of DRR is used to assign quantum to the traffic flows to meet their QoS requirements. Instead of bytes being used as a deficit, the actual airtime being consumed by flow is used as a deficit in their queuing structure. The proposed technique is a promising way to provide QoS in WiFi based networks.

Similar to [12], authors in [13] used Airtime Time-Excess Round Robin (ATERR) scheduling mechanism which aims to fairly distribute the airtime between clients. The authors recalculated the quantum value of each queue after every packet transmission to estimate the time taken by packets to reach the destination. Using such estimations and ATERR, they were able to guarantee slice satisfaction and thus QoS satisfaction. However, the slices were created manually by network administrator and then resources are managed for slices. Authors in [14] have employed control theory to target channel access parameters like Contention Window (CW) however, their objective was to maximize network throughput with different traffic types. They did not user QoS satisfaction while devising their solution. Similarly, authors in [15] [16] have also targeted EDCA parameters to optimize and achieve better QoS however, their solution remain limited to 4 QoS access categories of WiFi and cannot be applied to current IoT networks with diverse range of QoS requirements.

Different from other works, we have created a platform which can autonomously create slices and manage slice resources (airtime) by autonomously estimating their throughput requirements. We have controlled airtime through a scheduling mechanism proposed by [17] that considers Modulation and Coding (MCS) scheme being employed at lower layer to achieve better resource assignments as per channel conditions.

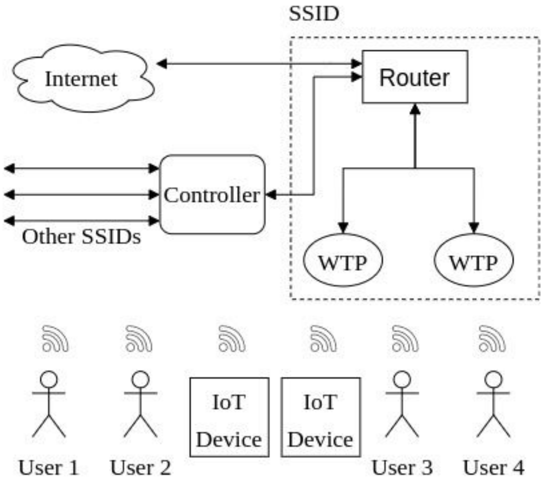


Fig. 1. Network Set-up.

III. NETWORK MODEL AND AUTONOMOUS NETWORK SLICING

The network model used in our paper consists of a SDN controller running in a AMD Ryzen 7 5700U laptop using Ubuntu 20.04 OS. The Access Points (APs) were connected through wired LAN through a router to the SDN controller as shown in Figure 1. The APs, also referred to as Wireless Terminal Points (WTPs) in this paper, were running OpenWRT 19.07 with our modified Empower-Agent running inside a PC Engines APU-2 board. The IoT devices were emulated with the help of raspberry Pi Zero running Raspian OS. The controller receives statistics from all WTPs about the different type of traffic flows present in the network. The traffic flows were emulated with the help of Iperf3 where TCP flows with different DSCP values were generated in the downlink. These statistics were used by the controller to autonomously create network slices and assign resources depending on their throughput estimates. These estimates were done using average packet size and packet arrival rate in each slice.

Since there can many types of traffic in an IoT network with different DSCP values, our proposed slicing and QoS management framework provides flexibility to create any number of slices and serve their QoS requirements however, standard WiFi can also serve 4 QoS classes. Although IEEE 802.11aa has proposed alternate video and voice category through credit based queuing [18], it still cannot provide better QoS in the network. Having more QoS classes allows the controller to render more granular control over the QoS requirements in the network.

A. Slice Resource Management

Slice Resource Management is implemented to autonomously configure QoS slices and assign resources for meeting slice requirements. The slice resource configuration needs to be modified every time a slice is created or removed

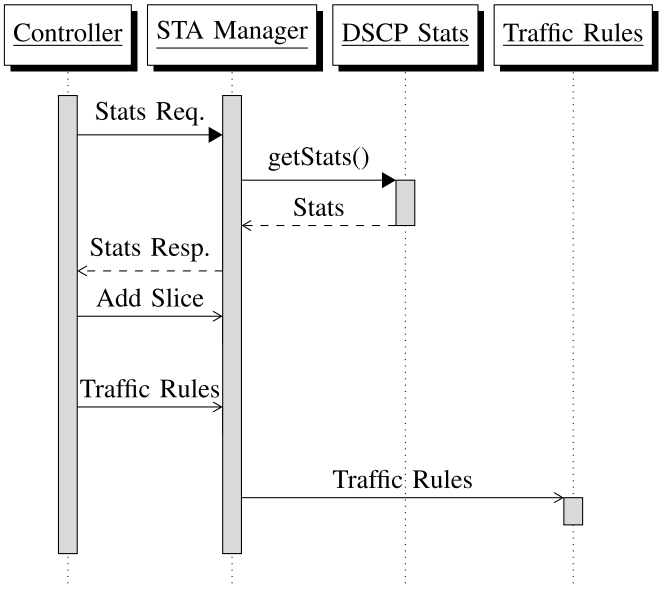


Fig. 2. Interaction between Controller and WTP.

or if the network conditions change. Based on these changes, the controller continuously monitors slice resource assignments for reconfiguration, if needed. In case of new flows in the network, the controller sends instructions to create a slice to all WTPs. If the slices already exist, the controller modifies the configuration of those slices as per QoS needs and sends the updated configuration to all WTPs. In case traffic flows stop, the corresponding slice is removed after few seconds to release resources for remaining slices.

To prioritize slice over one another and to create QoS priorities, quantum values to slice scheduler is assigned. However, this assignment can include other MAC and PHY layer parameters to develop more control over slice resources. The maximum quantum value of the AP is limited to 10000 to limit the scheduling latency of flows to 10ms. Thus the slices get quantum values which add up to a value of 10000. Giving a higher quantum proportion to a slice leads to the slice having higher throughput and low latency.

B. Traffic Rules

Traffic Rules (TR) is an application running in the WTP and is responsible for ensuring incoming network packets are inserted into the right slices as instructed by the controller. If the traffic flows with similar QoS requirements are having different DSCP values, TR app changes the DSCP field of packets to slice DSCP value to which to they belong as per their QoS requirements. Resultantly, the flow is handled as per slice policies defined by the controller.

C. DSCP Statistics App and Slice Throughput Estimation

For the controller to have an idea about the slice QoS requirements, it requires real-time statistics from all the WTP. The DSCP statistics is an application running in the WTP responsible for collecting and storing the slice packets statistics. The controller poll these statistics every second to estimate slice throughput requirements. These statistics include number of packets arriving per second in the slice as well as the average packet size. They can help estimate the slice throughput requirements which are then used for slice resource management application.

D. Packet Flow in QoS Slicing Framework

Figure 3. shows the packet flow inside the WTP. The packets arriving on the WTP first needs to be assigned a more appropriate DSCP before being sent to the stations. A traffic classification is required inside the AP that looks at packet DSCP value however, other traffic features like packet size, packet arrival rate, port No, Src and Dst addresses etc. can be used to determine its application type and QoS requirements. Afterwards, TR app can assign an appropriate DSCP to the flows depending on their QoS requirements. Currently, traffic classification is not implemented in our framework and is a work in progress. The traffic flows in our network are pre-tagged with DSCP values to differentiate between different type of flows. As the packet flows through the WTP towards the stations, DSCP stats element inside it collect packet statistics (packet rate and sizes) and then they are transmitted following slice scheduling policy.

Fig. 2 shows the interaction between the controller and the WTP. It starts with the controller requesting the statistics. The LVAP (station) manager collects the statistics from the DSCP Stats element and sends them back to the controller. The controller gets these statistics from all active WTPs and forms an idea of the whole network. It then analyzes these statistics to form slice groups and traffic rules. It either adds slices if they do not already exist or updates the existing slice configurations. In addition, it also sends the traffic rules to the station manager so that packets are put into the slice groups instead of making a slice for each DSCP.

IV. PERFORMANCE EVALUATION

The functionalities offered by the dynamic QoS Slicing platform were tested by organising 2 experiments. These experiments were conducted specifically to observe if 1.) the controller is able to create slices based on DSCP values in traffic or not 2.) the slices can maintain their prioritization 3.) AP is able to estimate slice throughput requirements and 4.) whether a new slice is created if a critical application/user demands a high priority. The experiments are framed in such a way that the WTP faces congestion after a certain time and needs to prioritize which packets to transmit and which ones to drop. This is because QoS satisfactions can be easily maintained in a lightly loaded network however, problem arises in case of increased network load. The set-up consists of one WTP, Raspberry Pi clients and a controller. The flows are generated using IPerf3 from the laptop and are sent to the client via the WTP.

A. Experimental Scenario 1

The aim of this experiment is to test whether the WTPs are able create slices as per flows in network and guarantee

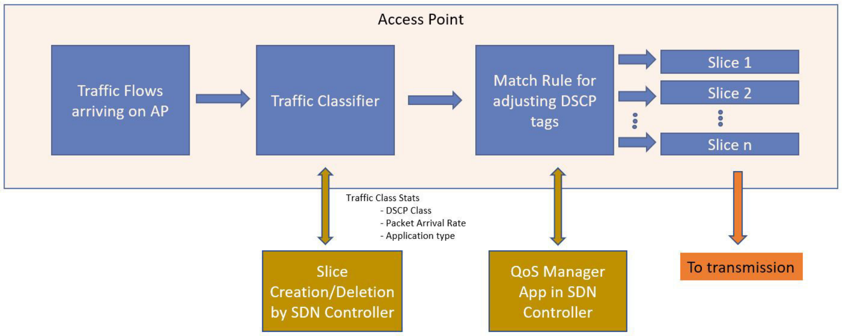
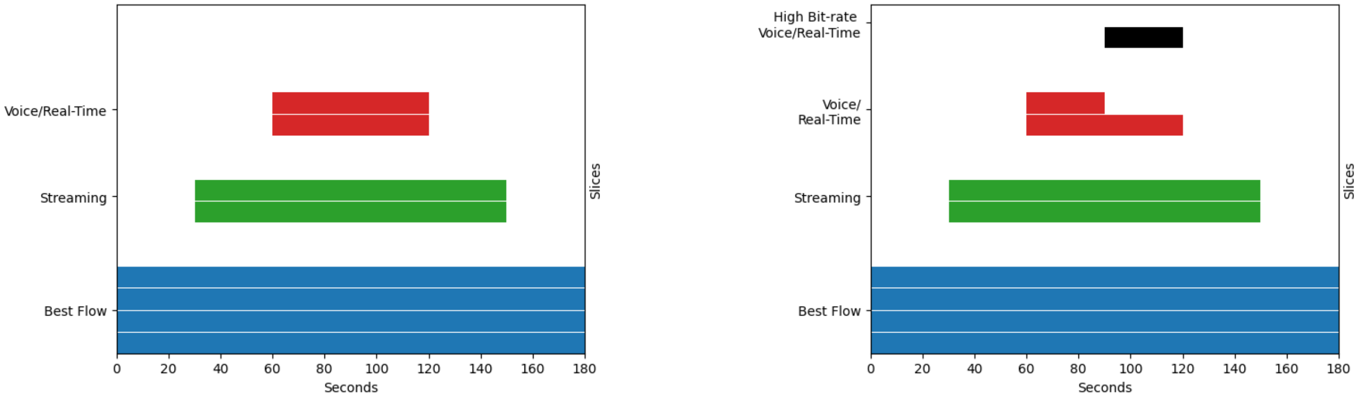


Fig. 3. Traffic Flow in 5G-Empower Controlled WiFi AP



(a) Experimental Scenario 1

(b) Experimental Scenario 2

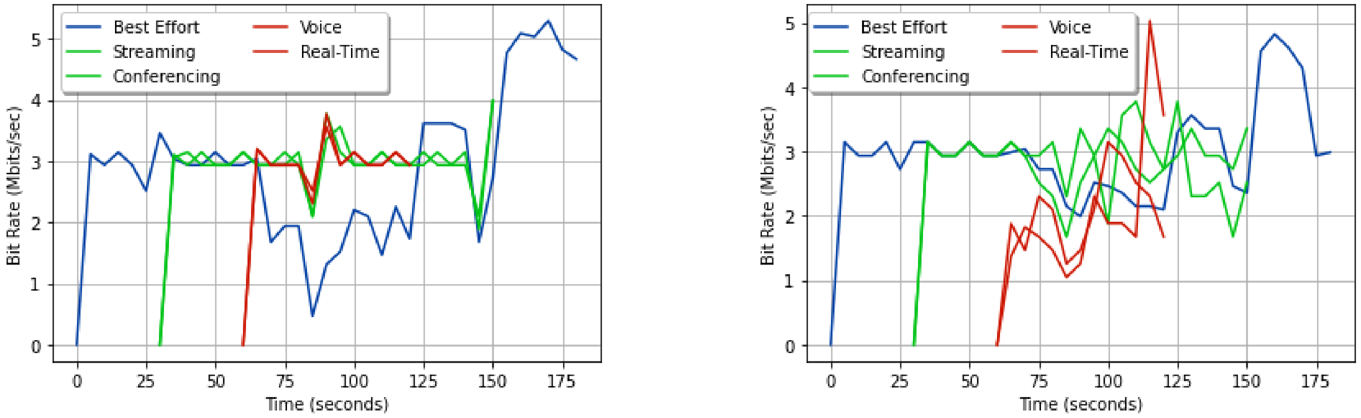
Fig. 4. Experimental Setups for Performance Evaluation

the slice prioritization made by the controller. The higher priority slices should be guaranteed their QoS requirements against the low priority slices. This also tests if the controller is able to create traffic rules to group similar flows in the same slices. The network has one WTP and one controller. At the start, i.e 0 seconds in Fig. 4(a) 4 Best Effort TCP Flows are transmitted to the WTP. They are all kept in a single slice and no prioritization takes place as of yet. After 30 seconds, 2 new flows are added. These flows have a DSCP of 32 and 38 respectively. Since these indicate streaming and conferencing applications, they need to be given a higher priority. The controller should make a new slice for DSCP 32 with a higher quantum value (for higher priority) than BE and should also make a traffic rule to change DSCP 38 into 32 so that both flows utilise the same slice. After the 1-minute mark, 2 more flows are added. These flows have a DSCP of 44 and 46. They indicate voice calls and real-time interaction and should be prioritized over the other flows. Therefore, the controller makes a new slice for DSCP 46 with a higher quantum than both the other slices. In addition, a new traffic rule should be made to change DSCP 44 into 46. After the 2-minutes mark, the latest 2 flows are now removed (DSCP 44 and 46). The controller needs to re-organise the slices with different quantum values, after which the configuration goes back to that of the 0 to 1-minute mark.

B. Experimental Scenario 2

The aim of this experiment is to test whether the controller is able to notice the change in the data rate of the flows and modify slice configurations dynamically to adapt to the new data rates.

The second scenario is made to test slice requirement estimations and slice management capability in case of high bit-rate requirement by a high priority slice in the network. At the start as shown in Fig. 4(b), 4 Best Effort TCP flows (Flow A) are transmitted to the WTP. They should all have the same priority since they would be kept in one slice. After 30 seconds, 2 new flows are added with the DSCPs 32 and 38 which are streaming and conferencing packets (Flow B). After the 1-minute mark, 2 new flows with DSCP 44 and 46 (Flow C) are added. So far the AP should have 3 slices, one for Best Effort, one for Streaming/Conferencing and one more for Real-Time flows. At the 90th second, the flow with DSCP 44 increases its data rate because of high bit rate voice traffic. The controller should notice that real time slice is demanding more traffic due to the increase in packet arrival rate of DSCP 44. To handle this increasing throughput demand, controller should create a separate slice for DSCP 44 (which was previously grouped with DSCP 46) to meet its rising demand and keep



(a) Proposed QoS Slicing

(b) Without QoS Slicing

Fig. 5. Comparison of Results for Scenario 1

it isolated form DSCP 46 traffic. In addition, it should also remove the previously existing rule for remarking its DSCP values to DSCP 46 for handling by real time slice. Increasing the quantum (airtime) without creating separate slice for DSCP 44 would not provide an isolated bandwidth to DSCP 44 because increased quantum would be shared between DSCP 44 and DSCP 46 traffic. After the 2-minutes mark, Flow C stops and the controller should be able to re-organise the slices with suitable quantum values. Finally, at the 150th second, Flow B stops as well and only best effort flows remain in the network.

C. Scenario 1 Results

The bit rate expectation of each flow was set to 3 Mbps. Each experiment was done twice, once with slicing enabled and once without slicing. This was done to check the benefits slicing provides.

D. Slice Grouping and Prioritization

In Fig. 5(a) Flow A (Best Effort) starts and stabilises at around 3 Mbps. Flow B (Streaming and Conferencing) also stabilises at 3 Mbps. However, once Flow C (Real time) starts, the network congestion begins due to slice requirements being greater than the available channel capacity. The AP now checks which flow has the highest priority and assign slice resources as per slice priority. Therefore, Flow C gets 3 Mbps. Similarly, since Flow B has second priority, it also gets 3 Mbps. Finally, the remaining bit rate is provided to Flow A. After Flow C is terminated, due to the available bit rate in the channel, Flow A shoots up as show in the figure. The rapid rise of throughput of Flow A is because of available channel capacity to the full buffered traffic of this TCP flow.

Fig. 5(b) shows the result of network with no slicing employed by the AP. Since the AP does not know how to prioritize the flows, each flow get equal share of the available channel capacity and AP is not able to prioritize high priority flows over low priority flows. Slicing gives the AP the ability to manage resources in a much more efficient and intentional way. Based on the slice priority, the AP can ensure that the bandwidth is distributed proportionally and user QoS requirements are met.

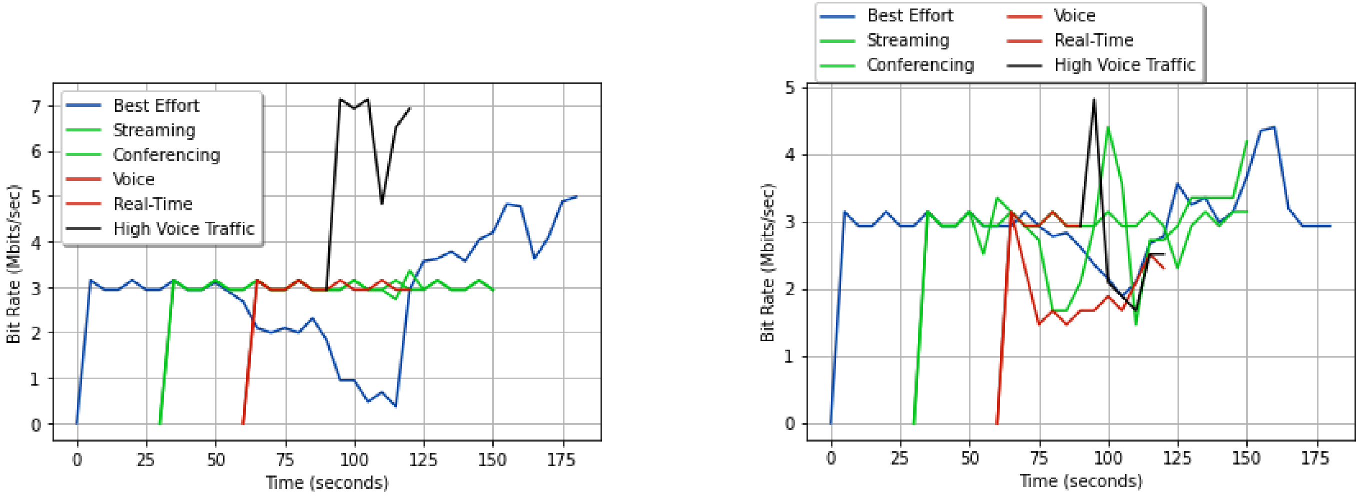
E. Scenario 2 Results

In Fig. 6(a) Flow A and B start one after the other and stabilise at 3 Mbps. At the one-minute mark, Flow C joins the networks. This leads to congestion and the AP reduces Flow A's bandwidth to cater to Flow B and C. At the 90th second, one of the flows in Flow C increases its bit rate to 7 Mbps. The AP detects this increase in slice throughput requirements and keeping in view the priority of this flow and its high bit rate requirement, the controller creates a separate slice for this flow (DSCP 44) and assigned required quantum to it to provide the desired 7 Mbps bit rate. The traffic requirement of DSCP 46 are still met as it is also high priority traffic flow and controller ensures this through suitable quantum assignments. This further decreases the bandwidth of Flow A and ensures the high priority slice's requirements are met satisfactorily.

In Fig. 6(b) the same network flow without slicing is shown. It firstly does not give priority to Flow C which leads to a delay in packet arrival. This would disrupt the voice quality of the end-users. Secondly, when the Flow demanded a higher bandwidth of 7 Mbps, the AP was not able to provide its desired bit-rate estimated by the empower-agent deployed in the AP (WTP). It can be seen from the results that our QoS slicing framework is able to create QoS slices autonomously based on traffic flows in the network. It can reassign flows to available QoS slices to ease out slice resource management and can prioritize high quality flows over other flows. In case of high throughput requirements by high priority flows, it can correctly detect those requirements and realign resources to meet these needs dynamically.

V. CONCLUSION AND FUTURE WORK

QoS Slicing has shown the potential to manage a network with diverse traffic patterns. Present proposals require manual slice creation/deletion and pre-defined requirements to meet



(a) Proposed QoS Slicing

(b) Without QoS Slicing

Fig. 6. Comparison of Results for Scenario 2

QoS in the network. In this paper, we have proposed a 5G-Empower based framework that dynamically creates and manages QoS slices in an SDN-controlled network. Our framework uses slice grouping to combine traffic flows with similar QoS requirements in order to keep the number of slices low which in turn leads to easy slice resource management. We have demonstrated that our QoS slicing framework can provide high quality QoS in wireless IoT networks in an autonomous and dynamic manner and can scale over larger number of traffic flows in the network.

Future work can be done to develop machine learning based efficient slice resource management algorithms. Currently, slice resources are handled through quantum assignments however, more physical and MAC layer parameters like MCS, frame sizes, transmit powers, channel RSSI etc. can be included in slice configurations for better control over the slices to manage diverse range of QoS requirements in IoT networks.

REFERENCES

[1] S. Kumar, P. Tiwari, and M. Zymbler, “Internet of things is a revolutionary approach for future technology enhancement: A review - journal of big data,” Dec 2019.

[2] S. Sun, K. Adachi, P. H. Tan, Y. Zhou, J. Joung, and C. K. Ho, “Heterogeneous network: An evolutionary path to 5g,” in *2015 21st Asia-Pacific Conference on Communications (APCC)*, pp. 174–178, 2015.

[3] M. Haghi Kashani, M. Madanipour, M. Nikravan, P. Asghari, and E. Mahdipour, “A systematic review of iot in healthcare: Applications, techniques, and trends,” *Journal of Network and Computer Applications*, vol. 192, p. 103164, 2021.

[4] F. T. Johnsen, Z. Zieliński, K. Wrona, N. Suri, C. Fuchs, M. Pradhan, J. Furtak, B. Vasilache, V. Pellegrini, M. Dyk, M. Marks, and M. Krzysztoń, “Application of iot in military operations in a smart city,” in *2018 International Conference on Military Communications and Information Systems (ICMCIS)*, pp. 1–8, 2018.

[5] T. Szigeti, J. Henry, and F. Baker, “Mapping Diffserv to IEEE 802.11.” RFC 8325, Feb. 2018.

[6] A. Arulappan and J. J, “Performance analysis of wlan under variable number of nodes using the adjustable parameters in edca,” *Journal of Theoretical and Applied Information Technology*, vol. 62, pp. 1–7, 04 2014.

[7] B. Finley, E. Boz, K. Kilkki, J. Manner, A. Oulasvirta, and H. Hämmäinen, “Does network quality matter? a field study of mobile user satisfaction,” *Pervasive and Mobile Computing*, vol. 39, pp. 80–99, 09 2016.

[8] P. Serrano, A. Banchs, P. Patras, and A. Azcorra, “Optimal configuration of 802.11e edca for real-time and data traffic,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2511–2528, 2010.

[9] G. O. Ugwu, U. N. Nwawelu, M. A. Ahaneku, and C. I. Ani, “Effect of service differentiation on qos in ieee 802.11 e enhanced distributed channel access: a simulation approach,” *Journal of Engineering and Applied Science*, vol. 69, no. 1, pp. 1–18, 2022.

[10] E. Coronado, S. N. Khan, and R. Riggio, “5g-empower: A software-defined networking platform for 5g radio access networks,” *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 715–728, 2019.

[11] M. Richart, J. Baliosian, J. Serrat, and J.-L. Gorricho, “Resource slicing in virtual wireless networks: A survey,” *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 462–476, 2016.

[12] M. Richart, J. Baliosian, J. Serrati, J.-L. Gorricho, R. Agüero, and N. Agoulmine, “Resource allocation for network slicing in wifi access points,” in *2017 13th International Conference on Network and Service Management (CNSM)*, pp. 1–4, 2017.

[13] M. Richart, J. Baliosian, J. Serrat, and J.-L. Gorricho, “Resource allocation and management techniques for network slicing in wifi networks,” in *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–6, 2020.

[14] A. Banchs, P. Serrano, P. Patras, and M. Natkaniec, “Providing throughput and fairness guarantees in virtualized wlans through control theory,” *Mobile Networks and Applications*, vol. 17, no. 4, pp. 435–446, 2012.

[15] K. Nakauchi, Y. Shoji, and N. Nishinaga, “Airtime-based resource control in wireless lans for wireless network virtualization,” in *2012 Fourth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 166–169, 2012.

[16] K. Guo, S. Sanadhya, and T. Woo, “Vifi: virtualizing wlan using commodity hardware,” in *Proceedings of the 9th ACM workshop on Mobility in the evolving internet architecture*, pp. 25–30, 2014.

[17] E. Coronado, R. Riggio, J. Villa1ón, and A. Garrido, “Lasagna: Programming abstractions for end-to-end slicing in software-defined wlans,” in *2018 IEEE 19th International Symposium on” A World of Wireless, Mobile and Multimedia Networks”(WoWMoM)*, pp. 14–15, IEEE, 2018.

[18] B. T. Vijay and B. Malarkodi, “A study of ieee 802.11aa,” in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–6, 2017.