The 18th International Conference on Mobility, Sensing and Networking (MSN 2022)

Online Service Provisioning and Updating in QoS-aware Mobile Edge Computing

Shuaibing Lu, Jie Wu, Pengfan Lu, Jiamei Shi,
Ning Wang, and Juan Fang

Outline

1. Background and Motivation
2. Model and Formulation
3. Service Update Decision based on Lyapunov Optimization
4. Online Optimization of Service Provisioning Strategy
5. Evaluations
6. Conclusions

Background and Motivation

Part 1

Background and Motivation

- **Cloud Data Center Networks (DCNs)**
  - supporting cloud-based applications for large enterprises
- **Mobile Edge Computing (MEC)**
  - deploying edge servers at base stations to supply computation, storage, and networking resources for multiple users

Background and Motivation

Motivation

- find an efficient strategy that can improve the QoS of mobile users by considering the cost constraint.
- determining which services are chosen to be placed in order to obtain a better performance when multiple users make the same decision at the same time.

Objective

- improve the QoS by minimizing the total delay while considering maintaining the long-term cost under the constraint.

Background and Motivation

- An illustrating example
  - ① u_3 moves from m_1 to m_4 at t ;
  - ② u_3 goes back to m_1 ;

Extreme solution 1: migrate or provision a replication of s_3 on m_4 which may bring a lower delay for user u_3.
total cost will be the maximum one

Extreme solution 2: retain service s_3 within m_1.
QoS of users will decrease

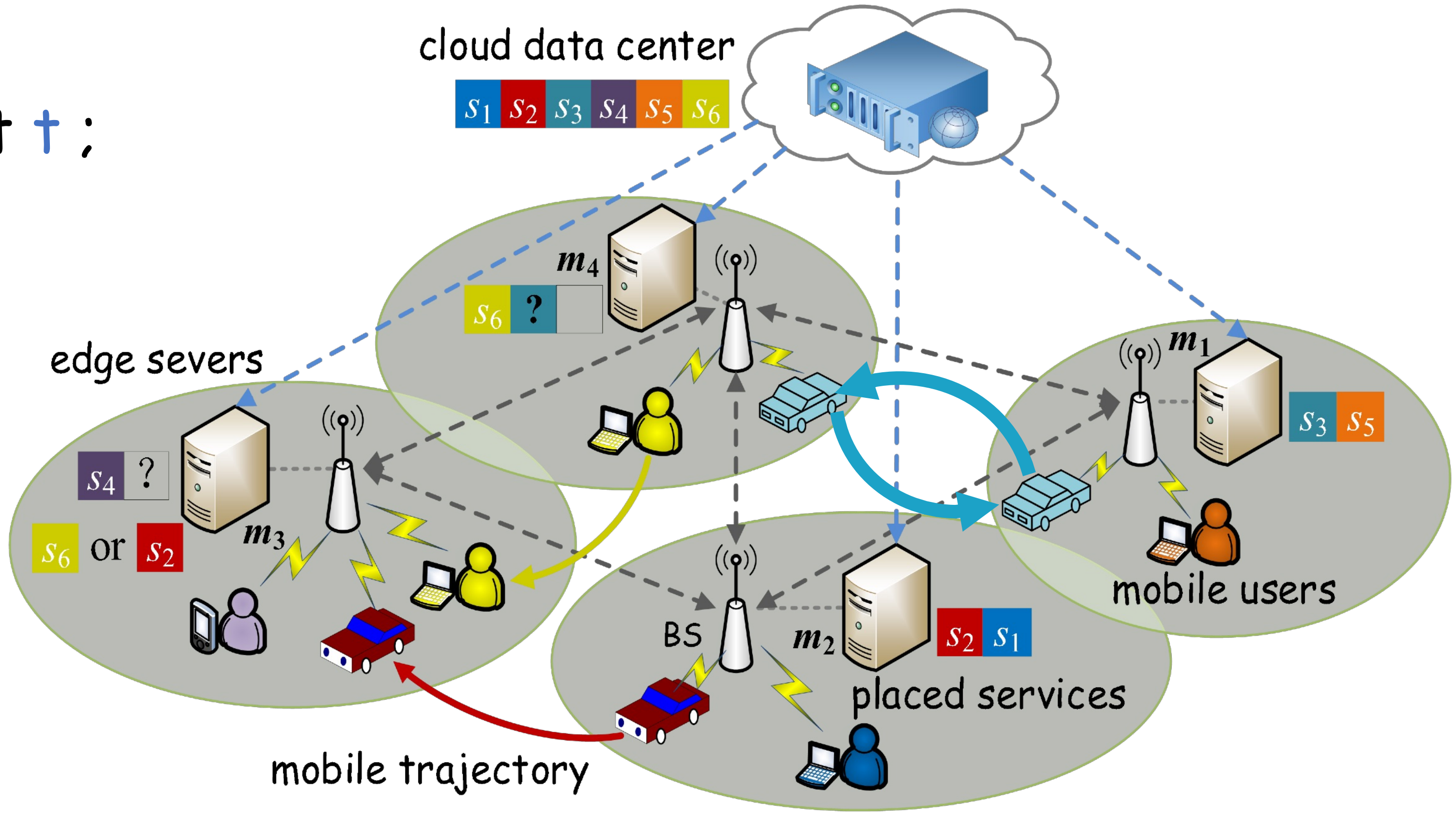


Fig.1. An illustrating example

Migration\Replication\Retaining
?

Model and Formulation

Part 2

Model and Formulation

Model

- system model: $\mathbf{S} = \{s_h\}$, $\mathbf{M} = \{m_j\}$, $\mathbf{U} = \{u_i\}$
- QoS model:

$$\mathbb{D}_{u_i}(t) = D^c_{u_i}(t) + D^l_{u_i}(t) + x_{s_i}(t) \cdot D^u_{u_i}(t)$$

computing delay

$$D^c_{u_i}(t) = \sum_{m_j \in \mathbf{M}} \frac{r_{u_i}(t)}{z^c_{m_j}}$$

$$D^l_{u_i}(t) = \sum_{m_j \in \mathbf{M}} \frac{d_{u_i}(t)}{t_{u_i,m_j}(t)}$$

communication delay

updating delay

$$D^u_{u_i}(t) = \Upsilon(s_i) + \Psi(s_i)$$

- cost model:

$$\mathbb{C}_{s_h}(t) = C^m_{s_h}(t) + C^r_{s_h}(t)$$

migration cost

replication cost

Model and Formulation

Formulation

objective function

P1: minimize $\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{|U|} \mathbb{D}_{u_i}(t)$ (1)

subject to

$$\mathbb{D}_{u_i}(t) = D_{u_i}^{c}(t) + D_{u_i}^{l}(t) + x_{s_i}(t)\cdot D_{u_i}^{u}(t), \quad (2)$$

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T}\sum_{h=1}^{|S|} \mathbb{C}_{s_h}(t) \le \bar{\Gamma}, \mathbb{D}_{u_i}(t) \le \bar{D}, \forall u_i \in \mathbf{U}, \quad (3)$$

$$\sum_{S_{m_i}\in S} W(\mathbf{S}_{m_i}(t)) \le R_{m_i}^{S}, \forall m_i \in \mathbf{M}, \quad (4)$$

$$x_{s_h}(t) \in \{0,1\}, \forall s_h \in \mathbf{S} \quad (5)$$

constraints

Part 3

Service Update Decision Strategy Based on Lyapunov Optimization

Updating Decision Based on Lyapunov Optimization

Decoupling based on Lyapunov Optimization

- decouple the original problem into per-frame deterministic problems by applying the Lyapunov optimization.
- we introduce a virtual queue $Q(t)$ which denotes the historical measurement of the extra cost of services at time slot t.
- queue updating mechanism

$$Q(t-1) = \max\{Q(t) + \mathbb{C}(t) - \bar{\bar{\Gamma}}, 0\}$$

total extra cost

long-term cost

Updating Decision Based on Lyapunov Optimization

Decoupling based on Lyapunov Optimization

- we take expectations and derive that the expected backlog over time slot in [0, T - 1] is less than the threshold.

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\mathbb{C}(t)] \le \lim_{T\to\infty}\frac{1}{T}\mathbb{E}[Q(T)] + \bar{\Gamma}$$

- we define a quadratic Lyapunov function for each slot t.

$$L(Q(t)) \triangleq \frac{1}{2}Q(t)^2$$

- we introduce the one-step conditional Lyapunov drift

$$\triangle\,(Q(t)) \triangleq \mathbb{E}[L\big(Q(t+1)\big) - L(Q(t))|Q(t)]$$

Updating Decision Based on Lyapunov Optimization

Decoupling based on Lyapunov Optimization

Lemma 1: Given the updating decisions of services in set $\mathbf{S}$ according to multiple mobile users $\mathbf{U}$ in each time slot t, the following statement holds:

$$\Delta\left(Q(t)\right) \leq \beta + Q(t)\mathbb{E}[(\mathbb{C}(t) - \bar{\Gamma})|Q(t)]$$

, where $\beta \triangleq \frac{1}{2}(\tilde{\mathbb{C}}(t)^2 + \bar{\Gamma}^2)$.

- According to the Lyapunov optimization framework, we obtain the upper bound of the Lyapunov drift function by introducing a Lyapunov drift-plus-penalty function in each time slot t.

$$P(t) \triangleq \Delta\left(Q(t)\right) + VE[D(t)|Q(t)]$$

non-negative parameter

Updating Decision Based on Lyapunov Optimization

Decoupling based on Lyapunov Optimization

- The performance of the service provisioning strategy is guaranteed by minimizing an upper bound of the following function.

$$P(t) \leq \beta + Q(t)E[(\mathbb{C}(t) - \bar{\Gamma})|Q(t)] + VE[D(t)|Q(t)]$$

minimizing the right side

transformation

- service provisioning and updating problem

P2: minimize $\beta + Q(t)(\mathbb{C}(t) - \bar{\Gamma}) + V\mathbb{D}(t)$ (12)

subject to (2)-(5). (13)

Updating Decision Based on Lyapunov Optimization

Optimal Services Updating Decision Strategy

Definition 1 (Optimal Service Updating (OSU) Problem): Given the distribution of users U, the topology of edge network G, and the function Θ(t), an OSU problem is how to find a decision for services in S to minimize P2 under the constraints at time slot t.

Scenario 1 :
OSU with no prediction

Scenario 2 :
OSU with prediction

Optimal Services Updating Decision Strategy——OSU with no prediction

- OSU problem without available information caused by the inaccurate prediction results or in the initial or training stages of mobile users in per-slot.

Definition 2 (conflict resolution factor): Let η_h indicate the conflict resolution factor of service s_h and $\eta_h = \mathbb{C}_{s_h}(t)/\mathbb{D}^l_{u_h}(t)$, where $\overline{\mathbb{D}^l_{u_h}(t)} = \mathbb{D}^l_{u_h}(t)|_{s_h \notin \mathbf{s}_{m_t}(t)}$.

Updating Strategy with No Prediction (USNP) Algorithm

- **Step 1**
  - each user in set $\mathbf{U}$, choose the updating decision by optimizing P2
- **Step 2**
  - check the feasibility of services on edge servers by checking whether

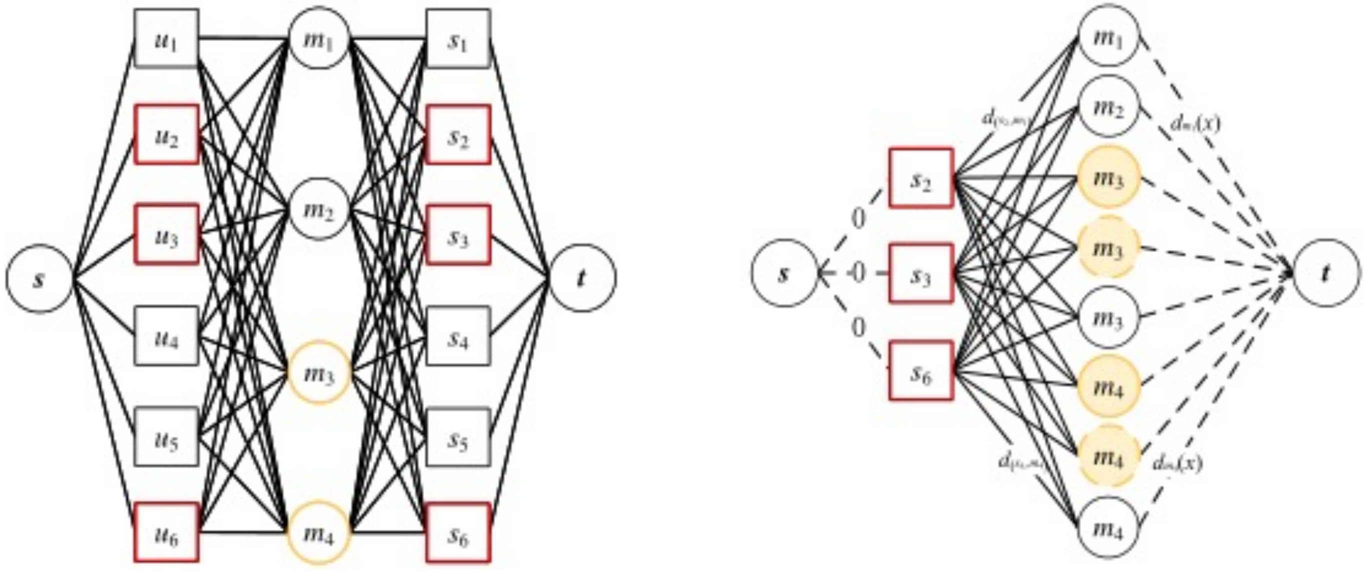
$$\sum_{S_{m_i} \in S} W(S_{m_i}(t)) \geq R_{m_i}^S$$

  - $\sum_{S_{m_i} \in S} W(S_{m_i}(t))$ denote the total number of services provisioning on m_i
- **Step 3**
  - Choose a service by an increasing order $i = argmin\{\eta_h\}$
- **Step 4**
  - service updating decision $\mathbf{X}(t)$

❑ Optimal Services Updating Decision Strategy——OSU with prediction

Lemma 2: The decision of the OSU problem can be solved by minimizing $\Theta(t)$, where $\Theta(t) = Q(t)\mathbb{C}(t) + V\mathbb{D}(t)$.

Definition 3 (Activity Set): Let $\hat{\mathbf{U}}(t)$ indicate the activity set of users at time slot t, where $u_i \in \hat{\mathbf{U}}(t)$ is the user whose current location $L_{u_i}(t)$ is going far away from the edge server for initial connection $L_{u_i}(t-1)$.



(a) original connectivity graph (b) extracted connectivity graph

Fig.2. The connectivity graphs of Fig1.

Updating Strategy with Prediction (USP) Algorithm

- **Step 1**
  - construct the original connectivity graph g based on the provisioning of $\mathbf{S}$, the connections of $\mathbf{G}$, and $\mathbf{U}$
- **Step 2**
  - calculate $\zeta_{u_i}(t) = (L_{u_i}(t-1), L_{u_i}(t))$
- **Step 3**
  - If $\zeta_{u_i}(t) = 1$, this denotes that u_i has gone away from the edge server at time slot $t - 1$. Then, construct the activity set by adding u_i into set $\hat{\mathbf{U}}(t)$,
  - Otherwise, it denotes that u_i always stays near the edge server from $t-1$ to t, and update $\mathbf{U}(t)$;
  - construct the activity set $\hat{\mathbf{U}}(t)$.

Updating Strategy with Prediction (USP) Algorithm

Step 4

- construct the extracted connectivity graph $\mathbf{G}^{\circ}$ based on the activity set $\hat{\mathbf{U}}(t)$

Step 5

- we replace the link with $|\hat{\mathbf{U}}(t)|$ parallel ones with weight $d_{m_i}(x)|_{u_x \epsilon \hat{U}(t)}$ between edge servers and destination t.

Step 6

- find a feasible service updating decision with min-cost flow of $\hat{\mathbf{U}}(t)$ and return the updating decision $\mathbf{X}(t)$.

Online Optimization of Service Provisioning Strategy

Part 4

Online Optimization of Service Provisioning strategy ($\boldsymbol{O-OSP_{\omega}}$) Algorithm

- the main idea of $\boldsymbol{O-OSP_{\omega}}$ is to leverage the prediction model to look forward the trajectories of users in multiple steps and use the information to realize the service provisioning.

Definition 4 (feasible decision frequency): Let $\varrho^{a}_{s_{h|\omega}}(t)$ indicate the feasible decision frequency of s_h under the value a°, where $\varrho^{a^{\circ}}_{s_{h|\omega}}(t) = \frac{1}{\omega}\sum_{x=0}^{x=\omega-1} f(A^{(x)}_{s_h}, a^{\circ})$.

A function to indicate whether the result in queue $A^{(x)}_{s_h}$ is equal to a° , i.e., $a_{s_h} = a^{\circ}$.

Online Optimization of Service Provisioning

Step 1

- get service updating decision $\mathbf{X}(t)$ using Algorithm 1

Step 2

- obtain the service updating decision $\mathbf{X}(t)$ using Algorithm 2 based on $\widehat{L}_{U|[\tau,\tau+\omega]}$, $\widehat{L}_{U|[\tau,\tau+\omega]}$ is the trajectory of user u_i in a ω time steps prediction window starting at time τ

Step 3

- set $\tilde{t} = (t-\tau)\ mod\ \omega$, and check whether the prediction steps are less than ω.

Step 4

- use a queue $A_{s_h}^{(x)}$ to record the decision values of service s_h in x time steps,
- let $\varrho_{s_{h|\omega}}^{a^\circ}$ indicate the feasible decision frequency of s_h under the value a°

Step 5

- update the service provisioning for services by feasible decision frequencies $X_{s_h}(\tilde{t}) = \arg\max_{a^\circ \in A_{s_h}^{(\omega)}} \{\varrho_{s_{h|\omega}}^{a^\circ}\}$.

Online Optimization of Service Provisioning strategy ($O-OSP_{\omega}$) Algorithm

Theorem 1: By applying OSP, the time-average system delay satisfies:

$$\frac{1}{T}\sum_{t=0}^{t=T-1}\mathbb{D}(t) \leq \frac{1}{2}(OPT+\beta+V|\mathbf{U}|\overline{D})+\epsilon+\frac{1}{\omega}W\cdot\alpha\cdot T.$$

Evaluations

PART 5

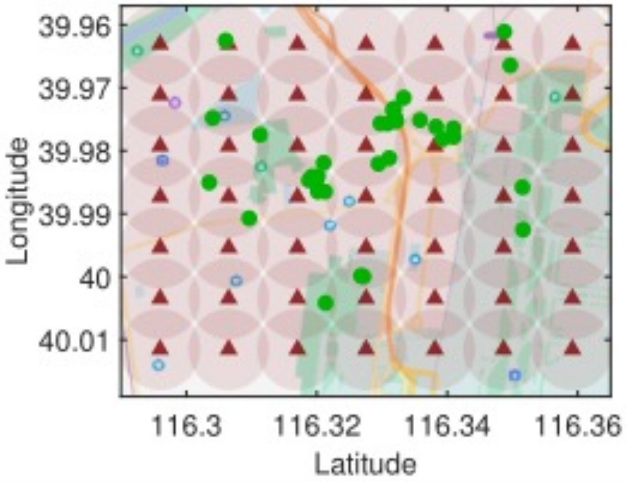
Evaluations

Basic Setting

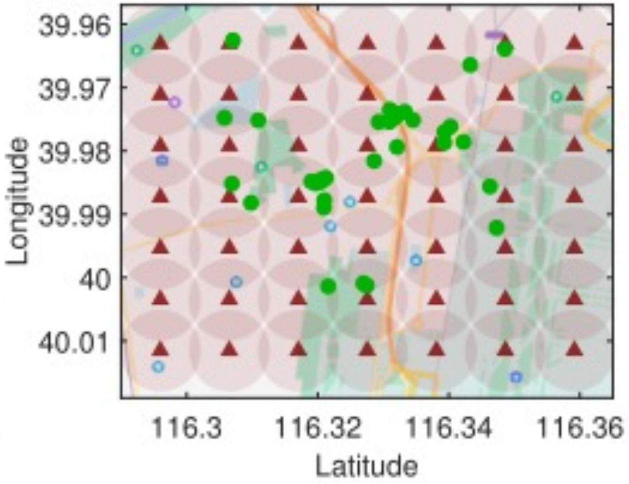
- Hardware: E5-2620 CPU, NVIDIA RTX5000 GPU, 128Gb memory, 2Tb hard disk.
- Dataset: Microsoft GPS trajectory dataset (182 users), 40 users were selected to construct $\mathbf{U}$.
- Range: 2.5km, user trajectories during 60 consecutive time slots.

Evaluations

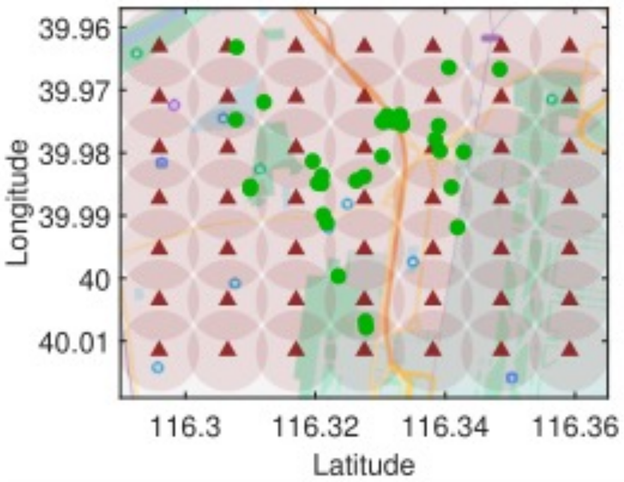
- Users distribution at different time slots.



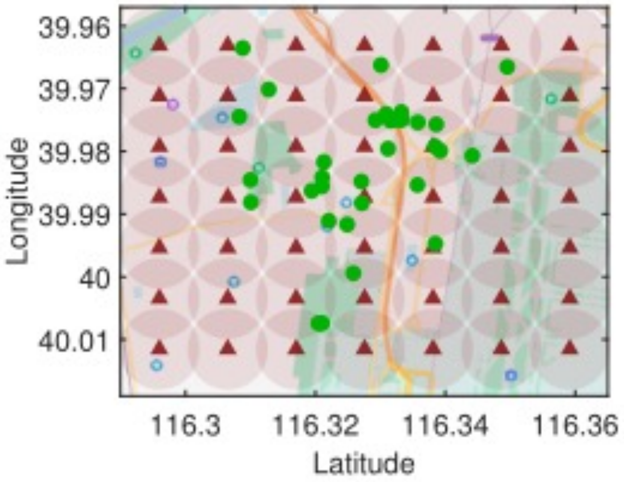
(a) time slot 0.



(b) time slot 20.



(c) time slot 40.



(d) time slot 59.

- setting 49 edge servers with the service range of 450 meters.
- computing capacity of each server to range from 2GHz to 5GHz.
- data size of each service is 1GB.
- storage of each edge server ranges from 5GB to 10GB.

Evaluations

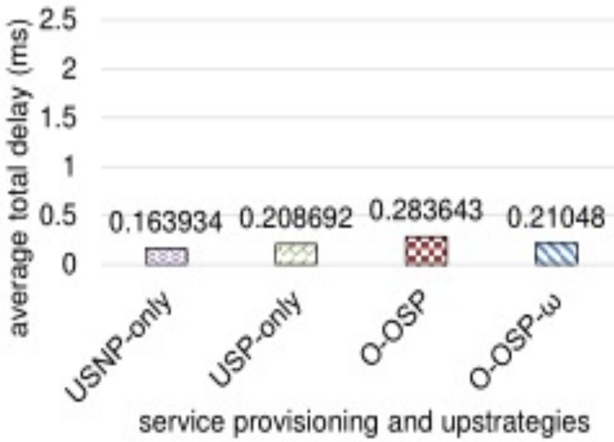
- **Three Comparison algorithms**
  - USNP-only: Services provisioning and updating without using the prediction information, and the decisions are only made by USNP .
  - USP-only: Services provisioning and updating by using the prediction information, and the decisions are only made by USP .
  - O-OSP: Online services provisioning and updating based on $\boldsymbol{O-OSP_{\omega}}$ without considering ω steps prediction.

Evaluations

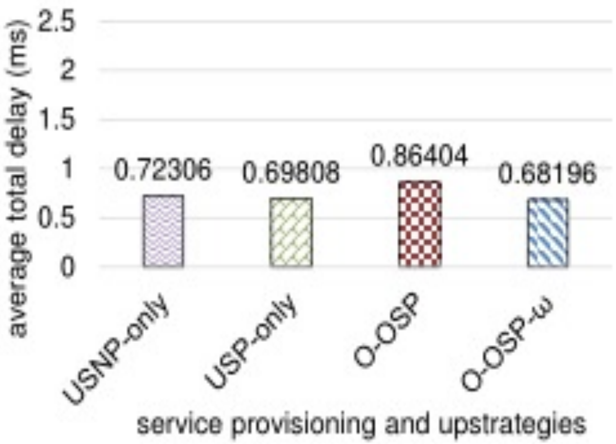
Experiment Results

Average total delay under different strategies

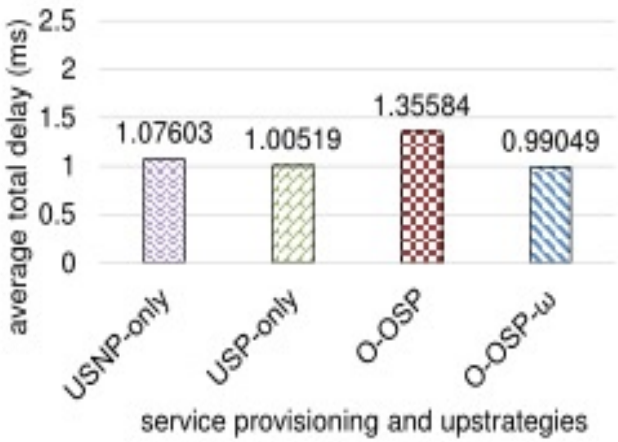
- The numbers and trajectories of users in set U affect the results of strategies
- Prediction with ω slots in $\boldsymbol{O-OSP_{\omega}}$ can effectively reduce the problem of service quality degradation caused by erratic activities of mobile users.



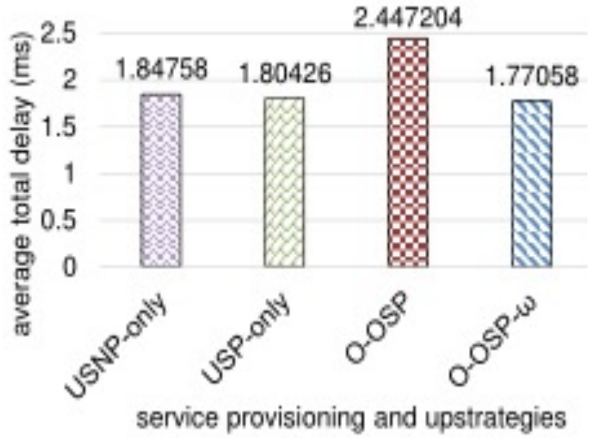
(a) # of users (10).



(b) # of users (20).



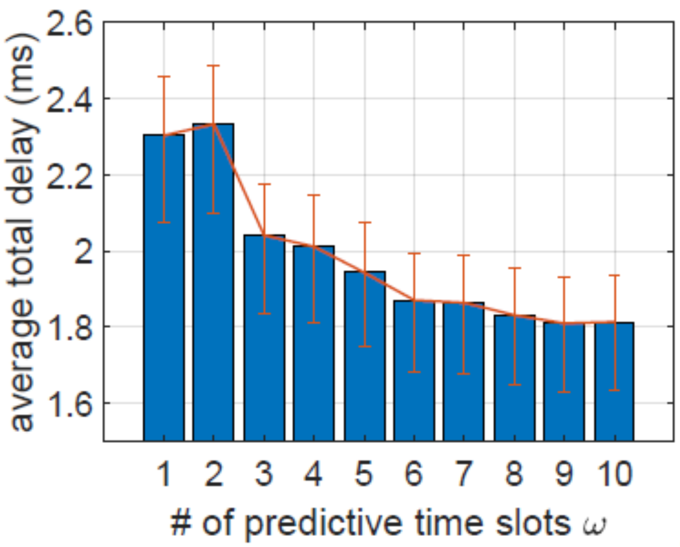
(c) # of users (30).



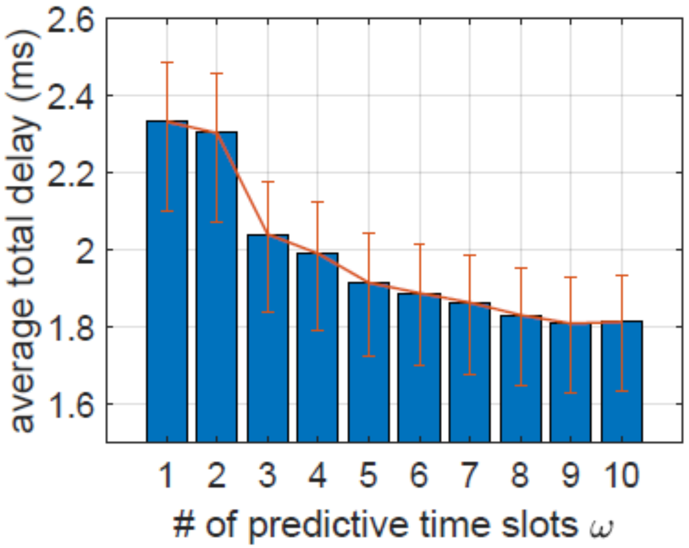
(d) # of users (40).

Experiment Results

- **Average total delay with different ω time slots**
  - The value of ω can influence the efficiency of $\boldsymbol{O-OSP_{\omega}}$
  - The accuracy of the chosen prediction model has little effect on the results of $\boldsymbol{O-OSP_{\omega}}$



(a) group with 71.7% accuracy.



(b) group with 56.6% accuracy.

Conclusions

Part 6

Conclusions

In this paper, we investigate the service provisioning and updating problem under the multiple-users scenario by improving the performance of services with the long-term cost constraint.

Contributions

- We first decouple the original long-term optimization problem into a per-slot deterministic one by using Lyapunov optimization.
- We propose two service updating decision strategies by considering the trajectory prediction conditions of users.
- We design an online strategy by utilizing the committed horizon control method while looking ahead to ω slots predictions.

Experiments

- Microsoft GPS trajectory dataset

Q&A