Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.39104

Deep Reinforcement Learning in Communication Networks

Jiayu Chen

Department of Engineering and Computing, University of Lancashire, Preston, UK
jchen65@lancashire.ac.uk

Abstract. In communication networks, during different traffic and channel conditions, scarce resources such as spectrum, power, bandwidth and computing resources need to be allocated. In this review, the authors examine how Deep Reinforcement Learning (DRL) can support such decisions, what its role is and why there is a need for it. The representative tasks and decisions made in the studies are discrete wireless allocation, continuous congestion control, hybrid edge offloading, graph-based routing, adaptive video streaming, and network slicing. Observations, actions, objectives, algorithm design, strengths and limitations are explored and contrasted between the tasks in each application. From the literature, one can see that Deep Q-Network agents can handle local wireless decision making; however, fine action choices and the multi-agent interactions lead to a higher coordination cost. Actor–critic controllers adjust sending rates according to feedback, but there are issues of fairness and transferability to unknown traffic. Although graph-based routing does not guarantee robustness to changes in demand, hybrid offloading reduces the search burden while retaining optimization overhead. DRL is most compelling when conditions change and adaptation is required, and when benefits can be shown based on appropriate networking baselines. Future research topics include data coverage, enforceable service constraints, as well as coordination among controllers of different time scales.

Keywords: Deep Reinforcement Learning, Communication Networks, Resource Allocation, Congestion Control

1. Introduction

Communication networks are wired or wireless systems that allow communication among devices that are used to provide Internet services. They must be able to divide bandwidth, transmission power, computing power and buffer space. However, congestion, interference and poor service quality can be caused by varying traffic volumes and demands for services. Although conventional optimization and fixed rules have their relevance, they may not be applicable if a good model is not available and if situations change quickly. Such a challenge encourages the use of a learning method based on interaction called reinforcement learning, and also Deep Reinforcement Learning (DRL), which uses neural networks to process the complex observations and decision spaces [1].

Modern reinforcement learning evolved from trial-and-error learning and optimal control, with temporal difference learning and Q-learning becoming important foundations in the 1980s [2]. A large breakthrough in DRL happened in 2015 when Mnih et al. demonstrated that deep Q-networks

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.39104

can learn control policies from high dimensional inputs [3]. Lillicrap et al. additionally used DRL for continuous control in simulated robotic manipulation and locomotion tasks [4]. This prompted studies on learning-based network management. For video bitrate selection, Mao et al. introduced Pensieve in the field of networking [5]. Li et al. investigated DRL for network slicing, showing how resource allocation could respond to differentiated service demands [6]. Nasir and Guo analysed multi-agent distributed power allocation, and Ye et al. proposed decentralised spectrum and power selection for vehicular communications [7, 8]. Jay et al. presented Aurora congestion control for the Internet and highlighted issues of fairness and generalisation [9]. Huang et al. later proposed learning-assisted computation offloading in wireless-powered edge networks [10]. The deployment issues have also been the subject of studies. Almasan et al. deployed DRL on top of GNNs for routing over unseen topologies [11] and Polese et al. proposed ColO-RAN to support evaluation of learned radio access network control [12]. These studies demonstrate across-the-board DRL potential, although their findings rely on varying observations, objectives, and assumptions. Focused reviews tying algorithm selection to networking requirements and deployment evidence are required.

This review covers DRL foundations, illustrative applications to networks, and real-world constraints. Section 2 discusses existing studies on the design of learning-based methods for some specific communication problems and their pros and cons. Section 3 presents future possibilities, such as hybrid optimization, offline learning, robust transfer, and network control time scale coordination. The key findings are summarized in Section 4.

2. Learning-based methods for communication tasks

Learning-based methods should be chosen according to the communication decision they support. The different choices of a channel, a sending rate and an edge server have different consequences. This section presents problem definitions as they relate to the applicable methods, along with their pros and cons. Some typical applications include wireless allocation, congestion control, offloading, routing, and service adaptation [5-12].

2.1. Discrete resource allocation and communication costs

Spectrum is finite and a higher power for a wireless transmitter could lower the rates of neighboring users. A Deep Q-Network (DQN) addresses discrete allocation by calculating the future return for every possible action. Experience replay replays past transitions and a target network offers a more stable learning target [3]. Observations might include Channel State Information (CSI), measured interference, and service indicators, with actions being based on pre-defined power levels. Reward should be based on the intended network utility, not just the transmitter's rate.

The DQN target for a non-terminal transition is

$$y_t = r_t + \gamma \max_{a'} Q_{\theta^-}(s_{t+1}, a') \tag{1}$$

Here, s_{t+1} is the next state, r_t is the immediate reward, γ is the discount factor, $\theta-$ denotes the target-network parameters, and a' is a candidate action in the next state. Maximization chooses the candidate whose predicted value is maximum [3]. In a terminal transition, the future value term is dropped.

Nasir and Guo use deep Q-learning for distributed power allocation with knowledge from adjacent links, considering the time delay in CSI [7]. It is useful in supporting local decisions with

DOI: 10.54254/2755-2721/2026.39104

regard to interference between transmitters. This decreases the need for repeated central optimisation but still creates signalling overhead to collect neighbour information. Considering latency and interference to V2I links, Ye et al. expand decentralized resource allocation to Vehicle-to-Vehicle (V2V) communication, allowing the agents to choose sub-bands and power levels [8].

These applications demonstrate the impact MADRL has on coupled wireless decisions [7, 8]. Finer power discretization enlarges the action space; however, the learning environment becomes non-stationary when the agents adapt simultaneously. For practical evaluation, deadline satisfaction and signalling cost should therefore be used in addition to throughput. A policy with a better average rate but that often misses vehicular deadlines would not address the original communication problem.

2.2. Continuous rate control and hybrid offloading

There are network decisions, like sending rates, that are inherently continuous. Breaking them down into numerous options can be inefficient when selecting actions. Actor–critic methods, by contrast, learn an actor that suggests actions and a critic that estimates the consequences. The continuous actions can be handled by Deep Deterministic Policy Gradient (DDPG) [4]. Clipping the objective function is a popular practice in proximal policy optimization (PPO) for preventing excessively large shifts in policy [13]. Soft Actor–Critic (SAC) combines experience reuse with an entropy-augmented objective that supports exploration [14]. These mechanisms offer helpful design options but are not, in and of themselves, performance benefits specific to any given network.

In Aurora, PPO is applied to Internet congestion control and adjusts the sending rate according to recent feedback [9]. It is formulated in relation to delay and packet loss that can happen at a later stage; it adapts to the varying conditions of the network rather than following some preset response rule. However, a controller that performs well in isolation may behave differently when it's running in conjunction with other protocols. Issues of fairness and unfamiliar bottleneck conditions are pertinent to its interpretation.

Mobile-Edge Computing (MEC) requires a different combination of decisions: whether to offload a task and how to split supporting resources. Huang et al.'s Deep Reinforcement Learning-Based Online Offloading (DROO) framework learns binary offloading decisions and incorporates them with resource optimization [10]. It is therefore not to be confused with a controller that directly learns all continuous allocations. It is a useful innovation as it reduces the burden of combinatorial search, while still keeping an optimization element. Extensions with task queues should also take waiting time into account, and the remaining computation must fit within the decision interval.

2.3. Routing with graph-based representations

Relationships between link, path and available capacity drive routing decisions. A learned policy based on a fixed-length feature vector may not be adequate if the network topology changes. A Graph Neural Network (GNN) provides a representation to process connectivity in the network, instead of treating each measurement as an independent input.

In optical networks, Almasan et al. combine GNNs with DRL and create a problem-specific action space for routing optimization [11]. They use structural representation and action design and test generalization to unseen topologies. This is relevant because learned routing would be more feasible if it avoided per-topology retraining.

But topology transfer and traffic transfer could be different requirements. Even if a policy deals with various graph types, it can still fail under different demand patterns or inaccurate link

DOI: 10.54254/2755-2721/2026.39104

measurements. Consideration should include blocking, or congestion, under different loads, and candidate paths must be feasible for deployment. These are not guarantees given by just using a GNN, but are engineering implications of the routing problem. Increased representation complexity must also be balanced against the time it takes to collect network information and make a routing decision.

2.4. Service-oriented adaptation

Service objectives must represent user experience goals or service pledges. Pensieve takes an Actor–critic approach to selecting video bitrates based on client observations [5]. Its Quality of Experience (QoE) target is a combination of visual quality, rebuffering and quality variation. A helpful property is that learning can support various QoE preferences, and altering those preferences can alter which policy looks best. Reporting separate parts is still required to make sense of an aggregate score.

Network slicing is a process in which resources are shared between services with varying QoS needs. Li et al.'s work explores DRL for priority-based core network management and radio resource slicing [6]. Adaptation can improve allocation over time as demand changes, but aggregate resource efficiency does not by itself establish service isolation. Even if the average delay in a slice is low, the delays may be unacceptable during demand spikes. ColO-RAN serves as an experimental platform to investigate programmable radio infrastructure for learned slicing and scheduling [12] and to link both algorithm design and practical control loops.

2.5. Comparative summary

The method best suited to a communication task depends on its action space, available observation space, and service objective. Although distributed Q-learning is able to learn wireless allocation locally, finer power selection expands the action space, and more wireless agents require more coordination [7, 8]. PPO-based congestion control tries to adapt sending rates to feedback, and fair sharing with other flows remains a concern [9]. DROO decreases the search space on binary offloading decisions and requires timely resource optimization [10]. Connectivity information can be used with graph-based routing; however, changes in demand and link measurements can restrict transfers [11]. While network slicing addresses differentiated service demand, Pensieve targets viewing quality directly; both need to be evaluated beyond an aggregate reward [5, 6].

Table 1. Strengths and limitations of learning-based methods across communication tasks

Communication task	Method or example	Main strength	Main limitation
Wireless allocation	Distributed Q-learning [7, 8]	Local interference-aware decisions	Coordination and signalling costs
Congestion control	Aurora [9]	Adaptive rate selection	Fairness and traffic transfer
Edge offloading	DROO [10]	Reduced discrete search burden	Remaining optimization overhead
Routing	GNN-based DRL [11]	Structural topology representation	Demand shifts and observation cost
Video streaming	Pensieve [5]	Direct QoE optimization	Sensitivity to reward preferences
Network slicing	DRL-based management [6, 12]	Demand-responsive allocation	Slice isolation during load spikes

DOI: 10.54254/2755-2721/2026.39104

These strengths and drawbacks are summarised in Table 1 without trying to directly compare different experiments. Learning-based approaches can also account for changes in traffic, channels or service preferences, and some do not require centralised optimisation at every update. Their range, however, depends on task requirements: wireless agents require signalling and coordination, and routing policies require reliable topology data and demand data. Comparisons should thus involve service outcomes such as deadline satisfaction, rebuffering, fairness violations and slice violations, as well as decision latency, signalling cost and retraining cost. This shows whether adaptation provides practical benefits over simpler controllers within the same limits.

3. Future prospects

The reviewed methods indicate that the appropriate learning method should be used to suit the network requirement. Key priorities are integration with existing optimisation, minimisation of unsafe data collection, enhancement of transfer and validation of the full control loop. These priorities address the limitations of the applications, rather than assuming that one large model can solve all network issues.

3.1. Hybrid optimization and structured decision-making

A hybrid approach can divide a communication problem between appropriate methods. DROO does this by incorporating learned offloading decisions and resource optimisation [10]. This could be expanded to scenarios where discrete placement decisions interact with continuous bandwidth or computing allocations.

For instance, a learning agent could select a suitable edge server, while an optimisation module assigns resources within the allowable capacity. This shrinks the learned action space and makes feasibility easier to examine. Yet, the combined time to make the decision, not just the time to make the neural-network inference, should be measured. It is also good to compare with a simple selection rule to establish the value added by learning.

3.2. Offline learning and efficient adaptation

Offline reinforcement learning learns from past transitions without additional interaction during training [15]. This can help networks as exploration can interrupt services, and can assist initial policy-making before controlled online testing.

The main problem is poor data coverage. The logs from an existing controller might be sparse, making the consequences of alternative actions difficult to estimate. A large data set is therefore not always informative. Future investigations should note the behaviour policy, action coverage and operating conditions in the records. Policies should be carefully evaluated when their proposed actions differ substantially from recorded behaviour.

Adaptation should also be selective. For traffic changes, only a service-specific component may need to be updated rather than retraining the entire controller. The aim is not to train faster, but to reduce data collection and adaptation costs without reducing performance.

3.3. Robust generalization and service constraints

The Almasan et al. routing results suggest additional study of representations that transfer across network structures [11]. A wider assessment would differentiate changes in topology, traffic and

DOI: 10.54254/2755-2721/2026.39104

measurement delay. Testing them separately and together would help determine what variations a learned policy can withstand.

Complementary treatments are needed for service constraints. Constrained Policy Optimization provides a framework for learning policies with expected-cost constraints [16]. However, satisfying expected-cost constraints does not guarantee that every packet will meet the deadline. Future design should include enforceable action limits, appropriate resource reservations and a fallback strategy for latency-sensitive apps.

It could be a vehicular controller which only transmits at certain power levels and employs a predetermined scheduling rule when observations are not reliable. When it comes to this proposed approach, deadline violations and recovery behaviour would have to be evaluated. It should not be considered safe simply because its average reward is better.

3.4. Coordination across network control time scales

Controllers in communication networks have varying time scales of operations. On the other hand, the transmission resources could be allocated very frequently via a radio scheduler, and the capacity change may be requested less frequently via a network-slicing controller [6, 12]. Transport congestion control and video adaptation are based upon the feedback from other locations along the communication path [5, 9]. If these controllers adapt independently, one may respond to a congestion notification after another has already changed the available capacity. Their objectives may be competing with each other as well: for example, a slice manager may make sure the capacity of a service is reduced, whereas the local scheduler will attempt to meet a surge of packet deadlines.

It seems like an important avenue is hierarchical coordination with explicit time scales and shared constraints. A slower controller could be used to set feasible resource budgets, and faster agents can make local decisions based on the budgets. This division reduces the joint decision space and makes capacity limits easier to enforce. It also poses new design questions, as aggregated demand might arrive too late to update the budget while the faster agent may be issued a command based on a stale view of the interference or traffic. Future research should state what information each controller gets, when it is updated, and how conflicts are resolved.

Evaluation needs to investigate network-level behaviours under peak load, mobility and unexpected disturbances including missed deadlines and control-traffic overhead. The ColO-RAN programmable radio platform can be used to test these coupled control loops beyond simulation [12] to determine if coordinated learning could lead to better service performance while remaining feasible at the operating speed of the network.

4. Conclusion

This review has related DRL methods to communication-network decisions and performance requirements. Discrete Q-learning supports local wireless resource allocation, although action discretisation and communication between agents increase coordination costs. But fairness with other flows is important even when using Actor–critic methods to adapt sending rates based on feedback. Hybrid edge offloading provides a smaller binary search space and retains optimisation overhead. Graph-based routing and service-oriented control show that topology, viewing quality and slice isolation influence objectives and evaluation. A universally best algorithm cannot be identified from these findings. Studies use different workloads, baselines and measures; hence, numerical gains cannot be compared or used as evidence of reliable deployment. Future work involves incorporating both learning and practical action sets, testing policies on unknown traffic and

DOI: 10.54254/2755-2721/2026.39104

topologies and coordinating controllers at multiple time scales. The real worth of DRL should be evaluated by sustained service performance, decision latency and reliability, rather than rewards alone.

References

[1] Luong, N. C., Hoang, D. T., Gong, S., Niyato, D., Wang, P., Liang, Y.-C., & Kim, D. I. (2019). Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys Tutorials, 21*(4), 3133–3174. doi:10.1109/COMST.2019.2916583.

[2] Sutton, R., & Barto, A. (2018). *Reinforcement learning*. MIT Press. Available at: https://mitpress.mit.edu/9780262039246/reinforcement-learning/ [Accessed 15 Sept. 2026].

[3] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529–533. doi:10.1038/nature14236.

[4] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arxiv.org*. Available at: https://arxiv.org/abs/1509.02971 [Accessed 16 Sept. 2026].

[5] Mao, H., Netravali, R., & Alizadeh, M. (2017). Neural adaptive video streaming with Pensieve. *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. doi:10.1145/3098822.3098843.

[6] Li, R., Zhao, Z., Sun, Q., I, C.-L., Yang, C., Chen, X., Zhao, M., & Zhang, H. (2018). Deep reinforcement learning for resource management in network slicing. *IEEE Access, 6*, 74429–74441. doi:10.1109/access.2018.2881964.

[7] Nasir, Y. S., & Guo, D. (2019). Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks. *IEEE Journal on Selected Areas in Communications, 37*(10), 2239–2250. doi:10.1109/jsac.2019.2933973.

[8] Ye, H., Li, G. Y., & Juang, B.-H. F. (2019). Deep reinforcement learning based resource allocation for V2V communications. *IEEE Transactions on Vehicular Technology, 68*(4), 3163–3173. doi:10.1109/tvt.2019.2897134.

[9] Jay, N., Rotman, N., Godfrey, B., Schapira, M., & Aviv Tamar (2019). A deep reinforcement learning perspective on internet congestion control. *PMLR, 97*, 3050–3059. Available at: https://proceedings.mlr.press/v97/jay19a.html [Accessed 22 Sept. 2026].

[10] Huang, L., Bi, S., & Zhang, Y. J. (2019). Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks. *IEEE Transactions on Mobile Computing*, 1–1. doi:10.1109/tmc.2019.2928811.

[11] Almasan, P., Suárez-Varela, J., Rusek, K., Barlet-Ros, P., & Cabellos-Aparicio, A. (2022). Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case. *Computer Communications*. doi:10.1016/j.comcom.2022.09.029.

[12] Polese, M., Bonati, L., D'Oro, S., Basagni, S., & Melodia, T. (2022). ColO-RAN: Developing machine learning-based xApps for open RAN closed-loop control on programmable experimental platforms. *IEEE Transactions on Mobile Computing*, 1–14. doi:10.1109/TMC.2022.3188013.

[13] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv.org*. https://arxiv.org/abs/1707.06347

[14] Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *Proceedings.mlr.press*. https://proceedings.mlr.press/v80/haarnoja18b.html

[15] Levine, S., Kumar, A., Tucker, G., & Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv:2005.01643 [cs, stat]. https://arxiv.org/abs/2005.01643

[16] Achiam, J., Held, D., Aviv Tamar, & Pieter Abbeel (2017). Constrained policy optimization. *PMLR*, pp. 22–31. https://proceedings.mlr.press/v70/achiam17a.html