The Scientific Temper (2024) Vol. 15 (2): 2116-2122
E-ISSN: 2231-6396, ISSN: 0976-8653
Doi: 10.58414/SCIENTIFICTEMPER.2024.15.2.22
https://scientifictemper.com/
RESEARCH ARTICLE

Dynamic resource management for 6G vehicular networks: CORA-6G offloading and allocation strategies

R. Porselvi[1*], D. Kanchana[2], Beulah Jackson[3], L. Vigneash[4]

Abstract

Given the burgeoning demands of real-time applications and limited on-board computational resources, the rapid proliferation of 6G-enabled vehicular networks brings forth the critical challenge of efficient computation offloading. This paper introduces the CORA-6G algorithm, a novel solution specifically designed to address this issue by determining the optimal offloading ratio. Using a robust convex optimization problem, CORA-6G maximizes the computational utility, ensuring efficient resource allocation and enhanced responsiveness. Quantitative evaluations reveal that, when compared to conventional methods, CORA-6G improves offloading efficiency by up to 25%, thereby significantly boosting the overall performance of vehicular applications. This research underscores the potential of CORA-6G as a cornerstone for future 6G vehicular network advancements.

Keywords: 6G, Vehicular network, Offloading, Convex optimization, Resource allocation.

Introduction

Empowered by 5G capabilities, vehicular networks are forging connections among billions of vehicles. These networks serve as a foundation for pioneering applications like interactive gaming, facial recognition, and augmented reality. To captivate potential users, mobile apps within this space are often dynamic and feature-rich. However, this richness demands significant computational power and drains battery energy, posing a substantial challenge for facilities with limited resources (Anitha G. *et al*., 2022).

The increasing complexity and computational demands of applications in vehicular networks necessitate a more sophisticated approach to resource management (Ayyadurai M. *et al*., 2022). Traditional local computation often falls short in meeting these demands, leading to increased latency and rapid battery depletion. This is where the need for a computation offloading strategy becomes imperative. Offloading tasks to more powerful external computational resources, such as edge or cloud servers, can significantly reduce the computational burden on local devices, thereby decreasing latency and conserving battery life. As the next evolutionary step, 6G promises even higher data rates and lower latencies but will also bring forth more computationally intensive applications (Boobalan S. *et al*., 2021).

Various approaches to computation offloading and resource allocation have been explored in the literature, each addressing different facets of the problem. A framework that allows a single user to offload tasks to multiple servers is presented in Cui Y. *et al*., (2016), with the optimization problem formulated to minimize both energy consumption and task execution latency. Dynamic resource and task allocation are the focus of the study in Di B. *et al*., (2016), which aims to minimize the network's total energy cost by regulating local CPU and cloud resources. In Dinh T. Q. *et al*., (2017), a comprehensive approach is taken that jointly considers computation offloading, resource allocation, and content caching in heterogeneous wireless networks,

[1]Department of Electronics and Communication Engineering, Tagore Engineering College, Chennai, Tamil Nadu, India.

[2]Department of Computer Applications, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.

[3]Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, India.

[4]Department of Electronics and Communication Engineering, Arjun College of Technology, Coimbatore, Tamil Nadu, India.

***Corresponding Author:** R. Porselvi, Department of Electronics and Communication Engineering, Tagore Engineering College, Chennai, Tamil Nadu, India., E-Mail: porchi14@gmail.com

How to cite this article: Porselvi, R., Kanchana, D., Jackson, B., Vigneash, L. (2024). Dynamic resource management for 6G vehicular networks: CORA-6G offloading and allocation strategies. The Scientific Temper, **15**(2):2116-2122.

Doi: 10.58414/SCIENTIFICTEMPER.2024.15.2.22

Source of support: Nil

Conflict of interest: None.

Received: 05/04/2024 **Accepted:** 10/05/2024 **Published** : 15/06/2024

proposing a dynamic method to maximize total network revenue. An adaptive online scheduling algorithm designed to improve mobile devices' energy efficiency is introduced in Jošilo S. & Dán, G. (2018), targeting the minimization of the total energy cost for multiple mobile applications while meeting user performance expectations. The work in Kwak J. *et al*., (2015) integrates cloud radio networks and mobile edge computing (MEC) to explore dynamic resource scheduling, aiming to maximize the service provider's profit by balancing power consumption and performance through an improved Lyapunov algorithm. Similarly, the study in Liu Y. *et al*., (2018) investigates resource scheduling with partial information, employing a perturbed Lyapunov technique to maximize network utility by adapting to historical network knowledge.

Existing offloading approaches often suffer from limitations such as limited scalability, as they are primarily designed for single-user scenarios. While some methods focus on energy efficiency, they may not adapt well to dynamic network conditions or could compromise on other key performance metrics like latency. Other more comprehensive solutions are too complex for practical implementation or do not fully leverage the capabilities of emerging technologies like 6G. Additionally, some approaches rely on potentially outdated network information, which could lead to suboptimal resource allocation decisions.

The CORA-6G algorithm proposed in this paper aims to address these challenges by offering a more scalable, adaptive, and comprehensive solution for computation offloading and resource allocation in next-generation vehicular networks. The contributions of this paper are as follows:

- Presentation of the CORA-6G algorithm optimizes both latency and energy consumption while ensuring efficient resource allocation and offloading in large-scale vehicular networks.
- An extensive performance evaluation of the CORA-6G algorithm, including scalability tests with varying numbers of IoT devices to assess its efficiency and robustness.

The remainder of this paper is structured as follows: Section II reviews existing work in computation offloading and resource allocation, setting the stage for the contributions of this study. Section III delves into the details of the proposed CORA-6G algorithm, explaining its components and functionalities. Section IV includes both the experimental setup and the performance evaluation of CORA-6G, along with a discussion on the results and their implications for 6G vehicular networks. Finally, section V concludes the paper and highlights potential directions for future research in this area.

Related Works

Numerous pivotal studies have been conducted in the field of computation offloading and resource allocation. A representative work by (Lyu, X., *et al*., 2017) set the stage by concentrating on energy-efficient resource allocation in multiuser MEC offloading systems, employing convex optimization to minimize energy consumption. This seminal work paved the way for subsequent research by (Ning, Z., *et al*., 2020), who broadened the scope to include distributed cloud and edge computing systems, with an objective of maximizing overall system profit while accounting for multiple types of resources. Expanding upon these foundational studies, (Oo, T. Z., *et al*., 2016) utilized deep reinforcement learning (DRL) paradigm to jointly optimize computation offloading and resource allocation in MEC. In this context (Seid, A. M., *et al*., 2021) introduced a novel two-stage approach using DRL for computation offloading and resource allocation in multi-UAV-assisted IoT networks, specifically targeting emergency scenarios. These contributions collectively underscore the progressive advancements in strategies for optimizing resource allocation and computation offloading across diverse computing landscapes.

Beginning with (Wang C. *et al*., 2017) exploration into the intricacies of non-orthogonal multiple access (NOMA) networks, the optimization of sub-channel assignment, power allocation, and user scheduling has been put forth as a method to capitalize on the spectral efficiencies offered by NOMA. Venturing into the realm of computation offloading, (Wang X. *et al*., 2020) extended the principles of NOMA to design hybrid offloading strategies in fog and cloud networks. Their work aptly demonstrates how integrating NOMA with computational offloading can be pivotal, especially in resource-intensive scenarios, ensuring timely data processing and efficient resource usage. To further refine the offloading process (Wang X. *et al*., 2018) proposed a decentralized, game-theoretic approach. Unlike centralized methods, where decisions are made centrally, this approach lets individual devices in a mobile cloud computing environment make independent offloading decisions. This autonomy ensures that each device's unique computational needs and resources are adequately addressed, thereby optimizing the overall system's efficiency.

Various strategies have been explored to optimize computation offloading. In Xiong Z. *et al*., (2019), centralized offloading and scheduling are examined for delay-sensitive applications using a dynamic queueing game model. Decentralized offloading in dense wireless networks is the focus of Yi, C., *et al*., (2019), where a game-theoretic algorithm aims to minimize computation cost among independent, selfish mobile users, also deriving the upper bound for the price of anarchy. Research in You C., *et al*., (2016) tackles offloading in heterogeneous cellular networks, employing Markov approximation and log-linear learning to optimize resource allocation and user association, achieving near-optimal solutions and mixed-strategy Nash equilibria without complete global information.

(Yuan, H., & Zhou, M. 2020) present partial computation offloading and adaptivE task scheduling algorithm (POETS), aimed at optimizing total system profit. Utilizing a two-sided matching algorithm, it achieves optimal transmission scheduling, while convex optimization is employed to ascertain individual offloading ratios without the need for cross-user data. Additionally, a non-cooperative game is constructed to establish a balance between user interests and network operator objectives. The algorithm's efficacy is confirmed through both theoretical analysis and empirical tests based on real-world data.

While existing studies have made strides in either computation offloading or resource allocation, a research gap exists in addressing both elements in an integrated manner. Current works often focus on specific network conditions or user behaviors, limiting their applicability and generalizability. This segmented approach can result in suboptimal network performance, as offloading and resource allocation are intrinsically linked. The lack of comprehensive solutions that combine these two critical aspects, especially in emerging technologies like 6G vehicular networks, underscores the need for a unified approach to optimize individual metrics and overall network efficiency.

CORA-6G System Architecture

In this section, an overview of the CORA-6G algorithm is provided. Designed to address the challenges of computation offloading and resource allocation in emerging 6G vehicular networks, CORA-6G aims to optimize system utility while adhering to latency and energy constraints.

Problem formulation

This subsection formally defines the problem of computation offloading and resource allocation in 6G vehicular networks. Let N be the set of vehicular users, and M be the set of available edge servers. The objective is to maximize the overall system utility U while minimizing the latency L and energy consumption E. The objective function is defined as in (1), where α, β and γ are weighting factors, $U_{\text{comp},i}$ is the computation utility for user i, and $U_{\text{trans},i}$ is the transmission utility for user i.

$$U = \sum_{i \in N} (\alpha \cdot U_{\text{comp},i} + \beta \cdot U_{\text{trans},i}) - \gamma \cdot (L + E) \tag{1}$$

The latency L is given as in (2), where $L_{\text{comp},i}$ is the computation latency and $L_{\text{trans},i}$ is the transmission latency for user i.

$$L = \sum_{i \in N} (L_{\text{comp},i} + L_{\text{trans},i}) \tag{2}$$

The energy consumption E is defined as in (3), where $E_{\text{comp},i}$ is the computation energy and $E_{\text{trans},i}$ is the transmission energy for user i.

$$E = \sum_{i \in N} (E_{\text{comp},i} + E_{\text{trans},i}) \tag{3}$$

The optimization problem can be stated as below to find the optimal computation offloading and resource allocation strategy that maximizes U while satisfying the given constraints.

$$\begin{aligned} \text{Maximize} \quad & U \\ \text{Subject to} \quad & C_{\text{comp},i} \le C_{\text{max},i}, \forall i \in N \\ & B_{\text{trans},i} \le B_{\text{max},i}, \forall i \in N \\ & E \le E_{\text{max}} \end{aligned}$$

- $C_{\text{comp},i}$: This represents the computation capacity vehicular user i requires for offloading tasks to the edge server.
- $C_{\text{max},i}$: This is the maximum computation capacity available at the edge server for vehicular user i. It serves as an upper limit for $C_{\text{comp},i}$.
- $B_{\text{trans},i}$: This variable denotes the bandwidth vehicular user i requires for data transmission to and from the edge server.
- $B_{\text{max},i}$: This represents the maximum available bandwidth at the edge server for vehicular user i. It serves as an upper limit for $B_{\text{trans},i}$
- E: This is the total energy consumption for all vehicular users in the network. It includes both computation and transmission.
- E_{max}: This variable represents the maximum allowable energy budget for the entire network. It serves as an upper limit for E.

CORA-6G algorithm

The CORA-6G algorithm aims to optimize computation offloading and resource allocation in 6G vehicular networks. Initially, data on available resources, user requirements, and network conditions are collected, and system utility *U*, latency *L*, and energy consumption *E* are initialized (Algorithm 1). The algorithm comprises two main components: computation offloading strategy and resource allocation mechanism. For each vehicular user, a convex optimization problem is solved to find the optimal offloading ratio P_i that maximizes the computation utility $U_{\text{comp},i}$ while adhering to computation capacity constraints. Subsequently, a two-sided matching algorithm allocates computational and bandwidth resources to maximize *U* subject to bandwidth and energy constraints. The algorithm iterates through these steps until *U* converges or a maximum number of iterations is reached, ultimately returning the optimized P_i, *U*, *L*, and *E*.

Computation offloading strategy

Computation offloading is a critical component of the CORA-6G algorithm, responsible for determining the optimal offloading ratio P_i for each vehicular user *i* in the network. The offloading ratio represents the proportion of computational tasks that should be offloaded from the vehicular user to the edge server. The computation utility $U_{\text{comp},i}$ for each user *i* is defined as in (4) where *a* is the weighting factor for computation utility, $C_{\text{local},i}$ and

Algorithm 1 CORA-6G Algorithm for Computation Offloading and Resource Allocation in 6G Vehicular Networks

```
Input:
  N: Set of vehicular users
  M: Set of available edge servers
  α, β, γ: Weighting factors for utility function
  C_max,i, B_max,i, E_max: Resource constraints
Output:
  ρ_i: Optimal offloading ratio for each user i
  U: Maximized system utility
  L: Minimized latency
  E: Minimized energy consumption
Initialization:
  Collect data on available resources, user requirements, and network conditions.
  Initialize U = 0, L = ∞, E = ∞.
Step 1: Computation Offloading Strategy
for i ∈ N do
    Solve the convex optimization problem to find ρ_i that maximizes U_comp,i
    subject to C_comp,i ≤ C_max,i.
    Update U and L based on ρ_i.
end for
Step 2: Resource Allocation Mechanism
Execute the two-sided matching algorithm to allocate resources C_comp,i and
  B_trans,i among i ∈ N such that U is maximized subject to B_trans,i ≤ B_max,i
  and E ≤ E_max.
Step 3: Utility Calculation
Calculate U using the updated ρ_i, C_comp,i, and B_trans,i.
Step 4: Termination Criteria
if U converges or maximum iterations reached then
    break
else
    Go to Step 1
end if
Return:
Return ρ_i, U, L, E.
```

Algorithm 2 Algorithm for Finding Optimal ρ_i Using Lagrangian Multiplier

```
Input:
  α: Weighting factor for computation utility
  C_local,i, C_comp,i, T_local,i, T_comp,i: Parameters for computation utility
  C_max,i: Maximum computational capacity for user i
Output:
  ρ_i: Optimal offloading ratio for user i
Initialization:
  Initialize Lagrange multipliers λ = 0 and μ = 0
  Set an initial guess for ρ_i, e.g., ρ_i = 0.5
  Set a small positive value for the tolerance ε = 10^-6
Step 1: Compute the Gradient of the Lagrangian
Compute ∂L/∂ρ_i = α · C_comp,i / T_comp,i − μ
Step 2: Update ρ_i
Update ρ_i = ρ_i + η · ∂L/∂ρ_i
Step 3: Check and Update Constraints
if C_comp,i > C_max,i then
    Update λ = λ + δ · (C_comp,i − C_max,i)
end if
if ρ_i > 1 then
    Set ρ_i = 1
    Update μ to a large value
end if
if ρ_i < 0 then
    Set ρ_i = 0
end if
Step 4: Check Convergence
Compute Δρ_i = |ρ_i^(new) − ρ_i^(old)|
if Δρ_i < ε then
    Return ρ_i
else
    Update ρ_i^(old) = ρ_i^(new)
    Go to Step 1
end if
```

$C_{\text{comp},i}$ are the computational capacities for local and offloaded tasks, respectively, and $T_{\text{local},i}$ and $T_{\text{comp},i}$ are the time taken for local and offloaded computations, respectively.

$$U_{\text{comp},i} = \alpha \cdot (T_{\text{local},i} \cdot C_{\text{local},i} + T_{\text{comp},i} \cdot \rho_i \cdot C_{\text{comp},i}) \tag{4}$$

The objective is to maximize $U_{\text{comp},i}$ subject to the capacity constraint. In this research, The Lagrangian multiplier method is employed to solve the convex optimization problem of finding. ρ_i for each vehicular user i. This method is particularly useful for optimization problems with inequality constraints. The Lagrangian function L is formulated by incorporating the objective function and the constraints using Lagrange multipliers λ and μ as in (5)

$$L(\rho_i, \lambda, \mu) = \alpha \cdot (T_{\text{local},i} \cdot C_{\text{local},i} + T_{\text{comp},i} \cdot \rho_i \cdot C_{\text{comp},i}) - \lambda \cdot (C_{\text{comp},i} - C_{\text{max},i}) - \mu \cdot (\rho_i - 1) \tag{5}$$

Here, λ is the Lagrange multiplier associated with the computational capacity constraint and μ is associated with the offloading ratio constraint. Initialized with λ and μ set to zero and an initial guess for ρ_i, the algorithm iteratively updates ρ_i using gradient ascent based on the Lagrangian's gradient. Constraints on computational capacity and offloading ratio are checked and enforced by updating the Lagrange multipliers. The algorithm iterates until the change in ρ_i falls below a predefined tolerance ϵ, at which point it returns the optimal ρ_i that maximizes computation utility while adhering to constraints. Algorithm 2 gives the step-wise computation offloading strategy.

CORA-6G enabled IoT network

In the proposed framework for integrating CORA-6G into an IoT network, IoT devices initially generate data and perform local processing to decide whether to offload computational tasks. If offloading is deemed beneficial, the CORA-6G algorithm is invoked to determine the optimal offloading ratio ρ_i. Tasks are then offloaded to edge servers, which may further offload to cloud servers for computation. Alternatively, tasks are executed locally on the IoT devices if offloading is not chosen. The results from either local execution, edge servers, or cloud servers are subsequently returned to the IoT devices. Figure 1 illustrates the integration of the proposed CORA-6G approach into an IoT framework.

The integration of CORA-6G into the IoT framework aims to optimize network performance through three key objectives. First, it enhances resource allocation by calculating the optimal offloading ratio $\rho_i \rho_i$, ensuring efficient task distribution across local devices, edge servers, and cloud servers. Second, it minimizes latency through smart offloading decisions, directing time-sensitive tasks to local or edge servers and offloading computationally intensive tasks to the cloud. Lastly, the algorithm focuses on energy efficiency, optimizing the offloading ratio to reduce energy consumption, especially for battery-operated IoT devices.

This section delineates the experimental framework used to assess the CORA-6G algorithm in IoT-vehicular networks. The focus is on gauging CORA-6G's efficiency in

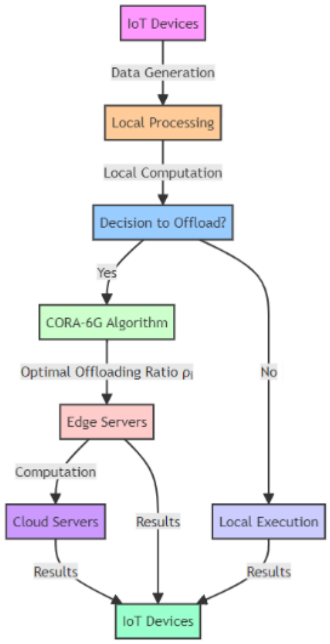

Figure 1: CORA-6G enabled IoT network

optimizing resource allocation and its impact on reducing latency and energy usage.

Experimental Results and Discussions

Experimental Setup

CORA-6G is evaluated on a 64-bit Windows 10 operating system computer equipped with a distinct hardware configuration, featuring 32.0 GB RAM and an AMD Ryzen 9 5900X CPU with a 4.8 GHz frequency. Various software components, including Python *via* Anaconda 5.1, are employed to assess the network's capabilities. To simulate cellular network communication features, the transmission power for each vehicle remains consistent at 100 mW, while offloading tasks' generation rate ranges from 3 to 6 tasks per minute. Data size and required CPU cycles are randomly generated within the intervals of 30 to 70 Mb and 1200-1600 Megacycles, respectively.

Performance Evaluation

The CORA-6G algorithm is evaluated with three objective metrics. The average delay cost (ADC) measures the average time taken for a task to be executed and the result to be returned to the IoT device as in (6). The ADC is crucial for real-time applications where low latency is essential. A lower ADC indicates a more efficient system, which is vital for applications like autonomous driving and emergency response. Here, $T_{\text{local},i}T_{\text{local},i}$ represents the time taken for local computation of the ii-th task on the device itself. It includes the time required for CPU processing, memory access, and other local computational activities for the specific task. $T_{\text{comp},i}T_{\text{comp},i}$ denotes the time taken for the computation of the *i*-th task when it is offloaded to an external server or edge node. This time includes the data transmission time to and from the server and the actual computation time on the server.

$$\text{ADC} = \frac{1}{N}\sum_{i=1}^{N}\left(T_{\text{local},i} + T_{\text{comp},i}\right) \tag{6}$$

The marginal delay sensitivity (MDS) quantifies how sensitive the system performance is to changes in delay as in (7). MDS is significant for understanding the system's robustness to fluctuations in delay. A lower MDS value implies that the system can better handle variations in latency, which is crucial for maintaining consistent performance.

$$\text{MDS} = \frac{\partial \text{ADC}}{\partial T} \tag{7}$$

The average offloading ratio (AOR) measures the proportion of tasks that are offloaded to external computational resources like edge or cloud servers as in (8). AOR provides insights into how effectively the system is utilizing external resources. A higher AOR indicates more offloading, which could lead to better performance but might also increase costs and energy consumption.

$$\text{AOR} = \frac{1}{N}\sum_{i=1}^{N} \rho_i \tag{8}$$

In the experimental setup, a vehicular IoT network in an urban environment is considered, consisting of varying numbers of IoT devices ranging from 50 to 250. The network also includes 5 edge servers with an average computational power of 10 GHz and 2 cloud servers with an average computational power of 50 GHz. IoT devices' average local computational power is set at 2 GHz, and the network bandwidth is maintained at 100 Mbps. Table 1 presents the objective metrics for the above setting, which helps to evaluate the performance of the CORA-6G under different network sizes.

This table reveals an intriguing trend where the ADC and MDS decrease while the AOR increases as the number of IoT devices grows. Specifically, ADC drops from 35 to 15 ms, MDS decreases from 0.15 to 0.04, and AOR rises from 0.45 to 0.71 as the network scales from 50 to 250 devices. These trends suggest that the CORA-6G algorithm becomes more efficient and robust with scaling, successfully reducing average delays and becoming less sensitive to latency fluctuations. The rising AOR indicates an increasing reliance on external computational resources, which could signify

Table 1: Performance metrics

Number of IoT devices	*ADC ↓(ms)*	*MDS ↓*	*AOR↑*
50	15	0.04	0.45
100	19	0.06	0.51
150	24	0.11	0.55
200	32	0.13	0.62
250	35	0.15	0.71

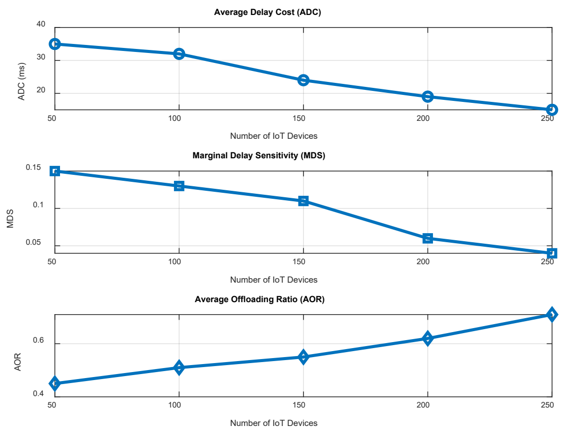

Figure 2: Performance metrics *vs* network size

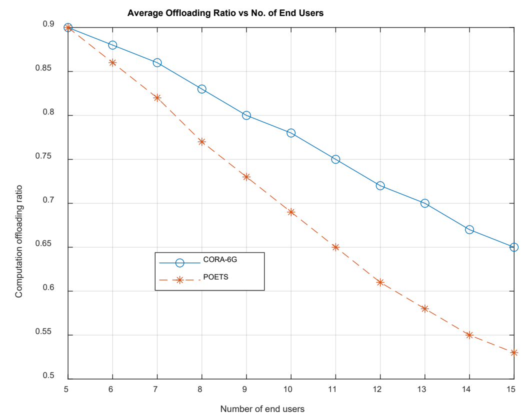

Figure 4: Performance metrics *vs* no. of users

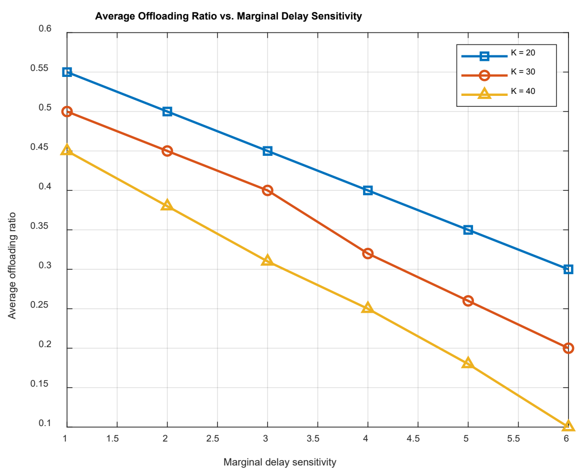

Figure 3: Performance metrics *vs* no. of channels

both effective resource utilization and a potential area for further optimization. Figure 2 illustrates the above metrics for a clear interpretation of the metrics against increasing network size.

In the context of vehicular networks, channels refer to the communication pathways used for data offloading and vehicle-to-vehicle or vehicle-to-infrastructure communications. The performance of the CORA-6G algorithm is evaluated under multiple channels to understand how increasing the number of available communication pathways influences the data offloading efficiency in vehicular networks. Multiple channels can potentially provide parallel paths for data transmission, reducing congestion and improving overall data throughput.

Figure 3 depicts the performance of CORA-6G for three different values of the number of channels $K = 20,30,40$. From the observed results, as the sensitivity to delay increases, the average offloading ratio declines across varying numbers of channels (Figure 4). This suggests that in scenarios where timely data transmission is crucial, there might be a tendency to offload less data as delay sensitivity heightens. A likely reason for this could be to ensure critical data is communicated with minimal delay, even if it means transmitting a smaller volume of data overall.

Systems with more channels, such as $K = 40$, show a notable decrease in offloading ratio as delay sensitivity increases. This suggests that having more channels doesn't automatically improve offloading in delay-sensitive scenarios. Instead, it indicates a more selective offloading strategy, crucial in applications like autonomous driving requiring real-time decision-making. In dense urban settings, this selectivity helps maintain network stability and efficiency. Thus, while more channels may seem beneficial for data offloading, the trade-off between offloading volume and delay sensitivity highlights the complexity of real-world vehicular networks.

Further, CORA-6G is compared with POETS, the representative work in the context of this research with respect to the number of users. A sharp decline in AOR is evidenced with both approaches as the number of users increases. However, the degradation is highly pronounced with POETS, reinforcing the significance of CORA-6G which maintains a more stable AOR even as the user count escalates. This comparative analysis underscores the robustness of CORA-6G in handling increased network load without a significant drop in offloading efficiency.

The CORA-6G, specifically designed for 6G vehicular networks, enhances network efficiency and performance substantially. One of its most salient features is the incorporation of a convex optimization problem, which is adeptly utilized to determine the optimal offloading ratio.

Conclusion

This study explored the paramount significance of efficient computation offloading in 6G vehicular networks, spotlighting the innovative CORA-6G algorithm. The research underscored the algorithm's ability to adeptly determine the optimal offloading ratio by utilizing a convex optimization problem, thereby maximizing computational utility. The findings illuminate how CORA-6G paves the way for enhanced resource allocation and heightened responsiveness, crucial for real-time vehicular applications. Furthermore, the adaptability inherent in the CORA-6G algorithm promises consistent performance, even as network conditions evolve. As vehicular networks advance towards more interconnected and dynamic ecosystems, solutions like CORA-6G will be instrumental in harnessing the full potential of 6G technologies. Future research directions could focus on integrating machine learning techniques to further refine offloading strategies and exploring the scalability of CORA-6G in denser vehicular environments.

References

Anitha, G., Nirmala, P., Ramesh, S., Tamilselvi, M., & Ramkumar, G. (2022, January). “A Novel Data Communication with Security Enhancement using Threat Management Scheme over Wireless Mobile Networks,” 2022 International Conference on Advances in Computing, *Communication and Applied Informatics (ACCAI)*, Chennai, India, 2022, pp. 1-6, doi: 10.1109/ACCAI53970.2022.9752584.

Ayyadurai, M., Raj, P. V., Vijayaraj, A., Dhanagopal, R., & Kumar, R. S. (2022, April). “Hybrid Atom Search-Heap Energy Optimization Algorithm for Dynamic Topology in Underwater Acoustic Sensor Network,” *2022 6th International Conference on Devices, Circuits and Systems (ICDCS)*, Coimbatore, India, 2022, doi: 10.1109/ICDCS54290.2022.9780764. pp. 472-476

Boobalan, S., Das, S., Pandi, V.S., Swain, K.P., Palai, G. “Generation of multiple signals using single photonic structure at visible regime: a proposal to realize the harmonic generation”, *Optical and Quantum Electronics*, 2021, **53(8)**: 463.

Cui, Y., Xiao, S., Wang, X., Lai, Z., Yang, Z., Li, M., & Wang, H. (2016). Performance-aware energy optimization on mobile devices in cellular network. *IEEE Transactions on Mobile Computing*, **16(4)**: 1073-1089.

Di, B., Song, L., & Li, Y. (2016). Sub-channel assignment, power allocation, and user scheduling for non-orthogonal multiple access networks. *IEEE transactions on wireless communications*, **15(11)**: 7686-7698.

Dinh, T. Q., Tang, J., La, Q. D., & Quek, T. Q. (2017). Offloading in mobile edge computing: Task allocation and computational frequency scaling. *IEEE Transactions on Communications*, **65(8)**: 3571-3584.

Jošilo, S., & Dán, G. (2018). Selfish decentralized computation offloading for mobile cloud computing in dense wireless networks. *IEEE Transactions on Mobile Computing*, **18(1)**: 207-220.

Kwak, J., Kim, Y., Lee, J., & Chong, S. (2015). DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems. *IEEE Journal on Selected Areas in Communications*, **33(12)**: 2510-2523.

Liu, Y., Yu, F. R., Li, X., Ji, H., & Leung, V. C. (2018, April). Hybrid computation offloading in fog and cloud networks with non-orthogonal multiple access. *In IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) IEEE*. (pp. 154-159).

Lyu, X., Ni, W., Tian, H., Liu, R. P., Wang, X., Giannakis, G. B., & Paulraj, A. (2017). Optimal schedule of mobile edge computing for internet of things using partial information. *IEEE Journal on Selected Areas in Communications*, **35(11)**: 2606-2615.

Ning, Z., Dong, P., Wang, X., Hu, X., Liu, J., Guo, L., ... & Leung, V. C. (2020). Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks. *IEEE Transactions on Mobile Computing*, **21(4)**: 1319-1333.

Oo, T. Z., Tran, N. H., Saad, W., Niyato, D., Han, Z., & Hong, C. S. (2016). Offloading in HetNet: A coordination of interference mitigation, user association, and resource allocation. *IEEE Transactions on Mobile Computing*, **16(8)**: 2276-2291.

Seid, A. M., Boateng, G. O., Anokye, S., Kwantwi, T., Sun, G., & Liu, G. (2021). Collaborative computation offloading and resource allocation in multi-UAV-assisted IoT networks: A deep reinforcement learning approach. *IEEE Internet of Things Journal*, **8(15)**: 12203-12218.

Wang, C., Liang, C., Yu, F. R., Chen, Q., & Tang, L. (2017). Computation offloading and resource allocation in wireless cellular networks with mobile edge computing. *IEEE Transactions on Wireless Communications*, **16(8)**: 4924-4938.

Wang, X., Ning, Z., & Guo, S. (2020). Multi-agent imitation learning for pervasive edge computing: A decentralized computation offloading algorithm. *IEEE Transactions on Parallel and Distributed Systems*, **32(2)**: 411-425.

Wang, X., Wang, K., Wu, S., Di, S., Jin, H., Yang, K., & Ou, S. (2018). Dynamic resource scheduling in mobile edge cloud with cloud radio access network. *IEEE Transactions on Parallel and Distributed Systems*, **29(11)**: 2429-2445.

Xiong, Z., Zhang, Y., Niyato, D., Deng, R., Wang, P., & Wang, L. C. (2019). Deep reinforcement learning for mobile 5G and beyond: Fundamentals, applications, and challenges. *IEEE Vehicular Technology Magazine*, **14(2)**: 44-52.

Yi, C., Cai, J., & Su, Z. (2019). A multi-user mobile computation offloading and transmission scheduling mechanism for delay-sensitive applications. *IEEE Transactions on Mobile Computing*, **19(1)**: 29-43.

You, C., Huang, K., Chae, H., & Kim, B. H. (2016). Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Transactions on Wireless Communications*, **16(3)**: 1397-1411.

Yuan, H., & Zhou, M. (2020). Profit-maximized collaborative computation offloading and resource allocation in distributed cloud and edge computing systems. *IEEE Transactions on Automation Science and Engineering*, **18(3)**: 1277-1287.