This article has been accepted for publication in IEEE Wireless Communications Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LWC.2023.3323421

Priority Aggregation Network with Integrated Computation and Sensation for Ultra Dense Artificial Intelligence of Things

Chao Ren, Zongrui He, Yingqi Li, Difei Cao, Haijun Zhang, *Fellow, IEEE*, and Arumugam Nallanathan, *Fellow, IEEE*

***Abstract*—In the realm of ultra dense Intelligence of Things, efficient utilization of computation and sensation (CAS) resources remains a challenge due to their uneven temporal and spatial distribution. Priority Aggregation Network (PAN) is introduced to employ priority as an intermediary to restore, combine, and exchange idle CAS resources' permissions for cross-temporal and cross-modal resource leasing. PAN awards priority to devices that lease their sensing and computing resources to others, facilitating efficient resource allocation. Numerical results indicate PAN maintains high cross-temporal task completion rates while conserving equipment resources, with increased willingness to pay for resources.**

***Index Terms*—Resources leasing; joint computation and sensation; resource management**

I. Introduction

The proliferation of Artificial Intelligence of Things (AIoT) devices has led to explosive growth, resulting in an increased demand for computational and sensory resources. Globally, the number of mobile devices is projected to exceed 70 billion, with a monthly traffic volume of over 25 exabytes. This exponential growth imposes a high task workload, but the utilization of computational and sensory (CAS) resources in AIoT devices remains uneven across time and space. For instance, some devices may have idle CAS capabilities during computing (and vice versa), while others may remain idle as a whole while charging. These idle resources may not require immediate utilization currently but could encounter significant workloads in the future, leading to resource wastage and constraints. While these resources may not currently require immediate computation or sensory resources, they could potentially encounter significant workloads in the future, leading to both resource wastage and constraints. Additionally, the heterogeneity and immediacy of CAS resources, which cannot be stored, integrated, or exchanged, further complicate addressing the uneven distribution of resources across temporal and spatial dimensions.

Various strategies have been explored to alleviate CAS task burdens in edge or dense networks. In Ref. [1], a game theory-based algorithm was introduced to offload traditional local computational tasks to the cloud. Considering AIoT devices, it is crucial to account for the energy consumption associated with computation tasks, as emphasized by the offloading algorithm detailed in Ref. [2]. Furthermore, studies have leveraged resource utilization across different layers through task offloading or task migration [3], [4] to counteract the uneven spatial distribution of computational resources. However, the dual requirements of AIoT entail not only robust computational prowess but also a versatile spectrum of sensors. To optimize resource allocation for sensation, an system offloads specific operations from the platform to the network edge with incentive for the workers to maximize their utility [5]. In Ref. [6] a collaborative sensing solution is explored for both ideal scenarios and real-world applications, addressing occlusion and sensor failure challenges in autonomous driving scenarios. The challenge remains of transcending the inherent heterogeneity of CAS resources, enabling seamless interaction and overcoming the limitations imposed by idle CAS resources.

To address this issue, we propose a priority aggregation network with integrated computation and sensation (PANICAS). PANICAS enables the integration and exchange of CAS capabilities by utilizing priority variables as intermediary transaction bridges, thereby mitigating the impact of heterogeneous CAS resources. It effectively leverages priority-based storage to aggregate the contributions of current CAS devices to the entire network in exchange for assistance with their own tasks. This transforms the challenge of 'storing and exchanging CAS resources' into a priority accumulation and consumption problem, effectively addressing the immediacy and uneven utilization across temporal and spatial domains. PANICAS establishes low-latency device groups, fostering resource-sharing dynamics and prioritizing high-demand devices within the CAS framework.

II. System Model

Within the PANICAS framework (see Figs. 1 and 2), AIoT devices have logically adjacency or experience time delays lower than τ. Each device within a range of size S has position information ($d_{x,j}$, $d_{y,j}$). CAS Priority Leasing (CASPL)

This work is supported in part by the National Natural Science Foundation of China (Grant No. 62201034, U22B2003 and 62341103), the Beijing Municipal Natural Science Foundation (Grant No. L212004-03), the Beijing Natural Science Foundation (Grant No. L212004), and Xiaomi Fund of Young Scholar (Corresponding author: Haijun Zhang.)

C. Ren, Z. He, Y. Li, D. Cao and Haijun Zhang are with the Beijing Engineering and Technology Research Center for Convergence Networks and Ubiquitous Services, University of Science and Technology Beijing, Beijing, 100083, China (e-mail: chaoren@ustb.edu.cn; m202310612@xs.ustb.edu.cn; erin_qi@foxmail.com; caodifei@hotmail.com; haijunzhang@ieee.org).

A. Nallanathan is with King's College London, London, UK. (e-mail: arumugam.nallanathan@kcl.ac.uk)

Manuscript received XXX; revised XXX.

This article has been accepted for publication in IEEE Wireless Communications Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LWC.2023.3323421

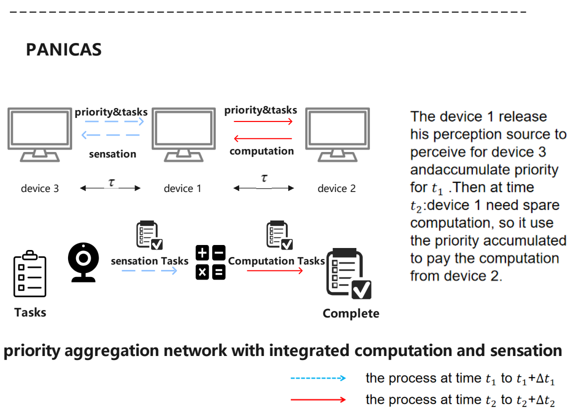

Fig. 1. System model of PANICAS.

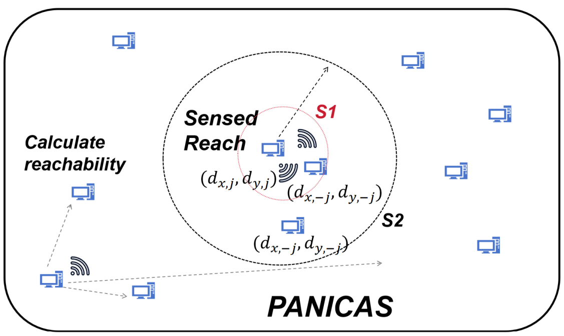

Fig. 2. Sensation and computation in PANICAS.

involves receivers prioritizing providers, and providers accumulating priority variables, facilitating the provision of computational or sensory aid. Fig. 1 exemplifies resource invocation over time and modalities, as device 1 transitions its role from a provider to a receiver at slots t_1 and t_2. Assuming a total of N CAS users, each CAS user j is identified with the ID d_j ($j \in [1, N]$), while $-j$ represents all other user numbers except d_j. CAS resources are merged and treated as a unified virtual resource set, with the minimum resource unit identified as $i \in \{\text{Type1Sense}, \text{Type2Compt}\}$, characterizing resources of different types/modalities. Each task T for each user may contain sub-tasks from the CAS type set, where $A_{i,j}$ represents the resource (i) amount required to complete the task for d_j.

During a CASPL slot, d_j provides an amount of $R_{\text{out},i,j}$ CAS resource i to other users d_{-j} with priority conversion parameter $P_{\text{out},i,j}$ (sale price, e.g., per MB/CPU Circle), and leases an amounts of $R_{\text{in},i,j}$ CAS resources i from users d_{-j} with priority conversion parameter $P_{\text{in},i,j}$ (purchase price). For all the CAS resource $i \in \{\text{Type1Sense}, \text{Type2Compt}\}$, the CAS priority accumulation value (token) for user d_j is denoted as T_{d_j}. The fixed amount of internal CAS resource i for user d_j is denoted as $R_{i,j}$. Thus, the demand of external CAS resource i by the execution task of user d_j is

$$DR_{i,j} = A_{i,j} - R_{i,j}, \tag{1}$$

where $DR_{i,j} > 0$ means d_j requires external resource, and $DR_{i,j} < 0$ means d_j can provide idle internal resource.

In a CASPL slot, d_j is willing to provide a portion $\alpha_{i,j} = \frac{R_{\text{out},i,j}}{R_{i,j}} \in [0, 1]$ of its fixed internal resource i based on the demand of both d_{-j} and its own strategy. Similarly, d_j is willing to use a portion $\beta_{i,j} \in [0, 1]$ of its token to lease external resource i based on the available inventory and its own strategy. We refer to $\alpha_{i,j}$ and $\beta_{i,j}$ as willingness factors.

Due to factors introduced by the CASPL process, such as transmission delay and CASPL execution, utilizing external CAS resources for remote sensing, controlling, or processing results in the services provided by d_{-j} being a discounted version of d_j's own CPU and sensors. To model this effect, we introduce a penalty factor. To account for transmission and CASPL execution delay, as well as remote control and processing of external CAS resources, we use $\Delta\tau_{i,j,-j}$ to assess the extra delay introduced by CASPL for user d_j, where $-j$ means all the CAS service providers except d_j. Note that $\Delta\tau_{i,j,j}$ means d_j uses its own CAS resource i with minimum extra delay. Thus, the penalty factor for relative total delay is then defined as:

$$\tau^D_{i,j,-j} = \frac{\Delta\tau_{i,j,-j}}{\Delta\tau_{i,j,j}} > 1. \tag{2}$$

To handle the heterogeneity of modalities, demand, and inventory information, selectivity factors are required for different situations. Sensing tasks are influenced by geographical factors and can be categorized into quantitative and qualitative perspectives, with the qualitative aspect further subdivided into two Levels: in non-splittable task level, task can only be accomplished by the device itself; in splittable task level, devices falling within range S_1 constitutes normal transmissions characterized by the same relative latency as the computation task. However, task decomposition in S_2 leads to higher latency, which is manifested in $\tau^D_{i,j,-j}$. Ensuring the target to be sensed, sensory resources providers from d_{-j} should satisfy the geographic location condition within a range S_j defined by user d_j's sensory task. The selectivity requires a filter $*$ to choose the corresponding users. Considering d_j's willingness to consume token and the total idle resource from other users, we introduce an updated version of $\beta_{i,j}$ as Equ. (3).

$$R_{\text{in},i,j} = \min\left(\frac{\beta_{i,j}}{\tau^D_{i,j,-j}}, \frac{\sum_{k\in *}\max(0, -DR_{i,k})}{T_{d_j}\tau^D_{i,j,-j}}\right)\frac{T_{d_j}}{P_{\text{in},i,j}}, \tag{3}$$

where $* = S_j$ if i = Type1Sense, and $* = -j$ if i = Type2Compt. In an ultra-dense AIoT scenario, nearby massive CAS users can provide superfluous CAS resources and priority token for d_j, resulting in $\sum_{k\in *}\{\cdot\} \to +\infty$. This yields $\beta^{Sel}_{i,j,-j} = \frac{\beta_{i,j}}{\tau^D_{i,j,-j}}$. Therefore, based on Equ. (3), we have a more simplified expression for $R_{\text{in},i,j}$ as

$$R^{\text{UD}}_{\text{in},i,j} = \frac{\beta_{i,j}}{\tau^D_{i,j,-j}}\frac{T_{d_j}}{P_{\text{in},i,j}}. \tag{4}$$

This means in an ultra-dense AIoT scenario, the amount of resource i to be provided (sold) is determined by willingness, tokens, delay and the priority conversion parameter (price).

III. Discussion and Analysis

A. Case I: $D_{i,j} > 0$

If $DR_{i,j} > 0$, it indicates that external resource i is required to complete the task, and d_j should receive more of this external resource i by adjusting $\alpha i, j$ and $\beta_{i,j}$. Therefore, in the CASPL process, it isnecessary to adhere to the constraint that $R_{\text{in},i,j} - R_{\text{out},i,j} \geq DR_{i,j}$. Based on Equ. (4) we have the condition for $R_{\text{out},i,j}$ as

$$0 \leq R_{\text{out},i,j} \leq \frac{\beta_{i,j}}{\tau^D_{i,j,-j}} \frac{T_{d_j}}{P_{\text{in},i,j}} - A_{i,j} + R_{i,j}. \tag{5}$$

Furthermore, based on Equ. (5), the amount of resource i needed to complete the task must satisfy

$$0 \leq A_{i,j} \leq \frac{\beta_{i,j}}{\tau^D_{i,j,-j}} \frac{T_{d_j}}{P_{\text{in},i,j}} + R_{i,j}. \tag{6}$$

This means that user d_j cannot accept large tasks that do not meet the condition in Equ. (6). Moreover, by using the definition of of $\alpha_{i,j}$, we can further specify its bounds as

$$0 \leq \alpha_{i,j} \leq \min\left(1, \frac{\beta_{i,j}}{\tau^D_{i,j,-j}} \frac{T_{d_j}}{P_{\text{in},i,j} R_{i,j}} - \frac{A_{i,j}}{R_{i,j}} + 1\right). \tag{7}$$

To fully characterize $R_{\text{out},i,j}$, we need to consider various factors such as d_j's willingness factor to provide i to other users d_{-j}, the total amount of priority token that other users willing to provide, and the total resource demand from other users. An updated version of $\alpha_{i,j}$, denoted as $\alpha^{Sel}_{i,j,-j}$ is presented in Equ. (8).

$$\begin{aligned}\alpha^{Sel}_{i,j,-j} &= \min\left(\alpha_{i,j}, \frac{\sum_{k\in *} \beta_{i,k} T_{d_k}}{P_{\text{out},i,j}}, \sum_{k\in *} \max(0, \tau^D_{i,j,*}\alpha_{i,k} R_{i,k})\right). \\ R_{\text{out},i,j} &= \alpha^{Sel}_{i,j,-j} R_{i,j}\end{aligned} \tag{8}$$

In the context of an ultra-dense AIoT Scenario, where $\sum_{k\in *}\{\cdot\} \to +\infty$, we have $\alpha^{Sel}_{i,j,-j} = \alpha_{i,j}$. Thus, based on Equ. (8), we have a simplified expression for $R_{\text{out},i,j}$ as

$$R^{\text{UD}}_{\text{out},i,j} = \alpha_{i,j} R_{i,j}. \tag{9}$$

In a specific case where d_j requires one external resource i_1 to complete the task (i.e., $D_{i_1,j} > 0$), and there are no limitations on the other resource i_2, d_j can utilize the following strategy to maximize its amount of the resource i_1 in shortage. In the first stage of CASPL, d_j provides both i_1 and i_2 to accumulate token and potentially receive more i_1 by consuming tokens to complete other sub-tasks. An update is made to the amount of token as $T^{'}_d = T_d + \sum_i R^{\text{UD}}_{\text{out},i,j} P_{\text{out},i,j} - R^{\text{UD}}_{\text{in},i_2,j} P_{\text{in},i_2,j}$. In the second stage, d_j uses all the updated tokens to purchase i_1. Finally, based on Equ. (4), the total amount of i_2 owned by d_j is determined considering the shortage of i_1 and the difference between the sale price and purchase price of i_1 (see Equ. (10)).

$$\begin{aligned} R^{total}_{i_1,j} =& (1-\alpha_{i_1,j}) R_{i_1,j} + R^{\text{UD}}_{\text{in},i_1,j} \\ =& (1-\alpha_{i_1,j}) R_{i_1,j} + \frac{\beta_{i_1,j}}{\tau^D_{i_1,j,-j}} \frac{T^{'}_{d_j}}{P_{\text{in},i_1,j}} \\ =& \left(\frac{\beta_{i_1,j}}{\tau^D_{i_1,j,-j}} \frac{P_{\text{out},i_1,j}}{P_{\text{in},i_1,j}} - 1\right) \alpha_{i_1,j} R_{i_1,j} + \\ & \left(1 - \frac{\beta_{i_2,j}}{\tau^D_{i_2,j,-j}}\right) \beta_{i_1,j} \frac{T_{d_j}}{\tau^D_{i_1,j,-j} P_{\text{in},i_1,j}} + \\ & \frac{\alpha_{i_2,j}\beta_{i_1,j}}{\tau^D_{i_1,j,-j}} \frac{P_{\text{out},i_2,j}}{P_{\text{in},i_1,j}} R_{i_2,j} + R_{i_1,j}. \end{aligned} \tag{10}$$

Based on Equ. (2) and $\beta_{i_2,j} \in [0,1]$, it is evident that only the first term in $R^{total}_{i_1,j}$ can be negative. Thus, by considering the scenarios where the sale price of i_1 is higher or lower than $\tau^D_{i_1,j,-j}$ times the purchase price, we can find following optimal strategy to maximize the amount of resource i_1.

1) Case $P_{out,i_1,j} \geq \tau^D_{i_1,j,-j} P_{in,i_1,j}$: If $\frac{\tau^D_{i_1,j,-j} P_{\text{in},i_1,j}}{P_{\text{out},i_1,j}} < \beta_{i_1,j} \leq 1$, d_j chooses to maximize the available amount of i_1 by setting the maximum value of $\beta_{i_1,j} = 1$, $\alpha_{i_1,j} = \min\left(1, \frac{1}{\tau^D_{i_1,j,-j}} \frac{T_{d_j}}{P_{\text{in},i_1,j} R_{i_1,j}} - \frac{A_{i_1,j}}{R_{i_1,j}} + 1\right)$ and $\alpha_{i_2,j} = 1$, while setting the minimum value of $\beta_{i_2,j} = 0$. This results in $R^{total,max}_{i_1,j} = \left(\frac{1}{\tau^D_{i_1,j,-j}} \frac{P_{\text{out},i_1,j}}{P_{\text{in},i_1,j}} - 1\right) \alpha_{i_1,j} R_{i_1,j} + \frac{T_{d_j}}{\tau^D_{i_1,j,-j} P_{\text{in},i_1,j}} + \frac{1}{\tau^D_{i_1,j,-j}} \frac{P_{\text{out},i_2,j}}{P_{\text{in},i_1,j}} R_{i_2,j} + R_{i_1,j}$

If $0 \leq \beta_{i_1,j} \leq \frac{\tau^D_{i_1,j,-j} P_{\text{in},i_1,j}}{P_{\text{out},i_1,j}}$, d_j chooses to maximize the available amount of i_1 by setting the maximum value of $\beta_{i_1,j} = \frac{\tau^D_{i_1,j,-j} P_{\text{in},i_1,j}}{P_{\text{out},i_1,j}}$ and $\alpha_{i_2,j} = 1$, while setting the minimum value of $\alpha_{i_1,j} = 0$ and $\beta_{i_2,j} = 0$. This results in $R^{total,max}_{i_1,j} = \frac{T_{d_j}}{P_{\text{out},i_1,j}} + R_{i_2,j} + R_{i_1,j}$

2) Case $P_{out,i_1,j} < \tau^D_{i_1,j,-j} P_{in,i_1,j}$: In this case, $0 \leq \beta_{i_1,j} \leq 1 < \frac{\tau^D_{i_1,j,-j} P_{\text{in},i_1,j}}{P_{\text{out},i_1,j}}$, and t$d_j$ chooses to maximize the available amount of i_1 by setting the maximum value of $\beta_{i_1,j} = 1$ and $\alpha_{i_2,j} = 1$, while setting the minimum value of $\alpha_{i_1,j} = 0$ and $\beta_{i_2,j} = 0$, resulting in $R^{total,max}_{i_1,j} = \frac{T_{d_j} + P_{\text{out},i_2,j} R_{i_2,j}}{\tau^D_{i_1,j,-j} P_{\text{in},i_1,j}} + R_{i_1,j}$

3) Conclusions: Based on the derived willingness factors, in the first CASPL stage: all of resource i_2 is sold in exchange for tokens to purchase i_1. For resource i_1, if the sale price of i_1 is higher than $\tau^D_{i_1,j,-j}$ times purchase price, d_j has two strategies: A) Adventurist strategy: Sell a fraction of i_1 based on $\alpha_{i_1,j} = \min\left(1, \frac{1}{\tau^D_{i_1,j,-j}} \frac{T_{d_j}}{P_{\text{in},i_1,j} R_{i_1,j}} - \frac{A_{i_1,j}}{R_{i_1,j}} + 1\right)$, and then use all token to purchase i_1; B) Conservative strategy: Use a fraction of the accumulated token based on $\beta_{i_1,j} = \frac{\tau^D_{i_1,j,-j} P_{\text{in},i_1,j}}{P_{\text{out},i_1,j}}$ to purchase i_1.

In this example, we can observe the cross-spatial domain and cross-modal usage of CAS resources.

B. Case II: $D_{i,j} \leq 0$

In this case, typically, the task load is low, and there is no extreme shortage of resources. Users do not prefer to acquire the maximum available resources. Instead, they can accumulate priority tokens. The exchange value after accumulating priority is given by

$$\int_t^{t+\Delta t} \left(\frac{P^t_{\text{out},i_1,j}}{P^{t+\Delta t}_{\text{in},i_2,j}} \alpha_{i_1} R_{i_1,j} + \frac{P^t_{\text{out},i_2,j}}{P^{t+\Delta t}_{\text{in},i_2,j}} \alpha_{i_2} R_{i_2,j}\right) dt. \tag{11}$$

This shows that even the previously limited resource i_1 with an inventory of $R_{i_2,j}$ can be accumulated across domains and time by transforming into token.

IV. CASPL ALGORITHMS

The CASPL process comprises two distinct process: sensory/computing resource matching and priority settlement.

A. CASPL Algorithm

For sensory resource matching, we consider the influence of geographic factors on the sensation demand, resulting in the filtration of a subset of CAS resources, denoted as i_1, from the resource set. Subsequently, we apply Algorithm 1 to perform the matching. The expression for the actual resource requirement to be procured is $\frac{\beta_{i_1,j} \cdot T_{d_j}}{Pin_{i_1,j} \cdot \tau^D_{i_1,j,-j}}$.

Algorithm 1 Sensary Resource Matching

Require: In slot t, user d_j with coordinates $(d_{x,j}, d_{y,j})$ receives the sensory task $d_{T_1,i_1,j}$.

If: $DR_{i_1,j}$=$A_{i_1,j}$-$R_{i_1,j}$ >0

Then: Consume priority $\rightarrow$ Obtain i_1

End If

Ensure:

Condition: $(d_{X,j}, d_{Y,j}) \in S_j$, $d = \{d_{j_1}, d_{j_2}, \cdots, d_{j_n}\}$
Resources: $I_1 = \{-DR_{i_1,j_1}, -DR_{i_1,j_2}, \cdots, -DR_{i_1,j_n}\}$

If :$T_{d,j} - \frac{DR_{i_1,j}}{Pin_{i_1,j} \cdot \tau^D_{i_1,j,-j}}$>0, Then: Next

  Else: Sell other resource until it is greater than 0

End If

Descending order yields $\alpha_{i_1,j_{max}} R_{i_1,j_{max}}$

While: $(-\alpha_{i_1,j} \cdot R_{i_1,j_{max}}) - \frac{\beta_{i_1,j} \cdot T_{d_j}}{Pin_{i_1,j} \cdot \tau^D_{i_1,j,-j}}$>0, Then: Next

  Repeat: Plus the next largest resource.

Until >0, break.

Finally: $\frac{\beta_{i_1,j} \cdot T_{d_j}}{Pin_{i_1,j} \cdot \tau^D_{i_1,j,-j}} = \sum_{j_1}^{j_n}(-\alpha_{i_1,j_m} \cdot DR_{i_1,j_m})$,where $0 < m \leq n$.

In computational resource matching, geography is not a consideration. We can directly repeat Algorithm 1 after removing the condition: $(d_{X,j}, d_{Y,j}) \in S_j$.

B. Priority Settlement

After each exchange process, both the provider and the receiver should update their priorities.

Algorithm 2 Priority settlement

Require: Before

For the receiver: $priority = T^t_{d_j}$

For the provider:$\{d_{j_1}, d_{j_2}, ..., d_{j_n}\}$,$\left\{T^t_{d_{j_1}}, T^t_{d_{j_2}}, ..., T^t_{d_{j_n}}\right\}$

Ensure: After

For the receiver:$T^{t+\Delta t}_{d_j} = T^t_{d_j} - \beta_{i,j} \cdot T^t_{d_j}$

For the provider:$\left\{T^{t+\Delta t}_{d_{j_1}}, T^{t+\Delta t}_{d_{j_2}}, ..., T^{t+\Delta t}_{d_{j_n}}\right\}$
general:$T^{t+\Delta t}_{d_{j_m}} = T^t_{d_{j_m}} + \frac{-\alpha_{i,j_m} \cdot R_{i,j_m}}{Pout_{i,j_m}}$,,where $0 < m \leq n$.

In Algorithm 2, following each exchange process, both the provider and the receiver must update their priorities. In addition, recording the priority transaction amount and the priorities held after the transaction enables priority settlement.

V. Numerical Results

In simulation, 30 AI devices are distributed with integer coordinates in a (10×10) rectangular grid. Within S_1, $\tau^D_{i_1,j,-j} = 3$, while within S_2, $\tau^D_{i_1,j,-j} = 5$. CASPL is compared with the non-lease model by assessing the impact of the β willingness factor on the completion rates of sensory and computational tasks over a period, where $R_i = [10, 12, 15]$, $A_i = [8, 11, 14]$, $\tau^D_{i_1,j,-j} = 3$, $Pin_i, Pout_i = [1, 1.2, 1.5]$, $R_i = [10, 12, 15]$, $A_i = [8, 11, 14]$ and $Pin_i, Pout_i = [1, 1.2, 1.5]$. Fig. 3 illustrates that an increase in β, indicative of a higher proportion of individuals willing to pay for resources, results in an increased likelihood of task completion. However, the geographic environment's limiting effect on perception is less pronounced compared to computational tasks. Furthermore, we consider a scenario where the resource selling price exceeds the entry price, incorporating an adventurous factor, $a_i = 0.1$. Cross-time sensing and calculation task completion involve maintaining task completion rates higher than those of the no-lease model over time. This is achieved by increasing resource utilization within the system while keeping equipment resources constant.

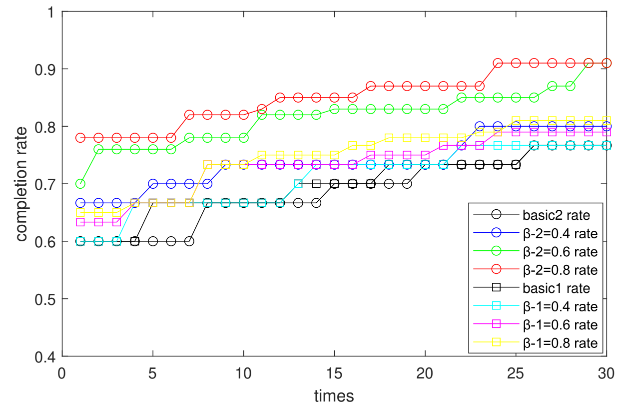

Fig. 3. Cross-time Sensing and calculating task completion rates

VI. Conclusion

In this paper, we consider the uneven distribution of resources and propose a PANICAS concept that considers the limitations of smart devices in sensation and computation, where tokens are used as an intermediary medium to leasing heterogeneous resources cross-time and cross-modal.

References

[1] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, April 2015.

[2] F. Samie, V. Tsoutsouras, L. Bauer, S. Xydis, D. Soudris, and J. Henkel, "Computation offloading and resource allocation for low-power iot edge devices," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Dec 2016, pp. 7–12.

[3] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587–597, March 2018.

[4] A. Boukerche, S. Guan, and R. E. De Grande, "A task-centric mobile cloud-based system to enable energy-aware efficient offloading," *IEEE Transactions on Sustainable Computing*, vol. 3, no. 4, pp. 248–261, Oct 2018.

[5] C. Ying, H. Jin, X. Wang, and Y. Luo, "Chaste: Incentive mechanism in edge-assisted mobile crowdsensing," in *2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, June 2020, pp. 1–9.

[6] Y. Han, H. Zhang, H. Li, Y. Jin, C. Lang, and Y. Li, "Collaborative perception in autonomous driving: Methods, datasets, and challenges," *IEEE Intelligent Transportation Systems Magazine*, pp. 2–22, 2023.