Вісник Тернопільського національного технічного університету
https://doi.org/10.33108/visnyk_tntu
Scientific Journal of the Ternopil National Technical University
2025, № 4 (120) https://doi.org/10.33108/visnyk_tntu2025.04
ISSN 2522-4433. Web: ***visnyk.tntu.edu.ua***

UDC 004.75

EDGE COMPUTING AND DATA OPTIMIZATION FOR REMOTE DIAGNOSTICS OF VEHICLES

Vitalii Bonar; Volodymyr Hotovych; Liubomyr Matiichuk

Ternopil Ivan Puluj National Technical University, Ternopil, Ukraine

Abstract. *This paper proposes the use of edge computing for remote vehicle diagnostics. Instead of continuously streaming raw telemetry, the on-board computer performs simplified analytics, adaptive sampling, and data compression, so that the outgoing network channel transmits event packets and short periodic summaries (summaries) of data. The system design focuses on three goals: to maintain the ability to make important diagnostic decisions, to keep the reconstruction error acceptable for analysis, and to notify of a fault even with weak or variable connectivity in a reasonable time. The paper proposes a simple and reproducible way to account for data volume and latency, shows how to configure basic controls for each signal, and discusses the use of multi-peripheral edge computing (MEC) to reduce latency. The result is significant savings in outbound traffic without compromising diagnostic capability.*

Key words: *remote diagnostics, edge computing, telemetry optimization, localized processing, diagnostics of vehicles.*

Submitted 05.09.2025
Revised 29.10.2025
https://doi.org/10.33108/visnyk_tntu2025.04.039 *Published 27.01.2026*

1. INTRODUCTION

Modern vehicles are distributed cyber-physical systems, the electronic control units (ECUs) of which produce telemetry of various nature: temperature and pressure in different systems, battery status, brakes status, various indicators of differecnt subsystems etc. The protocol used by the ECUs – UDS (Unified Diagnostic Services, ISO 14229), unifies the process of reading data by identifier, reporting and clearing diagnostic trouble codes (DTC), as well as performing various procedures. The DoIP (Diagnostics over Internet Protocol, ISO 13400) protocol transmits this data over IP/Ethernet and provides a higher level of diagnostic data throughput than CAN. At the same time, the UNECE R155 and ISO/SAE 21434 standards require ensuring an appropriate level of cybersecurity of the vehicle throughout its entire life cycle. The R156 standard defines the requirements for the software update management system, which determines how data is moved to and from vehicles. Together, these standards create both opportunities and limitations for remote diagnostics: they provide high-speed authenticated access when needed, but at the same time require resource-efficient and controlled data movement. This article discusses the benefits of moving analytical computations closer to the data source – the telematics control unit (TCU) or other network gateway with data caching at the edge computing node. This reduces latency and dependency on continuous connectivity, improves throughput, while still adhering to the principles of diagnostic and security standards.

Related research in the field of networks confirms the effectiveness of this direction. The ETSI standard for multi-peripheral edge computing (MEC) envisages the placement of computing power and data storage in or near the radio access network (RAN), providing standardized access protocols and guidelines for the deployment of operator-hosted edge computing services. For the operator, MEC diagnostics reduces the latency of data transmission in both directions and provides a near cache that can process and store bursts of packets and forward them later.

2. EXPERIMENTAL METHODS

Modern vehicles provide the ability to monitor from hundreds to thousands of different metrics that are useful for vehicle diagnostics. Some metrics have a high update rate, such as engine speed, acceleration, battery current, others have a low update rate. Telemetry that is intended to be sent to a remote server is filtered, processed, and sent by a TCU or other network gateway.

Three factors dominate the system design. First, cloud connectivity is inconsistent and often expensive, especially for fleets spanning regions with different operators and roaming agreements. Second, not all metrics are equally important for making diagnostic decisions at any given time. Third, edge computing resources are limited, but with the development of hardware, they are becoming increasingly powerful, making it possible to run non-trivial algorithms.

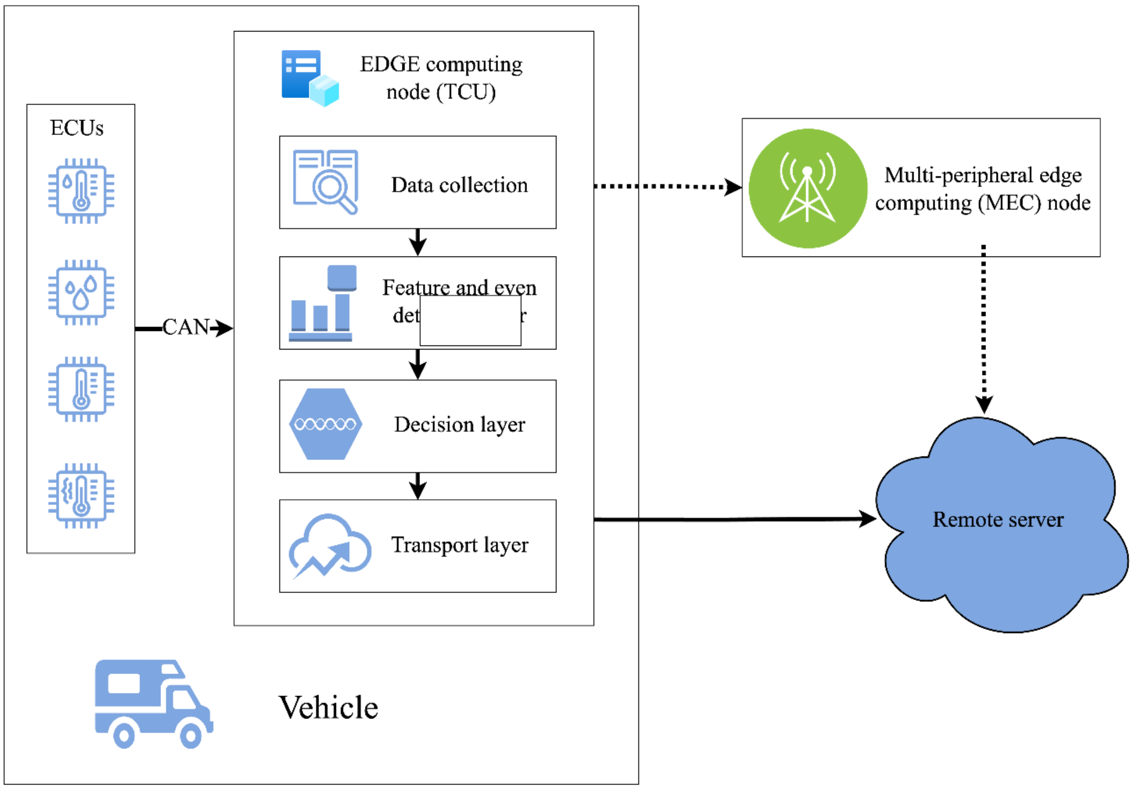

Figure 1. Proposed architecture for optimizing the transmission of collected signals using edge computing

In a remote diagnostics system (Fig. 1), which is focused on processing data with edge computing resources, the computing node is located inside the vehicle, usually integrated with the TCU, and has several signal processing layers. The signal acquisition layer receives traffic from the common bus and normalizes the data. The feature and event layer provides adaptive sampling, real-time change detection, and feature computation. The decision-making layer performs lightweight diagnostic logic – rules, statistical tests, or small trained models. The transport layer schedules data transmission over available communication channels, taking into account set of priorities, timing deadlines for individual features, and available size budgets. Additionally, MEC can be used – these are computing resources inside the mobile operator’s network, usually near a base station; they are not a car or a public cloud. MEC can be the first to receive urgent notifications, quickly check them without saving state, briefly buffer them, and forward them to the cloud, which reduces latency and latency fluctuations. This architecture follows the design rule: do as little as possible in the initial stage to ensure correct and timely decisions in the next stage, and reduce the amount of data by moving semantics to the edge nodes of computation.

A remote diagnostics system should provide reliable data necessary for making accurate decisions under conditions of limited bandwidth and unstable Internet connection, while maintaining low data transmission latency. The total amount of raw data received over a certain period of time t is calculated by formula (1):

$B_0 = \sum_{c=0}^{n} N_c \cdot s_c$, bytes, where (1)

n – total number of channels,

N_c – number of data instances in the current channel over a period of time t,

s_c – number of bytes in a single data instance in the current channel.

To optimize the transmission system, we introduce the concept of an edge computing node, which processes raw ECU data on the fly and forms packets from them for transmission. This reduces data resolution and compresses it. Thus, the transmitted payload becomes more efficient and is calculated by the formula:

$B = \sum_{c=0}^{n} (\kappa_c(\varepsilon) \cdot \alpha_c \cdot N_c \cdot s_c + \beta_c)$, bytes, where (2)

n – total number of channels,

ε – precision regulator for compression,

$\kappa_c(\varepsilon) \in (0,1]$ – channel compression ratio,

$\alpha_c \in (0,1]$ – data resolution factor,

s_c – number of bytes in a single data instance in the current channel,

N_c – number of data instances in the current channel over a period of time t,

β_c – bytes, channel metadata over time period t (for example – minimum, maximum, average value, etc.).

So, the three quantities that are important from an operational point of view are: data resolution factor (α_c), channel compression ratio $\kappa_c(\varepsilon)$￼) and channel metadata over time periodβ_c￼). The goal is to choose threshold values such that$B$ ￼ is an order of magnitude lower thanB_0￼, while preserving important information for diagnostic decision-making and analysis (limit crossings, sharp changes, data frequency changes, etc.) and a performance that depends on on-board information processing, not on the state of the Internet network.

3. RESULTS AND DISCUSSION

Let's consider an example of optimization over the period $t = 1h = 3600s$ with 3 signals – RPM (engine revolutions per minute), $T_{coolant}$ (coolant temperature), $V_{battery}$ (battery voltage level). Number of data instances in the channel N_c for the time period t we calculate using formula (3):

$N_c = f_c \cdot t$, where (3)

f_c – Hz, the data frequency of a particular channel.

An example of data optimization using edge computing is given in Table 1. As can be seen from the calculations, the total amount of processed data is reduced by ~30 times, which significantly reduces the load on the network.

Any process of lossy optimization of data and it’s transmission carries the risk of inconsistencies between the original and compressed data. Therefore, it is important to be able to adequately evaluate this process:

1. Accuracy of making diagnostic decisions based on the received data (invariance of discrete decisions, such as «has the coolant temperature exceeded a certain limit?»). For example, the system has a rule «Notify the user if the coolant temperature exceeds 110°C». With edge computing and data compression, the result of notification or no notification

should be the same as if the remote server had received the original raw data from the ECU. To check, you can: compare the % of coincidence of the system's decisions on the original and processed data, estimate the level of false positives and false negatives results.

2. Accuracy of representation – how close the reconstructed numbers are to the original. For example: after the packaging and unpacking processes, each recorded coolant temperature value should be within ±0.5°C of the true value. The deviation of the value is called the error. To estimate the error, it is necessary to compare it with the specified maximum error value for a given signal.

3. Timeliness of the received data – how quickly the system is able to detect and respond to errors. For example: the coolant temperature actually exceeds 110°C at time t=100s. If the system is able to detect the temperature limit being exceeded at time t=101s, then the detection time or delay is 1s. To check, you need to estimate the total time from the moment the event occurs on the ECU to the moment the notification appears to the user, including delays in the ECU, network, edge computing and any other system components.

Table 1

Example of data optimization by edge computing.

	RPM	$Tcoolant$	$V_{battery}$
f_c, Гц	10	1	5
$\kappa_c(\varepsilon)$	0.3	0.2	0.25
α_c	0.05	0.02	0.1
s_c, byte	2	2	2
β_c, byte	600	600	600
B_0, byte	72000	7200	36000
B, byte	1680	~629	1500
B_{0total}, byte	115200		
B_{total}, byte	~3808		

Optimization using edge computing is a series of trade-offs. The main idea of optimization is to make B as small as possible, but not at the cost of accuracy and timeliness of diagnostics. Each parameter of the equation can reduce the number of bytes in the final packet, at the risk of losing detail:

- A smaller compression ratio ($\kappa_c(\varepsilon)$) increases the level of data compression, reducing B (Fig. 2). However, a value that is too small blurs the signals and can hide the error.
- A smaller resolution of α_c saves bytes, but slows down the speed of error detection.
- Metadata and statistics β_c allow you to get a better summary of channels over a certain period of time, but increase the total number of bytes. You need to determine which parameters are really necessary and send only those.

A practical design pattern is to start with conservative error limits and thresholds that ensure that the transmitted data is consistent with the known data. Then gradually relax the parameters to achieve parity between throughput and accuracy. Using real-time telemetry to ensure that certain edge cases do not distort diagnostics. Often, it is possible to identify such channels that can be aggressively tweaked without affecting any diagnostic decisions, since their variations are either predictable or irrelevant.

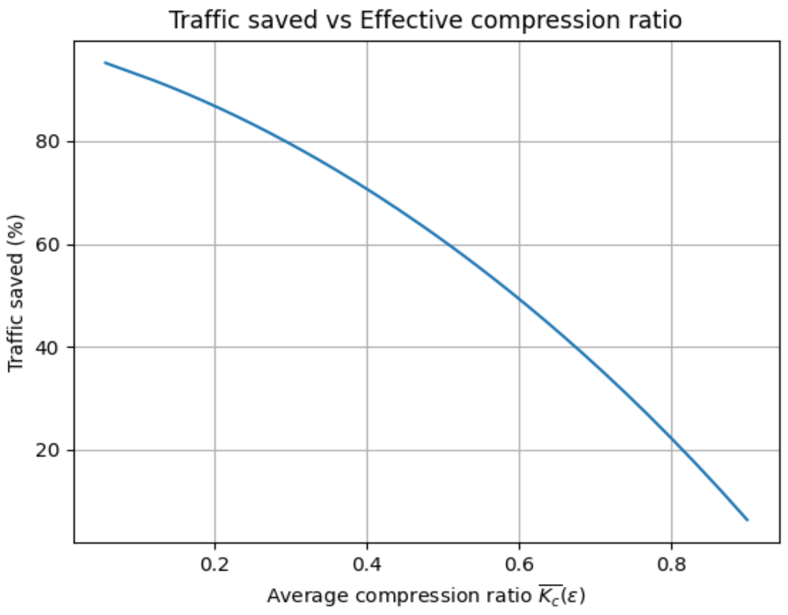

Figure 2. Saving network traffic using edge computing

Edge computing systems do not eliminate the need for raw data entirely. Some fault diagnostics require raw data streams from high-speed channels that are not available from metadata, statistical summaries, and fragmented data. Therefore, the edge computing node must be able to switch to full reporting mode when the vehicle temporarily transmits full raw data under certain conditions or remote commands. Another limitation is the validation of the learned components; while tiny models can be made reliable, proving their behavior under all operating conditions remains a challenge. Ultimately, the computing power and power consumption of vehicles, while constantly evolving, still limits what can be done without affecting other functions. Optimization of algorithms is necessary to avoid negative impacts on the vehicle itself or its individual components.

Telemetry analysis often includes sensitive information such as location etc. Systems with edge computing nodes improve privacy by default because they transmit less raw data and more aggregated summaries. Trusted edge computing runtimes protect various model parameters and keys. Mutual authentication and certificate pinning protect the transport layer. Payload signing ensures end-to-end data integrity so that compressed or aggregated data cannot be tampered with. Edge computing logic update mechanisms should be atomic, roll-backable, and certified. Diagnostics should not be left in inconsistent states during over-the-air updates. Security depends on provability and safe behavior. If the model fails to initialize or the feature extractor detects internal errors, the algorithm should fall back to conservative sampling rather than silently disabling telemetry. Error signals should be stored until acknowledged by the server. The system should maintain a local audit log so that post-incident analysis can reconstruct the behavior of the edge-scaling algorithm.

4. CONCLUSIONS

Remote diagnostics can only be successful if the right information is delivered quickly, reliably, and consistently. Edge computing provides the ability to shape, aggregate, and partially interpret telemetry where it is generated, so that the network delivers diagnostic decisions and compact data slices, rather than an uncontrolled stream of raw data. Based on clear design criteria such as decision accuracy, representation, and timeliness, significant reductions in network load can be achieved without compromising the diagnostic process.

References

1. Moshynska, Alina & Khrokalo, Oleksandr. (2024). Remote vehicle diagnostic system development based on the internet of things technology. Information and Telecommunication Sciences. 28–32. https://doi.org/10.20535/2411-2976.12024.28-32
2. Zheng, Yang & Li, Feifei & Luo, Feng. (2012). Vehicle Remote Diagnostic System Implementation Based on 3G Communication and Browser/Server Structure.
3. S. Douch, M. R. Abid, K. Zine-Dine, D. Bouzidi and D. Benhaddou (2022) "Edge Computing Technology Enablers: A Systematic Lecture Study," in IEEE Access, vol. 10, pp. 69264–69302. https://doi.org/10.1109/ACCESS.2022.3183634
4. George A. Shaji & George A.s & Baskar Dr & s, A. (2023). Edge Computing and the Future of Cloud Computing: A Survey of Industry Perspectives and Predictions. 02. 19–44. Doi: 10.5281/zenodo.8020101.
5. Cao K., Liu Y., Meng G. and Sun Q. (2020) "An Overview on Edge Computing Research", in IEEE Access, vol. 8, pp. 85714–85728. https://doi.org/10.1109/ACCESS.2020.2991734
6. Ajin V. W., Kumar L. D. and Joy J. (2016) "Study of security and effectiveness of DoIP in vehicle networks", International Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, India, 2016, pp. 1–6. https://doi.org/10.1109/ICCPCT.2016.7530357
7. Xu, Ning & Luo, Feng. (2025). Automotive DoIP Cybersecurity analysis. Advances in Engineering Innovation. 16. None-None. https://doi.org/10.54254/2977-3903/2025.21619
8. Kharche P., Murali M. and Khot G., "UDS implementation for ECU I/O testing", (2018) 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE), Singapore, 2018, pp. 137–140. https://doi.org/10.1109/ICITE.2018.8492642
9. Mishko O., Matiuk D., Derkach M. (2024) Security of remote iot system management by integrating firewall configuration into tunneled traffic. Scientific Journal of TNTU, vol. 115, no. 3, pp. 122–129. https://doi.org/10.33108/visnyk_tntu2024.03.122
10. Patra, Bhupesh & Tamrakar, Abha & Sharma, Rishabh. (2019). EDGE COMPUTING: EVOLUTION, CHALLENGES, AND FUTURE DIRECTIONS. Turkish Journal of Computer and Mathematics Education (TURCOMAT). 10. 741–745. https://doi.org/10.61841/turcomat.v10i1.14603
11. Xie, Lingfeng & Luo, Feng. (2016). Research and Implementation of the UDS Diagnostic System. 10.2991/icimm-16.2016.1. https://doi.org/10.2991/icimm-16.2016.1
12. Liashuk O., Hotovych V., Bonar V., Aulin V., Hrinkiv A. & Matiichuk L. (2024) The Concept of Remote Diagnostics of the Technical Condition of Vehicles During their Operation. Central Ukrainian Scientific Bulletin. Technical Sciences. 1. 29–39. https://doi.org/10.32515/2664-262X.2024.10(41).1.29-39
13. Kaiser Martin. (2015). Electronic control unit (ECU). https://doi.org/10.1007/978-3-658-03964-6_16
14. K. S. Dhananjayan, M. M, P. Kayalvizhi, P. Lakshmanan, P. N and O. S. Senthooriya (2024) "Development of a Telematic Control Unit for Capturing Vital Vehicle Data Without Using Company Fitted Telematic Ports", International Conference on Science Technology Engineering and Management (ICSTEM), Coimbatore, India, 2024, pp. 1–5. https://doi.org/10.1109/ICSTEM61137.2024.10561194
15. Nencioni, Gianfranco & Garroppo, Rosario & Olimid, Ruxandra. (2023). 5G Multi-Access Edge Computing: A Survey on Security, Dependability, and Performance. IEEE Access, pp. 1–1. https://doi.org/10.1109/ACCESS.2023.3288334
16. Starchenko V. (2021) Traffic optimization in wifi networks for the internet of things. Scientific Journal of TNTU (Tern.), vol. 104, no. 4, pp. 131–142. https://doi.org/10.33108/visnyk_tntu2021.04.131
17. Voichyshyn Yu., Holenko K., Horbay O., Honchar V. (2023) Methodology of analytical research of the microclimate of the bus drivers cab using the ANSYS-FLUENT software environment. Scientific Journal of TNTU, vol. 109, no. 1, pp. 90–98. https://doi.org/10.33108/visnyk_tntu2023.01.090

УДК 004.75

КОРДОННІ (EDGE) ОБЧИСЛЕННЯ ТА ОПТИМІЗАЦІЯ ДАНИХ ДЛЯ ДИСТАНЦІЙНОЇ ДІАГНОСТИКИ ТРАНСПОРТНИХ ЗАСОБІВ

Віталій Бонар; Володимир Готович; Любомир Матійчук

Тернопільський національний технічний університет імені Івана Пулюя, Тернопіль, Україна

***Резюме**. Запропоновано підхід до віддаленої діагностики транспортних засобів і виконанням первинної аналітика за допомогою кордонних обчислення (TCU/мережевий шлюз) та опційною* *оптимізацією за допомогою мульти-периферійних кордонних обчислень MEC. Отримано формалізацію компромісу «точність – своєчасність – обсяг», у якій першопочаткова кількість даних описується як $B_0=\sum_{c=0}^{n} N_c \cdot s_c$, а оптимізована – як $B=\sum_{c=0}^{n} (\kappa_c(\varepsilon)\cdot\alpha_c\cdot N_c\cdot s_c+\beta_c)$, де α_c є коефіцієнтом розширення даних, $\kappa_c(\varepsilon)$ – коефіцієнтом стиснення даних, B_c – обсягом метаданих. Показано на експерименті з трьома каналами (RPM, $T_{coolant}$, $V_{battery}$), що за налаштувань $\alpha_c=\{0.05; 0.02; 0.10\}$ і $k_c(\varepsilon)=\{0.30; 0.20; 0.25\}$ обсяг оптимізованих даних зменшується у ~30 разів ($B_{0total}=115200$ байт/год, $B_{total}=3808$ байт/год) без втрати здатності фіксувати значущі для діагностики події, а затримка спрацювання тригерів сповіщення становить 1с. Показано, що коректність діагностики забезпечується завдяки поєднання трьох чинників: точність – діагностичне рішення прийняте на основі сирих даних є таким самим, як і рішення, прийняте на основі оптимізованих даних; похибка представлення оптимізованих даних є найменшою; різниця часу прийняття діагностичного рішення на основі оптимізованих та сирих даних наближається до 0. Отримано методику налаштування параметрів оптимізації, в якій початкові параметри обираються з консервативних міркувань таким чином, щоб похибка діагностичних рішень була мінімальною для діагностики з використанням сирих і оптимізованих даних. Далі для знаходження оптимальних значень параметрів оптимізації, коефіцієнти α_c, $k_c(\varepsilon)$ та вміст метаданих B_c поступово зменшують до досягнення бажаного результату значень точності діагностики в порівнянні з першопочатковими даними та пропускної здантності мережі. Звідси можна зробити висновок, що раціональний вибір параметрів α_c, $k_c(\varepsilon)$, β_c разом із застосуванням MEC дозволяє суттєво скоротити кількість переданих даних без компромісів щодо інваріантності діагностичних рішень і своєчасності діагностичних сповіщень та слугує основою для проєктування систем діагностики із застосуванням кордонних обчислень.*

Ключові слова: *віддалена діагностика, кордонні обчислення, оптимізація телеметрії, локалізована обробка, транспортні засоби.*