Bridging the Gap between QoE and QoS in Congestion Control: A Large-scale Mobile Web Service Perspective

Jia Zhang¹, Yixuan Zhang¹, Enhuan Dong¹, Yan Zhang¹, Shaorui Ren¹, Zili Meng¹, Mingwei Xu¹, Xiaotian Li², Zongzhi Hou², Zhicheng Yang², Xiaoming Fu³

¹Tsinghua University  ²Meituan Inc.  ³University of Goettingen
Background
Gap between QoE and QoS

Applications value Quality of Experience (QoE).
- Request Completion Time
- Image Quality

Current CCAs optimize Quality of Service (QoS).

Latency
- Copa
- Swift

Loss
- BBR
- Vivace

Throughput
- Cubic
- Westwood
From the perspective of a large-scale mobile web service.

In this paper, we take request completion time (RCT) as QoE.
Motivation
Optimizing QoE for CCAs is challenging.

Convoluted relationship between QoS and QoE.

- QoE: User Experience, e.g. RCT, PLT
- QoS: Transport Capacity, e.g. RTT, Thpt., Loss

Optimal QoE ≠ QoS

What should CCA optimize towards?
Motivation
Optimizing QoE for CCAs is challenging.

Mismatched timescale between QoE and QoS.

▸ QoS-oriented CCAs
  ▸ fine-grained ACK information
  ▸ packet-level or RTT-level

▸ QoE:
  ▸ coarse-grained application metrics
  ▸ request level

How should CCA use QoE?
Insight
QoE-oriented CCA Selecting Mechanism

Always use the best one!

- Optimize the real goal.
- Match the time scale.
Design
Floo: QoE-oriented CCA Selecting Mechanism

Key Questions:
• How to select the best CCA for QoE?
  • CCA Selection Policy
• How to switch between CCAs without traffic interruption?
  • CCA Switching on the Fly
Design
Floo: QoE-oriented CCA Selecting Mechanism

Key Questions:
• How to select the best CCA for QoE?
  • CCA Selection Policy

• How to switch between CCAs without traffic interruption?
  • CCA Switching on the Fly
Design
CCA Selection Policy

Input:

▸ Application requirements and patterns  what app wants and behaves
▸ Network conditions  how network performs
▸ CCA characteristics  which aspect CCA prefers

Output: One of CCA candidates.

Challenge: Time-varying & Complex
Design
CCA Selection Policy

- Application requirements and patterns  
  *what app wants and behaves*

- Network conditions  
  *how network performs*

**Challenge: Time-varying & Complex**

- Response completion time
- Unsent size
- Current waiting time

- Bottleneck Bandwidth
- Packet Loss
- Round-trip Time

......
Design
CCA Selection Policy

- Application requirements and patterns
  *what app wants and behaves*

- Network conditions
  *how network performs*

Solution: monitor **both** and pre-process them!

- **App states**
  - Response completion time
  - Unsent size
  - Current waiting time

- **Network states**
  - Bottleneck Bandwidth
  - Packet Loss
  - Round-trip Time

**Features**
- min RTT, smooth RTT, RTT variance, ...
Design
CCA Selection Policy

App states ➔ Monitor ➔ Features ➔ Selector ➔ CCA

Network states

- CCA characteristics
  which aspect CCA prefers

Challenge: how to quantify CCAs’ preferences over different metrics?
Solution: Use Reinforcement Learning (RL) to select CCAs!

- Neural networks learn the implicit preferences of CCAs.
- End-to-end training towards QoE directly improves the performance
Key Questions:
• How to select the best CCA for QoE?
  • CCA Selection Policy

• How to switch between CCAs without traffic interruption?
  • CCA Switching on the Fly
Design
CCA Switching on the Fly

Challenge: How should we switch the CCA without interruption?

- Longer convergence time and performance deterioration.
Challenge: How should we switch the CCA without interruption?

- Longer convergence time and performance deterioration.
- Distorted path estimation results in abnormal behavior of new CCAs.

![Graphs showing CWND and RTT over time for Cubic and Copa, indicating a wrong decision with much larger CWND and queue loss event.]
Challenge: How should we switch the CCA without interruption?

- Target
  - Inherit the network path (fast and safe CCA convergence)
  - Retain the CCA characteristics (consistent with the original design goals)
Solution: Phase Migration

- Converged phase: CCA is confident about the current path condition and sending traffic now.
- Non-converged phase: CCA is not confident about the current path condition and not sending traffic with full speed.
Solution: Phase Migration

• Converged phase:
  • Floo directly enters the converged phase for the new CCA.
  
  But wait... what should the parameters (e.g., cwnd) be set?

• Non-converged phase:
  • Floo does not switch.
Design
CCA Switching on the Fly

Solution: Phase Migration + Variable Migration

• Sending rate variables:
  • CWND, pacing rate, etc.
  • Floo maps them with $CWND = \text{pacing rate} \times RTT$.

• Observation variables:
  • BtlBw, RTT, etc.
  • Floo preserves $BtwBw, RTT, \text{and loss}$ for all CCAs even if they do not require.

• Parameter variables:
  • Multiplicative-decrease factor during loss (Cubic), pacing gain (BBR), etc.
  • Floo does not change them.
Implementation

Put everything together...

Application Layer
  Application Status

Monitor

Metrics

Selector

Selected CCA

Switcher

CCA-related variables

Running CCA
  (Cubic, BBR, ...)

Conn Status

Flooo

Transport Layer (QUIC)
Evaluation
Experiment Setup

• Dianping, an mobile phone app with O(10M) daily active users.

• Different OS, HTTP versions, etc.

• A/B tests for 4 days with a fraction of users (5%), with >10M request logs.

• CCA candidates: Cubic, BBR, Copa, Westwood, and Vivace.
Evaluation
Large-scale Production Deployment

- Application performance – request completion time (RCT)

- 14% reduction on average
- 25% reduction at P99
Evaluation
Fine-grained Analysis

• Transport performance – throughput / latency
  • We further analyze 60 sets of traces for finer-grained transport layer metrics

(a) Stationary cellular scenarios.

(b) Highly variable scenarios.
Effectiveness of state migration

- Fast - Converge duration 2.1s -> 0.6s
- Safe - Loss rate ↓
- CCA consistency
- Effective – avg RCT 7%↓
Takeaway

- CCAs might be on the *Pareto-optimal frontier of QoS*, but different CCAs win in different scenarios *in terms of QoE*.
- Always selecting the best CCA can improve the QoE for applications.
- Floo monitors both network and application metrics, selects the best CCA with reinforcement learning, and ensures CCA switching consistency.
- Large-scale production deployment shows *14% improvement on QoE (request completion time)*.
Thank you!
jia-zhan18@mails.tsinghua.edu.cn

Bridging the Gap between QoE and QoS in Congestion Control: A Large-scale Mobile Web Service Perspective