Online Learning on the Programmable Dataplane

Kyle A. Simpson
✉ k.simpson.1@research.gla.ac.uk
FelixMcFelix   https://mcfelix.me

September 16, 2022

University of Glasgow

Main Thesis – Summarised

The sum of data-driven networking (DDN) and programmable dataplane (PDP) developments are key tools in improving management of computer networks.

- DDN automates, increases efficacy in control/optimisation problems.
- PDP hardware allows more precise input data for DDN, faster.
- ML/RL in PDP hardware lets us learn from & act on network events with less delay.
- PDP useful to reduce data rates – make host classification work at scale.

Motivation (I)

Many control/design mechanisms are already heuristic – CCAs, routing, network defence, content delivery/placement...

- DDN can offer improved mechanisms, or tailored control over tuning parameters.
- Catch: need these to run fast!

Many problems already feedback-loop like:

- Constant measure-act cycles.
- Clear optimisation criteria.

Problem: costs of ML inference and learning are high.

Problem: traditional network data are low-quality, sampled state.

Motivation (II)

Advances in PDP and In-Network Compute help here:

- Architectures designed for line-rate processing.
- More fine-grained state – timestamps, flow monitoring.
- Per-packet flow state *must* be handled in-situ.

Host-based execution has steering, stack costs that add to latency and harm throughput.

Supporting the thesis

How is this explored? **Three strands of work:**

Pure DDN	ML *in* PDP	PDP *for* ML
Ch. 4 & 6	Ch. 5	Ch. 6
RL for anti-DDoS, CCA detection	Online RL in the dataplane	In-dataplane telemetry reduction
More effective (env. dependent)	**Strong online latency + t’put benefits**	**Significant packet rate reduction**
Need to really understand environment, protocols	*Limited devices, sacrificed model capacity*	*Data reduction needs tailored to problem*

Other use cases and methods supported by recent literature – Ch. 2–3.

Wider takeaways and conclusions

- DDN can be very effective *when it works*:
    - Hard to deploy and design.
    - Deep knowledge of environments needed.
- Co-design is key!
    - Expertise on system dynamics – **network and HW/SW architectures**.
    - PDPs excellent for general offload...
    - ...but ML offload still needs deep human insight.
    - Future online works will need more insight at HW level *á la* Taurus.
- Security and explainability look bleak:
    - Extraordinary care needed around input data, training experience, and scope of outputs.