Q2oE: Balancing QoE Fairness and Preference for Video Streaming from Cooperative Edge Servers

Xinwei Huang[1] Yin Xu[1] Jinbo Cai[1] He Sun[1] Jie Wu[2] Mingjun Xiao[1]

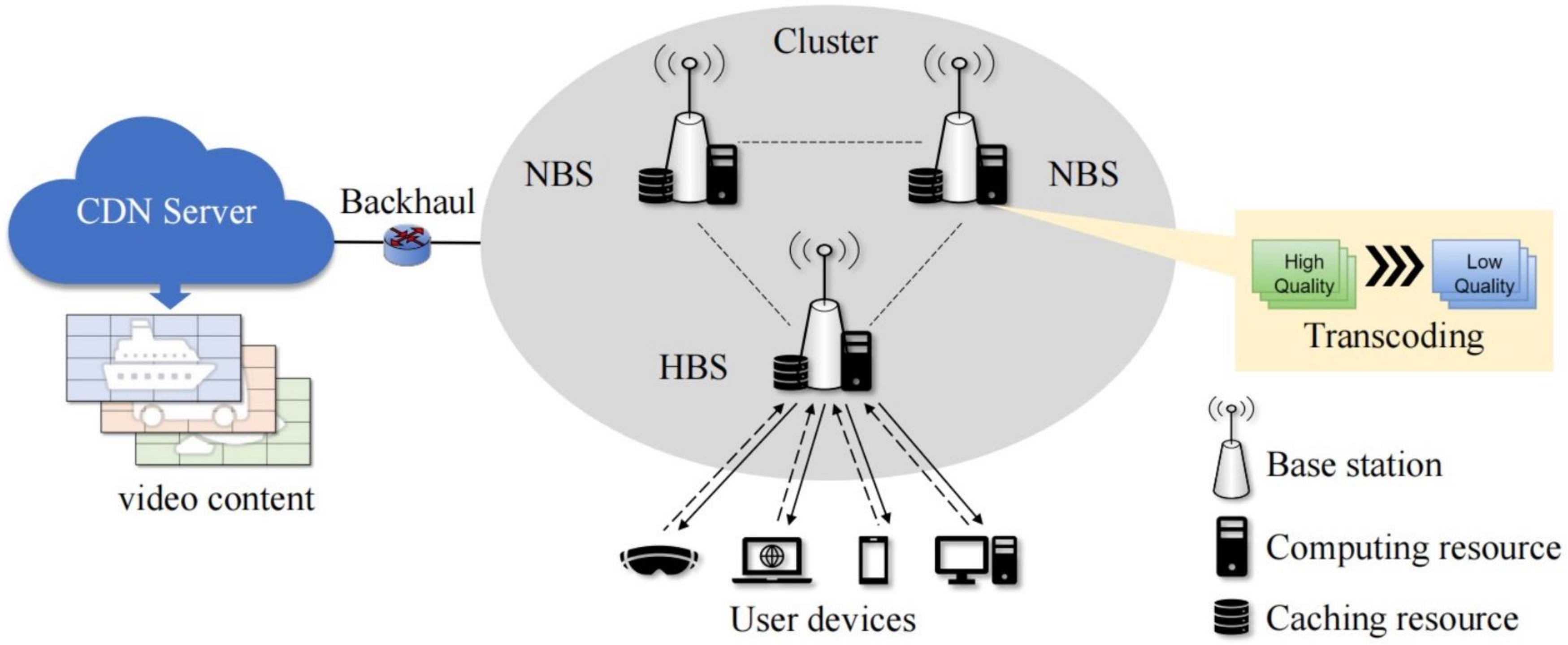
[1]University of Science and Technology of China

[2]Temple University

Outline

- Background & Motivation
- System Model
- Reinforcement Learning
- Evaluation & Conclusion

Background & Motivation



- Cache capacity is limited
- Bandwidth is limited
- Computational capacity is limited

- High video bitrate
- Minimal transmission delay
- Low rebuffering time
- Low frequency of bitrate variation

Background & Motivation

Motivation

- Traditonal QoE Models targets the "average user", thereby neglecting the individuality of each user;

- Traditional ABR algorithms are also agnostic to user preferences, because they usually optimize towards a fixed QoE model.

The differences between user preferences cannot be ignored, and the "average user" cannot represent all users.

Key Challenges

- **User Motivation**

Environmental factors also need to be considered, such as watching videos at home, watching videos while walking, or watching videos while driving. User QoE perception is different in these scenarios.

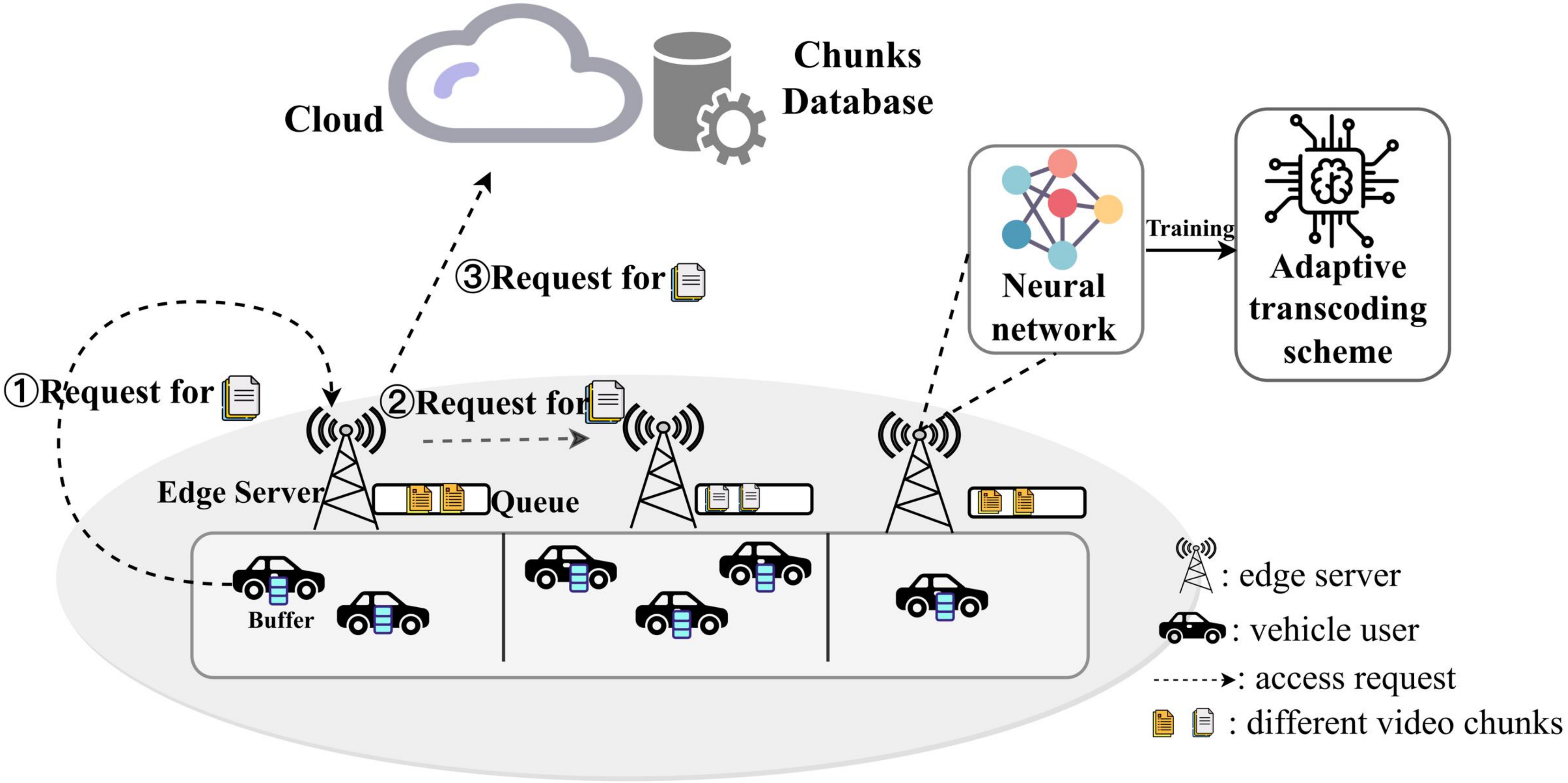
- **Collaboration between edge servers**
  - Collaborative caching
  - Collaborative transcoding

- **Preference and fairness modeling**

Outline

- Background & Motivation
- System Model
- Reinforcement Learning
- Evaluation & Conclusion

System Model



System Model

- **User-level QoE preference:**

$$P_i = p_i - \gamma * v_i$$

p_i represent the user $i's$ initial QoE preference, and P_i represent the user $i's$ true QoE preference.

p_i represents the user's preferred video chunk bitrate and latency tolerance.
γ is a hyperparameter.

- **QoE Model:**

$$QoE = Q_0 - \omega * I_c - \gamma * v - \delta * d$$

$$Q_0 = \max\left(1, \min\left(5, 1 + 4 \cdot \frac{c_1 \cdot \mathcal{V}_k}{c_2 + \mathcal{V}_k}\right)\right)$$

Q_0 is modeled with a Michaelis Menten function.
V_k is the bitrate level of the k^{th} video chunk, and c_1 and c_2 are the model parameters.

I_c represents fluctuations in the quality of continuous video chunks watched by the vehicle user. v stands for the vehicle's ability to move;
d stands for transmission delay;

System Model

$$QoE = Q_0 - \omega * I_c - \gamma * v - \delta * d$$

$$I_{c,k} = \frac{max(V_{k-1} - V_k, 0)}{V_k} * Q_{0,k}$$

The end-to-end latency for users to fetch the same bitrate video chunks from the edge server, neighbor edge server, and the cloud are randomly assigned following a uniform distribution within the range of [5, 10] (ms), [20, 50] (ms), and [100, 200] (ms)

The difference between the bitrate of the $(k-1)^{th}$ video chunk and the k^{th} video chunk represents the oscillation of the quality of two consecutive video chunks.
Here, only the downward fluctuation in video block quality will have negative effects, while an increase in bitrate often leads to positive effects.

System Model

- **Mobility Model:**

We use ρ to represent the probability that the vehicle will move to the next edge server, and $(1-\rho)$ to indicate the probability that the vehicle user does not move out of the current edge server.

$$\rho = \frac{1000 * v'}{3600 * R}$$

v' represents the vehicle's speed (e.g., 80km/h) and R represents the range of the edge server (e.g., 200m).

- **Fairness Model:**

$$c_t = \min_{i \in N}(QoE_{i,t}/M)$$

Maximize the lowest QoE value among M video chunks.

System Model

Q2oE Model:

To combine considerations of user-level preference and QoE fairness, we must formulate an optimization object to find the trade-off point.

$$max \sum_{t}^{T} (\alpha * \frac{1}{\sum_{i}^{N} |P_i - QoE_{i,t}|} + \beta * F_t)$$

- $P_i = p_i - v_i;$
- $QoE_{i,t} = Q_0^{i,t} - \omega \times I_c^{i,t} - \gamma \times v^{i,t} - \delta \times d^{i,t};$
- $F_t = c_t - c_{t-1};$
- $\alpha + \beta = 1;$

Outline

- Background & Motivation
- System Model
- Reinforcement Learning
- Evaluation & Conclusion

REINFORCEMENT LEARNING

State Space

- ✓ Location of the user vehicle L(t);
- ✓ Cache status of the edge server C(t);
- ✓ The velocity of the user's vehicle V (t);
- ✓ The QoE preference of the user P(t);

All these states possess the Markov property. Hence, the state can be represented as

$S_t = [L(t), C(t), V(t), P(t)]$;

Action Space

- The transcoding decision T(t) for the video chunks of_x0002_coverage users at the current edge server.
- The transcoding decision T′(t) for the video chunks for the users that are about to enter the coverage range of the current edge server

The action taken at time t can be defined as $A_t = [\mathrm{T(t)}, \mathrm{T'(t)}]$;

REINFORCEMENT LEARNING

Reward

Based on the previously mentioned optimization goals of user-level QoE preference and fairness, we propose the reward equation as:

$$r_{t,i} = \alpha * \frac{1}{\exp(P_i - QoE_{i,t})} * C + \beta * F_t$$

- For edge server j, the reward $r_{t,j}$ for the action it performs is the sum of all user rewards $r_{t,i}$ belonging to edge server j during time slot t.

$$r_{t,j} = \sum_{i}^{N_j} r_{t,i}$$

- The total rewards r_t accumulated across all edge servers.

$$r_t = \sum_{j}^{J} r_{t,j}$$

REINFORCEMENT LEARNING

Algorithm 1 algorithm of Q2D3

Initialize: Initialize each agent's evaluation network Q and target network Q' with random parameters; Initialize the replay buffer M;

```
for episode = 1, 2, ..., E do
    Reset the environment and get an initial observation state s_0;
    for t = 1, 2, 3, ..., T do
        for j = 1, 2, 3, ..., J do
            Cache video chunks based on the user's location and speed information;
            Select actions a_{j,t} based on current state and random exploration probability;
            Executes a_{j,t}, that is, executes corresponding transcoding actions;
        for i = 1, 2, 3, ..., N do
            Move and request video chunks from local edge servers;
            Calculate the value of QoE_{i,t} and the value of F_t, and then get the corresponding reward r_{t,i} according to Eq. (7);
        The environment return the state of the next time slot s_{t+1} and reward r_t according to Eq. (9);
        Store sample (s_t, a_t, r_t, s_{t+1}) in M;
        Sample a random minbatch of K samples from M;
        for j = 1, 2, 3...J do
            Update the evaluation network parameters θ with loss function by Eq. (10);
            Update the target network parameters θ' with loss function by Eq. (11);
```

Caching video chunks based on their popularity, and prioritizing the caching of those with higher bitrate levels whenever possible.

Due to the uncertain number of users within the edge node, we have capped the maximum number of requests to control the size of the action space.

$$L(\theta) = \frac{1}{K}\sum_{i=1}^{K}\left[r_t + \gamma Q'\left(s_{t+1}, a^*; \theta'\right) - Q\left(s_t, a_t; \theta\right)\right]^2$$

$$\theta' = \tau\theta + (1-\tau)\theta'$$

Outline

- Background & Motivation
- System Model
- Reinforcement Learning
- Evaluation & Conclusion

Experimental Settings

System settings

- 3 to 20 users with uniformly distributed QoE preference.
- Five edge servers.

Parameter settings

- E=30000; M=10000;
- Discount factor $\gamma = 0.5$;
- $\tau = 0.005$;
- Two fully connected hidden layers, each with 1024 neurons.

Compared Algorithms

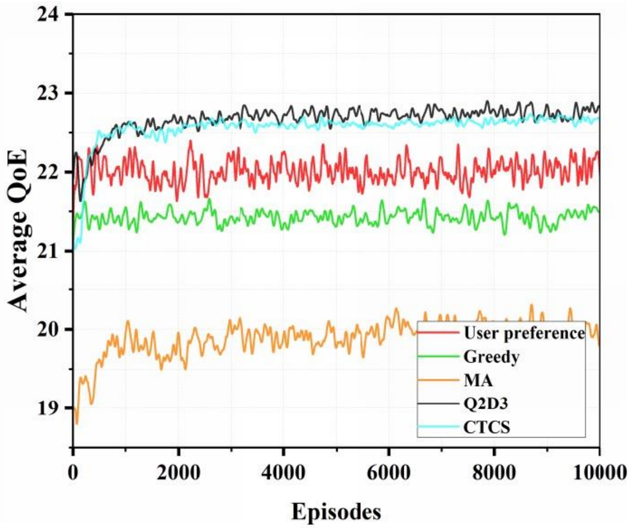
- DRL-CTCS
- Mobile-based algorithm(MA)
- GreedyMSMC

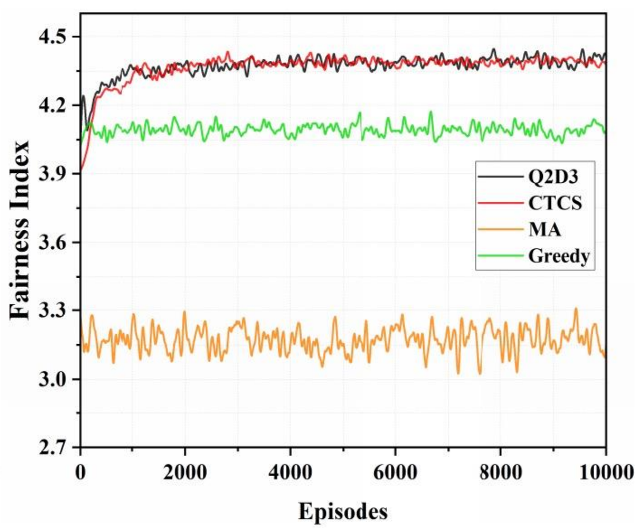
Evaluation Metrics

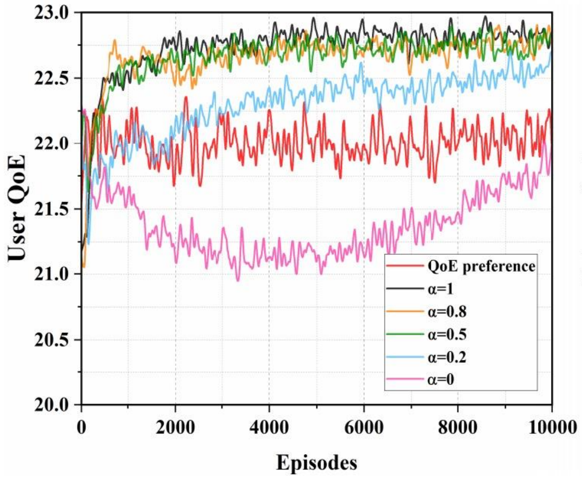
- QoE
- Fairness
- Delay

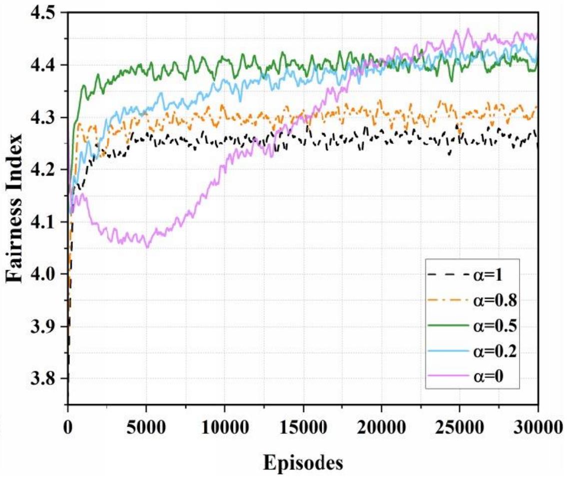
Evaluation

- Evolution of the average QoE and Fairness;





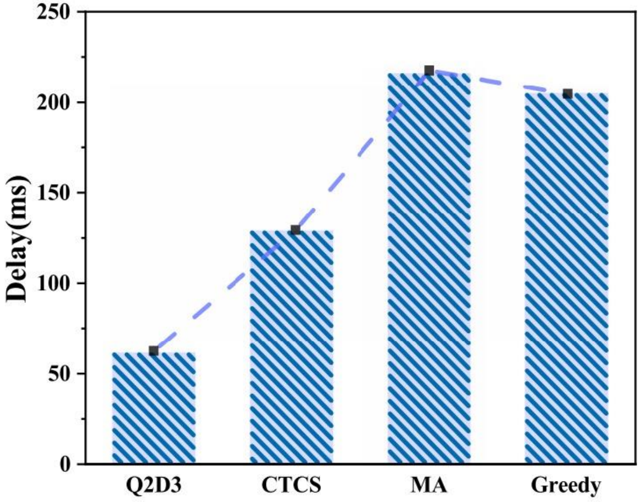


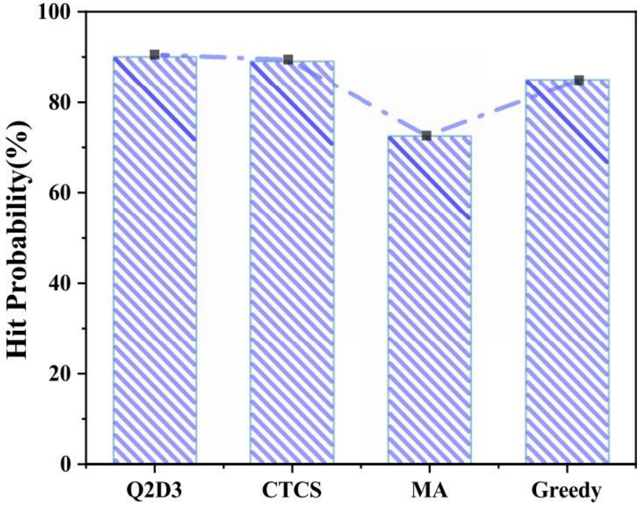


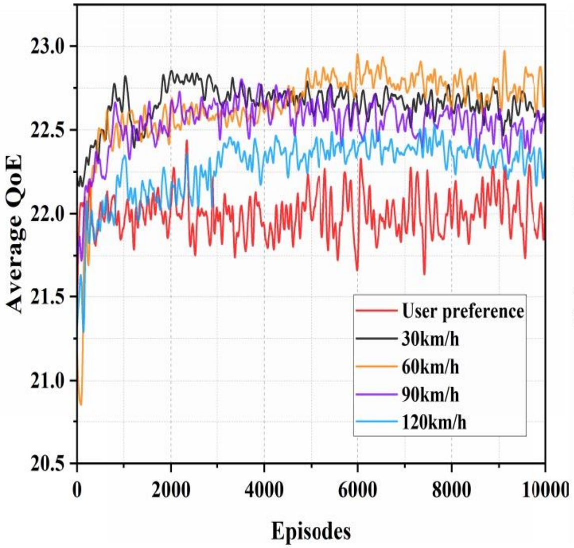
When we fix the $\alpha = 0.5$ and $\beta = 0.5$ in the reward function, the system's QoE and fairness show improvements compared to the other algorithms.

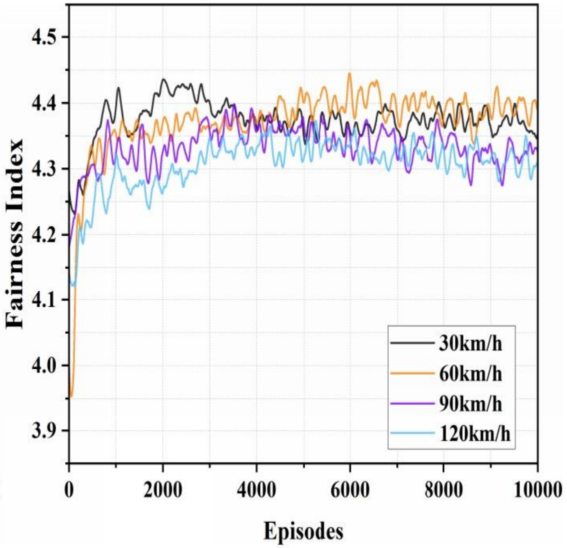
Evaluation

- Evolution of the delay and hit probaility;









Cooperative edge servers enhance both latency and cache hit rates.

The system performs well under various user mobility speeds.

Conclusion

01

- ✓ We propose the Q2oE model, a novel QoE model that combines user QoE preference and fairness.

02

- ✓ We present Q2D3, a reinforcement learning algorithm based on vehicle networks, to implement the Q2oE model.

03

- ✓ When edge server resources are limited, this algorithm can maximize satisfying users' QoE preferences while improving fairness as much as possible.

04

- ✓ We introduce a collaborative edge system where edge servers can collaborate with neighboring servers for caching and transcoding.

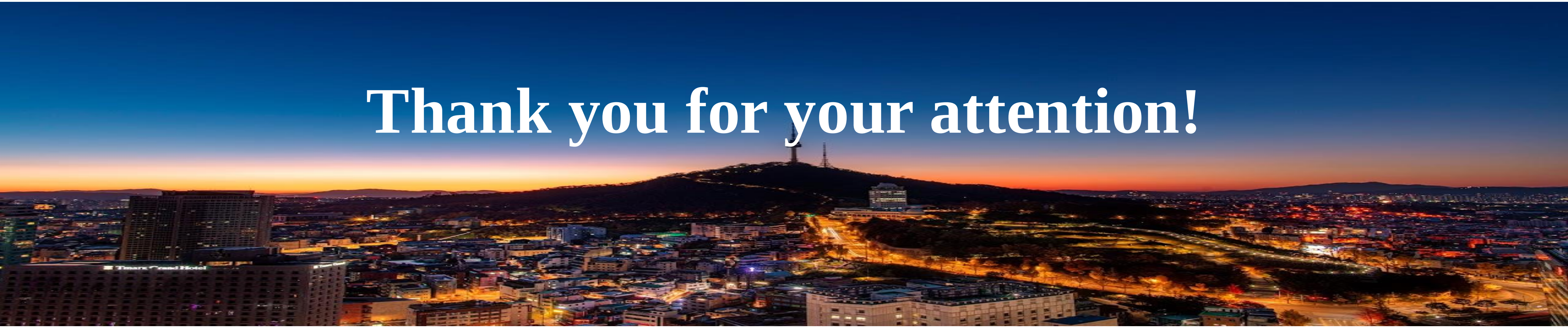
05

- ✓ Extensive simulation experiments have validated its performance.

University of Science and Technology of China

The 21th IEEE International Conference on Mobile Ad-Hoc and Smart Systems

Thank you for your attention!



Question?

Xinwei Huang xinweihuang@mail.ustc.edu.cn