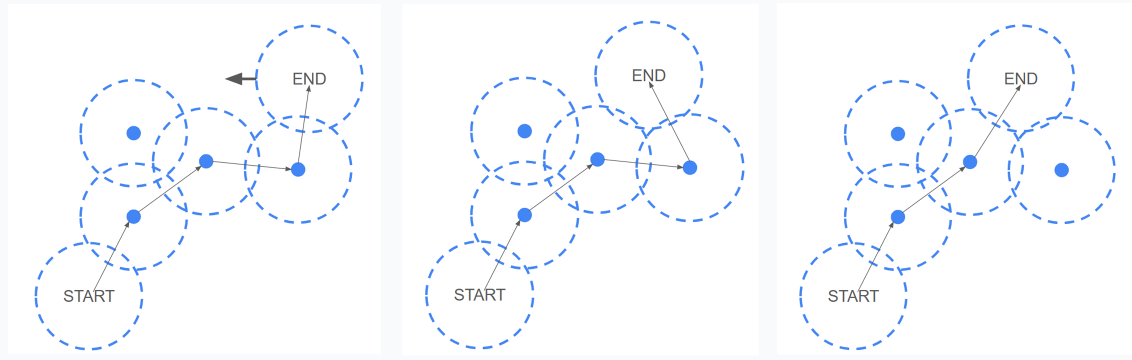
Learning Device-Layer Packet Scheduling in Mobile Networks

Nathan Wu | Quantitative Analysis Center, Wesleyan University | QAC Summer 2026

WESLEYAN UNIVERSITY

1 Decision-Making in Mobile Wireless Networks

Problem: what if no fixed network infrastructure available?
Ex: disaster response, remote sensing, military communication



A message is split into packets. A packet could require multiple transmissions, or hops, to reach its destination

Our setting

Mobile wireless network lets devices communicate directly with nearby devices instead of relying on routers, cell towers, or a fixed base station

Why are decisions in dynamic networks hard?

Network is dynamic: nodes move, wireless links appear and disappear, and traffic accumulates in packet queues. Because the topology changes over time, a good choice now may be not so good a few timesteps later

Packet decision

Which next hop should this packet move to now?

Available path to packet's destination changes as neighbors move

Device decision

Which queued packet should this device forward now?

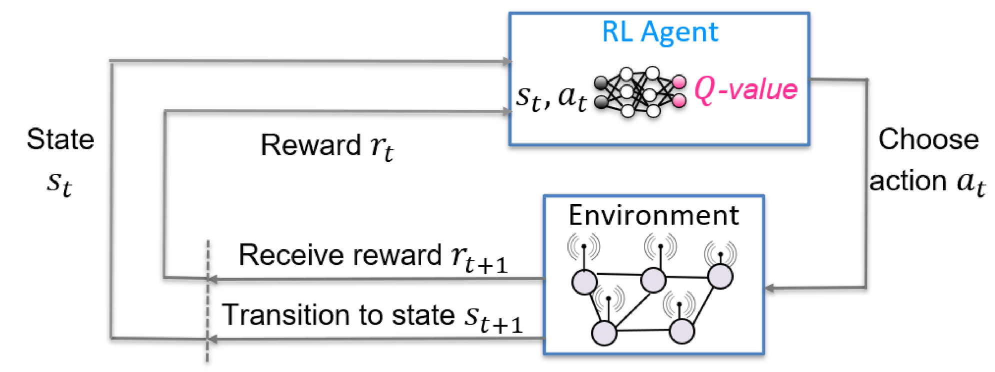
Some current neighbors are better opportunities for some packets vs. others

Goal: study device decision-making

How to **choose packet** whose current forwarding opportunity best supports goals of network operation?

2 Decision-Making Under Uncertainty

Reinforcement learning: for reasoning under uncertainty



Use RL when best action is uncertain and consequences appear later

RL agent is code: it observes data describing **state**, chooses an **action**, receives a **reward** (i.e., number), and updates estimates so actions with higher expected future reward are more likely to be chosen

State	Recorded description of agent's current situation
Actions	Choices available to the agent at that moment
Reward	Immediate consequence of the agent taking action

Q-values: estimate expected future reward

Q(s, a) = expected future reward for action a in state s

Neural network model

Universal function approximator: features in, estimated Q-value out

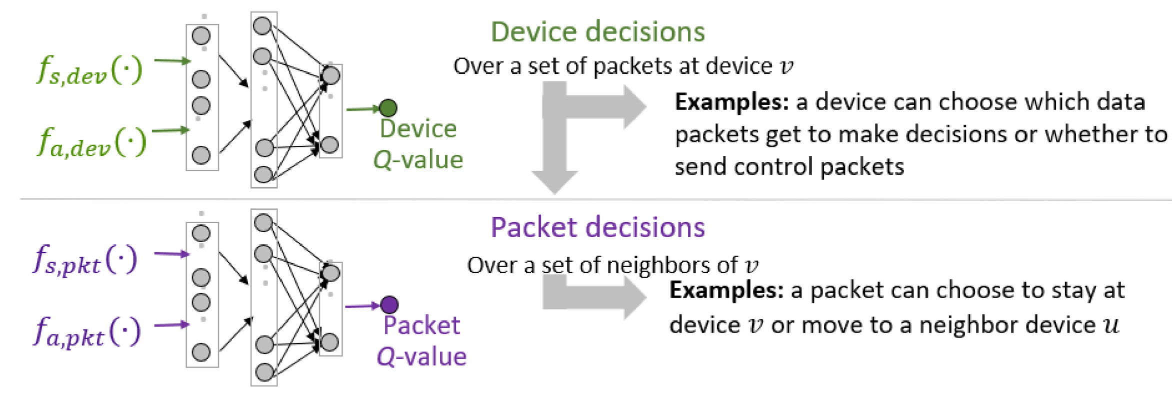
What are features? Values recorded describing environment. If two situations need different actions but have same features, RL agent cannot distinguish

Reward loophole

An RL agent does not understand human goals, only rewards

How can this go wrong? Suppose we design a robot to collect trash for us by rewarding it for placing objects in a trash bin. It might seem like the robot is doing its job correctly, but it may also be placing useful items that should not be thrown away in the trash as well.

3 Using RL for Device and Packet Decisions



Why RL?

Scheduling is **not just 1-step choice**, must take into consideration future. Sending one packet changes **future queue lengths**, **delay**, and **which packets get forwarding opportunities** later. RL is designed for repeated decisions where today's action affects tomorrow's state.

Existing packet agent

State	Describes current packet and its current device, and neighbor/link information
Action	Next-hop neighbor under consideration for packet
Reward	feedback giving immediate (not long-term) consequences of what happened to packet: Delivered? Dropped? Waiting? Forwarded?

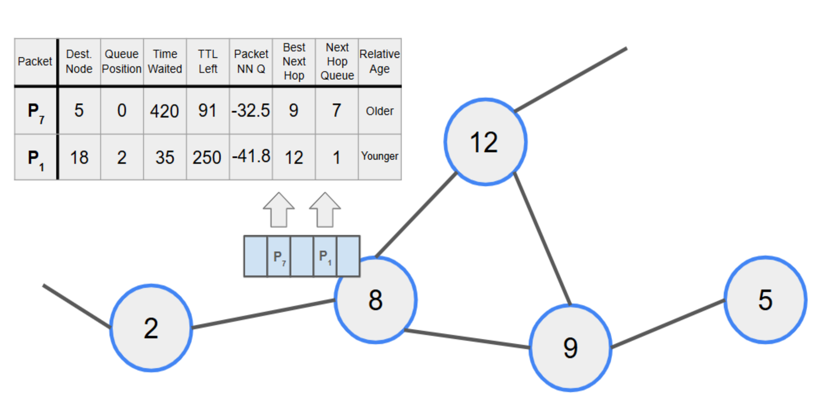
New device agent

State	Describes device's queue represented as candidate packet rows, summary of past forwarding decisions
Action	Queued packet under consideration to be forwarded
Reward	Feedback giving immediate (not long-term) consequences of packet choice

How state is recorded

- For each packet waiting, simulator records a row of features in datafile
- Each row describes that packet's current forwarding opportunity, such as the packet agent's **next hop Q-value**, how long it has **waited**, its **time-to-Live** (TTL), and next hop **congestion**
- Use **neural network** to scores these candidate rows. The chosen action is the selected packet, which uses the packet agent to choose the next hop

4 Designing a Device-Agent Reward Function



Node 8 selection timeline

What the device sees and records when choosing queued packets. Rewards are recorded after packet termination but are not seen by the device till training timesteps.

- **t = 416**: selects filler packet from front of queue; reward: -2.84
- **t = 418**: selects filler packet queue keeps changing; reward: -3.42
- **t = 420**: selects P7 old packet, good path; reward: -2.3
- **t = 421**: selects filler packet normal decision; reward: -3.29
- **t = 424**: selects P1 younger packet, but good path and low queue

Packet 7 timeline

The selected packet's outcome explains why the earlier device choice can receive a good reward.

- **t = 0**: P7 starts at node 2; reward: --
- **t = 1-419**: filler movement or waiting; reward: -1/step
- **t = 420**: P7 reaches node 8 queue; reward: -2
- **t = 421**: node 8 selects P7, P7 moves to node 9; reward: -2
- **t = 422**: P7 reaches node 5 delivered; terminal reward: 0

What info does the device have to help it make a decision?

Packet Q-value	Packet agent's estimate of routing promise for this packet
Time Waited	How long this packet has waited in the network
Time to Live (TTL)	Remaining lifetime before the packet expires
Next-hop queue length	How crowded the likely receiving device is

Our proposed reward combines 4 goals: Reward = $\alpha r + \beta s + \gamma f + \lambda q$

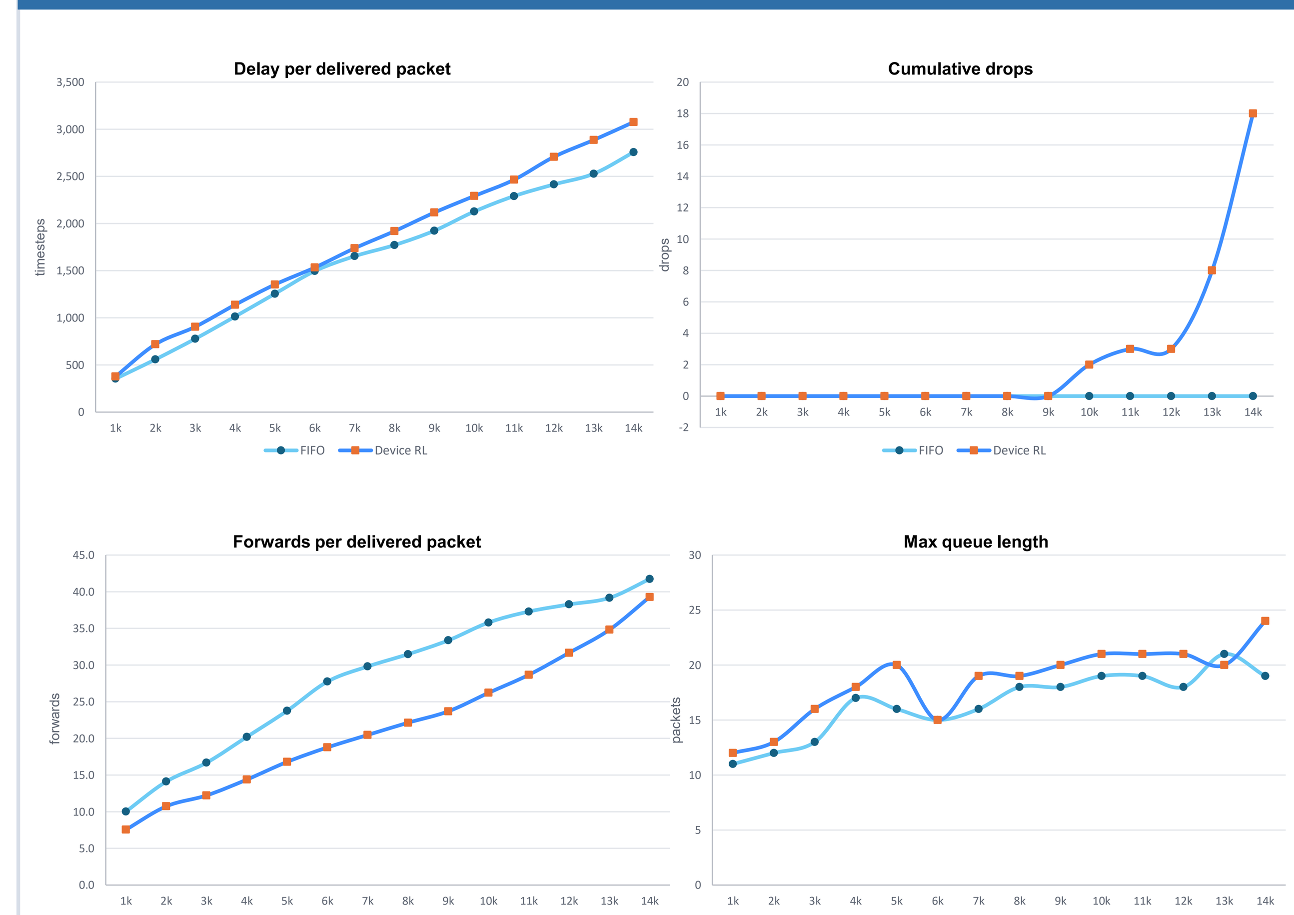
r: reliability	higher when packets are delivered; lower when packets drop or expire
s: speed	higher when delay is lower and the packet has not wasted much lifetime
f: fairness	higher for packets that have waited longer so they are not ignored
q: congestion	higher when the next-hop queue is short; lower when forwarding would add congestion

An example

Packet A	Large packet Q-value, short next-hop queue, waited several timesteps
Packet B	Front of queue, next hop is crowded, TTL is large
Reward logic	A should often score higher because it combines delivery promise, urgency, fairness, and low congestion

This is why the reward cannot optimize only one term. FIFO would pick B, and a congestion-only rule might ignore an urgent packet. The weighted reward makes the scheduler compare several network goals at the same time.

5 Results



6 Analysis & Moving Forward

Preliminary seeded test read: 14,000 timesteps

Packet agent retrained under FIFO; device policy tested without online training.

Metric	FIFO	Device RL	Read
Delivery rate	60.8%	51.0%	lower
Delay / delivered	2,756	3,076	higher
Forwards / delivered	41.7	39.3	fewer
Avg queue	9.44	11.64	higher
Drops	0	18	earlier

What the metrics suggest

- RL is comparable to FIFO meaning the **RL did learn**.
- Higher delay means RL packets are **taking longer to reach destination**.
- Earlier TTL drops suggest some packets take **inefficient paths**.
- Lower forwards means **RL is more selective**, but not always better.
- Higher queue lengths suggest packets wait **longer between chances to move**.

What next?

- Trace TTL drops and high delay per delivered packets: bad next hops = packet issue, bad selection = device issue.
- Retrain packet agent under device scheduling, not only FIFO.
- Compare selected vs skipped packets to see if old packets are actually being ignored.
- Check whether drops cluster around specific nodes; that would point to bottleneck behavior.