Adaptive Cooperative Transmission Design for Ultra-Reliable Low-Latency Communications via Deep Reinforcement Learning

Hyemin Yu
University of Victoria
Victoria, BC V8P 5C2, Canada
hmyu@uvic.ca

Hong-Chuan Yang
University of Victoria
Victoria, BC V8P 5C2, Canada
hy@uvic.ca

Abstract

Next-generation wireless communication systems must support ultra-reliable low-latency communication (URLLC) service for mission-critical applications. Meeting stringent URLLC requirements is challenging, especially for two-hop cooperative communication. In this paper, we develop an adaptive transmission design for a two-hop relaying communication system. Each hop transmission adaptively configures its transmission parameters separately, including numerology, mini-slot size, and modulation and coding scheme, for reliable packet transmission within a strict latency constraint. We formulate the hop-specific transceiver configuration as a Markov decision process (MDP) and propose a dual-agent reinforcement learning-based cooperative latency-aware transmission (DRL-CoLA) algorithm to learn latency-aware transmission policies in a distributed manner. Simulation results verify that the proposed algorithm achieves the near-optimal reliability while satisfying strict latency requirements.

1 Introduction

Next-generation wireless communication systems are expected to support a wide range of mission-critical applications, such as remote surgery, autonomous vehicles, and real-time virtual/augmented reality [7]. These use cases demand ultra-reliable and low-latency communication (URLLC) with packet error rates as low as 10^{-5} or even 10^{-7} and an end-to-end latency on the order of milliseconds [14]. However, it is challenging to satisfy such stringent URLLC requirements over wireless channels due to unpredictable channel fading and limited radio resources.

Cooperative communication has emerged as a promising solution to enhance transmission reliability in wireless networks by introducing an intermediate relay node between source and destination [6]. However, most existing works on two-hop transmission under latency requirements focus on one-shot transmission with no retransmissions [12], [5], [10]. Accordingly, any decoding error on either hop leads to transmission failure. Moreover, these one-shot schemes have assumed perfect knowledge of the channel state information (CSI) on both hops to optimally allocate channel uses across two communication links. Acquiring such global CSI requires excessive overhead and thus cannot comply with the tight latency budget of URLLC. The automatic repeat request (ARQ) protocols can enhance reliability via per-hop retransmissions without requiring global CSI [3]. Although effective, ARQ-based retransmission mechanisms inevitably increase transmission delay [4], [19]. Therefore, achieving URLLC in relay-aided transmission requires a novel design that jointly optimizes both reliability and latency rather than sacrificing one for the other.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI and ML for Next-Generation Wireless Communications and Networking (AI4NextG).

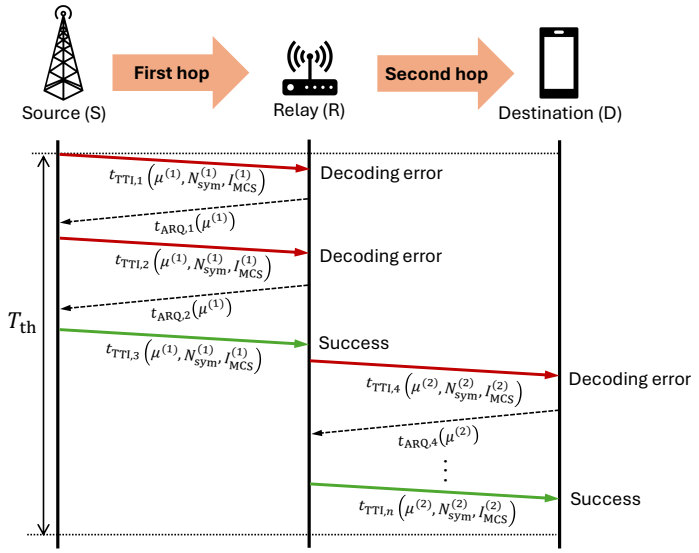


Figure 1: System model for two-hop relaying with ARQ protocols under the latency constraint T_{th}.

The Third Generation Partnership Project (3GPP) has introduced 5G new radio (NR) with key features, including adaptive modulation and coding (AMC), scalable numerology, and mini-slot scheduling to support URLLC services [1]. Previous works have optimized these features separately, focusing solely on AMC [8] or on scalable numerology [16], which limits the full potential of 5G NR. Very recently, Saatchi *et al.* demonstrated significant improvements in reliability by jointly optimizing numerology, mini-slot size, and modulation and coding scheme (MCS) under stringent latency constraints in a point-to-point single-carrier transmission system [13], which was extended to multicarrier transmission [18]. To the best of our knowledge, however, there has been no existing work that considers the impact of ARQ-based retransmission on the reliability under the latency constraint for two-hop relaying transmission.

Motivated by this, in this paper, we fill this gap by optimally configuring transmission parameters in every (re)transmission attempt on two hops to maximize the probability of successful packet delivery within a latency budget. Given the stringent latency constraint of URLLC, the resource configuration at the transceiver is performed only based on local CSI. To enable distributed operation without global CSI exchange/estimation, we propose a dual-agent reinforcement learning-based cooperative latency-aware transmission (DRL-CoLA) algorithm, where the source and the relay act as agents to learn latency-aware transmission policies from local observations and ARQ feedback. By doing so, the DRL-CoLA algorithm enables decentralized hop-specific execution while aligning both agents with the end-to-end latency constraint, thereby achieving URLLC over two-hop relaying transmission.

2 Two-Hop Transmission Under Latency Constraint

As depicted in Fig. 1, we consider a two-hop cooperative communication over 5G NR, where a source (S) transmits a delay-sensitive packet to a destination (D) via a half-duplex relay (R). To improve reliability, R helps forward a packet from S to D in a decode-and-forward manner, where only a successfully decoded packet at R is forwarded to D. We assume that the direct transmission from S to D is unavailable due to severe path loss, deep fading, and/or obstacles. Thus, the transmission from S to D via R will perform over two sequential hops, each of which supports scalable numerology, variable mini-slot, and AMC.

As specified in [2], 5G NR supports scalable numerology μ to adjust the subcarrier spacing by $2^{\mu} \times 15$ kHz, where $\mu \in \{0, 1, 2, 3, 4\}$. The mini-slot transmission in 5G NR further reduces latency by allowing $N_{\mathrm{sym}} \in \{2, 4, 7, 14\}$ orthogonal frequency division multiplexing (OFDM) symbols per mini-slot. The selection of numerology μ and the mini-slot size N_{sym} affects the subframe length, which is given in milliseconds (ms) by $T_{\mathrm{sf}} = N_{\mathrm{sym}}/(14 \times 2^{\mu})$ [13]. To improve link reliability, S

Table 1: MCS index table for URLLC service [1]

I_{MCS}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$R_C \times 1024$	30	50	78	120	193	308	449	602	378	490	616	466	567	666	772
Modulation	QPSK ($M = 2$)								16QAM ($M = 4$)			64QAM ($M = 6$)			

and R learn to adapt the MCS to the measured signal-to-noise ratio (SNR) and the remaining latency budget. We index the MCS for URLLC use cases specified in [1] by I_{MCS}, which is detailed in Table 1. The data packet of H bits must be successfully delivered to D within a latency budget T_{th}. To this end, S and R select numerology μ, mini-slot size N_{sym}, and MCS I_{MCS} for each (re)transmission attempt. Each hop transmission is performed over N_{sc} subcarriers, which is given by $N_{\mathrm{sc}} = \left\lfloor W/(2^{\mu} \times 15 \times 10^3) \right\rfloor$, where W is the available bandwidth, and $\lfloor . \rfloor$ is the floor function.

As the latency requirements for URLLC applications are typically shorter than the channel coherence time [12], we model the wireless channels between any two nodes as quasi-static flat fading, where all subcarriers experience the same fading within the latency constraint T_{th}. Let h_1 and h_2 denote the channel coefficients of S-R and R-D links, respectively, which are assumed to follow Rayleigh fading with unit mean $\mathbb{E}\left[|h_1|^2\right] = \mathbb{E}\left[|h_2|^2\right] = 1$. Accordingly, the instantaneous SNRs on the S–R and R–D links γ_1 and γ_2 are independent and exponentially distributed with means $\bar{\gamma}_1 \triangleq P_1 d_1^{-\eta}/\sigma^2$ and $\bar{\gamma}_2 \triangleq P_2 d_2^{-\eta}/\sigma^2$, where P_1 and P_2 are the transmit power at S and R, respectively, d_1 and d_2 are the respective link distances, η is the path loss exponent, and σ^2 is the power of the additive white Gaussian noise (AWGN).

In the first hop, S transmits its packet to R over $N_{\mathrm{sc}}^{(1)}$ subcarriers with selected numerology $\mu^{(1)}$, mini-slot size $N_{\mathrm{sym}}^{(1)}$, and MCS $I_{\mathrm{MCS}}^{(1)}$. While the number of subcarriers $N_{\mathrm{sc}}^{(1)}$ varies depending on $\mu^{(1)}$, the same mini-slot size and MCS are applied across all subcarriers. Let $R_C^{(1)}$ and $M^{(1)}$ denote the coding rate and the modulation order corresponding to $I_{\mathrm{MCS}}^{(1)}$, respectively. The number of symbols required to send H bits under the selected MCS $I_{\mathrm{MCS}}^{(1)}$ is given by $m_1 = \left\lceil H/\left(R_C^{(1)} \times M^{(1)}\right) \right\rceil$, where $\lceil . \rceil$ is the ceiling function. The number of subframes required to carry m_1 symbols is given by [18]

$$N_{\mathrm{sf}}^{(1)} = \left\lceil \frac{m_1}{N_{\mathrm{sc}}^{(1)} \times N_{\mathrm{sym}}^{(1)}} \right\rceil. \tag{1}$$

The transmission time interval (TTI) for the first-hop transmission attempt is given by

$$t_{\mathrm{TTI}}(\mu^{(1)}, N_{\mathrm{sym}}^{(1)}, I_{\mathrm{MCS}}^{(1)}) = N_{\mathrm{sf}}^{(1)} \times T_{\mathrm{sf}}^{(1)}. \tag{2}$$

For URLLC applications, the size of data packets is short and finite to comply with the stringent latency constraint [12]. Thus, the assumption of infinite blocklength underlying Shannon's capacity theorem is no longer valid [6]. Under such a finite blocklength regime, the decoding error probability becomes non-negligible and must be considered in performance analysis and system design. Applying the finite blocklength regime result, the decoding error probability at R for the first hop can be approximated as [11]

$$\varepsilon_1(\gamma_1, m_1) = Q\left(\ln 2 \sqrt{\frac{m_1}{V_1}} \left(\log_2(1+\gamma_1) - \frac{H}{m_1}\right)\right), \tag{3}$$

where $V_1 = 1 - (1+\gamma_1)^{-2}$ is the channel dispersion for S–R link, and $Q(x) = (1/\sqrt{2\pi}) \int_x^{\infty} \exp\left(-t^2/2\right) dt$ is the Gaussian Q-function. If a decoding error occurs at R, an ARQ request is sent over one OFDM symbol to S [13], [18]. Then, S selects the numerology, mini-slot size, and MCS to retransmit the packet. We assume that the ARQ request is always received successfully, and the overhead required to send an ARQ request is determined by the selected numerology [13].[1]

In the second hop, upon successful packet reception over first hop, R selects its own numerology $\mu^{(2)}$, mini-slot size $N_{\mathrm{sym}}^{(2)}$, and MCS $I_{\mathrm{MCS}}^{(2)}$ to transmit the decoded packet to D over $N_{\mathrm{sc}}^{(2)}$ subcarriers.

[1] Upon receiving an ARQ request, S reselects the numerology, mini-slot size, and MCS for retransmission. The number of OFDM symbols used in each (re)transmission attempt varies according to the newly selected MCS.

The number of symbols required to transmit H bits over the second hop can be obtained by $m_2 = \left\lceil H/\left(R_C^{(2)} \times M^{(2)}\right)\right\rceil$, where $R_C^{(2)}$ and $M^{(2)}$ are the coding rate and the modulation order associated with $I_{\mathrm{MCS}}^{(2)}$. Similar to (2), the TTI for the second hop transmission can be calculated by

$$t_{\mathrm{TTI}}(\mu^{(2)}, N_{\mathrm{sym}}^{(2)}, I_{\mathrm{MCS}}^{(2)}) = N_{\mathrm{sf}}^{(2)} \times T_{\mathrm{sf}}^{(2)}, \tag{4}$$

where $N_{\mathrm{sf}}^{(2)}$ is the number of subframes required for second hop transmission and can be obtained by (1). The decoding error probability at D in the second hop is given by

$$\varepsilon_2(\gamma_2, m_2) = Q\left(\ln 2\sqrt{\frac{m_2}{V_2}}\left(\log_2(1+\gamma_2) - \frac{H}{m_2}\right)\right), \tag{5}$$

where $V_2 = 1 - (1+\gamma_2)^{-2}$ is the channel dispersion for R–D link. If a decoding error is detected at D, the ARQ request is immediately sent to R. The (re)transmission attempts continue until the packet is successfully delivered to D or the latency budget is exhausted, which declares *packet loss*.

The total end-to-end transmission time consists of the TTIs for (re)transmissions and ARQ overheads. Let $i = 1$ and $i = 2$ denote agents S and R, respectively, corresponding to the hop in which they participate for packet transmission. The total transmission time $\mathcal{T}$ can be formulated by

$$\mathcal{T} = \sum_{i\in\{1,2\}} \sum_{n} \left(t_{\mathrm{TTI},n}(\mu^{(i)}, N_{\mathrm{sym}}^{(i)}, I_{\mathrm{MCS}}^{(i)}) + t_{\mathrm{ARQ},n}(\mu^{(i)})\right), \tag{6}$$

where $t_{\mathrm{TTI},n}(\mu^{(i)}, N_{\mathrm{sym}}^{(i)}, I_{\mathrm{MCS}}^{(i)})$ and $t_{\mathrm{ARQ},n}(\mu^{(i)})$ are the TTI and the ARQ response time in n-th (re)transmission attempt of agent $i \in \{1, 2\}$, respectively. Our goal is to maximize the probability of successful packet delivery over two hops by jointly optimizing numerology, mini-slot size, and MCS selection at S and R for each (re)transmission attempt under the latency constraint $\mathcal{T} \leq T_{\mathrm{th}}$.

3 Dual-Agent Distributed MDP Formulation

The uncertainties of wireless channels introduce random decoding errors. As a result, the actual time spent in each hop transmission is stochastic, which makes the total transmission time $\mathcal{T}$ a random variable. As the distribution of $\mathcal{T}$ is challenging to formulate, the latency constraint $\mathcal{T} \leq T_{\mathrm{th}}$ cannot be satisfied by traditional optimization techniques.[2] To address this issue, we formulate the adaptive transmission design of the two-hop cooperative communication system as a Markov decision process (MDP) [15]. Acting as independent agents, S and R optimally select transmission parameters via a sequential decision-making process to maximize end-to-end reliability while satisfying the latency constraint $\mathcal{T} \leq T_{\mathrm{th}}$. Based on the above discussion, the adaptive transmission design problem is formulated as the following MDP.

State space $\mathcal{S}^{(i)}$: Each agent observes the instantaneous SNR of its own link γ_i to adapt transmission parameters accordingly. Note that the decision at S affects not only its transmission in the first hop but also the remaining latency budget available for R to perform (re)transmissions in the next hop. Thus, additional information on the channel quality of the next hop is required to make latency-aware decisions. As the instantaneous channel state of the next-hop transmission cannot be obtained due to the limitations of the feedback channel, we assume that the average SNR of the next hop $\bar{\gamma}_{i+1}$ is available. Both agents must know the packet size H and the remaining latency budget in the current transmission attempt, denoted by τ_n. The state for agent i in n-th (re)transmission attempt is given by $s_n^{(i)} = (\gamma_i, \bar{\gamma}_{i+1}, H, \tau_n) \in \mathcal{S}^{(i)}$ with a dimension of 4. For the terminal hop ($i = 2$), there is no next hop to deliver the packet; therefore, we set $\bar{\gamma}_{i+1} = \infty$. The decision process has two absorbing terminal states,

- **Success**: If the packet is successfully transmitted to the intended receiver (R for the first-hop and D for the second-hop) within the latency budget, the process transitions to the **Success** state.

[2] To apply supervised deep learning approaches, we must have pre-labeled data on what the best transmission parameters or actions should be for each system state in advance. However, such data is challenging to obtain for stochastic environments where decoding outcomes vary with random channel realizations. To overcome this limitation, we adopt reinforcement learning to enable S and R to learn optimal decisions on transmission parameters by directly interacting with the environment.

- **Failure**: The process terminates in the **Failure** state if the remaining latency budget is exhausted before successful transmission.

Action space $\mathcal{A}$: The action space $\mathcal{A}$ consists of the available resource configurations in 5G NR, including the numerology μ, mini-slot size N_{sym}, and MCS I_{MCS}. The action space is defined by

$$\mathcal{A} = \left\{(\mu, N_{\text{sym}}, I_{\text{MCS}}) : \mu \in \{0,1,2,3,4\}, N_{\text{sym}} \in \{2,4,7,14\}, I_{\text{MCS}} \in \{1,\ldots,15\}\right\}, \quad (7)$$

where each action tuple $(\mu, N_{\text{sym}}, I_{\text{MCS}})$ has a dimension of 3. In n-th (re)transmission attempt, the action selected by agent i is given by $a_n^{(i)} = (\mu, N_{\text{sym}}, I_{\text{MCS}}) \in \mathcal{A}$, which directly affects the TTI and decoding error probability.

Transition dynamics: The transition probability $\mathcal{P}(s_{n+1}|s_n, a_n)$ is governed by the decoding error probability and the variation of latency budget, both of which determine whether a packet is successfully transmitted or further retransmissions are required. Given the selected action $a_n^{(i)}$, the remaining latency budget is updated by

$$\tau_{n+1} = \tau_n - t_{\text{TTI},n}(\mu, N_{\text{sym}}, I_{\text{MCS}}) - t_{\text{ARQ},n}(\mu). \quad (8)$$

For agent i, the probability of transitioning to the next state $s_{n+1}^{(i)}$ upon taking action $a_n^{(i)}$ in the current state $s_n^{(i)}$ depends on the occurrence of a decoding error and remaining latency budget τ_{n+1}. Thus, the state transition is defined by

$$\mathcal{P}(s_{n+1}^{(i)}|s_n^{(i)}, a_n^{(i)}) = \begin{cases} \mathbf{1}\{\tau_{n+1} \geq 0\} \times (1-\varepsilon_i), & \text{if } s_{n+1}^{(i)} = \textbf{Success}, \\ \mathbf{1}\{\tau_{n+1} < 0\} \times \varepsilon_i, & \text{if } s_{n+1}^{(i)} = \textbf{Failure}, \\ \mathbf{1}\{\tau_{n+1} \geq 0\} \times \varepsilon_i, & \text{otherwise}, \end{cases} \quad (9)$$

where $\mathbf{1}\{\cdot\}$ is the indicator function, and ε_i is the decoding error probability of agent i. If the transmission is successful within the latency budget, the process transits to the **Success** state with probability of $(1-\varepsilon_i)$. If the remaining latency is exhausted, the process moves to the **Failure** state. Otherwise, (re)transmission continues as long as there is sufficient latency.

Reward $\mathcal{R}^{(i)}$: Since S is responsible for packet transmission over two hops, its action must be evaluated depending on whether there is sufficient latency budget left for the next hop transmission. However, S cannot directly observe the transmission outcome of the second hop because R makes independent transmission decisions. To address this issue, we use the delay outage rate (DOR) [17] as a metric to estimate the likelihood of successful packet delivery in the next hop, given the remaining latency budget τ_{n+1}. According to [17], the delivery time required for agent i to send a packet of size H bits can be defined by

$$T_{\text{delivery}}^{(i)} = \frac{H}{W \log_2(1+\gamma_i)}. \quad (10)$$

As a delay outage occurs when the packet delivery time exceeds the latency budget τ_{n+1}, the DOR over the remaining latency τ_{n+1} is expressed as

$$\mathcal{P}\left(T_{\text{delivery}}^{(i)} > \tau_{n+1}\right) = 1 - \exp\left(\frac{-1}{\bar{\gamma}_i}\left(2^{\frac{H}{W\tau_{n+1}}} - 1\right)\right) \triangleq \mathcal{P}_{\text{DOR}}(\bar{\gamma}_i, \tau_{n+1}). \quad (11)$$

Based on the DOR, we define the reward for agent i as

$$\mathcal{R}_{n+1}^{(i)} = \begin{cases} 1 - \mathcal{P}_{\text{DOR}}(\bar{\gamma}_{i+1}, \tau_{n+1}), & \text{if } s_{n+1}^{(i)} = \textbf{Success}, \\ -1, & \text{if } s_{n+1}^{(i)} = \textbf{Failure}, \\ -0.1, & \text{otherwise}, \end{cases} \quad (12)$$

which gives less reward for actions that leave an insufficient latency budget τ_{n+1} for the next-hop transmission. For a successful terminal-hop transmission, the agent $i = 2$ receives a reward of 1, since the DOR for its next-hop transmission is zero with $\bar{\gamma}_{i+1} = \infty$. We impose a strong negative reward of -1 for entering the **Failure** state to discourage packet loss. A small negative reward of -0.1 is applied to suboptimal decisions that do not lead to the **Success** state but waste the remaining latency budget.[3]

[3] Both agents are penalized for actions that fail to reach the **Success** state yet consume the latency budget. Such a design encourages the agents to minimize unnecessary retransmissions. Thus, the number of retransmission attempts is implicitly optimized as the policy converges.

4 Dual-Agent Reinforcement Learning Solution

In two-hop cooperative transmission, two agents, S and R, make independent decisions. To facilitate the distributed learning, we propose a dual-agent reinforcement learning-based cooperative latency-aware transmission (DRL-CoLA) algorithm, where S and R learn the hop-specific transmission policies $\pi_i : s \mapsto a$, $i \in \{1, 2\}$, using only local observations and ARQ feedback. Since the formulated MDP involves a continuous state space and a discrete action space, we employ a deep Q-network (DQN) algorithm to approximate the Q-value function via deep neural networks (DNNs), i.e., $Q_i(s, a) \approx Q_i(s, a; \boldsymbol{\theta}_i)$ [9], where $\boldsymbol{\theta}_i$ is a parameter of the main Q-network for agent $i \in \{1, 2\}$. In each (re)transmission attempt, the agent observes its current state $s_n^{(i)}$ and selects the action $a_n^{(i)}$ following the ϵ-greedy policy. In turn, the environment will respond by providing the reward $\mathcal{R}_{n+1}^{(i)}$ and the next state $s_{n+1}^{(i)}$. Such experience is stored as a transition tuple $e_n = \left(s_n^{(i)}, a_n^{(i)}, \mathcal{R}_{n+1}^{(i)}, s_{n+1}^{(i)}\right)$ in a replay buffer $\mathcal{B}_i$. When training, a mini-batch $\mathcal{M}_i$ of stored experiences is sampled from the replay buffer $\mathcal{B}_i$ and used to calculate the mean squared error (MSE) loss function, given by

$$\mathcal{L}_i\left(\boldsymbol{\theta}_i\right) = \mathbb{E}_{e_n \sim \mathcal{M}_i}\left[\left(y_n^{(i)} - Q_i(s_n^{(i)}, a_n^{(i)}; \boldsymbol{\theta}_i)\right)^2\right], \tag{13}$$

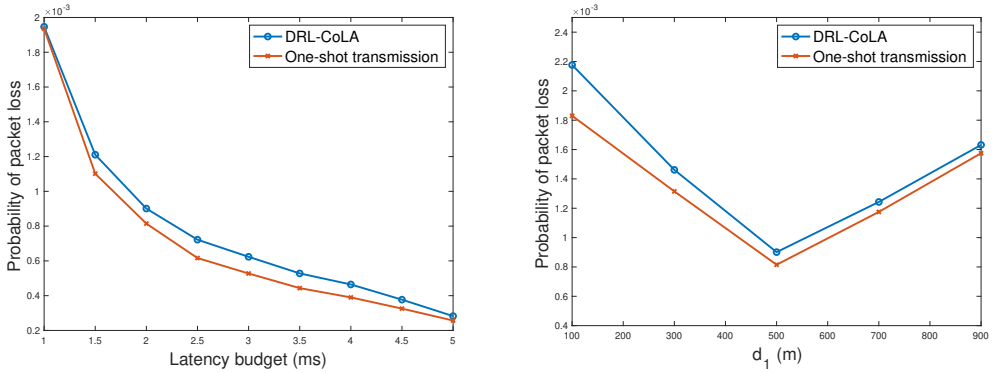
with the target value defined as $y_n^{(i)} = \mathcal{R}_{n+1}^{(i)} + \gamma \max_{a' \in \mathcal{A}} Q_i(s_{n+1}^{(i)}, a'; \boldsymbol{\theta}_i^-)$, where $\boldsymbol{\theta}_i^-$ is a parameter of the target Q-network, and γ is the discount factor with $0 \leq \gamma \leq 1$. The parameter of the main Q-network $\boldsymbol{\theta}_i$ is updated via the gradient descent method to minimize the loss function $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_i - \alpha \nabla_{\boldsymbol{\theta}_i} \mathcal{L}_i\left(\boldsymbol{\theta}_i\right)$, where α is the learning rate, and the gradient $\nabla_{\boldsymbol{\theta}_i} \mathcal{L}_i$ is calculated by

$$\nabla_{\boldsymbol{\theta}_i} \mathcal{L}_i(\boldsymbol{\theta}_i) = \mathbb{E}\Big[\big(y_n^{(i)} - Q_i(s_n^{(i)}, a_n^{(i)}; \boldsymbol{\theta}_i)\big) \times \nabla_{\boldsymbol{\theta}_i} Q_i(s_n^{(i)}, a_n^{(i)}; \boldsymbol{\theta}_i)\Big]. \tag{14}$$

Every E' episode, the parameter of the target Q-network $\boldsymbol{\theta}_i^-$ is copied from that of the main Q-network [9]. After training is completed, each agent obtains the trained Q-network $\boldsymbol{\theta}_i^*$. The optimal policy is derived by selecting the action that maximizes the optimal Q-value function in each state [15]. The process of training the Q-network is presented in **Algorithm 1**. The proposed DRL-CoLa algorithm for two-hop cooperative relaying is presented in **Algorithm 2**. These proposed algorithms are presented in Appendix A.

5 Simulation Results

In this section, we provide the simulation results to evaluate the effectiveness of the proposed DRL-CoLa algorithm. The simulation setting for experiments is presented in Appendix B.



(a) Probability of packet loss versus T_{th}.

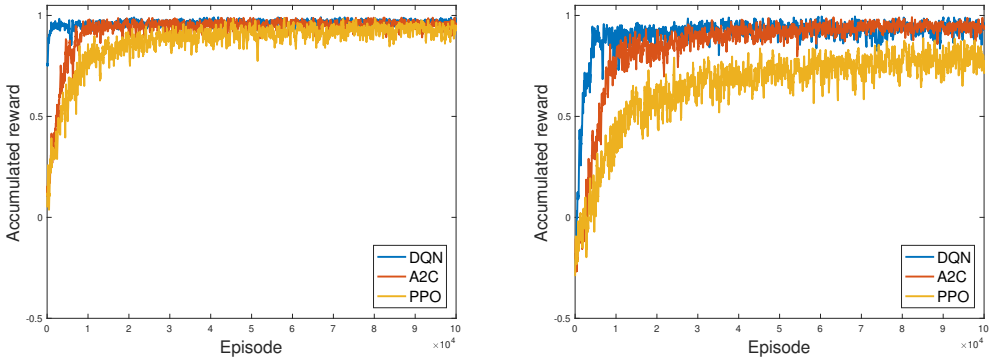
(b) Probability of packet loss versus d_1.

Figure 2: Comparison of end-to-end reliability between the proposed DRL-CoLa scheme and one-shot transmission.

In Figure 2 (a), we compare the proposed DRL-CoLA with the one-shot transmission scheme that optimally allocates symbols across both hops using global CSI. The one-shot transmission sets the

lower bound on packet loss probability. Across the tested latency regimes, DRL-CoLA achieves near-optimal packet loss performance even without global CSI by learning hop-specific policies from local CSI and ARQ feedback. This result indicates that decentralized, per-hop decision making can satisfy stringent URLLC requirements while avoiding the overhead of global CSI acquisition.

In Figure 2 (b), we plot the probability of packet loss versus the S–R distance d_1 while keeping $d_1 + d_2 = 1000$ m. The curve is V-shaped: probability of packet loss decreases as d_1 increases by bringing R closer to D, reaches a minimum at the symmetric placement $d_1 = d_2$, and then increases once d_1 exceeds 500 m. This is because moving R toward D improves the link quality of R–D but simultaneously degrades that of S–R. Given the symmetric simulation parameters and identical fading statistics on both hops, balancing the large-scale losses $d_1 = d_2$ leads to the lowest probability of packet loss. It is worth noting that packet loss is lower when $d_1 > d_2$ than when $d_1 < d_2$. The reason is that the second-hop transmission is performed on a tighter latency budget than the first-hop transmission. Thus, improving the channel quality for the R–D link increases the likelihood of completing packet delivery within such a tight latency budget, thereby reducing the packet loss.



(a) Rewards accumulated by S.

(b) Rewards accumulated by R.

Figure 3: Comparison of accumulated rewards achieved by DRL-CoLA when implemented with different reinforcement learning algorithms.

In Figure 3, we plot the accumulated rewards to demonstrate the convergence behavior of the proposed DRL-CoLA and to justify the selection of DQN for training both agents, S and R. We can see that the accumulated rewards of both agents increase steadily with training episodes and eventually stabilize. It confirms the convergence of the proposed DRL-CoLA algorithm. Moreover, DRL-CoLA trained with DQN shows faster convergence and achieves higher steady-state rewards compared to the A2C- and PPO-based implementation. This result indicates that the value-based DQN algorithm is more suitable for the considered cooperative transmission design, where a discrete action space allows DQN to learn near-optimal policies for both agents more efficiently than policy-gradient algorithms.

6 Conclusion

In this paper, we developed an adaptive transmission design for two-hop cooperative communication to meet the stringent URLLC requirements. We formulated the two-hop transmission process as an MDP to enable per-attempt radio resource configuration on the active hop for successful packet delivery across two hops within the latency budget. To solve the formulated MDP, we proposed the DRL-CoLA algorithm, where S and R learned decentralized, latency-aware policies from local observations and ARQ feedback. Simulation results showed that DRL-CoLA achieves near-optimal reliability comparable to the one-shot transmission scheme even without global CSI.

References

[1] 3GPP. NR; Physical Layer Procedures for Data (release 15). Technical Report TS 38.214, 3rd Generation Partnership Project (3GPP), 2020.

[2] 3GPP. NR; Physical Channels and Modulation (release 16). Technical Report TS 38.211, 3rd Generation Partnership Project (3GPP), 2021.

[3] F. E. Airod, H. Chafnaji, and H. Yanikomeroglu. Harq in full-duplex relay-assisted transmissions for urllc. *IEEE Open Journal of the Communications Society*, 2:409–422, 2021.

[4] J. Cheng and C. Shen. Relay-assisted uplink transmission design of urllc packets. *IEEE Internet of Things Journal*, 9(19):18839–18853, 2022.

[5] H. Di, X. Zhu, Z. Liu, and X. Tu. Joint blocklength and trajectory optimizations for urllc-enabled uav relay system. *IEEE Communications Letters*, 28(1):118–122, 2024.

[6] Y. Hu, M. C. Gursoy, and A. Schmeink. Relaying-enabled ultra-reliable low-latency communications in 5g. *IEEE Network*, 32(2):62–68, 2018.

[7] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim. Ultra-reliable and low-latency communications in 5g downlink: Physical layer aspects. *IEEE Wireless Communications*, 25(3):124–130, 2018.

[8] S. Mashhadi, N. Ghiasi, S. Farahmand, and S. M. Razavizadeh. Deep reinforcement learning based adaptive modulation with outdated csi. *IEEE Communications Letters*, 25(10):3291–3295, 2021.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[10] C. Pan, H. Ren, Y. Deng, M. Elkashlan, and A. Nallanathan. Joint blocklength and location optimization for urllc-enabled uav relay systems. *IEEE Communications Letters*, 23(3):498–501, 2019.

[11] Y. Polyanskiy, H. V. Poor, and S. Verdu. Channel coding rate in the finite blocklength regime. *IEEE Transactions on Information Theory*, 56(5):2307–2359, 2010.

[12] H. Ren, C. Pan, Y. Deng, M. Elkashlan, and A. Nallanathan. Joint power and blocklength optimization for urllc in a factory automation scenario. *IEEE Transactions on Wireless Communications*, 19(3):1786–1801, 2020.

[13] N. S. Saatchi, H.-C. Yang, and Y.-C. Liang. Novel adaptive transmission scheme for effective urllc support in 5g nr: A model-based reinforcement learning solution. *IEEE Wireless Communications Letters*, 12(1):109–113, 2023.

[14] C. She, C. Sun, Z. Gu, Y. Li, C. Yang, H. V. Poor, and B. Vucetic. A tutorial on ultrareliable and low-latency communications in 6g: Integrating domain knowledge into deep learning. *Proceedings of the IEEE*, 109(3):204–246, 2021.

[15] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2 edition, 2018.

[16] Z. Wang and V. W.S. Wong. Joint resource block allocation and beamforming with mixed-numerology for embb and urllc use cases. In *2021 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2021.

[17] H.-C. Yang, S. Choi, and M.-S. Alouini. Ultra-reliable low-latency transmission of small data over fading channels: A data-oriented analysis. *IEEE Communications Letters*, 24(3):515–519, 2020.

[18] H.-C. Yang and N. S. Saatchi. A deep reinforcement learning-based adaptive transmission design for trustworthy urllc support in 5g and beyond wireless systems. In *2024 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pages 1–5, 2024.

[19] C. Yin, R. Zhang, Y. Li, Y. Ruan, T. Tao, and D. Li. Power consumption minimization for packet re-management based c-noma in urllc: Cooperation in the second phase of relaying. *IEEE Transactions on Wireless Communications*, 22(3):2065–2079, 2023.

Appendix A
Proposed Algorithms

Algorithm 1 Training algorithm for Q-network

Input: Probability of exploration ϵ; replay buffer $\mathcal{B}_i$; mini-batch size $|\mathcal{M}_i|$; initial state $s_1^{(i)}$;main Q-network parameter $\boldsymbol{\theta}_i$; target Q-network parameter $\boldsymbol{\theta}_i^-$

Output: Remaining latency budget T_{rem}; terminal flag $done^{(i)}$, updated replay buffer $\mathcal{B}_i$; updated main Q-network parameter $\boldsymbol{\theta}_i$

Set $done^{(i)} \leftarrow$ False and $n \leftarrow 1$.
while not $done^{(i)}$ **do**
  Observe the current state $s_n^{(i)}$ and select the action $a_n^{(i)}$ using the ϵ-greedy policy.
  Receive the reward $\mathcal{R}_{n+1}^{(i)}$ and the next state $s_{n+1}^{(i)}$.
  Store transition $e_n^{(i)} = \left(s_n^{(i)}, a_n^{(i)}, \mathcal{R}_{n+1}^{(i)}, s_{n+1}^{(i)}\right)$ in the replay buffer $\mathcal{B}_i$.
  Randomly sample a mini-batch $\mathcal{M}_i$ of transition tuples from the replay buffer $\mathcal{B}_i$.
  Update the main Q-network by performing gradient descent on loss $\mathcal{L}_i\left(\boldsymbol{\theta}_i\right)$ in (13).
  if $s_{n+1}^{(i)}$ = **Success** or $s_{n+1}^{(i)}$ = **Fail**, **then**
    $done^{(i)} \leftarrow$ True.
  else
    $n \leftarrow n + 1$.
  end if
end while
Set the remaining latency by $T_{\text{rem}} \leftarrow \tau_{n+1}$.

Algorithm 2 Proposed DRL-CoLa algorithm

Input: Number of episodes $E_{\max}$; probability of exploration ϵ; epsilon decaying rate λ; frequency of updating target Q-network E'; mini-batch size $|\mathcal{M}_i|$; packet size H; latency budget T_{th}

Output: Trained Q-network parameters $\boldsymbol{\theta}_1^*$ and $\boldsymbol{\theta}_2^*$

Initialize the main Q-network $\boldsymbol{\theta}_i$ and the target Q-network with $\boldsymbol{\theta}_i^- \leftarrow \boldsymbol{\theta}_i$.
Initialize the relay buffer with $\mathcal{B}_i \leftarrow \emptyset$.
for $e = 1, \cdots, E_{\max}$ **do**
  // Perform first-hop transmission
  Initialize the state $s_1^{(1)}$ with packet size H and latency budget T_{th}.
  Invoke **Algorithm 1** to obtain the remaining latency T_{rem}, first-hop termination status $done^{(1)}$, updated replay buffer $\mathcal{B}_1$, and parameter $\boldsymbol{\theta}_1$.
  if $T_{\text{rem}} \geq 0$ and $done^{(1)} =$ True, **then**
    // Perform second-hop transmission
    Initialize the state $s_1^{(2)}$ with packet size H and latency budget T_{rem}.
    Invoke **Algorithm 1** to obtain the updated replay buffer $\mathcal{B}_2$ and parameter $\boldsymbol{\theta}_2$.
  end if
  Decay exploration rate $\epsilon \leftarrow \lambda\epsilon$.
  Every E' episode, update the parameters of the target Q-network by $\boldsymbol{\theta}_1^- \leftarrow \boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2^- \leftarrow \boldsymbol{\theta}_2$.
end for

Appendix B
Simulation Setting

In this appendix, we detail the simulation settings used to generate the results in Section 5. We set $P_1 = 30$ dBm, $P_2 = 30$ dBm, and $\eta = 2$. The bandwidth is set to $W = 480$ kHz [2] with the noise power spectral density $N_0 = 10^{-14}$ W/Hz (i.e., -110 dBm/Hz), which leads to the noise power as $\sigma^2 = N_0 \times W$. Unless otherwise specified, we set the latency budget as $T_{\mathrm{th}} = 2$ ms, the distance as $d_1 = d_2 = 500$ m, and $H = 256$ bits. The main and target Q-networks are implemented as DNNs with three fully connected layers, consisting of 64, 256, and 128 neurons, respectively, with *ReLU* activations. Additionally, the Adam optimizer is employed to train both Q-networks. The hyperparameters for training **Algorithm 2** are summarized in Table 2.

Table 2: Hyperparameters for **Algorithm 2**

Parameter	Description	Value
$E_{\max}$	Number of episodes	100000
E'	Frequency of updating target	2000
γ	Discount factor	0.95
α	Learning rate	10^{-5}
ϵ	Probability of exploration	1
λ	Epsilon decaying rate	0.999
$\mathcal{B}_i$	Size of replay buffer	10000
$\mathcal{M}_i$	Size of mini-batch	64