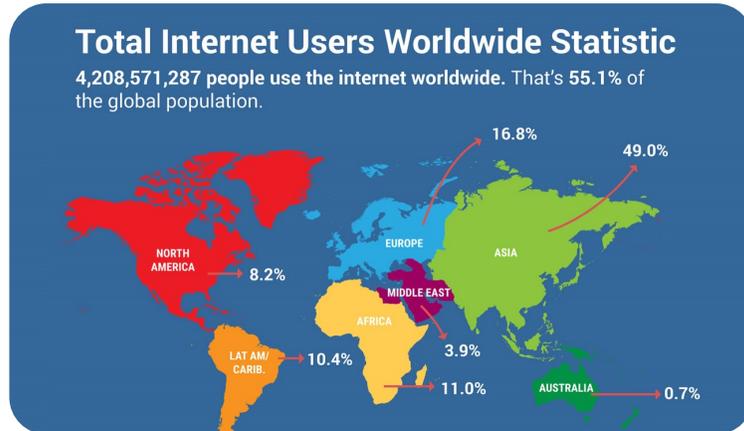


# Illuminating the hidden challenges of data-driven CDNs

Theophilus A. Benson  
Carnegie Mellon University

# Digital Services and their Importance



Impacts productivity

*"89% of remote workers lost 6.3% productivity due to **poor connectivity.**"*

[Zen Internet – broadband from U.K]



Impacts quality-of-life

*"19% elevation in stress due to a 6s **delay** in video loading"*

[Ericsson Mobility Report]



Impacts business revenue

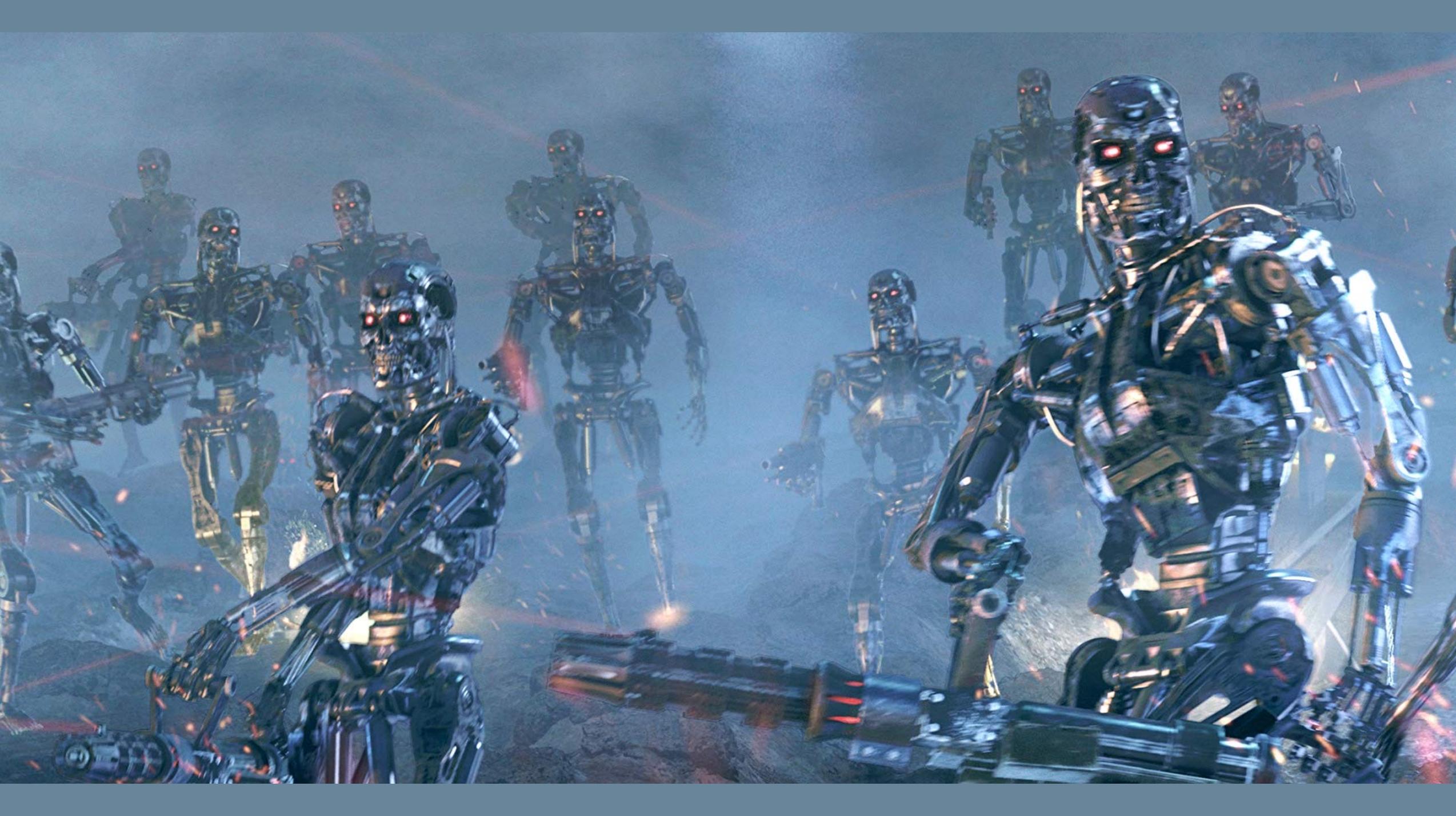
*"10% users lost for 1s **PLT inflation**"* [BBC study]

*"\$530K gain for 100ms **PLT improvement.**"*

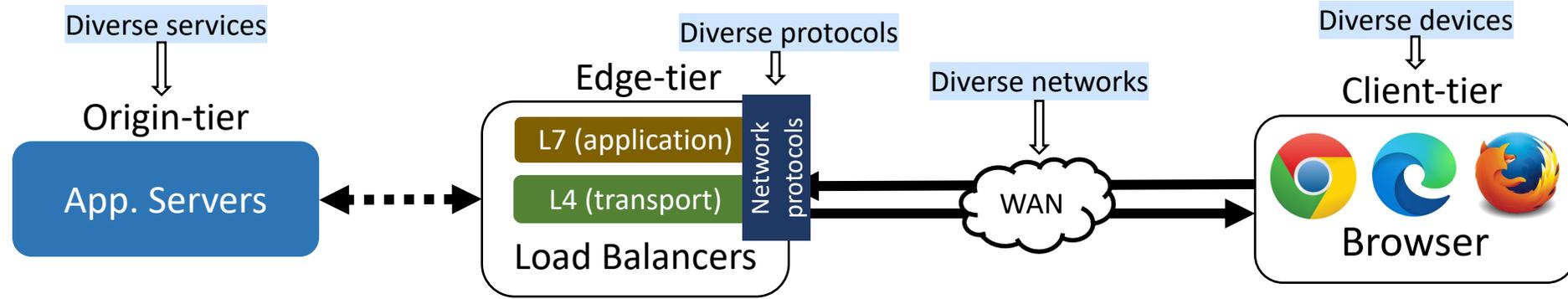
[Mobify report]

*"\$300K-100M loss for an hour **downtime.**"* <sup>2</sup>

[Atlassian report, Fastly study]



# Use of CDNs to Maintain QoE



**Data-driven == Systematic Approach to Tackle Diversity**

# Predictive Caching@Scale

## A scalable ML caching at the Edge

Vaishnav Janardhan  
Adit Bhardwaj

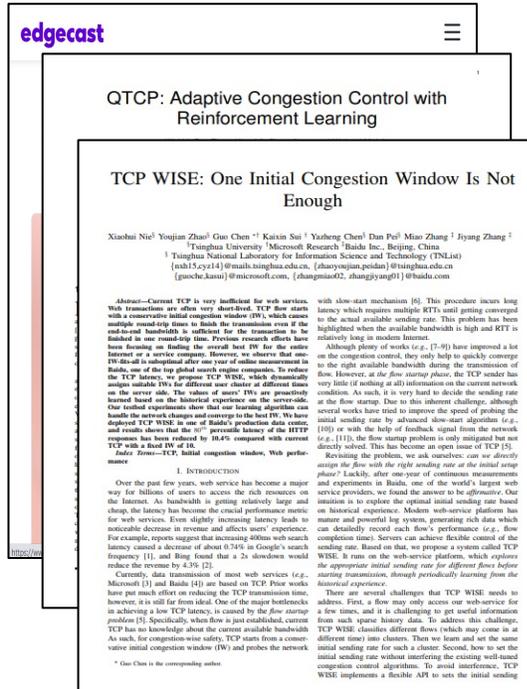
Engineering at Meta

Open Source ▾ Platforms ▾ Infrastructure Systems ▾ Physical Infrastructure ▾ Video Engineering & AR/VR ▾



POSTED ON JUNE 28, 2018 TO AI RESEARCH, DATA INFRASTRUCTURE

# Spiral: Self-tuning services via real-time machine learning



### QTCP: Adaptive Congestion Control with Reinforcement Learning

### TCP WISE: One Initial Congestion Window Is Not Enough

Xiaohu Nie<sup>1</sup> Youjun Zhao<sup>2</sup> Guo Chen<sup>1\*</sup> Kaixin Sui<sup>1</sup> Yazheng Chen<sup>1</sup> Dun Pei<sup>1</sup> Miao Zhang<sup>1</sup> Jiyang Zhang<sup>1</sup>  
<sup>1</sup>Tsinghua University <sup>2</sup>Microsoft Research <sup>3</sup>Baidu Inc., Beijing, China  
<sup>1</sup>niehx15.ez14@mails.tsinghua.edu.cn, [zhaojunyoujun.peidun]@tsinghua.edu.cn  
[guochu.kasui]@microsoft.com, [zhangmiao02, zhangjiyang01]@baidu.com

**Abstract**—Current TCP is very inefficient for web services. Web transactions are often very short-lived. TCP flow starts with a conservative initial congestion window (IW), which causes multiple round-trip times to finish the transmission even if the end-to-end bandwidth is sufficient for the transaction to be finished in one round-trip time. Previous research efforts have been focusing on finding the overall best IW for the entire Internet or a service company. However, we observe that one IW should be substituted after one year of online measurement in Baidu, one of the top global search engine companies. To reduce the TCP latency, we propose TCP WISE, which dynamically adapts the IWs for different user clusters at different times on the server side. The values of server IWs are proactively learned based on the historical experience on the server-side. Our method experiments show that our learning algorithm can handle the network changes and converge to the best IW. We have deployed TCP WISE in one of Baidu's production data centers and results show that the 50% per-connection latency of the RTT response has been reduced by 10.4% compared with current TCP with a fixed IW of 10.

**Index Terms**—TCP, initial congestion window, Web performance.

#### 1. INTRODUCTION

Over the past few years, web service has become a major way for billions of users to access the rich resources on the Internet. As bandwidth is getting relatively large and cheap, the latency has become the critical performance metric for web services. Even slightly increasing latency leads to noticeable decrease in revenue and affects users' experience. For example, reports suggest that increasing 40ms web search latency caused a decrease of about 0.74% in Google's search frequency [1], and Bing found that a 2s slowdown would reduce the revenue by 4.3% [2].

Currently, data transmission of most web services (e.g., Microsoft [3] and Baidu [4]) are based on TCP. Prior works have put much effort on reducing the TCP transmission time, however, it is still far from ideal. One of the major bottlenecks in achieving a low TCP latency, is caused by the flow startup problem [5]. Specifically, when flow is just established, current TCP has no knowledge about the current available bandwidth. As such, for congestion-wise safety, TCP starts from a conservative initial congestion window (IW) and probes the network with slow-start mechanism [6]. This procedure incurs long latency which requires multiple RTTs until getting converged to the actual available sending rate. This problem has been highlighted when the available bandwidth is high and RTT is relatively long in modern Internet.

Although plenty of works (e.g., [7-9]) have improved a lot on the congestion control, they only help to quickly converge to the right available bandwidth during the transmission of flow. However, at the flow startup phase, the TCP sender has very little of finding all information on the current network condition. As such, it is very hard to decide the sending rate at the flow startup. Due to this inherent challenge, although several works have tried to improve the speed of probing the initial sending rate by advanced slow-start algorithms (e.g., [10]) or with the help of feedback signal from the network (e.g., [11]), the flow startup problem is only mitigated but not directly solved. This has become an open issue on TCP [5].

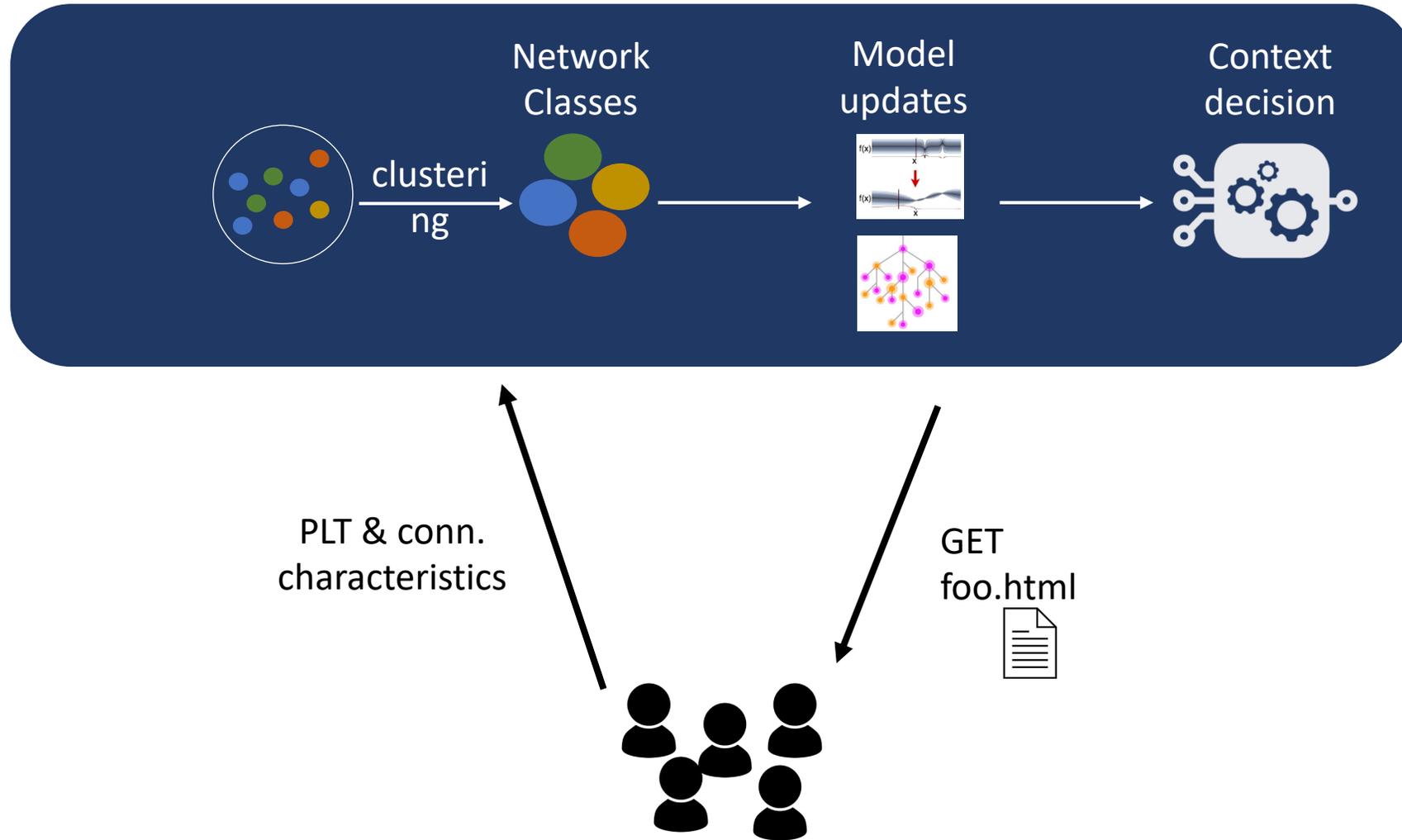
Revisiting the problem, we ask ourselves: can we directly assign the flow with the right sending rate at the initial startup phase? Luckily, after one-year of continuous measurements and experiments in Baidu, one of the world's largest web service providers, we found the answer to be affirmative. Our intuition is to explore the optimal initial sending rate based on historical experience. Modern web-serving platform has mature and powerful log system, generating rich data which can densely record each flow's performance (e.g., flow completion time). Servers can achieve flexible control of the sending rate. Based on that, we propose a system called TCP WISE. It runs on the web-service platforms, which explores the appropriate initial sending rate for different flows before starting transmission, through periodically learning from the historical experience.

There are several challenges that TCP WISE needs to address. First, a flow may only access our web-service for a few times, and it is challenging to get useful information from such sparse history data. To address this challenge, TCP WISE classifies different flows (which may come in at different times) into clusters. Then we learn and set the same initial sending rate for each a cluster. Second, how to set the initial sending rate without interfering the existing well-tuned congestion control algorithms. To avoid interference, TCP WISE implements a flexible API to set the initial sending

Data-driven == Systematic Approach to Tackle Diversity

Lacking systematic and principled investigations of  
the deployment implications of data-driven CDNs

# Configanator Workflow

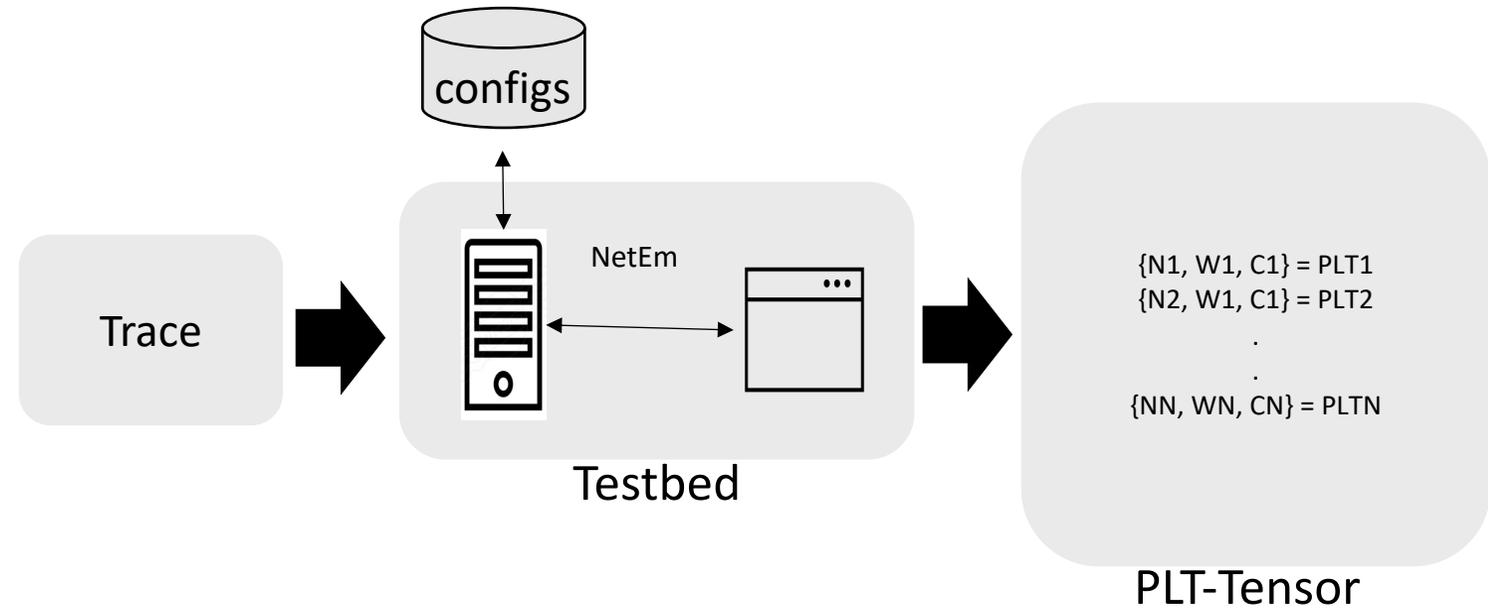


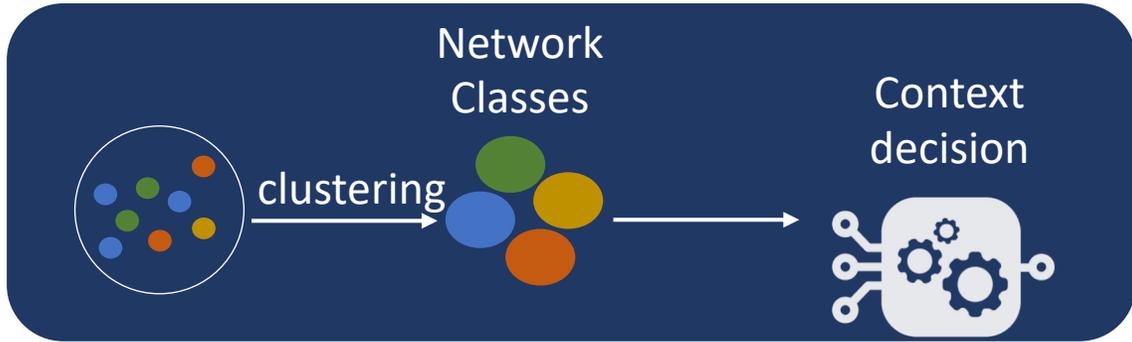
# Evaluation Setup

## Trace-driven simulation

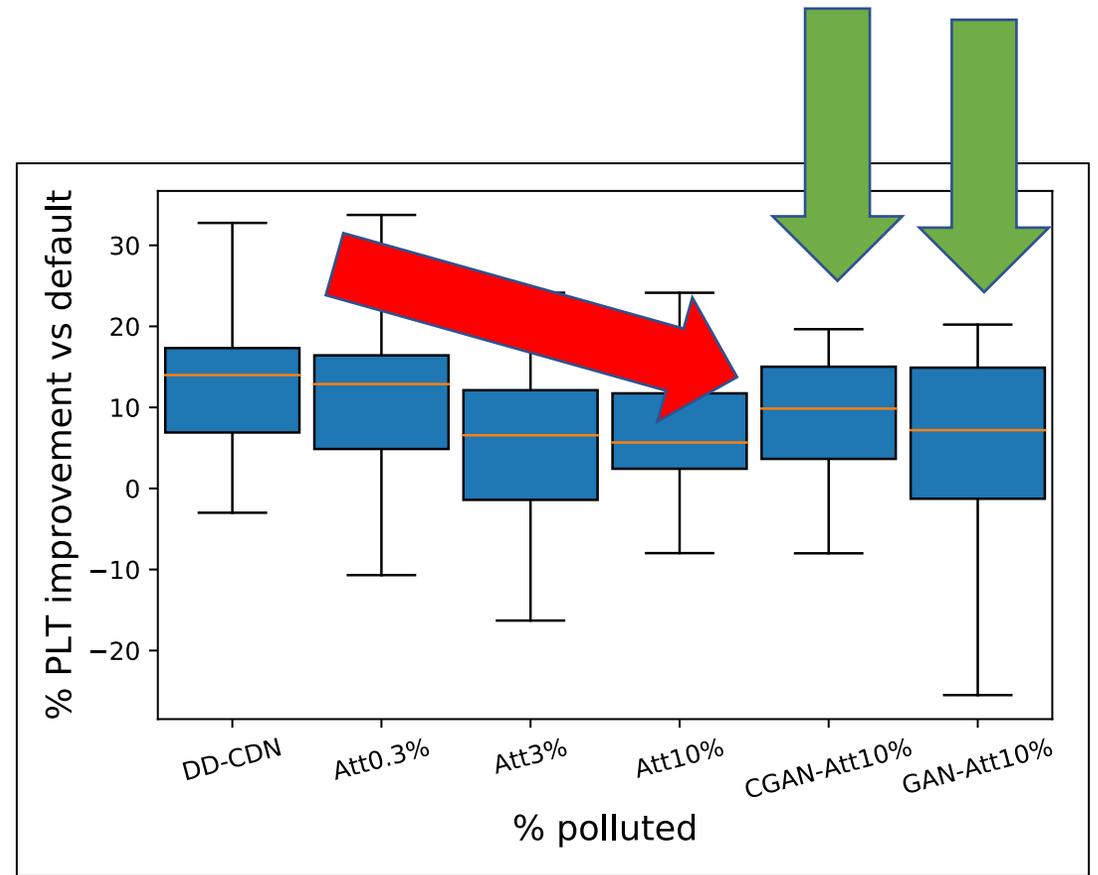
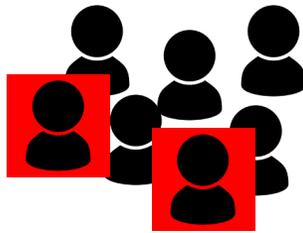
Traces from multiple regions.

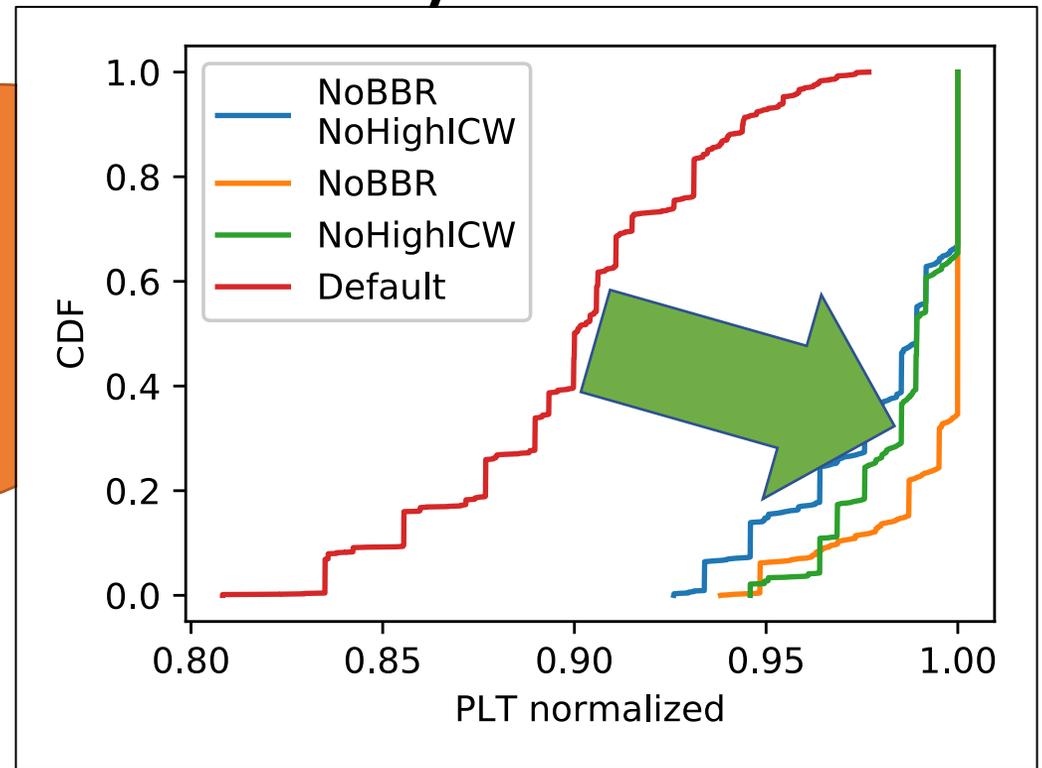
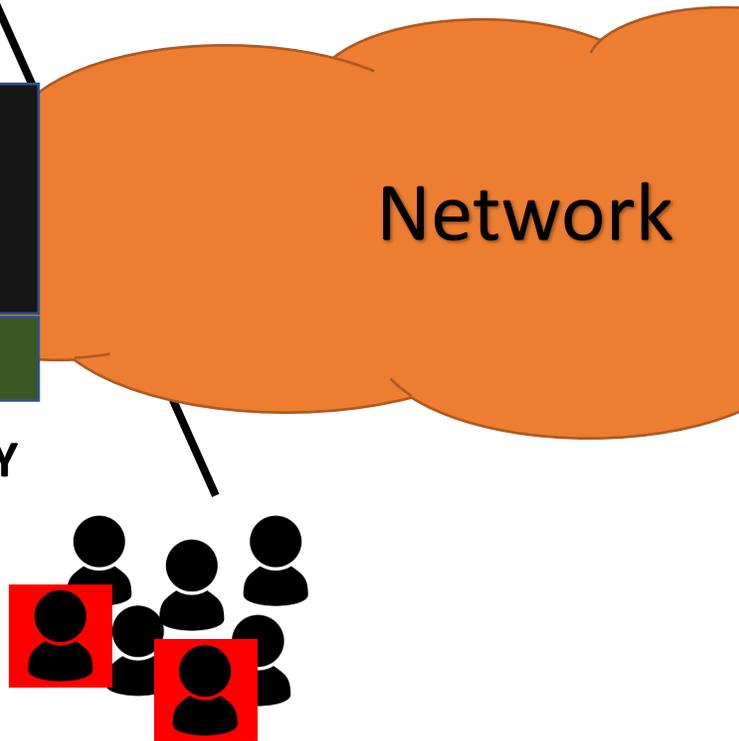
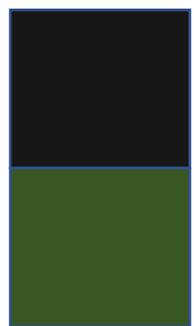
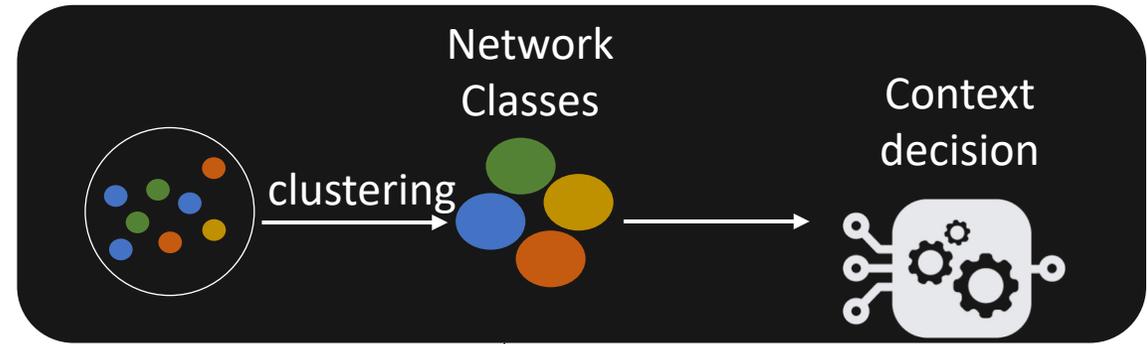
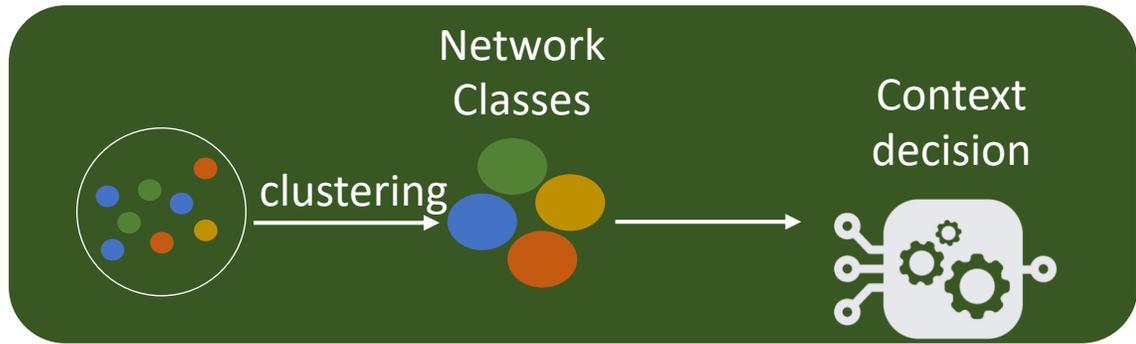
- Global (CDN trace)

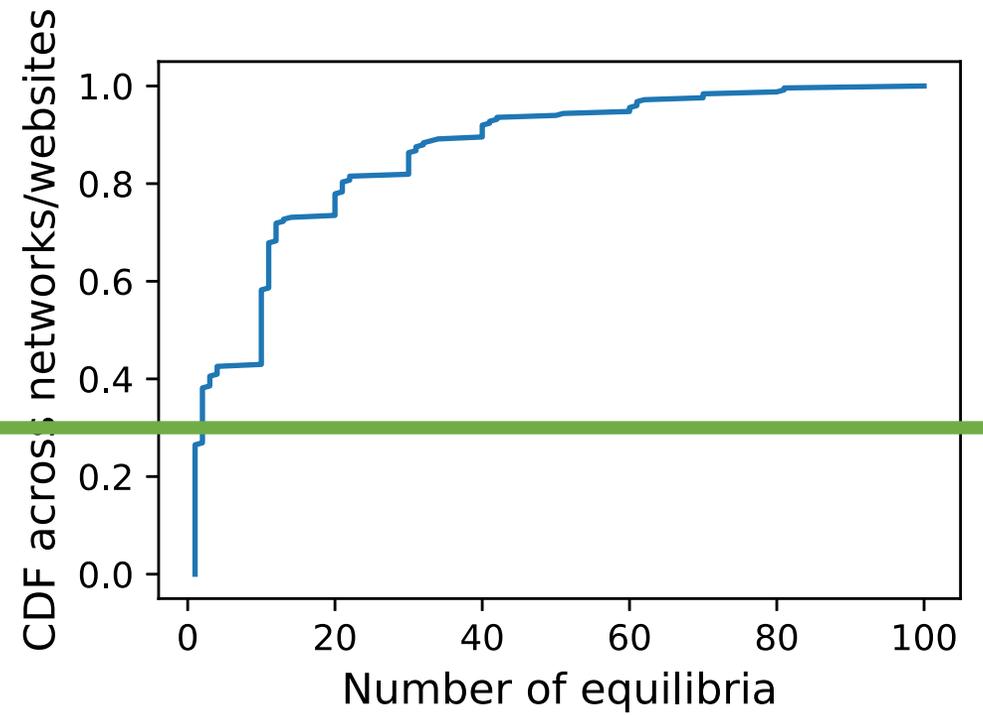
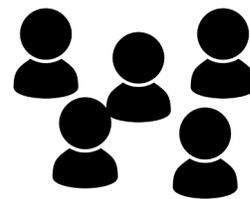
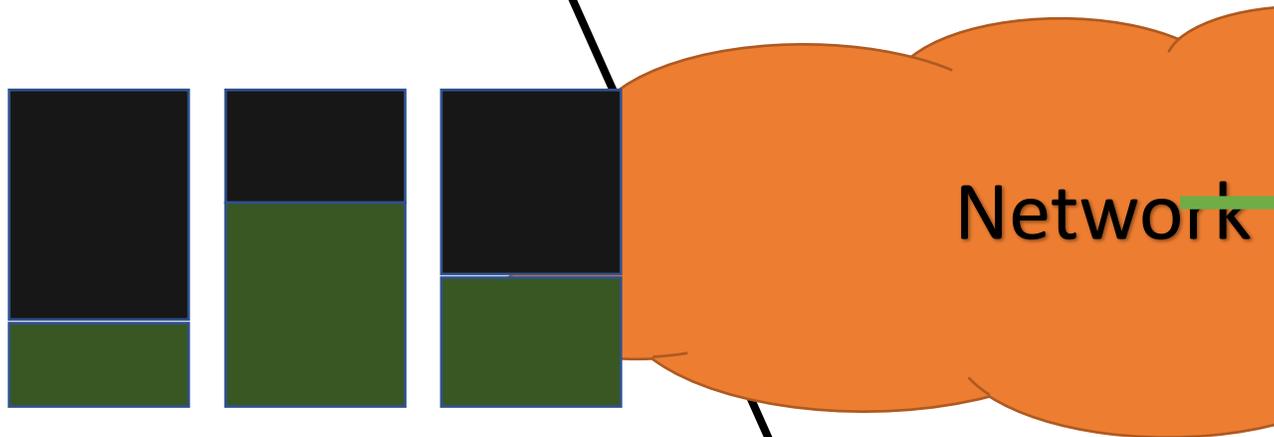
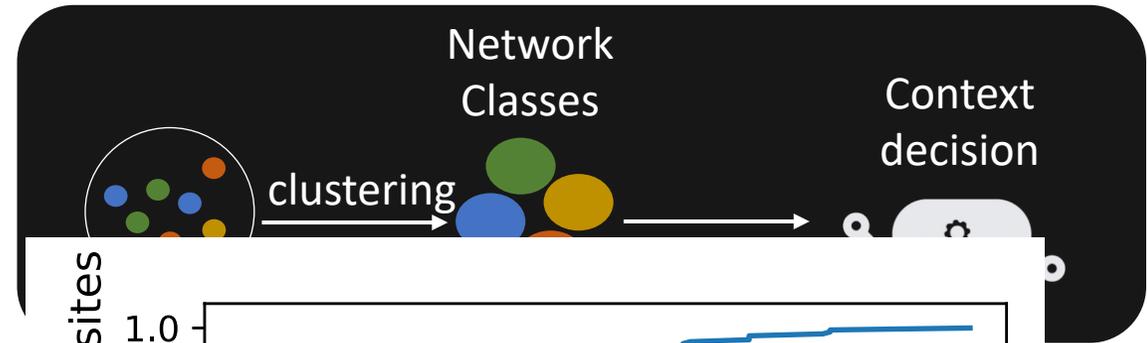
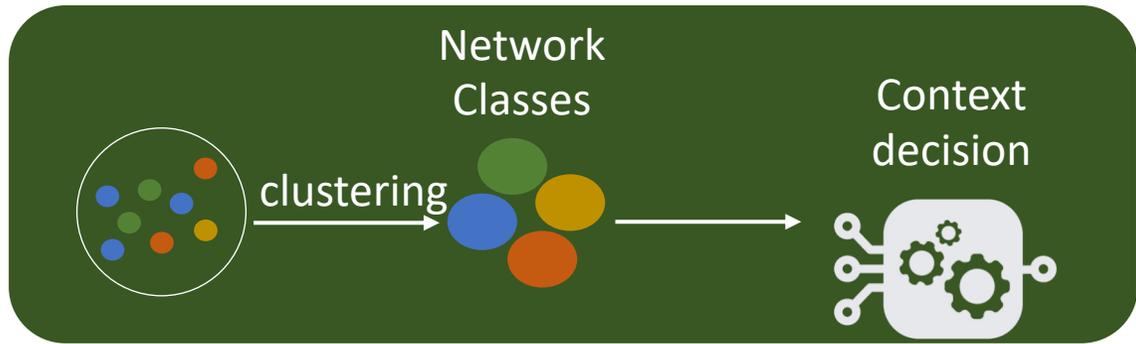




PLT & conn.  
characteristics



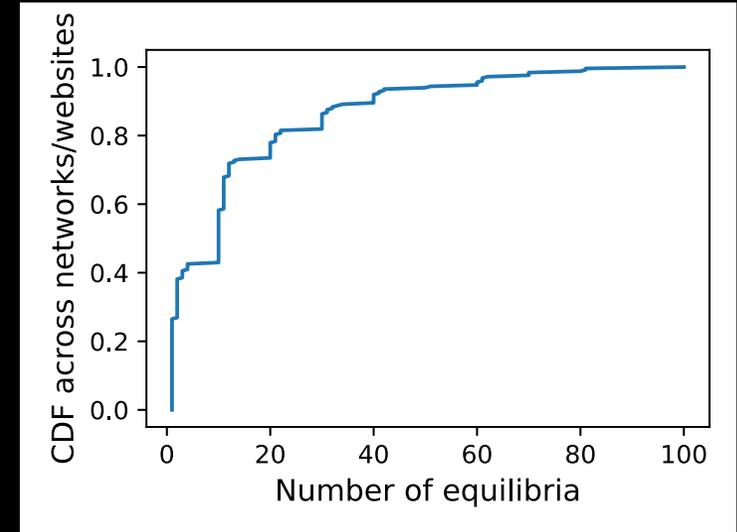
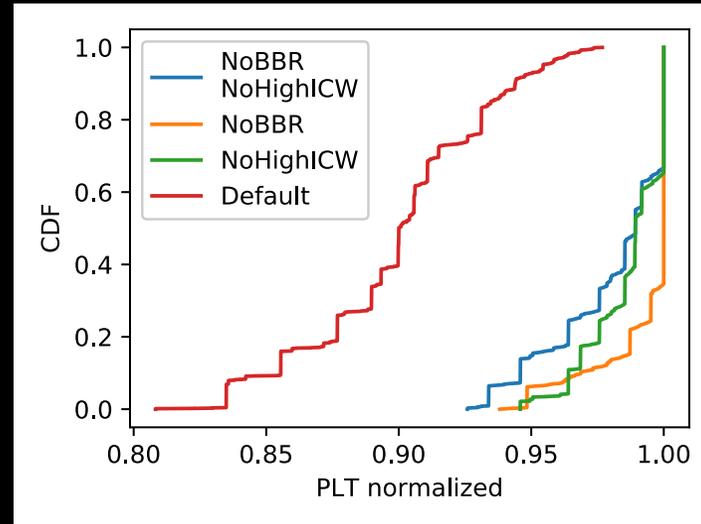
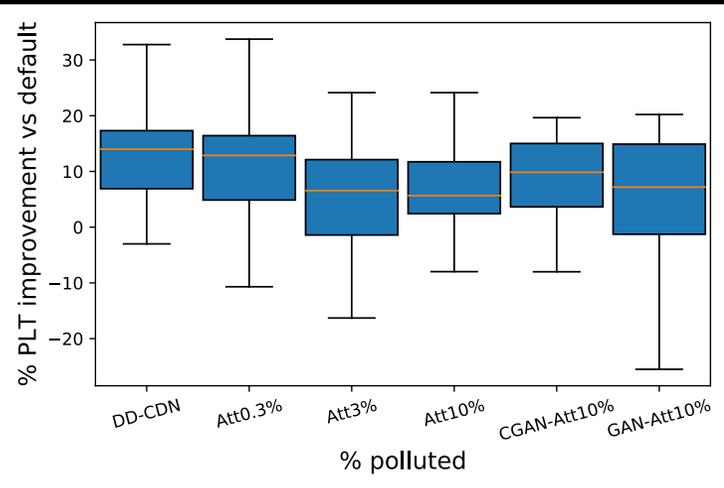




## Security

## Fairness

## Stability

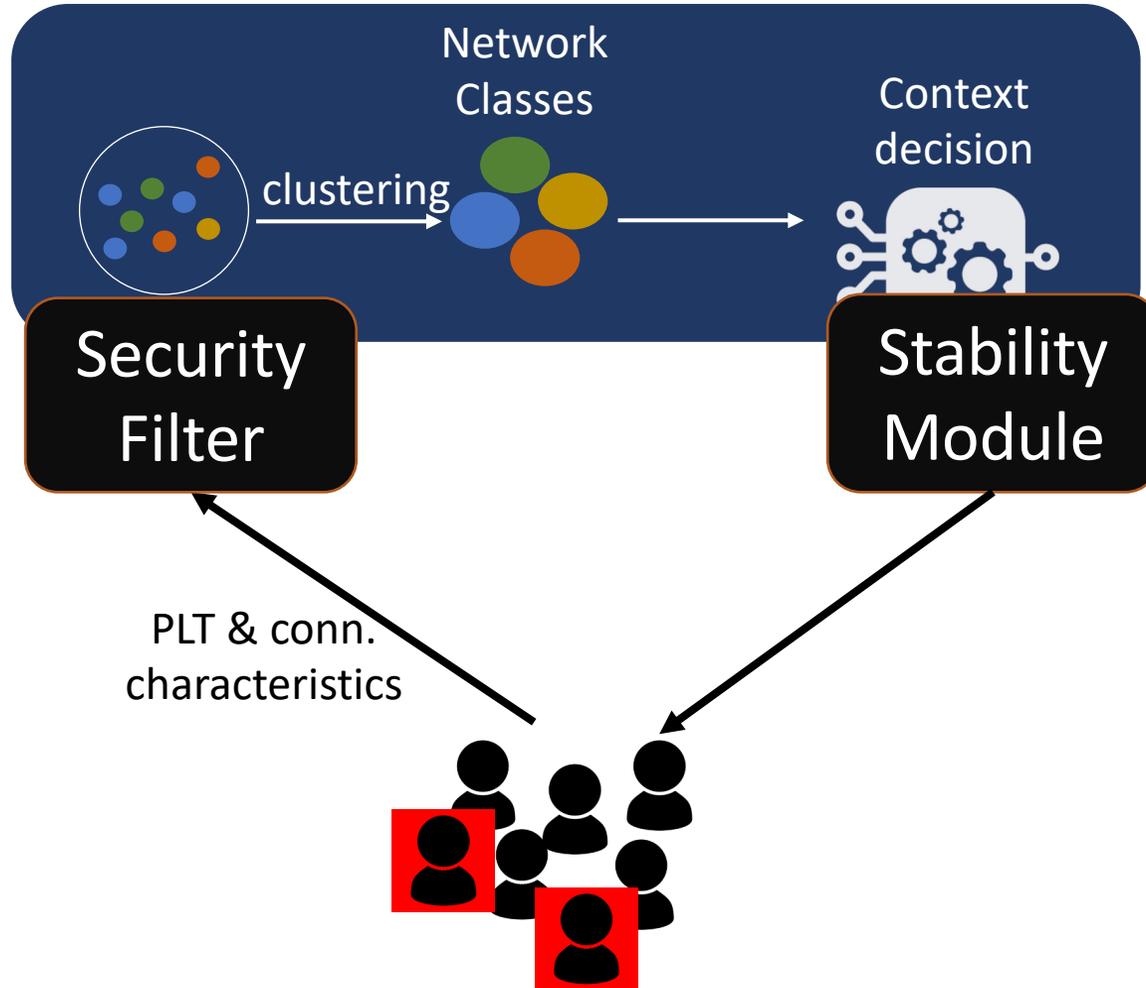


Use side-channel to detect anomalous inputs

Pre-Filter unfair configs

Pre-Filter configs that promote instability

# GuardRails



# Conclusion + Future Work

Growing deployment of  
Data-driven CDNs

Empirically investigated  
deployment challenges

Introduced a strawman  
solution: GuardRails

Plan to explore  
infrastructure support

