Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.086401

ARTICLE

Age-Energy Tradeoff in Vehicular MEC: Sensing, Transmission, and Computation Co-Optimization

Hui Zhang[1], **Mangang Xie**[1,*], **Baozhen An**[2] **and Jing Wei**[1]

[1]College of Artificial Intelligence and Computer Science, Northwest Normal University, No. 967 Anning East Road, Anning District, Lanzhou, China
[2]School of Information Science and Engineering, Lanzhou University, No. 222, Tianshui South Road, Chengguan District, Lanzhou, China

*Corresponding Author: Mangang Xie. Email: xiemg@nwnu.edu.cn

Received: 29 May 2026; Accepted: 30 July 2026; Published: 15 September 2026

ABSTRACT: Peak age of information (PAoI) and energy consumption (EC) are conflicting yet critical metrics in mobile edge computing (MEC)-assisted vehicular networks. Most existing studies overlook the joint effects of sensing, transmission, and computation. The main contributions of this work are threefold. First, we derive novel analytical expressions for the average PAoI and average EC under all three strategies, explicitly accounting for the energy and delay costs across the entire data processing chain. Second, we demonstrate that the partial computation offloading strategy is superior, effectively balancing the low latency of local processing with the high power of edge computing. Third, we formulate a weighted optimization problem to navigate the PAoI-EC tradeoff and identify an optimal offloading ratio that dynamically adapts to specific freshness and efficiency requirements. Numerical results demonstrate that jointly optimizing the offloading ratio, edge computing capability, and transmission power significantly improves performance. Our findings offer practical guidelines for designing timely and energy-efficient vehicular telematics systems.

KEYWORDS: Peak age of information; energy consumption; mobile edge computing; vehicular telematics; partial computation offloading

1 Introduction

1.1 Background and Motivation

With the iterative development of communication technologies and processor chip capabilities in recent years, innovative enabling technologies have emerged, accompanied by the proliferation of novel Internet of Things (IoT) applications, such as precision agriculture, remote medical therapy, and smart transportation systems [1–3]. Without exception, most of these applications have huge amounts of data and information to be processed, and these tasks are usually compute-intensive and latency-sensitive, requiring fast transmission rates and powerful processing capabilities to support [4]. Recent mobile edge computing (MEC) studies further emphasize that task offloading decisions should jointly consider timeliness, scalability, and energy costs, especially when multiple users compete for limited edge resources [5]. Vehicular and industrial IoT terminals suffer limited local computing and storage capacity, which leads to excessive latency for heavy sensing tasks; MEC deploys roadside edge servers to compensate for insufficient on-board hardware. Consequently, executing sensing data updates in both a timely and energy-efficient manner remains a paramount challenge.

In vehicular networks, dynamically changing road conditions and real-time vehicle status updates impose stringent temporal constraints on information processing. Maintaining data freshness is critical, as higher freshness enables vehicles to respond swiftly and accurately [6]. Conversely, outdated information may impair decision-making and escalate safety risks. While traditional metrics like throughput and latency are widely studied, they fail to adequately quantify information freshness. To address this limitation, the concept of Age of Information (AoI) has been introduced [7], defined as the time elapsed since the generation of the latest received data update from the destination's perspective. Additionally, energy consumption (EC) remains a pivotal concern in IoT systems. Balancing EC reduction with AoI optimization in vehicular networks constitutes a key research objective.

1.2 Related Work

A large amount of literature has been published in the last few years to study the AoI from different perspectives [8,9]. Peak age of information (PAoI), which is related to AoI, can likewise assess the freshness of information. Literature [10,11] has also done research on it. More recent studies have extended freshness-oriented scheduling to vehicular edge computing-assisted IoT systems and AoI optimization under unknown or delayed AoI observations [12,13]. And the freshness of data is also very important in connected vehicle systems, literature [14] investigated AoI-based vehicular control systems to reduce the communication traffic and improve the freshness of data at the same time. Literature [15] proposed a joint optimization approach for vehicular node update rates and in-vehicle device activation probabilities to minimize average PAoI. A hybrid coding scheme was investigated ... aiming to optimize both AoI and energy efficiency in vehicular communication [16]. Literature [17] further integrated unmanned aerial vehicle (UAV)-assisted relays with reconfigurable intelligent surfaces (RIS) under AoI and EC constraints to enhance decision-making in autonomous driving scenarios. In addition, UAV-assisted and digital-twin-enabled Vehicular Edge Computing (VEC) architectures have recently been investigated for AoI-aware offloading and model migration, confirming that information freshness is tightly coupled with mobility and edge resource dynamics [18,19]. Additionally, the work [20] developed a multi-agent scheduling framework for in-vehicle networks to minimize AoI under strict power and bandwidth limitations.

To enable intelligent driving functionalities, modern vehicles are equipped with extensive sensor arrays for real-time road condition monitoring [21,22]. These sensors continuously generate high-volume data streams, necessitating rapid computational processing to support timely system decisions. For example, Ref. [23] investigated MEC-assisted smart driving to improve the safety of autonomous driving networks, and Ref. [24] proposed a data offloading method for vehicular networks under the MEC architecture. Recent VEC offloading studies further consider intelligent offloading balance and reliability-aware task selection, highlighting the effects of dynamic topology, service reliability, and edge cooperation on latency-sensitive computation [25,26]. Literature [27] developed a non-orthogonal multiple access (NOMA)-assisted MEC analytical framework to characterize in-vehicle terminal mobility and evaluate its impact on task offloading, while literature [28] introduced a resource allocation algorithm for MEC-enabled uplink multiuser networks to minimize the weighted sum of latency and energy consumption. Additionally, MEC demonstrates significant potential in enhancing data freshness: literature [29] explored AoI optimization in UAV-assisted MEC systems through asynchronous offloading, and literature [30] examined distributed task offloading with resource allocation strategies. Some papers also consider PAoI, e.g., the papers [31,32] analyzed the PAoI optimization problem based on preemptive and non-preemptive offloading strategies in MEC systems. Furthermore, MEC exhibits notable energy efficiency advantages. Literature [33] designed a hybrid NOMA-based UAV-MEC network to reduce energy consumption for data-intensive time-critical devices, whereas literature [34] devised an iterative optimization method to minimize EC in cognitive radio-supported MEC systems without compromising quality of service requirements.

1.3 Main Contributions

Based on the above facts, this paper investigates the MEC system that combines the vehicle sensors, the vehicle's own local processor and the edge processor, and analyzes the average PAoI and EC of this system. In this system, the vehicle sensors sense and send computationally intensive packets to the vehicle's local or edge processors, and the packets follow a first-come-first-served (FCFS) policy in the computation and transmission queues, and their computation and transmission times follow an exponential distribution [35]. The main contributions of this paper are summarized as follows.

- We establish a full-chain analytical model for MEC-assisted vehicular telematics, where time-sensitive updates are generated by vehicle sensors and are then processed through local computation, edge computation, or partial computation offloading. Unlike existing AoI-aware MEC studies that mainly focus on transmission/offloading decisions, the proposed model jointly captures sensing-induced packet generation, wireless transmission, queueing delay, local/edge computation, and the corresponding energy costs.
- We derive average PAoI and average EC expressions for the three computation strategies, which take into account sensing as well as transmission and computation energy consumption. We also analyze the impact of offloading ratio, edge processor computation power, and transmission power on them.
- Given the inherent tradeoff between minimizing average PAoI and reducing EC, we propose an age-energy weighted objective function and optimize the offloading ratio to balance these two metrics.

1.4 Organization

The structure of this paper is outlined as follows. Section 2 presents the MEC-based vehicular telematics framework and defines key performance metrics. Section 3 formulates analytical expressions for average PAoI and EC, followed by numerical evaluations in Section 4. Finally, Section 5 concludes the paper with major findings.

2 System Framework and Performance Metric

2.1 System Framework

As illustrated in Fig. 1, the proposed MEC-enabled vehicular telematics system integrates sensing, transmission, and computational processes. The set of user vehicles is denoted as $V_n = \{V_1, V_2, V_3, \ldots, V_n\}$. Various onboard sensors (e.g., ranging sensors, collision monitoring sensors, temperature sensors) generate latency-sensitive, computation-heavy data packets following a Poisson arrival process with arrival rate λ_n. These sensing packets are forwarded to a decision module for task partitioning. We assume each sensor-generated packet carries an identifier (fixed header bits), which enables packet-server mapping. According to these identifiers, sensors route packets toward corresponding local or edge servers.

- *Local computation*: The sensors deployed on the vehicle generate packets of data, which are then all queued up in the vehicle's own processor's compute queue for sequential computation.
- *Partial computation*: Each packet generated by the vehicle sensors is sequentially processed by both the local processor and the edge server via task workload partitioning. Specifically, the local processor first executes a fraction $(1 - p)$ of the total central processing unit (CPU) cycles (e.g., data pre-processing or feature extraction). Subsequently, the remaining fraction p of the task data is placed into the transmission queue to be offloaded over the wireless channel to the edge server, which executes

the remaining final computation. Under this strategy, every packet sequentially passes through all three stages.

- *Edge computation*: The packets generated by the sensors on the vehicle are not computed by the vehicle's own processor, but are all sent sequentially through the transmission queue to the edge processor's computation queue to be computed sequentially.

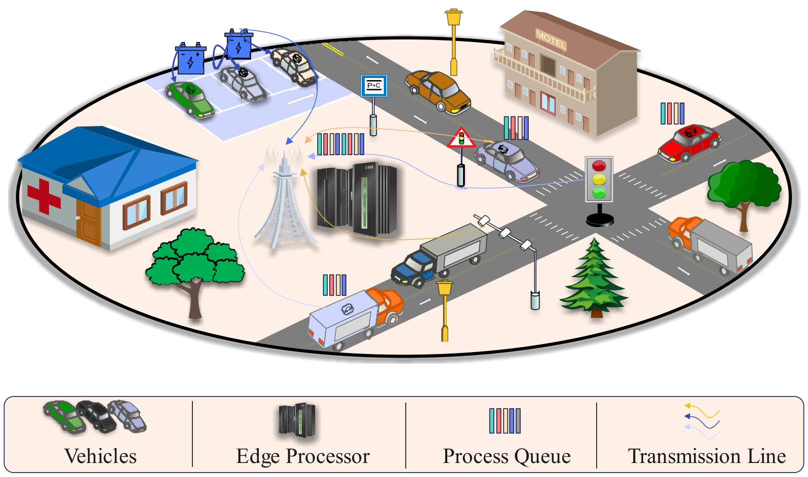


Figure 1: Synchronized transmission of perception and command data in automotive communication networks.

When partial computation offloading is adopted, tasks proceed through a three-stage processing pipeline consisting of local execution, wireless transmission, and edge execution. Within each stage, packets are served in the order of their arrival based on the FCFS principle. For the two extreme boundaries, the local computation scheme ($p = 0$) bypasses the transmission and edge queues, while the edge computation scheme ($p = 1$) bypasses the local processing queue. In addition, each queue is equipped with infinite buffering space to accommodate arbitrary volumes of packets and prevent overflow. The infinite-buffer assumption is adopted to obtain tractable closed-form PAoI expressions and to provide a baseline for evaluating the fundamental age-energy tradeoff. In practical vehicular MEC systems, however, the buffer size is finite. A finite buffer may prevent unbounded waiting time by dropping newly arriving or outdated packets when the queue is full. Such packet dropping can reduce excessive queueing delay and may improve information freshness under heavy traffic, but it can also decrease the probability of successful update delivery and waste sensing/transmission resources. Therefore, finite-buffer constraints would change the PAoI through both packet loss and queueing delay reduction, and the corresponding analysis requires incorporating the buffer size and packet dropping policy into the queueing model. For the subsequent PAoI analysis, it is hypothesized that packet computation and transmission times follow exponential distributions to reflect the stochastic behavior of these time intervals [36]. Specifically, packet transmission time (dependent on wireless channel conditions) may include retransmissions and backoffs. As demonstrated in [36], such complexities are well-captured by exponential distribution assumptions. Similarly, task computation time variability (e.g., influenced by computational complexity in applications like face recognition) is effectively characterized by exponential models [37]. These assumptions extend to MEC-based vehicular telematics systems. The probability density functions (PDFs) are defined as: edge server computation time C_E: $f_{C_E}(t) = \mu_E \exp(-\mu_E t)$; local server computation time C_L: $f_{C_L}(t) = \mu_L \exp(-\mu_L t)$; sensor queue transmission time C_T: $f_{C_T}(t) = \mu_T \exp(-\mu_T t)$. Here, μ_E, μ_T, and μ_L are used to characterize the service capacities of the edge server, communication link, and local server, respectively, where each capacity is measured by the average packet processing rate per unit time. To simplify the analysis, the transmission rate is assumed to be identical for all transmission queues, while processors of the same category are considered to provide uniform computational capability. The offloading ratio p $(0 \leq p \leq 1)$ denotes the proportion of tasks offloaded to the edge server.

The proposed task allocation model includes two limiting scenarios. In the first scenario, all computation requests are completed by the onboard processor without transmitting any packets to the edge infrastructure, resulting in an offloading ratio of $p = 0$. In the second scenario, every computation request is forwarded to the edge server for execution, and the local processor undertakes no computational workload, corresponding to $p = 1$.

2.2 Performance Metric

This study employs two performance metrics: average PAoI and average EC. We first define AoI as follows. Fig. 2 illustrates the temporal evolution of AoI, where the instantaneous AoI at the destination node at moment t is expressed as:

$$\Delta_n(t) = t - u_n(t), \tag{1}$$

where $u_n(t)$ denotes the generation timestamp of the latest received computation packet before time t.

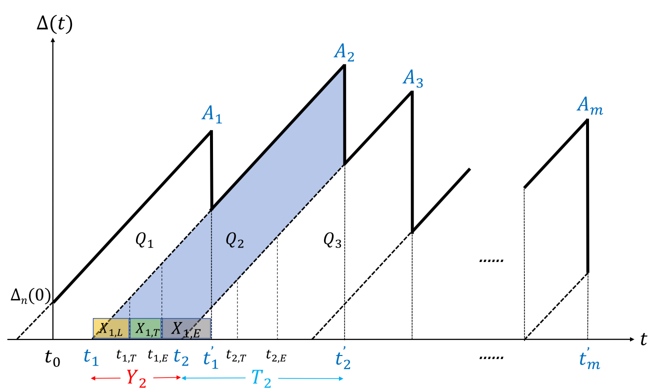


Figure 2: Example of the evolution of AoI for the vehicular networks based on MEC.

Let $P_{n,j}$ denote the j-th data packet $(j = 1, 2, \ldots)$ generated by sensor S_n after $t = 0$. The inter-generation time between consecutive packets $P_{n,j-1}$ and $P_{n,j}$ is defined as

$$Y_j = t_j - t_{j-1}, \tag{2}$$

where t_j corresponds to the generation instant of $P_{n,j}$.

The processing delay T_j, spanning from the generation of $P_{n,j}$ to the availability of its computation result, is formulated as

$$T_j = t_j' - t_j, \tag{3}$$

where t_j' marks the completion time of $P_{n,j}$ computation.

The PAoI value is denoted by $A_j = Y_j + T_j$. Therefore, the average PAoI expression to obtain is

$$\bar{A} = \lim_{n \to \infty} \frac{1}{n} \sum_{j=1}^{n} (Y_j + T_j). \tag{4}$$

Here, $Y_1 = t_1$ is initialized by setting $t_0 = 0$. Upon concluding this interval, the destination node acquires the computation result of the m-th packet.

Secondly, EC is another important performance metric in vehicular telematics-based MEC systems. The average EC expression for the system is

$$\bar{E} = \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} \left(E_j^{\text{Local}} + E_j^{\text{transmit}} + E_j^{\text{mec}} \right), \tag{5}$$

where E_j^{Local}, E_j^{transmit}, E_j^{mec} are the energy consumption of the j-th packet in the local processor, transmission process, and edge processor, respectively.

3 Average PAoI and EC Analysis

In this section, we aim to develop a generalized formula for calculating the average PAoI and average EC in the vehicular telematics based on MEC systems under consideration. Our objective is to comprehensively analyze both the average PAoI and EC, ensuring that our analysis applies broadly without compromising on generality.

We use D_n to denote the size of the computational task, and H_n to denote the number of CPU cycles required to complete the task. Denote by F_{Local} and F_{mec} the computational power of the vehicle's local processor and edge server, respectively. Use μ_L to indicate the service rate of the vehicle's local server, denoted as

$$\mu_L = \frac{F_{\text{Local}}}{(1-p)\,H_n}. \tag{6}$$

The service rate μ_E of mobile edge server is denoted as

$$\mu_E = \frac{F_{\text{mec}}}{pH_n}. \tag{7}$$

The transmission queue service rate μ_T is expressed as

$$\mu_T = \frac{r_n}{pD_n}, \tag{8}$$

where r_n is the transmission rate, denoted as

$$r_n = B\log_2\left(1 + \frac{PW}{N_0}\right), \tag{9}$$

where B represents the channel bandwidth, P represents the uplink transmission power of each vehicle to the server over the channel, W represents the channel gain during the upload of the computing tasks for vehicles, and N_0 represents the white noise power.

3.1 Average PAoI

Since the sensor generates packets according to the Poisson process at a rate of λ_n, we get $\mathbb{E}[Y_j] = \frac{1}{\lambda_n}$ and $\mathbb{E}[Y_j^2] = \frac{2}{\lambda_n^2}$. So according to (4) we calculate the average PAoI as

$$\bar{A} = \frac{1}{\lambda_n} + \mathbb{E}[T_j]. \tag{10}$$

Let $t_{j,L}$ represent the arrival time of packet $P_{n,j}$ at the local server's computation queue, and $t_{j,E}$ denote its arrival time at the edge server's transmission queue. The queuing delays are defined as follows: $X_{j,L}$: queuing delay in the local server's computation queue; $X_{j,T}$: queuing delay in the sensor-aware transmission queue; $X_{j,E}$: queuing delay in the edge server's computation queue. They are denoted as

$$\begin{aligned} X_{j,L} &= t_{j,T} - t_j, \\ X_{j,T} &= t_{j,E} - t_{j,T}, \\ X_{j,E} &= t'_j - t_{j,E}. \end{aligned} \tag{11}$$

Then, T_j can be written as

$$T_j = X_{j,L} + X_{j,T} + X_{j,E}. \tag{12}$$

We denote $W_{j,E}$, $W_{j,T}$, $W_{j,L}$ represent the waiting times of packet $P_{n,j}$ in the edge server computation queue, sensor transmission queue, and local server computation queue, respectively. Similarly, $C_{j,E}$, $C_{j,T}$ and $C_{j,L}$ denote the computation time at the edge server, transmission time, and local server computation time for $P_{n,j}$. The total queuing delay at each queue is the sum of the corresponding waiting time and processing time, leading to the following expressions

$$\begin{aligned} X_{j,E} &= W_{j,E} + C_{j,E}, \\ X_{j,T} &= W_{j,T} + C_{j,T}, \\ X_{j,L} &= W_{j,L} + C_{j,L}. \end{aligned} \tag{13}$$

By substituting (13) into (12), the total processing delay T_j becomes

$$T_j = W_{j,E} + C_{j,E} + W_{j,T} + C_{j,T} + W_{j,L} + C_{j,L}. \tag{14}$$

Since the process of generating packets by sensor sensing is uncorrelated with packet transmission and computation, substituting (14) into (10) and utilizing the fact that the packet computation times $C_{j,E}$, $C_{j,T}$, $C_{j,L}$ are independent of Y_j yields

$$\bar{A} = \frac{1}{\lambda_n} + \mathbb{E}[C_{j,E}] + \mathbb{E}[C_{j,T}] + \mathbb{E}[C_{j,L}] + \mathbb{E}[W_{j,E}] + \mathbb{E}[W_{j,T}] + \mathbb{E}[W_{j,L}]. \tag{15}$$

Notice that PAoI tends to infinity when any one or more of $W_{j,E}$, $W_{j,T}$, $W_{j,L}$ expectations tend to infinity. Therefore, to ensure the stability of the system, it is assumed that the arrival rate of packets in each queue is lower than its service rate.

Then find the average PAoI. Based on the above derivation, the average PAoI under the partial computation offloading scheme is calculated as

$$\bar{A} = \frac{1}{\lambda_n} + \frac{1}{\mu_E} + \frac{1}{\mu_T} + \frac{1}{\mu_L} + \mathbb{E}[W_{j,E}] + \mathbb{E}[W_{j,T}] + \mathbb{E}[W_{j,L}]. \tag{16}$$

Lemma 1. *For a Poisson process $K(t)$ with rate λ and an exponential random variable $X(\mathbb{E}[X] = 1/\gamma)$, the number of arrivals $S = K(X)$ within the interval $[0, X]$ follows a geometric distribution [38]. This probability mass function (PMF) is expressed by the following equation*

$$Pr(K(X) = k) = (1-\alpha)\alpha^k, k \geqslant 0, \tag{17}$$

where $\alpha = \lambda/(\gamma + \lambda)$.

Applying (17), the PMF for the packet count in the edge server queue during $P_{n,j}$ generation is

$$\Pr\left(K_{j,e} = k\right) = \left(\frac{\lambda}{\mu_E}\right)^k \left(1 - \frac{\lambda}{\mu_E}\right), \tag{18}$$

thus, we obtain the expression for $\mathbb{E}[W_{j,E}]$ as

$$\mathbb{E}[W_{j,E}] = \mathbb{E}[K_{j,E}]\mathbb{E}[C_D] = \frac{\lambda}{\mu_E(\mu_E - \lambda)}. \tag{19}$$

Similarly, we compute $\mathbb{E}[W_{j,T}]$ and $\mathbb{E}[W_{j,L}]$ as

$$\mathbb{E}[W_{j,T}] = \frac{\lambda}{\mu_T(\mu_T - \lambda)}, \tag{20}$$

$$\mathbb{E}[W_{j,L}] = \frac{\lambda_n}{\mu_L(\mu_L - \lambda_n)}. \tag{21}$$

We then substitute (19)–(21) into (16) to obtain a closed expression for the average PAoI under the partial computation offloading scheme with an offloading ratio of p as

$$\bar{A} = \frac{1}{\lambda_n} + \frac{1}{\mu_E} + \frac{1}{\mu_T} + \frac{1}{\mu_L} + \frac{\lambda}{\mu_E(\mu_E - \lambda)} + \frac{\lambda}{\mu_T(\mu_T - \lambda)} + \frac{\lambda_n}{\mu_L(\mu_L - \lambda_n)}, \tag{22}$$

where $\lambda = \sum_{n=1}^{N} \lambda_n$.

3.2 Average EC

We denote E_{Local} as the energy consumed by a packet that needs to be computed on the vehicle's local processor, computed as

$$E_{\text{Local}} = k_1(F_{\text{Local}})^2 H_n(1-p), \tag{23}$$

where k_1 denotes the intrinsic coefficient of the vehicle processor.

Denote T_{trans} as the time during transmission of a packet that needs to be transmitted to the edge processor's computation queue for processing, calculated as

$$T_{\text{trans}} = \frac{D_n p}{r_n}. \tag{24}$$

Denote E_{trans} as the EC of the packet during transmission, calculated as

$$E_{\text{trans}} = P T_{\text{trans}}. \tag{25}$$

We denote E_{mec} as the EC of a packet that needs to be computed in the edge processor during processing, computed as

$$E_{\text{mec}} = k_2 (F_{\text{mec}})^2 H_n p, \tag{26}$$

where k_2 denotes the intrinsic coefficient of the edge processor.

In some of the partial computation offloading strategy we have considered, the total EC of the MEC-assisted vehicular telematics system includes the energy consumed for computation at the vehicle's local processor, the energy consumed for packet transmission to the edge processor, and the energy consumed for computation at the edge processor, denoted as

$$E = E_{\text{Local}} + E_{\text{trans}} + E_{\text{mec}}. \tag{27}$$

We then substitute (23), (25) and (26) into (27) to obtain a closed expression for the average EC under the partial computation offloading scheme with an offloading ratio of p as

$$\overline{E} = \frac{1}{N}\left(k_1 (F_{\text{Local}})^2 [H_n (1-p)] + \frac{P(D_n p)}{r_n} + k_2 (F_{\text{mec}})^2 (H_n p) \right). \tag{28}$$

3.3 Age-Energy Tradeoff

In vehicular MEC systems, information freshness conflicts with energy consumption. Autonomous driving and real-time traffic control require low-latency sensing data processing with limited power budgets, which motivates our joint optimization design. To address these dual requirements, we aim to optimize key control parameters, including the offloading ratio p and transmission power P. However, a fundamental conflict exists: minimizing PAoI often necessitates frequent data updates and rapid processing, which inherently escalates EC; conversely, energy conservation strategies (e.g., reducing transmission frequency or computational load) may degrade information freshness. To resolve this inherent tradeoff, we construct a weighted cost function to balance PAoI and EC.

$$C = \alpha \bar{A} + \beta \bar{E}, \tag{29}$$

where $\alpha + \beta = 1$, with α and β representing the weights assigned to PAoI and EC, respectively. The optimization problem is formulated to minimize C

$$P_1: \quad \arg\min_{p} C_{\text{partial}}(p), \tag{30a}$$

$$\text{s.t.} \quad p_{\min} \le p \le p_{\max}, \tag{30b}$$

where $p_{\min}$ and $p_{\max}$ denote the bounds of the offloading ratio, and the subscript "partial" corresponds to the partial computation offloading strategy.

Since the weighted cost under the partial computation offloading strategy exhibits a unimodal trend with respect to the offloading ratio in the considered parameter region, the optimal offloading ratio can be efficiently searched by a bisection-based method. Compared with the exhaustive search, the proposed algorithm significantly reduces the number of candidate evaluations while achieving nearly the same weighted cost.

To solve P_1 and find the optimal offloading ratio p^* under a partial computation offloading strategy, we first employ an exhaustive search to identify the optimal offloading ratio p^*. For computational efficiency, a suboptimal binary search-based algorithm is further proposed, which is detailed in Algorithm 1.

Algorithm 1: Bisection method for solving P_1

Require: λ_n, H_n, D_n, F_{mec}, F_{Local}, $p_{\min}$, $p_{\max}$, convergence tolerance ε, gradient perturbation step δ
Ensure: p^*
Let $L = p_{\min}$, $R = p_{\max}$
while $(R - L) > \epsilon$ **do**
  $M = (L + R)/2$
  $C_l = C_{\text{partial}}(M - \delta)$ {Evaluate cost at left perturbation}
  $C_r = C_{\text{partial}}(M + \delta)$ {Evaluate cost at right perturbation}
  $\text{Grad} = (C_r - C_l)/(2 \cdot \delta)$ {Central finite difference gradient}
  if $|\text{Grad}| < \varepsilon$ **then**
    $p^* = M$
    break
  else if Grad > 0 **then**
    $R = M$ {The objective function is increasing; minimum lies to the left}
  else
    $L = M$ {The objective function is decreasing; minimum lies to the right}
  end if
end while
$p^* = (L + R)/2$

4 Numerical Simulations

Based on the mathematical analysis and theoretical derivations in Section 3, this section presents numerical results under various strategies, along with a comparison with Monte Carlo simulation experiments, where all vehicles are assumed to have the same update generation rate λ_n. Unless otherwise specified, system-related parameters and their corresponding values are listed in Table 1.

Table 1: System-related parameters.

Notation	Value	Notation	Value	Notation	Value	Notation	Value
k_1	0.05	P	24 dBm	k_2	0.01	N_0	−80 dBm
D_n	100 Mbits	λ_n	0.4	H_n	0.8 Gcycles/s	N	4
B	10 MHz	F_{Local}	1.0 Gcycles/s	W	−10 dBm	F_{mec}	30 Gcycles/s

To validate the analytical expressions, we further conduct a discrete-event Monte Carlo simulation. In each simulation run, packet arrivals are generated according to a Poisson process, and the transmission, local computation, and edge computation times are independently generated according to exponential distributions with the corresponding service rates. Packets are served following the FCFS discipline in each queue. The average PAoI and EC are obtained by averaging over a sufficiently large number of delivered packets after removing the initial transient period. The simulated results are then compared with the analytical results derived in Section 3. This comparison is presented not only for the main performance curves in Figs. 3–6, but also for the weighted cost vs. α/β in Fig. 7b,c, where simulation markers are overlaid on the analytical lines to validate the theoretical derivations under different processor capabilities and transmission powers. We evaluate the performance of the proposed framework by comparing it against two boundary baselines—pure local computation ($p = 0$) and pure edge computation ($p = 1$)—as well as two static-ratio partial computation offloading configurations (specifically $p = 0.3$ and $p = 0.7$). For all evaluated configurations, identical wireless channel parameters, transmission powers, and computational capabilities are maintained to guarantee a rigorous and fair comparison. This comprehensive suite demonstrates how dynamically tuning the offloading ratio over changing network conditions outperforms both single-node processing boundaries and static unoptimized offloading heuristics. We further compare the proposed bisection-based search with exhaustive search. The exhaustive search evaluates all candidate offloading ratios within the feasible interval, whereas the bisection-based method iteratively narrows the search range. Simulation results show that the two methods obtain almost identical optimal offloading ratios, while the bisection-based method requires far fewer objective-function evaluations. This confirms the computational efficiency of Algorithm 1.

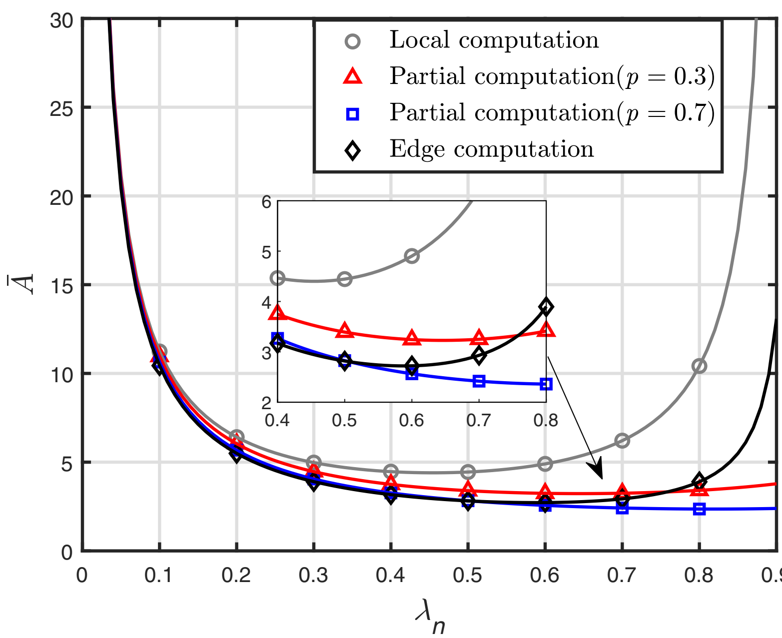


Figure 3: Average PAoI on the update generation rate λ_n.

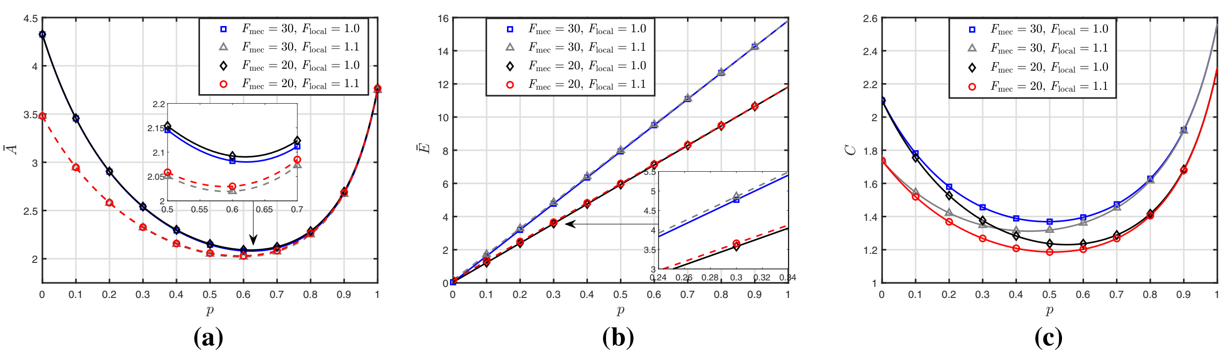
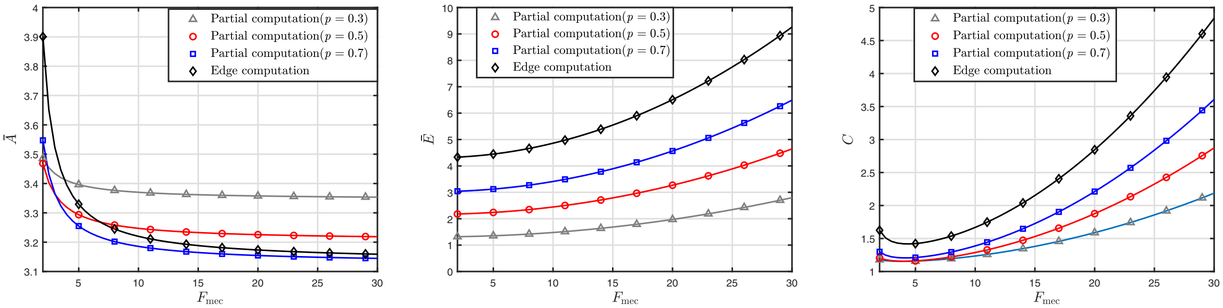
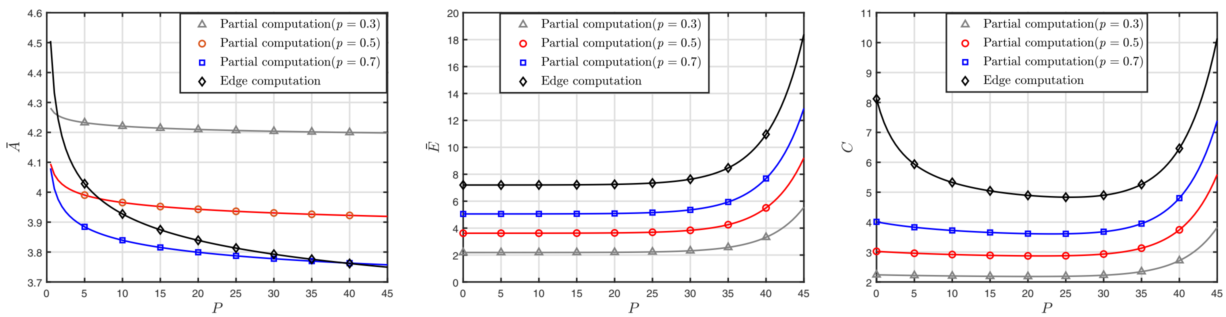


Figure 4: Offloading ratio p vs. average PAoI, average EC, weighted sum C for different F_{mec} and F_{Local}. (**a**) PAoI presents U-shape with optimal p^*, limited by local bottleneck at small p and transmission congestion at large p. (**b**) EC grows monotonically with p due to extra transmission and edge computation cost. (**c**) Weighted cost C is also U-shaped; stronger local processor yields smaller p^*, while stronger edge server yields larger p^*.



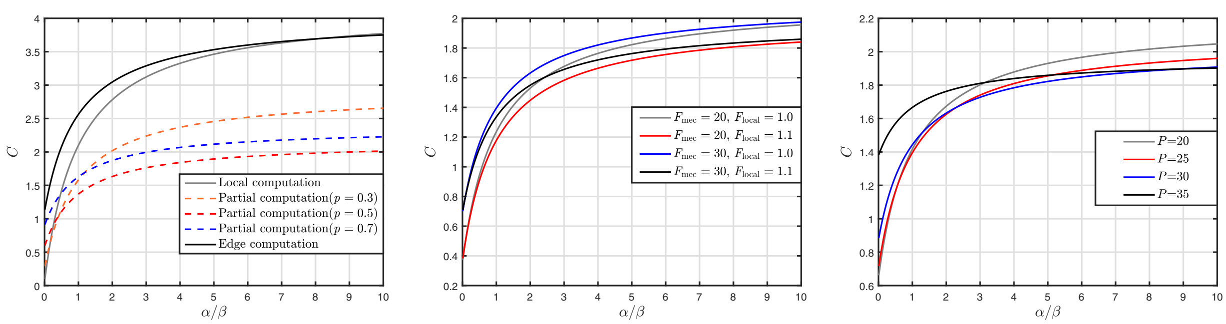
(a) (b) (c)

Figure 5: Edge processor computing power F_{mec} vs. average PAoI, average EC, weighted sum C for different computation strategy. (**a**) PAoI decreases with F_{mec}, and partial offloading outperforms full-edge strategy. (**b**) EC rises with F_{mec} as edge energy scales with F_{mec}^2. (**c**) There exists an optimal F_{mec} to minimize C, since excessive edge power brings heavy energy penalty.



(a) (b) (c)

Figure 6: Transmission power P vs. average PAoI, average EC, weighted sum C for different computation strategy. (**a**) Higher P reduces PAoI; pure local computation has fixed PAoI without transmission. (**b**) EC monotonically increases with P, and larger p leads to steeper growth. (**c**) C follows U-shape with optimal P within 20–30 dBm.



(a) (b) (c)

Figure 7: Weighted sum C vs. α/β. (**a**) C rises and saturates with larger α/β, and partial offloading achieves the minimal cost. (**b**) Larger F_{mec} reduces C, with wider performance gaps when prioritizing PAoI. (**c**) High P cuts C under energy priority but raises C under freshness priority. Simulations match analytical curves to validate derivations.

Fig. 3 shows that the partial computation offloading strategy achieves the lowest average PAoI across all λ_n, outperforming both pure local and pure edge strategies. Local computation suffers from severe queueing overload when $\lambda_n > 0.6$, while edge computation is bottlenecked by the transmission link when $\lambda_n > 0.8$.

These results confirm that balancing local processing and edge offloading is essential for maintaining information freshness.

In order to investigate the effect of the offloading ratio on the system, Fig. 4 shows the relationship between the average PAoI, average EC and the weighted sum of the average AoI and EC vs. the offloading ratio p. Firstly, from Fig. 4a, it can be seen that the average PAoI decreases and then increases with increasing p for different F_{mec} and local, which indicates that there exists an optimal p^* that makes the average PAoI of the system to be the lowest, in addition, it can be seen that, when p is small, more packets are computed at the local processor, so the two curves with larger F_{Local} have a lower average PAoI. As p increases, more packets are computed at the edge processor. Since the edge processor can easily meet the computational demand, the gap between the four curves keeps decreasing and finally even overlaps. As shown in Fig. 4b, despite the edge processor's computational superiority, its higher energy demands—combined with transmission energy costs—lead to a monotonic rise in average EC as p increases. While offloading more tasks to edge servers accelerates processing speed, this gain is counterbalanced by significantly elevated EC. To evaluate the PAoI-EC tradeoff, Fig. 4c illustrates the weighted sum C against p. Assuming equal priority for PAoI and EC ($\alpha = \beta = 0.5$), C exhibits a U-shaped trend akin to Fig. 4a, initially decreasing then increasing with p. Moreover, for different F_{mec} and F_{Local}, there always exists an optimal offloading ratio p^* to minimize C.

The edge processor computational power has an equally important impact on the system, and since the edge processor is not involved in data processing in the local computation strategy, the strategy is not considered in this part. Fig. 5 illustrates the edge processor computational power vs. average PAoI, average EC, weighted sum C for both partial and edge computation strategies. In Fig. 5a, as F_{mec} rises, the average PAoI declines across all four curves. Whereas in Fig. 5b, the system's average EC grows with higher F_{mec} and p, as elevating these parameters amplifies both computational and transmission energy consumption. To analyze the balance between average PAoI and EC, Fig. 5c illustrates the impact of F_{mec} on the weighted sum C, with $\alpha = \beta = 0.5$.

System transmission power is also a key metric, and Fig. 6 plots the relationship of average PAoI, average EC, weighted sum C with transmission power P under different strategies. The average PAoI of the system in Fig. 6a decreases as P increases, and the more packets need to be transmitted, the greater the impact, especially for the strategy of computing at the edge. The average EC increases with increasing P in Fig. 6b, and the magnitude of the change is more pronounced the larger the offloading ratio p is, because the larger P is, the more energy is consumed for transmission, and a larger offloading ratio P increases the data volume for wireless transmission, which leads to an increase in the system EC. As for the weighting and C, it can be seen in Fig. 6c that P optimizes C for both partial and edge computing strategies in the interval of 20 to 30.

As a complement to the above study, Fig. 7 plots the weighted cost C vs. the weight ratio α/β under different configurations, where solid lines represent analytical results and markers denote Monte Carlo simulation results. Specifically, Fig. 7a compares the three computation strategies (local, edge, and partial) with fixed system parameters, showing that C increases monotonically and eventually saturates as α/β grows, and the partial offloading strategy consistently achieves the lowest cost. Fig. 7b examines the impact of edge processor computing power F_{mec} under the partial offloading strategy. The analytical curves and the overlaid simulation markers exhibit close agreement, confirming the accuracy of our derivations. It is observed that a higher F_{mec} reduces the weighted cost for all α/β, but the reduction is more pronounced when α/β is small (i.e., when energy efficiency is prioritized). Fig. 7c further investigates the effect of transmission power P under the same partial offloading strategy. The simulation results again match the analytical predictions closely. We find that increasing P lowers C when α/β is relatively small, but this benefit gradually diminishes as α/β becomes large, because the energy consumption penalty dominates the weighted objective. In all subfigures, the close alignment between analytical and simulation results validates the proposed theoretical framework.

It has been found through previous studies that some of the computational strategies achieve better performance, and for different F_{mec} there always exists an optimal p^* to minimize C. As observed through numerical simulations, considering $F_{\text{Local}} = 1$ Gcycles/s for all strategies, Fig. 8 evaluates the performance of the proposed algorithm for $F_{\text{mec}} = 35, 30, 25, 20$. Algorithm 1 yields a sub-optimal offloading ratio when $F_{\text{mec}} = 35$, $F_{\text{mec}} = 25$, $F_{\text{mec}} = 20$. Specifically, for $F_{\text{mec}} = 35$, the exhaustive search yields $p^* = 0.44$, while Algorithm 1 produces $p^* = 0.45$. At $F_{\text{mec}} = 25$, the exhaustive method results in $p^* = 0.53$, compared to $p^* = 0.52$ for Algorithm 1. When $F_{\text{mec}} = 20$, the exhaustive approach achieves $p^* = 0.55$, whereas Algorithm 1 attains $p^* = 0.54$. Notably, at $F_{\text{mec}} = 30$, both Algorithm 1 and the exhaustive search yield identical optimal offloading ratios($p^* = 0.49$). These experimental findings confirm that Algorithm 1 achieves a near-optimal solution with reduced computational complexity.

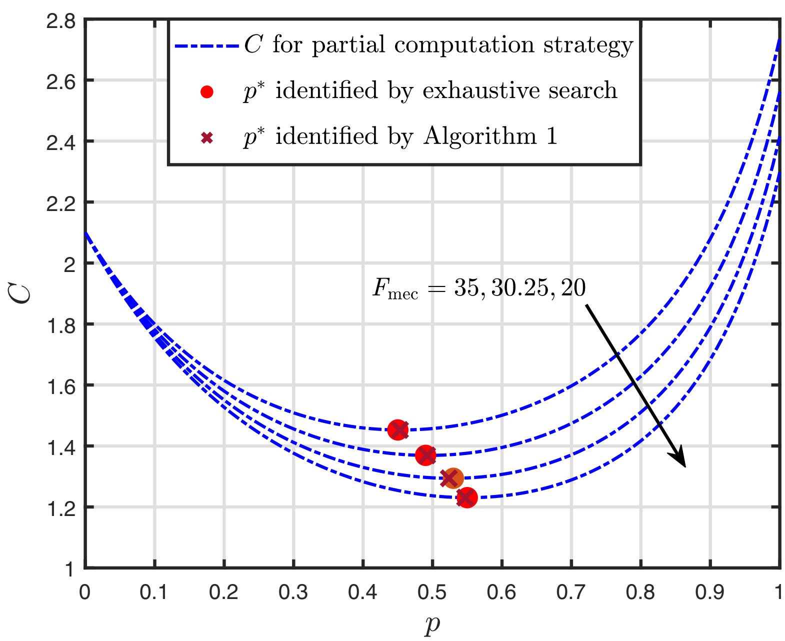


Figure 8: Optimal p^* for partial computation offloading strategies under different F_{mec}.

As shown in Fig. 9, the proposed adaptive workload partitioning scheme outperforms all four benchmark algorithms under all packet arrival intensities: The traditional delay-energy MEC only optimizes local-edge computation delay without joint transmission queue constraints and end-to-end sensing PAoI modeling, leading to persistently higher weighted cost compared with our joint optimization framework. The PAoI-only minimization strategy blindly offloads most tasks to edge servers to reduce PAoI, which introduces excessive transmission energy consumption and keeps its cost above the proposed scheme. The deep Q-network (DQN) reinforcement learning offloading suffers inherent flow-dependent training bias and cannot provide closed-form global optimal partitioning solutions. The static fixed ratio $p = 0.5$ heuristic lacks dynamic adjustment capacity for time-varying arrival intensity, resulting in a stable performance gap with our analytical optimal strategy.

The experimental results highlight that the average PAoI and EC of various computation strategies are contingent on the offloading ratio, and an optimal ratio can invariably be identified to achieve a balance between these metrics. Furthermore, the selection of computational capabilities and transmission power in edge processors plays a pivotal role in optimizing age and energy efficiency within MEC-assisted vehicular telematics systems.

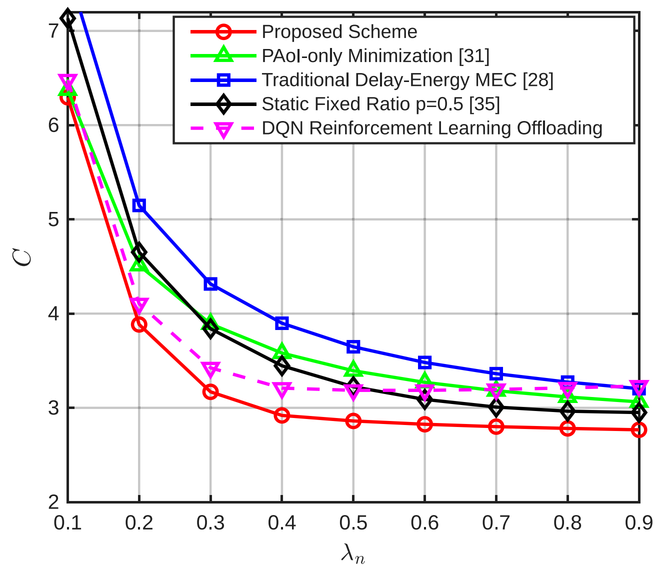


Figure 9: Weighted cost C vs. update generation rate λ_n under different state-of-the-art benchmarks.

5 Conclusion

This paper has investigated the critical tradeoff between information freshness and energy consumption in MEC-assisted vehicular networks. Our main contributions are threefold. First, we developed a novel analytical framework that jointly models the sensing, transmission, and computation processes, deriving closed-form expressions for the average PAoI and EC under local, edge, and partial computation offloading strategies. Second, we demonstrated that the partial computation offloading strategy is superior, effectively balancing the low latency of local processing with the high power of edge computing to achieve a better PAoI-EC tradeoff. Third, we formulated a weighted optimization problem, identifying that an optimal offloading ratio exists and that system performance can be finely tuned via the edge processor's computing power and transmission power. In summary, this paper provides theoretical closed-form derivations and a lightweight optimization method to build low-latency, energy-efficient vehicular MEC systems.

Acknowledgement: None.

Funding Statement: This work was supported in part by the National Natural Science Foundation of China under Grants 62501483.

Author Contributions: Hui Zhang: conceptualization, methodology, software, validation, formal analysis. Mangang Xie: conceptualization, methodology, supervision, formal analysis, writing—review & editing. Baozhen An: conceptualization, methodology, software, validation, writing—original draft, writing—review & editing. Jing Wei: validation, writing—review & editing. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: All simulation parameters and theoretical formulas in this manuscript can reproduce the numerical results.

Ethics Approval: Not applicable, no human/animal subjects involved.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Abdelfattah IS, Abd El-Malek AH, Zakariya AY, Rabia SI. Minimizing age of information and energy consumption in a computation-intensive status update system. In: Proceedings of the 2025 IEEE Wireless Communications and Networking Conference (WCNC); 2025 Mar 24–27; Milan, Italy.
2. Wang W, Zhu X. Reliable multidimensional resource scheduling for heterogeneous computing networks via coded distributed computing and hypergraph neural networks. IEEE Internet Things J. 2025;12(22):48242–61. doi:10.1109/JIOT.2025.3604217.
3. Mohapatra H. Task offloading and edge computing in IoT—gaps, challenges and future directions. Comput Mater Contin. 2026;87(3):8. doi:10.32604/cmc.2026.076726.
4. Muhammad A, Sorkhoh I, Samir M, Ebrahimi D, Assi C. Minimizing age of information in multiaccess-edge-computing-assisted IoT networks. IEEE Internet Things J. 2021;9(15):13052–66. doi:10.1109/JIOT.2021.3139044.
5. Aggarwal S, uzZaman MA, Bastopcu M, Ulukus S, Basar T. Distributed offloading in multi-access edge computing systems: a mean-field perspective. IEEE Trans Mob Comput. 2026;25(6):7592–610. doi:10.1109/TMC.2025.3642928.
6. Ji B, Zhang X, Mumtaz S, Han C, Li C, Wen H. Survey on the internet of vehicles: network architectures and applications. IEEE Commun Stand Mag. 2020;4(1):34–41. doi:10.1109/MCOMSTD.001.1900053.
7. Xie M, Gong J, Jia X, Wang Q, Ma X. Age and energy analysis for L-th best relay enabled cooperative status update systems with short packet communications. IEEE Trans Veh Technol. 2023;72(10):6294–308. doi:10.1109/TVT.2022.3233406.
8. Hosseini SS, Azmi P, Mokari N. Minimizing average age of information in reliable covert communication on time-varying channels. IEEE Trans Veh Technol. 2024;73(1):651–9. doi:10.1109/TVT.2023.3303674.
9. Song C, Xie M, Lu G, Zhang S, Jia X. Adaptive power partitioning and frame length optimization for AoI minimization in frame slotted ALOHA-assisted NOMA networks. J King Saud Univ Comput Inf Sci. 2026;38:359. doi:10.1007/s44443-026-00764-w.
10. Xie M, Wei J, Jia X, Wang H, Wang Q. Collaborative trajectory and resource optimization for AoI minimization in multi-UAV networks. IEEE Trans Commun. 2026;74:9327–42. doi:10.1109/TCOMM.2026.3698849.
11. Hui ETZ, Madhukumar AS. Mean peak age of information analysis of energy-aware computation offloading in IIoT networks. In: Proceedings of the 2024 IEEE 99th Vehicular Technology Conference (VTC-Spring); 2024 Jun 24–27; Singapore, Singapore. p. 1–6. doi:10.1109/VTC2024-Spring62846.2024.10683346.
12. Xie X, Zhong T, Wang H. Scheduling for maximizing the information freshness in vehicular edge computing-assisted IoT systems. IEEE Trans Intell Transp Syst. 2025;26(3):4140–51. doi:10.1109/TITS.2024.3514099.
13. Zhao Z, Kadota I. Optimizing age of information without knowing the age of information. In: Proceedings of the IEEE INFOCOM 2025—IEEE Conference on Computer Communications; 2025 May 19–22; London, UK. p. 1–10. doi:10.1109/INFOCOM55648.2025.11044597.
14. Onozuka Y, Imae A, Sakaue Y, Tamaki S, Taguchi K, Narikawa S, et al. AoI-based communication traffic reduction for networked railway vehicle control. In: Proceedings of the 2024 IEEE International Conference on Industrial Technology (ICIT); 2024 Mar 25–27; Bristol, UK. p. 1–6. doi:10.1109/ICIT58233.2024.10541007.
15. Li Z, Xiang L, Ge X. Age of information modeling and optimization for fast information dissemination in vehicular social networks. IEEE Trans Veh Technol. 2022;71(5):5445–59. doi:10.1109/TVT.2022.3154766.
16. Xie M, An B, Jia X, Zhou M, Lu J. Simultaneous update of sensing and control data using free-ride codes in vehicular networks: an age and energy perspective. Comput Netw. 2024;252:110667. doi:10.1016/j.comnet.2024.110667.
17. Qi W, Yang C, Song Q, Guan Y, Guo L, Jamalipour A. Minimizing age of information for hybrid UAV-RIS-assisted vehicular networks. IEEE Internet Things J. 2024;11(10):17886–95. doi:10.1109/JIOT.2024.3359666.
18. Goudarzi S, Soleymani SA, Anisi MH, Jindal A, Xiao P. Optimizing UAV-assisted vehicular edge computing with age of information: a SAC-based solution. IEEE Internet Things J. 2025;12(5):4555–69. doi:10.1109/JIOT.2025.3529836.
19. Chen X, Bi Y, Xing H, Zheng D, Marina MK. Model migration in digital twin-empowered vehicular edge computing with AoI-aware decentralized bilevel learning. IEEE Trans Mob Comput. 2026;25(3):3953–68. doi:10.1109/TMC.2025.3621052.

20. Wang J, Cao T, Wang X, Zhang M, Guan J. Resource scheduling in vehicular networks with age of information and channel awareness. In: Proceedings of the 2022 IEEE/CIC International Conference on Communications in China (ICCC); 2022 Aug 11–13; Foshan, China. p. 344–9. doi:10.1109/ICCC55456.2022.9880785.
21. Qiu B, Wang Y, Xiao H, Zhang Z. Deep reinforcement learning-based adaptive computation offloading and power allocation in vehicular edge computing networks. IEEE Trans Intell Transp Syst. 2024;25(10):13339–49. doi:10.1109/TITS.2024.3391831.
22. Sorkhoh I, Assi C, Ebrahimi D, Sharafeddine S. Optimizing information freshness for MEC-enabled cooperative autonomous driving. IEEE Trans Intell Transp Syst. 2022;23(8):13127–40. doi:10.1109/TITS.2021.3119961.
23. Zhang X, Yang B, Yu Z, Cao X, Alexandropoulos GC, Zhang Y, et al. Reconfigurable intelligent computational surfaces for MEC-assisted autonomous driving networks: design optimization and analysis. IEEE Trans Intell Transp Syst. 2025;26(1):1286–303. doi:10.1109/TITS.2024.3486555.
24. Huang CM, Chiang MS, Dao DT, Su WL, Xu S, Zhou H. V2V data offloading for cellular network based on the software defined network (SDN) inside mobile edge computing (MEC) architecture. IEEE Access. 2018;6:17741–55. doi:10.1109/ACCESS.2018.2820679.
25. Wu Y, Fang X, Min G, Chen H, Luo C. Intelligent offloading balance for vehicular edge computing and networks. IEEE Trans Intell Transp Syst. 2025;26(5):5792–803. doi:10.1109/TITS.2025.3549493.
26. Ma L, An Y, Liu S, Xiao Y, Wu N, Li Y. PRITO: performance-reputation integrated task offloading for reliable vehicular edge computing. IEEE Trans Mob Comput. 2026;25(6):7716–34.
27. Li L, Fan P. Latency and task loss probability for NOMA assisted MEC in mobility-aware vehicular networks. IEEE Trans Veh Technol. 2023;72(5):6891–5. doi:10.1109/TVT.2022.3233195.
28. Zhang H, Liu X, Xu Y, Li D, Yuen C, Xue Q. Partial offloading and resource allocation for MEC-assisted vehicular networks. IEEE Trans Veh Technol. 2023;73(1):1276–88. doi:10.1109/TVT.2023.3306939.
29. Qin X, Li Y, Ma N, Zhang Y, Han K, Meng L, et al. Timeliness-oriented asynchronous task offloading in UAV-edge-computing systems. IEEE Trans Netw Sci Eng. 2024;11(1):900–12. doi:10.1109/tnse.2023.3310055.
30. Kim M, Jang J, Choi Y, Yang HJ. Distributed task offloading and resource allocation for latency minimization in mobile edge computing networks. IEEE Trans Mob Comput. 2024;23(12):15149–66. doi:10.1109/TMC.2024.3440124.
31. Zhu J, Gong J. Optimizing peak age of information in MEC systems: computing preemption and non-preemption. IEEE/ACM Trans Netw. 2024;32(4):3285–300. doi:10.1109/TNET.2024.3384706.
32. Jiang Y, Liu J, Humar I, Chen M, AlQahtani SA, Hossain MS. Age of information based computation offloading and transmission scheduling in mobile-edge-computing-enabled IoT networks. IEEE Internet Things J. 2023;10(22):19782–94. doi:10.1109/JIOT.2023.3283287.
33. Kota NR, Naidu K. Minimizing energy consumption in H-NOMA based UAV-assisted MEC network. IEEE Commun Lett. 2023;27(9):2536–40. doi:10.1109/LCOMM.2023.3300467.
34. Elnaim AA, Babeker A, Barakat M, Gaid ASA, Saeed MM, Ahmed ZE, et al. Energy consumption for cognitive radio network enabled multi-access edge computing. In: Proceedings of the 2023 3rd International Conference on Emerging Smart Technologies (eSmarTA); 2023 Oct 10–11; Taiz, Yemen. p. 1–5. doi:10.1109/eSmarTA59349.2023.10293270.
35. Yu R, Liu Y, Lan X, Chen Q. Optimal offloading percentage for fresh status updates in mobile edge computing networks. In: Proceedings of the 2022 7th International Conference on Computer and Communication Systems (ICCCS); 2022 Apr 22–25; Wuhan, China. p. 524–9. doi:10.1109/ICCCS55155.2022.9845977.
36. Yates RD, Kaul SK. The age of information: real-time status updating by multiple sources. IEEE Trans Inf Theory. 2019;65(3):1807–27. doi:10.1109/TIT.2018.2871079.
37. Sthapit S, Thompson J, Robertson NM, Hopgood JR. Computational load balancing on the edge in absence of cloud and fog. IEEE Trans Mob Comput. 2019;18(7):1499–512. doi:10.1109/TMC.2018.2863301.
38. Tang Z, Sun Z, Yang N, Zhou X. Age of information of multi-user mobile-edge computing systems. IEEE Open J Commun Soc. 2023;4:1600–14. doi:10.1109/OJCOMS.2023.3294942.