Stroboscope: Declarative Network Monitoring on a Budget

Olivier Tilmans
Université catholique de Louvain

USENIX NSDI’18
April 11, 2018

Joint work with
T. Bühler (ETH Zürich), I. Poese (BENOCS), S. Vissicchio (UCL) and L. Vanbever (ETH Zürich)

Adapted from original picture © Michael Magg, 2007, CC-BY-SA 3.0
Consider this example ISP network topology

![ISP Network Topology Diagram]

- **Router**
- **Link**
- **Border router**
- **Customer peering**

1.2.3.0/24
What is the ingress router for this packet arriving at router D?
Which paths does the traffic follow?

![Diagram](image_url)
Which paths does the traffic follow?

Tracking flows network-wide requires to match measurements across multiple vantage points.
Which paths does the traffic follow?

Tracking flows network-wide requires to match measurements across multiple vantage points.

NetFlow, ProgME [ToN’11], FlowRadar [NSDI’16]
Is traffic load-balanced as expected?

1.2.3.0/24
Is the latency acceptable?

Packets

Time($t$)

Time($t + 25$ms)
Are there losses?

Time($t$)  Time($t + 25\text{ms}$)

A --- B --- C --- D

W --- Z --- V
Are there losses?

Fine-grained data-plane performance metrics require **packet-level visibility** over individual flows.
Fined-grained network monitoring is widely researched

- Gigascope [SIGMOD’03]
- Planck [SIGCOMM’14]
- Everflow [SIGCOMM’15]
- Compiling Path Queries [NSDI’16]
- Trumpet [SIGCOMM’16]
- Marple [SIGCOMM’17]
Fined-grained **ISP** network monitoring poses unique and unmet challenges

- No control over end hosts

Gigascope [SIGMOD’03]

Planck [SIGCOMM’14]

Everflow [SIGCOMM’15]

Compiling Path Queries [NSDI’16]

Trumpet [SIGCOMM’16]

Marple [SIGCOMM’17]
Fined-grained **ISP** network monitoring poses unique and unmet challenges

- No control over end hosts
- Limited data-plane flexibility

---

**Gigascope** [SIGMOD'03]

**Planck** [SIGCOMM'14]

**Everflow** [SIGCOMM'15]

**Compiling Path Queries** [NSDI'16]

**Trumpet** [SIGCOMM'16]

**Marple** [SIGCOMM'17]
Fined-grained ISP network monitoring poses unique and unmet challenges

- No control over end hosts
- Limited data-plane flexibility
- Limited monitoring bandwidth

Gigascope [SIGMOD'03]
Planck [SIGCOMM'14]
Everflow [SIGCOMM'15]
Compiling Path Queries [NSDI'16]
Trumpet [SIGCOMM'16]
Marple [SIGCOMM'17]
Stroboscope: Declarative Network Monitoring on a Budget

- Collecting traffic slices to monitor networks
- Adhering to a monitoring budget
- Using Stroboscope today
Consider the following flow of packets
Consider the following flow of packets
Stroboscope activates mirroring for the flow

Collector
Packets are copied and encapsulated towards the collector.

Collector

Mirrored packet
The mirroring rule is deactivated after a preset delay
Stroboscope stores the traffic slice for analysis
Stroboscope periodically toggles the mirroring rule
Stroboscope periodically toggles the mirroring rule
Stroboscope periodically toggles the mirroring rule
Stroboscope collects multiples traffic slices over time
Stroboscope works with currently deployed routers

- Most vendors provide traffic mirroring and encapsulation primitives
- The collector activates mirroring for a flow by updating one ACL
- Routers autonomously deactivate mirroring rules using timers
- Traffic slices can be as small as **23 ms** on our routers (Cisco C7018)
Consider the following forwarding path
Stroboscope activates mirroring rules along a path

MIRROR 1.2.3.0/24 ON [A B C D]
Traffic slices are collected

\[ \sim 25 \text{ms} \]

\[ \begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
3 & 4 & 5 & 6 & X & 8 & & \\
3 & 4 & X & 6 & 7 & 8 & 9 & \\
4 & X & 6 & 7 & 8 & 9 & 10 & 11 \\
\end{array} \]
A CONFINE query mirrors any packet leaving a region

MIRROR 1.2.3.0/24 ON [A B C D]
CONFINE 1.2.3.0/24 ON [A B E C D]
A CONFINE query mirrors any packet leaving a region

MIRROR 1.2.3.0/24 ON [A B C D]
CONFINE 1.2.3.0/24 ON [A B E C D]
Counting packets missing in all last hops of a path estimates loss rates
Counting packets partially following the path estimates load-balancing ratios
Analyzing matching packets across traffic slices enables fine-grained measurements at scale
Analyzing matching packets across traffic slices enables fine-grained measurements at scale

Forwarding paths discovery, timestamp reconstruction, payload inspection, …
Stroboscope: Declarative Network Monitoring on a Budget

- Collecting traffic slices to monitor networks
- Adhering to a monitoring budget
- Using Stroboscope today
Stroboscope defines two types of queries

- MIRROR
- CONFINE
Stroboscope defines two types of queries
MIRROR queries reconstruct the path taken by packets

MIRROR 1.2.3.0/24 ON [A B C D]
Fewer mirroring rules reduces bandwidth usage

MIRROR 1.2.3.0/24 ON [A B C D]
Too few mirroring rules creates **ambiguity**

**MIRROR 1.2.3.0/24 ON [A B C D]**
Too few mirroring rules creates ambiguity

The **Key-Points Sampling** algorithm minimizes mirroring rules and guarantees non-ambiguous reconstructed paths.
Stroboscope defines two types of queries
CONFINE queries mirror packets leaving a confinement region

CONFINE 1.2.3.0/24 on [A B E C D]

Edge Mirroring rule
Fewer mirroring rules minimizes control-plane overhead

CONFINE 1.2.3.0/24 ON [A B E C D]
The lower bound is a multi-terminal node cut

\[
\text{CONFINE 1.2.3.0/24 on } [A\ B\ E\ C\ D]
\]
The lower bound is a multi-terminal node cut

The **Surrounding** algorithm minimizes mirroring rules and guarantees to mirror any packet leaving the confinement region.
Query activations must be scheduled to meet the budget
Stroboscope divides the monitoring budget in timeslots

- Timeslot (≈30 ms)
- 40 Mbps
- 150 ms
Stroboscope requires traffic demand estimations
Stroboscope conservatively estimates traffic demands

Prefix measured in prior iterations?
Stroboscope conservatively estimates traffic demands

 Prefix measured in prior iterations? yes → Peak observed demand

Yes
NetFlow records
Assume the query requires the full budget

No
NetFlow estimation
Stroboscope conservatively estimates traffic demands

- Prefix measured in prior iterations?
  - yes: Peak observed demand
  - no: Prefix with enough NetFlow records?

Stroboscope conservatively estimates traffic demands

Prefix measured in prior iterations? yes → Peak observed demand

Prefix with enough NetFlow records? yes → NetFlow estimation

Prefix measured in prior iterations? no → Prefix with enough NetFlow records?
Stroboscope conservatively estimates traffic demands

Prefix measured in prior iterations?
  yes → Peak observed demand
  no → Prefix with enough NetFlow records?
    yes → NetFlow estimation
    no → Assume the query requires the full budget
Consume queries are scheduled in all timeslots

10 Mbps
Q1 Q2 Q3

15 Mbps
Q4 Q5

20 Mbps
Q6

0* Mbps

40 Mbps

Confine query
Stroboscope first approximates a minimal sub-schedule.
Stroboscope first approximates a minimal sub-schedule, optionally optimizing for the optimal bin-packing solution.
Stroboscope replicates the sub-schedule
Stroboscope replicates the sub-schedule, and minimizes budget leftovers
Stroboscopy achieves deterministic sampling
Stroboscope: Declarative Network Monitoring on a Budget

- Collecting traffic slices to monitor networks
- Adhering to a monitoring budget
- Using Stroboscope today
Selecting mirroring locations in realistic ISP topologies is fast
Schedules can be quickly approximated
Stroboscope tracks the rate of mirrored traffic in real time.
Measurement campaigns are stopped early if the estimated demand are exceeded.
Exceeding the total budget schedules the query once per measurement campaign.
Stable recorded traffic rates are used for future estimations.

Source A → Collector → Destination B

Traffic rate [Kb/s]
- real
- mirrored
- budget

Time [s] 0 1 10 12 17 20
Traffic rate [Kb/s] 0 1000 2000 3000 5000

MIRROR ON [A B]
Stroboscope exceeds the monitoring budget for at most one timeslot
Stroboscope: Declarative Network Monitoring on a Budget

- Traffic slicing as a first-class data-plane primitive
- Strong guarantees on budget compliance and measurement accuracy
- Measurement analysis decoupled from measurement collection
Stroboscope
Declarative Network Monitoring on a Budget

Stroboscope enables fine-grained monitoring of any traffic flow by instructing routers to mirror millisecond-long traffic slices in a programmatic way. Stroboscope takes as input high-level monitoring queries together with a budget and automatically determines which flows to mirror, where to place the mirroring rules and when to schedule these rules to maximize coverage while meeting the input budget.

Overview
Backup slides
NetFlow brings a poor visibility over traffic in ISP networks
Stroboscope defines a declarative requirement language

MIRROR 1.2.3.0/24 ON [A B C D], [A E C D]
MIRROR 1.2.3.0/24 ON [A -> D]

CONFINE 1.2.3.0/24 ON [A B E C D]
CONFINE 1.2.3.0/24 [A -> D]

MIRROR 1.2.3.0/24 [ -> D]
CONFINE 1.2.3.0/24 [ -> D]

USING 15 Mbps DURING 500 ms EVERY 5 s
The placement algorithms minimize the mirroring rules.
Schedules can be computed by two pipelines

![Graph showing time in seconds vs. input query count](image)

- **Optimized schedule**
- **Approximation**

![Graph showing optimization gain vs. fraction of experiments](image)