Real-time Performance Analysis of High-Speed, International Science Network Flows

Kyle A. Simpson[1], Richard Cziva[2], Yatish Kumar[2], Chin Guok[2]
[1]*University of Glasgow*, [2]*Energy Sciences Network (ESnet)*
k.simpson.1@research.gla.ac.uk

Science networks carry **high-speed, critical traffic** from scientific instruments to researchers and laboratories. Instruments at CERN generate **petabytes of data per second**, which is actively distributed on a global compute and storage infrastructure. The majority of science flows use TCP, with countless variants of congestion control algorithms in use. **Measuring these flows is a balancing act**: traffic sampling is cheaper, yet packet-based processing is required for tasks such as microburst detection [1]. Collecting telemetry for every packet would improve existing measures and enable new analyses, but presents a difficult engineering challenge.

We present a system to perform **per-packet, fine-grained monitoring**. We use **P4 programmable hardware** to implement a telemetry agent that converts packets into a digest with a **nanosecond-accurate timestamp**. Several of these switches are being deployed in our multi-100 Gbit/s network. Digests are analysed by dedicated collector servers, **engineered to run at wire-rate**.

Architecture

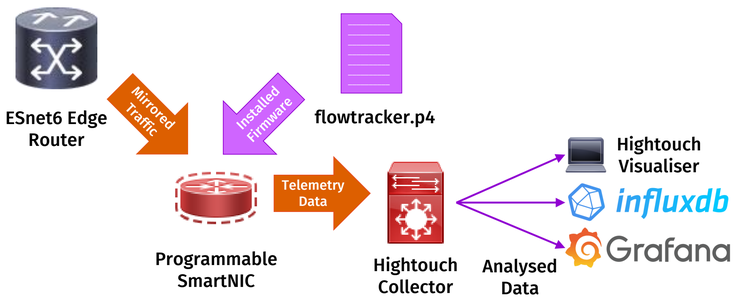

Fig. 1: Integration of a High Touch collector and flow timestamping at the network edge.

To better monitor flows in the next iteration of our large science network—**ESnet6**—we move flow analysis **to the network edge** (Fig. 1). TCP packets are mirrored, converted to **digests with nanosecond timestamps at MAC ingress** by P4-compatible [2] Agilio CX SmartNICs, and forwarded to collectors for processing. Collectors make data available by **both a live WebSocket stream** for real-time use cases, and via **mid-term storage in a time series database** for second-stage collectors or visualisation.

This model allows **data reduction, timestamping, and processing at line-rate**, while the control plane lets us determine where flow filtering applies, and which processing occurs. Custom, easily deployable P4 firmware lets us **adapt** our collectors, formats, applications as the network evolves.

Point Rates

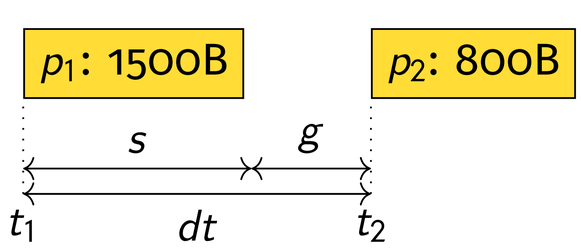

Fig. 2: Point rates. Note that p_1 and p_2 may be from different flows.

Accurately measuring the ingress time of every packet allows us to approximate the time each packet spends on the wire. In turn, we can measure the **point rate** associated with each packet. Taken individually, these are counter-intuitive, but can reveal **interesting distributional characteristics** in aggregate (Fig. 5).

Collectors–Stateful TCP Analysis

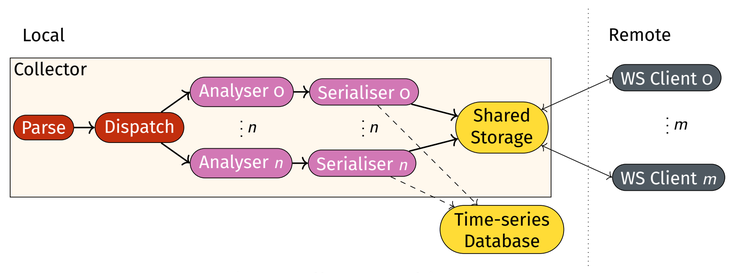

Fig. 3: Collector architecture.

We implement a collector application in **Go** to perform TCP flow analysis from High Touch packets, including the following algorithms:

- Rate Monitoring (point estimate and sliding window).
- Packet loss and retransmission detection.
- Online SRTT via Karn's algorithm [3].
- Bytes-in-flight tracking.
- Congestion window size estimation via Ghasemi et al.'s algorithm [4].

We use a **pipelined design** (Fig. 3) to maximise throughput where possible.

Metric	Parse	Dispatch	Analysis	Serialisation
Packet rate (kpps)	1300	775	310–485	675
Flow rate, jumbo frames (Gbit/s)	93.6	55.8	22.3–34.9	48.6
Flow rate, 1500 B (Gbit/s)	15.6	9.3	3.7–5.8	8.1

Tab. 1: Pipeline processing rates.

We instrument each stage using **Prometheus**; the ingress pipeline then has a peak capacity of **55.8 Gbit/s between all flows**, and **22.3–34.9 Gbit/s per flow** (depending on which analyses are performed). Deployment machines will be considerably more powerful, and experiments in *Rust* suggest higher parsing throughout.

Measuring At Midpoints

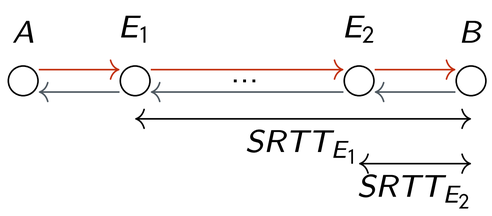

Fig. 4: SRTT computation of unidirectional flows as seen by collectors. Packets carrying data are marked by a red arrow.

Given that many flows are effectively unidirectional in byte volume, TCP analysis performed between endpoints means that **certain metrics describe a portion of the path** (SRTT, bytes in flight). Gathering these metrics at **both WAN ingress/egress** may be useful for diagnosing anomalous path behaviour.

TCP Flavour Rate Distribution

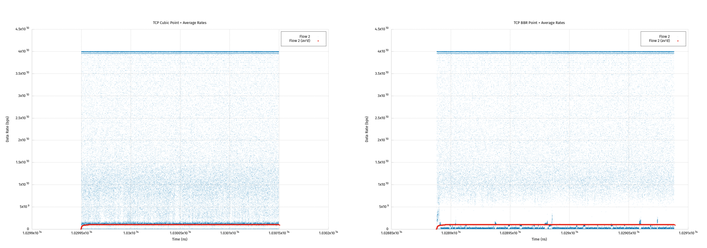

Fig. 5: Point (blue) and sliding window (red) rates at 1 Gbit/s. TCP Cubic is on the left, BBR is on the right.

Distributions of point rates over time vary for different flows—**suggesting e.g., different congestion control algorithm behaviour**. Close inspection reveals that **BBR point rates have more structured distribution**.

Next Collectors

Stateful TCP analysis of this kind acts as a **pilot project** for the feasibility of the system. We intend to explore other collector designs, including:

- **Peak utilisation tracking** (i.e., at 1 s granularity).
- **Monitoring microbursts** in the network via timing information.
- Visualisation/measurement of **queueing**.

TCP Flavour Identification

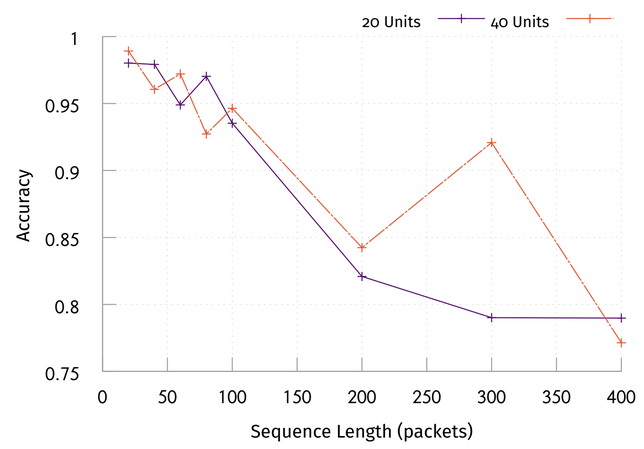

Fig. 6: Accuracy in telling apart TCP Cubic and BBR flows.

Motivated by the differences in point rate distribution between flows, we have have begun preliminary testing of TCP flavour identification using **long short-term memory** (LSTM) networks [5], which are well suited to time-series classification. We currently see **98.9 % prediction accuracy in as few as 20 measurements** for unmultiplexed flows.

Datasets

We provide telemetry packet captures and collector output for solo and pairs of multiplexed flows for $\{100, 200, \ldots, 1000\}$Mbit/s, using TCP *Cubic*, *BBR*, *Reno* and *Vegas* in *iperf3*. This is available at:
`https://tinyurl.com/esnet-ht-data`.

Future Work

- **Correlating results between collectors** (and switch-to-switch time synchronisation [6]).
- Move control of **which analyses are performed** to the control plane.
- Investigate **analysis *on* SmartNICs**.
- Handling **asymmetric flows**.

References

[1] X. Chen et al., ***Catching the microburst culprits with snappy***, in *Proceedings of the Afternoon Workshop on Self-Driving Networks, SelfDN@SIGCOMM 2018, Budapest, Hungary, August 24, 2018*, ACM, 2018, pp. 22–28.

[2] P. Bosshart et al., ***P4: programming protocol-independent packet processors***, *Computer Communication Review*, vol. 44, no. 3, pp. 87–95, 2014.

[3] P. Karn and C. Partridge, ***Improving round-trip time estimates in reliable transport protocols***, *Computer Communication Review*, vol. 17, no. 5, pp. 2–7, 1987.

[4] M. Ghasemi et al., ***Dapper: Data plane performance diagnosis of TCP***, in *Proceedings of the Symposium on SDN Research, SOSR 2017, Santa Clara, CA, USA, April 3-4, 2017*, ACM, 2017, pp. 61–74, ISBN: 978-1-4503-4947-5.

[5] S. Hochreiter and J. Schmidhuber, ***Long short-term memory***, *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] P. G. Kannan et al., ***Precise time-synchronization in the data-plane using programmable switching ASICs***, in *Proceedings of the 2019 ACM Symposium on SDN Research, SOSR 2019, San Jose, CA, USA, April 3-4, 2019.*, ACM, 2019, pp. 8–20, ISBN: 978-1-4503-6710-3.