SHORT-PAPER

Poster: Circa: Re-imagining Network Telemetry from an Approximation-First Perspective

MILIND SRIVASTAVA, Carnegie Mellon University, Pittsburgh, PA, United States

VYAS SEKAR, Carnegie Mellon University, Pittsburgh, PA, United States

Published: 04 August 2024

Citation in BibTeX format

ACM SIGCOMM Posters and Demos '24: ACM SIGCOMM 2024 Conference: Posters and Demos
August 4 - 8, 2024
NSW, Sydney, Australia

Conference Sponsors:
SIGCOMM

ACM SIGCOMM Posters and Demos '24: Proceedings of the ACM SIGCOMM 2024 Conference: Posters and Demos (August 2024)
https://doi.org/10.1145/3672202.3673742
ISBN: 9798400707179

Poster: Circa: Re-imagining Network Telemetry from an Approximation-First Perspective

Milind Srivastava, Vyas Sekar
Carnegie Mellon University

CCS CONCEPTS

• **Networks** → **In-network processing**; **Network monitoring**; **Network measurement**.

ACM Reference Format:
Milind Srivastava, Vyas Sekar, Carnegie Mellon University. 2024. Poster: Circa: Re-imagining Network Telemetry from an Approximation-First Perspective. In *ACM SIGCOMM 2024 Poster and Demo (SIGCOMM Posters and Demos '24), August 4–8, 2024, Sydney, NSW, Australia.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3672202.3673742

KEYWORDS

Network telemetry, Monitoring, Sketches, Approximation-first

1 INTRODUCTION

Telemetry systems are widely used to collect data from distributed endpoints, analyze data in conjunction to gain valuable insights, and store data for historical analytics. These systems consist of four stages (Figure 1): collection, transmission, analysis, and storage. Collectors at the endpoint collect various types of data, which is then transmitted to a central server for analysis. This data is used for multiple downstream tasks, such as dashboard monitoring and anomaly detection. Finally, this data is stored in long-term storage to aid retrospective analytics and debugging.

Different telemetry systems incur bottlenecks at different stages. For instance, a city-wide deployment of IoT sensors to measure air pollution would be bottlenecked by data collection and wireless transmission from the low-power IoT sensors. In contrast, network telemetry from a large datacenter may be bottlenecked at analyzing TBs of network traffic in real-time, and storing it to enable future debugging and longitudinal analysis.

Problem: With increasing volumes of data to be collected and analyzed, and the increasing need for real-time analytics, telemetry costs are rising across the stack. Thus, simply collecting all data, transmitting it for analysis, and storing it exactly has become prohibitively expensive.

While there are multiple proposals to reduce telemetry costs at different stages, these are often stopgap and myopic. For instance, a canonical method to reduce collection cost is to sample data. While sampling incurs low effort and is used widely, it often results in poor cost-accuracy trade-offs, forcing the operator to choose between low cost and high accuracy. To reduce transmission cost, one might employ techniques to naively aggregate data across multiple collectors. However, this results in low-fidelity inflexible telemetry, ultimately hurting the operator.

ACM ISBN 979-8-4007-0717-9/24/08.
https://doi.org/10.1145/3672202.3673742

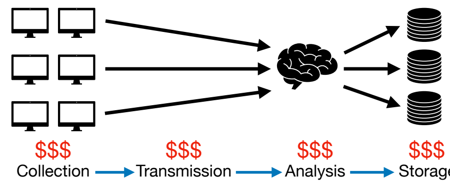

Figure 1: Telemetry costs are rising across the stack.

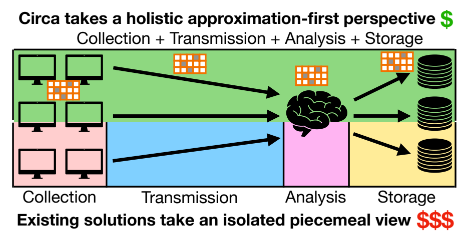

Figure 2: Existing solutions take an inefficient isolated view of the telemetry stack. Circa takes a holistic approximation-first view to enable efficient telemetry using sketches.

A key limitation of existing solutions is that their design is built for exact telemetry and support for approximation is added as an afterthought. Consequently, these systems are unable to exploit the full potential of approximation primitives.

Opportunity: Sketch-based telemetry has emerged as an attractive solution to address operator woes of costly telemetry. A sketch is a randomized approximation algorithm that can compute accurate statistics on a data stream with low resource cost. Sketches have sparked the community's interest, leading to numerous works on new sketches [10, 13, 15], sketch-based optimizations [11, 12, 16], and sketch-configuration frameworks [6, 14]. Thus, sketches present an opportunity to re-imagine the telemetry stack from the ground-up to provide accurate low-cost telemetry.

2 OUR VISION

To fully unlock the potential of sketches, our vision is to leverage such approximation primitives as first-class citizens in the telemetry stack. We envision Circa, a paradigm-shifting approximation-first telemetry system. Instead of existing solutions that leverage exact telemetry or leverage myopic solutions in isolation, Circa takes a holistic bird's eye view of the telemetry stack and considers approximation primitives like sketches as first-class citizens (Figure 2). Such a holistic approximation-first view allows Circa to unlock cross-stack opportunities for optimization and allow operators to leverage these benefits with low effort.

We describe three high-level requirements in our vision. (1) Circa must be **usable** to allow operators to practically leverage the benefits of sketches, with low effort. This entails providing familiar high-level interfaces to operators (e.g. SQL or PromQL [2]) to specify telemetry intents, and building systems that integrate well with current telemetry solutions such as Prometheus [1] and OpenTelemetry [4]. (2) Circa must be **efficient** in using sketches to provide highly accurate telemetry, while minimizing telemetry costs. This entails deciding which sketches to deploy, where to deploy them, and how to configure them to provide the best cost-accuracy trade-off. (3) Circa must be **adaptive** to changes in the data and intent workloads. This entails dynamically deploying new sketches and re-configuring already deployed sketches with low overhead in response to workload shifts.

We envision four building blocks to realize Circa.

Collection: Instead of collecting all raw data, data can be directly ingested into sketches at the collector. The main challenge here is to deploy sketches with minimum resource usage (e.g. memory or CPU cores) while preserving sufficient fidelity to enable high-accuracy downstream analytics.

Transmission: Instead of transmitting all the collected raw data, sketches can be used to aggregate data, at one collector or across collectors, and transmit them. Similar to collection, the challenge is to transmit sketches with minimum resource usage (such as transmission power or bandwidth) and high information density to enable high-fidelity and accurate downstream analytics.

Analysis: Instead of computing SQL or PromQL [2] queries over raw data, sketches could be used to aggregate data and compute queries over them. While sketches are known to provide low-level statistics over data, the challenge here is to provide high coverage over diverse SQL/PromQL queries using sketch-based primitives.

Storage: Instead of storing all the raw data that was transmitted and analyzed, sketches could be used to aggregate data along certain dimensions and at different time-scales to enable high-fidelity, high-accuracy and efficient retrospective analytics. Similar to analysis, the challenge here is to support diverse historical queries while still providing massive cost reductions from using sketches.

We aim to build Circa in a modular fashion where an operator can choose which building blocks to enable. For instance, the operator may want to collect data exactly but enable approximation while transmitting it and for downstream analysis. In another situation, the operator may want exact data collection and analysis but may want to reduce long-term storage costs using approximation.

3 PRELIMINARY WORK

To demonstrate the benefits of Circa's approximation-first approach, we design a proof-of-concept system for sketch-based collection.

Consider a scenario where a network operator specifies a set of high-level telemetry intents to measure, with different accuracy requirements, using resource-constrained programmable data planes. Here, there are several practical roadblocks in using sketches. While there are several low-level sketch-primitives [10, 13, 15] and optimizations [11, 12] that exist, the operator has no way to translate their high-level intents into sketch deployments in the data plane. Existing sketch-based solutions provide *unusable* low-level interfaces to operators that entail high effort. If the operator naively tries to satisfy high-level intents using sketches, they incur poor cost-accuracy trade-offs.

Our sketch-based collection system leverages the following insight: instead of viewing each telemetry intent in isolation, we take a holistic view of all the intents to be measured by the operator. The input is a set of telemetry intents; each intent specifies a metric to measure (e.g., entropy or heavy hitters) over a flow-key of network traffic (e.g., srcIP or 5-tuple) and its required accuracy. (1) We explicitly map out the capability of different sketches in measuring different intents. For e.g., MRAC [9] can measure entropy, CountSketch [8] can measure heavy hitters, while UnivMon [10] can measure both these metrics. (2) Using offline network traces, we learn error-resource profiles that empirically capture the variation of sketch error for different resource configurations. (3) We formulate an optimization problem that encodes the coverage capability of sketches and error-resource profiles for each sketch, and derives the optimal data and control plane configuration which minimizes the resource usage of sketches in the data plane while satisfying the accuracy requirements of intents. The data plane configuration specifies the sketches to deploy and their resource configurations. The control plane configuration maps each required intent to a sketch deployed in the data plane.

We compare our system with strawman solutions for selecting sketches for each intent and allocating resources to each sketch. For the former, we consider: (a) randomly selecting a sketch for each intent, and (b) selecting a universal sketch (UnivMon [10]) to measure all intents. For the latter, we consider: (a) allocating resources equally to each sketch, and (b) allocating resources to each sketch proportional to the number of intents it measures.

We evaluate our system on diverse intents based on subsets of 5 metrics – entropy, heavy hitters, cardinality, change detection, and flow size distribution. Given a fixed resource bound, our system reduces the error from strawman solutions by **up-to 12x** on a single network node. For network-wide sketch deployment in a Clos topology [7], our system consumes **up-to 60x** less resources than strawman solutions to satisfy intents with the required accuracies.

4 FUTURE DIRECTIONS

We describe two directions towards realizing Circa's vision.

Collection: We are interested in adapting sketch-based collection dynamically changing intent and traffic workloads. For instance, given a DDoS attack, operators would want to estimate additional intents such as to record heavy-hitting flows in the network. Simultaneously, sketch accuracy would drop due to increased traffic flow. Thus, existing sketches must be re-configured with higher resources to maintain high accuracy intent estimates, while a new sketch might have to be deployed at multiple collectors in the network to estimate the new operator intent.

Analysis: Existing solutions for analysis entail ingesting streaming data, such as network events, into monitoring frameworks such as Prometheus [1] or time-series/analytics databases [3, 5]. While these systems are performant for exact telemetry, they offer poor support for sketches. We want to explore how sketches could be deployed in a "drop-in" manner in front of existing analytics/monitoring platforms to transparently accelerate SQL/PromQL queries using sketches.

REFERENCES

[1] Prometheus, 2023. Available from: https://prometheus.io/.
[2] Prometheus Query Language (PromQL), 2023. Available from: https://prometheus.io/docs/prometheus/latest/querying/basics/.
[3] Clickhouse, 2024.
[4] OpenTelemetry: High-quality, ubiquitous, and portable telemetry to enable effective observability, 2024.
[5] Timescale, 2024.
[6] Agarwal, A., Liu, Z., and Seshan, S. Heterosketch: Coordinating network-wide monitoring in hetero-geneous and dynamic networks. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)* (2022), USENIX Association.
[7] Al-Fares, M., Loukissas, A., and Vahdat, A. A scalable, commodity data center network architecture. In *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication* (New York, NY, USA, 2008), Association for Computing Machinery, p. 63–74.
[8] Charikar, M., Chen, K., and Farach-Colton, M. Finding frequent items in data streams. In *International Colloquium on Automata, Languages, and Programming* (2002).
[9] Kumar, A., Sung, M., Xu, J. J., and Wang, J. Data streaming algorithms for efficient and accurate estimation of flow size distribution. In *ACM SIGMETRICS* (2004).
[10] Liu, Z., Manousis, A., Vorsanger, G., Sekar, V., and Braverman, V. One sketch to rule them all: Rethinking network flow monitoring with Univmon. In *ACM SIGCOMM* (2016).
[11] Namkung, H., Liu, Z., Kim, D., Sekar, V., and Steenkiste, P. Sketchlib: Enabling efficient sketch-based monitoring on programmable switches. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)* (2022).
[12] Namkung, H., Liu, Z., Kim, D., Sekar, V., and Steenkiste, P. Sketchovsky: Enabling Ensembles of Sketches on Programmable Switches. In *USENIX NSDI* (2023).
[13] Song, C. H., Kannan, P. G., Low, B. K. H., and Chan, M. C. Fcm-sketch: generic network measurements with data plane support. In *ACM CoNEXT* (2020).
[14] Sun, H., Huang, Q., Sun, J., Wang, W., Li, J., Li, F., Bao, Y., Yao, X., and Zhang, G. Autosketch: Automatic sketch-oriented compiler for query-driven network telemetry. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)* (2024), USENIX Association.
[15] Yang, T., Jiang, J., Liu, P., Huang, Q., Gong, J., Zhou, Y., Miao, R., Li, X., and Uhlig, S. Elastic sketch: Adaptive and fast network-wide measurements. In *ACM SIGCOMM* (2018).
[16] Zhang, Y., Chen, P., and Liu, Z. OctoSketch: Enabling Real-Time, continuous network monitoring over multiple cores. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)* (2024), USENIX Association, pp. 1621–1639.