Asynchronous Federated Learning Technique for Latency Reduction in STAR-RIS enabled VRCS

Shivam Chaudhary*, Ishan Budhiraja*, Rajat Chaudhary*, Neeraj Kumar†, and Sujit Biswas‡
*School of Computer Science Engineering and Technology, Bennett University, Greater Noida, India
†Computer Science & Engineering, Thapar Institute of Engineering and Technology, Patiala, Punjab
‡Cybersecurity and FinTech, Computer Science Department, City, University of London, and Research Associate (Honorary) at CBT, University College London (UCL), UK
Emails: e22soep0063@bennett.edu.in, ishan.budhiraja@bennett.edu.in, rajat.chaudhary@bennett.edu.in, neeraj.kumar@thapar.edu, sujit.biswas@city.ac.uk

***Abstract*—With the advent of smart and autonomous vehicles, a number of novel data-intensive and latency-critical vehicular communication applications have emerged. However, dynamic vehicular mobility and urban environments introduce severe propagation challenges, leading to increased latency. In order to reduce latency in Vehicle Road Cooperative Systems (VRCS), this research introduces a unique architecture that combines Asynchronous Federated Learning (AFL) with Simultaneously Transmitting and Reflecting Reconfigurable Intelligent Surfaces (STAR-RIS). The proposed system leverages a Markov Decision Process (MDP)-based optimization framework to minimize latency by jointly optimizing STAR-RIS elements and offloading decisions. Our approach allows vehicles to asynchronously update global models, ensuring robust learning while adapting to dynamic network conditions. The simulation results show that the recommended strategy provides at least a 20% reduction in latency in AFL when compared to FL.**

***Index Terms*—Latency Reduction, STAR-RIS, VRCS, V2X, AFL.**

I. INTRODUCTION

The need for quick and low-latency communication is essential in the rapidly developing field of intelligent transportation systems (ITS). However, The quality of the propagation in Vehicle to everything (V2X) communication links is frequently degraded as a result of the channels' rapid varying by the high versatility of vehicles and the intricacy of the cities communication environment, like high structures obstructing the channels [1]. New technologies like STAR-RIS offer a novel approach to improving vehicular communication network propagation quality. The ability of STAR-RIS to control electromagnetic waves in a dynamic manner improves signal strength, coverage, and reduces latency [2]. Autonomous driving, collision avoidance, and traffic management all rely on real-time data exchange, so this capability is especially useful in VRCS. In VRCS, federated learning (FL) opens the door to minimizing latency [3].

FL enables decentralized data processing and model training across multiple edge vehicles, in contrast to conventional centralized machine learning techniques. This not only preserves data privacy but also mitigates the need for extensive data transmission to a central server, thereby significantly reducing latency [4]. FL makes it possible for vehicles and roadside units to work together to learn and improve system performance in the context of vehicular networks by facilitating the local updating and aggregation of models [5].

The integration of STAR-RIS with FL in VRCS addresses a synergistic way to deal with latency minimization. The proposed framework aims to improve communication efficiency and decrease latency by utilizing STAR-RIS's signal propagation enhancement capabilities and FL's decentralized nature.

A. Related Work

Recent research has explored the synergy between STAR-RIS and FL to address latency issues in vehicular networks. The authors in [6],[7] enhance the UAVs' transmission power via intelligent reflecting surfaces for V2X Communication. The authors in [8] demonstrated that deploying STAR-RIS in urban scenarios significantly improves signal propagation, resulting in reduced end-to-end communication delays. The STAR-RIS is crucial in modifying hybrid users' decoding sequence for effective interference reduction and omnidirectional coverage extension with federated learning [9]. In order to solve the issues of resource scheduling and transmission mode selection for the STAR-RIS infrastructure for virtual reality surveillance installed on drones, the authors offer a collaborative optimisation approach.

Authors in [10] took this a step further by integrating FL with RIS, showing that this combination can optimize both communication latency and model training efficiency. STAR-RIS assisted offloading scheme based on non orthognal multiple access (NOMA) can effectively reduce the latency compared with benchmark schemes [11]. The authors in [12],[13] takes into consideration a STAR-RIS assisted mobile edge computing(MEC) system and formulate a problem for the minimization of communication latency and weighted-sum computing in order to optimize the volume of offloading data, servers' edge computing resources. In [14],[15] researchers have looked into a number of strategies to deal with latency and energy management in integrated systems, such as resource allocation algorithms and adaptive beamforming, which allow STAR-RIS to dynamically modify its configuration in response to changes in the environment .

B. Motivation and Contribution

The reduction of latency, which is essential for real-time applications like collision avoidance and traffic management, faces new challenges when STAR-RIS is implemented in vehicular networks. Federated Learning (FL), with its decentralized nature, offers an expected arrangement by empowering vehicles to cooperatively gain proficiency with a common model while keeping their data local. This lowers latency and protects data privacy by requiring fewer frequent data transmissions to a central server.

- A latency-minimization strategy that takes into account the reconfigurability of the STAR-RIS and the dynamic nature of the vehicular network is incorporated into the framework. This procedure guarantees that the communication links are optimized and streamlined continuously, addressing the latency issues basic for V2X applications.
- The federated learning model is adjusted to the vehicular scenario, considering the versatility of vehicles, the varying network conditions, and the presence of STAR-RIS. The model empowers vehicles to cooperatively train on their local data, diminishing the requirement for centralized data aggregation and thus minimizing latency.

C. Organization

This research article is organized as: System model is described in Section II. The problem formulation and MDP formation is described in section III and Section IV describes proposed solution using federated learning. Section V representing the results and discussion, while the proposed article is concluded in Section VI.

II. System Model

In this section, $\mathcal{K} = \{k_1, k_2, \ldots, k_K\}$ is the set of MEC servers. Set of STAR-RIS is given by $\mathcal{J} = \{j_1, j_2, \ldots, j_J\}$. We analyse an STAR-RIS-equipped multi-vehicle scene where m vehicles are served by MEC-driven base station (BS) with help from STAR-RIS as shown in fig. 1. At time slot t, the m-th vehicle's trajectory coordinate is $\mathcal{G}_m = (i_m^t, j_m^t)$. While moving, the vehicle can select its offloading method. Vehicles with a maximum communication distance of x_0 can communicate with the STAR-RIS. Vehicle d will not communicate with STAR-IRS if it exceeds x_0. Directly from vehicle m to the MEC server (MS), the channel vector is represented as $\widehat{g}_{m,k}$, which comprises Non-Line-of-Sight (NLoS) $g_{m,k}^{NLOS}$ and Line-of-Sight (LoS) $g_{m,k}^{LoS}$ [16].

$$g_{m,k} = \sqrt{\nu x_{m,k}^{rho}}\left(\sqrt{\frac{\chi}{1+\chi}} g_{m,k}^{Los} + \sqrt{\frac{1}{1+\chi}} g_{m,k}^{Nlos}\right) \quad (1)$$

$$= \sqrt{\nu x_{m,k}^{\rho}} \widehat{g}_{m,k} \quad (2)$$

where $g_{m,k}^{Los}$ represents ULA's array response, χ stands for the Rician factor, ν for the factor of distortion, ρ = -4 for the

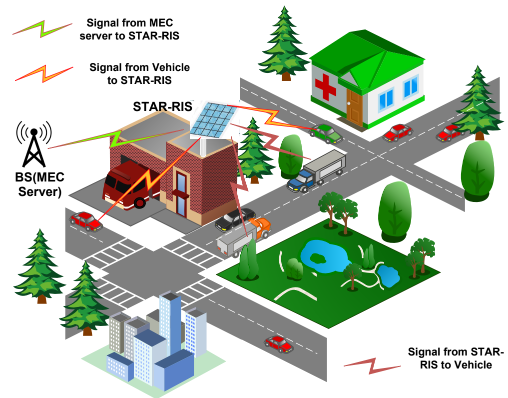

Fig. 1: System model.

fading factor of the path, and $g_{m,k}^{NLos}$ stands for the Complex Gaussian distribution with zero mean and unit variance. We can make every effort to address the aforementioned building-related signal fading issues with the use of STAR-RIS. From vehicle m to the MS, the channel vector g_m is expressed as

$$g_m = g_{m,k} + H\Phi g_{s,k} \quad (3)$$

where the channel gain between STAR-RIS and vehicles is represented by $H = x_{m,s}^{\rho}$. The gain of the channel under Rayleigh fading is represented by $g_{m,k}$, which is given by $\sqrt{\nu x_{m,k}^{\rho}} \widehat{g}_{m,k}$. The distance between the mth vehicle and the MS is given by $x_{m,k}$. The coefficient matrices at the STAR-RIS are denoted by $\Phi^{\nu} = diag(\sqrt{\lambda_1^{\nu}} \exp^{j\phi_1^{\nu}}, \sqrt{\lambda_2^{\nu}} \exp^{j\phi_2^{\nu}}, \ldots.. \sqrt{\lambda_E^{\nu}} \exp^{j\phi_E^{\nu}}), \nu \in \{S, F\}$ where $\phi_e^{\nu} \in [0, 2\pi)$,$\forall e \in \{1, 2, 3, \ldots, E\}$ represents the phase shift for the e-th element. The array STAR-RIS is made up of e × e reflecting components. S is the region consists of vehicles in transmission region and F is the region consists of vehicles in reflection region. The gain in the channel between STAR-RIS and the MS is denoted by $g_{s,k}$ and is given by $g_{s,k} = x_{s,k}^{\rho}$, where $x_{s,k}$ is the distance between MS and STAR-RIS. The overall duration for the MS execution includes processing and offloading tasks and transmission delays. The rate of transmission is provided by

$$R_{m,k}(t)$$

$$= W_{m,k}(t) log_2\left(1 + \frac{q_m(t)|v_m(t) g_m(t)|^2}{\sum_{r=1, r\neq m}^{w} q_r(t)|v_r(t) g_r(t)|^2 + \sigma_0}\right) \quad (4)$$

where the combined data rate from the vehicle to the STAR-RIS is indicated by $R_{m,k}$. The beamforming vector is denoted by $v_m(t)$ (e.g., $v_m^H(t) v_k(t) = 1$). The variance of the Gaussian noise is σ_0, and its mean is zero. The bandwidth between vehicle m and the MS is represented by $\mathcal{W}_{m,k}(t)$.

A. Execution Model

Vehicles must send vehicular tasks via STAR-IRS to the MS before the MS can begin executing the r-th partial task $Y_{m,r}$. The delay in transmission is provided by

$$L_{m,k,r}(t) = \frac{Y_{m,r}(t)}{R_{m,k}(t)} \tag{5}$$

The following represents the overall execution time:

$$L_{m,r}^{mce} = \frac{Y_{m,i}(t)O(t)}{u_s(t)} + L_{m,k,r}(t) + L^{past} \tag{6}$$

where O represent calculation density, u_s is the speed of MS at time t. The task data offloading is the primary cause of the overall delay for MS, as can be shown from the above calculations. Because the input is much higher compared to the output of task offloading, the return delay L^{Past} of the results is quite small [17].

Assume that the task carried out in vehicle m has a data size of $Y_{m,r}(t)$. The computational resources for local execution are represented by $\mu_l(t)$, and the time of local execution is provided by

$$L_{m,r}^{lco} = \frac{Y_{m,r}(t)O(t)}{\mu_l(t)} \tag{7}$$

III. Problem Formulation and MDP Formation

The complete execution delay to compute offloading and wireless communication for every vehicle to carry out tasks at time slot t is provided by

$$L(t) = \sum_{m=1}^{W}\sum_{r=1}^{pra} \beta_{m,r(t)} L_{m,r}^{lco}(t) + (1-\beta_{m,r(t)}) L_{m,r}^{mce}(t) \tag{8}$$

where pra denotes the total number of task portions. Therefore, the overall delays in the wireless communication and compute offloading life cycle are as follows.

$$L = \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T} L(t) \tag{9}$$

$$= \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T}\sum_{m=1}^{W}\sum_{r=1}^{pra} \beta_{m,r(t)} L_{m,r}^{lco}(t) + (1-\beta_{m,r(t)}) L_{m,r}^{mce}(t) \tag{10}$$

Taking (10) into consideration the following optimization problem is formulated:

$$\begin{aligned} \mathcal{P.F.} \quad &: \quad \min L \qquad (11)\\ s.t. \quad \mathbb{X}_1 \quad &: \quad \sum_{r=1}^{pra} \beta_{m,r} \in \{0,1\},\\ \mathbb{X}_2 \quad &: \quad q_m \le Q_{max},\\ \mathbb{X}_3 \quad &: \quad W_{m,k} \le W_{max},\\ \mathbb{X}_4 \quad &: \quad m \in W, t \in T, \quad . \end{aligned}$$

Our objective is to reduce the latency L from (11). The discrete variable β in constraint $X1$ makes problem (11) a non-convex optimisation problem. The power of the channel q_m between the MS and the vehicle is kept under the maximum power Q_{max} by constraint $X2$. Constraint $X3$ ensures that, over the course of the vehicle's life, the bandwidth of the channel $W_{m,k}$ between the MS and the vehicle m does not exceed the maximum bandwidth W_{max}.

A. Markov Decision Process

- **State Space:** The state space at time slot t consists of the vehicle's trajectory location $\mathcal{G}(t)$, vehicle connection $\mathcal{B}(t)$, and vehicle velocity $\mathcal{C}(t)$. $\mathcal{G}(t)$= $[(i_1^t, j_1^t), \ldots, (i_m^t, j_m^t)]$. $\mathcal{B}(t) = [\mathcal{B}_1(t), \ldots, \mathcal{B}_m(t)]$(i.e $\mathcal{C}(t) = 0$ indicates that the vehicle is not connected to the STAR-RIS, $\mathcal{C}(t) = 1$ denotes that the vehicle connects to the STAR-RIS at time slot t). $\mathcal{C}(t) = [\mathcal{C}_1(t), \mathcal{C}_2(t), \ldots, \mathcal{C}_m(t)]$. So, the state space is $\mathcal{S}(t) = [\mathcal{G}(t), \mathcal{B}(t), \mathcal{C}(t)]$.
- **Action Space:** The selective action of the vehicle range is [0, 1] as indicated by the action $a_{m,t}$ at time slot t. The vector $\mathfrak{A} = (\beta_{1,t}, \beta_{2,t}, \ldots, \beta_{m,t})$ gives the action set $\mathfrak{A}$, and the action $\beta_{1,t}$ is taken by the agent. where $\beta_{1,t} = 0$ indicates the task is carried out locally by the vehicle, while $\beta_{m,t} = 1$ suggests that the task should be transmitted to the MS.
- **Reward:** At time slot t, when the agent is in the states $\mathcal{S}$, the reward for selecting the action β_t is given as $R(\mathcal{S}(t), \beta(t))$.

In order to minimize latency, we convert the optimisation problem into a joint optimization of the STAR-RIS components and offloading decision. We reframe the delay minimization problem as an agent reward R maximisation challenge. As a result, the reward function for the T time slots is provided as

$$R = \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T} L(t) \tag{12}$$

IV. Proposed Solution using Federated Learning

FL is a procedure that permits AI models to be produced jointly by scattered devices while maintaining the secrecy of data [18]. Updating models using centralised data isn't always appropriate when working with massive amounts of data. During the learning process, FL shares the parameters of the machine learning model (such as gradient) rather than gathering data. The global model is created by the central server by combining the model update parameters from all of the clients. The global model is provided to each client, which then uses its own data to improve it. The central server combines and weighs in an adaptive manner successively as they get new models from other devices, according to an asynchronous updating technique provided in [19]. The FL framework Flower [20], which we utilised to complete our work, offers an abstraction of the gRPC-based communication capabilities.

A. Architecture for Decentralized system

Since asynchronous decentralised algorithms need not rely on a centralised server, they can have any kind of connection topology and do not require centralised communication. Furthermore, the asynchronous progress of learning rounds eliminates latency resulting from imbalanced computer resources.

A server-client interaction exists in centralised FL, where clients acquire knowledge (improve) and the server compiles. Every device in decentralised FL is in charge of both aggregation and learning. In this instance, we investigate the arbitrary arrangement of aggregation and learning in asynchronous decentralised FL. In order to accomplish aggregation and learning in that sequence, we create AsyFL(Algorithm 1).

B. Algorithm proposed

For each vehicle, the rounds run asynchronously, and all algorithms are applicable to vehicle m. M is the total number of vehicles, D_m is the vehicle m's local data, vehicle m's total local data quantity is represented by n_m and $E_{m,r}$ represents the model parameters for device m at the conclusion of round $r-1$. Consequently, E_{b,r_b} stands for the most recent model parameters for vehicle m at that time. Finetune (E_m, D_m) indicates that E_m used D_m to learn on vehicle m. AsyFL is shown by Algorithm 1. For each vehicle, we start the model with the same set of parameters. Every round consists of two steps: Using the amount of data as weights, we first aggregate the models with the connected vehicles by calculating model weighted average.

We then carry out learning using local data. Expanding upon AsyFL, we further suggest algorithms uAsyFL (Algorithm 2), which include model aggregation through update history derived from BrainTorrent [21]. In this step, aggregation is limited to models changed since the previous round. It is anticipated that this strategy will stop the model's quality from declining as a result of reaggregating with previous models. U_m is a vector that shows the connected vehicles' version information for vehicle m at the end of the preceding round. An increment of one in the vehicle version number is indicated by increment $u^m(m)$.

C. Learning Process

It is expected that every vehicle has local data and a local model. The following steps are involved in the learning process as shown in fig 2:

- Ask the linked vehicles for model specifications and version details.
- Get model specifications and version data from the vehicles that are connected.
- Use the version information to aggregate the models.
- Utilising local data, update the local model.

V. Results and Discussion

All of the simulation's algorithms are run in a Python simulation environment. Table I displays the specific parameter settings. The evaluation PC is equipped with an NVIDIA RTX3090 GPU, 64GB GB of RAM and an Intel i7@ 3.60 GHz CPU. When the edge computing capability u_s grows in the range $[1\times10^{10}, 4\times10^{10}]$cycles/s, as shown

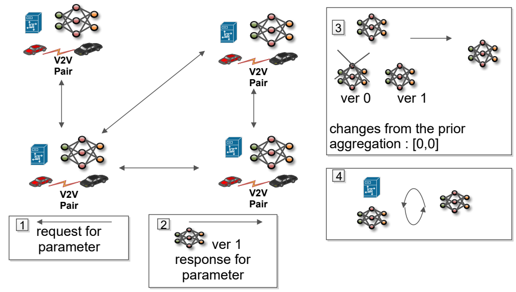

Fig. 2: Learning Process.

Algorithm 1 AsyFL

Set up local models first $\{E_{m,0}\}_{m=1}^{M}$
for every iteration, $r = 0, 1, \ldots$ **do**
  $E_m \leftarrow n_m.E_{m,r}$
  $N \leftarrow n_m$
  for $b \in$ {connected vehicle indexes} **do**
    $E_m \leftarrow E_m + n_b.E_{b,r_b}$
    $N \leftarrow N + n_b$
  end for
  $E_m \leftarrow \frac{E_m}{N}$
  $E_{m,r+1} \leftarrow$ Finetune(E_m, D_m)
end for

Algorithm 2 uAsyFL

Set up local models first $\{E_{m,0}\}_{m=1}^{M}$
Set the version vectors $\{u^m\}_{m=1}^{M}$ to their initial values.
for every iteration $r = 0, 1, \ldots$ **do**
  $u_{previous} \leftarrow u_m$
  $u_{updated} \leftarrow$ obtain variants for each and every other vehicle
  $E_m \leftarrow n_m.E_{m,r}$
  $N \leftarrow n_m$
  for $b \in$ {connected vehicle indexes} **do**
    if $u^b_{updated} > u^b_{previous}$ **then**
      $E_m \leftarrow E_m + n_b.E_{b,r_b}$
      $N \leftarrow N + n_b$
    end if
  end for
  $E_m \leftarrow \frac{E_m}{N}$
  $E_{m,r+1} \leftarrow$ Finetune(E_m, D_m)
  $u^m \leftarrow u_{updated}$
  Increment $u^m(m)$
end for

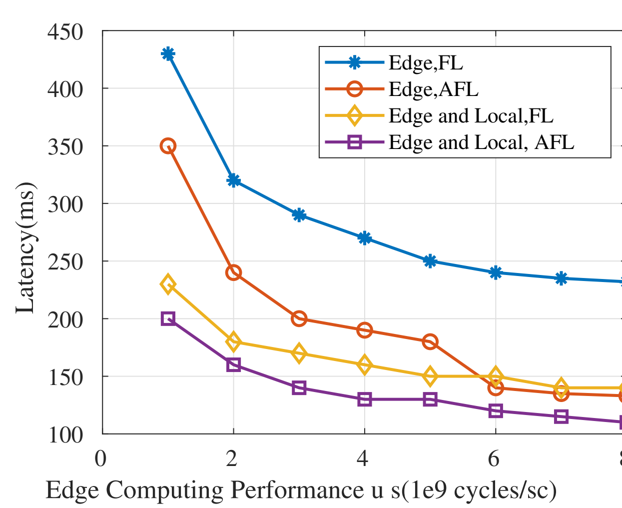

Fig. 3: Latency versus edge computing capability u_s

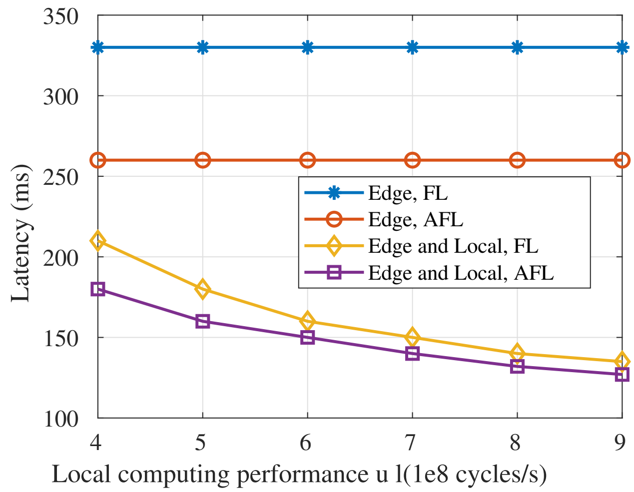

Fig. 4: Latency versus local computing capability u_l

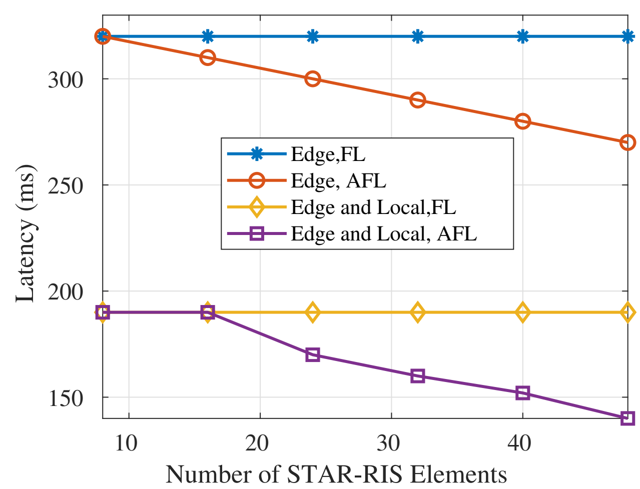

Fig. 6: Latency versus Number of STAR-RIS elements

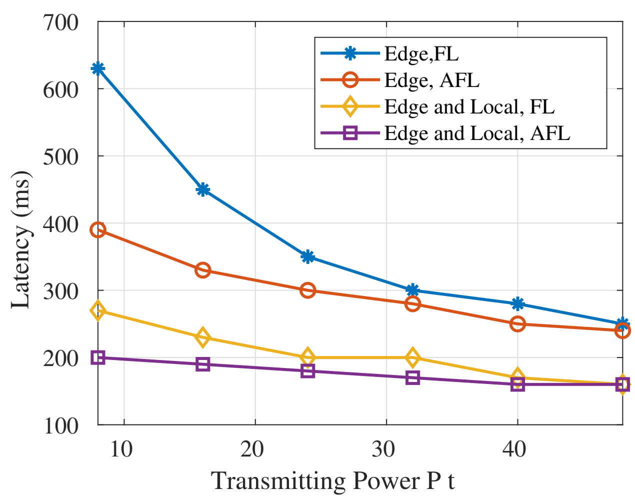

Fig. 5: Latency versus Transmitting power P_t

in Fig. 3, the latency is greatly reduced; however, a slight reduction is also visible when u_s increases in the range $[4 \times 10^{10}, 8 \times 10^{10}]$. This is because, for small u_s, the latency imposed by edge computing predominates, whereas, for big u_s, the latency imposed by communication does.

According to Fig. 4, ELFL (Edge Local Federated Learning) and ELAFL (Edge Local Asynchronous Federated Learning) are observed to drop proportionately to $\frac{1}{u_l}$. Nevertheless, even with the increase in u_l, EFL (Edge Federated Learning) and EAFL (Edge Asynchronous Federated Learning)stay the same. This is due to the fact that the latency of shifting all computing tasks to the edge is unaffected by changes in the local computing capacity.

The transmission's power P_t is adjusted within the range of [0, 50] dBm in Figure 5. Note that when transmission power increases, the delay associated with each method decreases. Moreover, the delay decreases significantly when Pt rises in the range [0, 30] dBm, but only marginally when Pt rises in the range [30, 50] dBm. Additionally, when Pt increases, the latency differences between the schemes utilising FL STAR-RIS and AFL STAR-RIS get less. Fig. 5 illustrates how effectively the suggested ELAFL functions in low transmission power scenarios.

The total number of STAR-RIS elements varies in Fig. 6. Note that the schemes with AFL have decreasing latency as the number of STAR-RIS elements rises, but the schemes with FL do not vary in latency. When the number of STAR-RIS elements increases in the range [20, 50], the latency is significantly decreased, but only a small as the number of STAR-RIS elements increases in the range [1, 20], a reduction in latency may be observed. Additionally, when E increases, the latency differences between the schemes using FL STAR-RIS and AFL STAR-RIS gets more.

TABLE I: Parameters for Simulation

Parameters	**Values**
Cellular cell's Radius	500m
Size of task data	400 KB
Carrier Frequency	5MHz
The MEC server's frequency	8GHz
BS transmission power	5W
computation density	1000 cycle/bit
The channel's bandwidth	180 KHz
Channel Power Gain	-30dB
maximum power used for transmission	1KW
Path loss exponent	4
frequency of local computing	1 GHz
Learning Rate	0.001
Small-batch Size	32
Factor of Discount	0.9
Starting Exploration	1
Ending Exploration	0.01
Total steps of exploration	1000
Capacity for replay storage	1000
Step Count for Every Epoch	20
Episodes	100

VI. Conclusion

In this research, we have created a unique framework for VRCS latency reduction by merging AFL with STAR-RIS. Our approach makes use of STAR-RIS's dynamic reconfigurability to optimise the propagation environment and boost signal strength and coverage. AFL lowers latency and delays related to synchronous communication and data aggregation by enabling vehicles to update the global model asynchronously. Our system tackles the unique challenges posed by vehicle networks, including the requirement for high mobility, variable network circumstances, and real-time responsiveness. Our thorough simulations validate the effectiveness of the strategy, showing that it outperforms current decentralised solutions.

References

[1] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming," *IEEE transactions on wireless communications*, vol. 18, no. 11, pp. 5394–5409, 2019.

[2] Q. Wu, S. Zhang, B. Zheng, C. You, and R. Zhang, "Intelligent reflecting surface-aided wireless communications: A tutorial," *IEEE transactions on communications*, vol. 69, no. 5, pp. 3313–3351, 2021.

[3] S. Chaudhary, A. Nehra, I. Budhiraja, R. Chaudhary, and A. Bansal, "Star-ris based resource scheduling and mode selection for drone assisted 5g communications," in *IEEE INFOCOM 2024-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 01–06, IEEE, 2024.

[4] X. Gu, G. Zhang, Y. Ji, W. Duan, M. Wen, Z. Ding, and P.-H. Ho, "Intelligent surface aided d2d-v2x system for low-latency and high-reliability communications," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 11, pp. 11624–11636, 2022.

[5] Z. Du, C. Wu, T. Yoshinaga, K.-L. A. Yau, Y. Ji, and J. Li, "Federated learning for vehicular internet of things: Recent advances and open issues," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 45–61, 2020.

[6] S. Chaudhary, R. Chaudhary, I. Budhiraja, A. Bhardwaj, A. Nehra, and S. Martha, "Improving the transmission power of uavs with intelligent reflecting surfaces in v2x," in *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, pp. 1–6, IEEE, 2023.

[7] V. Vishnoi, P. Consul, I. Budhiraja, S. Gupta, and N. Kumar, "Deep reinforcement learning based energy consumption minimization for intelligent reflecting surfaces assisted d2d users underlaying uav network," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, IEEE, 2023.

[8] B. Song and F. Wang, "Star-ris-aided secure communications for mec with delay/energy-constrained qos using multi-agent deep reinforcement learning," *Ad Hoc Networks*, vol. 154, p. 103370, 2024.

[9] W. Ni, Y. Liu, Y. C. Eldar, Z. Yang, and H. Tian, "Star-ris integrated nonorthogonal multiple access and over-the-air federated learning: Framework, analysis, and optimization," *IEEE internet of things journal*, vol. 9, no. 18, pp. 17136–17156, 2022.

[10] T. H. T. Le, L. Cantos, S. R. Pandey, H. Shin, and Y. H. Kim, "Federated learning with noma assisted by multiple intelligent reflecting surfaces: Latency minimizing optimization and auction," *IEEE Transactions on Vehicular Technology*, 2023.

[11] N. Li, B. Chen, and X. Tao, "Star-ris assisted offloading based on hybrid noma: A time minimization perspective," *IEEE Transactions on Vehicular Technology*, 2024.

[12] A. A. Al-Habob, O. Waqar, and H. Tabassum, "Latency minimization in phase-coupled star-ris assisted multi-mec server systems," in *2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–7, IEEE, 2023.

[13] N. Joshi, I. Budhiraja, D. Garg, S. Garg, B. J. Choi, and M. Alrashoud, "Deep reinforcement learning based rate enhancement scheme for ris assisted mobile users underlaying uav," *Alexandria Engineering Journal*, vol. 91, pp. 1–11, 2024.

[14] S. Chaudhary, I. Budhiraja, R. Chaudhary, N. Kumar, D. Garg, and A. M. Almuhaideb, "Quantum federated reinforcement learning based joint mode selection and resource allocation for star-ris aided vrcs," *IEEE Internet of Things Journal*, 2024.

[15] S. Chaudhary, I. Budhiraja, R. Chaudhary, S. Garg, B. J. Choi, and M. Alrashoud, "Proximal policy optimization based sum rate maximization scheme for star-ris-assisted vehicular networks underlaying uav," *Alexandria Engineering Journal*, vol. 118, pp. 700–710, 2025.

[16] T. Liu, J. Tong, Q. Guo, J. Xi, Y. Yu, and Z. Xiao, "On the performance of massive mimo systems with low-resolution adcs and mrc receivers over rician fading channels," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4514–4524, 2020.

[17] X. Xu, B. Shen, S. Ding, G. Srivastava, M. Bilal, M. R. Khosravi, V. G. Menon, M. A. Jan, and M. Wang, "Service offloading with deep q-network for digital twinning-empowered internet of vehicles in edge computing," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1414–1423, 2020.

[18] E. T. M. Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Communications Surveys & Tutorials*, 2023.

[19] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous federated optimization," *arXiv preprint arXiv:1903.03934*, 2019.

[20] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusmão, *et al.*, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.

[21] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, "Braintorrent: A peer-to-peer environment for decentralized federated learning," *arXiv preprint arXiv:1905.06731*, 2019.