Privacy-preserving Network Congestion Control (1739966)

Hussein Darir Hussein Sibai Chester Cheng Sayan Mitra Geir Dullerud Nikita Borisov
University of Illinois at Urbana-Champaign

Project goals

A. *Develop algorithms and analysis tools for building congestion-aware traffic routing algorithms with provable privacy guarantees;*

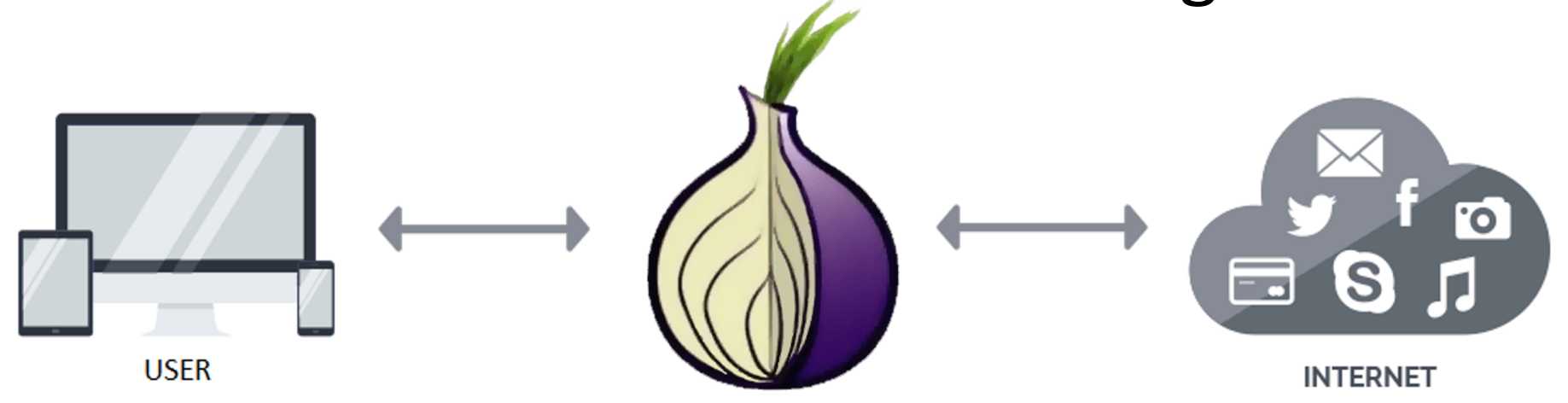
B. *Develop the foundations, algorithms, and experimental systems for studying the trade-off between privacy and efficiency in different networks; and*

C. *Of particular interest are communication networks and other networks used for collection and dissemination of behavioral information.*

The Tor network

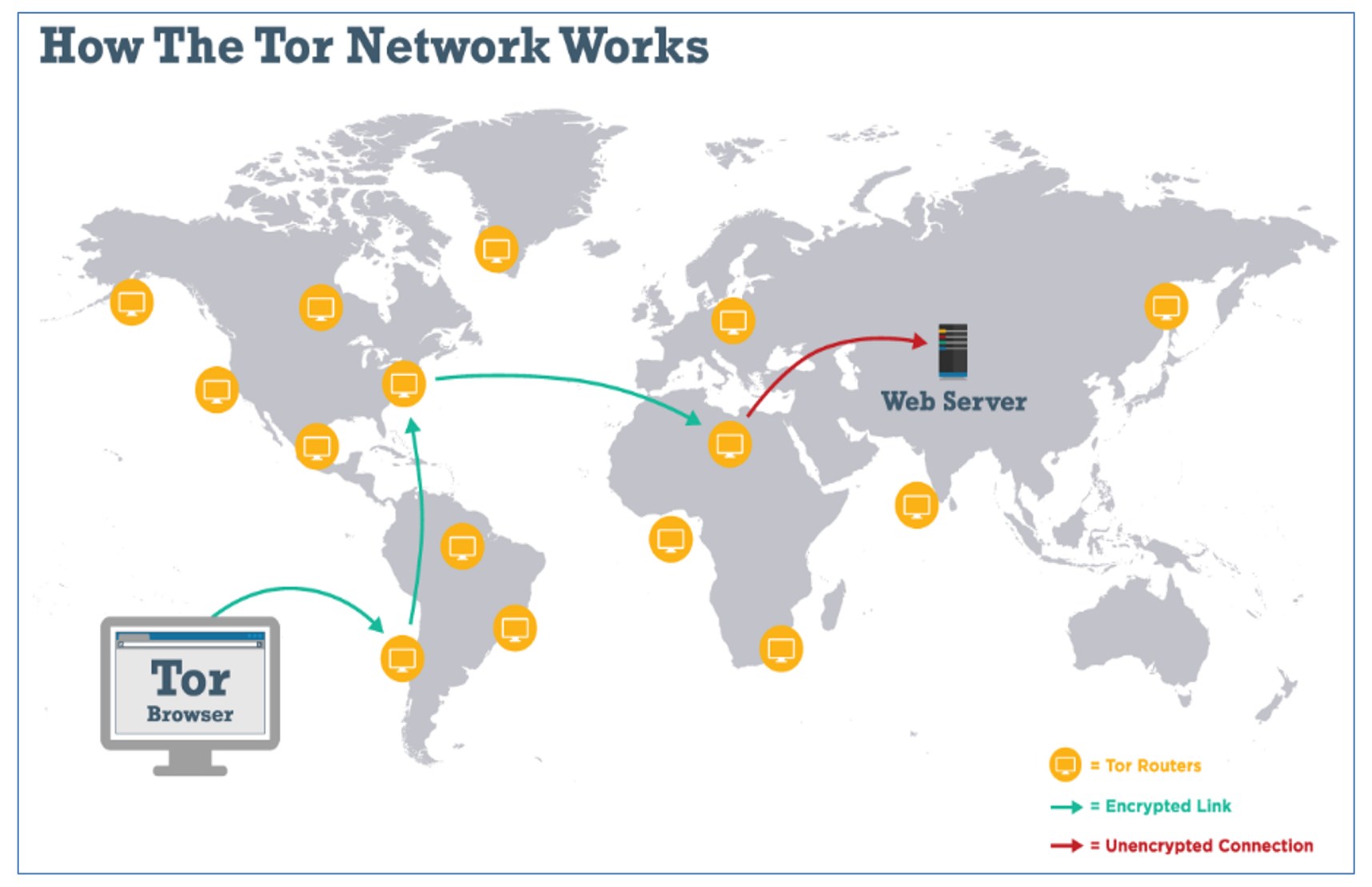
Tor | Browser

- Our first step has been to study the problem of load-balancing in path selection in anonymous networks such as **Tor**.
- Users are increasingly turning to anonymous communication networks to protect themselves from surveillance and online tracking.



What is Tor?

- "Tor is free software and an open network that helps the user defend against traffic analysis, a form of network surveillance that threatens personal freedom and privacy."[1]

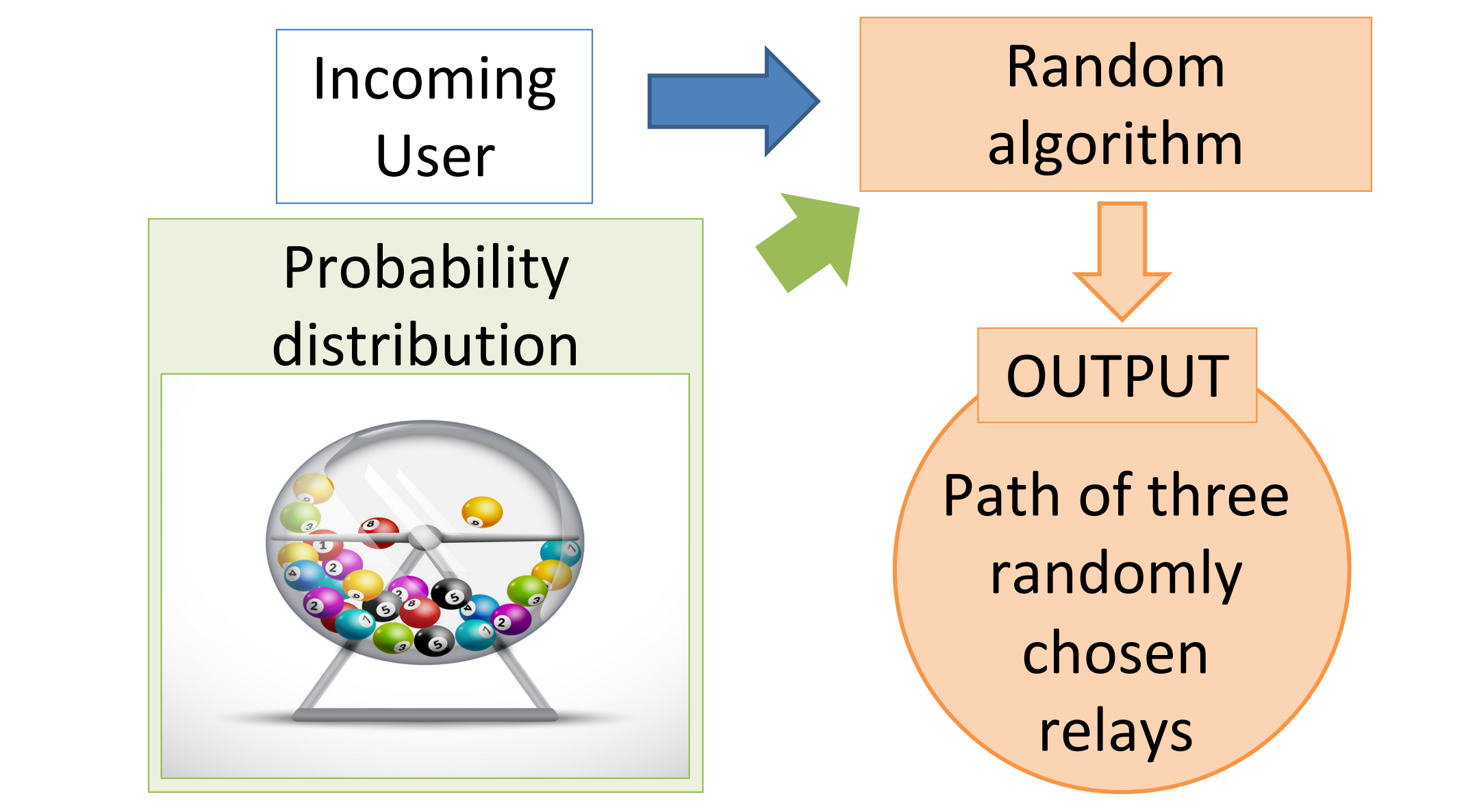


- To achieve anonymity in Tor, users' traffic is routed across a series of servers, called relays.
- Each user's path through the network, called a circuit, typically transits three of them.

Tor can be SLOW!

SLOW Tor SLOW

- Users choosing the paths imperfectly is a main reason.
- Currently relays are chosen randomly weighted by their estimated capacities.
- Current method of estimating capacity of relays is not accurate.



Incoming User

Random algorithm

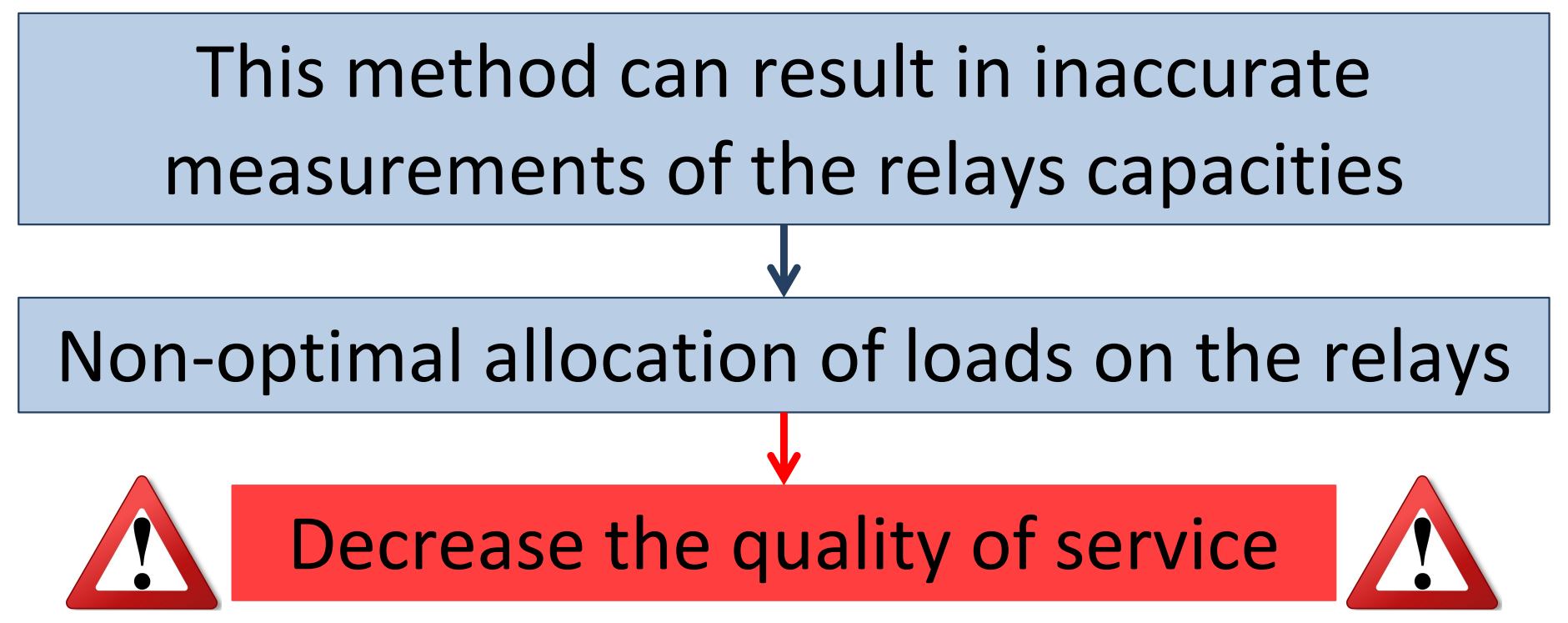
Probability distribution

OUTPUT

Path of three randomly chosen relays

Relays capacities estimation

- Currently, a server periodically creates test paths that pass through all relays in the network and measures their allocated bandwidths.
- These bandwidths are then assumed to be the capacities of the corresponding relays that are released to the public.

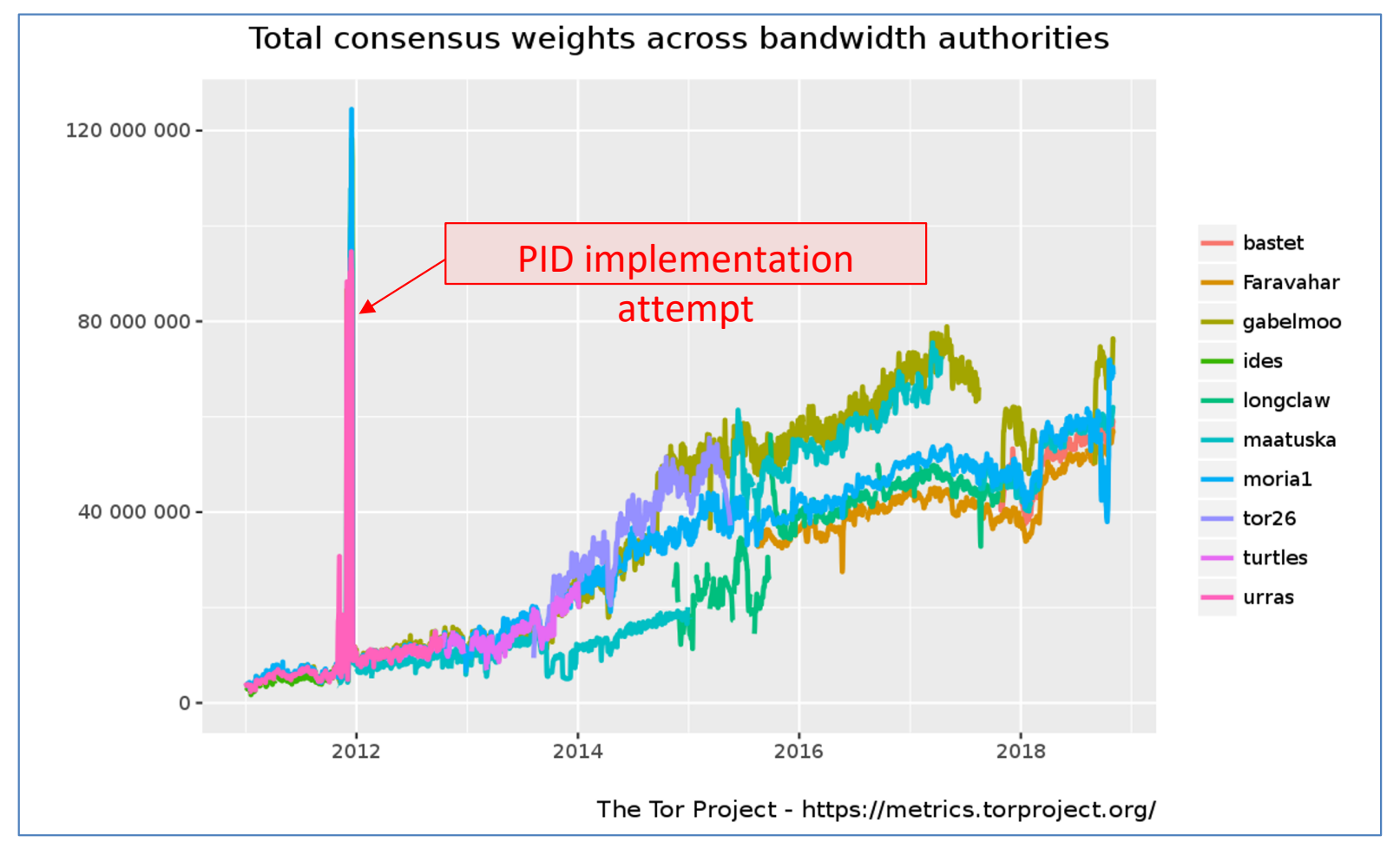


This method can result in inaccurate measurements of the relays capacities

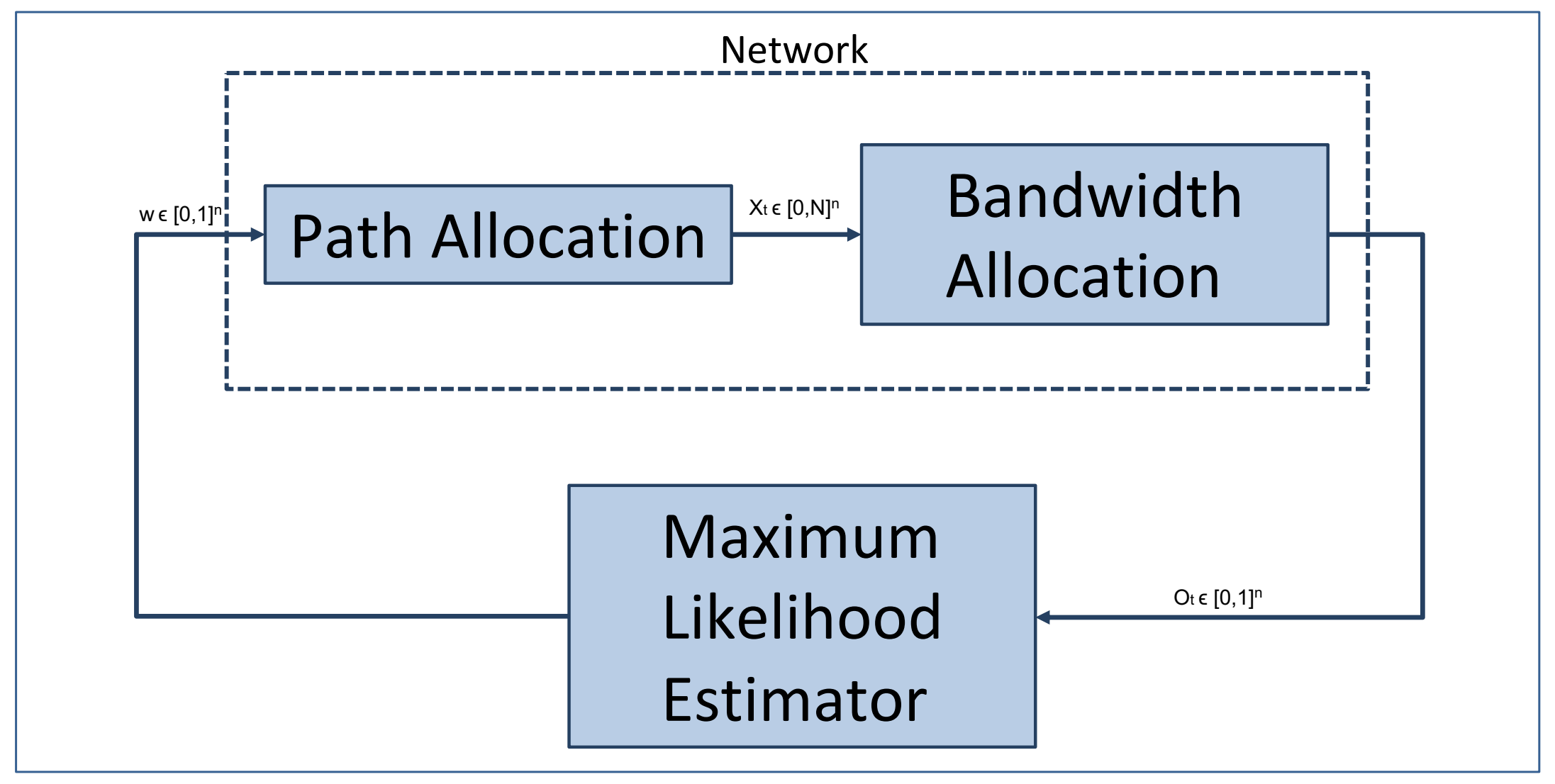
Non-optimal allocation of loads on the relays

Decrease the quality of service

- There were failed attempts to solve the problem using PID controller.

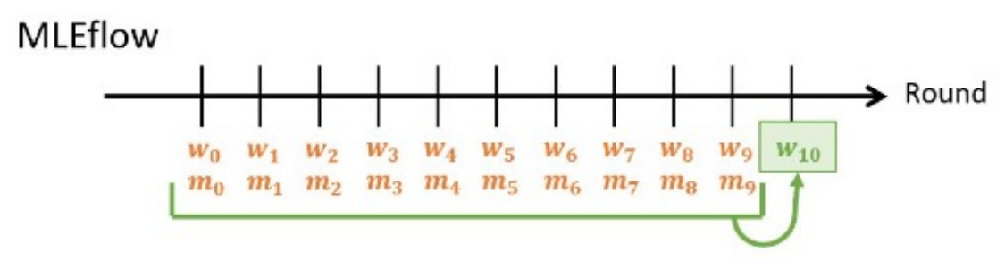


The Tor Project - https://metrics.torproject.org/



MLEFlow

- We developed an algorithm, **"MLEFlow"**, that result in provably accurate estimates of the relays capacities using maximum likelihood analysis.


MLEflow
Round
w_0 w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8 w_9 w_{10}
m_0 m_1 m_2 m_3 m_4 m_5 m_6 m_7 m_8 m_9

- Given the whole history of measurements, the current capacities estimates and using maximum likelihood analysis, we derived a closed form solution for the optimal update of the estimates.

$$C^H_{t+1}[j] \approx \exp\left(\frac{\sum_{i=0}^{t} \frac{1}{m_i[j]} \log(m_i[j]\lambda_s w_i[j])}{\sum_{i=0}^{t} \frac{1}{m_i[j]}}\right)$$

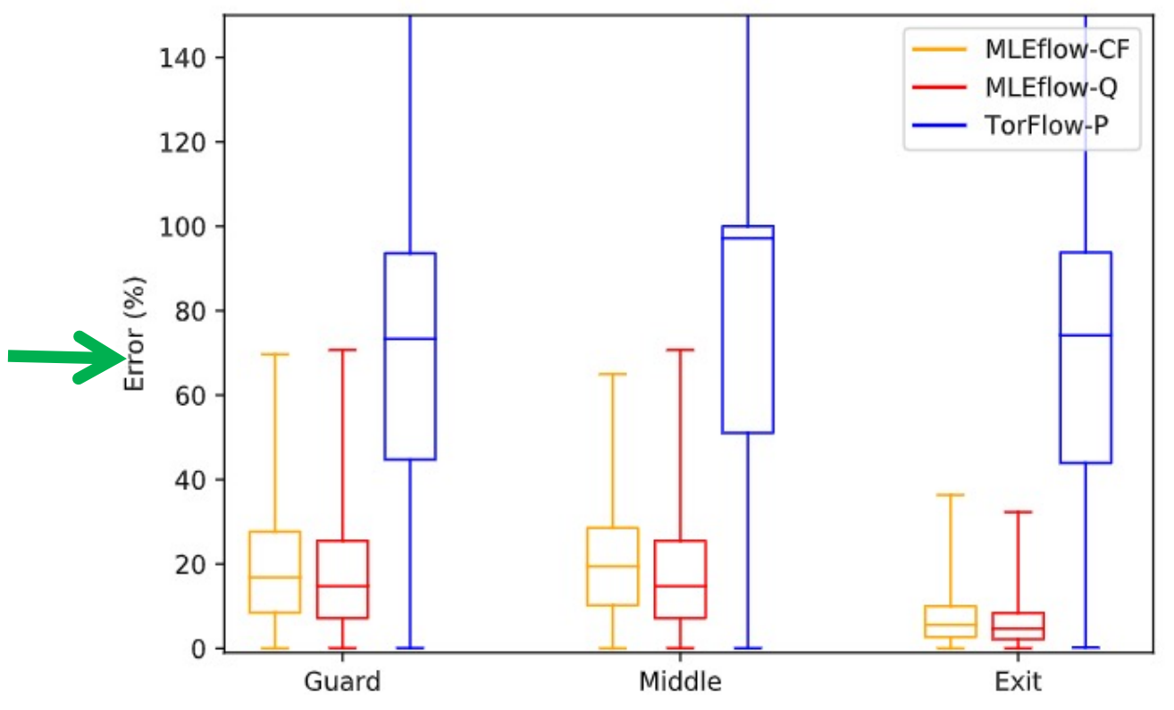
✓ As the rate of users' arrival to the network, λ_s, increases, the **expected value of our estimates converges to the actual capacities**.

✓ **The variance of MLEFlow estimates goes to zero** as the number of iterations, t, increases.

Python simulation results

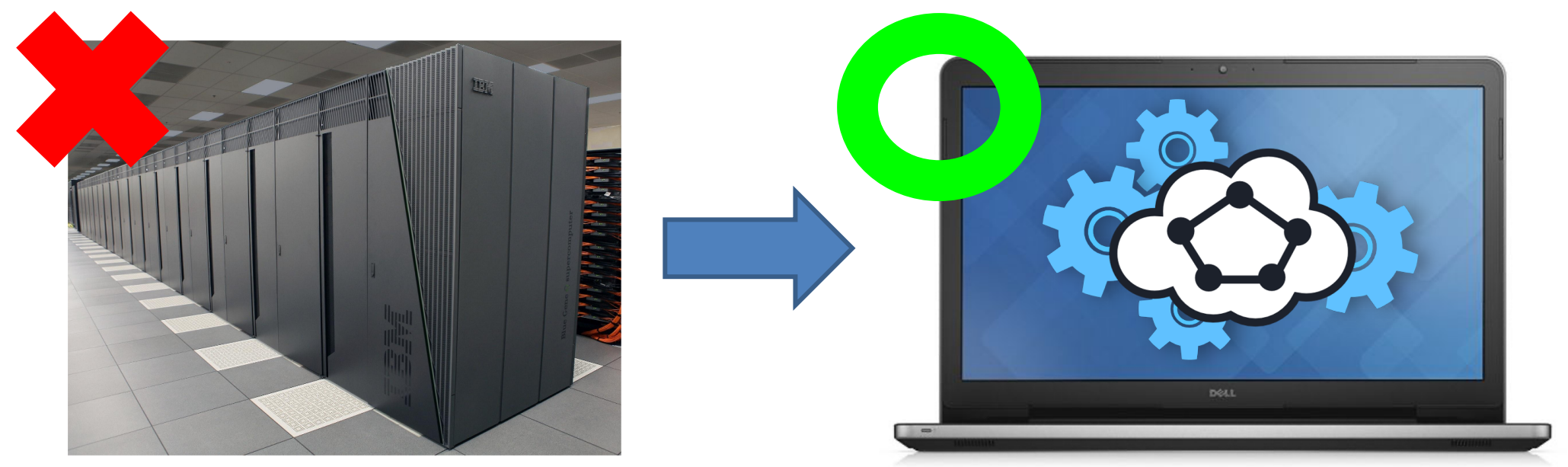
- We evaluated MLEFlow using a flow-based simulation of the Tor network.
- We consider a network analogous to the current Tor network with 6037 relays (100% network).

✓ Significantly **lower estimation error** than the currently deployed algorithm Torflow.



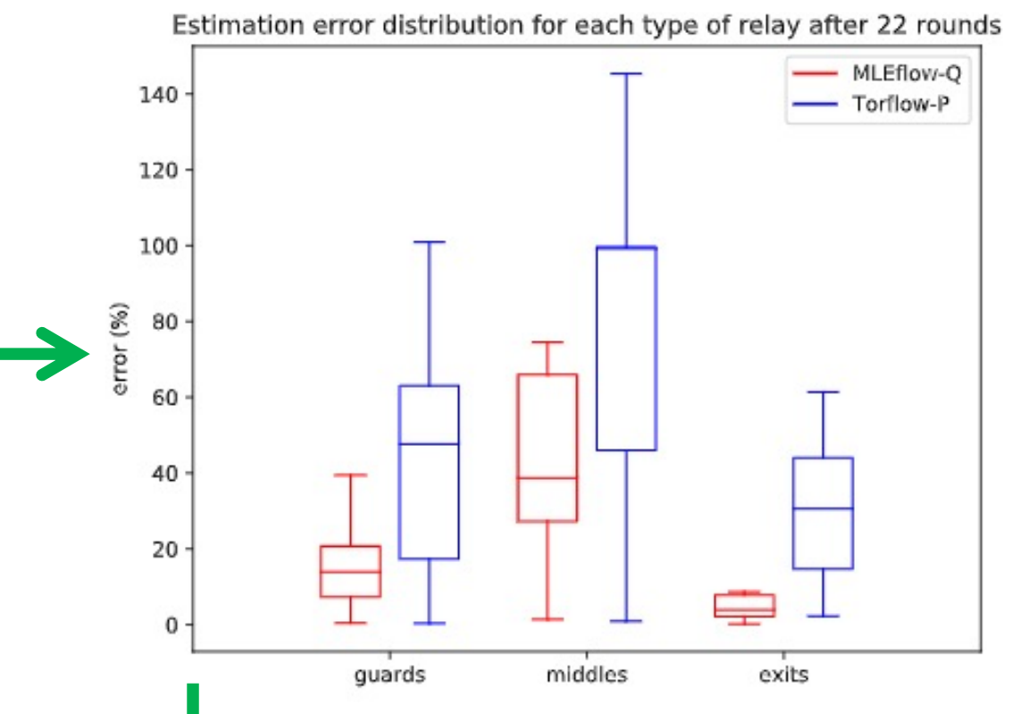
MLEFlow is significantly more accurate because of using past measurements in estimation.

Shadow simulation results

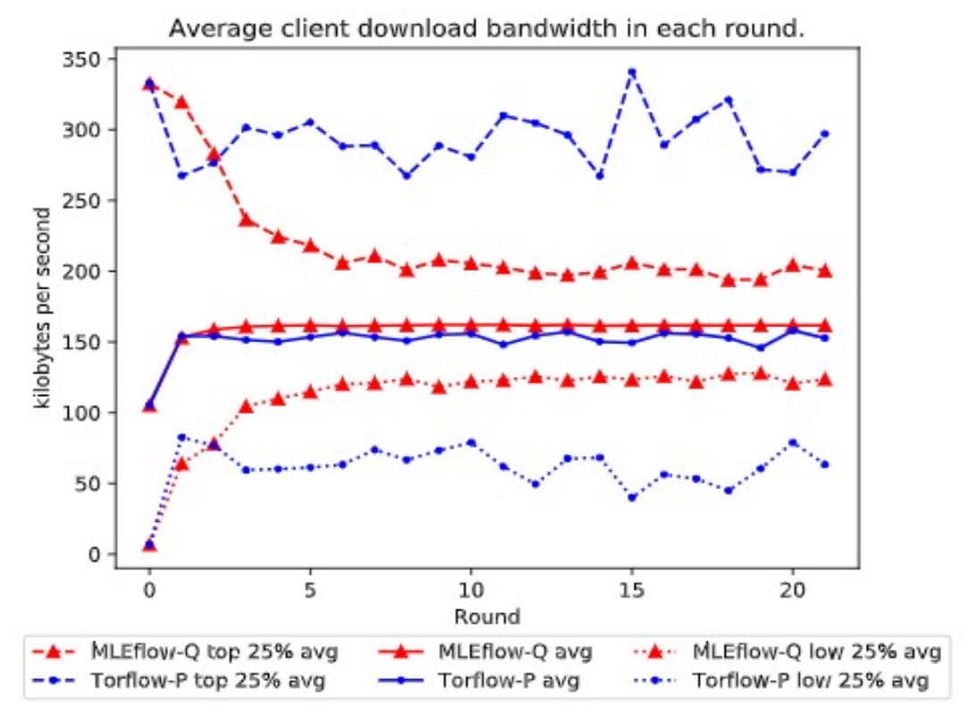


- Shadow creates an environment that allows simulated network connection between virtual nodes (clients, relays, and servers).
- Shadow runs Tor directly out of the box.
- Efficient network simulation in a single box.
- We simulated the developed algorithm in shadow for a 3% network.

✓ Significantly **lower estimation error** than the currently deployed algorithm Torflow.

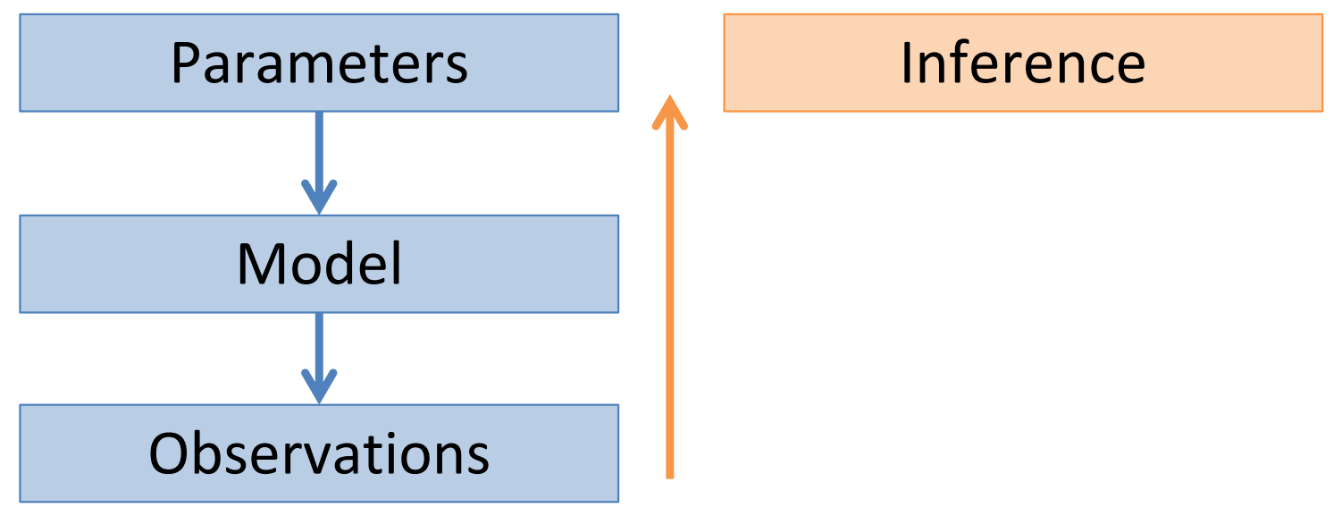


✓ **Fairer distribution** of bandwidth between users = INCREASE service's quality



Future work: Probabilistic programming

- **Probabilistic programming** is a programming paradigm in which probabilistic models are specified and inference for these models is performed automatically.



- Make use of probabilistic programming language (Python pyro) in order to estimate capacities of Tor relays.

Project Website

https://wiki.illinois.edu/wiki/display/MitraResearch/Privacy-preserving+Network+Congestion+Control%3A+Theory+and+Applications

References

[1] https://www.torproject.org/
[2] Hussein Darir, Hussein Sibai, Nikita Borisov, Geir E. Dullerud, Sayan Mitra: TightRope: Towards Optimal Load-balancing of Paths in Anonymous Networks. In WPES '18: 2018 Workshop on Privacy in the Electronic Society, Oct. 15, 2018, Toronto, ON, Canada.
[3] https://metrics.torproject.org/
[4] Working paper/under review: Hussein Darir, Hussein Sibai, Chester Cheng, Nikita Borisov, Geir E. Dullerud, Sayan Mitra: MLEFlow: Learning from History to Improve Load Balancing in Tor.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1739966.