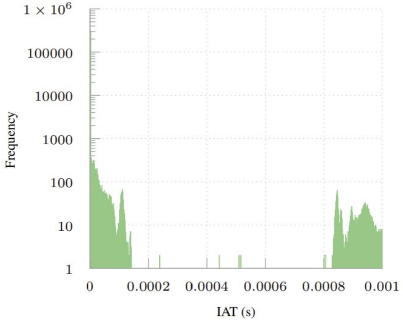
Seiðr: Dataplane Assisted Flow Classification Using ML

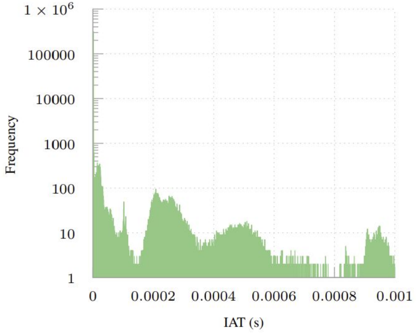
Kyle A. Simpson, Richard Cziva, Dimitrios P. Pezaros.
University of Glasgow & Lawrence Berkeley National Laboratory

IEEE GLOBECOM 2020.

Introduction and Motivation



(a) TCP Cubic

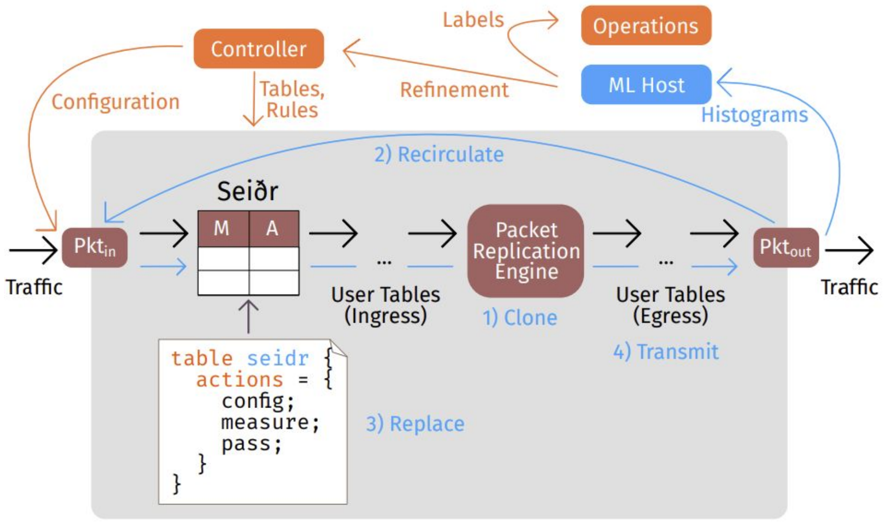


(b) TCP BBR

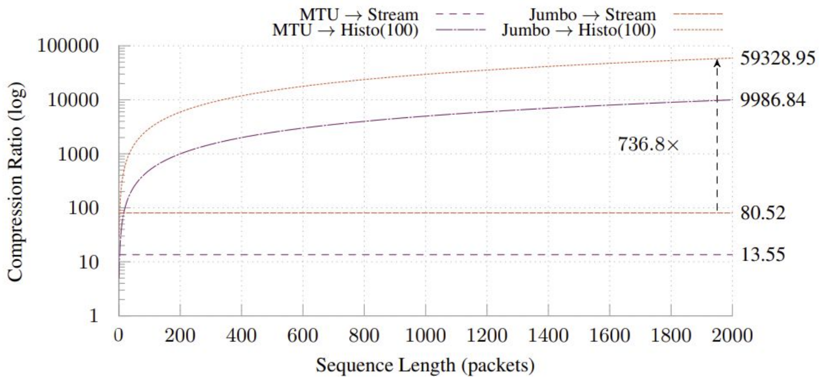
- Want to understand network usage and behaviour: flow classification.
  - E.g., TCP flavours, application type/behaviour.
  - Main differences may lie in *distribution* of a feature.
- Programmable NICs & switches can convert to telemetry, reducing data rate.
  - Per-packet info, accurate timing...
  - But how to scale to > 100Gbps? Many flows with small packets?
- SOLUTION: aggregate measurements in the dataplane.

How?

- For flows matched via control plane (`table seidr`):
    - Maintain hash table of histograms.
    - Record packet IAT/field/property in matched histogram.
    - Forward packet onto next tables.
    - If enough data, generate histogram via clone + recirculate + replace.
- Compatible with P4 *Portable Switch Architecture*.
- Histograms sent to any classifier/collector.
- Runtime configurable.



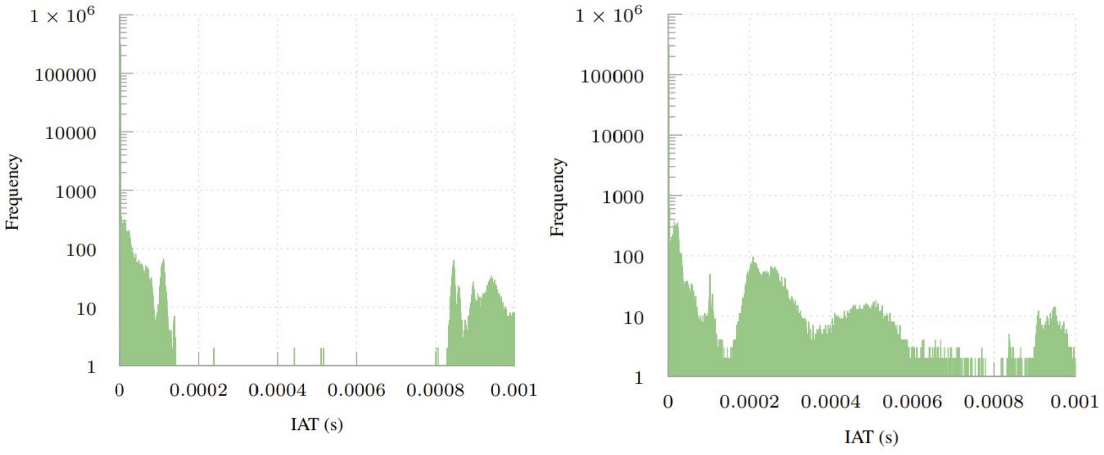
Calculated data rate reduction



- Packet conversion to telemetry gives O(1) volume reduction...
  - But no reduction in packet rate!
  - 1Mpps client ingest bottleneck.
- Histograms also reduce packet rate by 1/seq_len
  - Overcome host ingest bottleneck
  - Linear volume reduction (O(seq_len))
- 100Gbps in => 10 Mbps out, 33.3kpps

Use case: TCP flavour detection

- Other works show TCP BBRv1 *unfairness*.
- Can't control CCA usage in large (transit) WANs.
- BBR's algorithm has key differences.
- Main IAT differences:
  - Distributional
  - Fine-grained (sub-ms)



(a) TCP Cubic

(b) TCP BBR

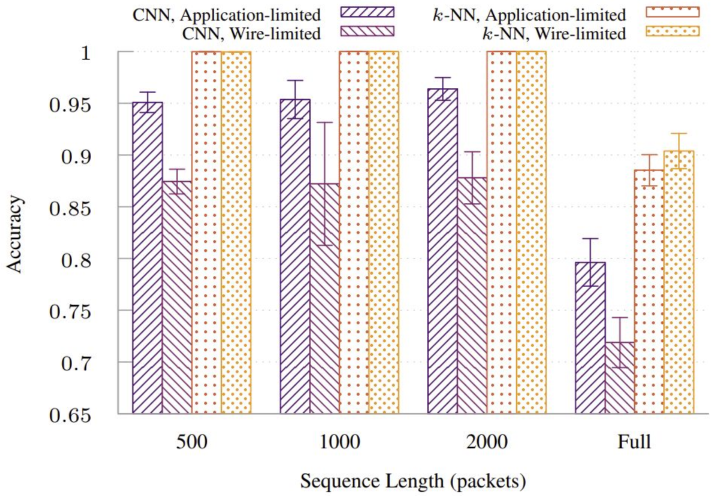
BBR Classification – methods and cost

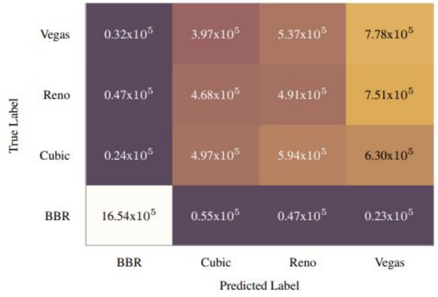
- Approaches
  - Convolutional Neural Networks
  - *k*-Nearest Neighbours
- CNNs longer to train, cheaper to run
  - Low memory use, fast per-histogram test time.
- Online analysis with *k*NNs not feasible
  - Dataset size, memory cost, execution cost.
  - Train cost paid *every time* model *k*NN is created.

Family	Online/Subsequence	$n_{classes}$	Train	Test	Memory
CNN	✓	2	(43 ± 2) min	(49.1 ± 9.2) µs	409.76 KiB
	✓	4	(243 ± 2) min	(50.5 ± 1.7) µs	410.27 KiB
	✗	2	(1.82 ± 0.47) s	(161.3 ± 3.9) µs	409.76 KiB
	✗	4	(7.94 ± 0.50) s	(137.7 ± 1.2) µs	410.27 KiB
k-NN	✓	2	(21.4 ± 1.2) min	(323 ± 69) µs	2.1 GiB
	✓	4	—	—	12.58 GiB
	✗	2	(0.20 ± 0.06) s	(54.0 ± 0.3) µs	332.8 KiB
	✗	4	(2.20 ± 0.04) s	(517.0 ± 5.0) µs	2.0 MiB

Classification accuracy

- Good at detecting BBR:
  - Online CNNs see >= 85% accuracy, peak F1=0.965.
  - ...Accuracy falls to ~50–60% if we add other TCP flavours. *(Not shown)*
  - *BUT*—BBR still incredibly distinct from predecessors.
  - Conclusion: BBR's *timer*-based approach is detectable vs. classical *cwnd*-based.
- Future?
  - QUIC can use BBR: can we unmask flows this way?
  - Other flows with interesting temporal properties? VoIP?
  - BBR v2? FastTCP?





(a) *Application-limited.*
$F1_{BBR} = 0.935$, $F1 = 0.486$.

Conclusions

- Statistical aggregation can be done in P4.
  - Here, via histogramming.
  - Compliant with the *Portable Switch Architecture*.
  - **Significant data and packet rate reduction**.
- TCP BBR can be told apart from its predecessors.
  - Timer-based algorithm => **inter-arrival times differ**.
  - Differences in **distribution of measurements**.
  - Doesn’t work on older variants.
- Histogram reduction works well with this type of classification.
  - Good performance with CNNs, *k*NNs.
- Future—QUIC unmasking?

Questions?