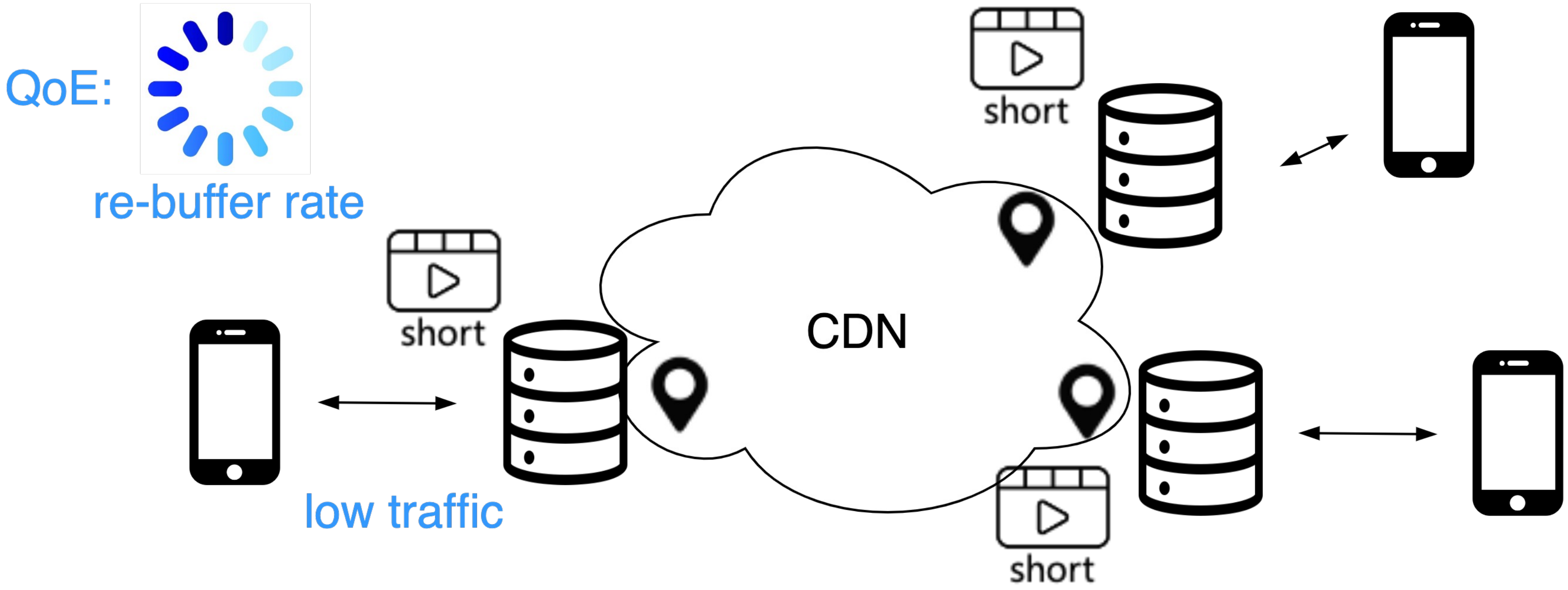
Learning Production-Optimized Congestion Control Selection for Alibaba Cloud CDN

Xuan Zeng[1*], Haoran Xu[2*], Chen Chen[1*], Xumiao Zhang[1], Xiaoxi Zhang[2], Xu Chen[2], Guihai Chen3, Yubing Qiu[1], Yiping Zhang[1],Chong Hao[1], Ennan Zhai[1]

[1]Alibaba Cloud [2]Sun Yat-sen University [3]Nanjing University

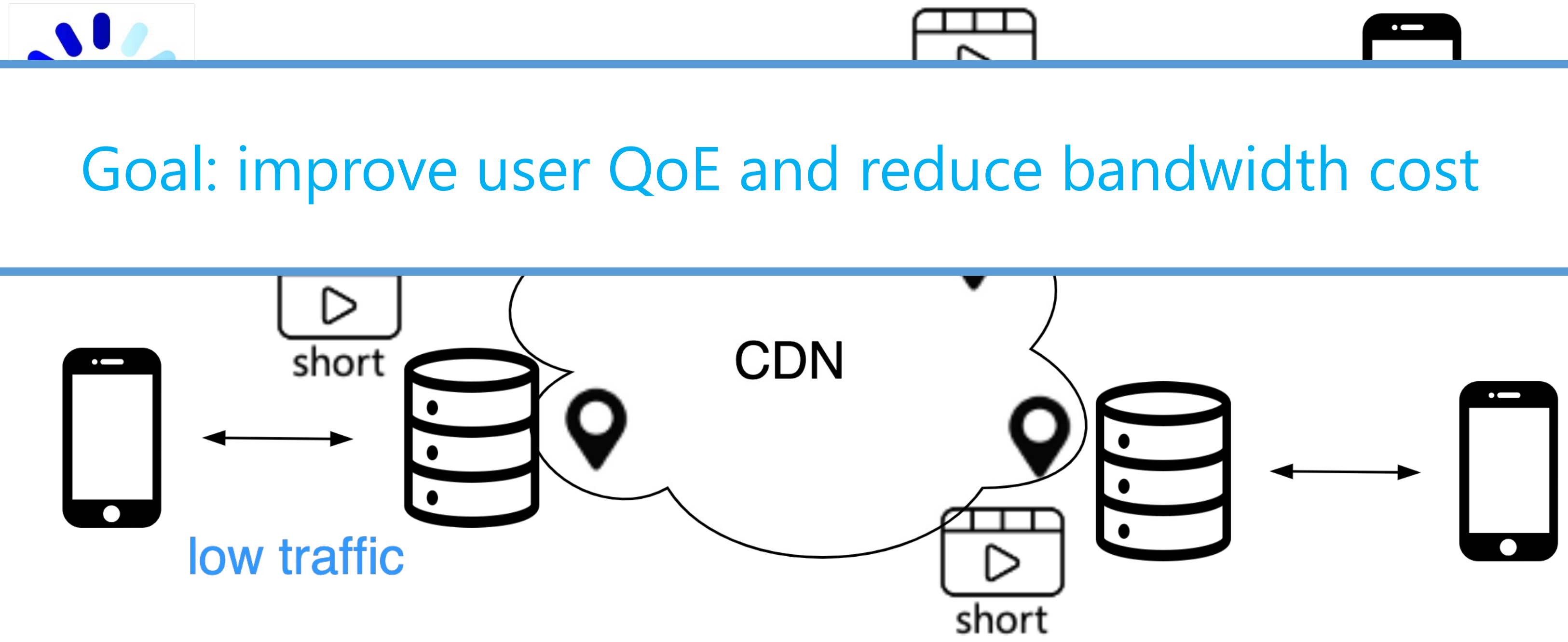
Short Video Service on CDN

- CDN handles 70% global traffic.
- Short Video as major workload



Short Video Service on CDN

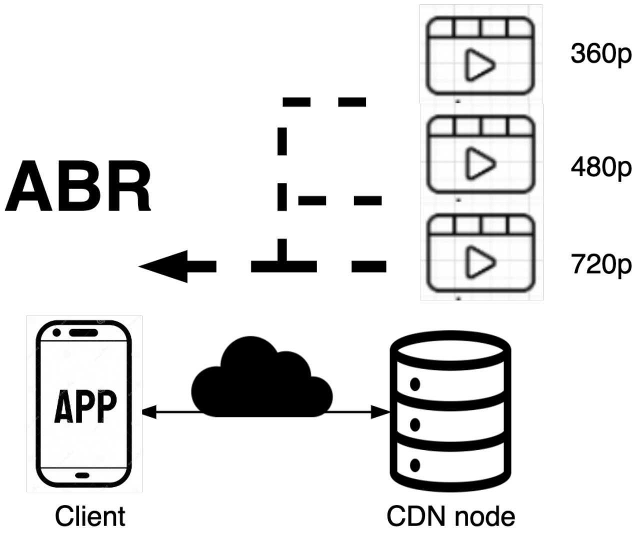
- CDN handles 70% global traffic.
- Short Video as major workload



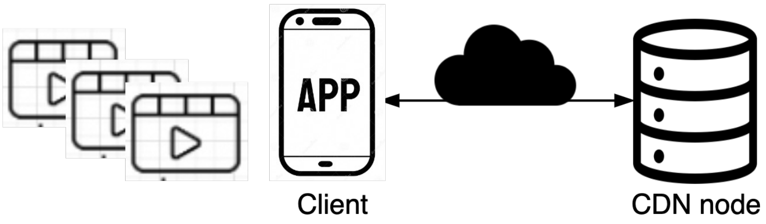
Goal: improve user QoE and reduce bandwidth cost

Approach #1: application-layer strategies

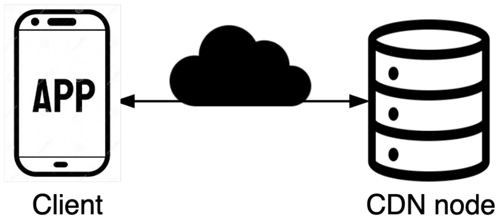
ABR



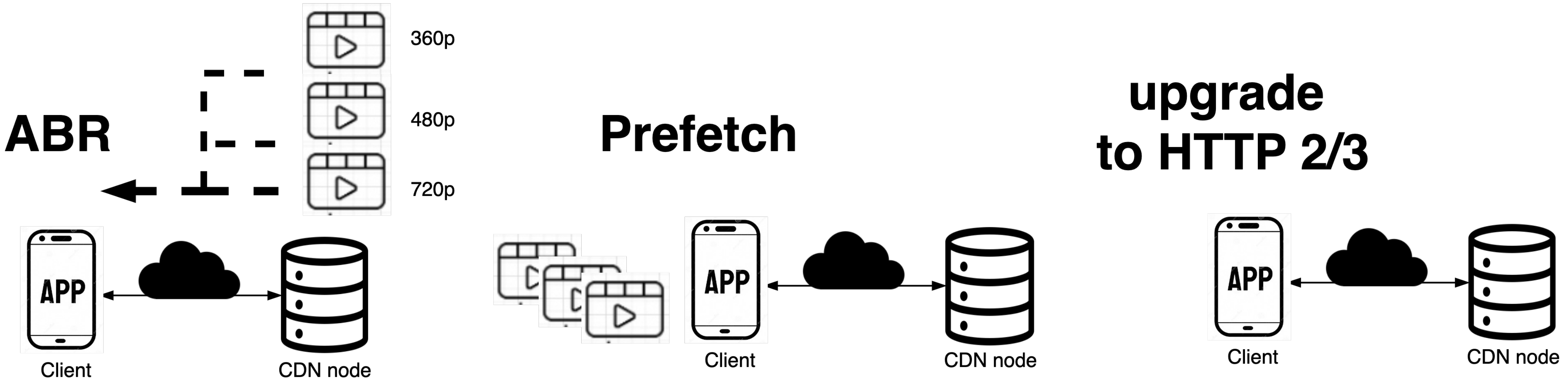
Prefetch



upgrade to HTTP 2/3



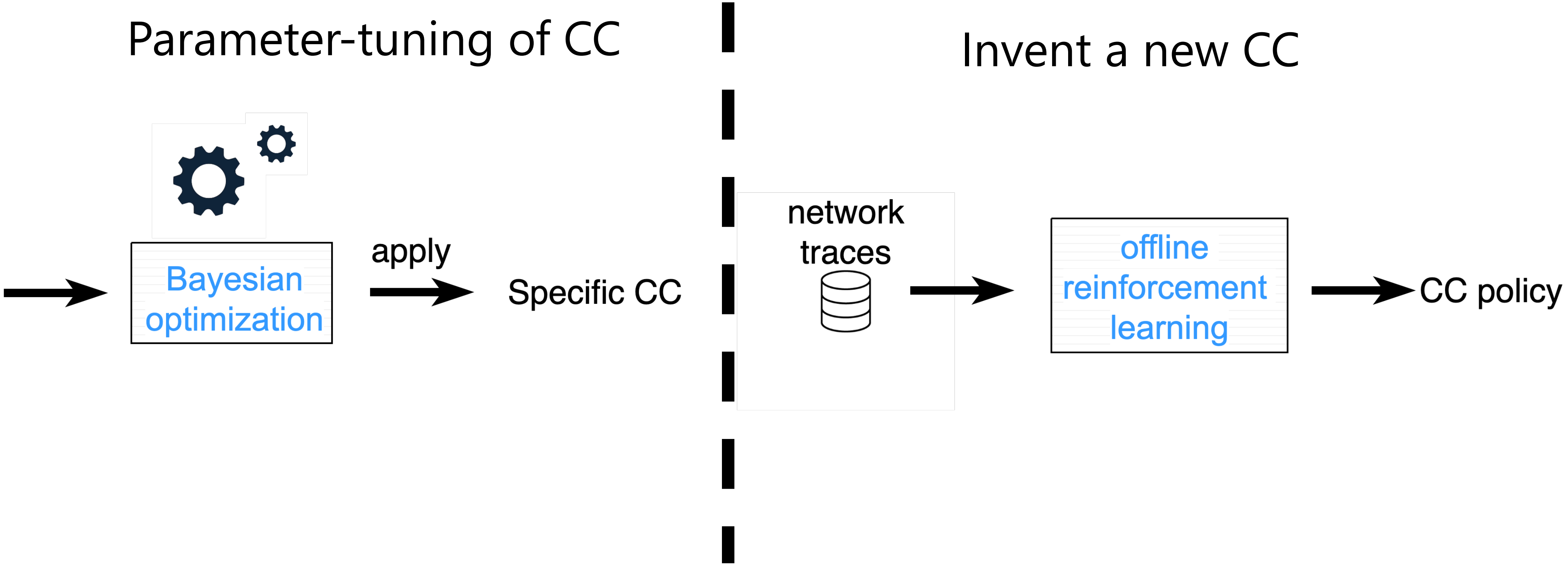
Approach #1: application-layer strategies



- Implemented at client & out of control for CDN

Approach #2: transport-layer strategies

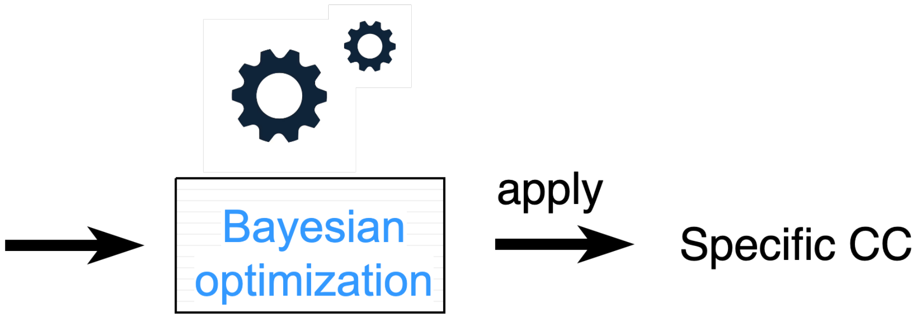
- Adjust a specific type of CC (Congestion Control)



Approach #2: transport-layer strategies

- Adjust a specific type of CC (Congestion Control)

Parameter-tuning of CC



- Can't fix inherent design limitations

Invent a new CC

- Inadequate production testing

Approach #2: transport-layer strategies

- Adjust a specific type of CC (Congestion Control)

Parameter-tuning of CC

Invent a new CC

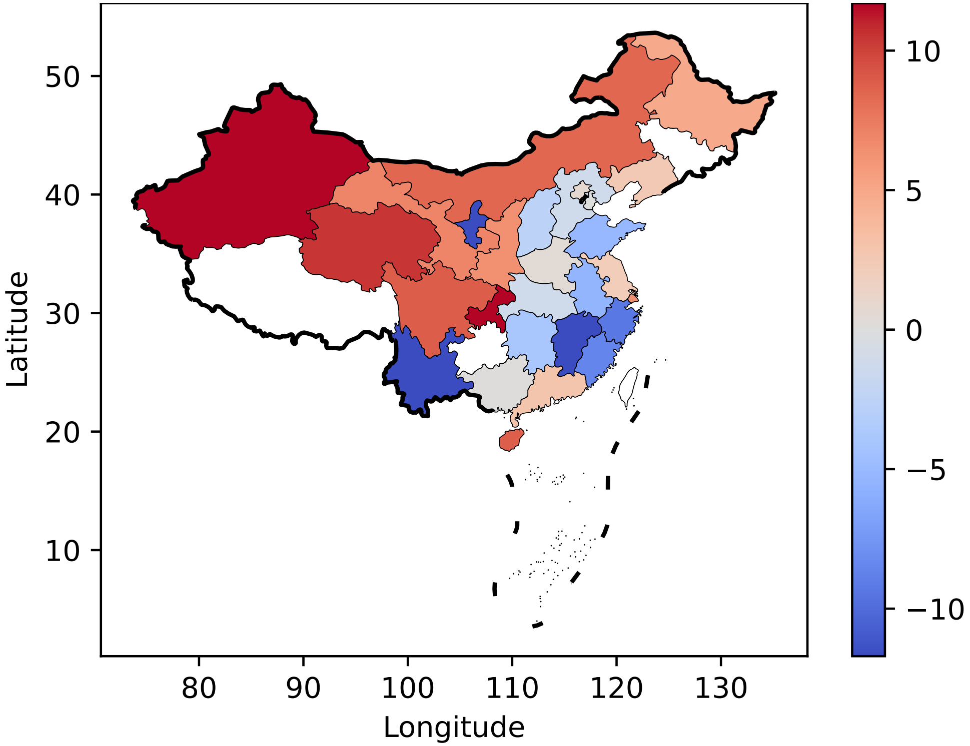
A single CC is insufficient; measurement reveals no CC is always optimal !

- Can't fix inherent design limitations

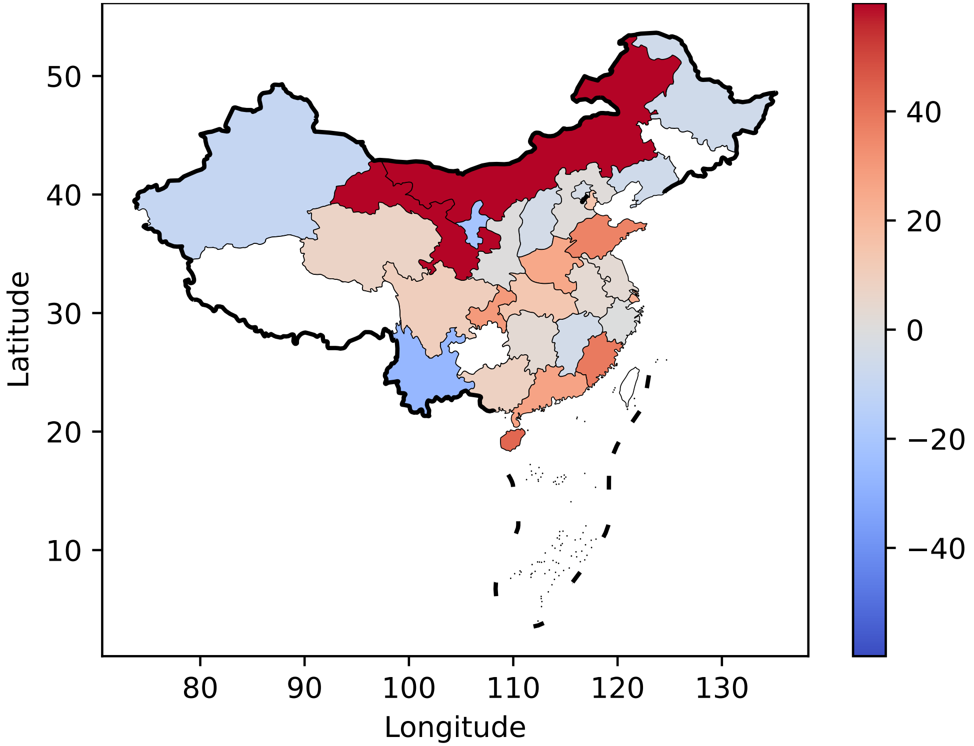
- Inadequate production testing

A/B test of Cubic vs BBR across region

- Vary significantly across regions



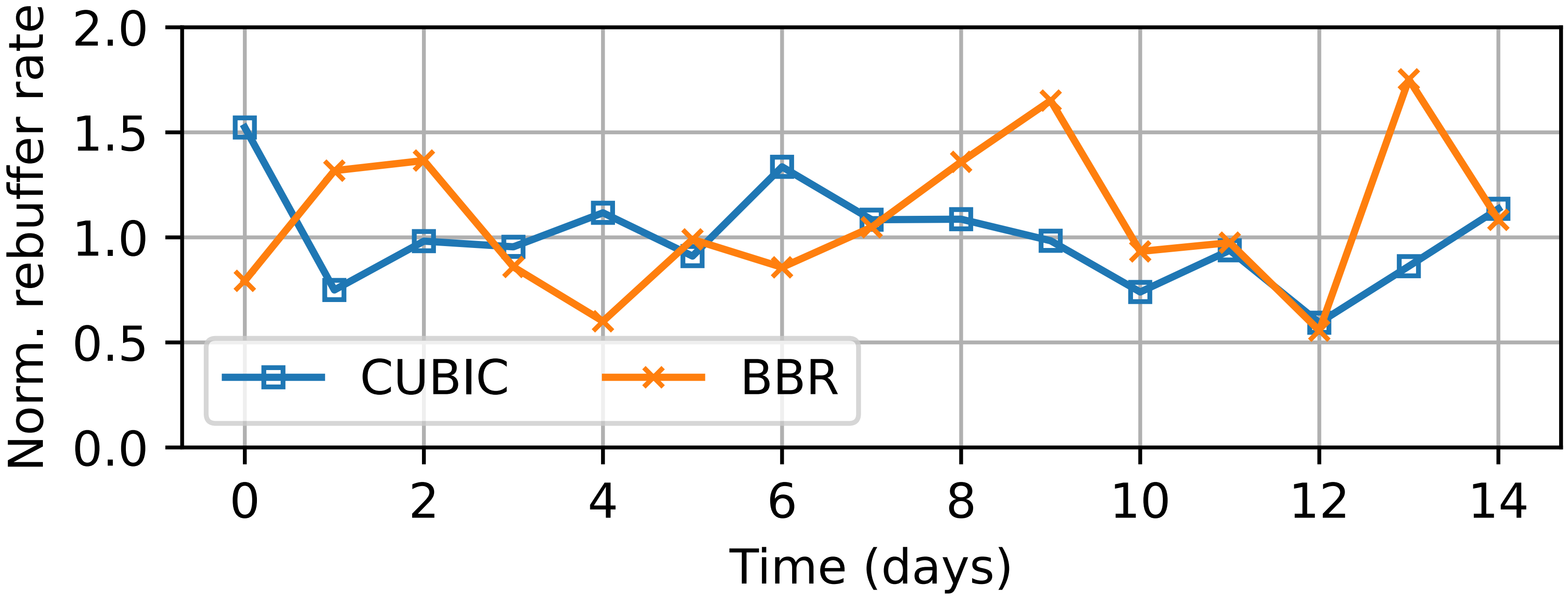
Rebuffer rate



bandwidth cost

A/B test of Cubic vs BBR across time

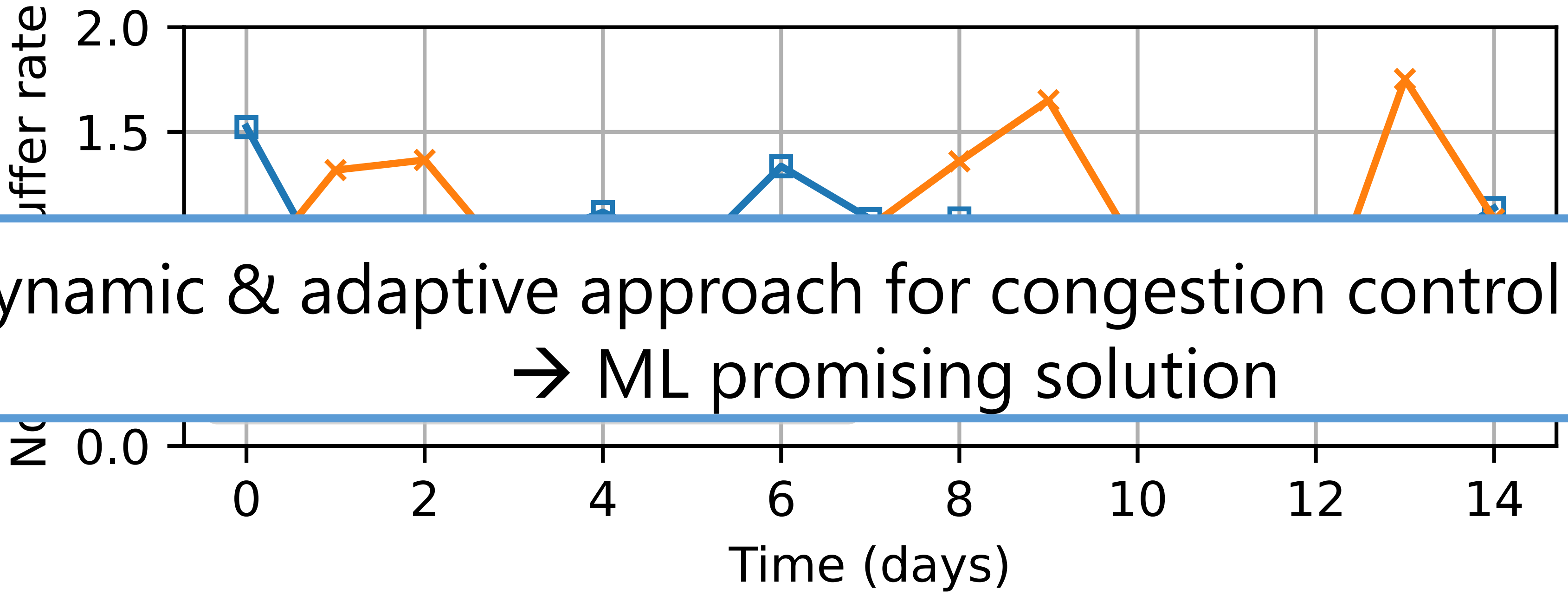
- Fluctuate over time



Cubic vs BBR over time
on one CDN node

A/B test of Cubic vs BBR across time

- Fluctuate over time



Need dynamic & adaptive approach for congestion control selection (CCS)
→ ML promising solution

Cubic vs BBR over time
on one CDN node

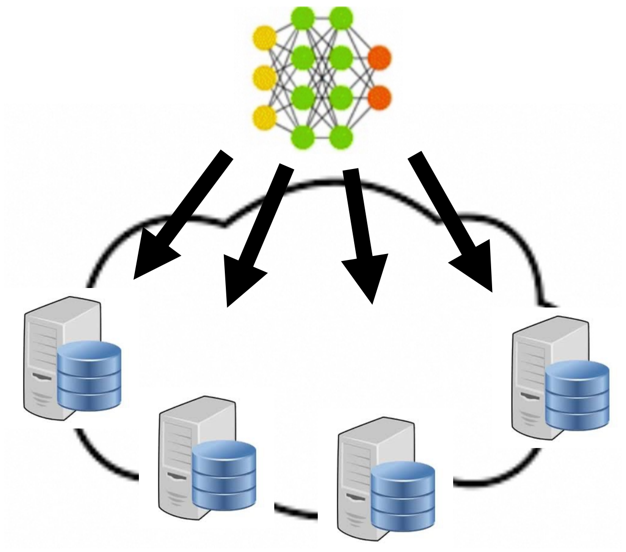
Our approach

- AliCCS: the first ML-based Congestion Control Selection (CCS) framework for Short Video delivery in production CDN
  - surpass each CC's limitations
  - worst-case guarantee (select from well-established CCs)

Key challenges in applying ML for CCS in production CDN

Key challenges in applying ML for CCS in production CDN

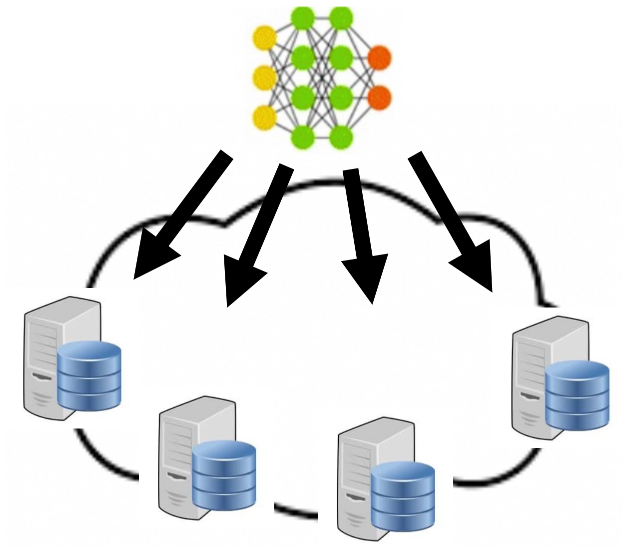
Scalability & Generalization



- Impractical 1 model per node

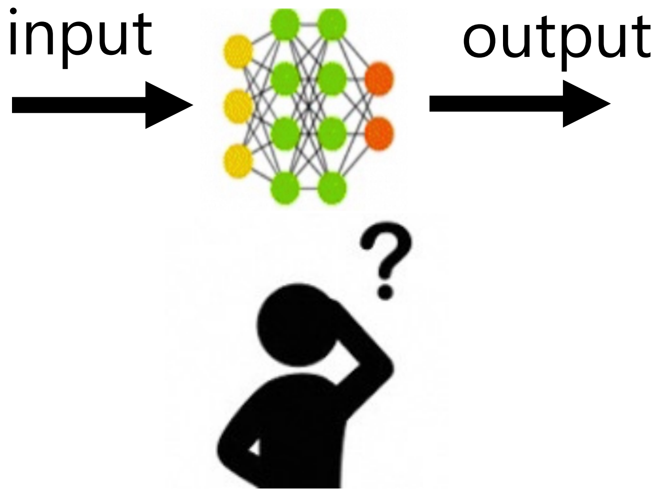
Key challenges in applying ML for CCS in production CDN

Scalability & Generalization



- Impractical 1 model per node

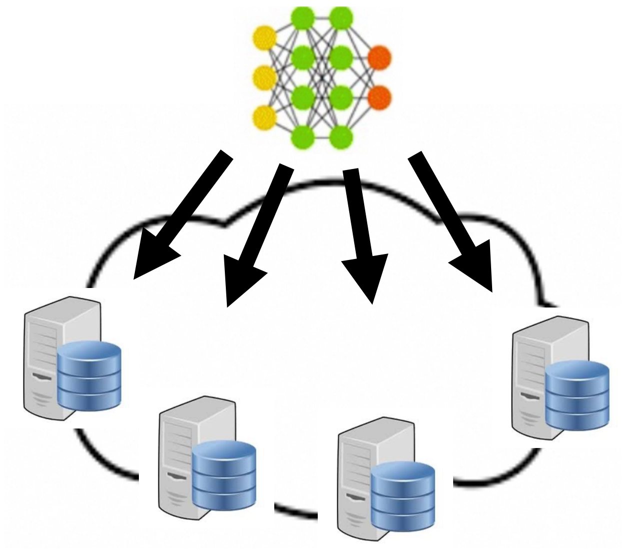
Interpretability



- Troubleshooting & iterate

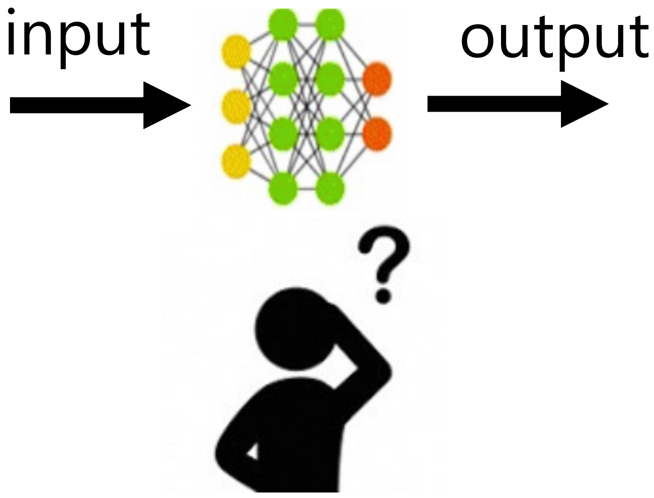
Key challenges in applying ML for CCS in production CDN

Scalability & Generalization



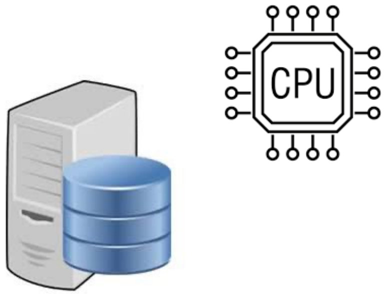
- Impractical 1 model per node

Interpretability



- Troubleshooting & iterate

Reduce Inference delay



- Minimize impact of model Inference delay

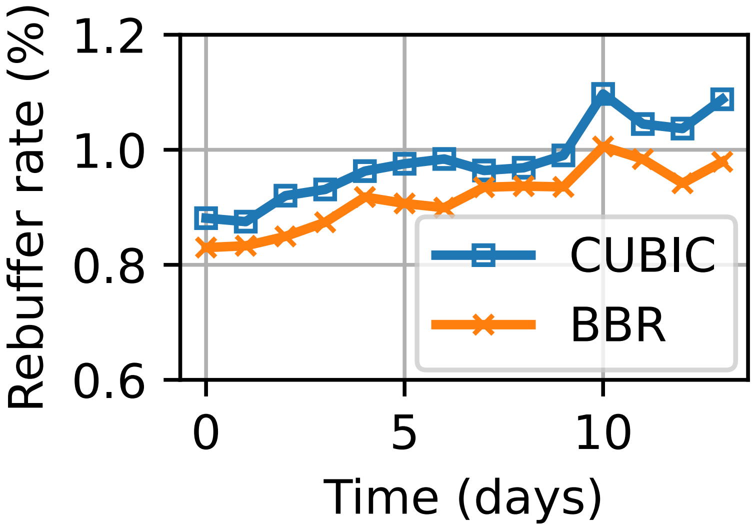
1. Model Scalability and generalization

2. Model prediction interpretability

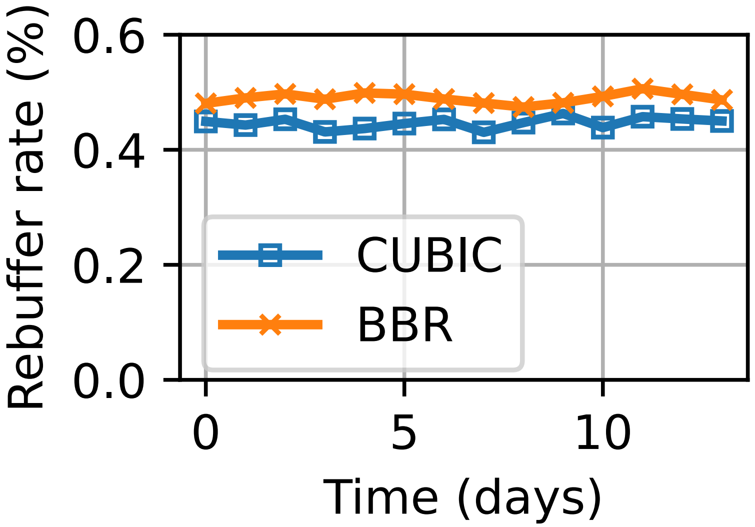
3. Model inference overhead

Key observation: what impacts CCS?

- Cubic vs BBR based on network types

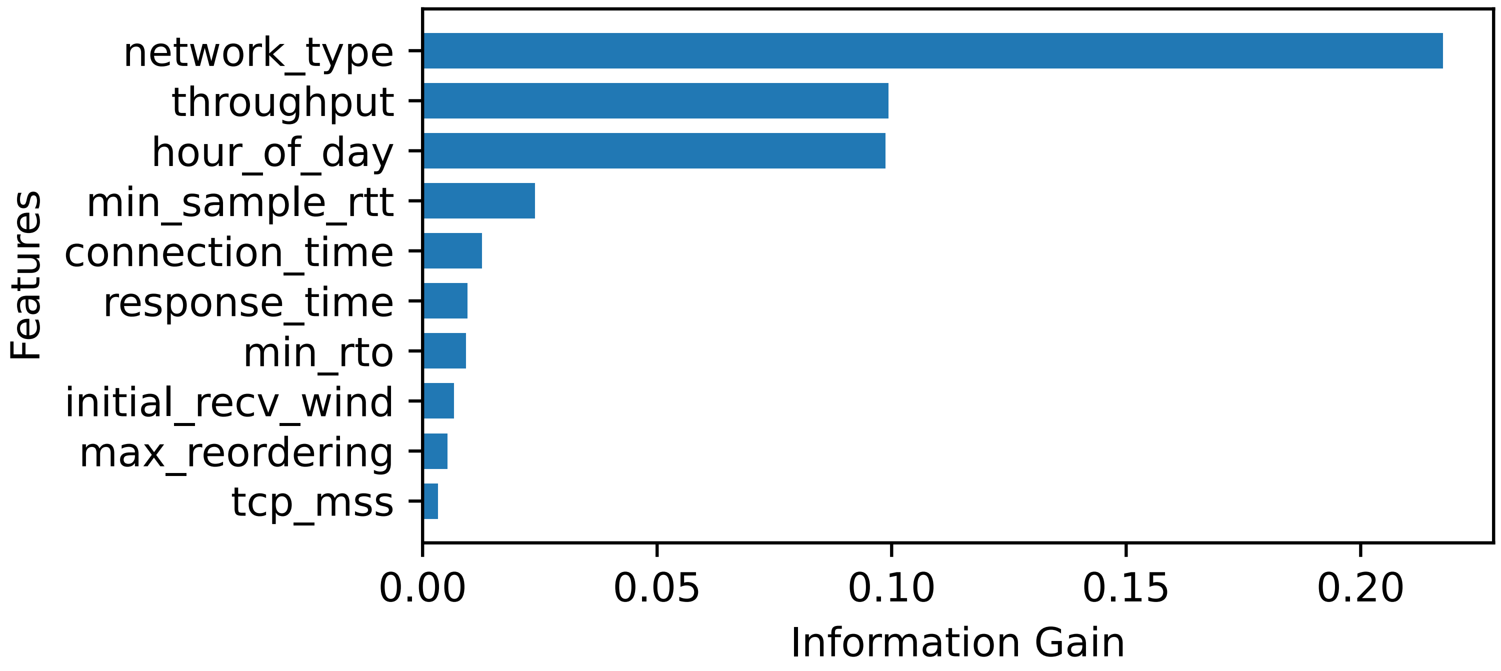


4G

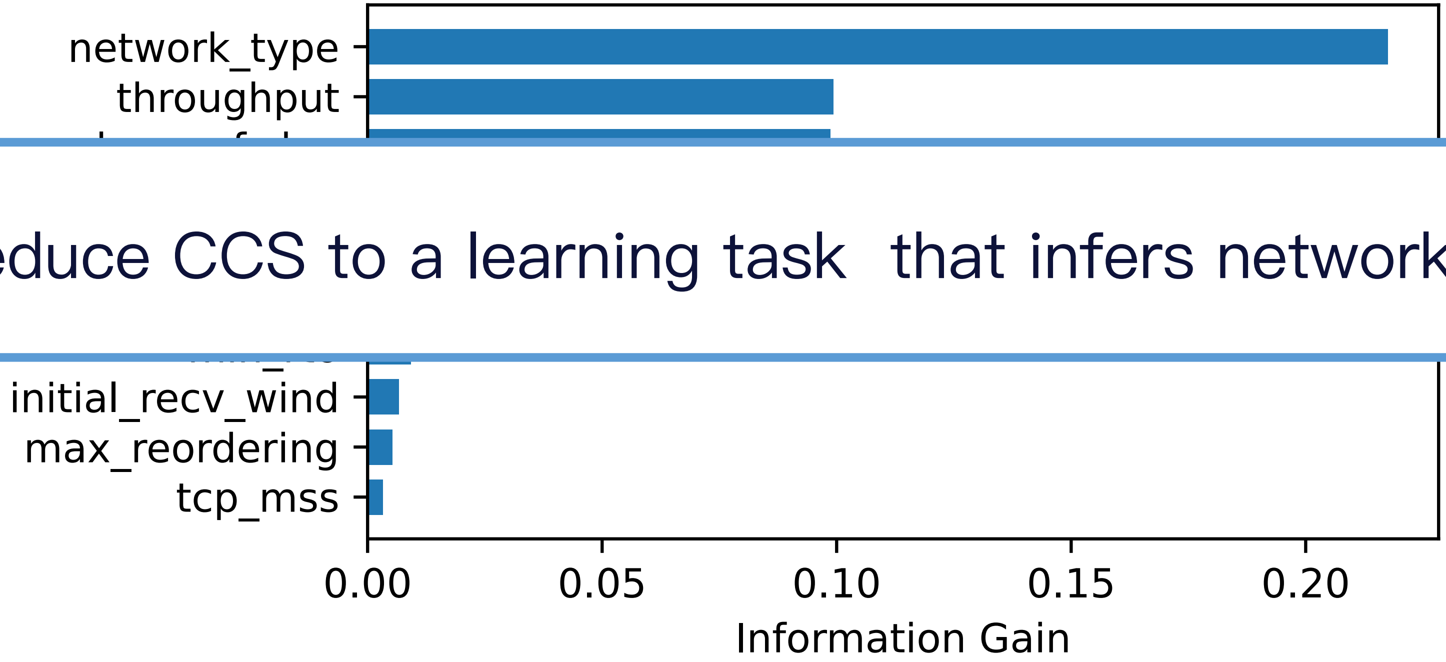


Wi-Fi

Key observation on feature analysis



Key observation on feature analysis



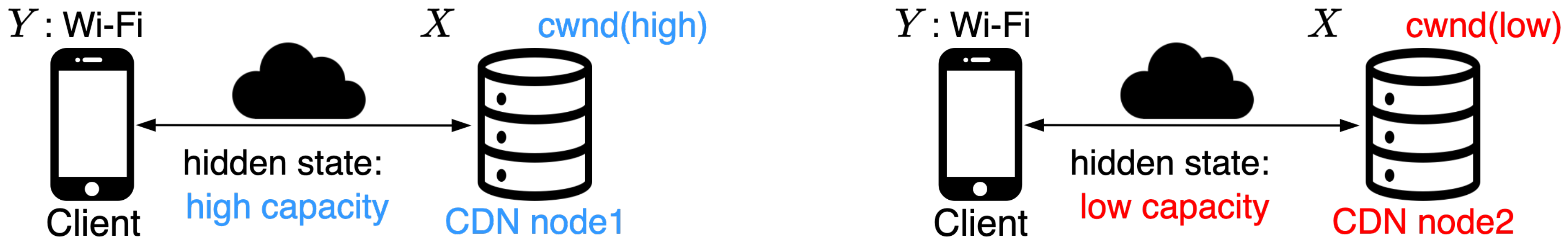
Reduce CCS to a learning task that infers network types from TCP statistics

Model design: learn a generalized model

- Issue with hidden states in the network

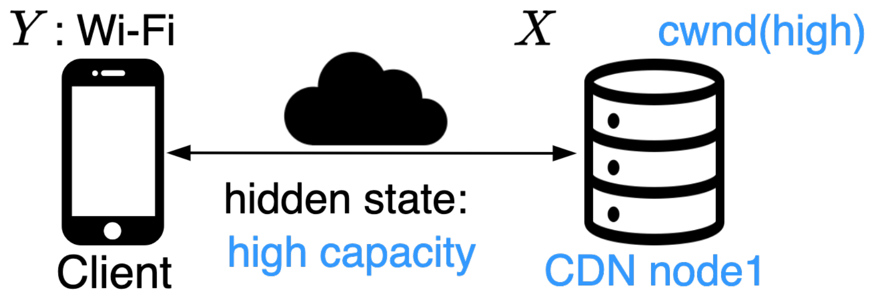
Model design: learn a generalized model

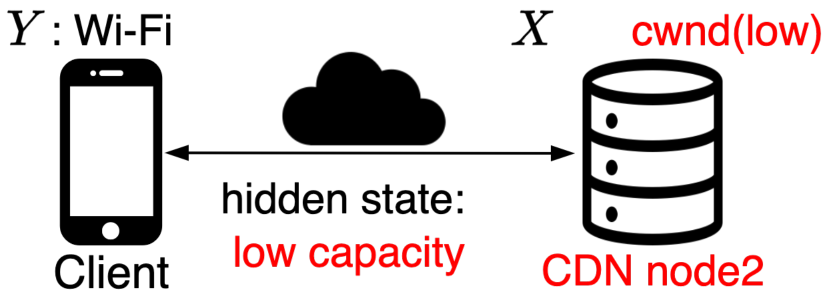
- Issue with hidden states in the network
  - E.g., cwnd as feature X



Model design: learn a generalized model

- Issue with hidden states in the network
  - E.g., cwnd as feature X



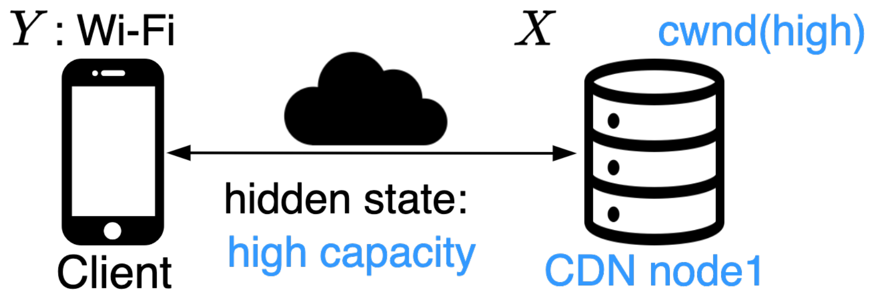


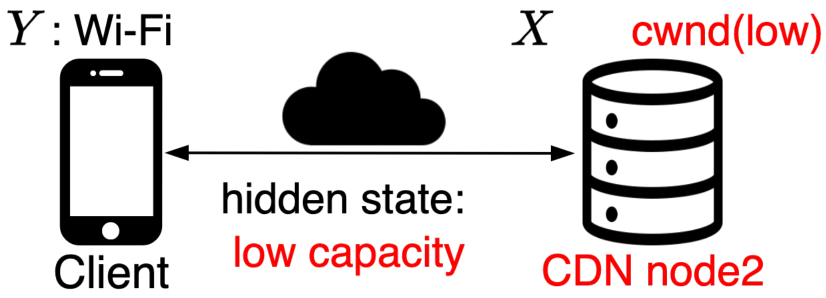
Data: Y:Wi-Fi ←→ X: high cwnd

Y:Wi-Fi ←→ X: low cwnd

Model design: learn a generalized model

- Issue with hidden states in the network
  - E.g., cwnd as feature X



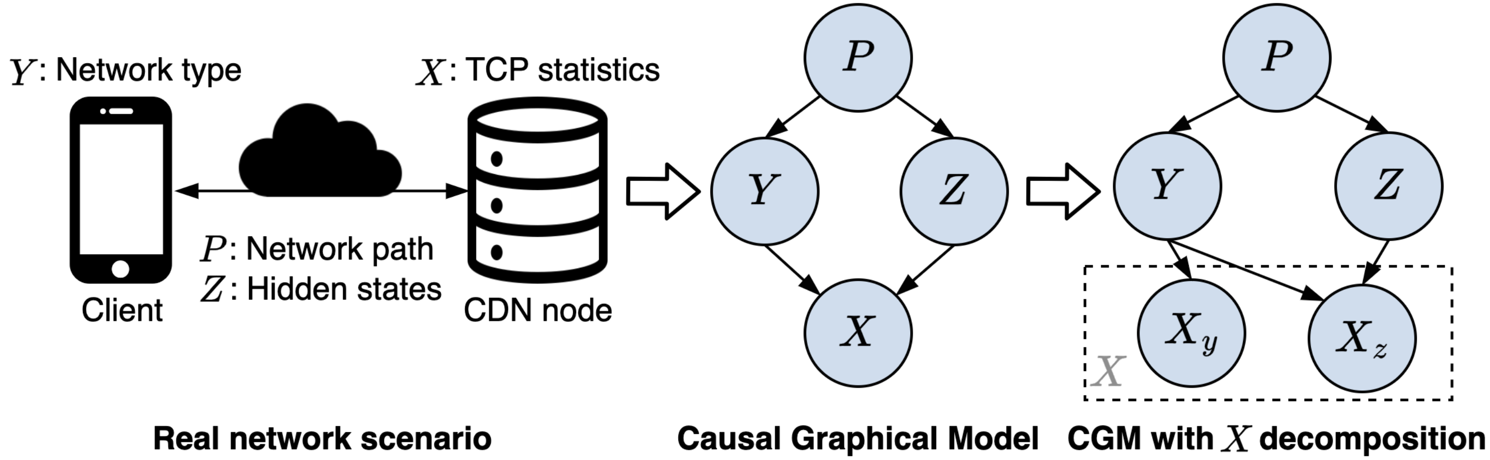


Data: Y:Wi-Fi ←→ X: high cwnd Y:Wi-Fi ←→ X: low cwnd

Inconsistency causes trouble in learning unified model !

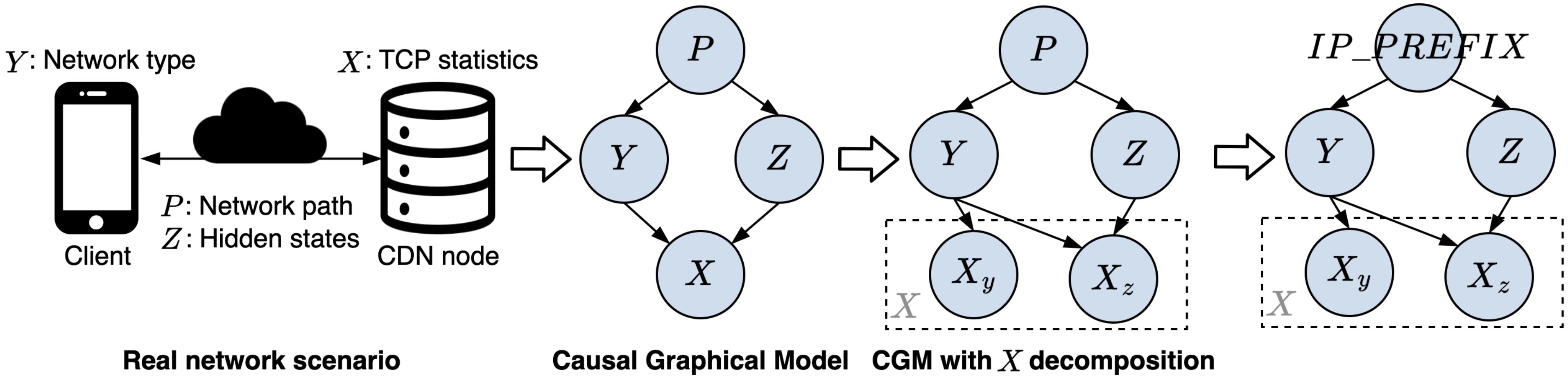
Model design: key intuition

- Key observation: one-to-one mapping between /24 IP prefix and Internet path (or hidden states)



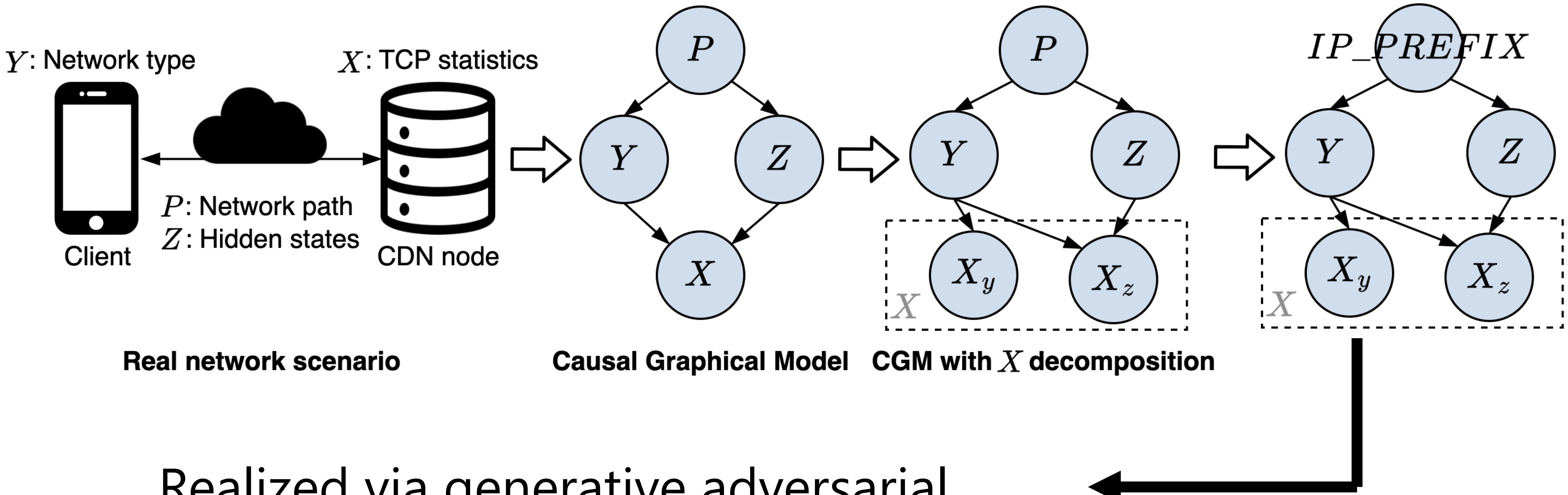
Model design: key intuition

- Representation invariant to /24 IP prefix (thus invariant to hidden states)



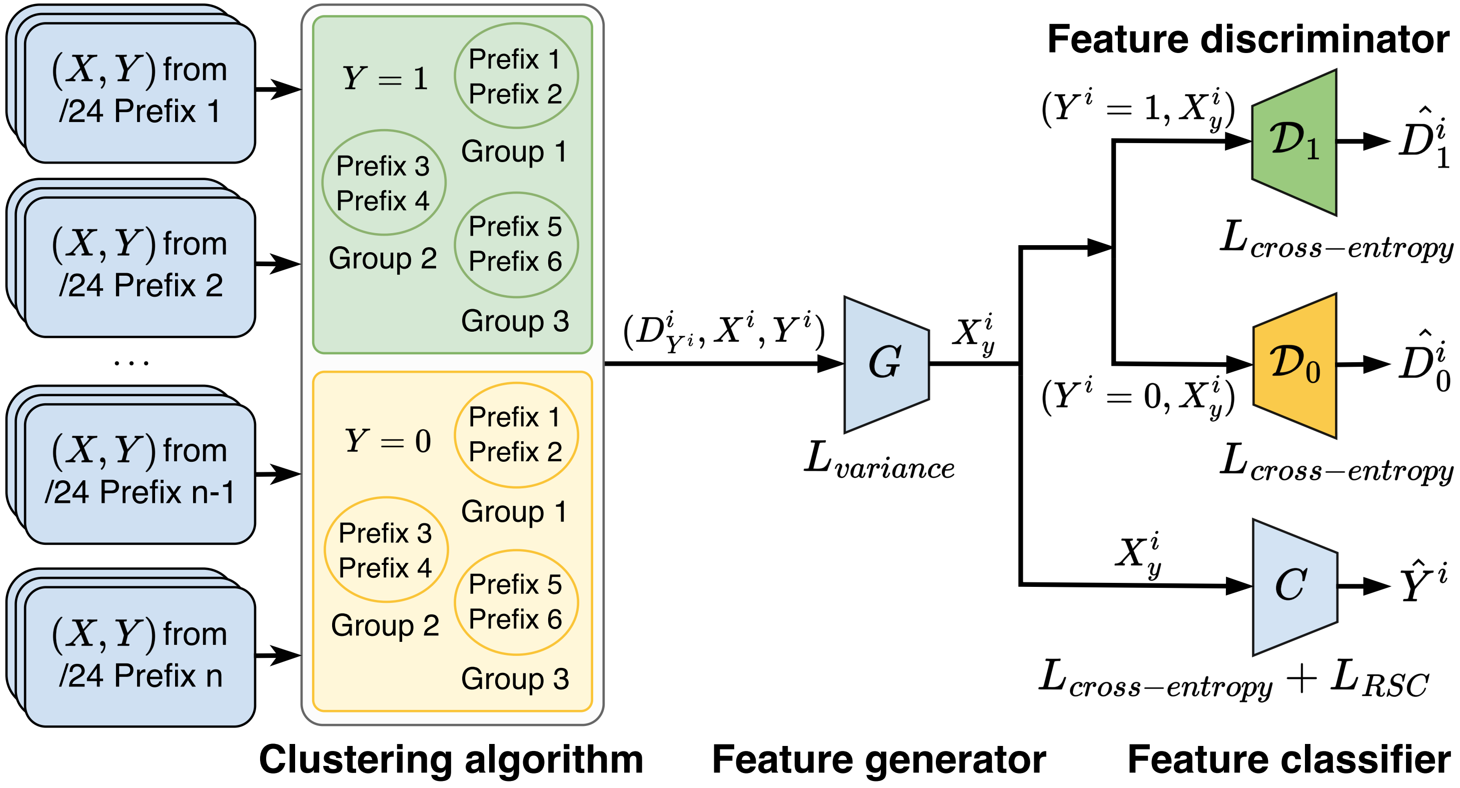
Model design: key intuition

- Representation invariant to /24 IP prefix (thus invariant to hidden state)



Realized via generative adversarial network(GAN)

Model design: GAN-based realization



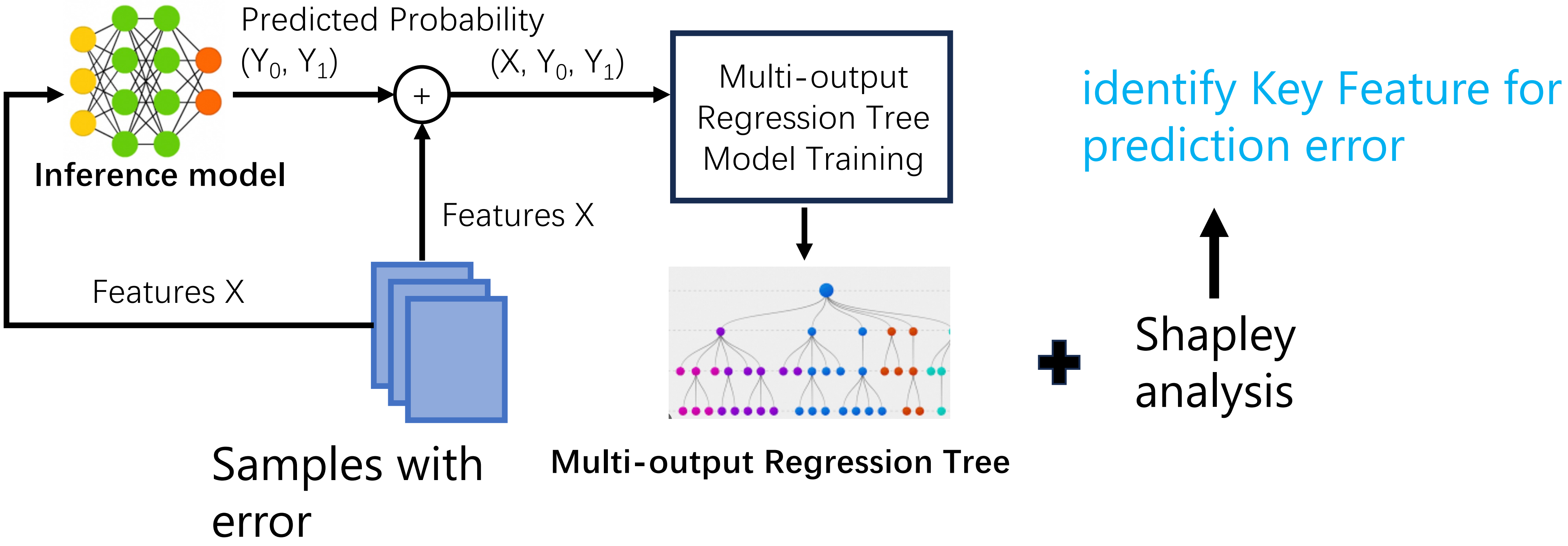
1. Model Scalability and generalization

2. Model prediction interpretability

3. Model inference overhead

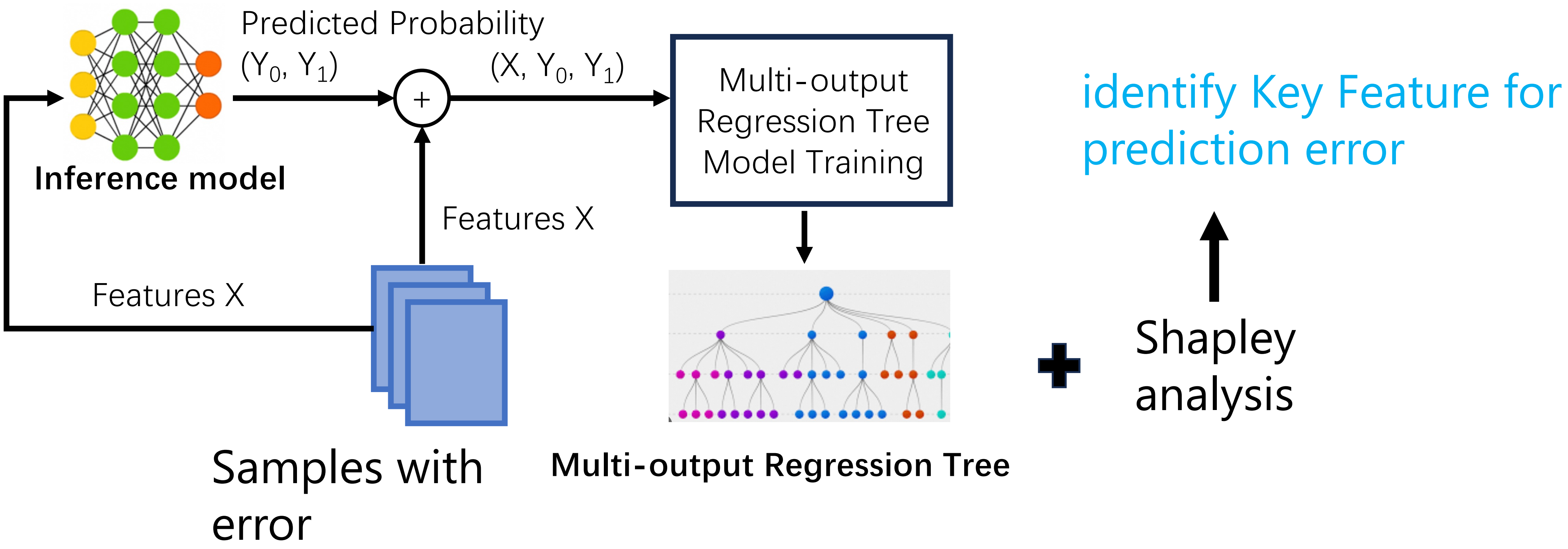
Interpretability design: distillation

- Distillation to decision trees



Interpretability design: distillation

- Real example: low accuracy for small ISP
  - Identified over-reliance on TCP MSS
  - Reduced MSS reliance ➔ boost accuracy.



1. Model Scalability and generalization

2. Model prediction interpretability

3. Model inference overhead

Key observations to reduce overhead

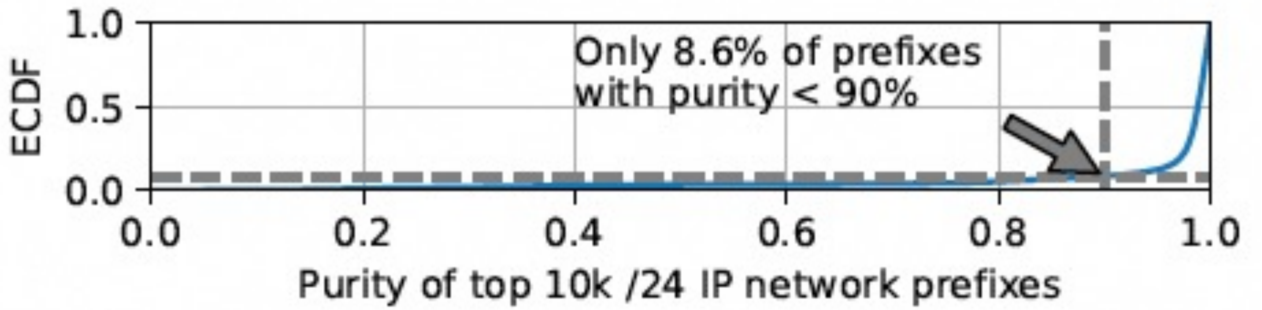
Consistent CCS in temporal stability

a /24 IP prefix dominated by either Wi-Fi or 4G for more than 2 hours with 87% probability

CCS stays the same for hours

Save CPU: Cache inference results for specific paths for hours.

Consistent CCS in IP prefix similarity

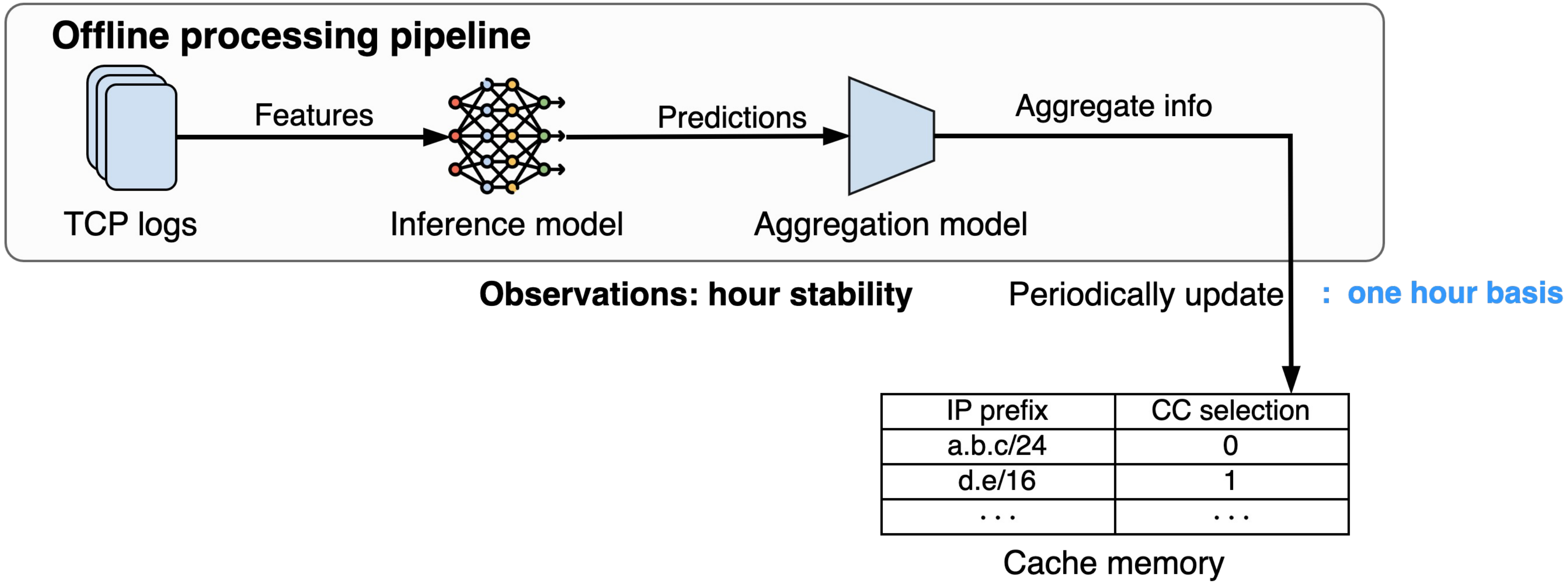


CCS stays the same under /24 prefix

Save Memory: aggregate cached results under the same IP prefix into a single entry

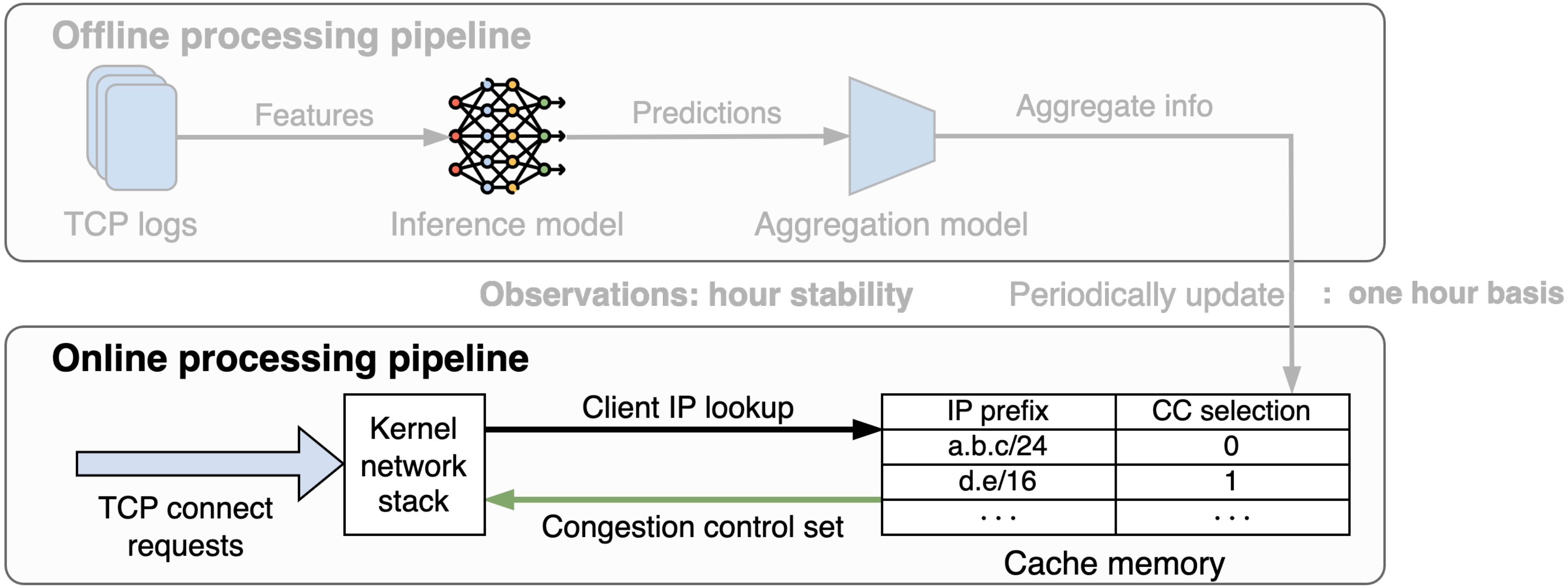
AliCCS design: reduce inference overhead

- Inference & save in cache



AliCCS design: reduce inference overhead

- Search in cache, avoiding sequential inference



Inference efficiency

- Comparison with Non-optimized online inference

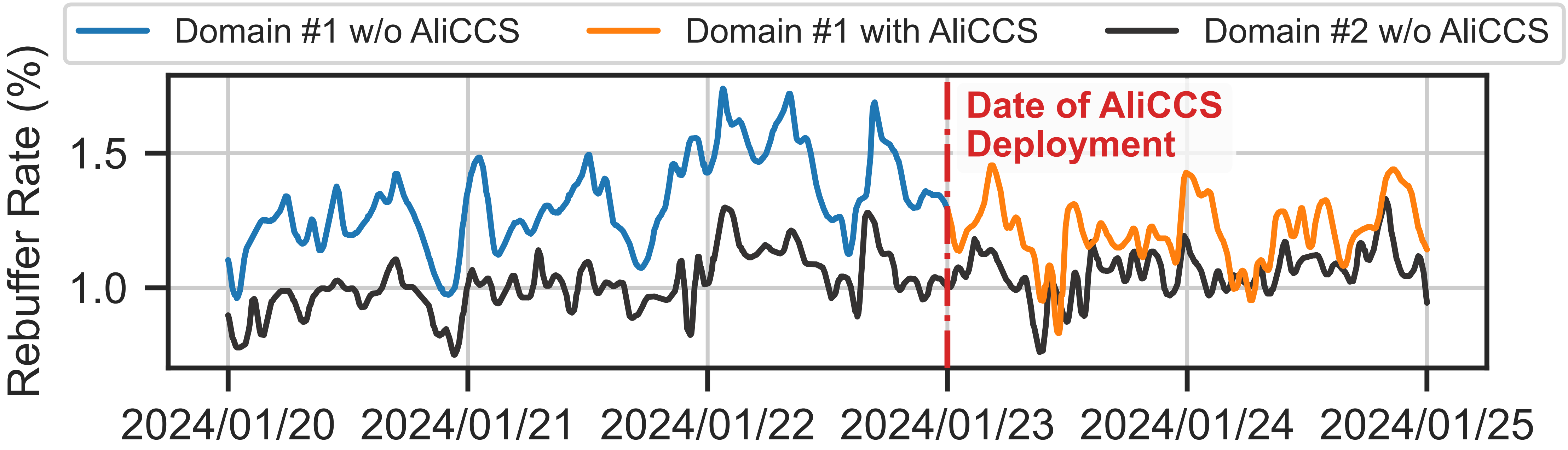
	Processing delay	Max. QPS
Baseline: online inference	10417 ns	7.6k
Online-offline decoupled	162 ns	18.4k
Improvement	64.30×	2.42×

Evaluation: real-world deployment

- 400 nodes, almost all provinces in China & southeast Asia
- Compare AliCCS vs baseline (statically use Cubic)

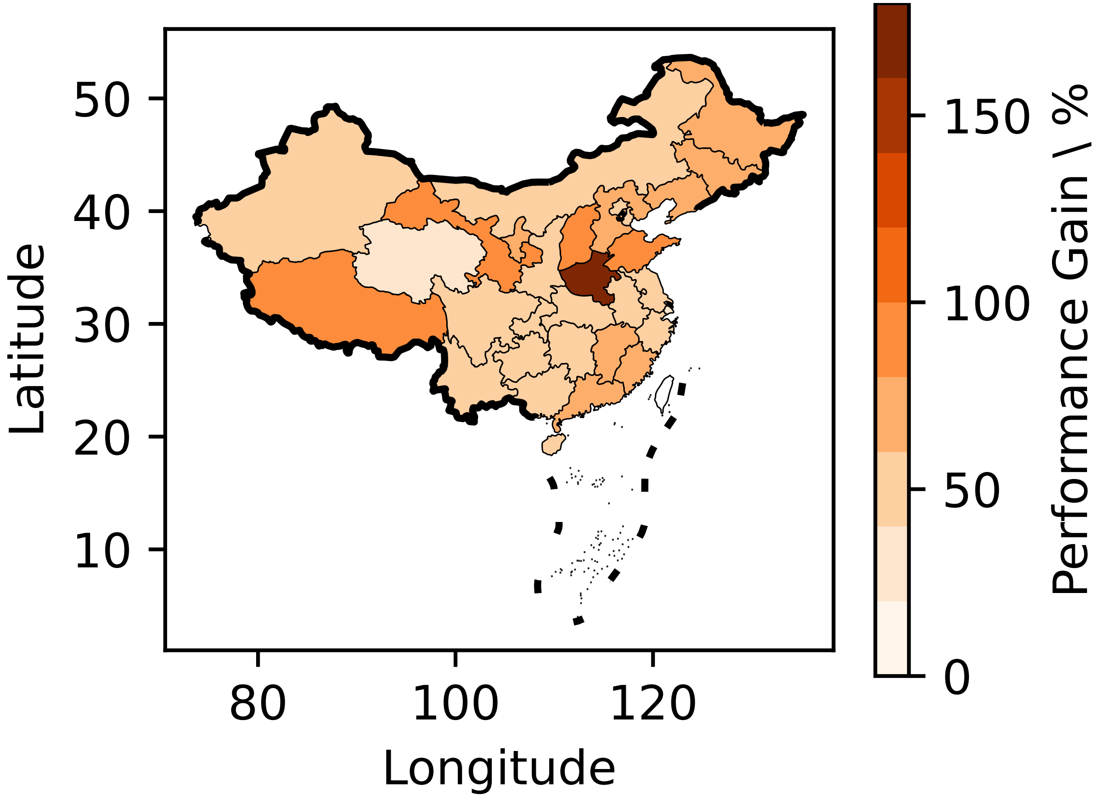
Evaluation: real-world deployment

- Avg. rebuffer-rate reduction of 4.9%
- 2%–3% reduction can ensure customer retention



Evaluation: real-world deployment

- Retransmission rate reduction of 25.5%–174.3%
- Avg. reduction of 62.7%

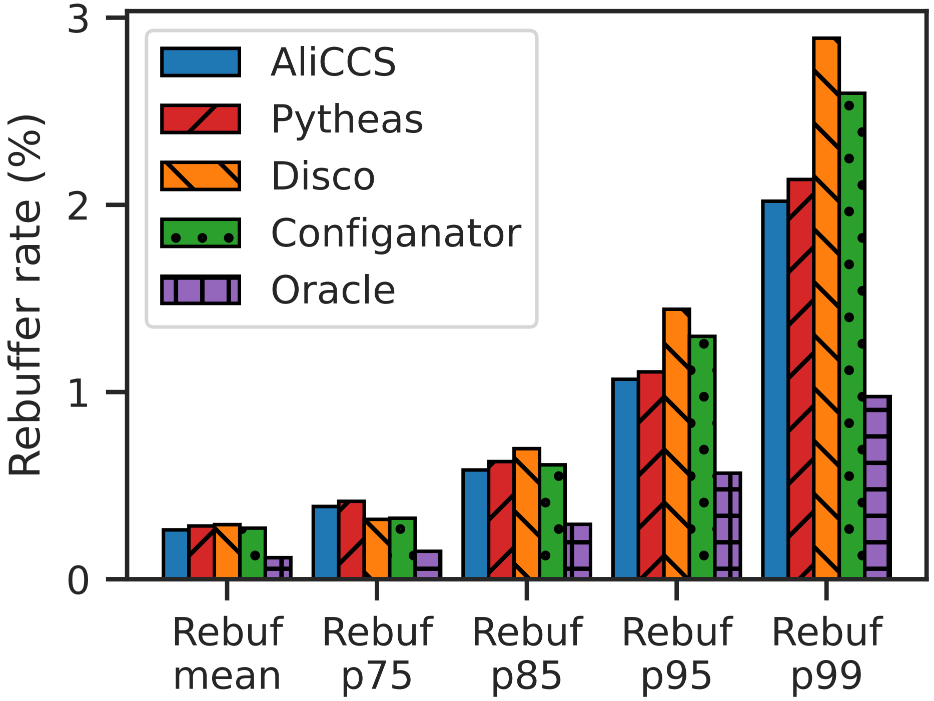


Evaluation: trace-driven comparison

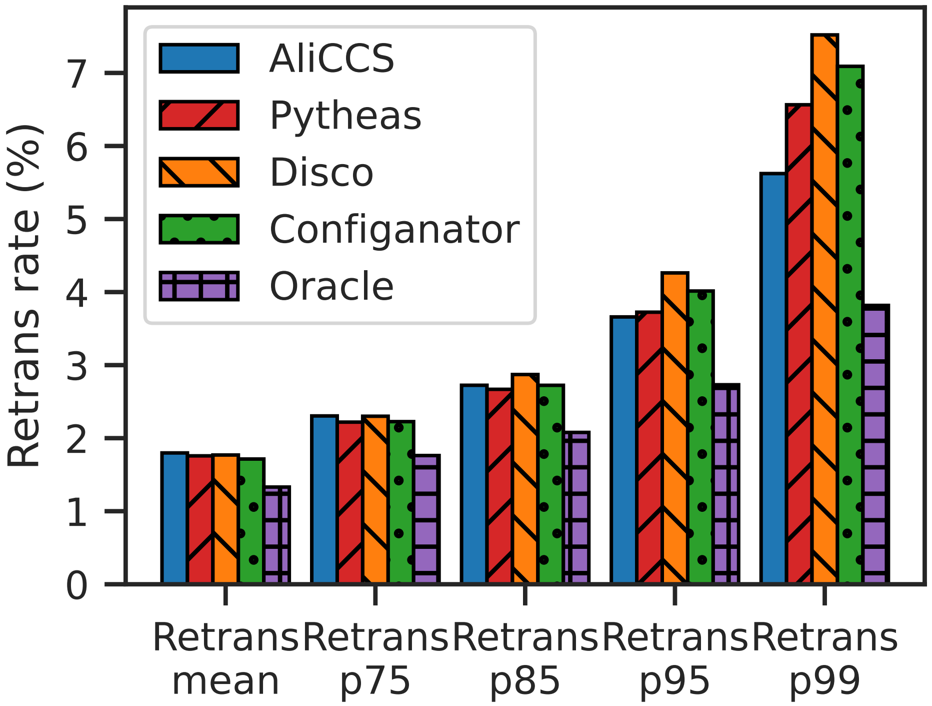
- Compare with state of the art
- List of baselines:
  - Pytheas: online learning
  - Disco: tree-based ML model
  - Confignator: tree-based ML model + bayeasian optimization
  - Oracle: theoretical upper bound

Evaluation: trace-driven comparison

All similar in low percentile (good network condition)

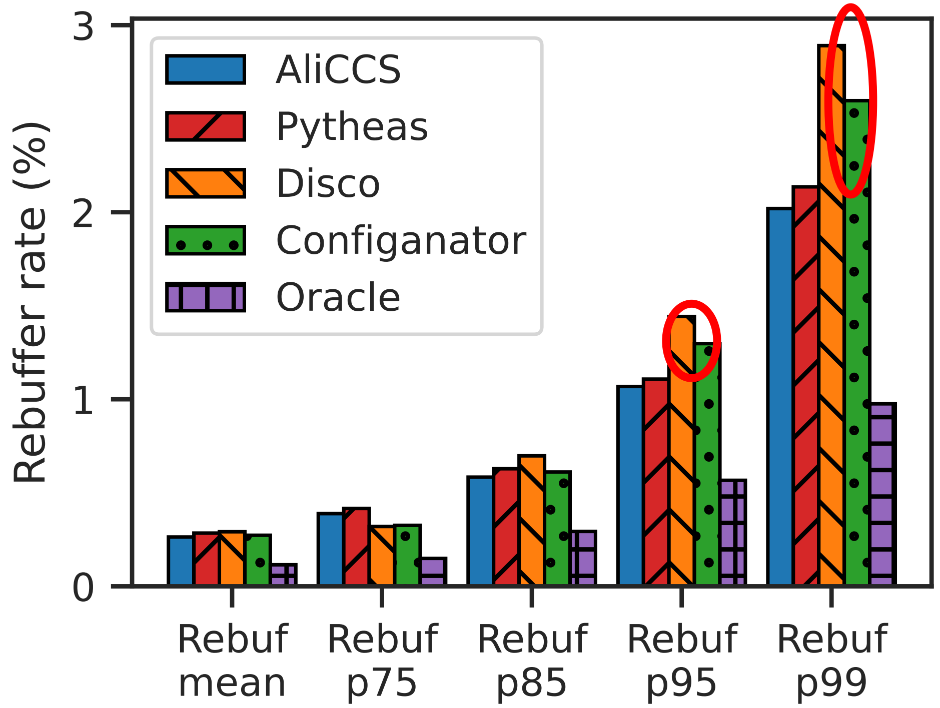


Rebuffer-rate

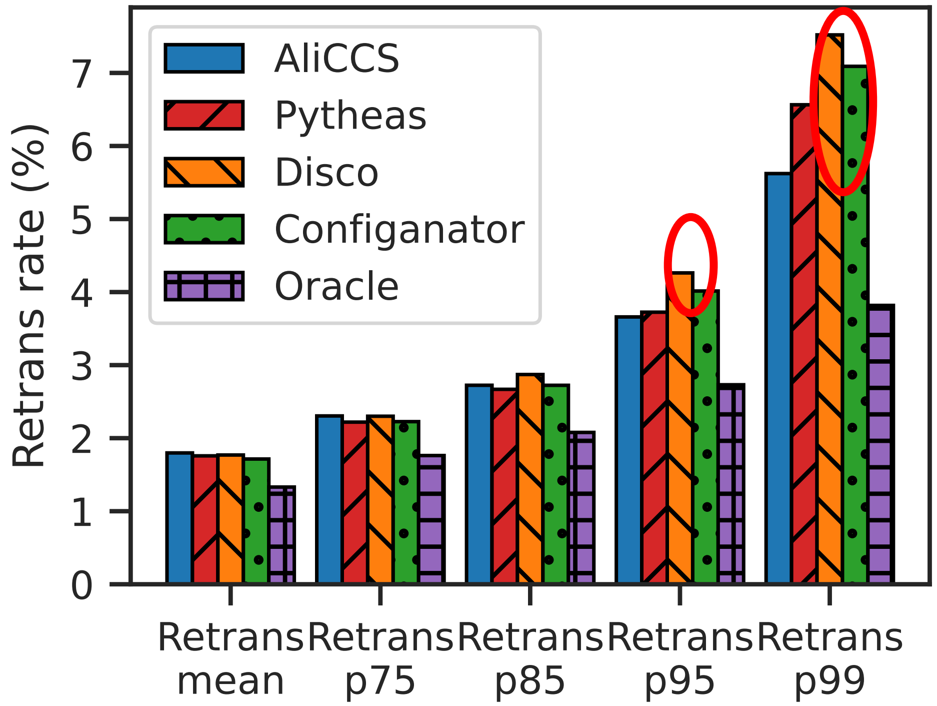


Retransmission rate

Evaluation: vs Disco & Confignator

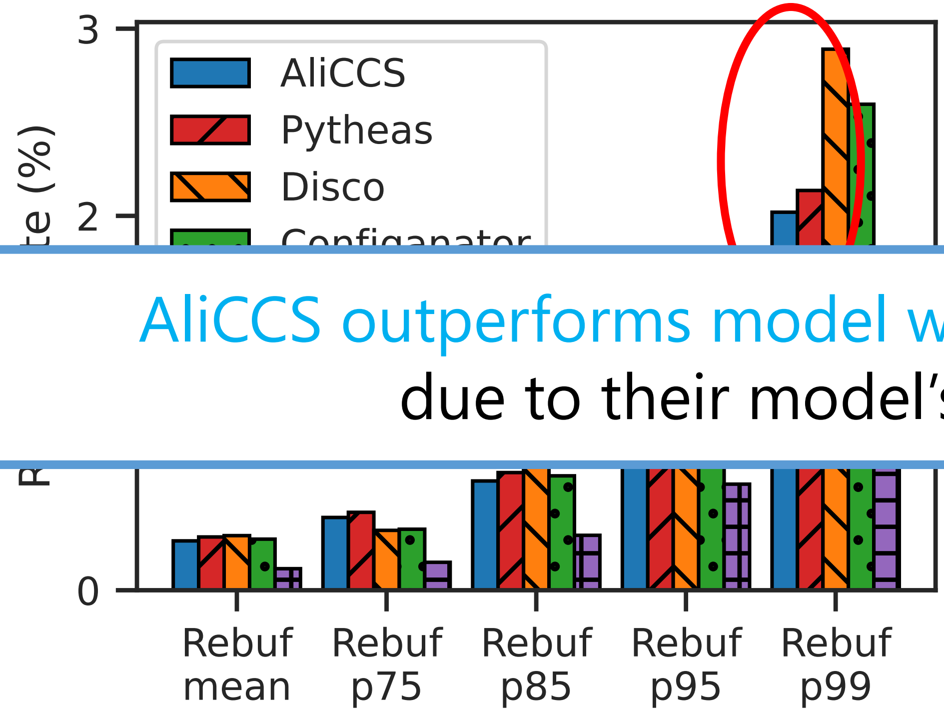


Rebuffer-rate

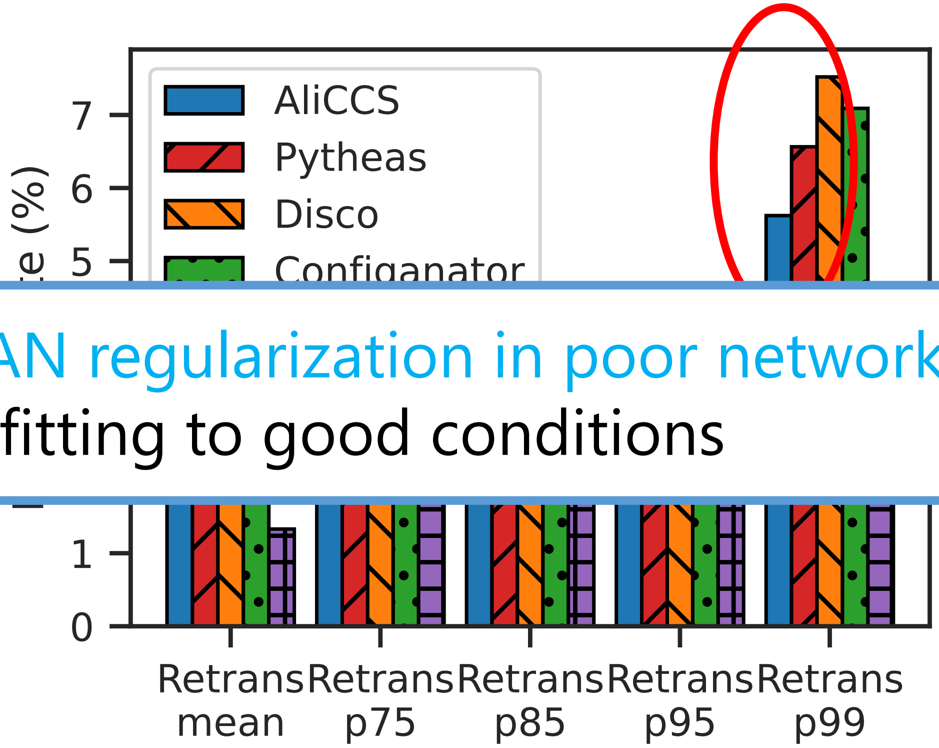


Retransmission rate

Evaluation: vs Disco & Confignator



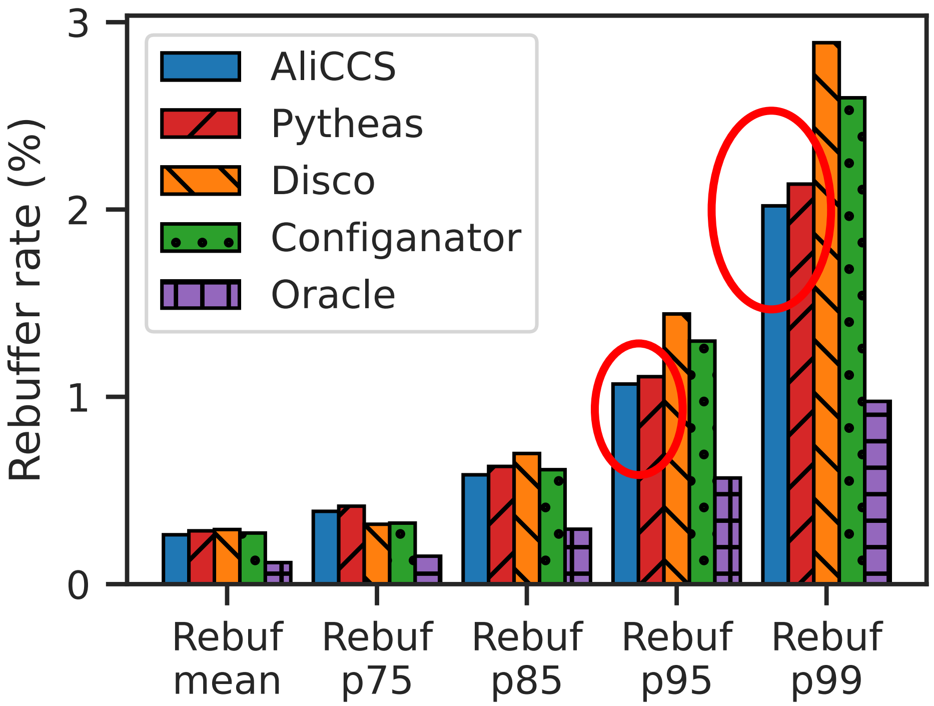
Rebuffer-rate



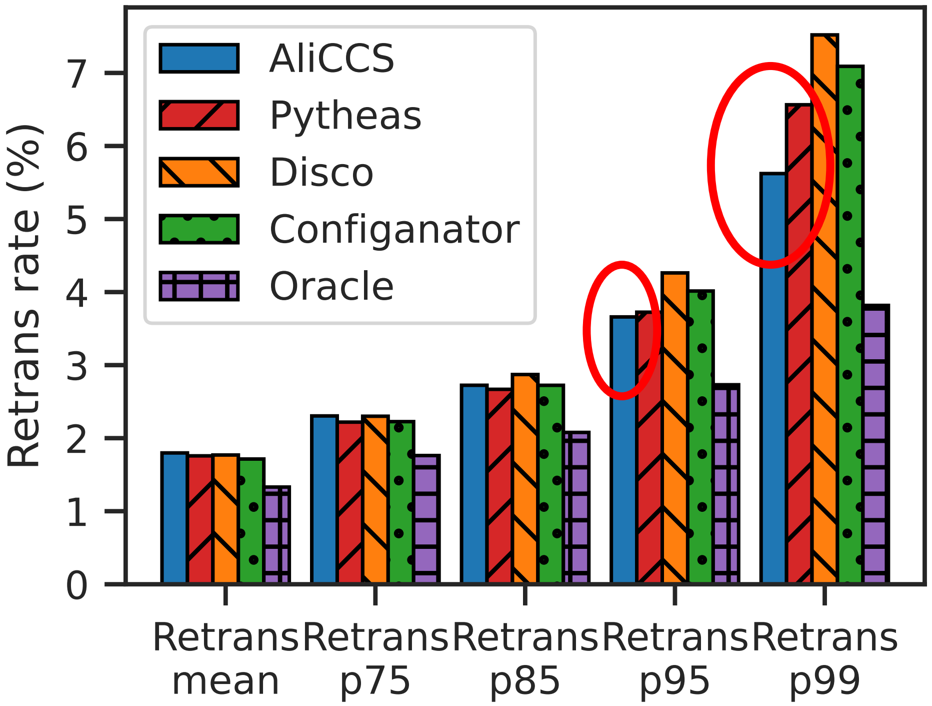
Retransmission rate

AliCCS outperforms model with GAN regularization in poor networks,
due to their model's overfitting to good conditions

Evaluation: vs Pytheas

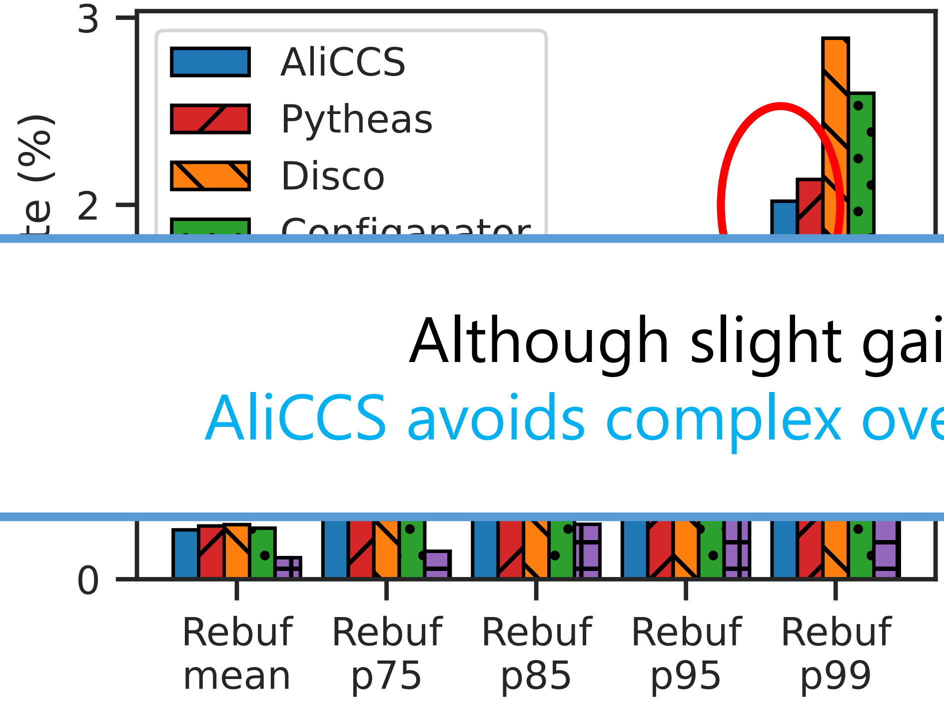


Rebuffer-rate

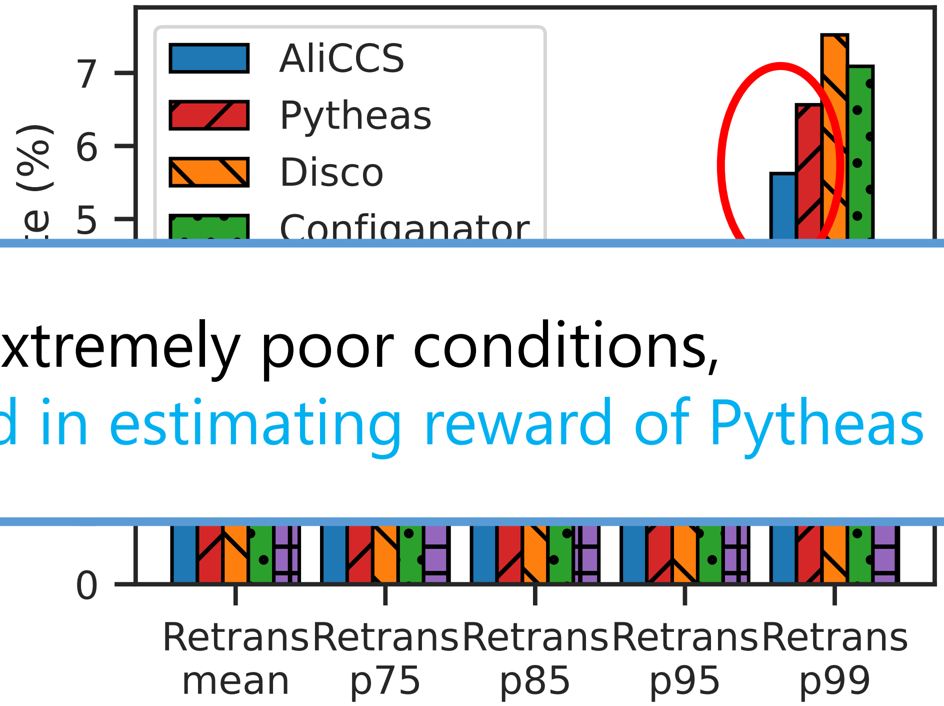


Retransmission rate

Evaluation: vs Pytheas



Rebuffer-rate



Retransmission rate

Although slight gain in extremely poor conditions,
AliCCS avoids complex overhead in estimating reward of Pytheas

Lessons learned

- Maintain throughput stability: Crucial for short video CC design
- Deploy a fallback strategy: helpful to avoid degrading below default configuration in low-confidence scenarios
- Expect higher gain in IPv6: enhanced CCS performance in IPv6 observed