Guaranteeing critical communications by using fuzzy logic based intelligent control systems in Wi-Fi Networks

Xhulio Limani*, Gilson Miranda Jr.*, Carlos F. M. e Silva†,
Xiaoman Shen‡, Chun Pan‡, Xingfeng Jiang‡, Chi Zhang‡, Johann M. Marquez-Barja*
*{xhulio.limani, gilson.miranda, johann.marquez-barja}@uantwerpen.be
‡{shenxiaoman, panchun, jiangxingfeng, zhangchi157}@huawei.com
†carlosfmesilva@gmail.com
*University of Antwerp, IDLab, imec, Antwerp, Belgium
‡Huawei Technologies Co., Ltd., P.R China

***Abstract*—In the face of intensifying global challenges, including climate change, geopolitical turbulence, and both natural and anthropogenic disasters, the urgency of effective disaster management has been brought to the forefront. Risk reduction, preparedness for response, and recovery is of crucial importance in this context. Communication technologies play an essential role in disaster management, and among the technologies used, Wi-Fi networks play a significant part. When traditional communication systems are overloaded or break down during a disaster scenario, Wi-Fi networks can provide an alternative means of information dissemination and coordination for response teams, as well as communication for victims affected. In this paper, we introduce a novel slicing algorithm that, using a logic similar to human thinking, aims to dynamically optimize the resource allocation over Wi-Fi networks for critical services. By exploiting In-band Network Telemetry (INT) techniques to monitor the network, our work aims to dynamically prioritize certain critical services by applying network slicing techniques in real time even under difficult network conditions, thus improving the role of Wi-Fi networks in disaster management situations.**

***Index Terms*—Network Slicing, Airtime, Wi-Fi, Testbed, In-band Network Telemetry, Fuzzy Logic.**

I. Introduction

Global crises challenges such as climate change, geopolitical unrest, natural and human-caused disasters are becoming an increasingly important concern. Effective disaster management, including risk reduction, response preparation, and recovery, has never been more critical [1]. Communication technologies play an essential role in disaster management [2], and among such technologies, Wi-Fi networks can play a significant part. When traditional communication systems are overloaded or break down during a disaster scenario, Wi-Fi networks can provide an alternative means of information dissemination and coordination for response teams, as well as communication for affected victims. During disaster scenarios, communication networks are often subject to significant stress that can be caused by increased demand [2], infrastructure damage, or power outages. When multiple critical services, such as real-time voice communications (VoIP), Programmable Logic Controllers (PLCs) for remote control, and streaming video for information dissemination, compete for the same network resources as in Figure 1, the situation further deteriorates. Preserving the Quality of Service (QoS) for critical services under these conditions is critical, yet challenging.

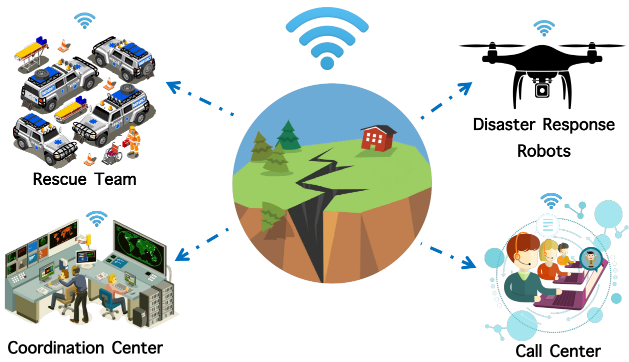

Fig. 1. Disaster Scenario

The problem of network overload due to the lack of resources and consequent degradation of services during crisis scenarios is a critical limitation for the exploitation of Wi-Fi networks as an effective aid for disaster management. However, the behavior of Wi-Fi networks, particularly under stress conditions, is complex and difficult to model from a mathematical perspective. This complexity arises from the multitude of variables and parameters that influence the performance of the network. Furthermore, due to the random access nature of Wi-Fi networks, they are inherently non-deterministic, adding another layer of difficulty in prediction and modeling. Due to this complexity, it is not possible to control specific variables to change the output of the network in a predictable manner, as there is no precise mathematical model.

The use of mechanisms that can efficiently manage network resources is crucial to ensure that critical services receive the necessary network resources to properly function, without compromising the overall network stability. In this paper, we introduce a slicing algorithm that is designed to dynamically prioritize critical services over a Wi-Fi network using a similar logic to human thinking. Unlike the binary *"true"* or *"false"* of Boolean logic, the algorithm mirrors human logic by interpreting variables in a spectrum, such as *"completely true"* to *"completely false"*, including elements such as *"partially *true"* or *"partially false"*. This nuanced approach allows for decisions that better reflect real-world complexity, enhancing network resource management.

By utilizing In-band Network Telemetry (INT) techniques [3] for network monitoring and to provide real-time feedback, our work focuses on dynamically allocating the appropriate amount of *airtime* to different services. *Airtime* refers to the amount of time a Wi-Fi network dedicates to transmitting data from a certain service. Our goal is to uphold the QoS required by certain critical services, even under strenuous network conditions, thus improving the role of Wi-Fi networks in disaster management situations.

Recent studies [4]–[12] have proposed various solutions for Wi-Fi network slicing, such as dynamic resource allocation mechanisms [4], [5], Mixed Integer Linear Programming optimization [6], [8], and Software-Defined Radio Access Network (SD-RAN) management approaches [7], [12]. However, most of these studies do not delve into practical implementations with real hardware, thus eluding intrinsic physical phenomena and engineering challenges, such as modifications to the packet scheduler of a Wi-Fi AP. Additionally, the inclusion of telemetry options, particularly INT, remains sparse with a notable exception being [12]. Furthermore, the frameworks used on the real testbed works do not support the latest Wi-Fi standards (Wi-Fi 5 and 6). Our work addresses these gaps, providing a comprehensive solution compatible with current Wi-Fi standards and validated in real-world scenarios.

The remainder of the paper is organized as follows: Section II provides a detailed discussion on the relevance of network slicing for Wi-Fi networks, Section III discusses the metrics and benchmarks employed to evaluate the performance of the proposed solution. Section IV presents the design and implementation details of the slicing algorithm. Section V describes our experimental setup and the results obtained. Finally, conclusions are drawn in Section VI.

II. Slicing on Wi-Fi Networks

Network slicing represents a revolutionary approach that facilitates the creation of multiple virtual networks within a common physical infrastructure. Each resulting virtual network is referred to as a slice [13]. Each individual slice can be precisely calibrated to meet a unique set of bandwidth, latency, and reliability requirements providing a versatile solution capable of handling a wide range of data traffic. For example, a single network slice can be properly tuned to provide low-latency connections, ideal for real-time applications such as emergency alert systems or remote control of rescue robots when it may be too dangerous for humans to enter certain areas. Conversely, another slice can be adjusted to meet high bandwidth requirements, essential for real-time video streaming that can be crucial for situational awareness.

By leveraging slicing for Wi-Fi networks, significant efficiency improvements can be achieved, especially when using Software-Defined Network (SDN) techniques to centrally and dynamically manage the allocation of resources for network services according to their priorities [14]. Hence, INT serves as a pivotal element for real-time network monitoring, allowing the system to swiftly adapt to variations in demand, the introduction of new services, or changes in user behavior. Therefore, by integrating slicing, SDN, and INT we can significantly improve the allocation of resources to the Wi-Fi network making it more robust and reliable for disaster management applications.

A. The Resource: Airtime

Since Wi-Fi networks operate in a shared medium, multiple devices will compete to access the wireless channel to transmit and receive data. A fair distribution of communication opportunities, known as *airtime*, among connected devices is crucial for the efficient functioning of the network [15].

Airtime refers to the limited amount of time available for data transmission within a wireless network, i.e., the time over the air. Ideally, a certain amount of airtime should be equitably distributed among all connected devices to ensure balanced network performance. Airtime is a valuable resource because it significantly affects the overall capacity and performance of the Wi-Fi network.

B. Software-Defined Network Controller

In this work, we used an SDN platform that is based on an architecture previously developed for Time-Sensitive Networks (TSNs) and can be used to configure general-purpose networks [16]. The architecture adopts a controller/agent approach, with the Time-Sensitive Network Controller (TSNC) serving as the centralized entity, while the Time-Sensitive Network Agent (TSNA) is implemented on Network Elements (NEs) that require fine-grained control.

The control plane uses IPv4 addresses only, while the data plane is based on IPv6. This design choice allows the use of IPv6 extended headers to implement INT, which enables detailed per-hop and per-flow monitoring. Moreover, this also provides a clear distinction between both control and data planes. Figure 2 illustrates the components of the TSNC that are relevant to this study, with a particular focus on the elements that enable the network programmability. The main communication between TSNC and TSNA is done through the Centralized Network Controller (CNC) module. The internal interface facilitates communication between the various modules, serving as a central point of interaction. This interface allows modules to query topology information, link states, and apply configurations to NEs based on their User Identifications (UIDs). Data plane telemetry reports are sent directly to the Subscriber socket of the Monitor module.

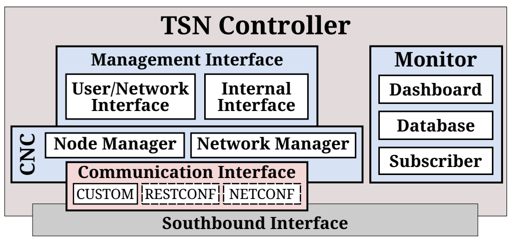

Fig. 2. TSNC architecture

Figure 3 shows the TSNA components for an AP. The key component of TSNAs that enables network slicing is the Click framework [17], a software-based modular router designed to provide flexibility and modularity. Other components of TSNA include the INT Manager, which manages the In-band Network Telemetry framework. This framework adds timestamps and packet counters as extended headers on IPv6 data packets and generates periodic reports with this data. Based on these reports, the Monitor module of the TSNC can calculate QoS metrics of throughput, delay, jitter, and Packet Loss Ratio (PLR).

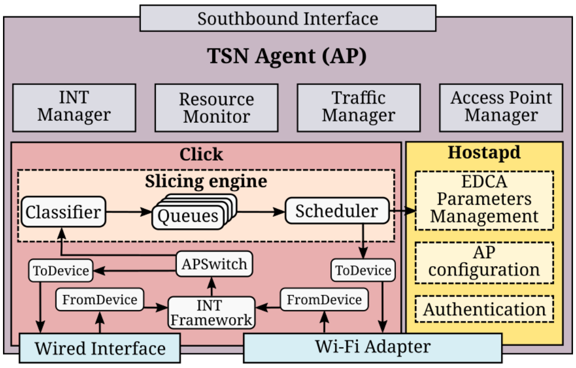

Fig. 3. TSNA architecture

The Resource Monitor is responsible for continuously monitoring the status of the NE in terms of services, network interfaces, hardware, and topology changes. During TSNA startup, the Resource Monitor analyzes the capabilities of the node and communicates them to the TSNC through an announcement message. Finally, the Traffic Manager handles traffic classification, filtering, and shaping. This module processes the rules received from the TSNC to label packets with specific Differentiated Services CodePoint (DSCP) values based on the 5-tuple that defines a flow: source IP, destination IP, Source Port, Destination Port, and Protocol. Depending on the DSCP value, packets in the flow are routed to different queues within the Click framework and handled by the scheduling/slicing algorithm.

C. *In-band Network Telemetry (INT)*

Data plane telemetry is obtained via INT and implemented on Click. The TSNA continuously monitors telemetry information and publishes the the Monitor element of the TSNC. The framework described by Haxhibeqiri et al. [18] has been adapted to add packet counters (packets transmitted/received), byte counters (bytes transmitted/received), and timing information (timestamps at each hop) as extended IPv6 packet headers. Telemetry information is collected for each flow identified as a 5-tuple: source address, destination address, source port, destination port, and transport protocol.

Figure 4 provides an overview of our INT mechanism. The controller sends an INT configuration to the INTSource element of an NE, informing the 5-tuple of the flow to be monitored. The INTSource intercepts each packet leaving the NE and adds the INT header with packet counters and timestamps if the 5-tuple is from a flow being monitored. The INTInter at the AP also checks the packets and appends telemetry data. This happens at each hop until the packet with all the telemetry reaches the INTSink at the destination. The INTSink extracts the telemetry data, forwards the original data packet to higher network layers, and transmits a report through the control plane connection to the controller.

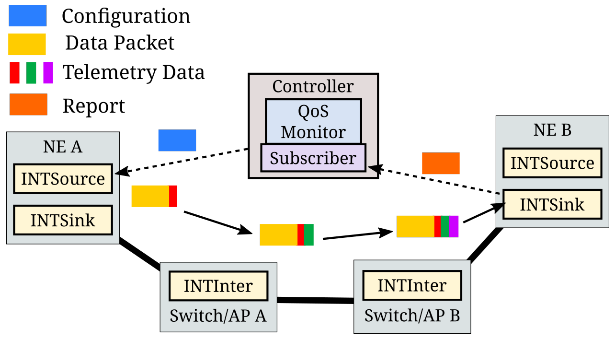

Fig. 4. INT report mechanism

In the controller, the QoS Monitor collects a sequence of packets and calculates throughput, delay, jitter, and PLR metrics. Throughput is calculated based on bytes transmitted in a given time period, while delay and jitter are calculated by comparing the timestamps of when the packet was generated and delivered to its final destination. The PLR is calculated from the difference between transmitted and received packets in a period.

III. Benchmark

Queuing plays a pivotal role in Wi-Fi networks as it manages the distribution of network resources among various data flows, ensuring that no single flow monopolizes the network. This is particularly crucial when different services have varying priorities and QoS requirements. In this work, we evaluate the network behavior when using FQ-CoDel (Fair Queuing and Controlled Delay) [19] as a baseline, as it is the default queuing discipline (qdisc) used by our Linux-based Wi-Fi AP. FQ-CoDel is a combination of two algorithms: Fair Queuing (FQ) and Controlled Delay (CoDel). FQ sorts the outgoing traffic into different flows trying to prevent any single data flow from monopolizing the network. Each of these flows essentially gets its own First-In-First-Out (FIFO) queue and then is managed by CoDel. CoDel monitors the amount of time that packets spend in the queue. If this so-called "sojourn time" exceeds a certain threshold, indicating potential congestion, CoDel drops packets from the front of the queue. This division allows network resources to be equally distributed but does not take into account the different priorities of the services and the QoS requirements.

A. *Baseline Experiment*

To establish a performance baseline, we conducted an experiment with a specific setup: two client nodes, an AP, and a server, as illustrated in Figure 5. We utilized a traffic generator with diverse traffic patterns. We considered downlink traffic, from the server to the client nodes, to capture the behavior of the Wi-Fi network. These patterns, as detailed in Table I, included: i) a VoIP service over UDP with a data rate of 20 Mbps to Client 1; ii) a Messaging service over TCP with a data rate of 4 Mbps also to Client 1; and iii) a Video service over UDP with a data rate of 40 Mbps to Client 2. All flows originate from the server node.

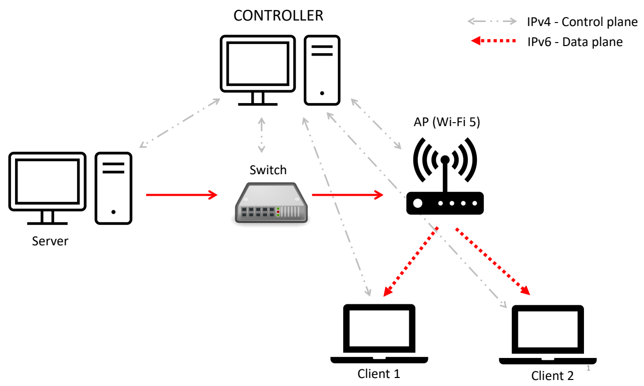

Fig. 5. Testbed Architecture

TABLE I
TRAFFIC PATTERN

Service	Destination	Protocol	Data Rate
VoIP	Client 1	UDP	20 Mbps
Messaging	Client 1	TCP	4 Mbps
Video	Client 2	UDP	40 Mbps

The graph in Figure 6 shows that under normal network conditions (prior to saturation with background traffic), the traffic traverses the network steadily. However, a noticeable deterioration in the performance of different services is observed once the background traffic starts. This degradation can be attributed to the FQ-CoDel scheduling algorithm, which does not take into account the required QoS and priority of the different services, thus allocating an uneven share of airtime to the different flows. Therefore, it is necessary to adopt more effective scheduling methods to perform resource slicing and allocate the necessary resources for the different network services, according to their priorities.

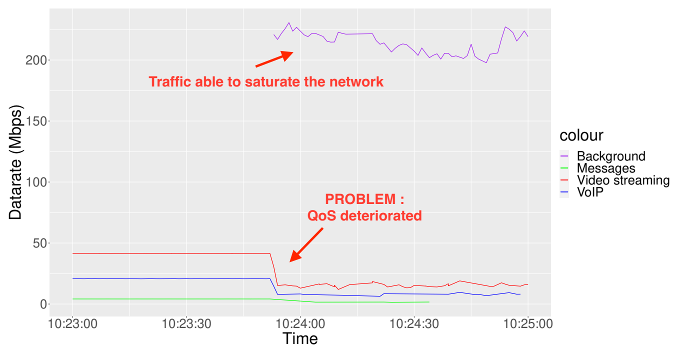

Fig. 6. Network behaviour without network slicing

IV. Fuzzy Logic for Airtime Slicing

Fuzzy Logic is able to manage uncertainties and non-linearity inherent in complex systems such as telecommunication networks. Leveraging fuzzy linguistic variables and rules, this logic can interpret and manipulate imprecise and vague data, thus offering a considerable advantage over traditional control systems. The Fuzzy-based slicing algorithm in this work plays a critical role in the decision-making for airtime allocation. It is responsible for creating, modifying, and efficiently managing network resources in order to adapt the airtime allocation under varying network conditions dynamically.

This algorithm is designed as an extension of the Network Manager submodule, located within the CNC from Figure 2. The structure of our algorithm, depicted in Figure 7, employs a systematic sequence of steps to control the allocation of airtime across the network. The algorithm is composed six main steps: i) configuration; ii) network statistics collection; iii) network evaluation; iv) determining network status; v) Decision Maker; and vi) Schedule Update. Figure 8 illustrates the process flow of the algorithm and the control elements involved.

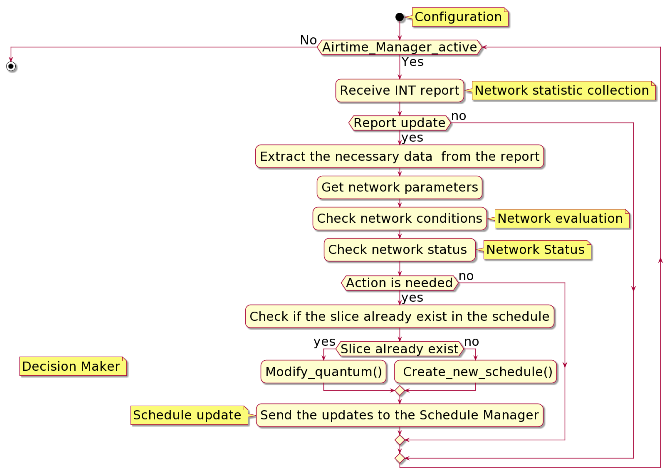

Fig. 7. Slicing Algorithm

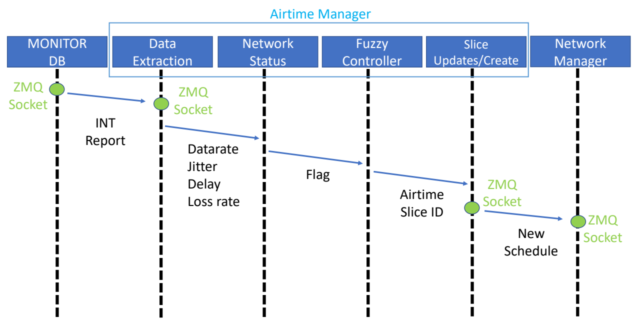

Fig. 8. Process flow

The first phase is the configuration of the ZeroMQ [20] sockets for communication with other components of the controller. The algorithm starts in phase (ii) by collecting the most recent INT reports from the database located on the TSNC. The INT reports are decoded by extracting information such as network service id, message date and time, and performance metrics of data rate, delay, jitter, and packet loss ratio. Using comparative analysis, if the target QoS metrics are not met, a network slice for the service is configured for the service. The algorithm continuously assesses if the QoS values for the service align with the expected values, and generates a flag value that indicates whether the allocation of resources for a service must be modified. In practice, it indicates if the service is sufficiently served with the allocated airtime, or if it operates with an excess of resources. In cases where a surplus exists, the extra airtime can be reallocated to other slices.

The Fuzzy Controller, detailed in Figure 9, employs four distinct fuzzy sets representing various network parameters: data rate, delay, jitter, and airtime. Each of these parameters is linguistically modeled using triangular and trapezoidal membership functions, enabling the controller to reason in a human-like manner.

The membership functions [21] used in our work are triangular and trapezoidal. These functions map the crisp inputs (exact numerical values) to fuzzy sets which represent vague concepts such as 'low', 'high', 'acceptable', 'very low', and 'very high', providing a control strategy more adaptable to human thinking and able to handle the ambiguities inherent in these parameters. This controller design allows for the customization of a control strategy based on the slice type, demonstrating its adaptability. Different membership functions are set to map input variables depending on the slice type.

For instance, in the scenario where the network slice corresponds to a network service targeted as QoS, such as video streaming or VoIP, the variables to be regulated are defined through certain membership functions. Those functions differ from a service targetted as Best-effort (BE), e.g., messaging, which utilizes a distinct set of membership functions. This part of the process highlights the adaptability of the algorithm to ensure efficient resource allocation and optimal network performance. In the end, the results of these processes are sent to the CNC to apply the necessary updates.

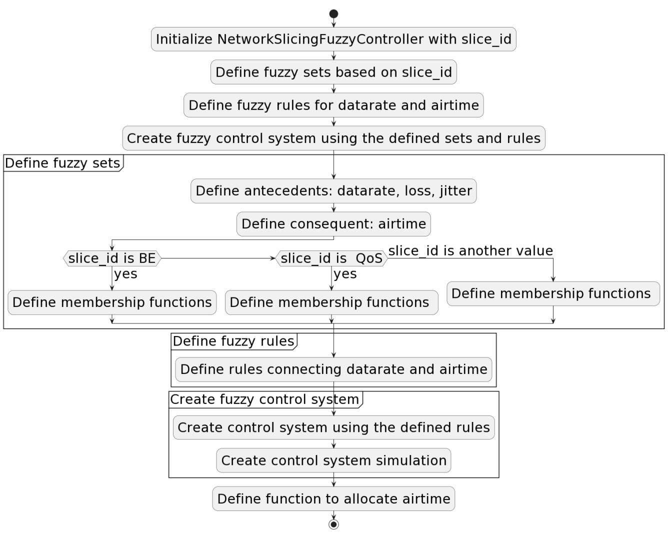

Fig. 9. Fuzzy controller algorithm

V. Experimentation

In this section, we describe how we conducted our experiments to validate our work and the effectiveness of the algorithm. We first discuss the network topology, then the hypothesized scenario, and finally, the parameters used as QoS for evaluating the network. The experiment was conducted in a real testbed by reproducing the topology of a typical Wi-Fi network, shown in Figure 10.

A. Testbed

Our testbed consists of 5 nodes and a controller. Three nodes serve as endpoints - two clients and a server, while two nodes function as bridge, switch, and AP. The two clients are connected to the same AP with Wi-Fi 5 (IEEE 802.11ac) features enabled. The AP is connected to a network switch, creating a path for data transmission and reception with other networks. The switch is connected to a wired node (Server) through a reliable Ethernet connection. All nodes have a logical connection to a central controller. In a realistic deployment, this connection is not required for Wi-Fi clients. The traffic flows originate at the server and traverse the network to reach the clients.

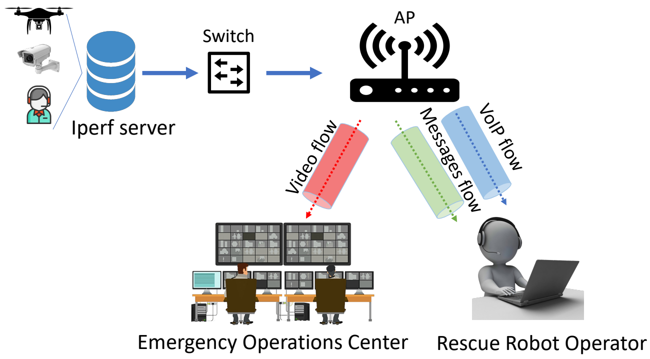

Fig. 10. Testbed scenario

In Figure 10 we hypothesized a scenario following a catastrophic earthquake, where the primary communication infrastructure is out of order, so the emergency services rely on a single Wi-Fi AP that serves two main clients: the Emergency Operations Center for video streaming of disaster-stricken areas, the field emergency response team using VoIP for real-time communication, and has control of the rescue robots requiring low latency for real-time navigation. The quality of critical services must be maintained for effective disaster management despite high background traffic.

To emulate different network services, we used Iperf[1], known for its ability to generate network traffic with specified characteristics. To replicate the characteristics of specific services, we configured Iperf according to the parameters reported in Table I.

B. Evaluation

Within the adopted architecture, our goal is to evaluate the performance and effectiveness of our algorithm in an environment reflecting realistic network conditions. The presence of multiple clients requires the division of AP resources, introducing a crucial level of complexity into network operations.

We evaluated our algorithm by benchmarking it against the FQ-CoDel algorithm, the default algorithm in Linux-based

[1] https://www.iperf.fr/

APs. The rationale for selecting FQ-CoDel as a comparison point is its widespread adoption in operational systems, a significant number of which are Linux-based. Such popularity makes FQ-CoDel a pertinent and practical standard for comparison. We conducted the tests under conditions of network saturation, to push the capabilities of our algorithm to their limits providing valuable insights into their performance and adaptability in situations of high demand, mirroring real network conditions.

In Figure 11 we present the behavior of the traffic within the Wi-Fi network. We compare the two different scheduling algorithms, FQ-CoDel and Fuzzy, applying them consecutively, to observe any disparities in network behavior, which can be attributed to the different scheduling algorithms. Ideally, different types of services labeled as background, messages, video streaming, and VoIP have different priority levels in disaster scenarios.

In Figure 11 from $t1$ to $t2$, FQ-CoDel is the algorithm driving the network traffic scheduling. From $t1$ to $t2$, we can notice high traffic in terms of data rate, referred to as background service. However, the other three types of services (messages, video streaming, and VoIP) have a moderate data rate.

The key moment comes at time $t2$ when we change the traffic scheduling algorithm to Fuzzy. Because of this, a sharp change in the behavior of each service at time $t2$ can be seen in terms of data rate. This change essentially occurs because the Fuzzy algorithm creates slices on the Wi-Fi network, assigning more airtime, to higher-priority services and fewer resources to lower-priority services. The increase in data rate starting from time $t2$ for messaging, video streaming, and VoIP services highlights how the network performance for such flows was affected by the background traffic. The background service demands more network resources than the other services, and such resources were granted by the FQ-CoDel algorithm. However, these are not granted after $t2$ by Fuzzy.

Subsequently, we examined the difference in network performance in terms of data rate, jitter, delay, and packet loss rate. We compared the Cumulative Distribution Function (CDF) of each metric for both scheduling algorithms, as shown in Figures 12-15. In addition, Table II shows the 99th percentiles of the measurements and the gains against FQ-CoDel.

The CDF of the data rate in Figure 12 shows how all curves with a solid line are shifted to the right compared to the dashed lines, demonstrating how the data rate of the various services is higher by applying our algorithm, compared to FQ-CoDel. More precisely, on average there is a data rate gain of more than 100% for the video and VoIP slices, and 44% for the Messaging service when we switch from FQ-CoDel to Fuzzy, as indicated in Table II.

For the delay and jitter, the CDFs shown in Figures 13 and Figure 14, highlight the improvement of the metrics with over 94 % of gains across all services. This confirms the reliability gains offered by the algorithm in dealing with critical real-time services. Finally, the CDF of packet loss ratio in Figure 15 shows a significant reduction from using the Fuzzy algorithm, reducing losses by 73% on average compared to the FQ-CoDel algorithm, meaning that crucial data can be transmitted more reliably.

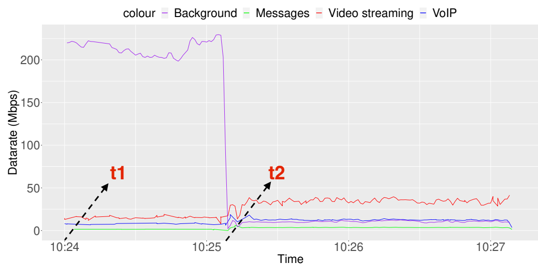

Fig. 11. Network behavior with network slicing

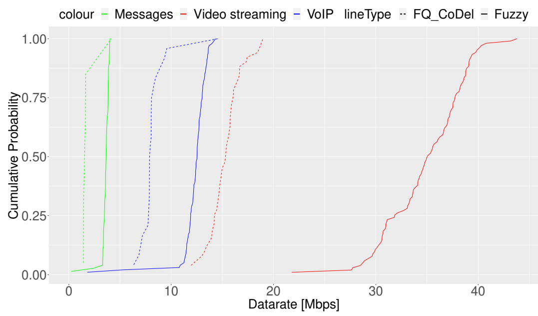

Fig. 12. Comparison of the CDF of data rate considering both algorithms

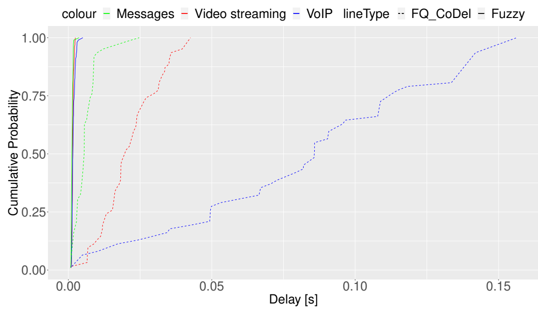

Fig. 13. Comparison of the CDF of delay considering both algorithms

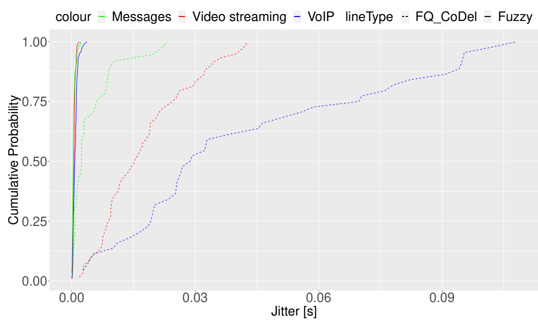

Fig. 14. Comparison of the CDF of jitter considering both algorithms

VI. Conclusion

Our study presents an approach for supporting communications under disaster management scenarios using Wi-Fi

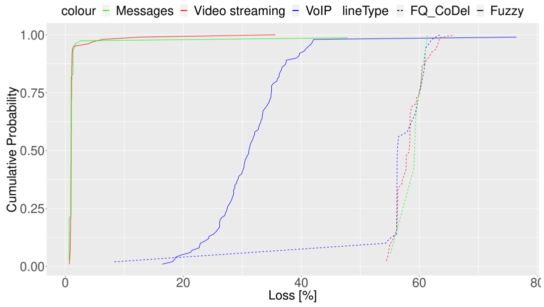

Fig. 15. Comparison of the CDF of packet loss considering both algorithms

TABLE II
PERFORMANCE GAINS - 99TH PERCENTILE

Metric	Algorithm	Video S.	Messages	VoIP
Datarate Gain %	FQ-CoDel - Fuzzy	127.27	44.33	122.22
Delay Gain %	FQ-CoDel - Fuzzy	98.47	97.39	94.17
Jitter Gain %	FQ-CoDel - Fuzzy	97.18	97.61	96.64
Loss Gain %	FQ-sCoDel - Fuzzy	87	58	99.1

networks. We apply network slicing to Wi-Fi networks to effectively allocate network resources dynamically and customized to critical services, ensuring their correct performance even when the network is saturated. The mechanism we created exploits INT techniques and Fuzzy logic to handle the uncertainties and nonlinearities present in complex telecommunication networks.

The systematic sequence of steps of the algorithm checks the allocation of airtime on the network, ensuring optimal allocation of network resources and performance. Through our experiments, we validated the reliability and effectiveness of our work. We hypothesized a disaster scenario using real equipment, demonstrating that our approach significantly improves the performance of critical services in Wi-Fi networks even under saturation conditions.

In future research, we aim to extend our work within the context of Wi-Fi 6. We plan to evaluate the impact of INT by studying channel congestion through a detailed analysis. Additionally, we intend to enhance our algorithm by integrating the Adaptive Neuro-Fuzzy Inference System (ANFIS). This integration will enable the algorithm to autonomously adapt to changing conditions. The ANFIS approach, which combines the ability to handle uncertainty and imprecision inherent in fuzzy logic with the capacity to learn from data, presents a potent methodology for the modeling and control of complex systems.

ACKNOWLEDGMENTS

This work is supported by the Flemish Government through FWO SBO project MOZAIK S003321N.

REFERENCES

[1] H. Ouyang, X. Tang, R. Zhang, *et al.*, "Resilience building and collaborative governance for climate change adaptation in response to a new state of more frequent and intense extreme weather events," *International Journal of Disaster Risk Science*, vol. 14, pp. 162–169, 2023. DOI: 10.1007/s13753-023-00467-0.

[2] K. Ali, H. X. Nguyen, Q.-T. Vien, and P. Shah, "Disaster management communication networks: Challenges and architecture design," in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, 2015, pp. 537–542. DOI: 10.1109/PERCOMW.2015.7134094.

[3] L. Tan, W. Su, W. Zhang, *et al.*, "In-band network telemetry: A survey," *Computer Networks*, vol. 186, p. 107 763, 2021, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2020.107763.

[4] 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*, 2020, pp. 1–6. DOI: 10.1109/NOMS47738.2020.9110407.

[5] M. Richart, J. Baliosian, J. Serrat, J.-L. Gorricho, and R. Agüero, "Slicing with guaranteed quality of service in wifi networks," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1822–1837, 2020. DOI: 10.1109/TNSM.2020.3005594.

[6] F. Fami, N. Hammami, C. Pham, and K.-K. Nguyen, "Slicing wi-fi networks for differentiated iot service provisioning," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 2022, pp. 866–871. DOI: 10.1109/WCNC51071.2022.9771816.

[7] S. H. Amur, K. Zia, A. Chiumento, and P. Havinga, "Autonomous network slicing and resource management for diverse qos in iot networks," in *2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2023, pp. 160–165. DOI: 10.1109/PerComWorkshops56833.2023.10150306.

[8] F. Fami, C. Pham, and K.-K. Nguyen, "Towards iot slicing for centralized wlans in enterprise networks," in *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, 2020, pp. 1–6. DOI: 10.1109/ISNCC49221.2020.9297339.

[9] T. Høiland-Jørgensen, P. Hurtig, and A. Brunstrom, "Polifi: Airtime policy enforcement for wifi," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019, pp. 1–6. DOI: 10.1109/WCNC.2019.8885440.

[10] M. Nerini and D. Palma, "5g network slicing for wi-fi networks," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2021, pp. 633–637.

[11] P. H. Isolani, N. Cardona, C. Donato, *et al.*, "Airtime-based resource allocation modeling for network slicing in ieee 802.11 rans," *IEEE Communications Letters*, vol. 24, no. 5, pp. 1077–1080, 2020. DOI: 10.1109/LCOMM.2020.2977906.

[12] P. Isolani, J. Haxhibeqiri, I. Moerman, *et al.*, "Sd-ran interactive management using in-band network telemetry in ieee 802.11 networks," *J Netw Syst Manage*, vol. 31, no. 11, 2023. DOI: 10.1007/s10922-022-09692-2.

[13] S. Wijethilaka and M. Liyanage, "Survey on network slicing for internet of things realization in 5g networks," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 957–994, 2021. DOI: 10.1109/COMST.2021.3067807.

[14] S. Singh and R. Jha, "A survey on software defined networking: Architecture for next generation network," *Journal of Network and Systems Management*, vol. 25, pp. 321–374, 2017. DOI: 10.1007/s10922-016-9393-9.

[15] M. Richart, J. Baliosian, J. Serrat, J.-L. Gorricho, and R. Agüero, "Slicing in wifi networks through airtime-based resource allocation," *J. Netw. Syst. Manage.*, vol. 27, no. 3, 2019, ISSN: 1064-7570. DOI: 10.1007/s10922-018-9484-x.

[16] G. Miranda, E. Municio, J. Haxhibeqiri, J. Hoebeke, I. Moerman, and J. M. Marquez-Barja, "Enabling time-sensitive network management over multi-domain wired/wi-fi networks," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2023. DOI: 10.1109/TNSM.2023.3274590.

[17] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," in *ACM Transactions on Computer Systems (TOCS)*, ACM, vol. 18, 2000, pp. 263–297.

[18] J. Haxhibeqiri, P. H. Isolani, J. M. Marquez-Barja, I. Moerman, and J. Hoebeke, "In-band network monitoring technique to support sdn-based wireless networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 627–641, 2021. DOI: 10.1109/TNSM.2020.3044415.

[19] P. E. McKenney, J. Morton, and D. Täht, *The flow queue codel packet scheduler and active queue management algorithm*, Internet Requests for Comments, RFC, Jan. 2018. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc8290.

[20] ZeroMQ contributors, *Zeromq*, Web page at https://zeromq.org/get-started/, 2023.

[21] Y. Dote, "Introduction to fuzzy logic," in *Proceedings of IECON '95 - 21st Annual Conference on IEEE Industrial Electronics*, vol. 1, 1995, 50–56 vol.1. DOI: 10.1109/IECON.1995.483332.