

RESEARCH ARTICLE

Optimizing Information Freshness in Dual-Connection Cellular Networks for Real-Time Applications

ABABAKR IBRAHIM RASUL¹ AND HAKEM BEITOLLAHI^{1,2}¹Department of Computer Science, Soran University, Soran 44008, Iraq²School of Computer Engineering, Iran University of Science and Technology (IUST), Tehran 13114-16846, Iran

Corresponding author: Hakem Beitollahi (hakem.beitollahi@visitors.soran.edu.iq)

ABSTRACT Emerging cellular networks integrate diverse technologies like millimeter wave (mmWave) to deliver high capacity but face challenges like blockage sensitivity. Dual connections (DC) enhance throughput but ensuring information freshness (Age-of-Information, AoI) is crucial for real-time applications. This paper proposes a novel approach for AoI optimization in DC cellular networks (4G/5G) with TCP multipath transport. We address the stochastic nature of AoI and channel conditions with a combined random and stochastic optimization approach. Modeling the system as a multi-queue system with LTE and mmWave connections, we propose a difference learning-based algorithm that dynamically selects the optimal queue for each packet. This eliminates the need for statistical information like packet arrival rates, enabling real-time adaptation. Compared to existing methods, our algorithm significantly reduces AoI under various network conditions (over 40%), demonstrating its effectiveness for real-time applications.

INDEX TERMS Age-of-Information, AoI, dual connectivity, cellular networks, 4G, 5G, MPTCP, real-time applications.

I. INTRODUCTION


The proliferation of real-time applications, such as self-driving cars and remote surgery, demands mobile networks with ultra-low latency and high information freshness [4], [7]. These applications require timely delivery of critical data that reflects the latest state of the environment (e.g., real-time sensor data from a moving vehicle). To address these demands, 4G/5G cellular networks with dual connections (DC) have been proposed [6], [21], [27]. In DC networks, a user equipment (UE) can connect to both a macro base station (MeNB) and a nearby small cell (SeNB) simultaneously, potentially increasing network capacity [6], [21], [27].

However, ensuring information freshness, quantified by Age-of-Information (AoI), remains a challenge in DC networks. This is due to the heterogeneous nature of LTE and mmWave connections (commonly used for DC), as well as

the dynamic channel variations that can impact information delivery [1], [15]. Traditional network performance metrics like delay or throughput do not directly address information freshness. Additionally, existing AoI optimization techniques in DC networks often struggle to adapt to these changing conditions and might rely on statistical information that might not be readily available.

We explore LTE-mmWave DC for real-time applications due to its high capacity and wider coverage compared to LTE. However, recent studies show significant limitations in outdoor-to-indoor (Otl) scenarios (e.g., [19]), highlighting the need for solutions that address dynamic channel quality for low-latency real-time communication.

To address these challenges, this paper proposes a novel approach for optimizing information freshness in DC networks with Multipath TCP (MPTCP). We leverage the capabilities of MPTCP for traffic load sharing across both connections and propose a reinforcement learning-based approach for dynamic traffic load distribution. This approach allows the network to adapt to real-world uncertainties and

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Khalil Afzal .

optimize information freshness in a dynamic and stochastic environment. Our work makes three key contributions as follows:

- **Focus on Information Freshness (AoI) in Dual Connections:** We address information freshness (AoI) in dual connections, a new perspective compared to the traditional focus on metrics like delay or throughput.
- **Reinforcement Learning for Dynamic Traffic Load Distribution:** We propose a novel approach for optimizing traffic load distribution in a dynamic and stochastic environment. Unlike previous works reliant on known information or statistical data, our approach utilizes reinforcement learning to adapt to real-world uncertainties such as channel variations and traffic fluctuations. This enables us to achieve an optimal policy for maximizing information freshness in dual connections.
- **Technology-Independent Problem Model:** Additionally, our problem model is independent of the specific technology used for connections, allowing for broader applicability across different cellular network generations.
- **Performance Evaluation:** Our solution demonstrates significant improvements in AoI minimization compared to baseline scheduling algorithms under different environmental conditions (above 40%), paving the way for more efficient and reliable data delivery in real-time applications over mobile networks.

The rest of the paper is organized as follows. Section II presents a background on the literature of the work. Section III discusses related work. Section IV presents the proposed method for the traffic load sharing problem in dual connection sub-flows. Section V evaluates the proposed method and finally Section VI concludes the paper.

II. BACKGROUND

A. DUAL CONNECTIVITY (DC)

Dual connectivity (DC) is a key technology in cellular networks that allows a user equipment (UE) to connect and communicate with two base stations (eNBs) simultaneously [27]. This capability, typically implemented in active/active mode for LTE networks, offers several advantages, including significantly improved per-user throughput, enhanced reliability due to redundancy, and smoother handovers during cell transitions. There are two main types of DC implementations: active/active mode, where both connections (typically to a Master eNB (MeNB) and a Secondary eNB (SeNB)) are used for data transmission simultaneously, and active/standby mode, where one connection is primary and the other serves as a backup [1]. The MeNB and SeNB operate on different carrier frequencies.

It's important to note that while DC provides multiple paths for data transmission, these paths are interconnected by the X2 interface, which is a backhaul connection between the MeNB and SeNB [1], [27]. This X2 interface might introduce

latency due to being non-ideal (e.g., using microwaves) compared to the direct connection between the UE and the base stations.

Dual connectivity (DC) is not limited to LTE networks and is also applicable in 5G deployments [1]. This technology allows a user equipment (UE) to connect to two base stations simultaneously, offering potential benefits like improved throughput and reliability. In the context of 5G, a particularly relevant application of DC is connecting to both a 4G (LTE) network and a 5G network concurrently [14]. This capability is especially useful for Non-Standalone (NSA) 5G deployments, which leverage the existing LTE core network while utilizing the new 5G NR air interface for data transmission. Figure 1 illustrates DC technology in cellular networks.

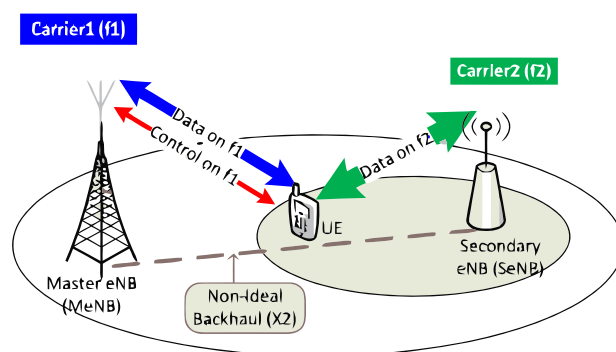


FIGURE 1. Dual connectivity (DC) technology in cellular networks.

Dual connections can be established using different technologies. These technologies include 4G (LTE), 5G (NR), and WiFi [1], [15], [21]. The primary and secondary connections can be configured in various ways. Among the, the four main configurations are:

- **LTE-LTE:** This configuration utilizes two LTE connections, potentially from different base stations.
- **LTE-WiFi:** This configuration combines an LTE connection with a WiFi connection, offering a broader range of connectivity options.
- **LTE-mmWave:** This configuration utilizes an LTE connection as the primary connection and a mmWave connection as the secondary connection, leveraging the strengths of both technologies for high capacity and wider coverage.
- **mmWave-mmWave:** This configuration uses two mmWave connections, potentially from different access points, offering extremely high capacity but requiring careful management due to the sensitivity of mmWave signals.

In this research, we focus specifically on the **LTE-mmWave mode**. This mode is particularly interesting because it combines the wider coverage and reliability of LTE with the high capacity of mmWave, making it well-suited for real-time applications that require both high data rates and reliable information delivery.

B. MULTIPATH TCP (MPTCP)

The availability of multiple paths provided by DC paves the way for utilizing Multipath TCP (MPTCP), a transport layer protocol specifically designed to manage traffic across these paths [5]. Unlike traditional TCP, which utilizes a single path, MPTCP offers significant advantages by enabling the creation of subflows. These subflows are essentially independent data streams that can be distributed across available network connections. This capability allows MPTCP to address challenges like varying delays and congestion on individual paths by dynamically adjusting traffic distribution [26]. MPTCP implements various functionalities to manage these subflows effectively. It can discover and select appropriate paths for each subflow, perform congestion control to avoid overwhelming any single path, and ensure in-order delivery at the receiver despite the potential for reordering of packets across different paths [26].

C. TRAFFIC SCHEDULING IN MPTCP

After establishing subflows and managing available paths, MPTCP needs to make informed decisions about how to distribute packets across these paths. Several factors influence these scheduling decisions, including the latency (round-trip time), bandwidth, and packet loss rate experienced on each subflow [24]. MPTCP utilizes various scheduling mechanisms to optimize traffic distribution based on these factors. Two commonly used scheduling mechanisms are

- **Lowest-RTT-First:** This is the default mechanism in the Linux kernel and prioritizes subflows with the lowest round-trip time (RTT). Packets are preferentially sent on paths with the shortest delay, aiming for faster delivery [8].
- **Round Robin:** This mechanism distributes packets equally across all subflows in a circular fashion, regardless of network conditions. While offering fairness, it might send packets on congested paths, potentially impacting performance [8].

III. RELATED WORK

Reference [10] maximizes throughput via window size adjustment for mmWave links under unfavorable conditions. However, it relies on fixed rates and RTTs, which are unrealistic in dynamic environments. Their focus on minimum RTT might prioritize low-delay paths over high-throughput ones. Our Q-learning approach addresses these limitations by dynamically adapting to real-time conditions, optimizing for AoI minimization instead of just throughput.

Reference [23] proposes an MDP-based flow control for energy efficiency and timely transmission, considering vibration and mmWave blockage. While it reduces delay difference and avoids blockage penalty, its distance-based blockage estimation is inaccurate. Our Q-learning approach learns from real-time information for better blockage probability estimation, leading to improved AoI minimization.

Reference [2] proposes a heuristic MPTCP scheduling algorithm for energy-efficient throughput. It balances energy consumption with maximizing information transfer. However, its static approach might not adapt well to dynamic channels. Our Q-learning method addresses this by continuously learning from real-time information, allowing for dynamic adaptation and AoI minimization while considering both energy and channel dynamics.

Reference [17] proposes traffic scheduling for downlink in dual-connection scenarios, maximizing utilized bandwidth while meeting constraints. However, it focuses solely on throughput, neglecting real-time needs. Our Q-learning approach considers AoI minimization alongside throughput, leading to better performance for real-time applications.

Reference [12] proposes a DQN-based MPTCP scheduling algorithm for LTE-WiFi connections to improve throughput. It uses a simple neural network and reward based on ACKs, achieving higher throughput compared to baselines. However, it neglects AoI and complex channel dynamics. Our approach addresses this by using a more comprehensive reward function (AoI, delay, throughput) and potentially exploring deeper neural networks for better learning.

Reference [11] proposes asynchronous RL for multipath congestion control in MPTCP. It uses learned rules to adapt congestion window sizes based on network conditions. This approach is effective in heterogeneous networks and avoids the “curse of dimensionality” by employing a rule-based method. A key strength is its asynchronous design, enabling real-time decision-making without delays.

Reference [25] uses deep RL for MPTCP packet scheduling, considering QoS features (throughput, RTT, loss) and offering asynchronous training for real-time adaptation. However, it shows lower performance in out-of-order packet queue length compared to Round Robin. This could be due to exploration-exploitation trade-offs, network architecture, or reward function design.

Previous approaches do not fully consider the dynamic nature of mmWave channel quality. Recent research has shown that device thermal limitations can significantly impact sustained mmWave throughput due to rising skin temperature [18]. This can lead to reductions in available mmWave channels and potential fallback to 4G, impacting achievable throughput and potentially increasing latency. Our work addresses this gap by proposing an MPTCP scheduling approach that incorporates real-time thermal information to adapt scheduling decisions and minimize AoI in mmWave deployments with dynamic channel quality.

IV. PROPOSED METHOD

Many existing works on traffic load sharing in dual-connectivity scenarios neglect a crucial metric for real-time applications: Age of Information (AoI). AoI measures the time elapsed since the last update was received, and minimizing AoI ensures the information utilized by applications remains fresh and timely. Traditional approaches for traffic optimization might struggle in dynamic network

environments where channel conditions, packet arrival rates, and queue lengths constantly change.

To address this challenge, we propose a novel approach that leverages reinforcement learning (RL) to optimize traffic load sharing for minimizing average AoI in dual-connectivity MPTCP networks. RL enables our method to learn and adapt to real-time network conditions, making it well-suited for this dynamic problem.

Our approach utilizes a Markov Decision Process (MDP) framework to model the traffic load sharing problem. Within this framework, a reinforcement learning agent observes the network state, takes actions (adjusting traffic distribution across paths), and receives rewards based on the resulting AoI. One specific RL technique we employ is Q-learning, which allows the agent to learn an optimal policy for traffic load sharing that minimizes the average AoI.

A. SYSTEM MODEL AND ASSUMPTIONS

We consider a dual-connectivity scenario with an active/active mode architecture, where a self-driving car acts as the user equipment (UE). The UE establishes connections with both a master eNB (5G) and a secondary eNB (LTE) as shown in Figure 2. Packets representing sensor data or control signals are generated by the UE according to a Poisson distribution with a mean arrival rate of λ . A Traffic Load Sharing Module is responsible for determining how to distribute these packets across queues for the LTE and mmWave transmission paths. The capacities of these channels are dynamic, leading to variable data output rates at any given time. In the following sections, we will elaborate on the specific system components and the information assumptions made for our proposed method.

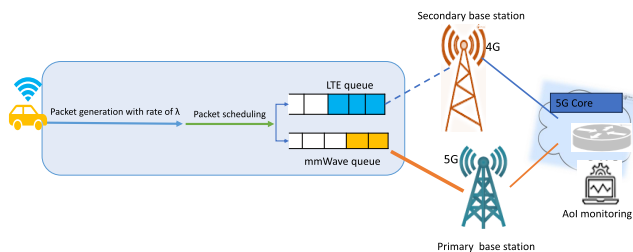


FIGURE 2. The system model.

1) THE WIRELESS LTE CHANNEL MODEL

The quality of both LTE and mmWave channels can vary significantly due to factors like noise, path loss, fading, and shadowing. To model the variable nature of the LTE channel, we adopt the well-established 3GPP Spatial Channel Model (SCM) with parameters suitable for our urban macro-cellular deployment scenario [22]. This model captures the effects of path loss, multipath propagation, and spatial correlation between antennas. The channel gain between the UE and the eNB antennas in the LTE channel can be expressed as [22]:

$$h_t = \sqrt{\zeta} \times D^{-\alpha} \quad (1)$$

where h_t represents the channel coefficient between the UE and antenna t , D represents the distance between the UE and the antenna, α is the path loss exponent, a parameter depending on the specific environment (e.g., urban macro), and ζ accounts for shadowing effects and is typically modeled as a log-normal random variable.

2) THE WIRELESS mmWave CHANNEL MODEL

The mmWave channel exhibits distinct characteristics compared to the LTE channel. To model these characteristics, we adopt a two-level Finite State Markov Chain (FSMC) model [22].

- **Level 1: LoS/NLoS:** The first level represents the large-scale fading effects and determines whether the channel is in a Line-of-Sight (LoS) or Non-Line-of-Sight (NLoS) state. The transition between these states depends on factors like blockage and mobility.
- **Level 2: Small-Scale Fading:** The second level models the small-scale fading within the established LoS or NLoS condition from Level 1. This level typically employs a specific fading model, such as the Rician fading model for NLoS conditions, to capture the rapid fluctuations in the received signal strength.

By using a two-level FSMC model, we can effectively capture the key aspects of the mmWave channel, including the impact of blockage and mobility on LoS/NLoS conditions, as well as the small-scale fading effects that influence signal strength.

Due to the significant impact of blockage in mmWave channels, the antenna can experience three main channel states relative to the user equipment (UE):

- **Line-of-Sight (LoS):** A direct line of sight exists between the UE and the antenna.
- **Non-Line-of-Sight (NLoS):** The signal path is obstructed, but a usable path still exists.
- **Outage:** The link between the UE and the antenna is completely blocked.

To model the transitions between these states over time, we employ a three-state Markov chain. In a Markov chain, the probability of transitioning to a new state depends only on the current state, and not on the history of previous states. The transition probabilities between LoS, NLoS, and Outage states are defined by a transition probability matrix (\mathbb{P}_y) as follows [16].

$$\mathbb{P}_y = \begin{bmatrix} P_{LoS|LoS} & P_{LoS|NLoS} & P_{LoS|Out} \\ P_{NLoS|LoS} & P_{NLoS|NLoS} & P_{NLoS|Out} \\ P_{Out|LoS} & P_{Out|NLoS} & P_{Out|Out} \end{bmatrix}$$

The specific values in the \mathbb{P}_y matrix depend on various factors such as the mobility of the UE, the dynamics of blockage in the environment, and the width of the antenna beam.

Path loss, which refers to the signal attenuation as it travels from the base station to the user equipment (UE), plays a crucial role in cellular network performance. In fifth-generation (5G) networks, path loss characteristics can vary

depending on the propagation environment [22] (line-of-sight or non-line-of-sight) [Eq. (2)].

$$PL(R) = \begin{cases} (1 - y_t)C_L R^{-\alpha_L} + y_t C_N R^{-\alpha_N} & y_t = 0 \text{ or } 1 \\ \infty & y_t = 2 \end{cases} \quad (2)$$

where, R denotes the distance between the base station and the UE (in meters), y_t is an indicator variable ($y_t = 0$ indicates line-of-sight (LOS) propagation, $y_t = 1$ indicates non-line-of-sight (NLOS) propagation, and $y_t = 2$ is an invalid value), C_L and C_N are constants associated with LOS and NLOS paths, respectively (in dB), and α_L and α_N are path loss exponents for LOS and NLOS paths, respectively.

Building upon the path loss model described by Eq. (2), we can now determine the transmit power (P_t^{Tx}) required to send a packet during a time slice as follows [13]

$$P_t^{Tx} = (2^{M/\tau W} - 1) \frac{N_0}{|g_t|^2 G_s^{max} G_b^{max} PL(R)} \quad (3)$$

where, M represents the number of bits transmitted per symbol, τ signifies the time slot duration (in seconds), N_0 represents the noise power spectral density (in Watts/Hz), g_t represents the channel gain between the user equipment and the base station, G_s^{max} and G_b^{max} are the maximum antenna gains at the base station and user equipment, respectively, $PL(R)$ is the path loss at distance R from the base station (in dB), as defined in Eq. (2).

As mentioned in [16], the link between the user equipment (UE) and the base station experiences fading, which refers to the random fluctuations in received signal strength. To account for this phenomenon, a Nakagami fading model with different parameters for line-of-sight (LOS) and non-line-of-sight (NLOS) propagation is employed.

3) TRAFFIC GENERATION MODEL

Traffic generation at the user equipment (UE) follows a Poisson probability distribution. This means that packets are generated at random points in time according to a discrete-time process with $1ms$ intervals ($t = 0, 1, 2, \dots$). At each time slot, we need to decide whether to place the newly generated packet(s) in the LTE or mmWave queue based on the current network conditions.

The number of packets arriving in the n th time slot, denoted by K_n , follows a Poisson distribution with mean arrival rate λ [3]. This arrival process models the random nature of packet generation by the UE. The size of each packet is set to $6MB$, which is a typical value for data packets containing sensor information or control signals.

Queuing Model in Base Stations: Packets generated by the user equipment need to be placed in one of two queues: LTE or mmWave. We assume that both queues have unlimited capacity and their lengths change dynamically. It's crucial to maintain a balance between arrival and departure rates to avoid excessive queuing and information delays. In each time slot (n), the lengths of the LTE queue (WnM) and

mmWave queue (WnS) are updated based on the following equations [9]:

$$W_{n+1}^M = I_n^M + W_n^M - C_n^M \quad (4)$$

$$W_{n+1}^S = I_n^S + W_n^S - C_n^S \quad (5)$$

In these equations:

- I_n^M and I_n^S represent the number of packets arriving in the mmWave and LTE queues, respectively, during time slot n .
- C_n^M and C_n^S represent the number of packets departing from the mmWave and LTE queues, respectively, during time slot n . These outgoing packets are calculated using Equations (6) and (7).
- R_n^M and R_n^S represent the output rates of the mmWave and LTE queues, respectively. These rates depend on factors like channel capacities and the traffic load sharing decisions made by the reinforcement learning agent (details on these decisions will be provided later).
- L represents the size of each packet (6 MB as specified earlier).
- t represents the duration of each time slot (1 ms as specified earlier).

$$C_n^M = \frac{R_n^M}{L} * t \quad (6)$$

$$C_n^S = \frac{R_n^S}{L} * t \quad (7)$$

By dynamically adjusting the routing of packets between the queues based on real-time network conditions, we aim to minimize the queue lengths and consequently, the Age of Information (AoI) experienced by the applications.

4) THE AoI MODEL

To measure the freshness of the data received at the destination, we employ the Age of Information (AoI) criterion. AoI represents the time elapsed since the last successful packet reception at the destination.

If u_t denotes the generation time of the most recently received packet at time t , the AoI of the source node at the destination is defined as a random process, denoted by ϕ_t (Equation (8)). A lower AoI value signifies fresher information available at the destination.

$$\phi_t = t - u_t \quad (8)$$

The symbol ϕ_t represents the AoI value at the beginning of time slot t . This value belongs to a set $\phi = 1, 2, \dots, \phi_{max}$. The maximum AoI, ϕ_{max} , defines an upper limit for the information age and is determined based on the specific requirements of the application (e.g., latency constraints).

Equation (9) describes how the AoI evolves over time:

$$\phi_{t+1} = \begin{cases} \min(\phi(t) + 1, \phi_{max}) & a_t = 0(\text{idle}) \\ 1 & a_t = 1(\text{transmit}) \end{cases} \quad (9)$$

In Equation (9), a_t is a binary variable indicating the system state in time slot t :

- $a_t = 0$ (idle): No packet was successfully transmitted in the current time slot. In this case, the AoI is incremented by 1, capped by the maximum AoI (ϕ_{max}).
- $a_t = 1$ (transmit): A packet was successfully transmitted in the current time slot. This resets the AoI to 1, reflecting the arrival of fresh information.

A table summarizing the symbols used in our system modeling is provided in Table 1.

TABLE 1. Symbols used in system modeling and their description.

description	Symbol
Length of the time period	τ
The distance from the origin node to the base stations	R
package size	L
mmWave channel bandwidth	W^M
LTE channel bandwidth	W^S
Carrier frequency dependent constant in LoS state	C_L
Carrier frequency dependent constant in NLoS state	C_N
Path loss coefficient in LoS state	α_L
Path loss coefficient in NLoS state	α_N
Path loss between source node and base stations with link length R	$PL(R)$
Additional noise power	N_0
The main corner gain of the origin node	G^{max}
Main corner gain gNB	G_b^{max}
Small scale fading gain	g_t
Power required to send a packet in time slice t	P^{Tx}
Number of incoming packets to the mmWave queue	I_n^M
Number of incoming packets to the LTE queue LTE	I_n^S
Number of outgoing packets from the mmWave queue	C_n^M
Number of outgoing packets from the LTE queue	C_n^S
Number of packets in mmWave queue	w_n^M
Number of packets in the LTE queue	w_n^S
Output rate from the mmWave queue	R_n^M
The exit rate from the LTE queue	R_n^S
channel gain coefficient between transmitter and receiver	$h_t \in H = \{0, 1, \dots, H\}$

B. PROBLEM FORMULATION

In this section, we formulate the problem of traffic load sharing between the LTE and mmWave channels as a single Markov decision process (MDP) framework. An MDP provides a suitable framework for modeling sequential decision-making problems under uncertainty.

The MDP is defined by a quadruple (Φ, A, P, C) :

1) **State Space (Φ):** The state space, denoted by Φ , represents all possible states of the system at any given time slot. Each state includes relevant information such as:

- The current AoI values for both the LTE queue (ϕ_{LTE}) and the mmWave queue ($mmWave$).
- The number of packets in the LTE queue (w_n^S) and the mmWave queue (w_n^M).

2) **Action Space (A):** The action space, denoted by A , represents the set of possible actions, $A = \{0, 1\}$, that the decision-making entity (the UE) can take in each state. These actions correspond to how the UE manages the incoming traffic:

- $a_t = 0$: The UE can choose to place a new packet in the LTE queue.
- $a_t = 1$: The UE can choose to place a new packet in the mmWave queue.

The number of incoming packets to the queue in equations (10) and (11) is no longer simply represented by I_n^M and I_n^S , respectively (remember equations 4 and 5). This is because two influential factors affect the number of packets entering the queue:

- **Poisson Random Process:** This process captures the inherent randomness in packet arrivals, often modeled using an arrival rate parameter.
- **Selected Process:** This refers to a specific mechanism or policy that governs how packets are directed towards either the mmWave (M) or LTE (S) queue.

To account for these factors, the equations are updated as follows:

$$W_{n+1}^M = a_t K_t + W_n^M - C_n^M \quad (10)$$

$$W_{n+1}^S = (1 - a_t) K_t + W_n^S - C_n^S \quad (11)$$

where, a_t represents the probability of a packet being directed towards the mmWave queue based on the selected process. $(1 - a_t)$ represents the probability of a packet entering the LTE queue. K_t is the arrival rate of packets at time step t , likely modeled using a Poisson distribution.

3) **Transition Probability (\mathbb{P}):** The transition probability, denoted by \mathbb{P} , defines the probability of transitioning from one state (Φ_t, a_t) to another state $(\Phi_t + 1)$ given that action a_t is taken in state Φ_t according to equation 12. These probabilities depend on various factors such as:

- The effectiveness of the chosen action (e.g., successful packet transmission in the chosen queue).
- The channel capacities of the LTE and mmWave channels at the next time slot.
- The arrival of new packets in the next time slot.

$$\mathbb{P}(s_{t+1} | s_t, a_t) = \mathbb{P}_\phi(\phi_{t+1} | \phi_t, a_t) \quad (12)$$

4) **Cost Function (C):** The cost function, denoted by C , assigns a cost or reward to each state-action pair (Φ_t, a_t) . The cost function is typically designed to reflect the goal of the system. In our case, we aim to minimize the overall Age of Information (AoI) experienced by the application at the destination. The cost function should penalize actions that lead to high AoI values in the queues. The cost function is calculated as equation 13

$$C_t(s_t, a_t) = \phi_t \quad (13)$$

It is clear that the amount of ϕ_t depends to the system state and the chosen action.

Goal of Optimization: Since Age-of-Information (AoI) is a stochastic (random) process, the optimal approach for minimizing it is to calculate its long-term average.

This average cost is denoted by the symbol \bar{C} (Eq. 14).

$$\bar{C} = \lim_{n \rightarrow \infty} \frac{1}{N} \mathbb{E} \sum_{t=1}^N C(s_t, a_t) \quad (14)$$

Equation (Eq. 14) expresses the average cost as the limit, as the number of time steps (N) approaches infinity, of the average cost across those time steps. The summation (\sum) iterates through each time step (t) from 1 to N , considering the cost ($C(s_t, a_t)$) associated with the state (s_t) and action (a_t) taken at that time step. The expectation (\mathbb{E}) operator accounts for the randomness inherent in the AoI process.

Optimal Policy: The objective for the source node is to determine the optimal policy (π^*). This policy dictates the action (a_t) to be taken in each state (s_t) to minimize the long-term average cost function (\bar{C}_π) across all possible conditions. Equation (Eq. 15) formalizes this optimization problem. The arg min notation indicates that we are searching for the policy (π) that minimizes the average cost (\bar{C}_π).

$$\pi^* = \operatorname{argmin}_\pi \bar{C}_\pi \quad (15)$$

We will delve into the specifics of using Bellman's equations to calculate the optimal policy (π^*) in the next section.

C. BELLMAN EQUATIONS AND OPTIMAL POLICY

Bellman equations are a cornerstone of dynamic programming, a powerful technique for solving optimization problems involving sequential decisions. These equations express the optimal value of a decision problem at a specific state as the sum of the immediate reward (or cost) and the discounted expected value of the remaining problem.

In the context of Markov Decision Processes (MDPs), where the state of the system depends only on the most recent state, not the entire history, Bellman equations typically employ state-action value functions.

State-Action Value Function (Eq. 16): This function, denoted by $Q^\pi(s, a)$, represents the expected long-term cost incurred by taking action a in state s while following policy π . It is defined by Equation (16):

$$Q^\pi(s, a) = [c(s, a) - \bar{C}^{\pi^*}] + \sum_{s' \in S} \mathbb{P}(s'|s, a) Q^\pi(s', a) \quad (16)$$

- $c(s, a)$: Immediate cost of taking action a in state s .
- \bar{C}^{π^*} : Minimum average cost achievable with the optimal policy (π^*).
- $\sum_{s' \in S} \mathbb{P}(s'|s, a) Q^\pi(s', a)$: Expected future cost based on the transition probabilities ($\mathbb{P}(s'|s, a)$) of reaching state s' after taking action a in state s , considering the value of following policy π in that future state.

Finding the Optimal Policy (Eq. 17 and Eq. 18): The optimal policy, denoted by π^* , dictates the action that minimizes the expected long-term cost in each state. We can find the optimal policy using the state-action value function and Bellman equations.

Equation (17) is a special case of Eq. (16) where the policy π is the optimal policy (π^*). Here, $Q^{\pi^*}(s, a)$ represents the expected long-term cost under the optimal policy.

$$Q^{\pi^*}(s, a) = [c(s, a) - \bar{C}^{\pi^*}] + \sum_{s' \in S} \mathbb{P}(s'|s, a) Q^{\pi^*}(s', a) \quad (17)$$

Finally, Equation (18) formally defines the optimal policy selection:

$$\pi^*(s) = \operatorname{argmin}_{a \in A_s} Q^{\pi^*}(s, a) \quad (18)$$

This equation states that the optimal action in a given state s is the one that minimizes the state-action value function $Q^{\pi^*}(s, a)$ under the optimal policy itself (π^*). In simpler terms, the optimal policy chooses the action that leads to the lowest expected long-term cost when following the optimal policy.

D. REINFORCEMENT LEARNING AS A SOLUTION

Limitations of Model-Based Methods:

- 1) **Model Inaccuracy:** Building a perfect model of the environment is often impossible. Changes in the environment (e.g., weather, traffic patterns) can quickly render the model invalid. (Eq. 16)
- 2) **Modeling Difficulty:** In many real-world scenarios, accurately modeling the probabilistic structure of the system can be highly complex or even impractical. (Eq. 17)
- 3) **Limited Statistical Knowledge:** Obtaining accurate statistical data about random processes in the operational environment (e.g., data arrival rates, channel quality variations) can be challenging. (Eq. 18)

Given these limitations, reinforcement learning (RL) emerges as a compelling approach. RL algorithms can learn optimal policies **without** requiring a complete or accurate model of the environment. They achieve this by interacting with the environment, receiving rewards or penalties for their actions, and using these experiences to improve their decision-making over time.

1) POLICY LEARNING THROUGH Q-LEARNING

This section presents a policy learning algorithm based on the standard Q-learning method. Q-learning is a popular RL technique that estimates a state-action value function. This function helps the agent learn which actions are most beneficial in different states. Algorithm 1 shows pseudocode of the Q-Learning algorithm.

Line 1: the first line sets up the initial conditions for the learning process by initializing the Q-value table (representing the agent's knowledge about the environment) and potentially a tracking mechanism for exploration (visit count table), along with a counter for the number of learning steps taken.

Lines 3-8: The loop repeatedly executes these steps, enabling the agent to interact with the environment,

Algorithm 1 The Pseudocode of Q-Learning

```

1: Initialization:  $\hat{Q}(s, a) = 0 \forall s, a$ ;  $visit(s, a) = 0 \forall s \in S, a$ ;  $t = 0$  (initial time index)
2: Main Learning Loop:
3: while  $t < \text{Max Iter Num}$  do
4:   Take action from the action set  $A$  in  $\epsilon$ -greedy fashion;
      
$$a_t \leftarrow \begin{cases} \arg \min \hat{Q} & \text{with probability } 1 - \epsilon \\ \text{a random action} & \text{with probability } \epsilon \end{cases}$$

5:   Compute  $C(s_t, a_t)$  by 11;
6:   Observe the next system state  $s_{t+1}$ ;
7:   Compute  $\hat{Q}_{t+1}(s_t, a_t)$  by 18
8:    $t = t + 1$ ;
9: end while

```

learn from experiences (rewards and state transitions), and gradually refine its estimations of Q-values. This iterative process ultimately guides the agent towards learning an optimal policy, which specifies the best action to take in each state to maximize cumulative reward.

Line 4: The ϵ -greedy policy helps the agent decide whether to send a packet through the LTE or mmWave channel, balancing the need to explore both options (potentially discovering which is more reliable or efficient) with the desire to exploit its current understanding and choose the channel with the lower expected cost (represented by the Q-values).

Line 5: This line in the Q-learning pseudocode likely calculates the reward received for taking the chosen action (a_t) in the current state (s_t). The reward function (13) directly translate AoI values into rewards. Lower AoI (meaning fresher information) could lead to higher rewards, and vice versa.

Line 6: This line focuses on observing the overall state transition resulting from the action. This information, along with the received reward, is then used to update the Q-values in line 7, allowing the agent to learn how its actions impact the system's behavior and ultimately influence the AoI.

Line 7: performs a crucial step in the Q-learning process. By updating the Q-values based on experiences (rewards, state transitions), the agent gradually learns which actions are most beneficial in different states, leading it towards an optimal policy for minimizing AoI.

Lines 8 and 9: manage the loop's progress and termination. The time index is incremented within each iteration, and the loop continues as long as the maximum number of iterations hasn't been reached. Once the termination condition is met, the loop exits, and the Q-learning algorithm has completed its learning process.

2) LEARNING THE STATE-ACTION VALUE FUNCTION

To use the Q-learning algorithm in the first time slice, the source node initializes the Q-value function ($Q(s, a)$) with zeros for all possible state-action pairs in the first iteration.

In the next step in each iteration step of the algorithm according to equation 19: The source node being in the current state (s_t), choosing an action (a_t) using the ϵ -greedy policy, calculating the immediate cost ($C_t(s_t, a_t)$) associated with the chosen action in the current state, and finally, observing the next state (s_{t+1}) resulting from the action.

$$\hat{Q}_{t+1}(s_t, a_t) \leftarrow (1 - \mu_t)\hat{Q}_t(s_t, a_t) + \mu_t(C(s, a) - \bar{C}^{\pi^*} + \min_{a_{t+1} \in A_{s_{t+1}}} \hat{Q}_t(s_{t+1}, a_{t+1})) \quad (19)$$

In equation 19, the optimal cost function (\bar{C}^{π^*}) would represent the average cost achieved by the optimal policy. However, calculating this value beforehand is generally not feasible for complex systems.

Approximating the Optimal Cost: The solution proposed here is to replace the unknown optimal cost (\bar{C}^{π^*}) with an estimate of the Q-value for a fixed state-action pair ((s^*, a^*)). This pair acts as a reference point, and the true optimal cost is approximated by the limit of the Q-value at this reference point as the learning process progresses ($\lim_{t \rightarrow \infty} \hat{Q}_t(s^*, a^*)$).

Q-Learning Update Rule: Equation 20 shows the specific Q-learning update rule used in our application.

$$\hat{Q}_{t+1}(s_t, a_t) \leftarrow (1 - \mu_t)\hat{Q}_t(s_t, a_t) + \mu_t(C(s, a) - \hat{Q}_t(s^*, a^*) + \min_{a_{t+1} \in A_{s_{t+1}}} \hat{Q}_t(s_{t+1}, a_{t+1})) \quad (20)$$

It incorporates the following elements:

- Previous Q-value: $\hat{Q}_t(s_t, a_t)$
- Learning rate: $(1 - \mu_t)$ (incorporates a decaying learning rate μ_t)
- Reward (replaced by cost here): $C(s_t, a_t)$
- Reference Q-value: $\hat{Q}_t(s^*, a^*)$
- Minimum Q-value of next state actions: $\min_{a_{t+1} \in A_{s_{t+1}}} \hat{Q}_t(s_{t+1}, a_{t+1})$.

Learning Rate: The text emphasizes the importance of a decreasing learning rate (μ_t) over time. This ensures that the agent gradually converges to a stable Q-value function. Equation 21 specifies the standard conditions for the learning rate function:

$$\mu_t \rightarrow 0, t \rightarrow \infty, \sum_t (\mu(t))^2 < \infty \quad (21)$$

where,

- $\mu_t \rightarrow 0$ as $t \rightarrow \infty$: The learning rate approaches zero asymptotically as the learning progresses.
- $\sum_t (\mu(t))^2 < \infty$: The sum of the squared learning rates over time must be finite.

Adaptive Learning Rate: Equation 22 defines a specific example of a decreasing learning rate function based on the visit count ($visit_t(s, a)$) for each state-action pair ((s, a)).

$$\mu_t = \frac{1}{1 + (visit_t(s, a))^{0.65}} \quad (22)$$

where,

- $visit_t(s, a)$: This value represents the number of times the specific state-action pair has been encountered during learning.

- The learning rate (μ_t) is calculated with a denominator that increases as the visit count grows to a power of 0.65. This effectively reduces the learning rate as the agent encounters a state-action pair more frequently.

V. PERFORMANCE EVALUATION

This section evaluates the performance of our proposed Q-learning based MPTCP scheduling approach through simulations. We compare its effectiveness in minimizing Age of Information (AoI) with Round Robin and RTT-based scheduling algorithms.

A. SIMULATION SETTINGS AND PARAMETERS

We simulate the scenario described in Section IV-A and our protocol using the MATLAB environment. The origin node connects to two stations, representing a fifth-generation (mmWave) and fourth-generation (LTE) antenna, as depicted in Figure 2. The distance to the antennas ranges from 100 meters to 300 meters, and the transmitted power varies between 30 and 35 dBm.

The surrounding environment is modeled as uncertain, with data packet generation and channel quality changing over time. The simulation operates in fixed 1-millisecond time intervals. During each interval, the source node produces data between 1 and 6 MB. Channel quality is assumed to be constant within a time interval but can change between intervals.

For the LTE channel, we model the channel gain factor using a Markov chain with 10 distinct quality states. The mmWave channel has three discrete states: Line-of-Sight (LoS), Non-Line-of-Sight (NLoS), and Outage. These states change randomly throughout the simulation. The LTE and mmWave channels have bandwidths of 20 MHz and 5 MHz, respectively, and the data rate adapts based on channel conditions.

Table 2 summarizes the key simulation parameters used in our proposed scenario. The table includes details about the network topology, channel models, and data generation rates.

TABLE 2. Simulation parameters.

Parameter definition	Symbol	Value	Unit
Primary station bandwidth	W^M	5	Mb/s
Secondary station bandwidth	W^S	20	Mb/s
Time period	τ	1	ms
The distance from the origin node to the stations	R	[100, 150, 200, 250, 300]	m
Packet size	M	[5, 5.2, 5.4, 5.6, 5.8, 6]	MB
Carrier frequency-dependent constant in LoS state	C_L	-60	dB
Carrier frequency dependent constant in NLoS state	C_N	-70	dB
Main corner gain of the origin node	G_s^{main}	10	dB
Main corner gain gNB	G_b^{main}	10	dB
Additional noise power in mmWave	N_{mmWave}	$-174(dBm/Hz) + 10\log W + NF$	dBm
Additional noise power in LTE	N_{LTE}	-50	dB
Noise figure	NF	10	dB
power	P_T^x	[30, 31, 32, 33, 34, 35]	
Path loss coefficient in LoS state	α_L	2.1	
Path loss coefficient in NLoS state	α_N	3.17	
Nakagami extinction coefficient in LoS	N_L	2	
Nakagami extinction coefficient in NLoS	N_N	3	

B. RESULTS AND ANALYSIS

Here, we examine how the Q-learning agent, which observes channel states, data rates, and other relevant information, learns an optimal scheduling policy to minimize AoI.

We compare its performance with the baseline algorithms under different environmental conditions, such as:

- **Varying Traffic Load:** We investigate how the algorithms perform under different data generation rates at the source node (e.g., 1-3 Mb vs 4-6 Mb).
- **Channel Quality Distribution:** We analyze the impact of the distribution of channel qualities within the LTE channel (e.g., more frequent high-quality states vs more frequent low-quality states).

1) CONVERGENCE ANALYSIS

Here, we analyze the convergence of the Q-learning algorithm used in our proposed method. Convergence refers to the ability of the algorithm to stabilize its estimates of the Q-value function as the learning progresses.

Specifically, we focus on the convergence of the Q-value for a specific state-action pair, denoted as (s, a) . Figure 3 indicates that the Q-value for (s, a) converges after approximately 800,000 learning iterations, reaching a value of 3.5.

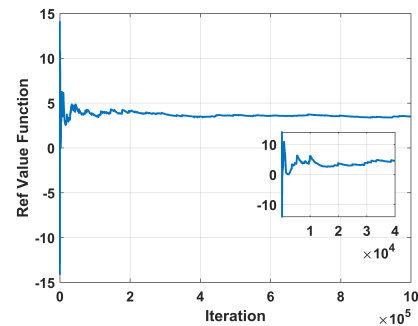


FIGURE 3. Convergence of the Q-value for state-action function.

2) IMPACT OF ARRIVAL RATE ON AVERAGE AoI

Figure 4 shows the average AoI for all three algorithms under increasing arrival rates (λ). As expected, the average AoI exhibits an upward trend with increasing λ , indicating higher congestion in the channel queue. The Q-learning algorithm demonstrates the best performance in terms of AoI minimization across all arrival rates. Results show that Q-learning significantly decreases AoI on average by 33% and 64% compared to Round Robin and the LL algorithm, respectively, for different arrival rates (e.g., packets per second). This can be attributed to its ability to dynamically

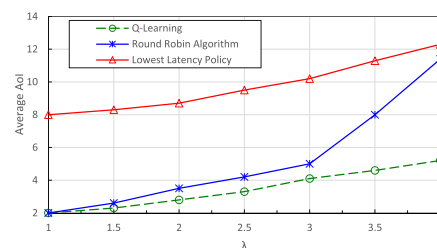


FIGURE 4. Average AoI vs. arrival rate.

adapt to congestion by selecting paths based on real-time information, including channel quality and queue lengths.

While the Round Robin algorithm performs worse than Q-learning at higher arrival rates, its performance is comparable at lower arrival rates. This could be due to the lack of adaptation of Round Robin under heavy traffic which leads to suboptimal performance.

The LL algorithm consistently exhibits the highest AoI among all three methods. This is likely because it prioritizes path delay without considering other crucial factors like packet arrival rate or buffer occupancy, leading to suboptimal scheduling decisions under congested network conditions.

3) IMPACT OF PACKET SIZE ON AVERAGE AoI

Figure 5 depicts the average Age-of-Information (AoI) for the three packet scheduling algorithms (Q-learning, Round Robin, and LL) as the packet size increases. As expected, we observe a slight increase in AoI for all algorithms with larger packets. This is because larger packets require more time to transmit over the network, leading to a slight delay in delivering fresh information to the destination.

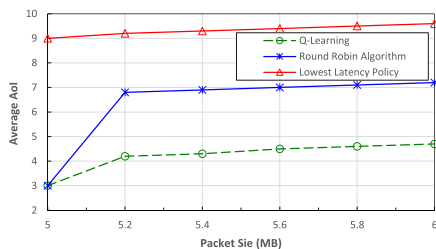


FIGURE 5. Average AoI vs. packet size.

However, the Q-learning algorithm demonstrates a slower increase in AoI compared to Round Robin and the LL algorithm as packet size grows. This is likely because the Q-learning agent can prioritize smaller packets when the queue is congested. By transmitting smaller packets first, the agent reduces the overall queue length and transmission delays, ultimately leading to lower AoI compared to Round Robin’s fixed scheduling strategy, which might treat all packet sizes equally.

The LL algorithm again exhibits the highest AoI among all methods due to its reliance on a single criterion (potentially path delay) that might not be optimal for minimizing AoI when packet sizes change.

4) IMPACT OF TRANSMIT POWER ON AVERAGE AoI

Figure 6 explores the impact of varying transmit power (30 dBm to 35 dBm) on average AoI for all three algorithms. As expected, we observe a downward trend in AoI with increasing power. This is because higher power translates to a stronger signal and improved channel quality, allowing for faster transmission rates and reduced queueing delays. Consequently, packets experience lower AoI.

The Q-learning algorithm demonstrates the most significant decrease in AoI compared to Round Robin at lower

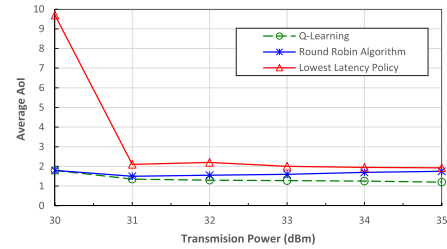


FIGURE 6. Average AoI vs. transmit power.

power levels. Notably, it outperforms Round Robin by 16% in terms of AoI reduction. This advantage can be attributed to the agent’s ability to exploit the improved channel quality at higher power. By strategically selecting paths and potentially prioritizing critical packets, the Q-learning agent can significantly reduce AoI compared to Round Robin’s fixed scheduling, which might not fully adapt to these dynamic channel conditions.

While the LL algorithm still exhibits the highest AoI overall, its AoI also decreases with increasing power due to the faster transmission rates. However, its reliance on a single criterion likely hinders its ability to optimally exploit the improved channel quality for AoI minimization compared to the more adaptive Q-learning approach.

5) IMPACT OF DISTANCE ON AVERAGE AoI

Figure 7 illustrates how the distance between the source node and the stations affects average AoI for all three algorithms. As the distance increases (up to 350 meters in our simulation), the average AoI exhibits an upward trend. This is because greater distances lead to signal attenuation and potentially higher bit error rates. Consequently, transmissions take longer, queues grow larger, and ultimately, AoI increases.

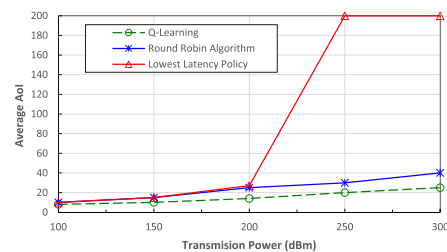


FIGURE 7. Average AoI vs. distance.

Interestingly, in this scenario, the Q-learning and Round Robin algorithms demonstrate similar performance in terms of AoI. This might be because both algorithms prioritize successfully transmitting packets even with the increased delay caused by larger distances. Q-learning’s ability to adapt to varying channel conditions might not be as advantageous in this case compared to Round Robin’s simpler scheduling, as both methods might struggle to find optimal paths under significant signal degradation. Despite similar performance at larger distances, it’s important to remember that Q-learning

still outperforms Round Robin on average by 21%. This emphasizes its overall effectiveness in optimizing AoI across different channel conditions, particularly when signal degradation is less severe.

The LL algorithm again exhibits the highest AoI at all distances. This is likely because its reliance on a single criterion (e.g., path delay) might not effectively account for the substantial changes in signal quality over long distances. While it might choose paths with lower initial delay, the deteriorated signal at faraway stations can significantly increase transmission times and AoI compared to the other two algorithms.

VI. CONCLUSION AND FUTURE WORK

The ever-growing demand for real-time applications in areas like autonomous vehicles and remote surgery necessitates stricter guarantees on data freshness in mobile communication networks. In this paper, we addressed the challenge of minimizing Age of Information (AoI) — a metric crucial for real-time applications — in the context of Multipath TCP (MPTCP) traffic scheduling over dual connections (e.g., LTE and mmWave). Our work investigated the use of reinforcement learning to find an optimal policy for dynamically allocating traffic between the two paths in a dual connection scenario. The proposed approach considers the dynamic and stochastic nature of the environment, including:

- Fluctuating channel quality
- Varying traffic arrival patterns
- Changes in packet transmission rates
- Power level adjustments

To achieve this, we first formulated the problem as a stochastic optimization problem and then leveraged the power of reinforcement learning to solve it. This approach allows the scheduling policy to adapt to real-time changes in network conditions, unlike traditional static scheduling methods.

Our key innovation lies in incorporating AoI into the MPTCP scheduling framework for dual connections. This focus on AoI optimization is particularly relevant for real-time applications where timely data delivery is paramount. Additionally, our work utilizes MPTCP, a more advanced transport protocol compared to traditional TCP, making it suitable for future network environments. Furthermore, the non-dependence of our problem model on the specific technology used in the dual connection allows for broader applicability across different network scenarios.

As potential future research directions, we can explore extending this work to consider additional network complexities such as congestion control and security mechanisms. Additionally, investigating the effectiveness of alternative reinforcement learning algorithms tailored specifically for the dynamic characteristics of mobile networks could be an interesting avenue for further exploration.

As another potential future work, the findings from the real-world measurement study [20] provide valuable insights into the dynamic nature of cellular network deployments, particularly regarding the performance variations across

different bands (low, mid, high/mmWave) and deployment modes. These insights will be leveraged to inform the development and evaluation of our MPTCP scheduling approach with AoI minimization, ensuring its effectiveness in adapting to real-world network conditions for minimal data transmission delays.

REFERENCES

- [1] M. Agiwal, H. Kwon, S. Park, and H. Jin, "A survey on 4G-5G dual connectivity: Road to 5G implementation," *IEEE Access*, vol. 9, pp. 16193–16210, 2021.
- [2] P. Dong, R. Shen, Y. Li, C. Nie, J. Xie, K. Gao, and L. Zhang, "An energy-saving scheduling algorithm for multipath TCP in wireless networks," *Electronics*, vol. 11, no. 3, p. 490, Feb. 2022.
- [3] V. Hakami and M. Dehghan, "Distributed power control for delay optimization in energy harvesting cooperative relay networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 4742–4755, Jun. 2017.
- [4] B. He, J. Wang, Q. Qi, H. Sun, J. Liao, L. Lu, and Z. Han, "Learning-based real-time transmission control for multi-path TCP networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 9, no. 5, pp. 1353–1369, Nov. 2023.
- [5] I. Mahmud, T. Lubna, and Y.-Z. Cho, "Performance evaluation of MPTCP on simultaneous use of 5G and 4G networks," *Sensors*, vol. 22, no. 19, p. 7509, Oct. 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:202538867>
- [6] J. Jung, S. Lee, J. Shin, and Y. Kim, "Self-attention-based uplink radio resource prediction in 5G dual connectivity," *IEEE Internet Things J.*, vol. 10, no. 22, pp. 19925–19936, Sep. 2023.
- [7] S. Jung, H. Kim, X. Zhang, and S. Dey, "GaMiCO: Game-slicing based multi-interface computation offloading in 5G vehicular networks," *J. Commun. Netw.*, vol. 25, no. 4, pp. 491–506, Aug. 2023.
- [8] B. Y. L. Kimura, D. C. S. F. Lima, and A. A. F. Loureiro, "Packet scheduling in multipath TCP: Fundamentals, lessons, and opportunities," *IEEE Syst. J.*, vol. 15, no. 1, pp. 1445–1457, Mar. 2021.
- [9] F. Kooshki, M. A. Rahman, M. M. Mowla, A. G. Armada, and A. Flizikowski, "Efficient radio resource management for future 6G mobile networks: A cell-less approach," *IEEE Netw. Lett.*, vol. 5, no. 2, pp. 95–99, Sep. 2023.
- [10] C. Lee, S. Song, H. Cho, G. Lim, and J.-M. Chung, "Optimal multipath TCP offloading over 5G NR and LTE networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 293–296, Feb. 2019.
- [11] W. Li, H. Zhang, S. Gao, C. Xue, X. Wang, and S. Lu, "SmartCC: A reinforcement learning approach for multipath TCP congestion control in heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2621–2633, Nov. 2019.
- [12] J. Luo, X. Su, and B. Liu, "A reinforcement learning approach for multipath TCP data scheduling," in *Proc. IEEE 9th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2019, pp. 276–280.
- [13] S. Manap, K. Dimiyati, M. N. Hindia, M. S. A. Talip, and R. Tafazolli, "Survey of radio resource management in 5G heterogeneous networks," *IEEE Access*, vol. 8, pp. 131202–131223, 2020.
- [14] A. Narayanan, M. I. Rochman, A. Hassan, B. S. Firmansyah, V. Sathya, M. Ghosh, F. Qian, and Z.-L. Zhang, "A comparative measurement study of commercial 5G mmWave deployments," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2022, pp. 800–809.
- [15] F. Neto, J. Granjal, and V. Pereira, "A survey on security approaches on PPDR systems toward 5G and beyond," *IEEE Access*, vol. 10, pp. 117118–117140, 2022.
- [16] O. Onireti, A. Imran, and M. A. Imran, "Coverage, capacity, and energy efficiency analysis in the uplink of mmWave cellular networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 3982–3997, May 2018.
- [17] M.-S. Pan, T.-M. Lin, C.-Y. Chiu, and C.-Y. Wang, "Downlink traffic scheduling for LTE—A small cell networks with dual connectivity enhancement," *IEEE Commun. Lett.*, vol. 20, no. 4, pp. 796–799, Apr. 2016.
- [18] M. I. Rochman, D. Fernandez, N. Nunez, V. Sathya, A. S. Ibrahim, M. Ghosh, and W. Payne, "Impact of device thermal performance on 5G mmWave communication systems," in *Proc. IEEE Int. Workshop Tech. Committee Commun. Quality Rel. (CQR)*, Sep. 2022, pp. 1–6.
- [19] M. I. Rochman, V. Sathya, and M. Ghosh, "Outdoor-to-indoor performance analysis of a commercial deployment of 5G mmWave," in *Proc. IEEE Future Netw. World Forum (FNWF)*, Oct. 2022, pp. 519–525.

- [20] M. I. Rochman, V. Sathya, N. Nunez, D. Fernandez, M. Ghosh, A. S. Ibrahim, and W. Payne, "A comparison study of cellular deployments in Chicago and Miami using apps on smartphones," in *Proc. 15th ACM Workshop Wireless Netw. Testbeds, Experim. Eval. CHaracterization*, Jan. 2022, pp. 61–68.
- [21] T. Sylla, L. Mendiboure, S. Maaloul, H. Aniss, M. A. Chalouf, and S. Delbruel, "Multi-connectivity for 5G networks and beyond: A survey," *Sensors*, vol. 22, no. 19, p. 7591, Oct. 2022. [Online]. Available: <https://www.mdpi.com/1424-8220/22/19/7591>
- [22] C.-X. Wang, J. Bian, J. Sun, W. Zhang, and M. Zhang, "A survey of 5G channel measurements and models," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3142–3168, 4th Quart., 2018.
- [23] Y. Wang, C. Sun, F. Jiang, and J. Jiang, "Blocking- and delay-aware flow control using Markov decision process," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Aug. 2020, pp. 905–910.
- [24] Y. Xing, K. Xue, Y. Zhang, J. Han, J. Li, and D. S. L. Wei, "An online learning assisted packet scheduler for MPTCP in mobile networks," *IEEE/ACM Trans. Netw.*, vol. 31, no. 5, pp. 2297–2312, Feb. 2023.
- [25] H. Zhang, W. Li, S. Gao, X. Wang, and B. Ye, "ReLeS: A neural adaptive multipath scheduler based on deep reinforcement learning," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 1648–1656.
- [26] J. Zhao, J. Liu, H. Wang, C. Xu, and H. Zhang, "Multipath congestion control: Measurement, analysis, and optimization from the energy perspective," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 6, pp. 3295–3307, May 2023.
- [27] S. Zinno, A. Affinito, N. Pasquino, G. Ventre, and A. Botta, "Prediction of RTT through radio-layer parameters in 4G/5G dual-connectivity mobile networks," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2023, pp. 213–218.



ABABAKR IBRAHIM RASUL received the B.S. degree in computer science from the University of Salahadin, Soran, in 2008, and the M.Sc. degree in networking professional from the University of Sheffield Hallam, U.K., in 2013. He is currently pursuing the Ph.D. degree with Soran University, Iraq. He is also a Lecturer with Soran University. His research interests include network designs, network performance evaluation, wireless communication systems, and network security.



HAKEM BEITOLLAHI received the B.S. degree in computer engineering from the University of Tehran, in 2002, the M.S. degree from the Sharif University of Technology, Tehran, Iran, in 2005, and the Ph.D. degree from Katholieke Universiteit Leuven, Belgium, in 2012. He is currently an Assistant Professor with the School of Computer Engineering, Iran University of Science and Technology. His research interests include network security, real-time systems, and hardware accelerators.

• • •