Learning an Adaptive Forwarding Strategy for Mobile Wireless Networks: Resource Usage vs. Latency

Victoria Manfredi, Alicia Wolfe, Xiaolan Zhang, Bing Wang

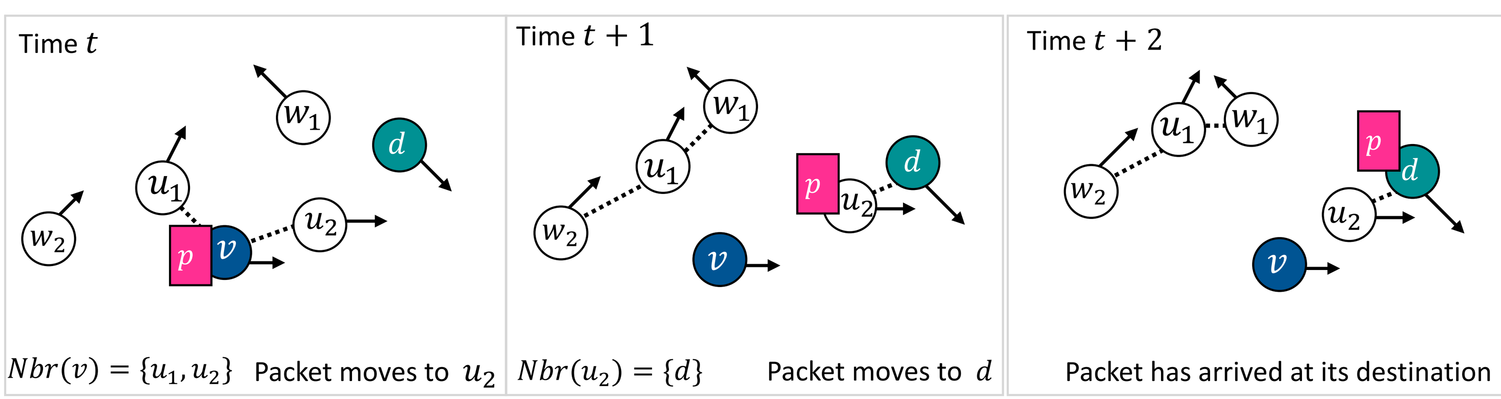
Reinforcement Learning for Real Life Workshop @ NeurIPS 2022

Abstract

Mobile wireless networks present several challenges for any learning system, due to uncertain and variable device movement, a decentralized network architecture, and constraints on network resources. In this work, we use deep reinforcement learning (DRL) to learn a scalable and generalizable forwarding strategy for such networks. We make the following contributions: i) we use hierarchical RL to design DRL packet agents rather than device agents, to capture the packet forwarding decisions that are made over time and improve training efficiency; ii) we use relational features to ensure generalizability of the learned forwarding strategy to a wide range of network dynamics and enable offline training; and iii) we incorporate both forwarding goals and network resource considerations into packet decision-making by designing a weighted DRL reward function. Our results show that our DRL agent often achieves a similar delay per packet delivered as the oracle forwarding strategy and outperforms all other strategies including state-of-the-art strategies, even on scenarios on which the DRL agent was not trained.

1. Packet Forwarding in Mobile Wireless Networks

Devices are moving: infrastructure-less network simplifies deployment

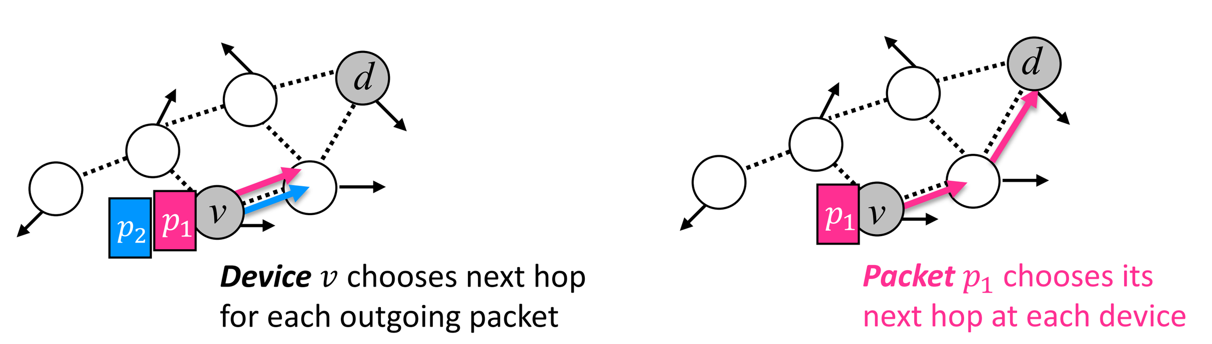


Making optimal forwarding decisions is hard: changing network conditions, limited communication windows, decentralized architecture, competing network goals….

Solution: learn how to forward using deep reinforcement learning

2. Who chooses actions?

Normally a device chooses a packet's next hop … but a device's state doesn't track what happens to the packet



Solution: Use **packet agents** to simplify s, a, s', r experience sequence and reward definition

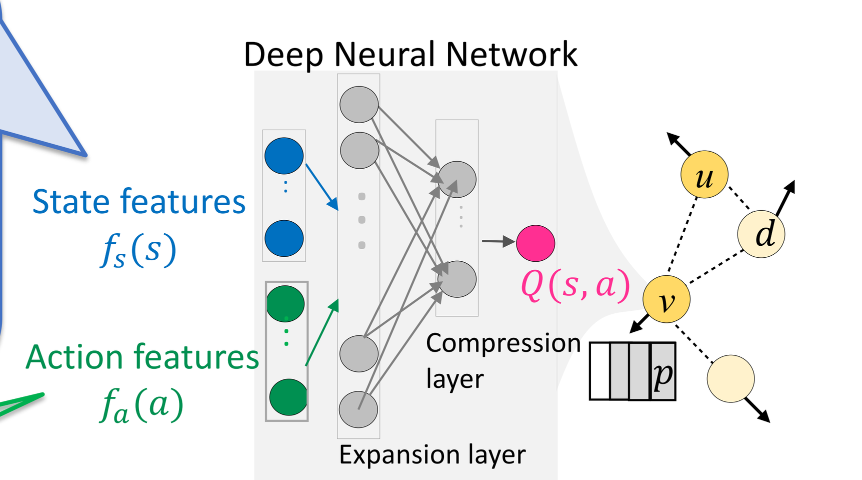
3. Generalizable States and Actions

How to use the same forwarding strategy for topologies with different connectivity and device mobility?

Packet features $f_{packet}(p)$: p's TTL
Device features $f_{device}(v, d)$
- v's queue length, queue length for packets to d, node degree, node density

Neighbor features, $f_{neighbor}(Nbr(v), p, t)$
- summarize varying # of neighbors
- min, mean, max of $f_{device}(Nbr(v), p, t)$

Path features $f_{path}(v, d)$: distance from v to d

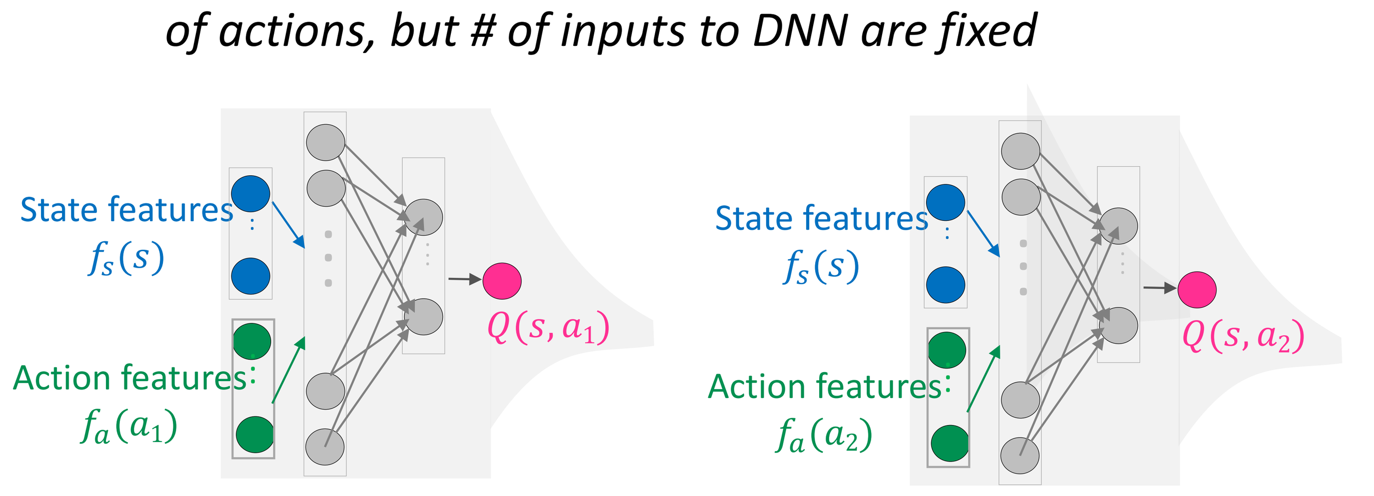


Packet p separately considers each action u
- **device features** $f_{device}(u, d)$,
- **neighborhood features** $f_{nbrhood}(u, d)$,
- **device features** $f_{device}(u, d)$, and
- **context features** $f_{context}(p, u)$ which indicate whether p has recently visited u

Solution: Use **relational features** that model the relationship between devices instead of describing a specific device

4. Varying Numbers of Actions

Varying numbers of neighbors means varying numbers of actions, but # of inputs to DNN are fixed



Solution: Use DNN separately to predict Q-value for **each action** available in a state

5. Competing Goals

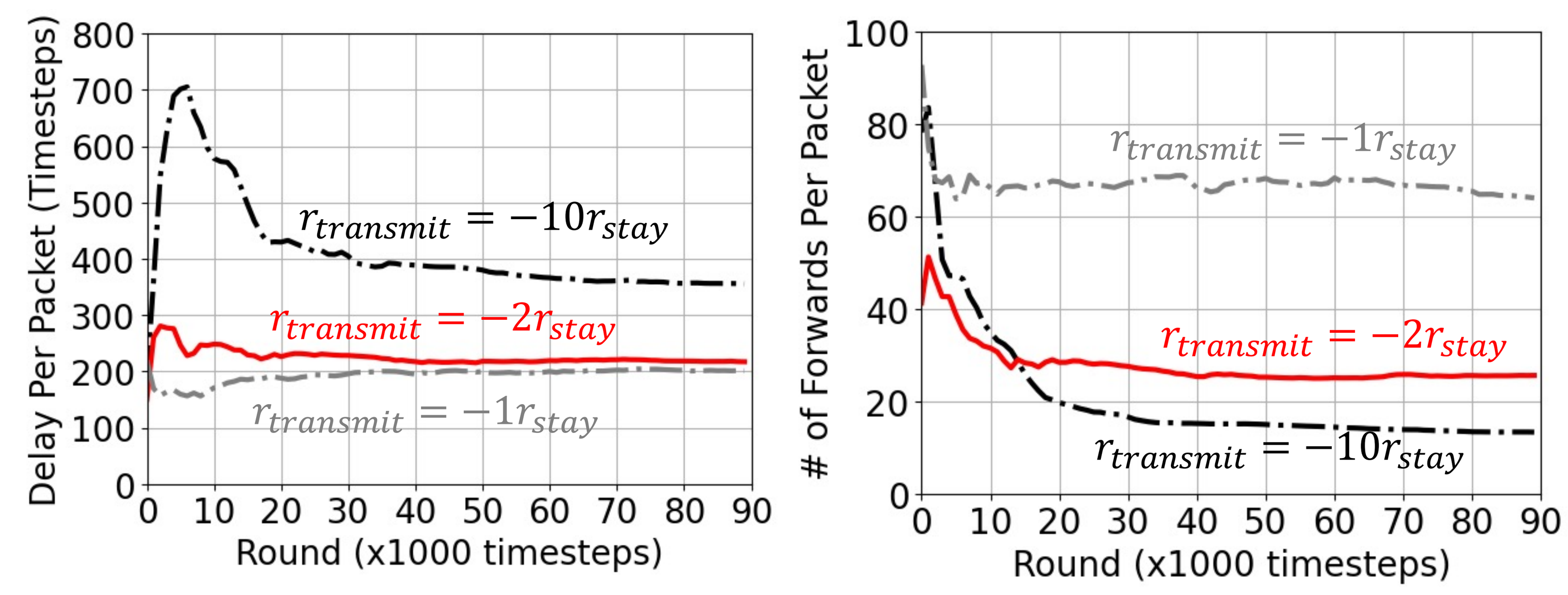
How to trade-off competing network goals such as packet delay vs. resource usage?

Solution: Use a **weighted** reward function

$$r_{stay} = -1, \quad r_{transmit} = \alpha r_{stay}, \quad r_{drop} = \frac{r_{transmit}}{1-\gamma}, \quad r_{delivery} = 0$$

$$\text{where } \alpha = -1, -2, -10 \text{ and } \gamma = 0.99$$

Impact of reward during training



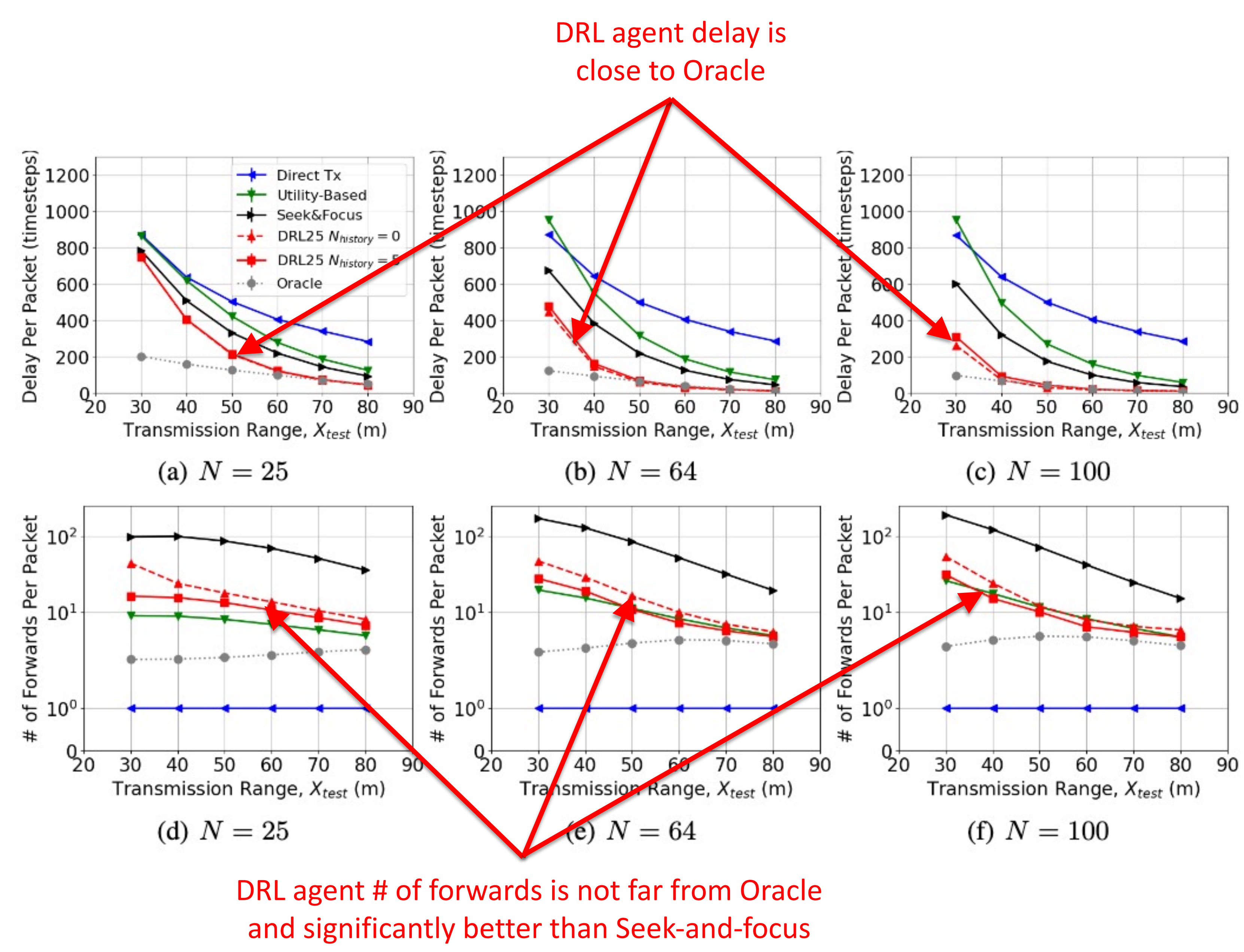
6. Efficient Training

Network architecture is decentralized, complicating training as exchange of training information is limited by communication opportunities

Solution: Use **offline training**, leveraging relational features. **Use fixed policy options** to model decisions that take multiple timesteps, like a packet waiting in a queue before making a forwarding decision.

7. Performance Evaluation

Compare the performance of DRL agent, an oracle-based delay minimizing strategy (Oracle), a transmission minimizing strategy (Direct transmission), and two state-of the art strategies (Utility and Seek-and-Focus)



Takeaways: DRL agent is trained on **25 device network with tx range of 50m**, generalizes to 64 and 100 device networks with **tx ranges of 30 to 80m**

8. Conclusions and Future Work

Conclusions: possible to use DRL to learn a scalable and generalizable forwarding strategy for mobile wireless networks. We propose three key ideas: i) packet agents, ii) relational features, and iii) a weighted reward function.

Future work: feature ablation, more diverse mobility, more features to characterize diverse mobility, device decision-making to complement packet agents, and refining reward function.

Acknowledgements: This material is based upon work supported by the National Science Foundation (NSF) under award #2154190. Results presented in this paper were obtained in part using CloudBank, which is partially supported by the NSF under award #2154190. The authors also acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing HPC and consultation resources that have contributed to the research results reported within this paper.