Random Access Sliced 5G-Advanced Network for Machine Type Communication

Paul Orim
dept. Electrical Engineering
University of Cape Town
Cape Town, South Africa
ormpau001@myuct.ac.za

Joyce Mwangama
dept. Electrical Engineering
University of Cape Town
Cape Town, South Africa
joyce.mwangama@uct.ac.za

***Abstract*—Massive machine-type communication (mMTC) has been highlighted as one of the major services that enable the development of the Internet of Things (IoT) paradigm with varying Quality of Service (QoS) requirements for 5G and beyond networks. However, it is extremely challenging to use a monolithic physical network to handle several mMTC applications with varying QoS needs due to the limitations experienced during the random access (RA) procedure, which leads to collisions and network overload. While 5G-Advanced promises better network performance and more connections through the introduction of machine learning, it also requires more access resources for the RA process. The introduction of network slicing to the access network can ameliorate these issues, leading to improvements in collision resolution and network congestion management. Consequently, to tackle the afore-mentioned challenges, we propose a network-slicing random access scheme (NSRAS) that combines network slicing and machine learning algorithms to dynamically vary the physical random access resources into multiple virtual resources to reduce collisions during the RA procedure while meeting the QoS requirements of the different types of machine-type communication devices (MTCDs). Our proposed scheme shows improvements in the average access delay and the outage proportion for MTCDs.**

***Index Terms*—random access, network slicing, massive access, power ramping, Q-learning.**

I. Introduction

The implementation of fifth-generation (5G) and beyond networks requires the satisfaction of the internet of things (IoT) paradigm. This is due to the IoT's reliance on machine-type communication (MTC) technology to link the countless devices and sensors that will comprise the communication ecosystem. Networks in the 5G and beyond era will require much larger capacity and varying latency than existing networks in order to serve the massive influx of machine-type communication devices (MTCDs). 5G is anticipated to reach its fullest and most comprehensive development stage through the implementation of 5G-Advanced between 2025 and 2030. This will lay the groundwork for additional use cases and more demanding applications than were previously possible, thereby bringing improvements in artificial intelligence (AI) and machine learning (ML) to network management and radio access network (RAN) layers for better efficiency and network performance [1]. Furthermore, these networks should be capable of handling massive volumes of tiny-sized data created by MTCDs while also ensuring that their quality of service (QoS) requirements are satisfied. These data packets are mostly uplink-dominated, tiny-sized data that could be transmitted during the synchronization and resource request scheduling known as the random access (RA) procedure. According to a forecast by Ericsson, the number of MTCDs that will be connected to cellular networks will approach 5.5 billion by 2028 [2]. Therefore, the attempt by a massive number of MTCDs with diverse QoS requirements to access these networks will lead to collisions and network overload in the physical random access channel (PRACH).

To mitigate these problems, 5G-Advanced suggests introducing machine learning and other 5G network features, like network slicing on the RAN layers, to improve the access network and meet the QoS requirements of different devices. Machine learning may aid in the optimization of network access resources (preambles) and the dynamic adjustment of network parameters based on certain traffic patterns. Furthermore, network slicing enables the establishment of virtualized networks that cater to certain devices or applications, resulting in improved QoS and tailored services. A single physical network is partitioned into multiple slices to accomplish the three important features of network slicing, namely, customization, isolation, and scalability. 5G-Advanced intends to deliver enhanced connections and enable an increasingly wide variety of devices or applications in the future by adding these advanced characteristics. The utilization of these advanced characteristics to solve the problem of collision and network overload in the PRACH, especially as the number of MTCDs becomes massive, is crucial for maintaining efficient and reliable communication in the future. By employing sophisticated algorithms and adaptive transmission techniques, the PRACH can effectively manage the increasing number of devices accessing the network simultaneously.

Therefore, we proposed a network-slicing random access scheme (NSRAS) that combines network slicing with Q-learning, a reinforcement learning strategy that requires no external assistance with the learning process, to solve the problem of collision and network overload during the RA procedure. By integrating network slicing with Q-learning, we can increase access resources (preambles) in a dynamic and efficient manner. The Q-learning algorithm serves as the orchestrator that learns from past experiences and makes

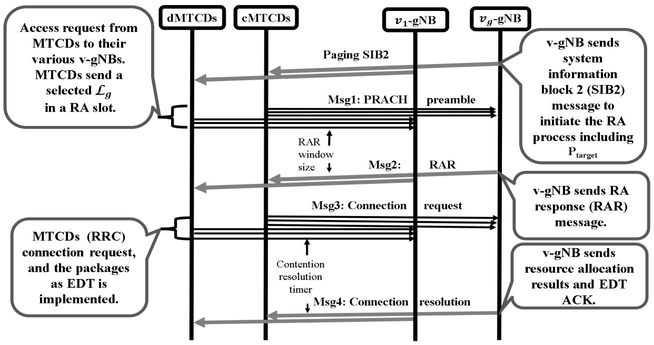

Fig. 1. Illustration of a Random Access Network in the physical layer

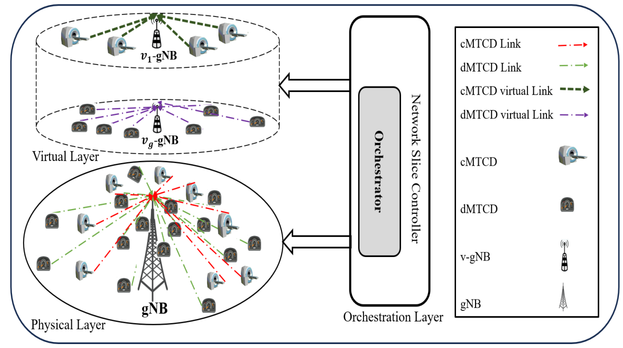

Fig. 2. 5G-Advanced RAN Network Slice framework for mMTC

informed decisions on how to allocate access resources effectively. This approach eliminates the need for external assistance, making it a self-adaptive solution to prevent collisions and network overload during the RA procedure. With this combined strategy, we can ensure that the access delay and outage proportions of the various types of MTCDs are met.

II. RELATED WORK

The problem associated with preamble collisions due to simultaneous massive access requests from MTCDs over cellular networks has seen 3GPP proffer schemes such as access class barring (ACB) and 3GPP backoff schemes that are usually ineffective during massive access as they tend to restrict access during collision and cause excessive access delay [3]. Consequently, researchers have been actively investigating alternate remedies to alleviate the problem. An effective strategy might involve deploying network slicing on the RAN networks to augment the availability of access resources, thereby mitigating the likelihood of collisions. While the authors in [4] discussed the coexistence of evolved mobile broadband (eMBB), ultra-reliable low-latency communication (URLLC) and mMTC in a sliced 5G RAN, Mancuso *et al.* [5] proposed a stochastic RAN slicing for the coexistence of both human type comminucation (HTC) and MTC devices. Although they both showed the importance of network slicing on the control plane, their schemes did not consider the QoS of MTCDs and their use of resource reservation limits the availability of preambles to other slices in the virtual networks. The congestion during the RA process was reduced by the authors in [6] using a priority-based scheme that sliced the RAN using reinforcement learning to tune the parameters of the 3GPP ACB scheme. Their scheme maximises access resources and reduces delay based on the service priority of the individual device. The success of their scheme is based on the limitations of the 3GPP ACB scheme, which controls access by restricting access to some service classes. In [7], the authors suggested a preamble slice orderly queue access (PSOQA) scheme that slices the preamble, weakens random access, and adds queuing access to the virtual layer. This scheme led to an increase in the access success rate and access security for devices. The isolation in multiple slices with different traffic characteristics for efficient network slicing during the RA procedure was studied in [8]. They came to the conclusion that hard splitting of the PRACH resources can fully achieve isolation of the RACH resources among different slices. Vikhrova *et al.* [9] considered three generic service types of the 5G use case and proposed an analytical framework that sliced and shared preambles among these service types. Using the 3GPP resource reservation scheme, they dedicated and shared subsets of preambles and allocated them to specific service types. However, the dedication of preambles to a specific service type means access resources will be limited to other service types. In [10], using a network sliced-enabled intelligent random access framework for mMTC, the authors were able to meet the QoS requirements of MTCDs. However, these solutions mostly used resource separation or reservation to accomplish the network slicing, which partitioned preambles and restricted the use of some physical resources by some service types.

In order to bridge the gap, we leverage the isolation and customisation characteristics of network slicing over the RAN and inculcate Q-learning to lower the outage proportion of MTCDs while still meeting their QoS requirements in terms of access delay. our NSRAS uses power ramping to partition the preambles, thereby enabling isolation and customization of the preambles without restricting less-priority service types from accessing these preambles when not in use. The Q-learning embedded in the orchestrator enables the dynamic allocation and scalability of the slice. By dynamically allocating and scaling the slices, Q-learning ensures efficient resource utilization and adaptability to changing network conditions.

III. SYSTEM MODEL

In 5G-Advanced New Radio (NR), two distinct RA procedures are accessible: the two-step NR RA procedure, which is also known as the two-step RACH; and the four-step NR RA procedure, which is also called the four-step RACH. The four-step RACH is adopted for this research, as shown in Fig. 1. This is due to its backward compatibility and the fact that the proximity of devices to the 5G node B (gNB) affects the effectiveness of the two-step RACH. Additionally, we took into account the requirement for power ramping in our scheme, which would render the two-step RACH inefficient [11]. We consider two types of devices: critical machine-type communication devices (cMTCDs) and delay-tolerant machine-type communication devices (dMTCDs). The physical layer comprises a single gNB with MTCDs deployed, as shown in Fig. 2. We consider a RAN with $\mathcal{G}$ slices where $\mathcal{G} = \{1, 2, 3, ..., g\}$. The MTCDs are informed of the slice to which they belong by the gNB via the system information block 2 (SIB2). Our approach prioritizes the virtualization of control plane resources, as MTCDs consist of tiny-sized data that can be transmitted during the RA process. We define a set of dMTCD in a slice as $\mathcal{N}_{d,g}$ where $\mathcal{N}_{d,g}$ = {1,2,3,..., $n_{\mathrm{d,g}}$}. The set of cMTCDs in a slice is defined as $\mathcal{N}_{c,g}$ = {1,2,3,..., $n_{\mathrm{c,g}}$}.

To investigate the network, we postulate the following:

- Due to the implementation of early data transmission (EDT), access requests and packets shall henceforth be used indiscriminately.
- cMTCDs are presumed to operate on MTC traffic class 1, which is utilized for monitoring, automation, and control and has a latency ranging from 10 to 1000 milliseconds [12].
- For MTCDs, there is no lag time between packet generation and access requests because requests are transmitted immediately as they are generated.

A. Slicing Strategy

The slicing strategy incorporates power ramping in the physical layer, with the slice preambles in the virtual layer formed by a combination of legacy preambles and network slice instance identities (NsiId). [13]. The physical layer isolation is accomplished by ramping the transmit power of the priority MTCDs before the RA process, allowing access to the MTCD with the higher preamble transmit power during collision. This virtually increases the number of available preambles by a multiple of the number of available slices. MTCDs in different slices may use the same preambles, but the gNB can tell the difference between the random access requests of MTCDs in different slices based on their power back-off factor. As a result, our suggested slice preambles may be used in the sliced RAN with just modest modifications to the 5G-Advanced four-step handshake RA protocol, as shown in Fig. 3. The MTCDs are informed of their slice allocation by the gNB using system information block 2 (SIB2) before the commencement of the RA procedure. The MTCDs send their slice preamble ($\mathcal{L}_g$), which comprises a legacy preamble and NsiId to it's v-gNB to begin the four-step RACH RA procedure. In the event of a collision caused by several MTCDs sending the same preamble, the gNB uses power ramping on the physical layer to prioritize the MTCD with higher preamble transmit powers and deny access to those with lower preamble transmit powers. The orchestration layer is implemented in the gNB, where the q-learning algorithm is embedded. The collision probability of the dMTCDs is used by the learning algorithm to dynamically vary the number of slices.

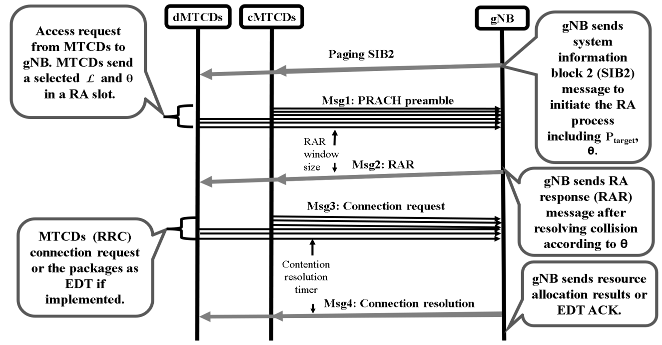

Fig. 3. Illustration of a Random Access Network in the physical layer

B. network-slicing random access

The random access technique is intended to significantly lower the average outage proportion of dMTCDs while satisfying the QoS requirements of cMTCDs. The MTCDs are activated within a time interval t $(0 \leqslant t \leqslant T)$ according to a beta distribution (f(t)) with probability density function (PDF) given as:

$$f(t) = \frac{t^{\alpha-1}(T-t)^{\beta-1}}{T^{\alpha+\beta-1}.Beta(\alpha,\beta)} \tag{1}$$

where $Beta$ (α, β) denotes the beta function with parameters α and β. We denote the arrival rate for cMTCD and dMTCD during *i*th RA opportunity (RAO) as:

$$\lambda_{c,g}(i)= n_{\mathrm{c,g}} \int_{t_i}^{t_{i+1}} f(t)dt \tag{2a}$$

$$\lambda_{d,g}(i)= n_{\mathrm{d,g}} \int_{t_i}^{t_{i+1}} f(t)dt \tag{2b}$$

Considering that before executing the RA process, MTCDs will choose slice preamble $\mathcal{L}_g$, which contains the power back-off factor (ρ). This enables cMTCDs to use the same preamble as other types of devices and yet be allowed access. The following formulas may be used to determine the success probabilities [14] for cMTCDs and dMTCDs:

$$\mathcal{P}_{c,g}=\sum_{m=1}^{\lambda_{c,g}} \binom{\lambda_{c,g}-1}{m-1} \mathcal{L}_g\rho \Big(\frac{1}{\mathcal{L}_g\rho}\Big)^{m} \Big(1-\frac{1}{\mathcal{L}_g\rho}\Big)^{\lambda_{c,g}-m} \mathcal{P}_{Msg3}, \quad (1 < m \leqslant g) \tag{3a}$$

$$\mathcal{P}_{d,g} \;=\; \sum_{m=1}^{\lambda_{d,g}} \;\mathcal{L}_g\rho\frac{1}{\mathcal{L}_g}(1-\frac{1}{\mathcal{L}_g})^{\lambda_{d,g}-1} \;, \qquad (m=1) \tag{3b}$$

Where m denotes the number of devices that selected the same preambles during the RA procedure. The difference is the above formulas for cMTCDs and dMTCDs reflects the implementation of power ramping in the physical layer which gives priority to the cMTCD slice in the case of collision. The collision probabilities for both types of devices can be calculated as:

$$\mathcal{Q}_{c,g} = 1 - \mathcal{P}_{c,g} \tag{4a}$$

$$\mathcal{Q}_{d,g} = 1 - \mathcal{P}_{d,g} \tag{4b}$$

We assume that the average time spent on a successful RA procedure is $\mathcal{T}_s$, likewise the average time spent on a failed RA procedure as $\mathcal{T}_f$. The back-off times for preamble re-transmission are given as $\mathcal{T}_{off\,c}$ and $\mathcal{T}_{off\,d}$. Likewise, the access class baring (ACB) resolution time can be denoted as $\mathcal{T}_a$. Therefore we can calculate the average delay as:

$$\mathcal{D}_{c,g} = (\mathcal{T}_s\mathcal{P}_{c,g} \quad + \quad \mathcal{T}_f(\prod_{i=1}^{\eta}\mathcal{Q}_{c,g}) \quad + \mathcal{T}_{off\,c})\psi \quad , c \in \mathcal{N}_{c,g}, \tag{5a}$$

$$\mathcal{D}_{d,g} = (\mathcal{T}_s\mathcal{P}_{d,g} + \mathcal{T}_f(\prod_{i=1}^{\eta}\mathcal{Q}_{c,g}) + \mathcal{T}_{off\,d} + \mathcal{T}_a)\psi \quad , d \in \mathcal{N}_{d,g} \tag{5b}$$

Where η_{max} represents the PreambleTransMax, which is the parameter that determines the maximum number of preamble re-transmissions a MTCD can make after encountering a collision during the RA procedure. Therefore we elucidate the preamble collision counter as η where $\eta = \{1, 2, 3, ..., \eta_{max}\}$ We define a binary function ψ that indicates congestion in the access network during the RA procedure, thereby not including fail access delay in the average access delay of successful access mathematically as:

$$\psi \;\; = \begin{cases} 1, if\ \eta \leqslant \eta_{max} \\ 0,\ \textit{otherwise} \end{cases}$$

The average number of failed access can be calculated as:

$$\mathcal{N}_{F\,c,g} = n_{\mathrm{c,g}}\lambda_{c,g}(\prod_{i=1}^{\eta_{max}}\mathcal{Q}_{c,g}) \tag{6a}$$

$$\mathcal{N}_{F\,c,g} = n_{\mathrm{d,g}}\lambda_{d,g}(\prod_{i=1}^{\eta_{max}}\mathcal{Q}_{d,g}) \tag{6b}$$

The outage proportion for each types of MTCD can be represented as follows:

$$\sigma_{c,g} \;\; = \frac{\mathcal{N}_{F\,c,g}}{n_{\mathrm{c,g}}} \tag{7a}$$

$$\sigma_{d,g} \;\; = \frac{\mathcal{N}_{F\,d,g}}{n_{\mathrm{d,g}}} \tag{7b}$$

IV. Q-LEARNING FUNDAMENTALS

To accomplish the abstraction of the virtual layer from the physical layer, we employ reinforcement learning, specifically Q-learning. In Q-learning, an agent takes actions in an environment to maximize its cumulative rewards based on values learned. In the context of abstracting the virtual layer from the physical layer, Q-learning enables the agent to learn the optimal actions to take. The agent becomes capable of autonomously deciding on the optimal steps to either increase or decrease the number of virtual layer(s) functionality from the physical layer by providing a reward mechanism and updating the Q-values depending on experienced results. The outcomes of the agent's actions, known as rewards, might be either favorable or unfavorable The reward for each action is determined in order to evaluate its performance. Finding the optimal course of action for the next iteration is the aim of the learning process. In order to learn Q-values, Q-learning iteratively updates estimates using the Bellman equation. The Bellman equation states that, under optimal policy assumptions, the best Q-value ($Q(s, a)$) for a state-action pair is the sum of the immediate reward from the action and the discounted future rewards and can be represented mathematically as:

$$Q(s, a) = r(s, a) + \gamma * maxQ(s^{'}, a^{'}) \tag{8}$$

Where $r(s, a)$ is the immediate reward for action a in state s, and $maxQ(s^{'}, a^{'})$ is the maximum expected future rewards for future actions and states. γ represents the discount factor ($0 \leqslant \gamma \leqslant 1$), The q-learning model iteratively updates the Q-values based on the selected learning rate (α), which accepts values ranging from 0 to 1 using the formula shown below.

$$Q(s, a) \leftarrow Q(s, a) + \alpha * [r(s, a) + \gamma * maxQ(s^{'}, a^{'}) - Q(s, a)] \tag{9}$$

A. Q-learning Orchestrator Algorithm

The proposed Q-learning orchestrator algorithm is implemented on the gNB where the number of virtual slices can be dynamically varied with respect to the collision proportion of dMTCDs as illustrated in algorithm 1. This is due to the fact that cMTCds are granted access during collision due to power ramping in the physical layer; therefore, dMTCDs suffer more collisions; hence, it is pertinent to use the collision proportion of the dMTCDs as a threshold. At the beginning of RAO, both cMTCDs and dMTCDs attempt the RA procedure in the same slice; however, as the collision proportion increases, the Q-learning algorithm learns from its environment and dynamically varies the number of slices. The Q-learning algorithm is defined as a Markov decision process (MDP) on a state space (s_k). The transition between states, represented by the action (a_k), is determined by the collision proportion of dMTCDs in the network. The objective of the algorithm is to satisfy the QoS demands of cMTCDs through the dynamic adjustment of slice quantity to increase the availability of access resources. This guarantees that the access of dMTCDS does not impede the access of cMTCDs, while also distributing the access of dMTCDs over time utilizing the 3GPP ACB scheme. The reward value (r_k) which is the difference between the actual collision probability ($\mathcal{Q}_{d,g}$) of dMTCDs and a predetermined threshold collision probability ($\mathcal{Q}_{th}$) can be represent as:

$$r_f = (\mathcal{Q}_{d,g} - \mathcal{Q}_{th}) \times 100 \tag{10}$$

Depending on the values of $\mathcal{Q}_{d,g}$ and $\mathcal{Q}_{th}$, the reward value r_k might be positive or negative. The negative reward value indicates that the collision probability for the dMTCDs is less than the needed threshold; hence, the number of slices remains in it's initial state (s_k). A positive reward value, on the other hand, signifies that action needs to be taken to remedy the collision problem. We establish three reward thresholds for positive rewards: the minimum, medium, and maximum positive reward values. A minimum-positive reward value (r_{fmin}) will necessitate the transition to the subsequent state, $s_k + 1$., hence augmenting the value of the ACB factor for dMTCDs. The medium-positive reward value (r_{fmed}) will require a transition to stage s_k+2, which involves an increment of the number of slices by $G + 1$. Ultimately, achieving the maximum positive reward value (r_{fmax}) will include transitioning to state $s_k + 3$, resulting in an increase in $G + 2$ slices.

Algorithm 1 Q-learning Orchestrator Algorithm

```
Initialize MTCD in terms of dMTCD and cMTCD
Initialize ALL MTCD to perform RA to v-gNB
Set number of slice G
Set T_K1, T_K2 and Q_th
Calculate Q_d,g according to equation 4B
Calculate r_f according to equation 10
Initialise Q_th, r_f, ε, α , epsilon_decay, min_epsilon
Initialise Q(s,a) = 0
Use Greedy strategy and generate random number y
    if y ≥ ε
        choose action a according to maxQ(s,a)
    else if y < ε
        choose action a randomly
    end if
    ε = max(ε * epsilon_decay, min_epsilon)
Execute action " a " and calculate reward r'_f
Set G = G'
Update Q(s,a)
Check Q convergence
```

V. PERFORMANCE EVALUATION

The simulation parameters outlined in Table 1 were used to validate the proposed scheme. We set the ($\mathcal{Q}_{th}$) at 0.4 and the maximum number of slices (g) was set at 3. We assume that the RA opportunities happen within 10 ms. The cMTCDs were designed to access other slices in order to meet their QoS requirements. We set the reward value range of $20\% > r_f \geqslant 0$ as r_{fmin}, $40\% \geqslant r_f \geqslant 20\%$ as r_{fmed}, and $r_f > 40\%$ as r_{fmax}. To ensure a fair comparison between the 3GPP ACB scheme and our proposed system for both cMTCDs and dMTCDs, we have included the back-off time in Table 1 for the 3GPP ACB scheme for both types of devices.

TABLE I
SIMULATION PARAMETERS

Parameter	Values
PRACH configuration Index	6
Number of Preambles	54
PreambleTransMax(η_{max})	5
Power back-off factor(ρ)	3
Shape parameter of Beta function(α)	3
Shape parameter of Beta function(β)	4
Back off time step 1($T_{\mathrm{off}c}$)	2 subframe
Back off time step 2 ($T_{\mathrm{off}d}$)	5 subframe
ACB resolution time (T_{a})	3 subframe
Number of resource block per slot	6RBs
Max MTCD transmission Power	23dBm
% of dMTCD	60
% of cMTCD	40
Pathloss (MTCDs to gNB)	35.2 + 35$\log_{10} d$, d in km

A. Performance Metrics

The effectiveness of our suggested plan was assessed using the performance metrics listed below, which emphasized the goal of this study:

- Average MTCD access delay: The average time it takes for each type of device to create a packet and access the gNB successfully.
- Average outage proportion: The average outage proportion is the ratio of failed access requests from MTCDs to the total access requests of these devices.

B. Performance Results

To accurately assess the performance of our suggested scheme in relation to the 3GPP ACB systems, we offer the results of our simulations, which are displayed in Figs 4 and 5. The simulation findings depicted in the said figures clearly demonstrate the superiority of our proposed scheme over the 3GPP ACB schemes. These results demonstrate considerable gains in terms of outage proportion and access delay, particularly for cMTCDs, and support the feasibility of our proposed approach.

Our scheme achieves a lower average access delay than the 3GPP ACB scheme, irrespective of the types of MTCDs, as can be seen in Fig. 4. This can be attributed to the implementation of power ramping embedded network slicing, which made more access resources available, and using Q-learning, access resources can be dynamically allocated to meet the QoS requirements of the various types of devices. Although the same back-off time was used for both schemes, the lack of slicing and intelligent access resource allocation means that in the 3GPP ACB scheme, MTCDs tend to experience more collisions, thereby having a higher accumulated back-off time.

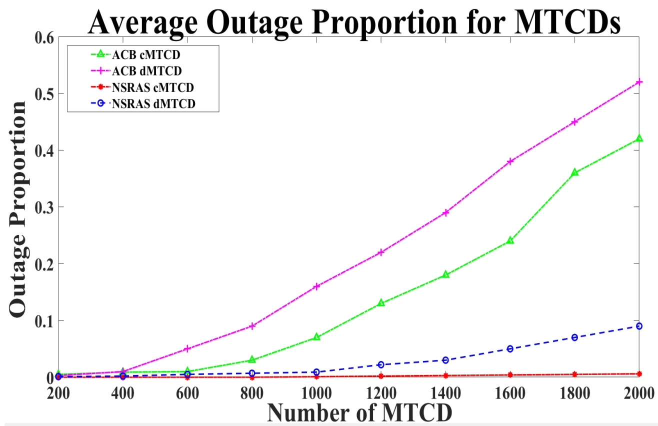

Fig. 4. Average delay of MTCDs

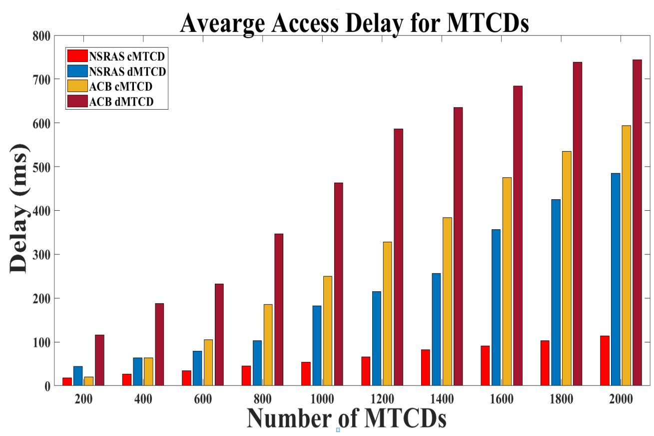

Fig. 5. Average Outage proportion for MTCDs

As a result, the 3GPP ACB scheme may struggle to effectively manage the increasing number of devices on the network, leading to longer access latency and potentially degraded performance for MTCDs. In contrast, our scheme's ability to intelligently allocate access resources based on device QoS requirements allows for more efficient and optimized access, resulting in lower collision rates and improved overall access delay. This highlights the importance of intelligent access management in ensuring a smooth and reliable connectivity experience for MTCDs.

Finally, Figure 5 shows the average outage proportion for both schemes, with our scheme having the lowest average outage proportion. This is explicable by the fact that more access resources become available, particularly as the number of slices increases. Thus, our scheme guarantees minimal downtime for key equipment by offering a better degree of accessibility and dependability. Additionally, the implementation of the Q-learning algorithm also contributes to improved overall network performance, as it allows for better distribution and utilization of resources across all devices and slices. Ultimately, our scheme proves to be highly efficient in ensuring uninterrupted and reliable communication for critical devices in a variety of scenarios.

VI. CONCLUSION

In conclusion, the problem of collision and network overload in 5G networks was studied, and we proposed a NSRAS to tackle these challenges. Our scheme combines network slicing and Q-learning to dynamically increase and allocate access resources to meet the QoS requirements of MTCDs. Through the simulation and analysis of the results, we validated that the average access delay and average outage proportion were enhanced.

ACKNOWLEDGMENT

The authors are grateful for the support from the University of Cape Town, the UCT-Telkom Centre of Excellence (CoE) in Broadband Networks and Applications, Rondebosch, Cape Town, and the University of Calabar, Nigeria.

REFERENCES

[1] 5G-Americas, "Becoming 5g-advanced: the 3gpp roadmap," Tech. Rep., December 2022.

[2] Ericsson Mobility Report, "Near-immortal devices and a sustainable deploy-and-forget future," *https://www.ericsson.com/en/reports-and-papers/mobility-report/reports/june-2023*, 2023.

[3] 3gpp, "Medium access control (mac) protocol specification (release 8), access, evolved universal terrestrial radio," *TS 36.321, V16. 4.0.*, 2021.

[4] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5g wireless network slicing for embb, urllc, and mmtc: A communication-theoretic view," *Ieee Access*, vol. 6, pp. 55 765–55 779, 2018.

[5] V. Mancuso, P. Castagno, M. Sereno, and M. A. Marsan, "Slicing cell resources: The case of htc and mtc coexistence," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 667–675.

[6] A. Turan, M. Koseoglu, and E. A. Sezer, "Reinforcement learning-based adaptive access class barring for ran slicing," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2021, pp. 1–6.

[7] J. Sun, M. Guo, and J. Liu, "Preamble slice orderly queue access scheme in cell-free dense communication systems," *Digital Communications and Networks*, 2023.

[8] "Random access for network slicing," *10th EAI International Conference on Mobile*, 2017.

[9] O. Vikhrova, C. Suraci, A. Tropeano, S. Pizzi, K. Samouylov, and G. Araniti, "Enhanced radio access procedure in sliced 5g networks," in *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE, 2019, pp. 1–6.

[10] B. Yang, F. Wei, X. She, Z. Jiang, J. Zhu, P. Chen, and J. Wang, "Intelligent random access for massive-machine type communications in sliced mobile networks," *Electronics*, vol. 12, no. 2, p. 329, 2023.

[11] S. P. Erik Dahlman and J. Skeold, "17. random access," in *5G nr : the next generation wireless access technology*. 125 London Wall, London EC2Y 5AS, United Kingdom: Academic Press, 2017, pp. 349–370.

[12] S. A. AlQahtani, "Analysis and modelling of power consumption-aware priority-based scheduling for m2m data aggregation over long-term-evolution networks," *IET Communications*, vol. 11, no. 2, pp. 177–184, 2017.

[13] 3GPP, "5g; 5g system; network slice selection services," *3GPP TS 29.531; Version 15.6.0, Release 15*.

[14] Y. Piao, Y. Kim, and T.-J. Lee, "Random power back-off for random access in 5g networks," *IEEE Access*, vol. 9, pp. 121 561–121 569, 2021.