Summarizing

Applications of Multi-Agent Deep Reinforcement Learning with Communication in Network Management: A Survey

This article is a survey about the application of MADRL with communication in network management. It proves that with the increasing complexity of network management, traditional centralized network management modes have gradually become unable to meet the demand. Traditional network management technologies usually rely on heuristic methods, which are hard to scale for automation and intelligent management in future networks. The article highlights the importance of deploying distributed and decentralized network engineering to improve network performance using Multi-Agent Deep Reinforcement Learning (MADRL) as a solution. It also emphasizes that in some partially observable distributed multi-agent systems, the mutual influence of adaptation strategies renders the system vulnerable to non-stationary problems—where each agent’s learning continuously affects the environment, making coordination and convergence challenging.

For the application of MADRL, the survey focuses on several network management domains. In traffic engineering, MADRL is used to optimize routing paths for data packets and balance the load among switches, routers, and links. The survey introduces key works in this field; for example, in [31] a MADRL routing method is proposed where each Autonomous System (AS) acts as an agent. The results show that a system with agent communication performs better than one where agents rely solely on their local observations to make decisions. Additionally, in transmit power control for wireless networks, network entities must control transmission power to reduce interference with other agents. Through communication, an agent can learn not only its own conditions but also the interference it causes or the power allocation of other agents, thereby finding a balance between improving transmission power and reducing interference. The article also touches on network security, presenting works where coordinated agent communication helps defend against threats like jamming and DDoS attacks.

After discussing these applications, the survey details the communication strategies that underpin MADRL in network management. The article categorizes the information exchanged among agents as "communication content," which refers to what information—such as an agent’s state, action, reward, or strategy—is encoded in the messages. While every agent internally uses both state and reward for learning, designers may choose to share only specific pieces of information externally to enhance coordination and reduce communication overhead. Furthermore, the survey defines "communication objects" as the targets of this information exchange. It categorizes them into three types: Neighbor Agents (agents share information only with those directly connected or in close proximity), All Agents (agents broadcast information to every other agent in the system), and Central Agent (agents communicate solely with a designated central controller that aggregates and redistributes decisions). Finally, it reviews "communication message processing" techniques—such as concatenation (directly combining agent messages), neural networks (learning effective communication patterns), and averaging (simplifying communication while potentially losing critical nuances)—which integrate the exchanged messages into each agent’s decision-making process. Together, these communication strategies enable MADRL systems to overcome the limitations of partial observability and non-stationarity, paving the way for more adaptive and intelligent network management solutions.

**Analysis of Independent Learning in Network Agents: A Packet Forwarding Use Case**

This article explores the use of Independent Q-Learning (IQL) for packet forwarding in Named Data Networking (NDN). The authors investigate whether independent reinforcement learning agents (where each agent learns without coordination with others) can efficiently make forwarding decisions in a distributed network. They propose an Independent Deep Q-Network Forwarding (IDQF) strategy and evaluate its performance against the basic Best Route (BR) strategy in NDN. The study highlights major challenges with independent learning in network environments, including non-stationarity, partial observability, and delayed rewards.

Problem Statement:  
 Traditional forwarding in large-scale networks often relies on centralized or cooperative approaches that are costly, complex, and prone to issues like single points of failure. The article addresses the challenge of deploying independent, autonomous agents for packet forwarding. However, such independent learning methods face significant difficulties due to non-stationarity (as agents adapt concurrently), partial observability (limited local information), and delayed rewards (latency in feedback), which collectively hinder convergence to optimal forwarding policies.

Proposed Solution and Methodology:

The article proposes an Independent Deep Q-Network Forwarding (IDQF) strategy for packet forwarding in Named Data Networking (NDN).In this approach, each router acts as an autonomous reinforcement learning agent that makes forwarding decisions based solely on its local observations.

State Representation:

Local Observations:  
 Each router gathers local network measurements such as:

Round-Trip Time (RTT)

Interest satisfaction ratio

Number of retransmitted packets

Time Window:  
 Observations are collected over a time window [t−δt,t] to average out fluctuations and provide a stable estimate of the network conditions.

Action Space:

The agent’s decision involves selecting the optimal next-hop interface for forwarding incoming Interest packets by evaluating the Q-value of each possible interface using a Deep Q-Network (DQN).

Reward Function:

The reward for a given time step t is defined as:

RW(t)=−(M1 k=t−δt∑t RTT(k)+C×R(t)) where:

* M is the number of data packets received in the interval [t−δt,t],
* RTT(k) is the round-trip time of a packet at time k,
* R(t) is the number of retransmissions during this time window,
* C is a constant penalty multiplier.

The DQN is trained using experience replay with parameters such as a decay rate of 0.001 and a learning rate of 1.0 over 50 episodes (each 60 seconds long). The experiments span various traffic loads by testing five different Interest packet request rates.

Environment Tested:  
 The experiments were conducted using ndnSIM in the NDN context. The network topology used is the Sprint topology, consisting of 11 routers and 18 links with varied capacities (1 Mbps, 2 Mbps, and 5 Mbps) and link delays. The testing scenario involves a single consumer requesting data from a single producer, with Interest packet rates varying from 100 to 300 packets per second, representing different levels of network load.

Results:

The IDQF strategy consistently underperforms compared to the basic Best Route (BR) approach, showing 10%-36% lower throughput and higher average end-to-end delays across all tested traffic rates.

Advantages:

Scalability: Unlike centralized RL, IDQF does not require coordination among agents, making it scalable for large networks.

Realistic Testing in NDN:

Identification of Key Challenges in Independent Learning: The article systematically outlines the limitations of IQL in networking, which helps guide future research.

Disadvantages:

Non-Stationarity Issues

Delayed Rewards Reduce Learning Efficiency: Since network conditions change dynamically, the delayed impact of forwarding decisions makes RL unstable.

Partial observations.