Edge Computing QoE Maximization in EV Parking Scenario

Yu-Chieh Lee, Yao Chiang, Hung-Yu Wei
Department of Electrical Engineering, National Taiwan University

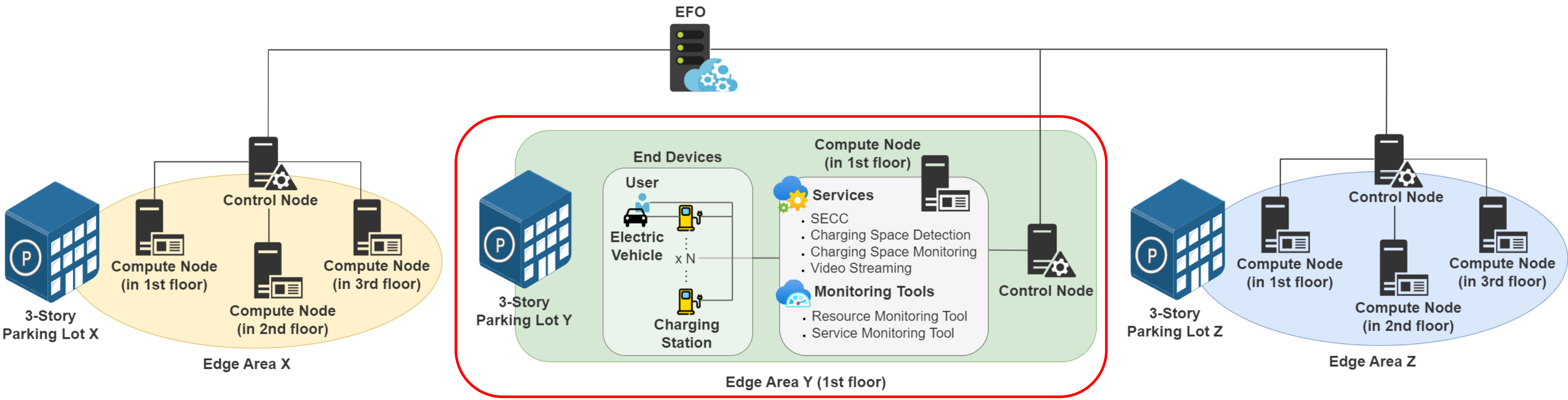
Introduction – EV parking + Edge computing

- Background

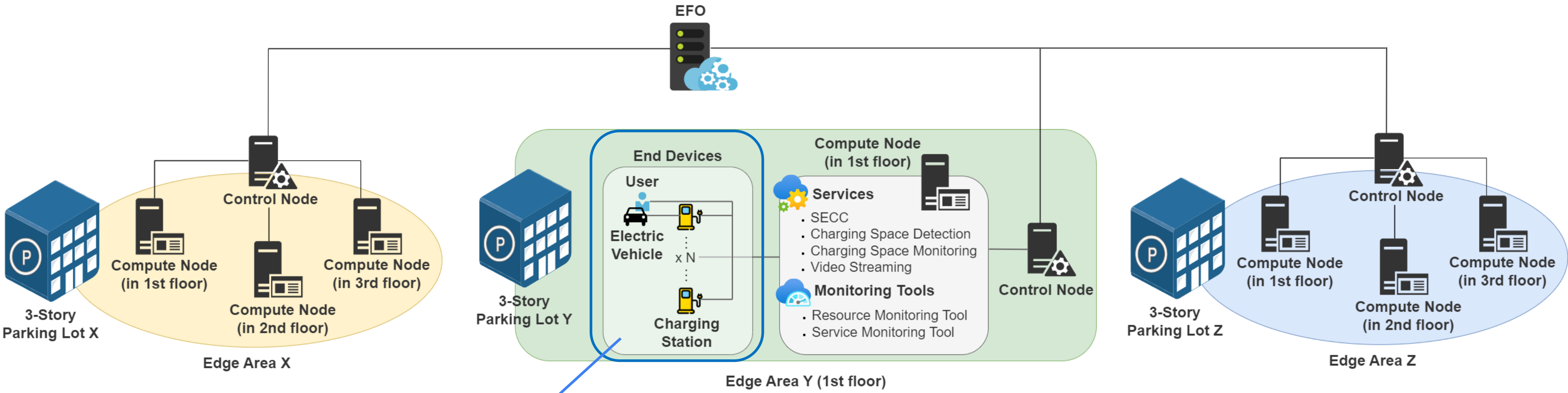
- Motivate us to propose
  - Scenario of smart EV parking lots + edge computing system
  - Methods to achieve efficient resource allocation and fast service provisions

System Model

- 3-tier edge computing architecture in IEEE 1935 standard
- 4 services included: SECC, Charging Space Detection, Charging Space Monitoring, Video Streaming



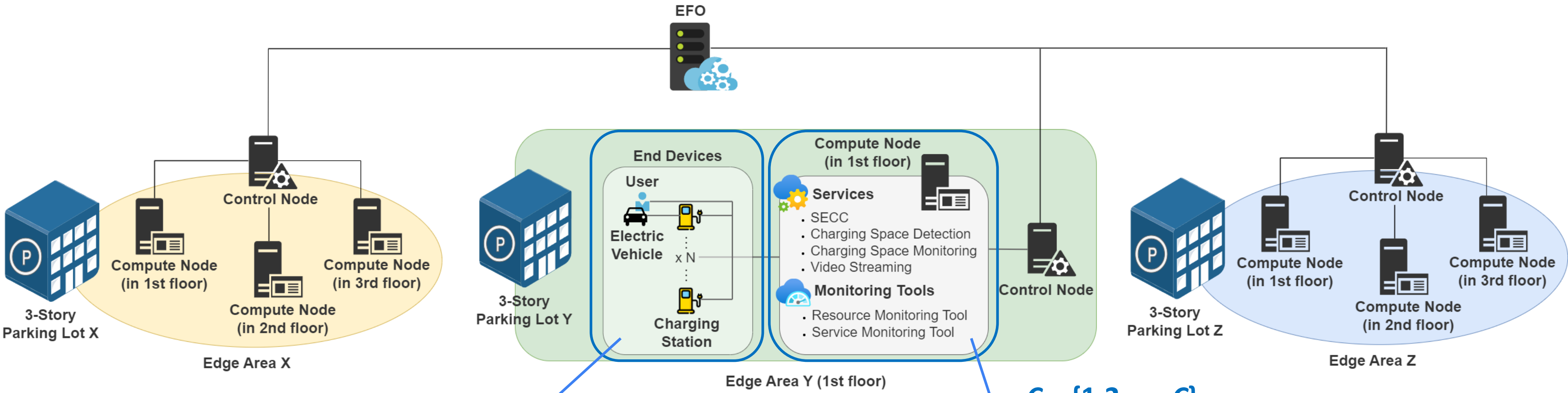
System Model



1. $\mathcal{U} = \{1,2, \ldots, U\}$

- EV owners' mobile devices + EV charging stations
- Request for services

System Model



1. *U* = {1,2, …, *U*}

- EV owners‘ mobile devices + EV charging stations
- Request for services

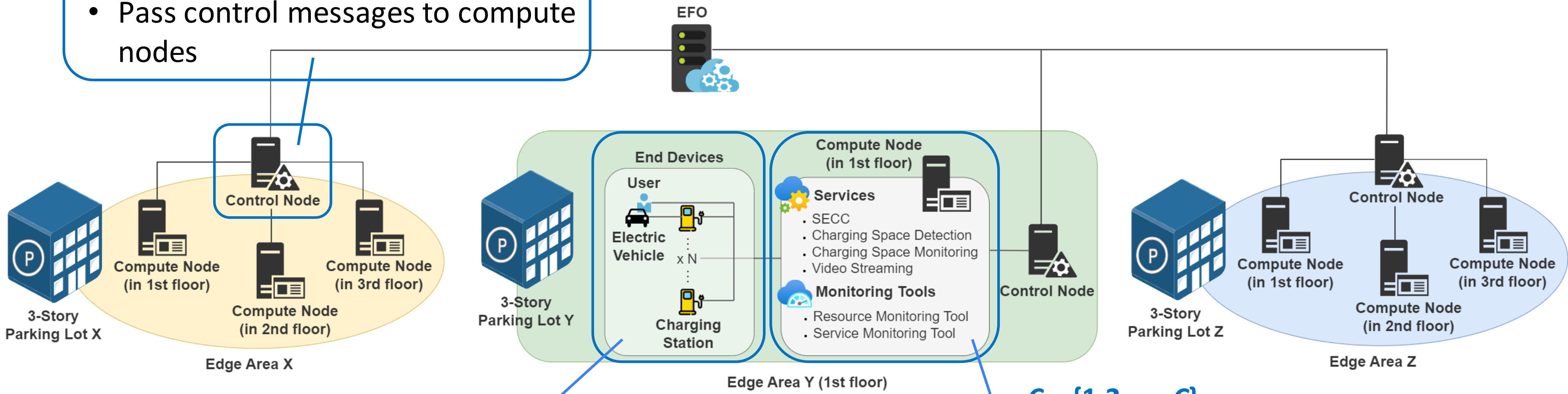
2. *C* = {1,2, …, *C*}

- **Computer-level** entity with **limited resources**
- Provides services
- Send resource and request information to control node

System Model

3.

- **Control-level** entity
- Collects data and pass to EFO
- Pass control messages to compute nodes



1. $U = \{1,2,\ldots,U\}$

- EV owners' mobile devices + EV charging stations
- Request for services

2. $C = \{1,2,\ldots,C\}$

- **Computer-level** entity with **limited resources**
- Provides services
- Send resource and request information to control node

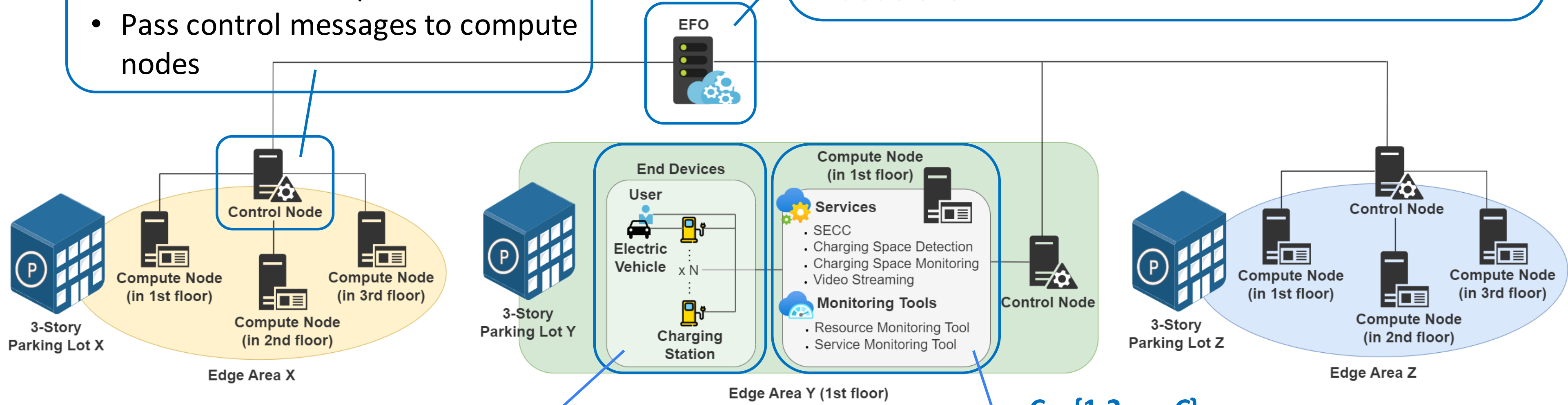
System Model

3.

- **Control-level** entity
- Collects data and pass to EFO
- Pass control messages to compute nodes

4.

- **Orchestrator-level** entity
- Global view of the system
- Runs proposed algorithm to **predict future requests**, make **service deployment and assignment decisions**



1. $U = \{1, 2, \ldots, U\}$

- EV owners‘ mobile devices + EV charging stations
- Request for services

2. $C = \{1, 2, \ldots, C\}$

- **Computer-level** entity with **limited resources**
- Provides services
- Send resource and request information to control node

QoE Models

$S = \{SECC, CSD, CSM, VS\}$

Average response time | Model accuracy | Weight factors

Supply equipment communication controller (SECC)

$$Q_{c,u}^{SECC} = \begin{cases} \frac{1}{\frac{1}{n}\sum_{i=1}^{n}(RTT_{c,u}+t_i^{SECC})} & \text{if no timeout} \\ -1 & \text{if timeout} \end{cases}$$

Video streaming (VS)

$$Q_{c,u}^{VS} = \alpha X_u - (1-\alpha) I^a$$
$$= \alpha X_u - (1-\alpha)(RTT_{c,u} + t^{VS})$$

Initial waiting time

Video resolution

Charging space detection (CSD)

$$Q_{c,u}^{CSD} = \beta \times \frac{1}{\frac{1}{n}\sum_{i=1}^{n}(RTT_{c,u} + t_i^{CSD})} + (1-\beta) \times 0.72$$

Charging space monitoring (CSM)

$$Q_{c,u}^{CSM} = \gamma \times \frac{1}{\frac{1}{n}\sum_{i=1}^{n}(RTT_{c,u} + t_i^{CSM})} + (1-\gamma) \times 0.9624$$

Service priority: SECC > CSD = CSM > VS

Main Problem Formulation

Objective

$$\max_{D} \sum_{t=0}^{q} \sum_{u \in U} \sum_{c \in C} \sum_{s \in S} v_{c,u}^{s}(t) \times \boxed{d_{c,u}^{s}(t)} \text{, where } v_{c,u}^{s}(t) = \boxed{prior^{s}} \times \boxed{R_{u}^{s}(t)} \times \boxed{Q_{c,u}^{s}}$$

- $d_{c,u}^{s}(t)$: whether deploy service *s* requested by user *u* on compute node *c* (boolean)
- $prior^{s}$: priority of service type *s*
- $R_{u}^{s}(t)$: request from end device *u* for service *s* (boolean)
- $Q_{c,u}^{s}$: corresponding QoE value

s.t.

Constraints

$$C1: \sum_{s \in S} \sum_{u \in U} \tau_{c,u}^{s}(t) < \tau_{c}, \forall c \in C$$

Total CPU limit of the compute node

$$C2: \sum_{s \in S} \sum_{u \in U} \omega_{c,u}^{s}(t) < \omega_{c}, \forall c \in C$$

Total memory limit of the compute node

$$C3: \sum_{s \in S} \sum_{u \in U} dl_{c,u}^{s}(t) < ul_{c}, \forall c \in C$$

Total uplink capacity limit of the compute node

$$C4: \sum_{s \in S} \sum_{u \in U} ul_{c,u}^{s}(t) < dl_{c}, \forall c \in C$$

Total downlink capacity limit of the compute node

$$C5: \sum_{s \in S} \sum_{c \in C} ul_{c,u}^{s}(t) < ul_{u}, \forall u \in U$$

Total uplink capacity limit of the end device

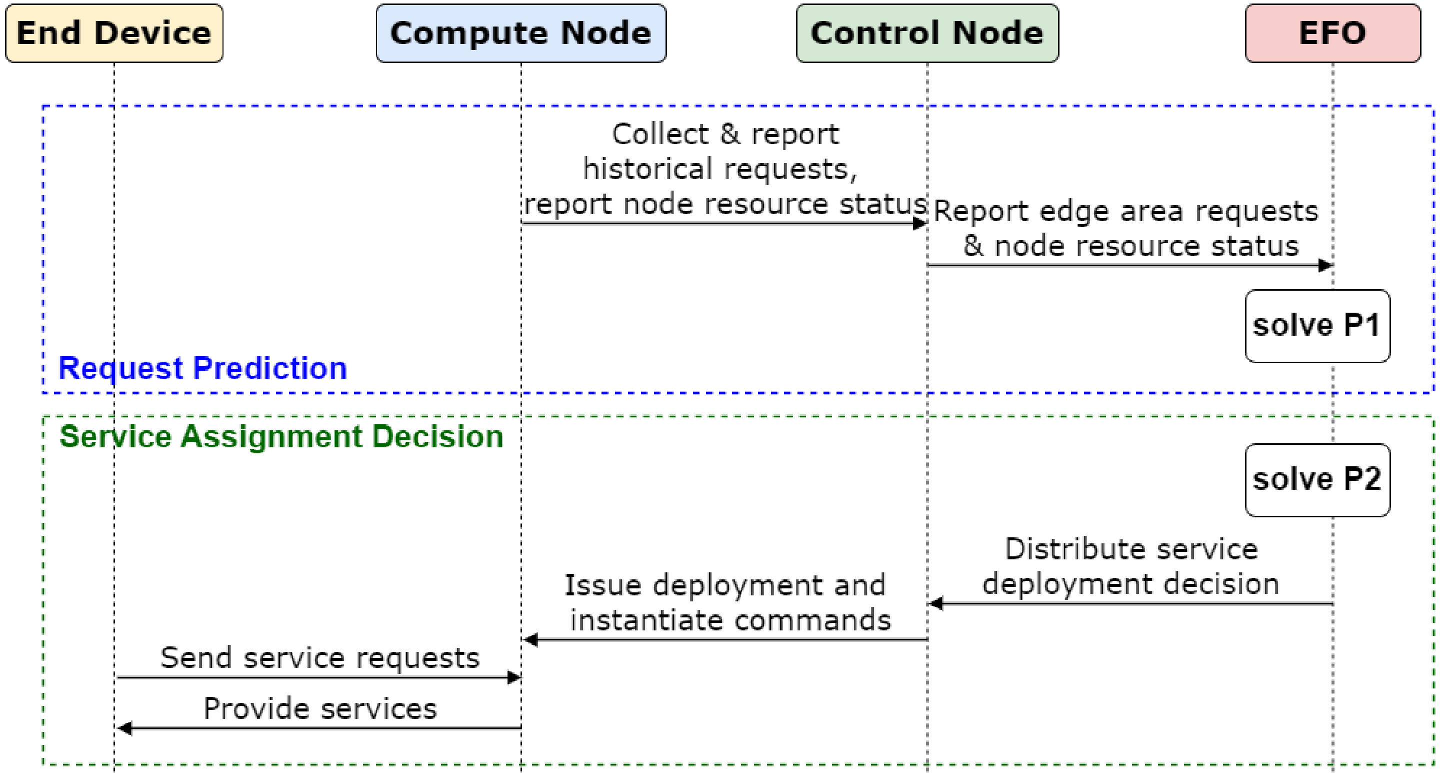
$$C6: \sum_{s \in S} \sum_{c \in C} dl_{c,u}^{s}(t) < dl_{u}, \forall u \in U$$

Total downlink capacity limit of the end device

$$C7: \sum_{c \in C} d_{c,u}^{s}(t) <= 1$$

Request from end device *u* for service *s* be accepted at most once

Proposed Method

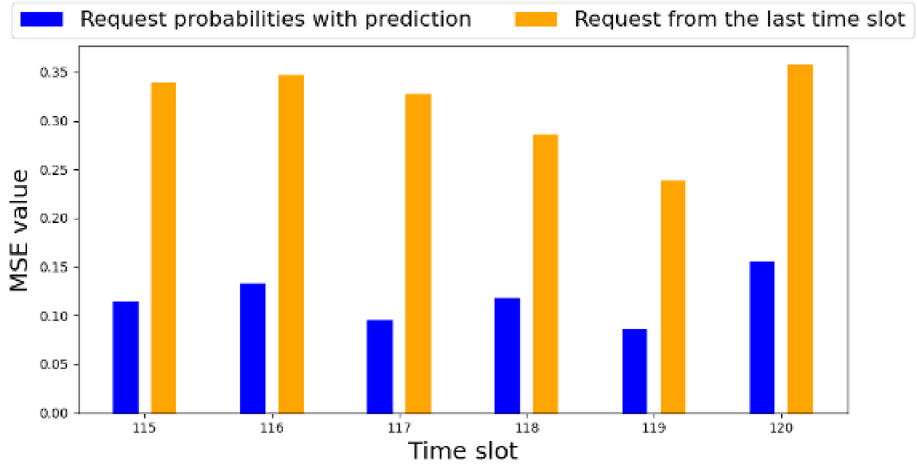


P1: Request Prediction – Method & Simulation Result

Average request data in the past *w* time slots as the probability of the next time slot

$$P_u^s(t) = \frac{1}{w} \sum_{x=t-w}^{t-1} R_u^s(x)$$

⇒ More similar to the real request data



P2: Service Assignment Decision

Goal: Assign requests to appropriate compute nodes in advance for **optimizing resource efficiency** and **fast service provision** based on

Original

$$v^s_{c,u}(t) = prior^s \times R^s_u(t) \times Q^s_{c,u}$$

$$\max_D \sum_{t=0}^{q} \sum_{u\in U} \sum_{c\in C} \sum_{s\in S} v^s_{c,u}(t) \times d^s_{c,u}(t)$$

Estimated

$$\overline{v^s_{c,u}(t)} = prior^s \times P^s_u(t) \times \overline{Q^s_{c,u}}$$

$$\max_D \sum_{u\in U} \sum_{c\in C} \sum_{a\in S} \overline{v^s_{c,u}(t)} \times d^s_{c,u}(t)$$

Estimated QoE Models

- **Exact RTT ⇒ Average RTT**
- **Exact execution time ⇒ Average execution time**

Original

$$Q_{c,u}^{SECC} = \begin{cases} \frac{1}{\frac{1}{n}\sum_{i=1}^{n}(RTT_{c,u}+t_i^{SECC})} & \text{if no timeout} \\ -1 & \text{if timeout} \end{cases}$$

$$Q_{c,u}^{CSD} = \beta \times \frac{1}{\frac{1}{n}\sum_{i=1}^{n}(RTT_{c,u}+t_i^{CSD})} + (1-\beta)\times 0.72$$

$$Q_{c,u}^{CSM} = \gamma \times \frac{1}{\frac{1}{n}\sum_{i=1}^{n}(RTT_{c,u}+t_i^{CSM})} + (1-\gamma)\times 0.9624$$

$$Q_{c,u}^{VS} = \alpha X_u - (1-\alpha) I^a = \alpha X_u - (1-\alpha)(RTT_{c,u}+t^{VS})$$

Estimated

$$\overline{Q_{c,u}^{SECC}} = (1 - prob_{timeout}) \times \frac{1}{\overline{RTT_{c,u}}+\overline{t^{SECC}}} + prob_{timeout} \times (-1)$$

$$\overline{Q_{c,u}^{CSD}} = \beta \times \frac{1}{\overline{RTT_{c,u}}+\overline{t^{CSD}}} + (1-\beta)\times 0.72$$

$$\overline{Q_{c,u}^{CSM}} = \gamma \times \frac{1}{\overline{RTT_{c,u}}+\overline{t^{CSM}}} + (1-\gamma)\times 0.9624$$

$$\overline{Q_{c,u}^{VS}} = \alpha X'_u - (1-\alpha)(\overline{RTT_{c,u}}+\overline{t^{VS}})$$

P2: Service Assignment Decision – Methods

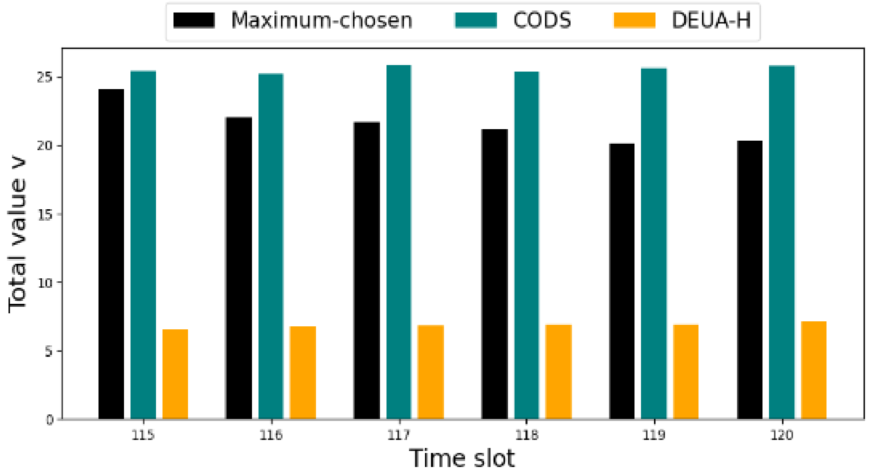
Method 1: Maximum-chosen Algorithm

- Greedy based
- Choose the one with maximum estimated *v* value if required resources are available

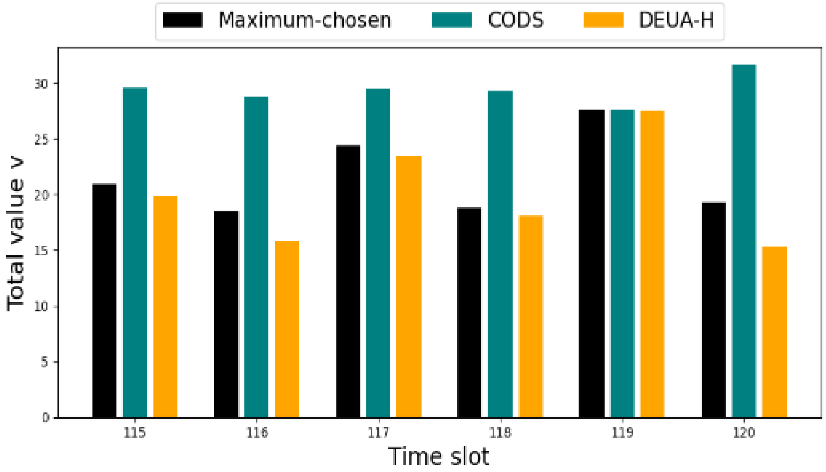
Method 2: Collaborative Optimal Decision Search (CODS)

- Integer linear programming (ILP) based
- Formulate objective function and constraints into an ILP problem

P2: Service Assignment Decision – Simulation Result



Predicted probabilities as inputs



Actual requests as inputs

CODS has the best performance.

Conclusion

- Proposing a scenario combining EV parking lots and an edge computing system
    - **IEEE 1935 standard**
- Formulate (estimated) QoE models for four frequently used services in EV parking scenario
- **Request prediction + Maximum-chosen / CODS method ⇒ Fast service provisions + resource efficiency maximization**

References

- [1] J. R. Bhat and S. A. Alqahtani, “6g ecosystem: Current status and future perspective,” IEEE Access, vol. 9, pp. 43 134–43 167, 2021.
- [2] S. S. Ravi and M. Aziz, “Utilization of electric vehicles for vehicle-togrid services: Progress and perspectives,” Energies, vol. 15, no. 2, p.589, 2022.
- [3] Y. Chiang, Y. Zhang, H. Luo, T.-Y. Chen, G.-H. Chen, H.-T. Chen, Y.-J. Wang, H.-Y. Wei, and C.-T. Chou, “Management and orchestration of edge computing for iot: A comprehensive survey,” IEEE Internet of Things Journal, 2023.
- [4] C. Lee, S. Park, T. Yang, and S.-H. Lee, “Smart parking with finegrained localization and user status sensing based on edge computing,” in 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), 2019, pp. 1–5.
- [5] X. Huang, P. Li, R. Yu, Y. Wu, K. Xie, and S. Xie, “Fedparking: A federated learning based parking space estimation with parked vehicle assisted edge computing,” IEEE Transactions on Vehicular Technology, vol. 70, no. 9, pp. 9355–9368, 2021.
- [6] A. Luo, J. Yuan, F. Liang, Q. Yang, and D. Mu, “Load forecasting of electric vehicle charging station based on edge computing,” in 2020 IEEE 3rd International Conference on Computer and Communication Engineering Technology (CCET), 2020, pp. 34–38.
- [7] ETSI, “Mobile edge computing (mec); framework and reference architecture,” ETSI, DGS MEC, vol. 3, pp. 1–18, 2016.
- [8] IEEE standard for edge/fog manageability and orchestration, IEEE standard 1935-2023, 2023.

References

- [9] T.-Y. Chen, Y. Chiang, J.-H. Wu, H.-T. Chen, C.-C. Chen, and H.-Y. Wei, “Ieee p1935 edge/fog manageability and orchestration: Standard and usage example,” in 2023 IEEE International Conference on Edge Computing and Communications (EDGE), 2023, pp. 96–103.
- [10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.
- [11] A. B. Shetty, J. Rebeiro et al., “Facial recognition using haar cascade and lbp classifiers,” Global Transitions Proceedings, vol. 2, no. 2, pp.330–335, 2021.
- [12] Y. Chiang, C.-H. Hsu, and H.-Y. Wei, “Collaborative social-aware and qoe-driven video caching and adaptation in edge network,” IEEE Transactions on Multimedia, vol. 23, pp. 4311–4325, 2021.
- [13] Z. Xu, G. Zou, X. Xia, Y. Liu, Y. Gan, B. Zhang, and Q. He, “Distanceaware edge user allocation with qoe optimization,” in 2020 IEEE International Conference on Web Services (ICWS). IEEE, 2020, pp. 66–74.

Thank you very much for listening.

Appendix – Parameters Settings

TABLE I
PARAMETER SETTINGS.

Parameter	Value	Parameter	Value
$prior^{SECC}$	$4/9$	$prior^{CSD}$	$2/9$
$prior^{CSM}$	$2/9$	$prior^{VS}$	$1/9$
$prob_{timeout}$	0.01	$\overline{t^{SECC}}$	9 (ms)
$\overline{\tau^{SECC}}$	0.79 (core/user)	$\overline{\omega^{SECC}}$	366 (MB/user)
$\overline{dl^{SECC}}$	86 (kbps/user)	$\overline{ul^{SECC}}$	21 (kbps/user)
β	0.9	$\overline{t^{CSD}}$	5244.8 (ms)
$\overline{\tau^{CSD}}$	0.23 (core/user)	$\overline{\omega^{CSD}}$	430 (MB/user)
$\overline{dl^{CSD}}$	2.1 (kbps/user)	$\overline{ul^{CSD}}$	327 (kbps/user)
γ	0.7	$\overline{t^{CSM}}$	768.2 (ms)
$\overline{\tau^{CSM}}$	0.17 (core/user)	$\overline{\omega^{CSM}}$	49.3 (MB/user)
$\overline{dl^{CSM}}$	2.9 (kbps/user)	$\overline{ul^{CSM}}$	1.09 (Mbps/user)
α	$5/7$	$\overline{t^{VS}}$	0 (ms)
$\overline{\tau^{VS}}$	0.0033 (core/user)	$\overline{\omega^{VS}}$	14.1 (MB/user)
$\overline{ul^{VS}}$	59 (kbps/user)	τ_c	4 (cores)
ω_c	32 (GB)	ul_c	1 (Gbps)
dl_c	1 (Gbps)	ul_u	100 (Mbps)
dl_u	100 (Mbps)	w	6

Appendix – Maximum Chosen Algorithm

Algorithm 1 Maximum-chosen Algorithm

Input: $P_u^s,\ \forall s \in S, \forall u \in U$

Output: Total value $value$ and service deployment and assignment decisions D

Let $value \leftarrow 0$

Initialize $\overline{V} = \{\overline{v_{c,u}^s} = prior^s \times P_u^s \times \overline{Q_{c,u}^s} \mid \forall s \in S, \forall c \in C, \forall u \in U\}$

Initialize $D = \{d_{c,u}^s = 0 \mid \forall s \in S, \forall c \in C, \forall u \in U\}$

Initialize $RC = \{rc_c = \{\tau_c, \omega_c, dl_c, ul_c\} \mid \forall c \in C\}$

Initialize $RE = \{re_u = \{dl_u, ul_u\} \mid \forall u \in U\}$

while There exist non-zero value in set $\overline{V}$ **do**

    Find maximum value $\overline{v_{c',u'}^{s'}}$ in set $\overline{V}$

    if CheckResourceEnough(s', c', u') **then**

        $value \leftarrow value + \overline{v_{c',u'}^{s'}}$

        $d_{c',u'}^{s'} \leftarrow 1$

        for $c \in C$ **do**

            $\overline{v_{c,u'}^{s'}} \leftarrow 0$

        end for

    else

        $\overline{v_{c',u'}^{s'}} \leftarrow 0$

    end if

end while

function CheckResourceEnough(s, c, u)

    if all value in $(rc_c - \{\overline{\tau^s}, \overline{\omega^s}, \overline{ul^s}, \overline{dl^s}\}) > 0$ **and** all value in $(re_u - \{\overline{dl^s}, \overline{ul^s}\}) > 0$ **then**

        $rc_c \leftarrow rc_c - \{\overline{\tau^s}, \overline{\omega^s}, \overline{ul^s}, \overline{dl^s}\}$

        $re_u \leftarrow re_u - \{\overline{dl^s}, \overline{ul^s}\}$

        return True

    else

        return False

    end if

end function