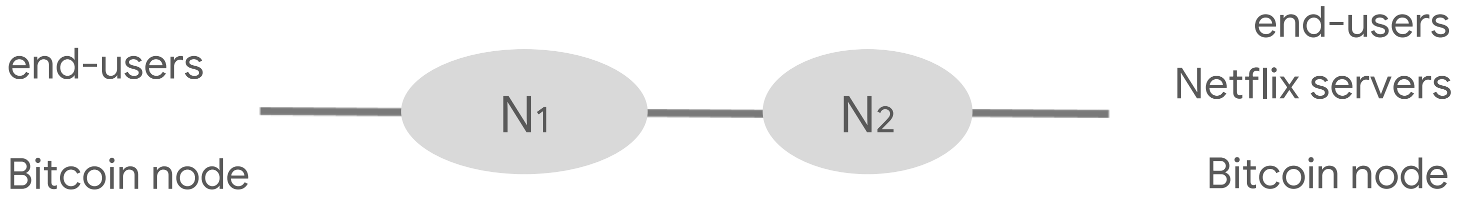
Internet Performance Transparency

Georgia Fragkouli

Private Thesis Defense
24/06/2022

EPFL

Internet's goal: enable end-systems to communicate *w/ good performance*



- Users need to trace performance attacks [Apostolaki et al. 2017]
- Networks need to prove competitive performance
- Regulators need to verify SLAs and neutrality rules

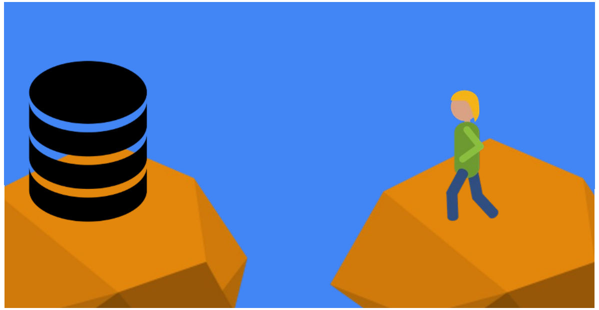
Users & regulators need to localize performance issues to networks

Why is localizing performance issues hard?

Networks

Generate performance measurements

Exaggerate network performance



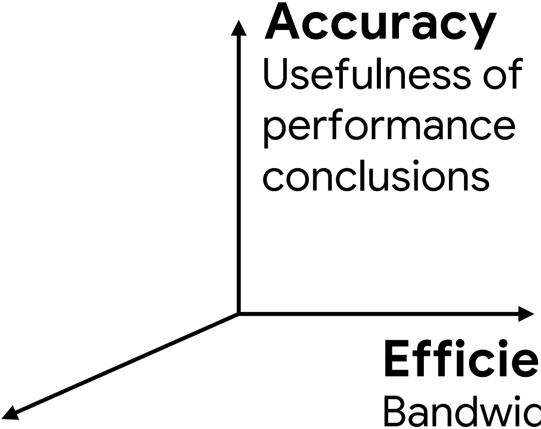
Users

Unreliable access to measurements

Reliably assess network performance

Bridge gap to enable network performance transparency

Transparency goals



Accuracy
Usefulness of performance conclusions

Anonymity
Hiding users' communication patterns

Efficiency
Bandwidth for publishing reports

No existing design with good balance

Existing designs rely on fine-grained reporting

- Networks report on individual packets
- Networks sample packet reports
- Networks accurately report fate of individual packets
  - Requires incentives for honestly reporting fate of individual packets
  - Reveals users' communication patterns

Inaccuracy because of unrealistic incentives & lower anonymity

Thesis

Accurate and efficient Internet performance transparency is possible
by adapting the incentive structure to the underlying honesty incentives
and combining incentives with mathematical tools;
adapting the report granularity eases the transparency-anonymity tussle.

Outline

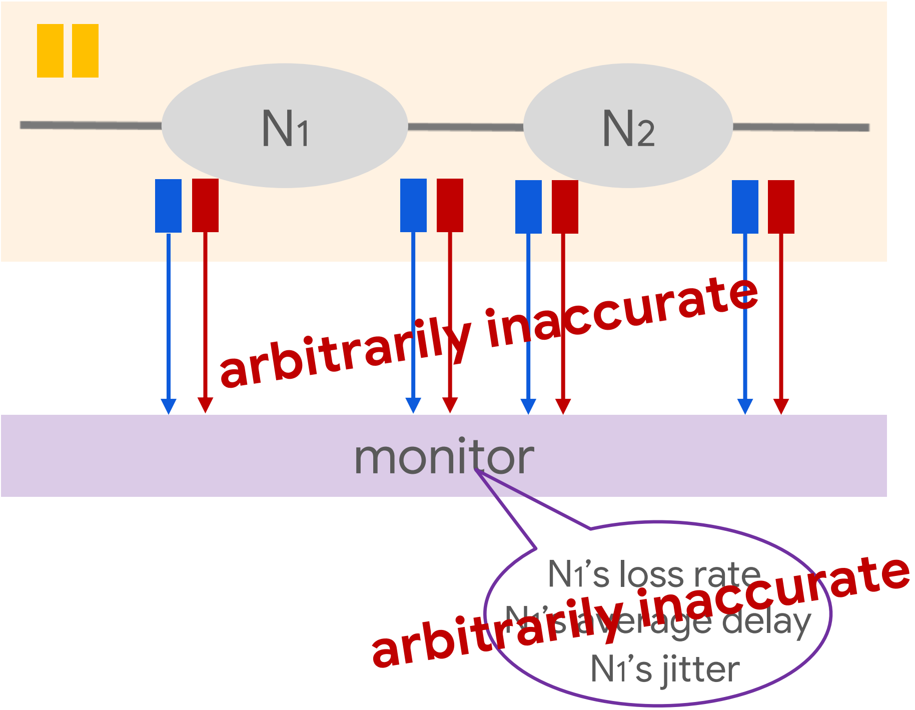
- **Accurate & efficient Internet performance transparency**
    - Split-responsibility for verifiable, user-based average metrics
    - Policy-based grouping of traffic for verifiable jitter

- Reconcile transparency with anonymity
    - Time granularity as noise
    - Adaptive reports for anonymity

Transparency protocols

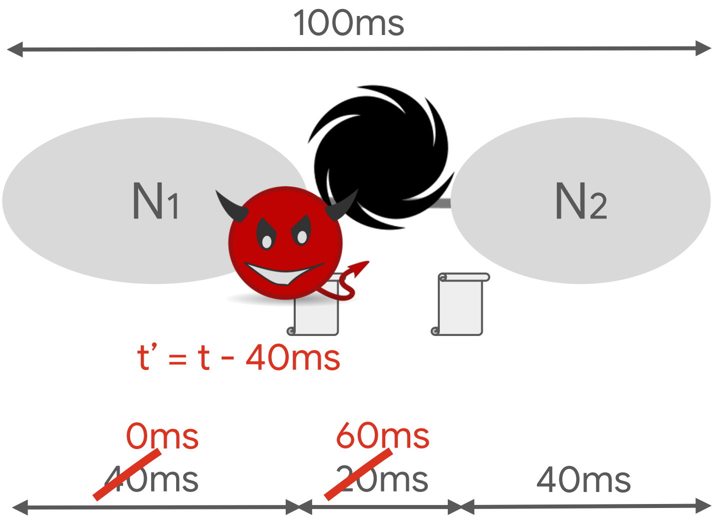
- **Data plane:** sampling packets
  - + consistent => same samples
  - + secure => representative samples

- **Control plane:** per-network performance estimation
  - loss rates & delay averages
  - jitter & neutrality



Need: accurate network statistics despite inaccurate packet reports

Packet delay

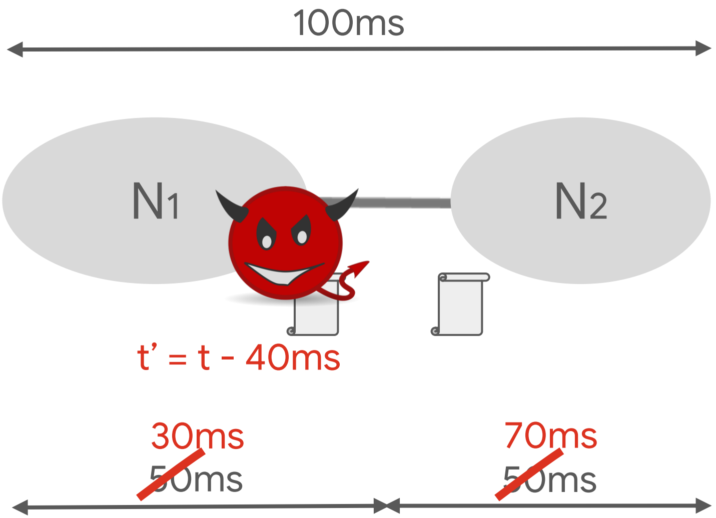


externalizability

someone has to take responsibility for orphan delay

Externalizability not enough for accuracy

Creating incentives for honesty through conflict

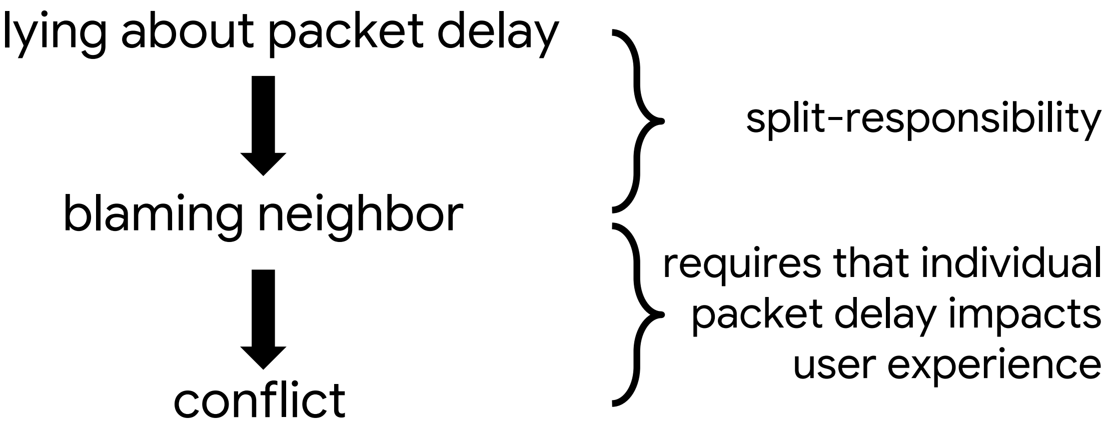


split-responsibility

lying about pkt delay
=> blaming neighbor
=> conflict

Networks have an incentive to honestly report packet delay

The impact of lying about individual packets



lying about packet delay

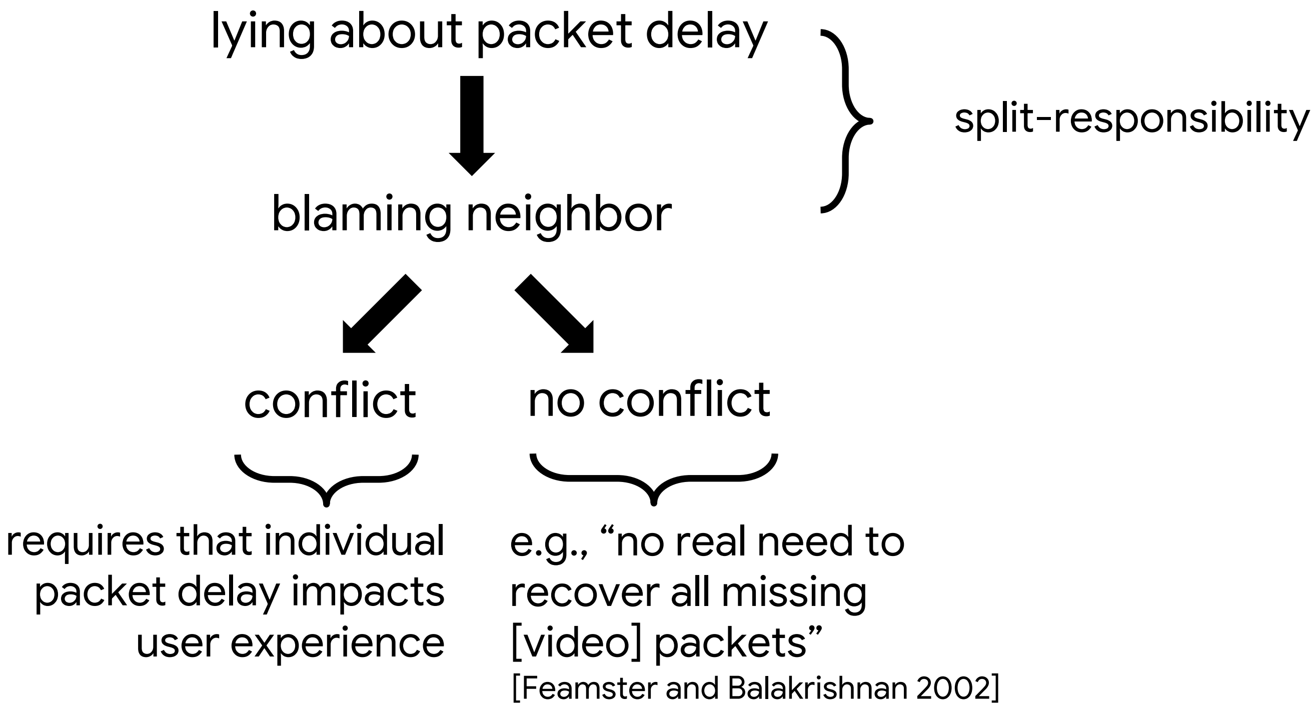
blaming neighbor

conflict

split-responsibility

requires that individual
packet delay impacts
user experience

The impact of lying about individual packets



lying about packet delay

blaming neighbor

split-responsibility

conflict

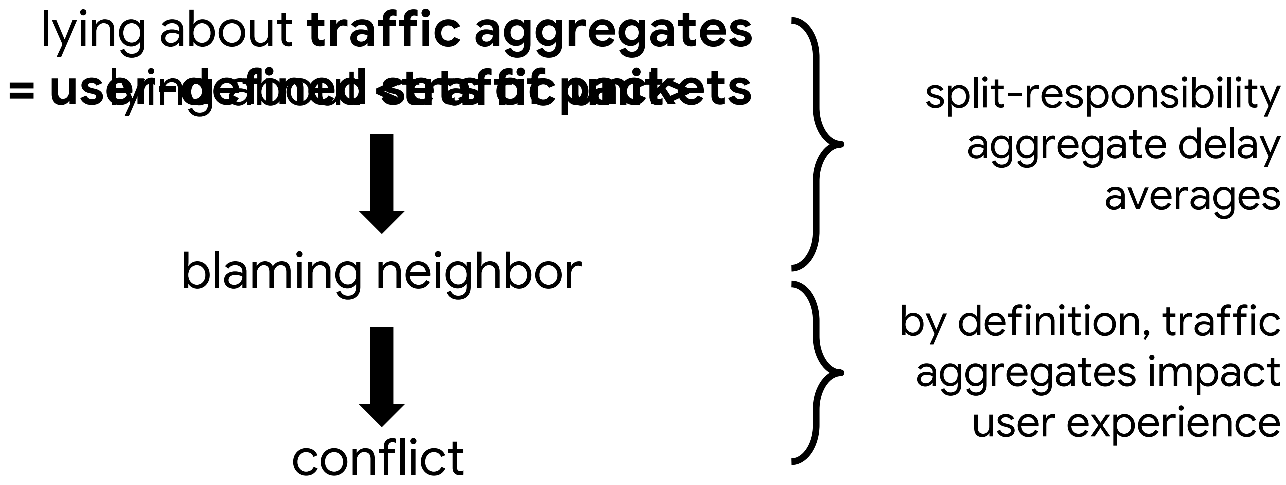
no conflict

requires that individual packet delay impacts user experience

e.g., “no real need to recover all missing [video] packets” [Feamster and Balakrishnan 2002]

Lying does not always lead to conflict => inaccurate packet delays

Accurate metrics from inaccurate packet delays



lying about **traffic aggregates**

= user-defined sets of packets

lying about **<traffic unit>**

blaming neighbor

conflict

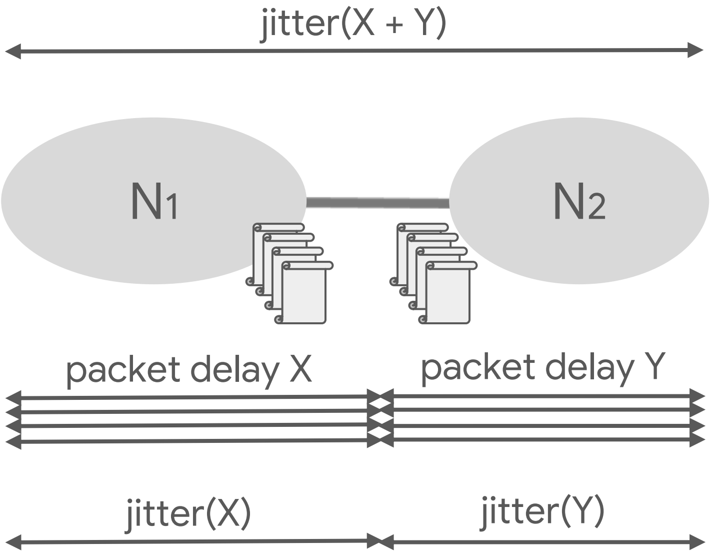
split-responsibility aggregate delay averages

by definition, traffic aggregates impact user experience

Accurate delay averages by adapting to user interests

Jitter

- Conflicts on jitter?
  - BUT jitter not externalizable:
    jitter(X + Y) = jitter(X) + jitter(Y) **+ 2cov(X,Y)**



Jitter not externalizable => conflicts not enough

Accuracy for jitter: a unifying perspective

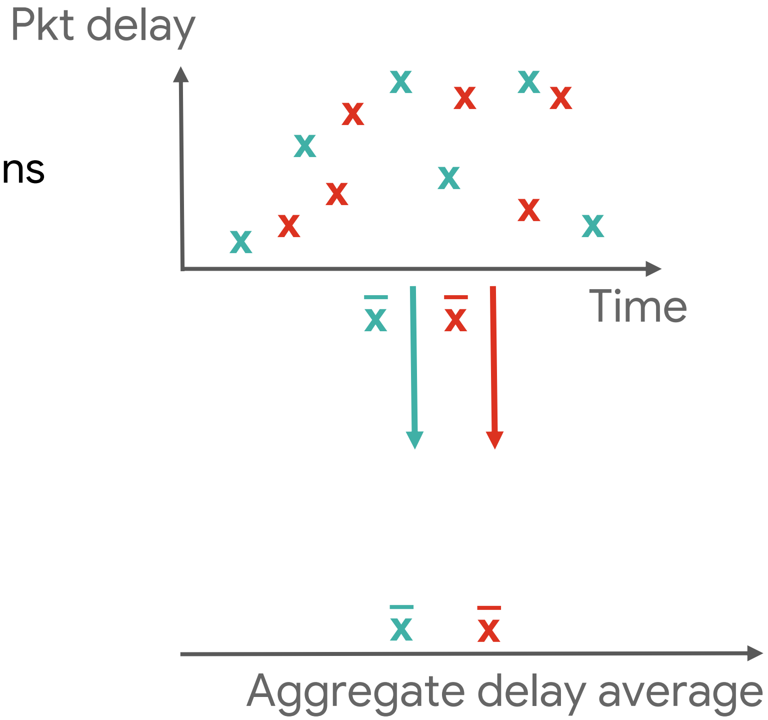
- **Similarly treated traffic** subject to **math constraints**

- Jitter reliably extracted from delay averages & math constraints

Neutrality

- Defining neutrality
  - exposing packets to same network conditions
  - ⇒ same packet delay distribution

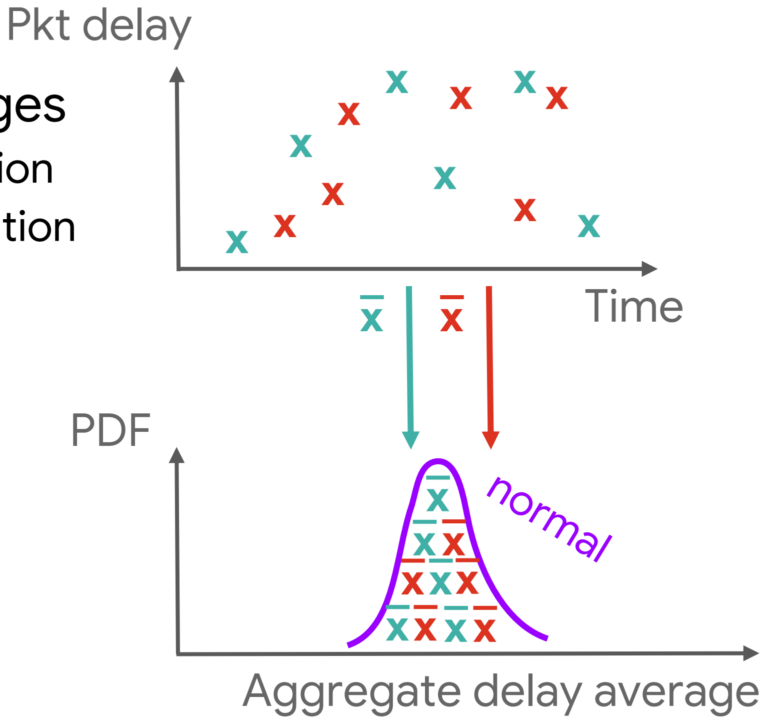
- Measuring neutrality
  - “draw” distributions & check if similar
  - BUT cannot directly see distributions



Gap between metric of interest and incentivizable info

Neutrality imposes constraints

- CLT ties together aggregate delay averages
  - each average follows same normal distribution
  - take many averages to draw normal distribution

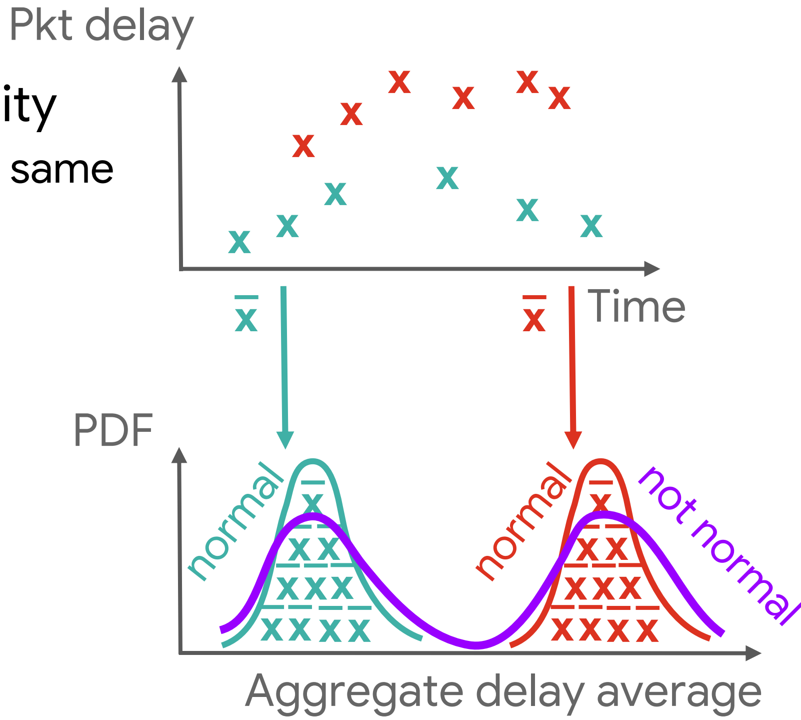


Reliably extract neutrality via normality check over delay averages

Impose neutrality on networks?

- No universal but **per traffic class** neutrality
  - traffic class = subset of packets treated the same
  - networks free to declare traffic classes
  - monitor checks normality within each class

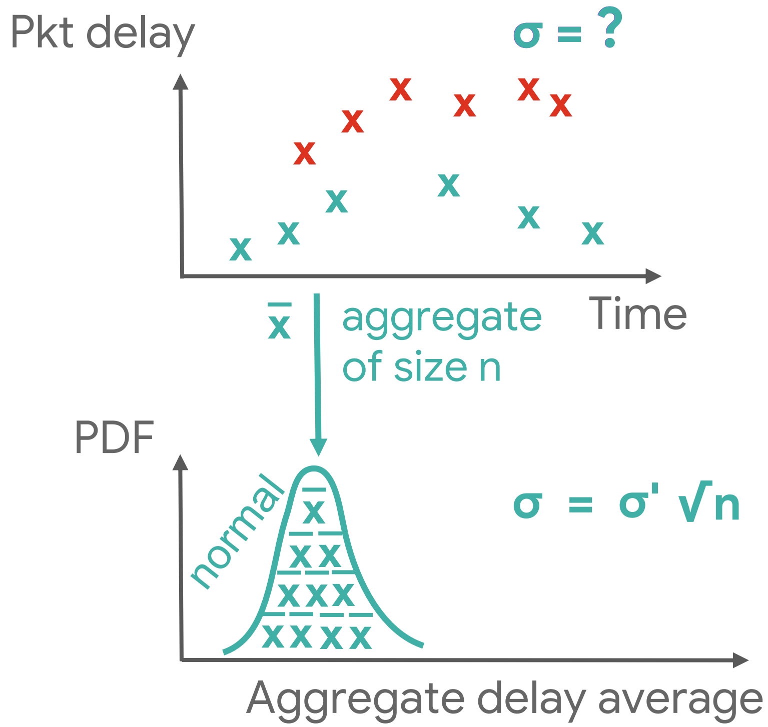
- Dishonest class declaration?
  - ⇒ networks risk failing normality checks
  - ⇒ incentive for honest class declaration



No universal neutrality but transparent class declaration

Jitter

- Estimate jitter one-class-at-a-time
  - allows using CLT
  - ⇒ jitter = known function of known quantities



Reliably extracted from delay averages

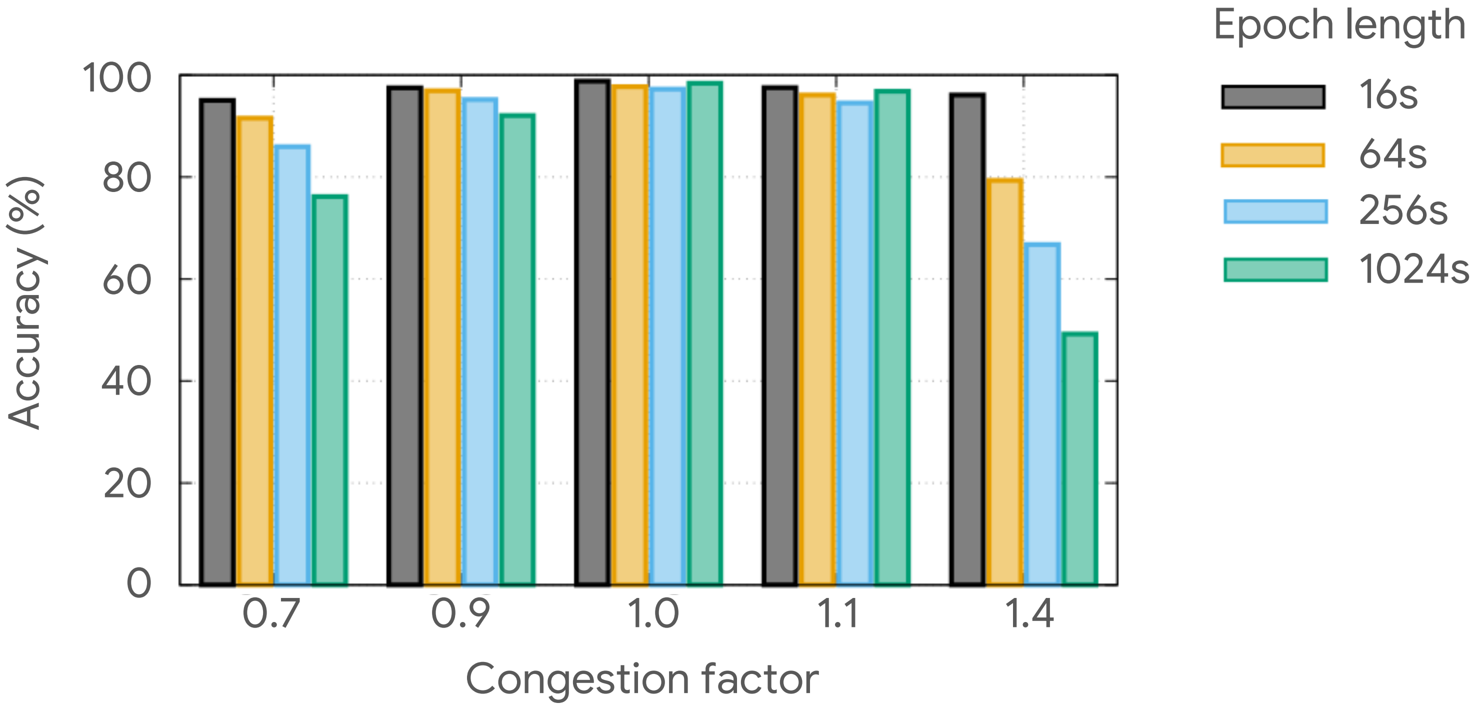
Recap

- Accurate averages via split-responsibility & alignment with user interests
- Class verification via normality check on accurate averages
- Accurate jitter via per-class verification

Experimental setup & methodology

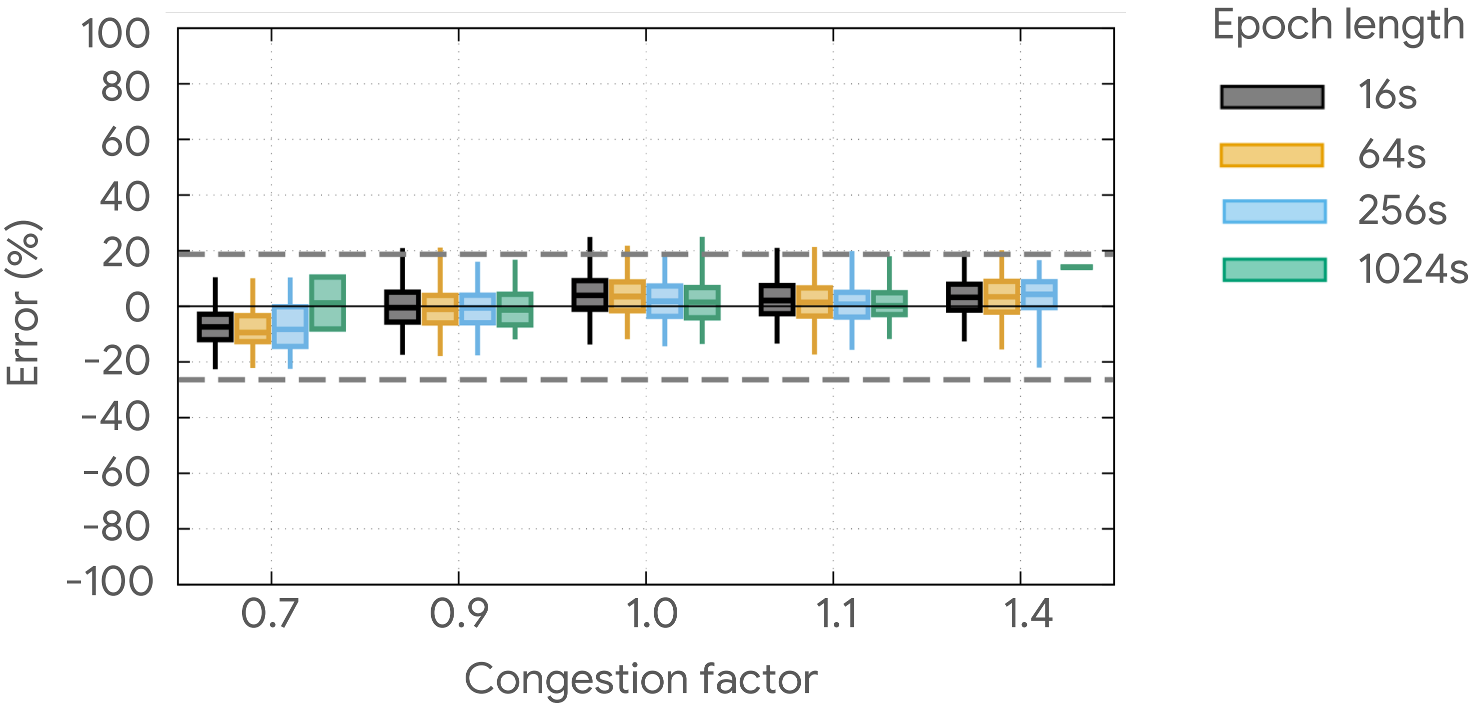
- Input traffic: 21 one-hour CAIDA traces
- Under different network conditions
- 5% sampling => <1% bandwidth overhead
- Non-overlapping, equi-length epochs
- Aggregation: /24 src-dst prefix pairs
- Metrics: per-epoch delay averages, neutrality & jitter

Neutrality verification accuracy



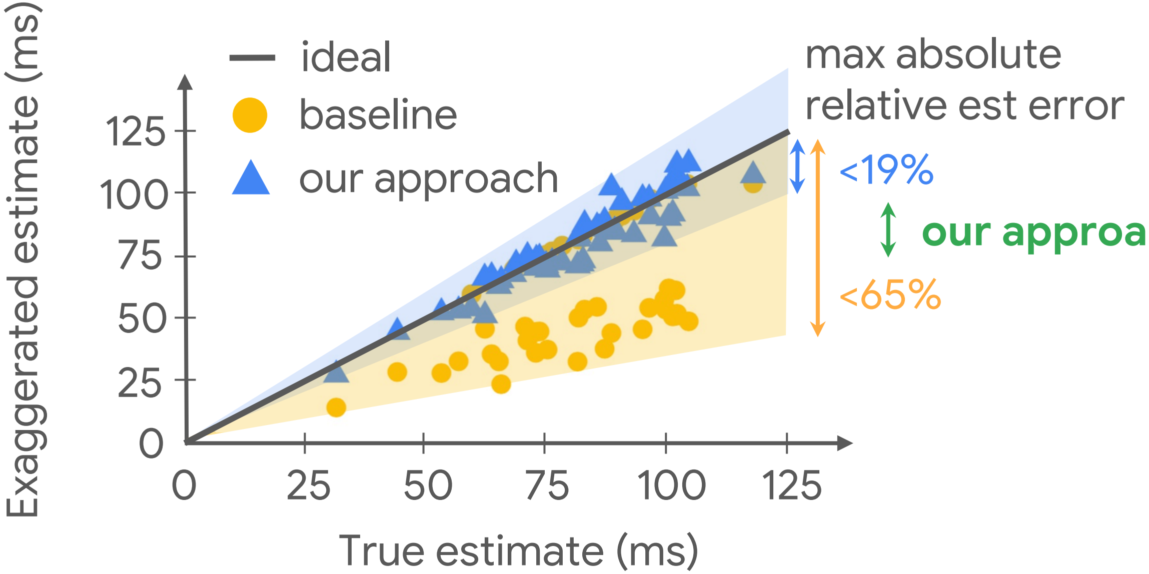
>78% accuracy across diverse scenarios

Jitter estimation accuracy

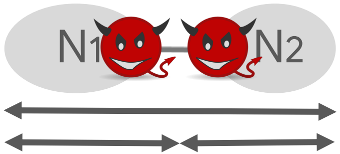


75th percentile <10% & 99th percentile <25%

Accuracy gains under dishonesty



our approach: 3x better



unaffected delay averages

3x better accuracy by relying on incentivizable information

Transparency

SLAs & neutrality

Anonymity

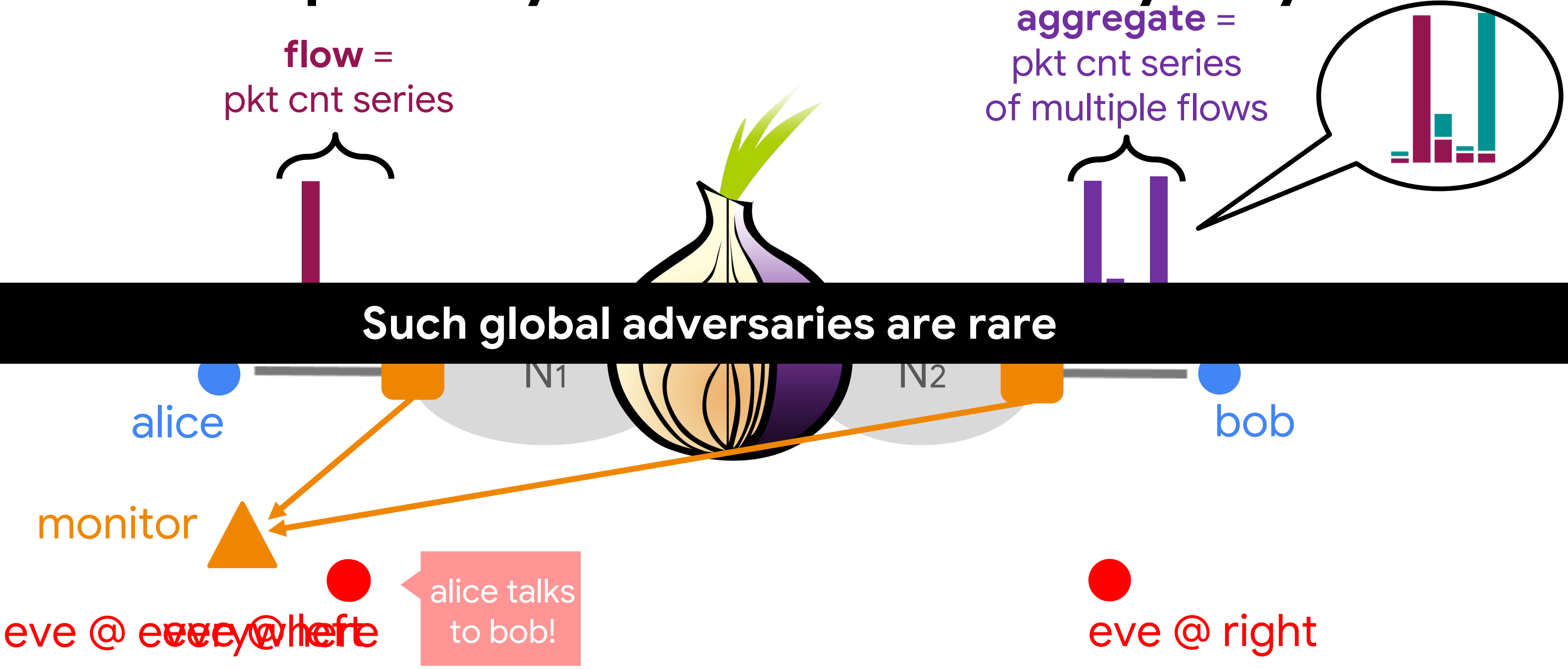
Tor-like overlays

Outline

- Accurate & efficient Internet performance transparency
  - Split-responsibility for verifiable, user-based average metrics
  - Policy-based grouping of traffic for verifiable jitter

- **Reconcile transparency with anonymity**
  - Time granularity as noise
  - Adaptive reports for anonymity

Transparency weakens Tor anonymity



flow =
pkt cnt series

aggregate =
pkt cnt series
of multiple flows

N1

N2

alice

bob

monitor

eve @ everywhere

eve @ left

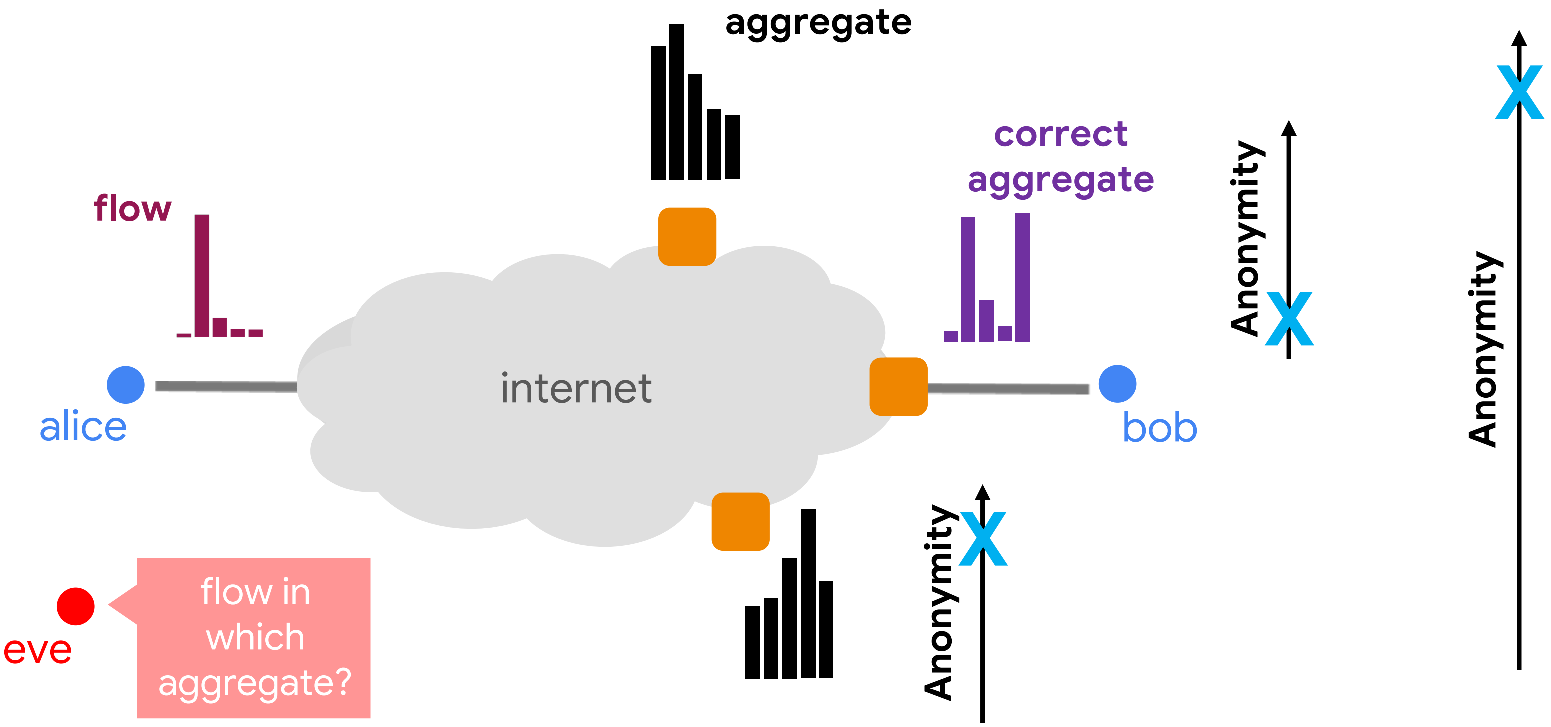
alice talks
to bob!

eve @ right

Such global adversaries are rare

Transparency introduces global adversary

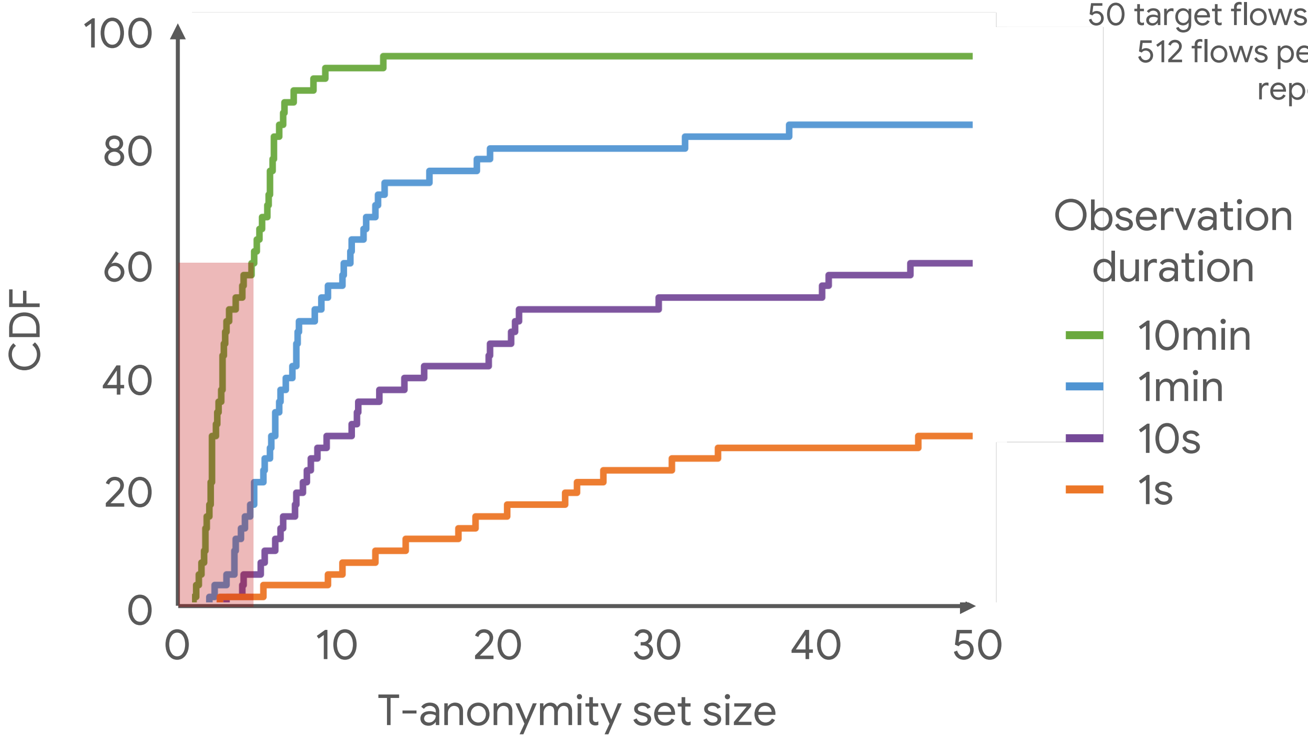
Quantifying anonymity



T-anonymity set size captures deviation from ground truth

Effect of transparency on anonymity

2018 CAIDA traces
50 target flows/aggregates
512 flows per aggregate
reports per 1ms



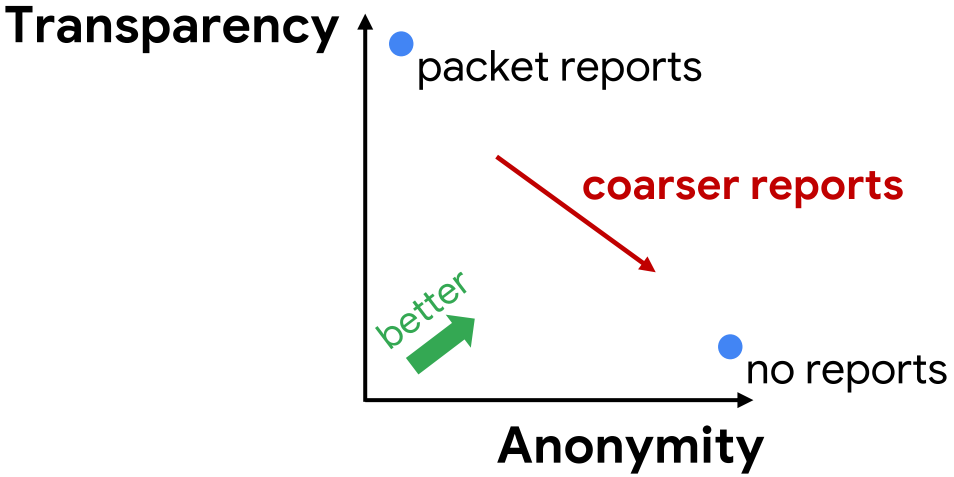
Given enough time, adversary deanonymizes ~60% of cases

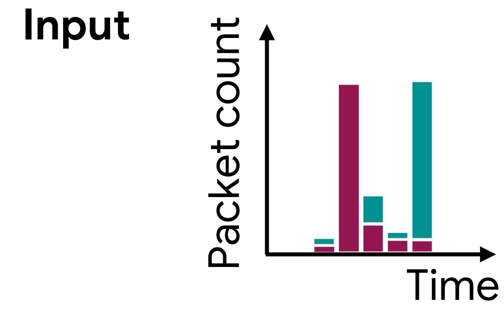
Constraints

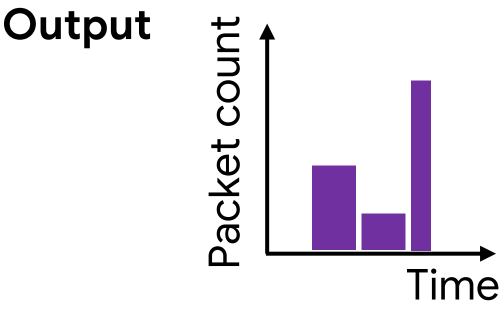
- Any flow could be a target
- No network coordination

Improve anonymity for all flows with network-local decisions

Time granularity as noise

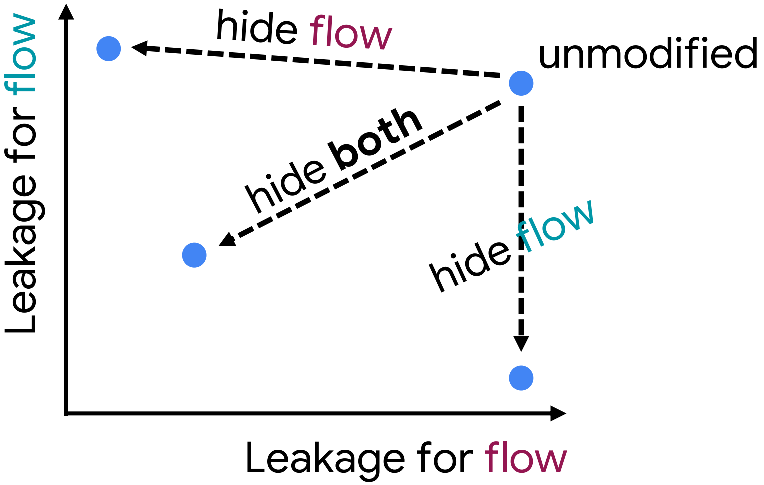


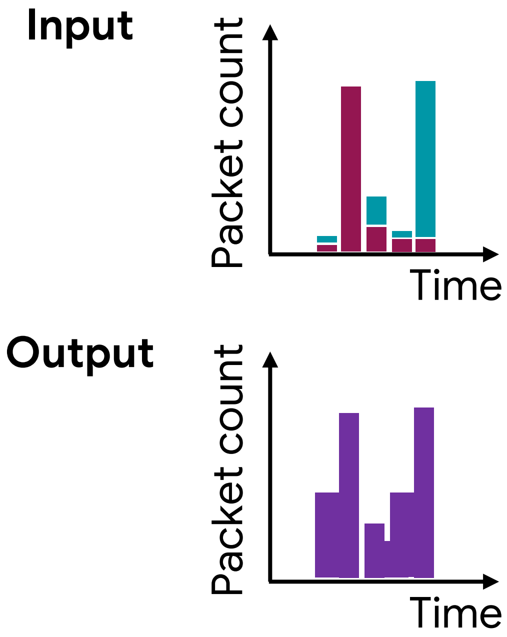




Hides sensitive flow patterns but impacts report utility

Networks adaptively time-bin reports

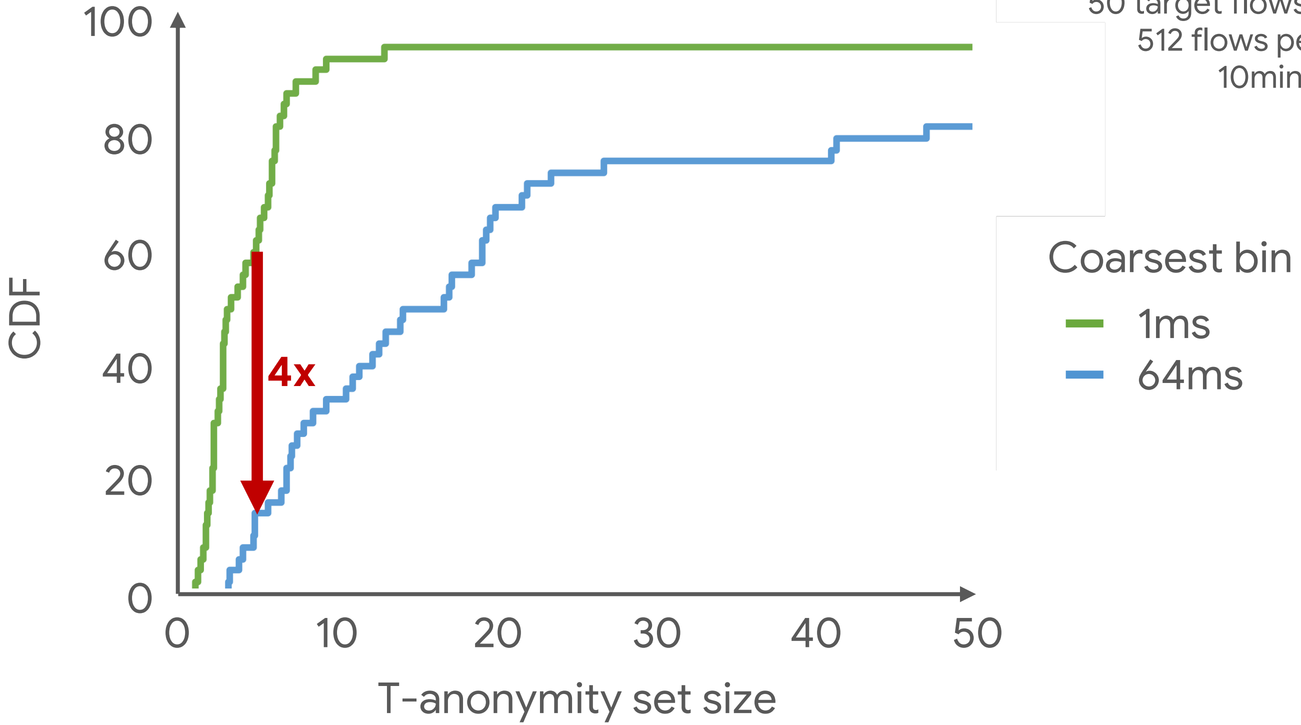




Pick the binning that minimizes leakage for most-leaking flow

Effect of coarser reports on anonymity

2018 CAIDA traces
50 target flows/aggregates
512 flows per aggregate
10min observation



4x improvement at sub-second granularity

Accurate, efficient & anonymous transparency

- Accurate & efficient Internet performance transparency
  - Split-responsibility for verifiable, user-based average metrics
  - Policy-based grouping of traffic for verifiable jitter

- Reconcile transparency with anonymity
  - Time granularity as noise
  - Adaptive reports for anonymity