Cooperative Caching for Pub/Sub System in Edge Computing

Tomoya Tanaka, Advisor: Prof. Dr. Tomio Kamada

Kobe Univerisity

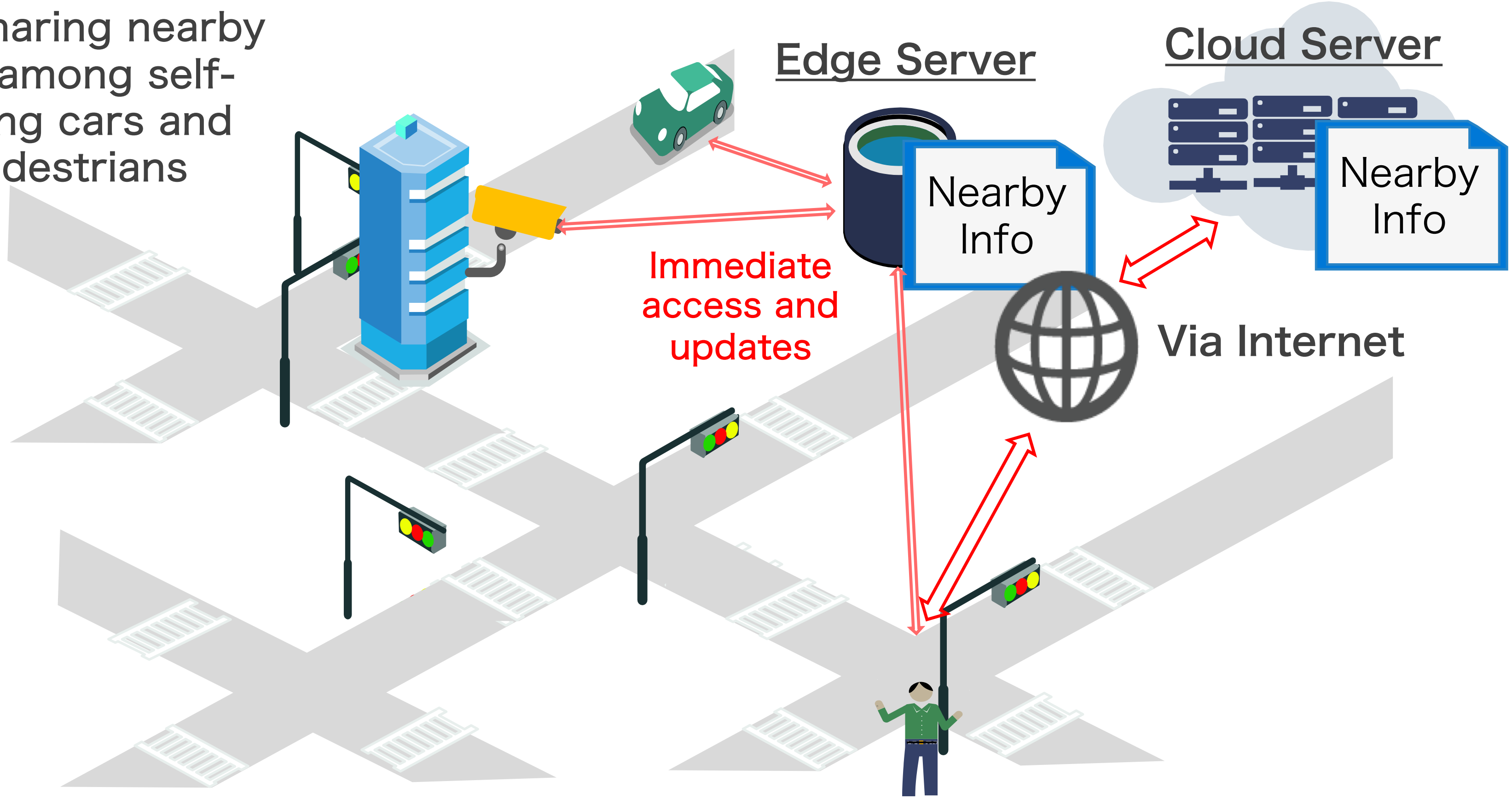
Computer Science and Systems Engineering Department

Introduction

- Edge Computing

  Cache content on geographically dispersed edge servers
  → Access to content with low-latency can be realized

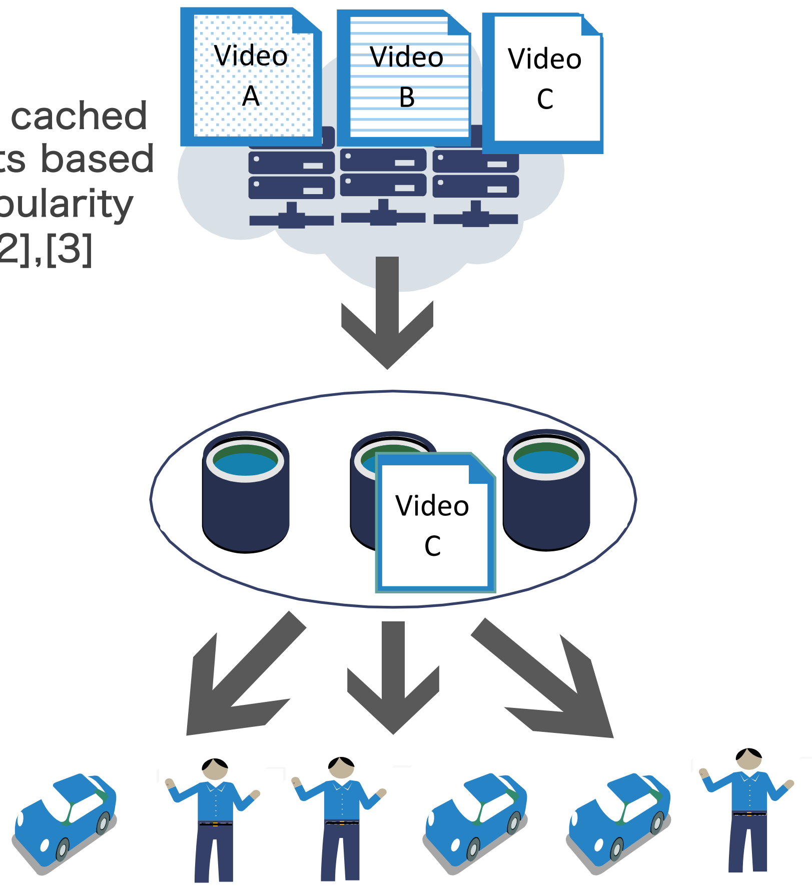
e.g. Sharing nearby info among self-driving cars and pedestrians



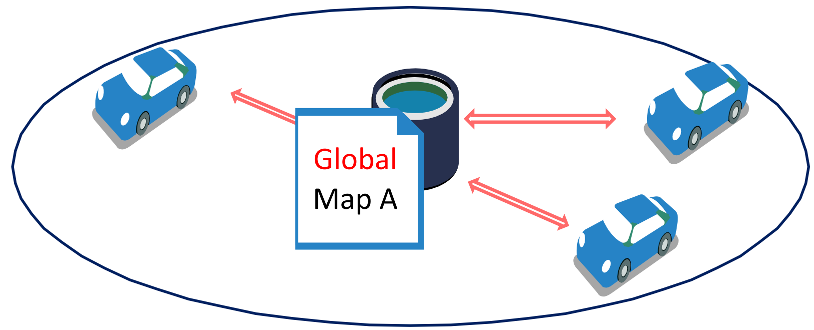
Related Works

Content caching on edge servers → Resource constraints

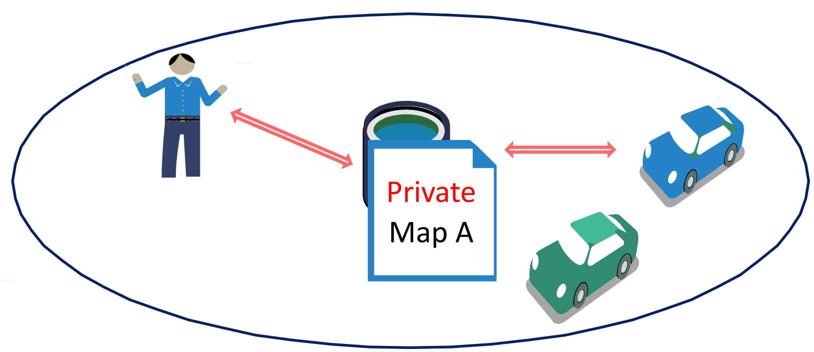
Decide cached contents based on popularity [1],[2],[3]



Locality conscious content shareing



Sharing public contents among nearby clients



Immediate content sharing among specific clients

We realize immediate content sharing among specific clients

Related Works

A system that can transfer content to specific destinations only [5, Nagato 2020]

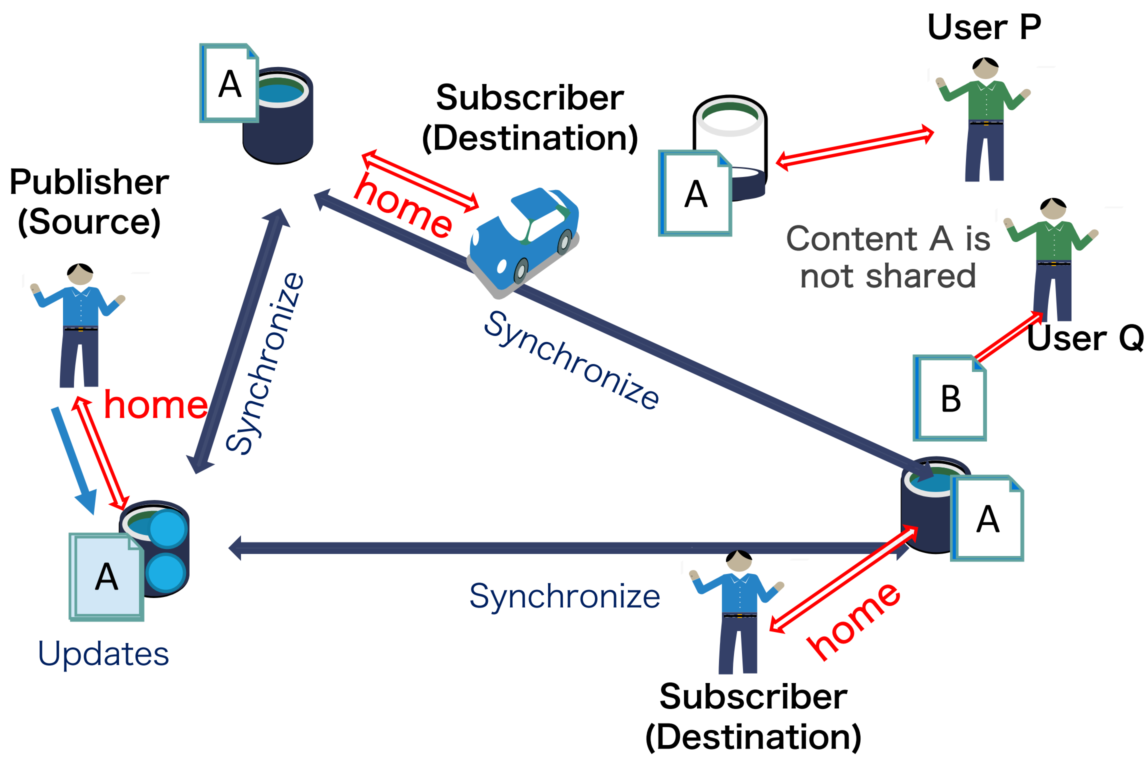
Characteristics

1. **Pub/Sub Distribution using cache**

   The client connects to a fixed server (home server) and performs Publish/Subscribe.

2. **Consistency**

   Synchronize content state between servers



Make content sharing real-time considering resource constraints

Related Works

A system that can transfer content to specific destinations only [5, Nagato 2020]

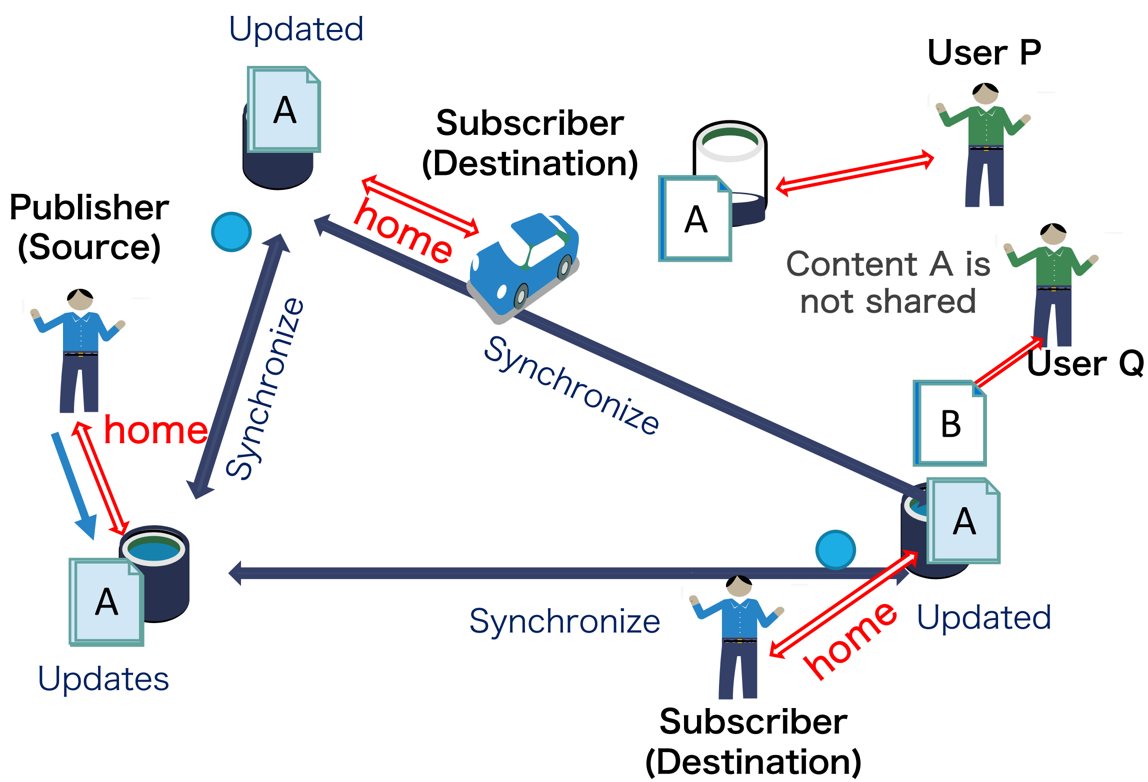
Characteristics

1. **Pub/Sub Distribution using cache**

   The client connects to a fixed server (home server) and performs Publish/Subscribe.

2. **Consistency**

   Synchronize content state between servers



Make content sharing real-time considering resource constraints

Related Works

A system that can transfer content to specific destinations only [5, Nagato 2020]

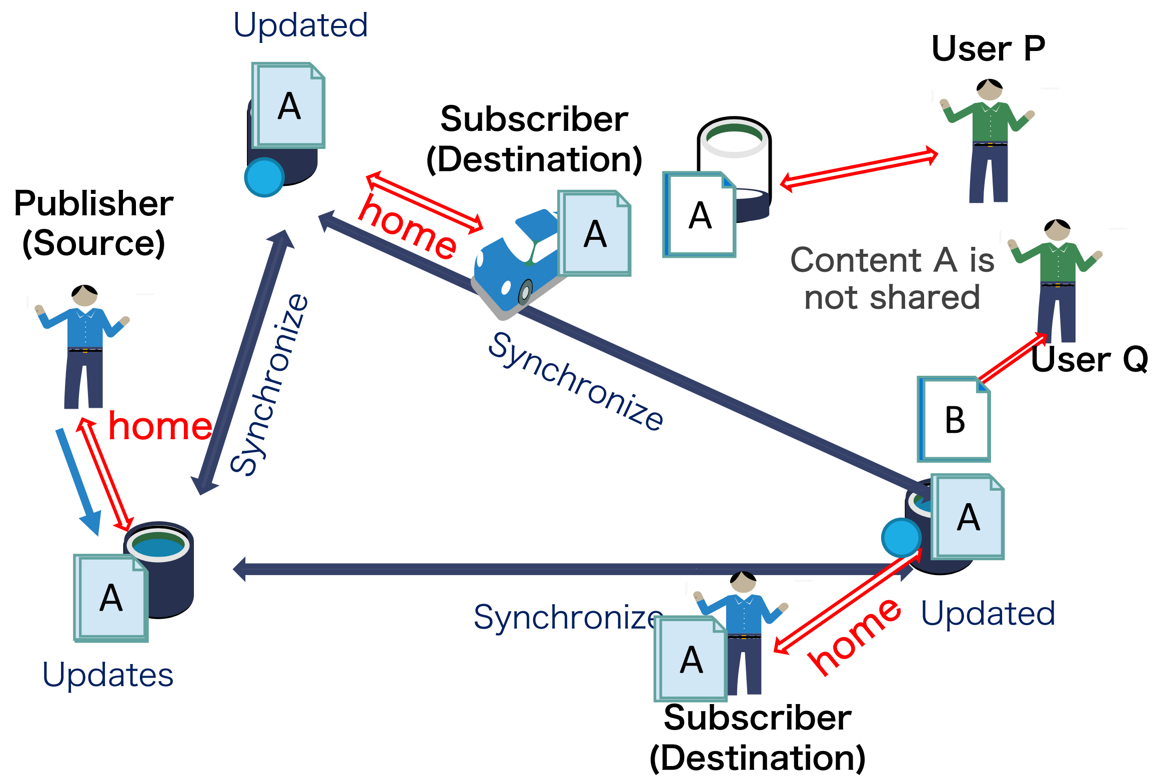
Characteristics

1. **Pub/Sub Distribution using cache**

   The client connects to a fixed server (home server) and performs Publish/Subscribe.

2. **Consistency**

   Synchronize content state between servers

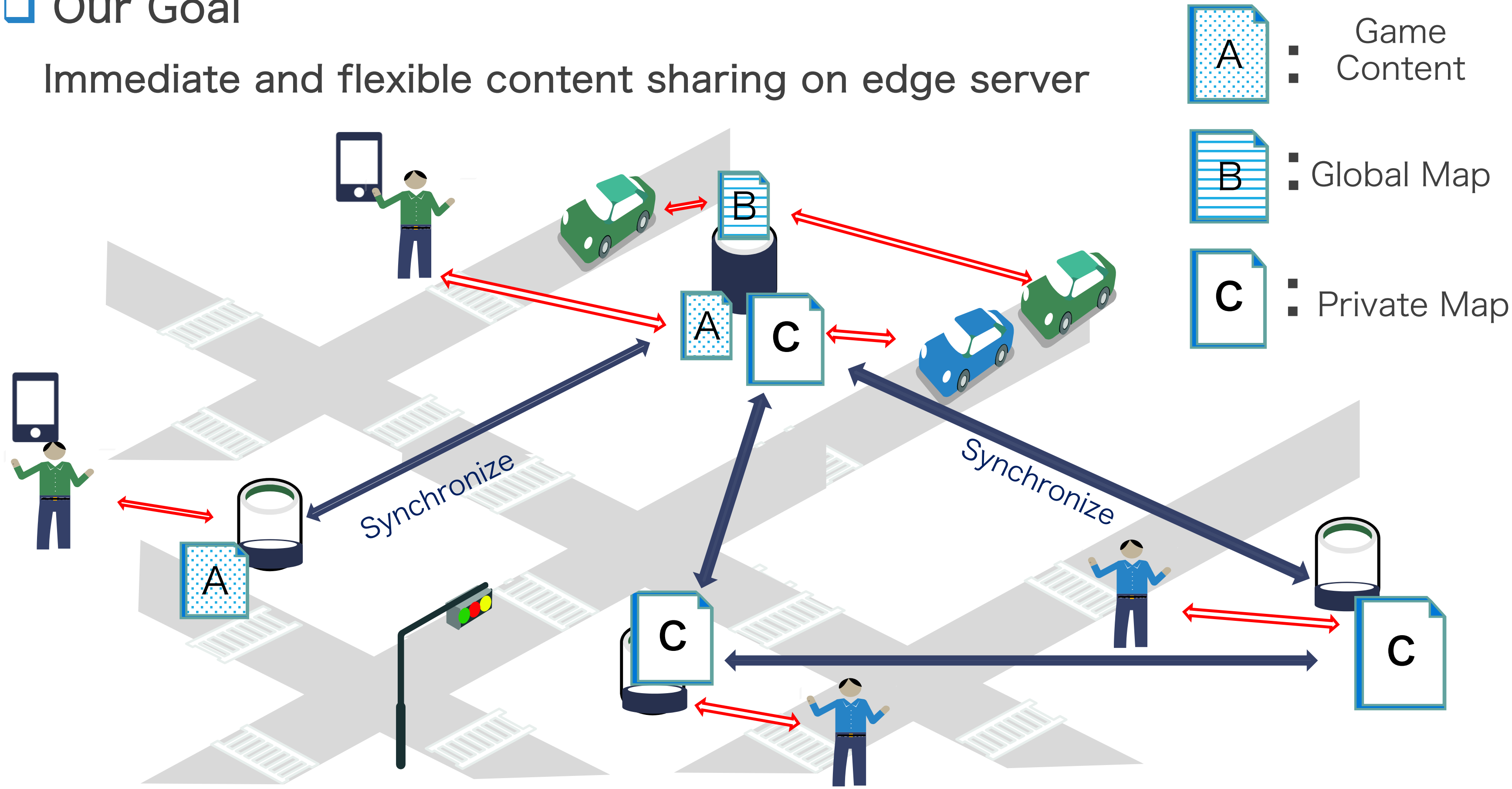


Make content sharing real-time considering resource constraints

Problem and Goal

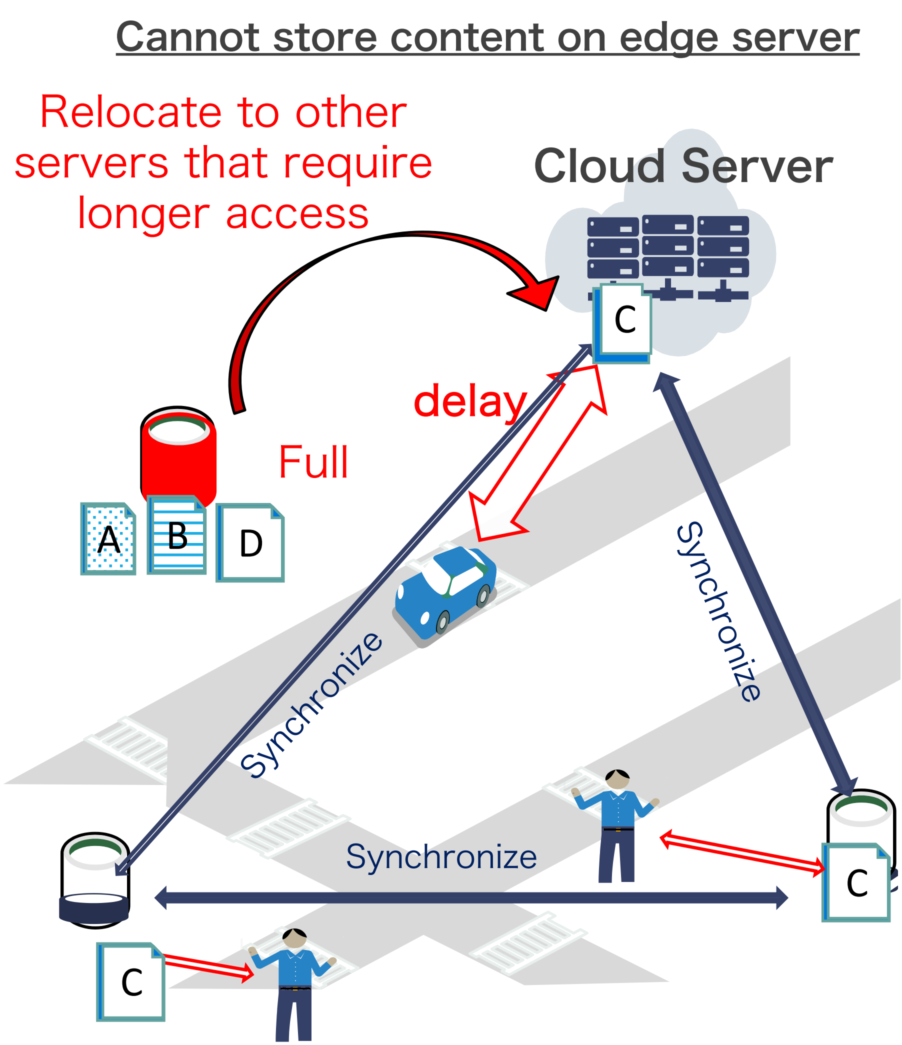
- Our Goal

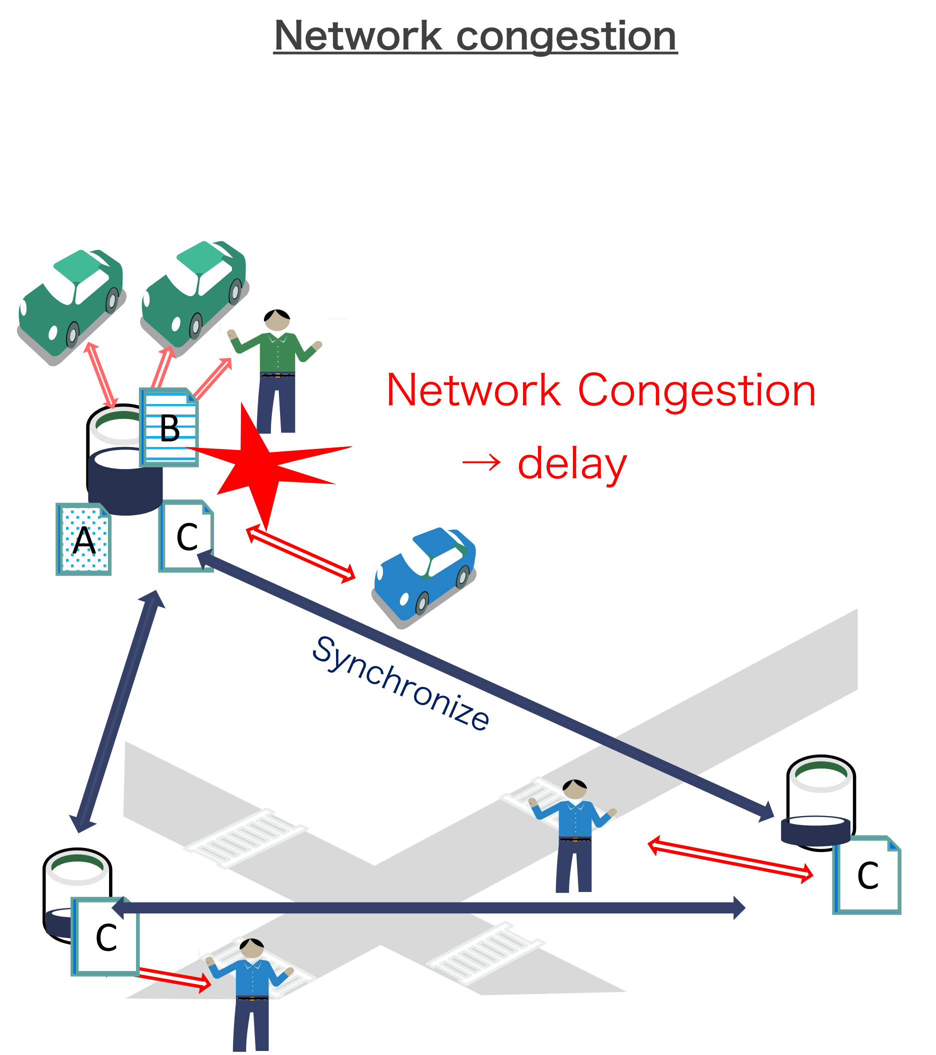
Immediate and flexible content sharing on edge server



cache and bandwidth can be over-consumed as the number of users increases

Case Study


Cannot store content on edge server
Relocate to other servers that require longer access
Cloud Server
C
delay
Full
A
B
D
Synchronize
Synchronize
Synchronize
C
C


Network congestion
Network Congestion
→ delay
B
A
C
Synchronize
C
C

Approach

Policy : Minimize delay for content transmission by efficient resource use

1. Formulation of delay for content delivery

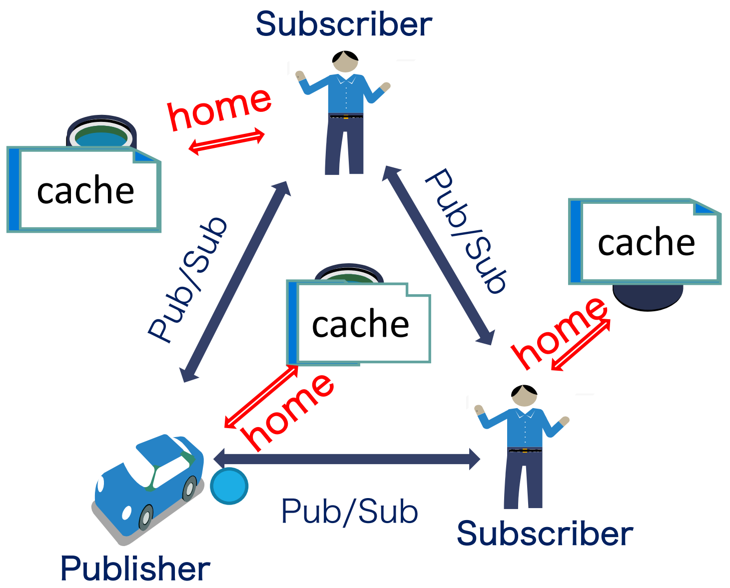
(Original modeling for interactive content sharing among clients)

<u>penalties</u>

- Exhaustion of capacity on edge servers
- work congestion
- Distance between a client and a home server

$$D = \frac{1}{M}\sum_{l=1}^{L}\sum_{m=1,c_m\in\mathcal{C}_l}^{M}\frac{1}{|\mathcal{U}_m|}\sum_{n=1}^{|\mathcal{U}_m|}(t_{mn}r_l\alpha + \psi(|\mathcal{C}_l|)(|\mathcal{C}_l| - C_{max})\beta + d_{ml}\gamma)$$

2. Algorithm Construction which minimize delay D



Leveraging strong pub/sub relationship

Approach

Policy : Minimize delay for content transmission by efficient resource use

1. Formulation of delay for content delivery

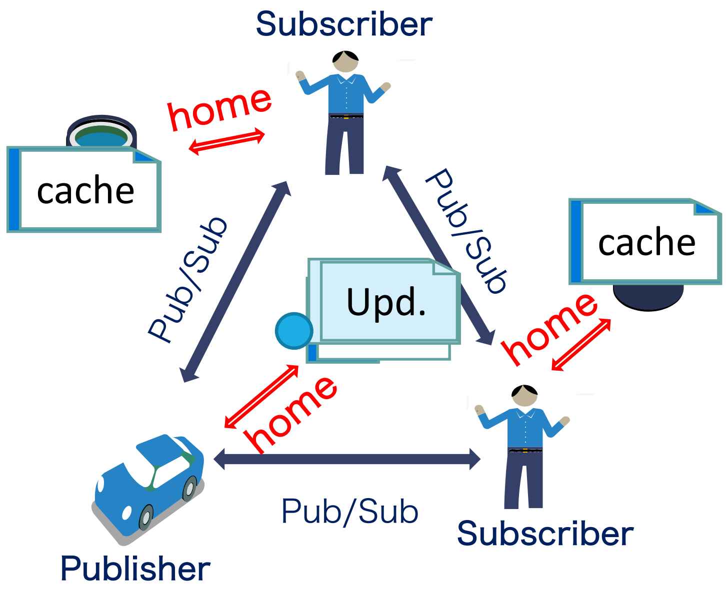
(Original modeling for interactive content sharing among clients)

<u>penalties</u>

- Exhaustion of capacity on edge servers
- work congestion
- Distance between a client and a home server

$$D = \frac{1}{M}\sum_{l=1}^{L}\sum_{m=1,c_m\in\mathcal{C}_l}^{M}\frac{1}{|\mathcal{U}_m|}\sum_{n=1}^{|\mathcal{U}_m|}(t_{mn}r_l\alpha + \psi(|\mathcal{C}_l|)(|\mathcal{C}_l| - C_{max})\beta + d_{ml}\gamma)$$

2. Algorithm Construction which minimize delay D



Leveraging strong pub/sub relationship

Approach

Policy : Minimize delay for content transmission by efficient resource use

1. Formulation of delay for content delivery

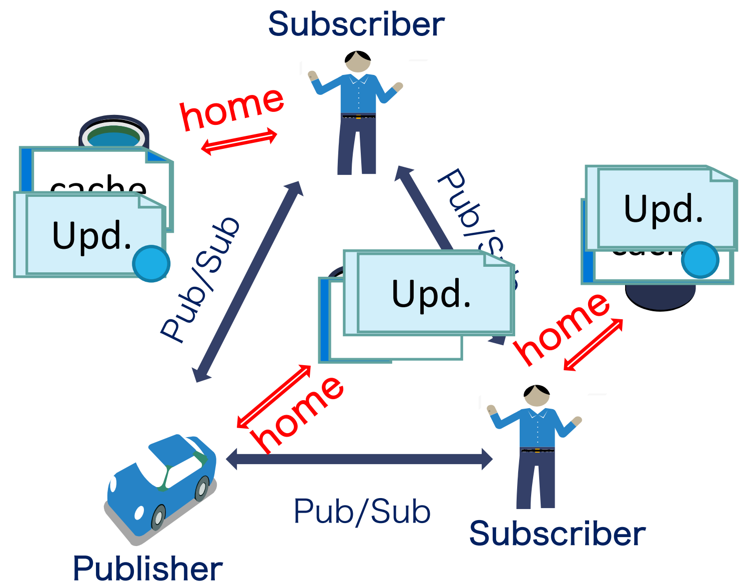
(Original modeling for interactive content sharing among clients)

<u>penalties</u>

- Exhaustion of capacity on edge servers
- work congestion
- Distance between a client and a home server

$$D = \frac{1}{M}\sum_{l=1}^{L}\sum_{m=1,c_m\in\mathcal{C}_l}^{M}\frac{1}{|\mathcal{U}_m|}\sum_{n=1}^{|\mathcal{U}_m|}(t_{mn}r_l\alpha+\psi(|\mathcal{C}_l|)(|\mathcal{C}_l|-C_{max})\beta+d_{ml}\gamma)$$

2. Algorithm Construction which minimize delay D



Leveraging strong pub/sub relationship

Approach

Policy : Minimize delay for content transmission by efficient resource use

1. Formulation of delay for content delivery

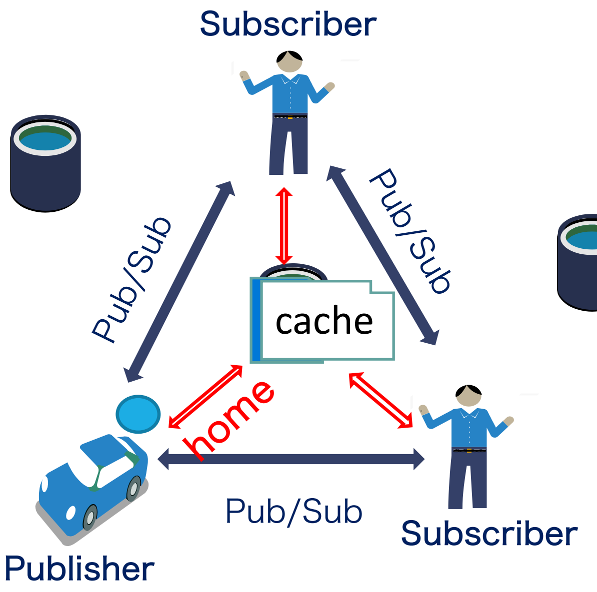
(Original modeling for interactive content sharing among clients)

<u>penalties</u>

- Exhaustion of capacity on edge servers
- work congestion
- Distance between a client and a home server

$$D = \frac{1}{M}\sum_{l=1}^{L}\sum_{m=1, c_m \in \mathcal{C}_l}^{M}\frac{1}{|\mathcal{U}_m|}\sum_{n=1}^{|\mathcal{U}_m|}(t_{mn}r_l\alpha + \psi(|\mathcal{C}_l|)(|\mathcal{C}_l| - C_{max})\beta + d_{ml}\gamma)$$

2. Algorithm Construction which minimize delay D



Leveraging strong pub/sub relationship

Approach

Policy : Minimize delay for content transmission by efficient resource use

1. Formulation of delay for content delivery

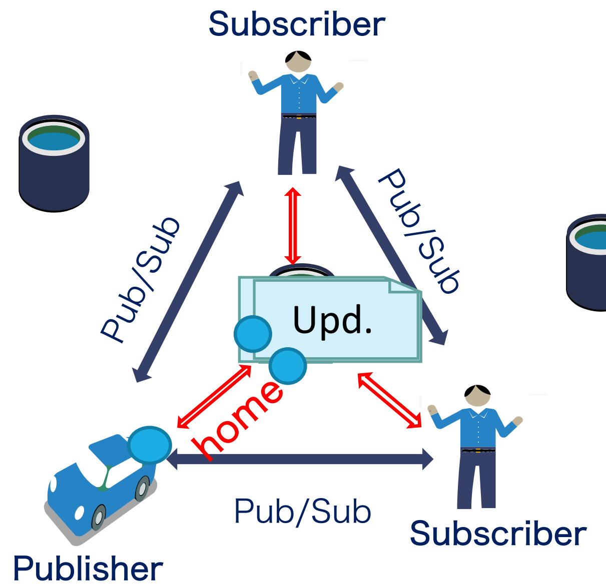
(Original modeling for interactive content sharing among clients)

penalties

- Exhaustion of capacity on edge servers
- work congestion
- Distance between a client and a home server

$$D = \frac{1}{M}\sum_{l=1}^{L}\sum_{m=1,c_m\in\mathcal{C}_l}^{M}\frac{1}{|\mathcal{U}_m|}\sum_{n=1}^{|\mathcal{U}_m|}(t_{mn}r_l\alpha + \psi(|\mathcal{C}_l|)(|\mathcal{C}_l| - C_{max})\beta + d_{ml}\gamma)$$

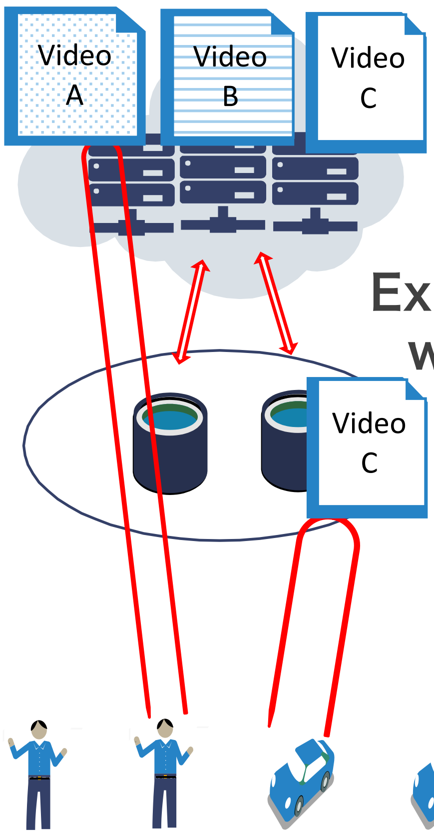
2. Algorithm Construction which minimize delay D



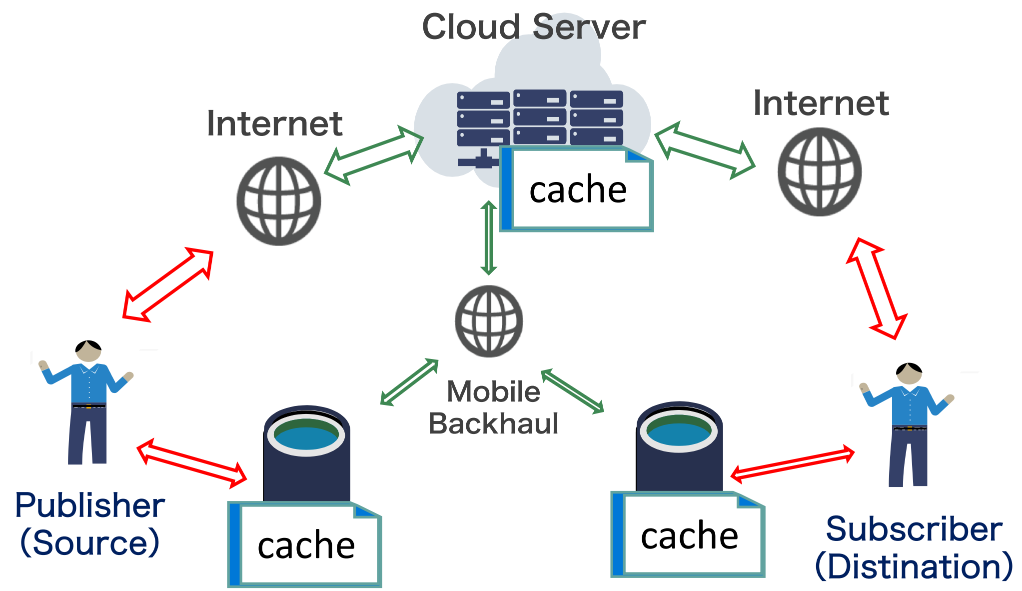
Leveraging strong pub/sub relationship

System Model

edge server capacity is full → Assume overflowed content is deployed on the cloud server



Existing work



Publisher side	Subscriber side
2. Network Congestion(β) 3. Distance in network (γ)	2. Network Congestion(β) 3. Distance in network (γ)

1. Deliver content via cloud server → α times delay

Cost Model

Policy : Minimize delay for content transmission by efficient resource use

Formulation of delay for content transmission

$$D = \frac{1}{M}\sum_{l=1}^{L}\sum_{m=1,c_m\in\mathcal{C}_l}^{M}\frac{1}{|\mathcal{U}_m|}\sum_{n=1}^{|\mathcal{U}_m|}(t_{mn} + t_{mn}r_l\alpha + \psi(|\mathcal{C}_l|)(|\mathcal{C}_l| - C_{max})\beta + d_{ml}\gamma)$$

- L :number of servers
- M :number of clients
- $\mathcal{U}_m$:number of updates by client m
- Base delay
- Content placed in the cloud
- Number of clients connected to server l
- Distance between clients and home servers

1st term : Content delivery delay under ideal conditions

2nd : cache overflow

Give α times delivery delay when transmitting via cloud server

3rd : network congestion

Give penalty β according to the network congestion

4th: Network distance

Give penalty γ according to the distance between users and home servers

Proposed Method

RLCCA（Relation and Locality conscious Cooperative Client Assignment)

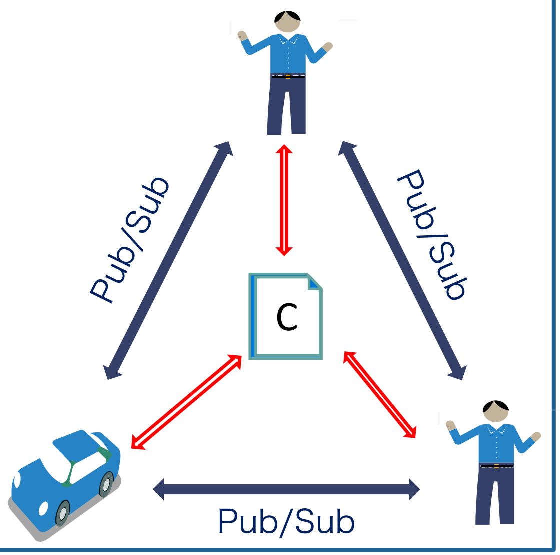
→ Heuristics which decrease delay effectively

1. Locality

Restrict connected servers to nearby servers

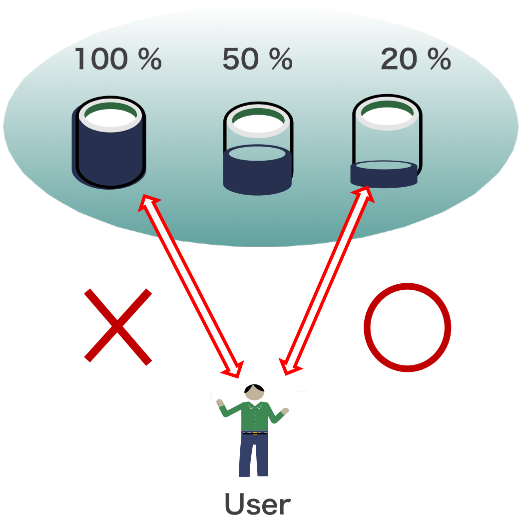
2. Relation

Levarage Pub/Sub relationship



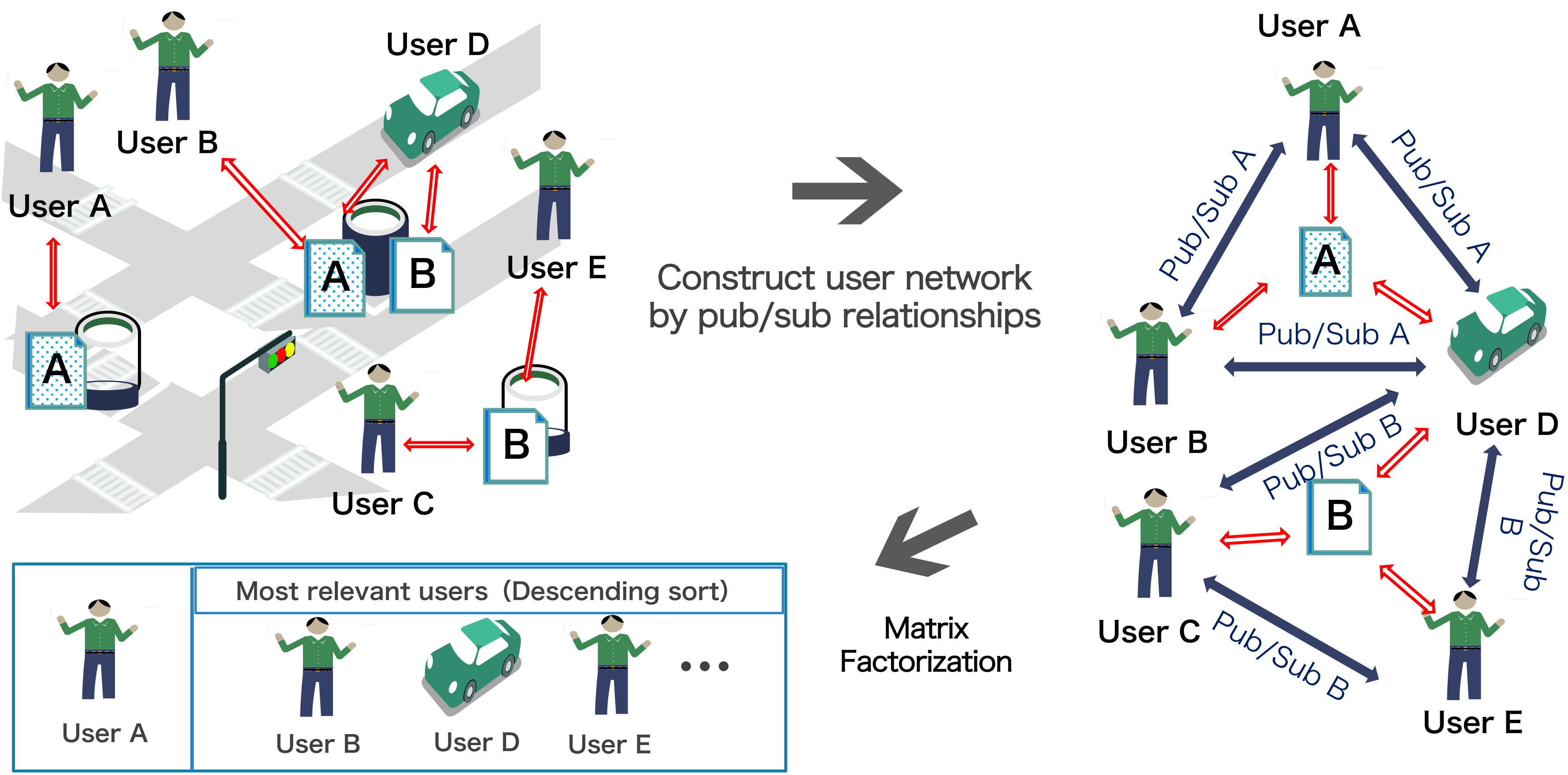
3. Cooperation

Cooperation according to the resource usage



Relation Conscious (Step 2)

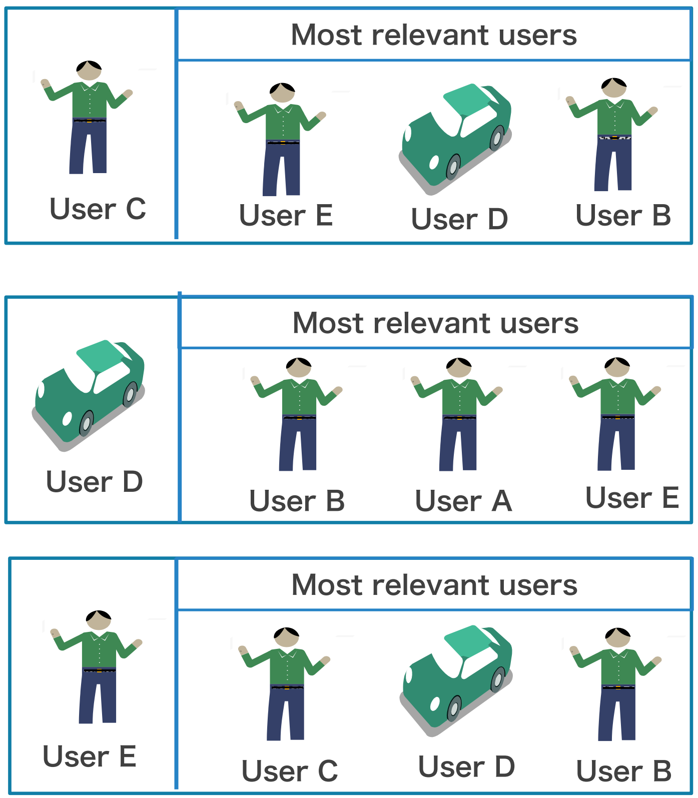
Extracting user-to-user connectivity as knowledge from Pub / Sub relationships

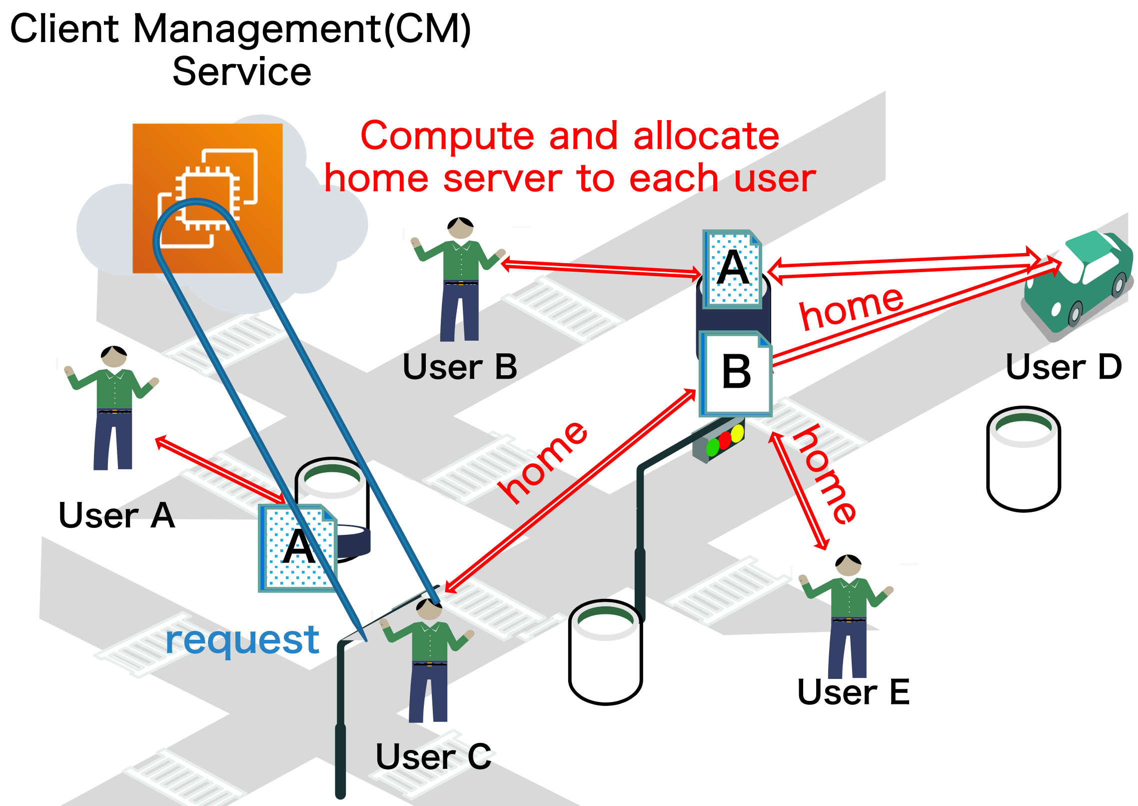


Aggregate cache by assigning highly relevant users to the same server

Relation Conscious (Step 2)

Relevant users refer to the same content cache

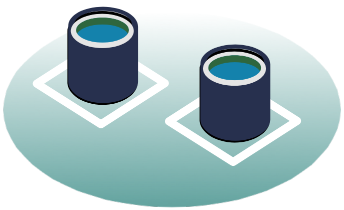




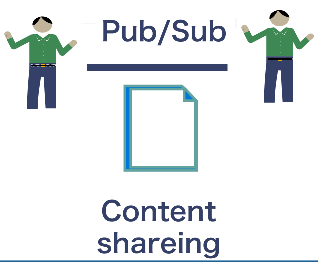
the cache consumption capacity of edge servers can be reduced

Simulation

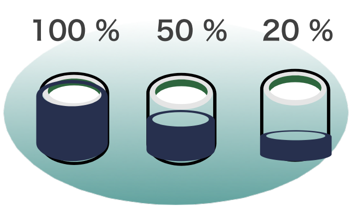
1. Locality



2. Relation



3. Cooperation



<u>Is the use of edge server resources more efficient and content transmission delays reduced?</u>

Metrics : Cost model of content delivery delay *D*

$$D = \frac{1}{M}\sum_{l=1}^{L}\sum_{m=1, c_m \in \mathcal{C}_l}^{M}\frac{1}{|\mathcal{U}_m|}\sum_{n=1}^{|\mathcal{U}_m|}(t_{mn}r_l\alpha + \psi(|\mathcal{C}_l|)(|\mathcal{C}_l| - C_{max})\beta + d_{ml}\gamma)$$

How much penalty is given to content delivery delay D due to excessive consumption of resources. The unit is ms (milli-seconds).

- Allocation to the nearest server
- Random Allocation

Scheme	Locality	Relation	Cooperation
LCA	○	×	×
RLCA	○	○	×
RLCCA	○	○	○

Result

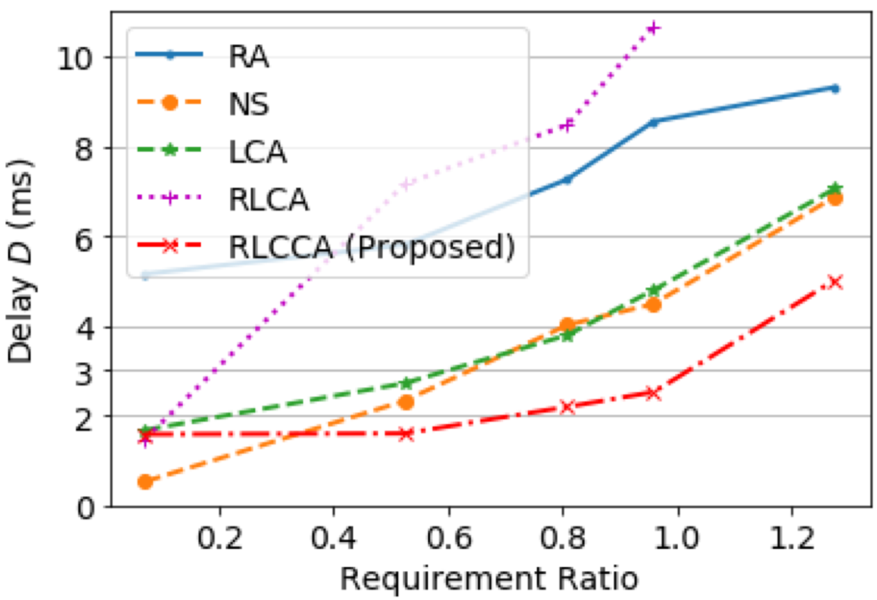
$$\text{Requirement Ratio} = \frac{\text{Number of content caches to be managed on the edge}}{\text{Number of content caches that can be placed on the edge}}$$

(Ratio of the number of contents requiring real-time delivery to the number that can be cached on the edge)

penalties

- Exhaustion of capacity on edge servers
- work congestion
- Distance between a client and a home server

Base content delivery delay : 5 ms



- **The proposed method can reduce the delay by 55% than conventional method**
- **When developers limit the content to be managed on the edge server in advance, the content can be delivered with few delay D**

References

[1] Q. Li, W. Shi, Y. Xiao, X. Ge, A. Pandharipande, “Content size-aware edge caching: a size-weighted popularity-based approach,” in IEEE Global Communications Conference (GLOBECOM), Dec. 2018.

[2] J. Ahn, S. H. Jeon, H. Park, “A novel proactive caching strategy with community- aware learning in CoMP-enabled small-cell networks,” IEEE Communication Let- ters, Vol. 22, No. 9, pp. 1918–1921, Sept. 2018.

[3] L. Hou, L. Lei, K. Zheng, X. Wang, “A Q-learning-based proactive caching strategy for non-safety related services in vehicular networks,” IEEE Internet of Things Journal, Vol. 6, No.3, pp. 4512–4520, June 2019.

[4] N. Itoh, H. Kaneko, A. Kohiga, T. Iwai, H. Shimonishi, “Novel packet scheduling for supporting various real-time IoT applications in LTE networks,” in IEEE International Workshop Technical Commitee on Communications Quality and Reliability (CQR), May 2017.

[5] T. Nagato, T. Tsutano, T. Kamada, Y. Takaki, C. Ohta, “Distributed key-value storage for edge computing and its explicit data distribution method,” IEICE Transactions on Communications, Vol. E103.B, No.1, pp. 20–31, Jan. 2020.

Conclusion

- Contribution

We propose content caching method by which content delivery with low-latency is realized considering edge server resource constraints

- Approach

1. Formulate content delivery delay between publisher and subscriber considering edge server resource constraints
2. Propose heuristics RLCCA（Relation and Locality conscious Cooperative Client Assignment)

- Result

✓Content delivery delay is decreased by 55% than existing methods

✓ When developers limit the content to be managed on the edge server in advance，the content can be delivered with few delay D

(Limitation of the content can be realized with previous methods)