Reinforcement Learning-based Congestion Control

A Systematic Evaluation of Efficiency and Fairness

Luca Giacomoni
University of Sussex

Challenges Faced by Congestion Control Protocols

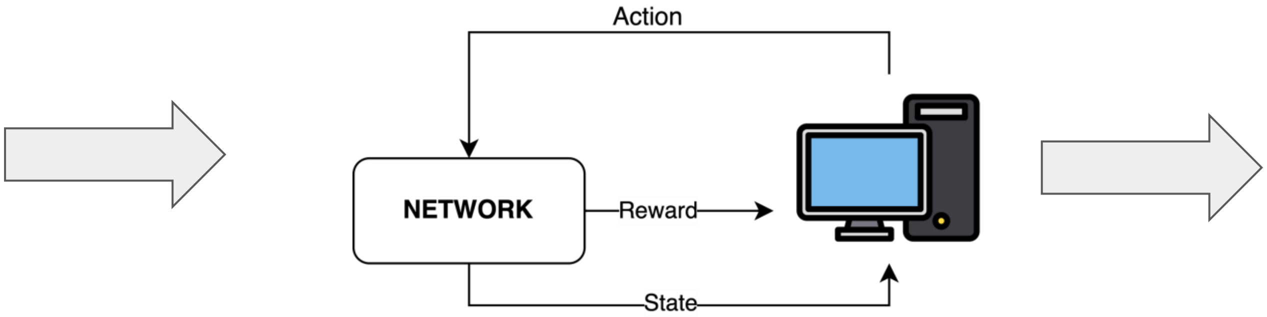
Congestion Control Protocols are responsible for efficient and fair distribution of network resources. However, they face several challenges:

- **Dynamic Network Conditions**: The volatile nature of network parameters such as bandwidth, delay, and loss rate requires CC protocols to adapt in real time, which is a complex task.
- **Adapting to New Technologies**: Emerging technologies, such as 5G or IoT devices, pose new challenges requiring CC protocols to adapt and evolve.
- **Performance in Diverse Scenarios:** CC protocols must perform effectively in varied network scenarios, like data centers, mobile networks, satellite communications, etc.

The Convergence of RL and CC

- RL-based CC optimises CC protocols by interacting with the environment and optimising a given reward function.
- Elements:
    - Reward function (throughput, delay, loss rate)
    - Observable network state space (delay variation, loss rates, throughput)
    - Action space (cwnd update value, transmission rate)
    - Learning algorithms (PPO, DDPG)

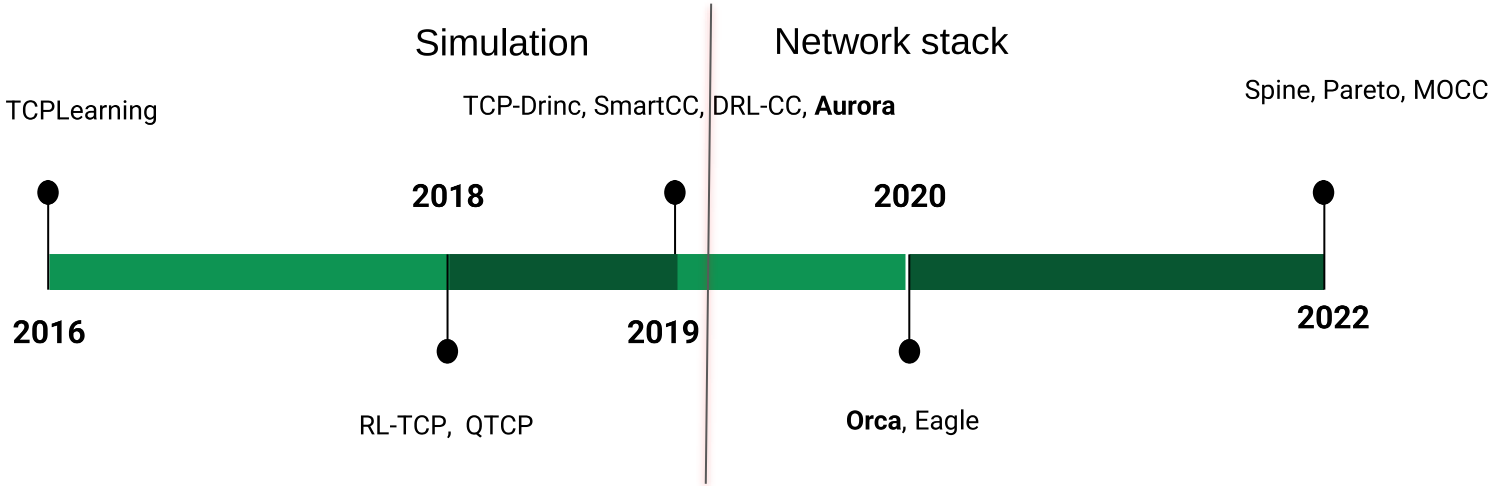
Model



Protocol

The Current State-of-the-Art

- RL-based approaches have seen varied implementations
  - Numerous simulation-based implementations (ns-2, ns-3 models)
  - Growing number of real network stack implementations (UDT, Linux Kernel)



The Real-World Viability of RL-CC (Ignoring Computational Complexity)

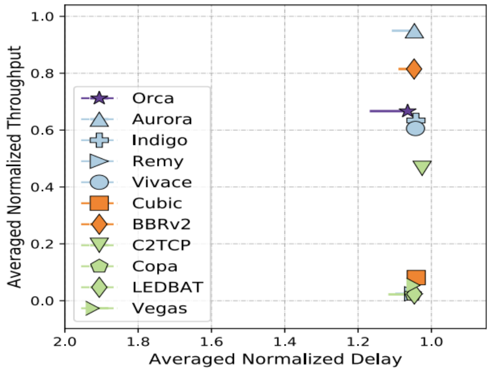
- Evaluation of methodologies of RL-CC approaches performance on real network stack present some limitations:
  - Single-flow efficiency focused
  - Black-box experimental setup difficult to interpret
  - Limited reproducibility

Challenges in Interpretation

- Hidden parameters and black-box evaluation obstruct interpretability. For example:

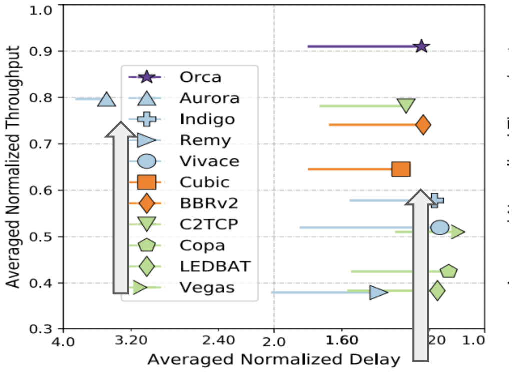
  - 1) Is the network buffer shallow? Does higher throughput necessarily indicates improved efficiency?
  - 2) What leads Aurora to fill the buffer when Cubic doesn't?

1)



(a) Inter-Continental experiments

2)



(b) Intra-Continental experiments

What can we do?

- Evaluation methodologies of CC offer a trade-off between **fidelity**, **flexibility**, **reproducibility** and **transparency**
    - Simulations: ++Reproducibility, - -Fidelity, ++Flexibility, ++Transparency
    - Emulation: +Reproducibility, -Fidelity, +Flexibility, +Transparency
    - “In the wild”: -Reproducibility, +Fidelity, -Flexibility, - -Transparency

- Challenges exacerbated by the **black box** nature of **RL-CC**
    - Decision making encoded into complex parametric functions (e.g. Neural Networks)

Objectives and Methodology

To develop a **transparent**, **flexible**, and **reproducible** benchmark for RL-CC efficiency and fairness.

- Methodology: Use Mininet emulation-based evaluation framework
- Measurement Metrics: Goodput, Congestion window sizes, srtt value, Retransmissions, Link and system utilization
- Several control variables:
  - Base RTT of competing flows (intra/inter)
  - Buffer size at the bottleneck (20%,100%, 400% the BDP)
  - Bottleneck bandwidth (10Mbps to 100Mbps)
  - AQM (CoDel and FQ)
  - TCP Friendliness
- **Transparency** and **reproducibility**:
  - Kernel config (e.g. TCP Buffers, segmentation offload)
  - All experiments code, configuration and data **open sourced**

Protocols Under Examination: Orca & Aurora

Orca

- Hybrid approach: trained agent acts on top of Cubic
- Reward: based on Power (Kleinrock's operational point)
- Control granularity: Coarse and fine
- Implementation: Linux kernel + application layer
- Training: on emulated environments

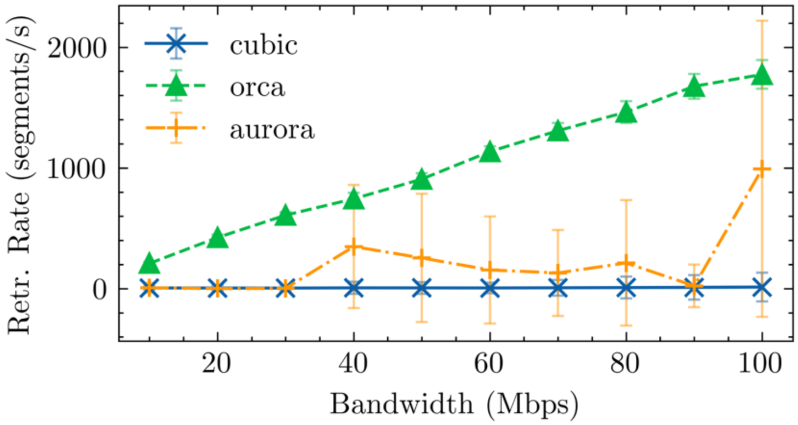
Aurora

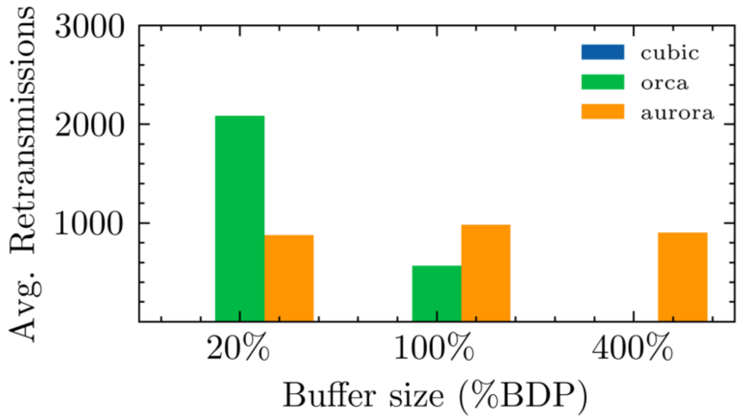
- Clean-slate approach: decision making purely RL-based
- Reward: linear combination of throughput, delay and loss
- Control granularity: Coarse
- Implementation: UDT
- Training: on custom simulator

Results

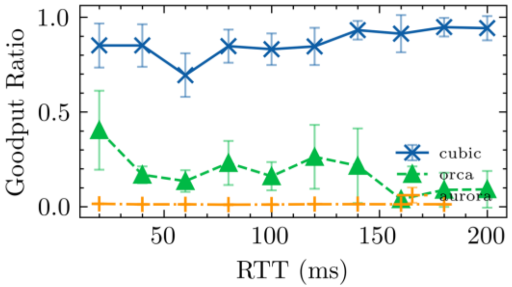
Goodput vs Retransmissions

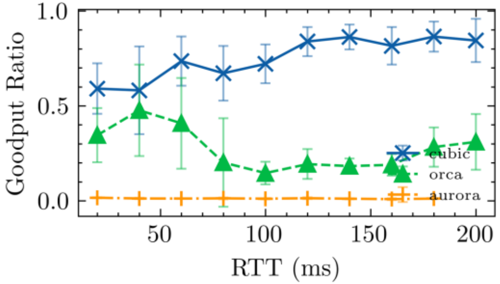
- For small buffers, Orca's aggressiveness can cause up to three orders of magnitude higher number of retransmissions than Cubic

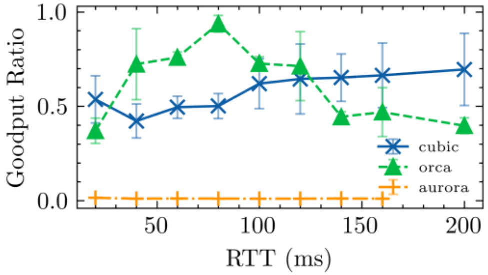




Intra RTT Fairness







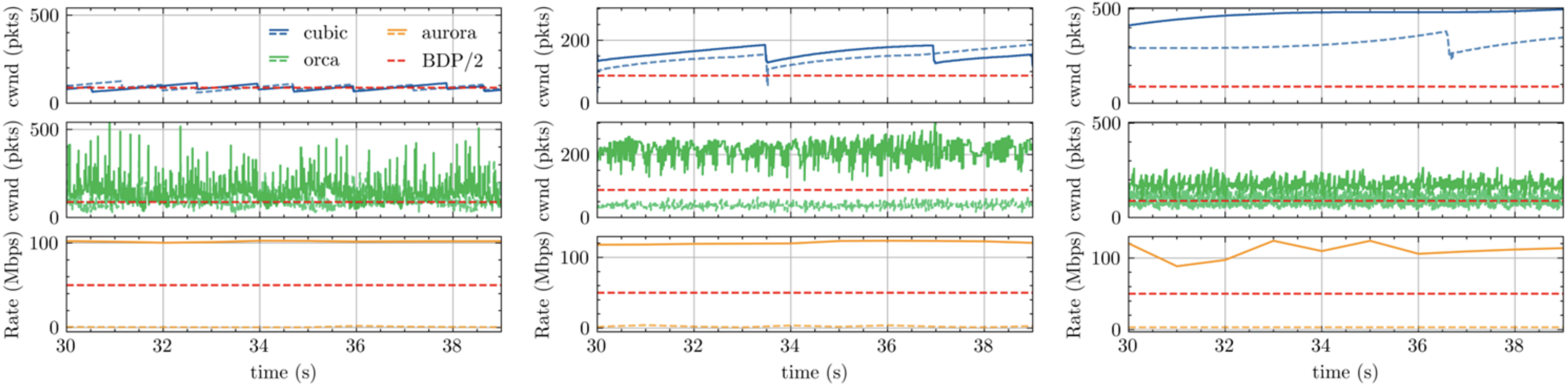
Buffer size: 20% BDP

Buffer size: 100% BDP

Buffer size: 400% BDP

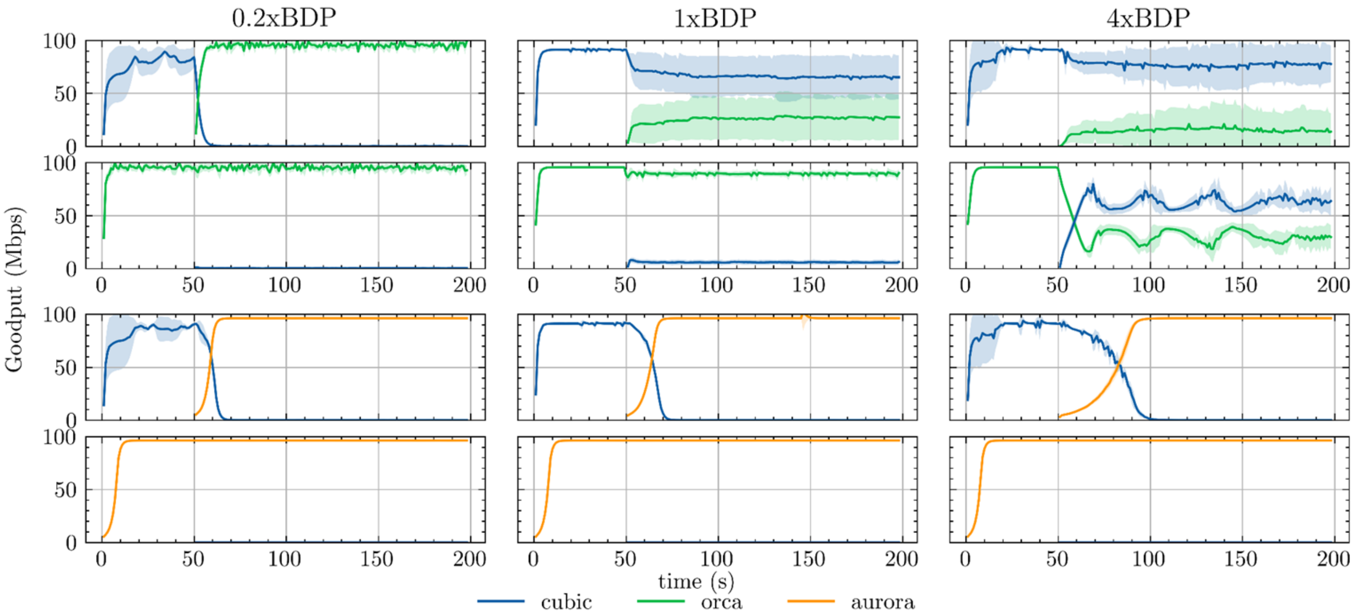
Goodput ratio for two competing flows in a dumbbell topology. Bottleneck capacity is 100Mbps, both flows experience the same base RTT (shown on x-axis), buffer capacity is set to 0.2x, 1x, and 4x the BDP.

Under the hood



(a) RTT: 20ms, Buffer Size: 0.2× BDP (b) RTT: 20ms, Buffer Size: 1× BDP (c) RTT: 20ms, Buffer Size: 4× BDP

Backward compatibility



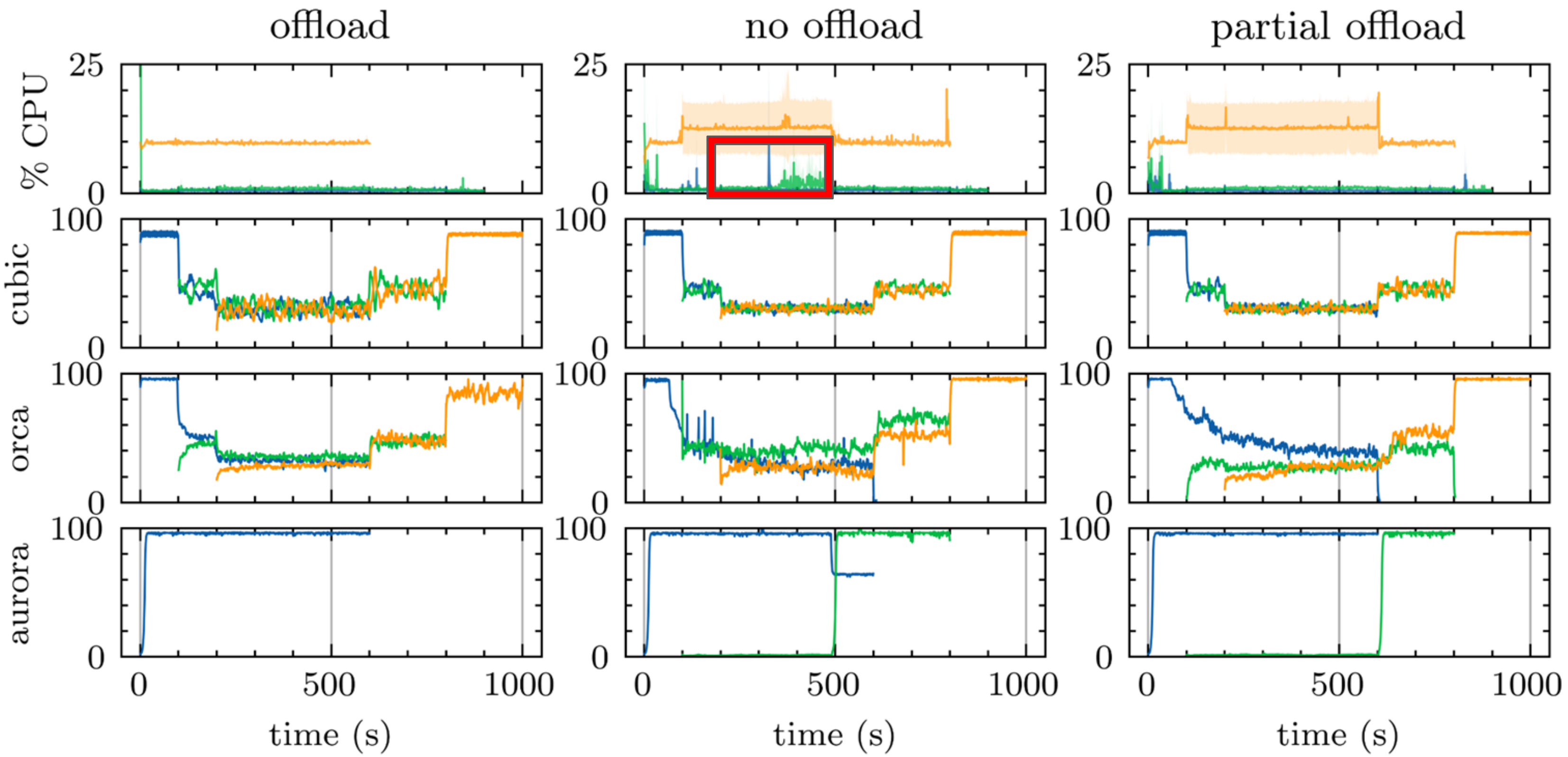
Goodput evolution for two competing flows (one being TCP Cubic) in a dumbbell topology. Bottleneck capacity is 100Mbps, both flows experience the same base RTT (100ms), buffer capacity is set to 0.2x, 1x, and 4x the BDP.

Conclusion

- The complexities of evaluating RL-based CC algorithms have resulted in under-reporting limitations in existing work, particularly with regards to efficiency and fairness.
- We show this through a systematic study of existing RL-based CC proposals with transparency and reproducibility as key objectives.
- We devise a methodology and set of benchmark experiments tailored to examine efficiency and fairness
- We present empirical data analysing the performance of Orca and Aurora.

Thank you!

Segmentation Offloading



Delay gain