E-Prosiding Seminar Antarabangsa Islam dan Sains 2021
"Wawasan Kemakmuran Bersama 2030"
Khamis, 9 September 2021
Universiti Sains Islam Malaysia

E-Prosiding
SAIS 2021

DELAY-AWARE 5G MILLIMETER-WAVE CELLULAR NETWORKS

Nurhajar Abdul Haddi[i], Fauzun Abdullah Asuhaimi[ii], Khairul Nabilah Zainul Ariffin[iii].

[i]Student, Faculty of Engineering and Build Environment, Universiti Sains Islam Malaysia,
[ii](*Corresponding author*) Senior Lecturer, Faculty of Engineering and Build Environment, Universiti Sains Islam Malaysia,
[iii]Senior Lecturer, Faculty of Engineering and Build Environment, Universiti Sains Islam Malaysia.
fauzun@usim.edu.my

Abstract

The potential of millimetre wave (mmWave) to provide dependability in 5G cellular networks has prompted the number of wireless communication network users to rise, indicating the progress of telecommunication standards. Despite the fact that the mmWave bands can support massive data rates, the effort required to deliver these data rates for end-to-end services while maintaining ultra-low-latency performance and reliability to support developing applications and use cases will necessitate a re-evaluation of all layers of the protocol stack. As a result, this paper examines millimetre wave characteristics and how they impact end-to-end transmission. In addition, this study examines the difficulties and potential solutions for offering dependable, end-to-end, and ultra-low-latency services in mmWave cellular networks. Aside from comparing mmWave and microwave, this article will also examine scheduling in order to mitigate the negative effects of intermittent connection in mmWave and fully use the capacity. This project's challenge and solution will be modelled using Deep Reinforcement Learning in order to do mathematical modelling and provide delay-propagation characteristics. Finally, the simulation results of physical modelling are provided to demonstrate the effectiveness of the suggested technique.

Keywords: *Delay, millimetre wave, Scheduling*

INTRODUCTION

One of the problems facing in today's wireless networks is the growing number of users and devices consuming more data traffic. Nevertheless, telecommunications companies have to limit them to the same radio spectrum frequency band they have always used. This implies that a finite amount of bandwidth is allocated to each device, leading to slower speeds and frequent disconnections. The lack of frequency band resources will become even more noticeable as the number of devices connected to wireless networks rises. This issue has a massive effect on the experience of consumers. Millimetre wave (mmWave) technology, however, provides a practical solution to this problem.

MmWave is the real technology beyond 5G network advancement. It is also known as a signal with a very high frequency. It is ideal for 5G networks since it can deliver high-frequency data. Millimetre wave is the band of frequencies between 30 GHz to 300 GHz where the wavelength range is between 10 millimetre and 1 millimetre. Figure 1 illustrate mmWave range in spectrum band between 30 GHz and 300GHz corresponding to the range of wavelength (10 mm to 1 mm) The mmWave frequency mechanism has vast amounts of adjoining raw bandwidth available, allowing for more data traffic for multiple multimedia application (Burghartz et al., 1998, Elmezughi et al., 2021).

It can be widely used in products and services, such as high-speed, point-to-point wireless local area networks (WLANs) and broadband access. Since mmWave allows higher data rates on lower frequencies, such as those used for Wi-Fi and current cellular networks, it is used in a wide range of mobile and wireless networks (Semiari et al., 2019). The mmWave technology also has been developed in cellular systems-based, resulting in broader bandwidth than the bandwidth of current cellular networks (*What Are Millimeter Waves? - Everything RF*, n.d.).

MmWave systems use many antenna elements at the base station and the user equipment to overcome the propagation challenges. This matter will lead to high directivity gains, fully directional communications, and possible noise-limited operations. MmWave communication is broadly considered as a promising technology for 5G cellular and next-generation WLANs. The new technology of cellular communication is estimated to support 1000 times more capacity and provide 100-times more smart devices than common mobile networks (Liu et al., 2017).

The 5G cellular wireless network will utilize the massive spectrum in mmWave bands (above 10 GHz) to enable these emerging technologies, potentially boosting the wireless capacity for Enhanced mobile broadband (eMBB) services and reduce the transmission delay for low-latency applications. However, the only drawback of mmWave is its short wavelength, which ranges from 1 to 10 millimetres. Due to the limited bandwidth, signals may be dispersed if they pass through an obstacle in their direction (Semiari et al., 2019).

In a wireless communications network, the total delay comprises four elements: propagation delay, transmission delay, processing delay, and queuing delay. Propagation delay can be translated as the time taken for a signal to propagates via a medium. Transmission delay is the time taken to push the packet's bit into the link in use. Processing delay is the time for processing a packet header and make a routing selection. Lastly, queuing delay is the time a packet spends in a buffer or routing queue. Generally, the total delay in a queuing process is mostly depends based on its queuing delay.

The other delay elements, however, can be considered negligible because the value is too small compared to other elements. Hence, for low latency systems with buffer, the critical challenge is to reduce the queuing delay substantially (Bertsekas & Gallager, n.d.),(Kurose et al., 2013). Even minor network delay may result in a significant increase in system response times, resulting in users' frustration, rebroadcasts, and application error. More data and delay can bound on network bandwidth and lead to routers congestion.

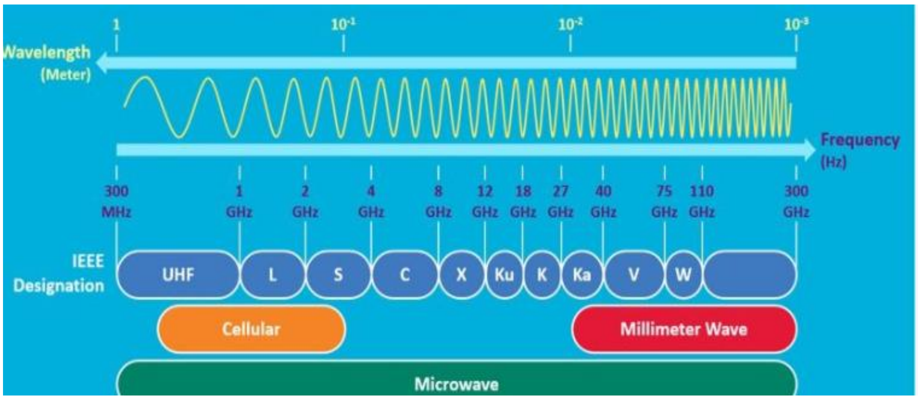

Figure 1: mmWave range in spectrum band between 30 GHz and 300GHz corresponding to the range of wavelength (10 mm to 1 mm) (What Are Millimeter Waves? - Everything RF, n.d.).

METHODOLOGY

In this section, a detail methodology of the process will be explained briefly. There are two parts for this methodology system model deep reinforcement learning (DRL) approach and delay optimal policy problem formulation. In the DRL, there will two sections that will be explained whereas the first section is about Markov decision Process (MDP) and the second section explains the Q-Learning involved to obtain the optimal policy.

System Model

This study considered two interfaces, as shown in Figure 2, where the system model of architecture of integrated sub6 GHz and mmWave is proposed. Due to high frequency feature of mmWave, its cell sizes are designed smaller than Sub6 GHz and have a higher in density. Even though mmWave can carry large data, its high-frequency bands can only travel on smaller area coverage (Lopez et al., 2019). Aside from that, 6 GHz and 4G LTE are intended to coexist in a close integration with 5G deployments below mmWave. To maintain efficiency and coverage, quick adaptability to switch channel conditions would allow shifting within across cells.

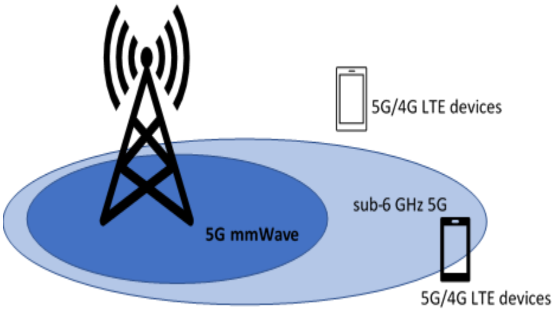

Figure 2: Dual sub6 GHz and mmWave architecture

Interface Scheduling

Optimal mmWave level throughput with bounded latency can be achieved by developing an efficient scheduling policy that is capable of determining interface(s) to choose and limiting the queue sizes of interfaces. The Sub6 GHz interface will be applied as the secondary data transfer mechanism. Figure 2 illustrates the load division component in the proposed system for dual-interface communications and data transmission using Sub6 GHz and mmWave. To achieve maximum mmWave throughput a constrained delay performance is by arranging traffic arrivals over the Sub6 GHz or mmWave.

In order to minimize the delay, scheduling process method is proposed. The scheduling process involves two servers which are mmWave and sub-6GHz server, in which the former has huge average service rate but with high dynamic nature (mmWave) whereas the later one has small average service rate but stable (sub6 GHz). Figure 3 depicts the system model of architecture of integrated Sub6 GHz and mmWave. The infinite type of head buffer is designed to hold all packets that will then be operated either by sub6 GHz or mmWave.

However, before scheduling process, the processing serves are utilized for vital data processing. Both servers operate at different average service rates, with mmWave processed 100 times more average service rate than sub6 GHz (Yao et al., n.d.). A buffer is added to the mmWave server during the queueing phase. Then, packets routed from the head buffer are stored. There are many factors of using independent queue for the mmWave interface. First, the processing server rate speed is similar to the mmWave's average service rate. Furthermore, due to mmWave high frequency, mmWave is extremely sensitive to blockage and making it difficult to forecast rapidly.

During this phase, the packets will undergo service time for both the processing and mmWave servers with the waiting time in the head buffer is excluded. This is approximately twice of mmWave service time. Other than that, the mmWave's performance will be reduced by half. However, with contradictory, the waiting time in the head buffer can be utilised to process better packets if the mmWave server has its own buffer for the processed packets. As a result, service times can be shortened.

Despite that, Sub6 GHz performed differently than mmWave since it is much slower than the processing server. Hence, the service time of the Sub6 GHz server able to neglect the delay of processing thus shows, when considering the cost of buffer, it is unnecessary as the Sub6 GHz server to have its own buffer. Relevantly, Sub6 GHz interface is considered to perform either a server or a buffer whereas the mmWave interface consists of a server and a buffer.

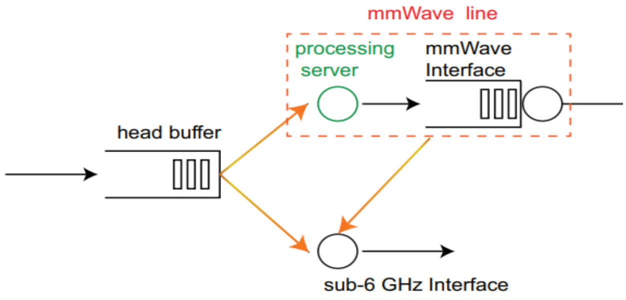

Figure 3: integrated Sub6 GHz –mmWave Scheduling architecture

Following that, the aim of maximizing utilisation of mmWave channel can be achieved by formulating an optional scheduling problem which result in bounded delay performance. The optimal threshold type policy is presented to determine when a data packet should be pushed either to the mmWave queue or Sub6 GHz queue. The optimal policy is when the scheduler only pushes the arrival traffic to the mmWave queue if and only if the queue length is lower than a threshold. The threshold-based scheduling policy is achieved by efficiently captured the dynamics of the mmWave channel, thus maximizing the channel usage. In order to perform threshold-based scheduling policy, mathematical computation is implemented based on a mathematical method.

In addition, the packets are supposed to be reneged to Sub6 GHz for service by implying impatient packet in the mmWave queue. It should be highlighted that the packet in the Sub6 GHz server cannot be returned to the mmWave queue or head buffer. There are new challenges caused from the addition of reneging flexibility to mmWave packets whether packets should be routed from the head buffer or mmWave queue to the Sub6 GHz queue. Furthermore, our problem should include the trade-off between waiting in the head queue (system entrance) and accessing the slow server which triggered by the slow-server problem, the trade-off between waiting in the mmWave queue and entering the slow server as well as the trade-off between dispatching packets from the head buffer and the mmWave queue.

Deep Reinforcement Learning Approach Threshold Type Policy

Deep reinforcement learning (DRL) combines reinforcement learning (RL) with deep learning by applying deep neural networks to create an artificial agent capable of learning optimum policies directly from high-dimensional sensory stimuli via end-to-end reinforcement learning. The reinforcement learning approach generates incentives depending on the agent's behaviour in a particular environment.

In this technique, the agent learns to interact with the environment using this method by structuring the experience in such a manner that the long-term rewards are maximised. The most frequently used reinforcement technique is Q-learning, with deep Q networks emerging from the combination of reinforcement learning and deep learning. Furthermore, substantially all RL issues may be expressed as MDPs, because an MDP can represent the observable environment for RL. As a result, in order to use a DRL method, the items in MDP must first be specified. In a real-world situation, the MDP's aim is to maximize the received rewards.

Markov Decision Process Elements

In MDP, three factors must be considered: state space, action space, and reward function. Let S and A be the discrete set of environment states and discrete set of actions, respectively. The system senses the state $s_{i,\,t} \in$ S and selects an action $a_{i,t} \in$ A at each timeslot t. The environment undergoes a transition to a new state as a result of the activity done., $s_{i,t\,+1} \in$ S according to probability $\Pr(s_{i,t\,+1} \mid s_{i,t}\,,\, a_{i,t})$ and generates a reward, $R_{i,t}(s_{i,t}\,,\, a_{i,t})$ to the representative A DRL technique is suggested in this work to determine the best strategy for delay minimization. However, before developing the threshold type policy, the state space, action space, and reward function must be specified.

i. **State space:** State can be expressed by a four-dimensional vector $\mathbf{q} \triangleq (\mathbf{q_0}, s_1, \mathbf{q_1}, s_2)$ with state space of $S(t) \triangleq N \times \{0, 1\} \times N \times \{0, 1\}$**.** Where q_0, q_1 denote the queue length of the head buffer and mmWave interface, respectively which q_0, $q_1 \in N$. Moreover, s_1, $s_2 \in \{0, 1\}$ denote condition of the processing server and Sub6 GHz interface either busy or idle, respectively.

ii. **Events:**
   - Arrival of a packet to the head buffer: The system's state changes with the arrival of one packet.
   - Moving of a packet from the mmWave interface: The departure of a packet from the mmWave queue changes the system state
   - Moving a packet from the Sub6 GHz interface: When a packet leaves the Sub6 GHz queue, the system's status changes.

- Processing completion: If the processing server delivers a packet to the mmWave queue, the system state changes

iii. **Action space:** $a(t) = \{A_s, A_1, A_2, A_b, A_r\}$ where

- *Static:* System state is unaffected. Hence,
  $A_s(q) \triangleq (q_0, s_1, q_1, s_2)$
- *Schedule to mmWave:* If the processing server is idle, A packet can be routed to the mmWave line. Hence,
  $A_1(q) \triangleq (q_0 -1, 1, q_1, s_2)$, where $\{q \mid q_0 \geq 1, s_1 = 0\}$
- *Schedule to Sub6 GHz:* If the processing server is idle, A packet can be routed to the mmWave Sub6 GHz . Hence,
  $A_2(q) \triangleq (q_0 -1, s_1, q_1, 1)$, where $\{q \mid q_0 \geq 1, s_2 = 0\}$
- *Schedule to Both interface*: dispatches two packets to the Sub6 GHz and processing servers simultaneously. Hence,
  $A_b(q) \triangleq (q0 -2, 1, q1, 1)$, where $\{q \mid q0 \geq 2, s_2 = s_1 = 0\}$
- *Reneging:* moves a packet from the mmWave line to the Sub6 GHz interface. Where $\{q \mid q_1 + s_1 \geq 1, l_2 = 0\}$. Ar_p and Ar_{mm} denote the reneging actions from the processing server and mmWave interface respectively. Therefore, we have:
  $At_p(q) \triangleq (q_0, 0, q_1, 1)$, $q \in \{q \mid l_1 = 1, s_2 = 0\}$
  $Atmm(q) \triangleq (q0, s1, q1, 1)$ $q \in \{q \mid q1 \geq 1, s2 = 0\}$

Q-Learning

DQN is produced when Q-Learning is coupled with a DNN. DQN, an abbreviation for DRL, employs a DNN to derive the cause and effect relationship between state-action pairs (si,t, ai,t) and then estimates its value function Q(si,t, ai,t ; i,t) (Mnih et al., 2015). However, when a DNN is used with Q-Learning, significant convergence and stability issues occur. First, a technique known as experience replay was introduced, in which the agent's encounters with the environment are recorded in memory and used to train the neural network via a random mini-batch process.

The second change is to utilise two distinct neural networks, one that is continually assessed and modified based on the agent's experience, and another, a target network, where the weights are updated on a regular basis. In addition, an online training mechanism is designed so that the values of the action-value function may be learnt based on the agent's interactions with the environment and observations. The essence is to approximate the optimal action value function using learned action value function from sampled rewards:

$$\mathrm{Qt} + 1(\mathrm{st}, \mathrm{at}) = (1 - \alpha)\, \mathrm{Qt}\, (\mathrm{st}, \mathrm{at}) + \alpha(\mathrm{R}(\mathrm{st}) + \gamma \max \mathrm{a} \in \mathrm{A}(\mathrm{st} + 1)\, \mathrm{Qt}\, (\mathrm{st} + 1, \mathrm{a}))$$

where $\alpha \in (0, 1)$ indicates the learning rate and $\gamma \in (0, 1)$ is the discounted factor It is possible to demonstrate that the system converges to the average reward case when γ

= 1. R(s) represents the reward seen in state S, which is set to $\frac{1}{S(t)}$ . . The Q-network may be trained specifically by changing the parameter to minimise the loss function, which is represented as follows:

$$L(\theta) \triangleq \frac{1}{\Omega} \sum_{w \in \Omega_t} \left(Q'_{i,w} - Q(s_{i,w}, a_{i,w}; \theta)\right)^2 \text{ --- (1)}$$

in which Ω_t indicates the index set of the random minibatch used at the t-th iteration, and $Q'_{i,w}$ is a value estimated using a Bellman equation, by fixing set of weights from the previous iterations of the learning procedure.

Delay Optimal Policy

The average delay reduction problem, according to Little's Law, is comparable to minimising the average number of packets in the system, which indicated as:

$$L = \lambda \times W,$$

where L indicates the time average number of users in a line or system, and and W signify the arrival rate and average waiting time per user, respectively. The question is then derived as:

$$\min_{\pi \in \Pi} \lim_{T \to \infty} \sup \frac{1}{T} \, E^{\pi} \left[\int_{t=1}^{T} (q[t] \cdot e) \, dt\right] \text{ -------- (2)}$$

where E^{π} indicates the conditional expectation given policy π, q[t] ∈ S(t) is the system state at time t, Π indicates the collection of all permissible policies, and $e = (1, 1, 1, 1)^T$ where T denotes the customers' average throughput.

Directly learning π is difficult. Q-learning is an alternate technique to solving the equation in reinforcement learning (2) (Watkins & Dayan, 1992).Despite of learning π, an action-value (also known as Q) function is also investigated to evaluate the theoretical discounted cumulative reward when some action a(t) in each state, s(t) is taken. After learning such an action-value function, the optimum policy may be created when the action with the greatest value in each state is taken. The core principle of Q-learning and many other reinforcement learning algorithms is to update the action-value function iteratively based on the formula of value iteration update:

$$Q_{t+1}(s_t, a_t) = (1 - \alpha)\, Q_t(s_t, a_t) + \alpha(R(s_t) + \gamma \max a \in A(s_{t+1})\, Qt\, (s_{t+1}, a)) \text{ ------ } \quad (3)$$

Given that the processing server and mmWave service rates are in the similar order, it is fair to infer that the anticipated time for packets to go via an empty mmWave line is shorter than that of an empty Sub6 GHz interface, such that:

$\frac{1}{\mu_p} + \frac{1}{p_a \, \mu_{mm}} < \frac{1}{\mu_{sub-6}}$, From here we can obtain properties:

(a) action A_1 has priority over action A_h:

$J_\beta(A_1(q)) \leq J_\beta(A_s(q))$ if $q_0 \geq 1$, $S_1 = 0$.

(b) action A2 has priority over action Ar:

$J_\beta(A_2(q)) \leq J_\beta(A_r(q))$ if $q_0 \geq 1$, $s_1 + q_1 \geq 1$, and $S_2 = 0$.

(c) A packet should be scheduled on the mmWave line whenever the mmWave line is empty and the head buffer is not empty:

$J_\beta(A_1(q)) \leq J_\beta(A_2(q))$ if $q = (q_0, 0, 0, 0)$ and $q_0 \geq 1$

Where Jβ (q) denote optimal expected total discounted delay function of initial state q. From properties above, three guidelines that provides characteristics of the optimal policy can be explained:

- Static is not preferable if the processing server is idle i.e., property (a).
- Keep the mmWave line busy i.e., Properties (a) and (c).
- Head buffer is the first choice for the Sub6 GHz interface i.e., property (b).

Based on these rules, we show that optimal policy for the discounted delay problem is of the threshold type, and is defined as follows: Dm (q) =

A_1 (q) i.e., $q = (q_0, 0, q_1, 1)$, $q_0 \geq 1$, or $q = (q_0, 0, q_1, 0)$, $q_0 \geq 1$, $q_0 + q_1 \leq m$,

A_2 (q) i.e., $q = (q_0, 1, q_1, 0)$, $q_0 \geq 1$, $q_0 + q_1 + 1 > m$, or $q = (1, 0, q_1, 0)$, $q_1 \geq m$,

At (q) if $q = (0, s_1, q_1, 0)$, $S_1 + q_1 > m$,

A_b (q) if $q = (q_0, 0, q_1, 0)$, $q_0 + q_1 > m$, $q_0 \geq 2$,

A_s (q) otherwise,

where Dm is a threshold policy with threshold m such that Dm defies all of the preceding rules to demonstrate Dm's optimality for the discounted delay issue, we label the action sets A1, Ah and A2, Ar as "not-adding-to-sub-6" and "adding-to-sub-6," respectively.

To determine the priority between the sets not-adding-to-sub-6 and adding-to-sub-6, the path consist of the head buffer, the processing server, and the mmWave queue is dubbed "Fast-lane" so that 'adding to sub6' has more priority over not adding-to-sub-6 when the Fastlane queue length exceeds a certain threshold m, i.e. This is demonstrated via value iteration. For convenience, the system state, s(t), is re-expressed as (x, q1, S2), where x determines the total of packets in processing server and the head buffer. It should be noted that if x > 0, the processing server should be busy with the first guideline.

RESULTS AND DISCUSSIONS

TensorFlow is used to evaluate the performance of the proposed method, and the same environment as in (Zhong et al., 2017) is taken into account. This section explains system parameters and discusses experimental findings.

Simulation Parameters

This simulation encounters four forms of delays: propagation delay, transmission delay, processing delay, and queuing delay. The mmWave and Sub6 GHz propagation delays are estimated to be 0.8ms and 0.1ms, respectively. The transmission delay is computed as follows:

$$\text{Transmission delay} = \frac{\text{Packet size}}{\mathit{Throughput}}$$

where the Throughput can be obtained by applying Shannon Hartley theorem where:

$$\mathbf{C = BW\ log\ (1 + SNR)},$$

And C stands for capacity, while BW and SINR stand for channel bandwidth and signal-to-noise ratio, respectively. In this simulation, the number of users is set to 15 with packet size = {3, 7...,20} and the SINR (in dBm) is chosen from the set y = {14,15, · · · ,25}. The service rate of the mmWave interface and the processing server of Sub6 GHz are in the same order, mm = p = 100 and sub-6 = 1. The probabilities of accessible and unavailable states are considered to be pa = 0.6 and P_{na} = 0.4, respectively.

In this paper, the deep neural network (DNN) is utilised in this article to approximate the action-value function, and it consists of three fully connected feedforward hidden layers of 256, 256, and 512 neurons, respectively. For updating the weights θ, the Adam algorithm (Kingma & Ba, 2014) is used, with the size of a minibatch set at 256. The replay memory D is assumed to retain N_D = most recent transitions, and training of starts only when D stores more than O = 300 transitions in each iteration. The total number of iterations is set to K = 10^5. The total number of iterations is set to K = 10^5. From 0.8 to 0, the frequency of investigating new actions drops linearly with the number of iterations. In precisely, for iteration k, is let to:

$$\varepsilon k = 0.8(1 - k/K)$$

The success rate and the average number of transition steps are used to measure performance. The success rate is calculated by dividing the total number of independent runs by the number of successful trials. A trial is done simultaneously if it progresses to the qualifications within 20-time frames. If a trial is successful, the average number of transition steps is defined as the average number of time frames involved to attain a goal state. The simulation results shown below are for arrival rates of 30 and 40 with average delay vs different thresholds.

Performance of DRL Algorithm

A simulation of the hypothesized algorithm's performance across 200k independent runs during the training process. The DRL algorithm's performance is an indicator of the loss of the Q function, which is calculated in (2). In general, Figure 4 demonstrates that as the number of iterations grows, the loss Q-function lowers until it becomes constant at the lowest loss function value after 90k training cycles. This demonstrates that the suggested method can converge, and that the scheduling strategy can make the best decision given any system state.

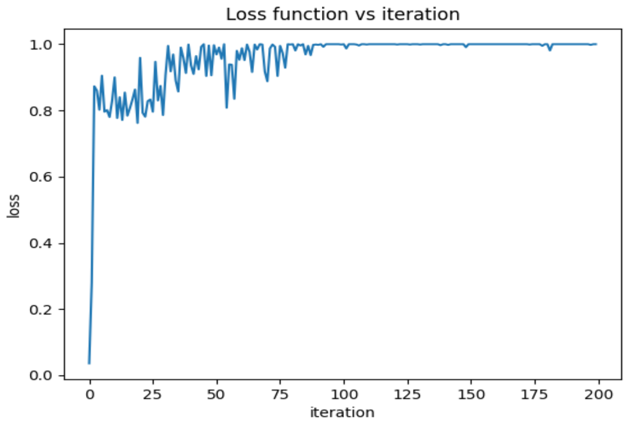

Figure 4: Loss function vs Iteration graph

Relationship of Average Delay and Optimal Threshold

Based on the Figure 5.1, it demonstrated that increasing a limited threshold reduces the average latency. According to the results, if there are a large number of packets added to the mmWave queue and the mmWave link becomes unavailable, a significant average delay occurs.

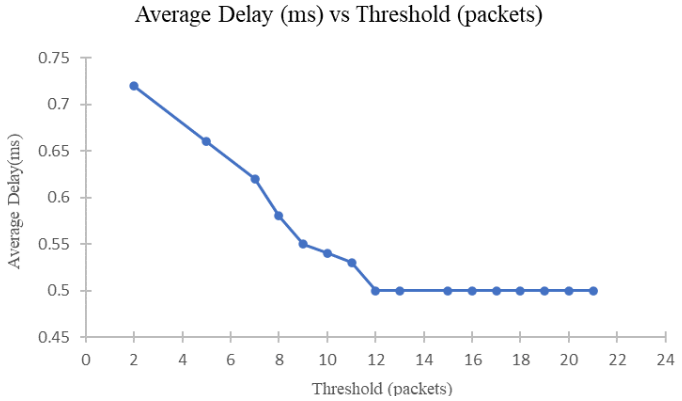

Figure 5.1: Average delay vs Threshold

The simulation initially analyses the trade-off between mmWave throughput (or, in the opposite case, link wastage) and average waiting time. Connection wastage is defined as the proportion of time slots in which packets are present in the system but the mmWave queue is empty and the mmWave link is available. Figure 5.2 depicts the trade-off between link wastage and average waiting time. According to the results, if there are a large number of packets added to the mmWave queue and the mmWave link becomes unavailable, a significant average delay occurs. On the other side, a conservative approach that causes link wastage owing to a shortage of packets in the mmWave queue is not desired.

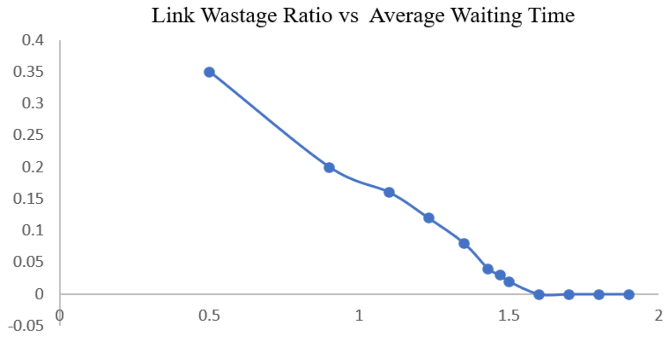

Figure 5.2: Tradeoff between delay and link wastage ratio

From that, the advantages of the Sub6 GHz interface in combating the impacts of blockage and intermittent connection are introduced and compared to the delay performance in a stem with and without the Sub6 GHz. The suggested threshold-type regulation is used for systems operating at less than 6 GHz. There is no scheduling policy for the system without the Sub6 GHz server because the system only has a mmWave interface. The study is conducted with various arrival rates. According to the results presented in Figure 5.3, for a certain arrival rate, the benefits of the Sub6 GHz interface become more obvious when the unavailable state grows. Given example, for an arrival rate of = 60, the integrated architecture combined with the threshold-based policy can reduce delay by up to 70%.

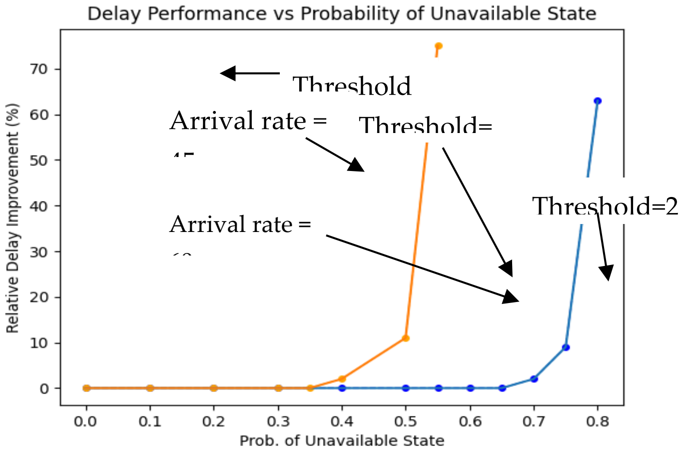

Figure 5.3: Delay vs Probability of unavailable state graph

CONCLUSION

In conclusion, integrated mmWave and sub-6GHz architecture for 5G cellular systems has been proposed in this study to diminish the negative effect of intermittent connectivity in mmWave and utilized the capacity in optimum. Deep Reinforcement Learning approach also has been adopted in this study to formulate the optimal scheduling policy for delay minimization. The results showed that Deep Reinforcement Learning give an excellent performance in determine the best policy for the integrated Sub6 GHz and mmWave architecture to combat blockage and dynamic nature of the mmWave communication.

REFERENCES

Bertsekas, Dimitri, and Robert Gallager. *Data Networks*.

Burghartz, J. et al. 1998. "RF Circuit Design Aspects of Spiral Inductors on Silicon." In *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, IEEE, 246–47, 443.

Elmezughi, Mohamed K., Thomas J. Afullo, and Nicholas O. Oyie. 2021. "Performance Study of Path Loss Models at 14, 18, and 22 GHz in an Indoor Corridor Environment for Wireless Communications." *SAIEE Africa Research Journal* 112(1): 32–45.

Frye, Robert C. 2001. "Integration and Electrical Isolation in CMOS Mixed-Signal Wireless Chips." *Proceedings of the IEEE* 89(4): 444–55.

"Integrating Sub-6 GHz and Millimeter Wave to Combat Blockage: Delay-Optimal

Scheduling | DeepAI." https://deepai.org/publication/integrating-sub-6-ghz-and-millimeter-wave-to-combat-blockage-delay-optimal-scheduling (February 21, 2021).

Kingma, Diederik P., and Jimmy Ba. 2014. "Adam: A Method for Stochastic Optimization." *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. https://arxiv.org/abs/1412.6980v9 (July 27, 2021).

Kurose, James F et al. 2013. *Computer Networking A Top-Down Approach Sixth Edition*.

Liu, Jia, Atilla Eryilmaz, Ness B. Shroff, and Elizabeth S. Bentley. 2017. "Understanding the Impacts of Limited Channel State Information on Massive MIMO Cellular Network Optimization." *IEEE Journal on Selected Areas in Communications* 35(8): 1715–27.

Lopez, Aida Vera et al. 2019. "Opportunities and Challenges of MmWave NR." *IEEE Wireless Communications* 26(2): 4–6.

Semiari, Omid, Walid Saad, Mehdi Bennis, and Merouane Debbah. 2019. "Integrated Millimeter Wave and Sub-6 GHz Wireless Networks: A Roadmap for Joint Mobile Broadband and Ultra-Reliable Low-Latency Communications." *IEEE Wireless Communications* 26(2): 109–15.

Watkins, Christopher J. C. H., and Peter Dayan. 1992. "Q-Learning." *Machine Learning 1992 8:3* 8(3): 279–92. https://link.springer.com/article/10.1007/BF00992698 (July 27, 2021).

"What Are Millimeter Waves? - Everything RF." https://www.everythingrf.com/community/what-are-millimeter-waves (December 19, 2020).