https://doi.org/10.31891/csit-2024-4-6
UDC 004.9

Kvitoslava OBELOVSKA, Artur HRYTSIV,
Oleh LISKEVYCH, Andriy ABZYATOV,
Lviv Polytechnic National University
Rostyslav LISKEVYCH
Interregional Academy of Personnel Management, Kyiv, Ukraine
Ukrainian Industrial Telecommunications LLC, Lviv, Ukraine

A MODEL OF AN ENHANCED COMPUTER GAME SERVER IN MULTIPLAYER ENVIRONMENTS

The rapid evolution of multiplayer gaming has led to increasingly complex virtual environments that require precise synchronous movement mechanics to be competitive. One of the main challenges of interacting with large numbers of users in real-time in a multiplayer environment is the effect of network delays on character movements. In addition to the constant component, network delays have a variable component that is random and at the same time can be different on different network segments when the server interacts with different clients. The article examines the operation of a computer game server and proposes a model of advanced character movement control for multiplayer environments that provides smooth transitions between animation states through the concept of client-side prediction. The model is based on the state transition diagram of the server and describes its operation during a multiplayer game. To analyze the processes implemented by the server, we defined its five states: listening state, packet delay check state, mobility check state, client data update state, and preauthorization data update state. The object of modeling is a random process characterized by discrete states and continuous time, the model of which is presented as a system of differential equations. The results of solving this system of equations are analytical expressions for estimating the probabilities of a computer game server being in each of the possible states depending on the intensity of transitions between states. The presented mathematical apparatus describes the influence of incoming requests of different intensities on maintaining the necessary quality of system operation. The resulting formulas can be used for further analysis of the server's operation in various scenarios. Based on them, recommendations for improving data exchange algorithms in the system can be developed.

Keywords: multiplayer game development, multiplayer server management, character movement, telecommunication network, model.

Квітослава ОБЕЛЬОВСЬКА, Артур ГРИЦІВ,
Олег ЛІСКЕВИЧ, Андрій АБЗЯТОВ,
Національний університет «Львівська політехніка»
Ростислав ЛІСКЕВИЧ
Міжрегіональна академія управління персоналом, Київ, Україна
ТОВ «Українські промислові телекомунікації», Львів, Україна

МОДЕЛЬ КОМП'ЮТЕРНОГОІГРОВОГО СЕРВЕРА В БАГАТОКОРИСТУВАЦЬКИХ СЕРЕДОВИЩАХ

Швидка еволюція багатокористувацьких ігор призвела до дедалі складніших віртуальних середовищ, які потребують точної синхронної механіки руху, щоб бути конкурентоспроможними. Однією з основних проблем взаємодії з великою кількістю користувачів у режимі реального часу в багатокористувацькому середовищі є вплив мережевих затримок на рухи персонажів. На додаток до постійної складової мережеві затримки мають змінну складову, яка є випадковою і може бути різною на різних сегментах мережі, коли сервер взаємодіє з різними клієнтами. У статті досліджено роботу комп'ютерного ігрового сервера та запропоновано модель вдосконаленого управління рухом персонажа для багатокористувацьких середовищ, яка забезпечує плавний перехід між анімаційними станами завдяки використанню концепції клієнтського передбачення. Модель базується на діаграмі переходів між станами сервера та описує його роботу під час багатокористувацької гри. Щоб проаналізувати процеси, реалізовані сервером, ми визначили його п'ять станів: стан прослуховування, стан перевірки затримки пакетів, стан перевірки мобільності, стан оновлення даних клієнта та стан оновлення даних попередньої авторизації. Об'єктом моделювання є випадковий процес, що характеризується дискретними станами та безперервним часом, модель якого представлена у вигляді системи диференціальних рівнянь. Результатом розв'язання цієї системи рівнянь є аналітичні вирази для оцінки ймовірностей перебування сервера комп'ютерної гри в кожному з можливих станів залежно від інтенсивності переходів між станами. Представлений математичний апарат описує вплив вхідних запитів різної інтенсивності на підтримку необхідної якості роботи системи. Отримані формули можна використовувати для подальшого аналізу роботи сервера в різних сценаріях. На їх основі можуть бути розроблені рекомендації щодо вдосконалення алгоритмів обміну даними в системі.

Ключові слова: розробка багатокористувацьких ігор, керування багатокористувацьким сервером, рух персонажів, телекомунікаційна мережа, модель.

Introduction

The evolution of multiplayer gaming has introduced increasingly intricate virtual worlds that demand precise movement mechanics to maintain immersion and competitiveness. In modern multiplayer gaming environments, character movement is a critical aspect that significantly impacts player experience and gameplay quality. Delays, inconsistencies, or inaccuracies in character movement can disrupt gameplay, leading to frustration and loss of engagement among players [1]. To address this, modern movement control systems must integrate

advanced prediction algorithms, adaptive error correction mechanisms, and efficient data handling strategies [2]. The importance of client predictability in multiplayer games and a new prediction algorithm that can be used for this is shown in [3]. The article [4] presents the fundamental concepts of synchronization in multiplayer online games and discusses typical approaches and algorithms used to support the synchronization of distributed game objects.

The player actions in multiplayer games need to be translated into responsive and realistic motion even in the presence of network instability. In addition to the constant component, network delays have a variable component that is random and at the same time can be different on different network segments when the server interacts with different clients. Ensuring smooth and accurate control of character movement across networked systems poses unique challenges, especially when considering the demands of real-time synchronization, network latency, and computational resource optimization. These challenges necessitate the development of systems that can handle the complex interplay between client-side prediction, server-side correction, and network communication [1]. This, in turn, requires improving existing data governance algorithms to provide new opportunities. The synchronization problem in client-server gaming systems is considered in [4,5]. The paper [4] uses the probability of interaction among participants and ranking of the delay among participants. The paper [5] presents fundamental concepts of synchronization in multiplayer online games and discusses typical approaches and algorithms used to support the synchronization of distributed game objects.

This research focuses on the design and implementation of an enhanced character movement control model tailored for multiplayer environments. It leverages a modular approach to optimize real-time interaction between players, incorporating elements such as network prediction, interpolation, and server-client synchronization. The study aims to bridge the gap between theoretical advancements and practical application in character movement systems by employing state-of-the-art techniques and frameworks like Unreal Engine.

The main contribution of this paper can be summarized as follows:

1. The state transition diagram of a computer game server that can be used to analyze its operation is designed.
2. Based on the state transition diagram, the analytical model of the computer game server operation process is proposed.
3. Mathematical expressions for determining the probability of a computer game server being in each of its states due to the intensity of transitions between its states have been obtained.

A model of a computer game server

The character movement control system developed for multiplayer environments represents a sophisticated software solution designed to ensure precise synchronization and smooth gameplay. This system is characterized by its advanced integration of network prediction, collision detection, and adaptive response to varying network conditions. Its modular structure and implementation within Unreal Engine as a plugin make it highly versatile and suitable for a wide range of multiplayer projects.

A key innovation of this system is the Character Client Network Prediction (CCNP) mechanism. This approach allows the client to estimate and predict the character's next position based on the player's input, current velocity, and previously synchronized data from the server. The prediction algorithm compensates for network latency by preemptively calculating positions before receiving server updates. This ensures that character movements appear smooth and responsive, even in high-latency environments. In addition, CCNP utilizes real-time error correction. When the server detects discrepancies between the predicted position and the actual state, it sends corrective updates that are seamlessly integrated into the gameplay without noticeable jitter.

Key features of the system include real-time position prediction, error correction mechanisms, and efficient data synchronization between the server and multiple clients. These features address challenges such as latency, packet loss, and discrepancies between client- and server-side states, ensuring seamless interaction among players. The system is particularly effective in scenarios involving complex movement, such as climbing, swimming, or traversing dynamic environments, as it ensures all clients maintain a consistent game state.

This innovative design not only mitigates the effects of network variability but also optimizes performance in large-scale multiplayer games. For example, the system adapts movement behaviors dynamically, reducing the load on the server by delegating non-critical calculations to the client through prediction mechanisms. By minimizing redundant synchronization data and handling minor corrections locally, the system significantly reduces the perceived latency and enhances the overall user experience.

As an object of further research, this system provides a foundation for exploring new algorithms and methods to enhance character movement control. The primary focus is on improving accuracy, reducing the impact of latency, and optimizing resource consumption in diverse multiplayer scenarios. To support such analysis, we propose a detailed model of the primary component's operation, encompassing its interaction with network layers, player inputs, and server-side logic. This model includes a feedback loop for continuous evaluation and adjustment of prediction and correction strategies based on real-world latency patterns.

The task involves developing a comprehensive framework for evaluating the system's efficiency, exploring potential optimizations, and validating its scalability in diverse multiplayer scenarios. This will contribute to the advancement of character movement control technologies, addressing the growing demands of modern game

development. By integrating and refining CCNP, the system not only meets current requirements but also sets the stage for future multiplayer advancements in terms of both technical efficiency and player experience.

A model of a computer game server

To study the server operation in more detail, we developed a model that describes its functioning during a multiplayer game. When studying different processes, different types of models are used, such as a model supported by artificial intelligence [6], analytical [7,8], ontology [9], a simulation model [10], a timed Petri net model [11], a game networking model [2,12], and others. The work [12] focuses on the challenges that arise when integrating network solutions into gaming environments and approaches to solving them. It is valuable that it is devoted to the choice of a network model for multiplayer games. The correct network model affects the gaming experience and performance, and its choice depends on the specifics of the game and the expectations of the players. We will use an analytical model similar to the one used in the article [13,14]. It considers the modeling object as a random process characterized by discrete states and continuous time, and represents its operation as a system of differential equations.

To analyze the processes occurring on the server, we have defined five of its states:

1. Listening state;
2. Packet delay check state;
3. Mobility check state;
4. Client data update state;
5. Pre-authorization data update state.

The server is in one of these states at any time and, when certain conditions are met, it transitions to another state. The server's operation can be described by a transition diagram between its states, which for the states we have selected has the form shown in Fig. 1.

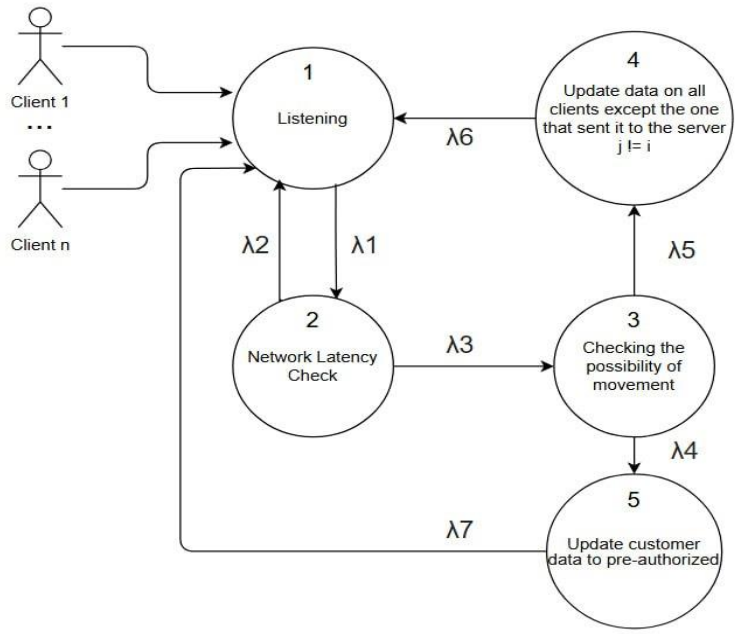

Fig. 1. Simplified diagram of server state transitions

The diagram demonstrates the process of the server when interacting with clients, focusing on data latency control and client action management. Initially and by default during operation, the server is in a listening state (standby, idle?) (state 1). In this state, it is constantly listening and ready to respond to client connection requests or new data from already connected users. When a connection is initiated, the server goes into the connection quality evaluation state (state 2), checking whether the data transfer delays do not exceed the set thresholds. If the delay is too long, indicating network problems, the server returns to the listening state (state 1) in anticipation of a better client connection.

If the delay corresponds to the acceptable parameters, the server goes to the next procedure (state 3) - checks the possibility of performing actions by the client, in particular, moving a client's controlled character. If movement is allowed, the server begins the process of reproducing the client's actions, synchronizing them with other clients (state 4). To ensure the correct operation of the network, traffic data is transmitted to all clients, except for the one that sent this data, which avoids duplication of information.

After successfully processing the move, the server returns to standby mode (from state 4 to state 1), ready for new requests or actions from clients.

If problems are detected during the movement possibility check (state 3) or motion playback the server

updates the client data to pre-authorized data (state 5), operationally returns to inspect delays and, if possible, eliminate the causes to ensure smooth and stable network operation.

The sequence of processes described above allows to the avoidance of failures and ensures the correct operation of the program in real-time.

Let us denote by λ_i the transition intensity of the server from one state to another. Then, following Figure 1:

- λ1 – the intensity of the server's transition from the listening state to the delay checking state;
- λ2 – the intensity of the server's transition from the delay checking state to the listening state;
- λ3 – the intensity of the server's transition from the delay checking state to the movement possibility check state;
- λ4 – the intensity of the server's transition from the movement possibility check state to the pre-authorize data update state;
- λ5 – the intensity of the server's transition from the movement possibility check state to the client data update state;
- λ6 – the intensity of the server's transition from the client data update state to the listening state;
- λ7 – the intensity of the transition of the server from the pre-authorize data update state to the listening state.

The server can be described as a random process characterized by discrete states and continuous time, so its mathematical model can be represented in the form of the following system of differential equations:

$$\begin{cases} \frac{dP_1(t)}{dt} = -\lambda_1 P_1(t) + \lambda_2 P_2(t) + \lambda_6 P_4(t) + \lambda_7 P_5(t) \\ \frac{dP_2(t)}{dt} = -(\lambda_2 + \lambda_3) P_2(t) + \lambda_1 P_1(t) \\ \frac{dP_3(t)}{dt} = -(\lambda_4 + \lambda_5) P_3(t) + \lambda_3 P_2(t) \\ \frac{dP_4(t)}{dt} = -\lambda_6 P_4(t) + \lambda_5 P_3(t) \\ \frac{dP_5(t)}{dt} = -\lambda_7 P_5(t) + \lambda_4 P_3(t) \end{cases} \quad (1)$$

To study the operation of the server in stationary mode, the system of equations (1) can be rewritten in the form of a system of algebraic equations:

$$\begin{cases} 0 = -\lambda_1 p_1 + \lambda_2 p_2 + \lambda_6 p_4 + \lambda_7 p_5 \\ 0 = -(\lambda_2 + \lambda_3) p_2 + \lambda_1 p_1 \\ 0 = -(\lambda_4 + \lambda_5) p_3 + \lambda_3 p_2 \\ 0 = -\lambda_6 p_4 + \lambda_5 p_3 \\ 0 = -\lambda_7 p_5 + \lambda_4 p_3 \end{cases} \quad (2)$$

The total probability that the system is in any of the discrete states is equal to 1, which gives the normalization identity:

$$\sum_{k=1}^{5} p_k = 1 \quad (3)$$

To solve the system, we will use the method of substitutions and represent all probabilities through the probability p_3. From the fifth equation of system (2), the probability p5 of the server being in the fifth state can be written as:

$$p_5 = \frac{\lambda_4}{\lambda_7} p_3. \quad (4)$$

The probability of the server being in the fourth state p_4 from the fourth equation of system (2) can be written as:

$$p_4 = \frac{\lambda_5}{\lambda_6} p_3, \quad (5)$$

and from the third equation of system (2), we get the probability p_2 of the server being in the second state:а з трертього рівняння системи (2) отримаємо ймовірність p_2 перебування сервера у другому стані:

$$p_2 = \frac{\lambda_4+\lambda_5}{\lambda_3} p_3. \quad (6)$$

From the second equation of system (2) taking into account equation (6), we get the probability of the server being in the first state p_1:

$$p_1 = \frac{(\lambda_2+\lambda_3)(\lambda_4+\lambda_5)}{\lambda_1 \lambda_3} p_3. \quad (7)$$

Substitute (4) - (7) in (3):

$$\frac{(\lambda_2+\lambda_3)(\lambda_4+\lambda_5)}{\lambda_1\lambda_3} p_3 + \frac{\lambda_4+\lambda_5}{\lambda_3} p_3 + p_3 + \frac{\lambda_5}{\lambda_6} p_3 + \frac{\lambda_4}{\lambda_7} p_3 = 1. \quad (8)$$

Let's find the probability p_3 of the server being in the third state due to the intensity of transitions between states.

$$p_3 = 1/\left[1 + \frac{\lambda_5}{\lambda_6} + \frac{\lambda_4}{\lambda_7} + \frac{(\lambda_4+\lambda_5)(\lambda_1+\lambda_2+\lambda_3)}{\lambda_1\lambda_3}\right]. \quad (9)$$

For simplicity, we mark:

$$\mathrm{B} = \left[1 + \frac{\lambda_5}{\lambda_6} + \frac{\lambda_4}{\lambda_7} + \frac{(\lambda_4+\lambda_5)(\lambda_1+\lambda_2+\lambda_3)}{\lambda_1\lambda_3}\right]^{-1} \quad (10)$$

and the probability of the server being in the movement possibility check state p_3:

$$p_3 = B. \quad (11)$$

Using this notation we will find the probabilities of the server being in all its states. As a result, we get:

- the probability that the server is in a listening state:

$$p_1 = \frac{(\lambda_2+\lambda_3)(\lambda_4+\lambda_5)}{\lambda_1\lambda_3} B; \quad (12)$$

- the probability that the server is in a latency check state:

$$p_2 = \frac{\lambda_4+\lambda_5}{\lambda_3} B; \quad (13)$$

- the probability of the server being in the client data update state:

$$p_4 = \frac{\lambda_5}{\lambda_6} B; \quad (14)$$

- the probability of the server being in the pre-authorized data update state:

$$p_5 = \frac{\lambda_4}{\lambda_7} B. \quad (15)$$

To verify the model, we will take the first equation of system (2), which was not used when finding the probabilities of the server (11-15) being in its various states. Let's define the right side of this equation:

$$\begin{aligned} -\lambda_1 p_1 + \lambda_2 p_2 + \lambda_6 p_4 + \lambda_7 p_5 &= -\lambda_1 \frac{(\lambda_2 + \lambda_3)(\lambda_4 + \lambda_5)}{\lambda_1\lambda_3} B + (\lambda_2 \frac{\lambda_4 + \lambda_5}{\lambda_3} + \lambda_6 \frac{\lambda_5}{\lambda_6} + \lambda_7 \frac{\lambda_4}{\lambda_7}) B = \\ &= -\frac{(\lambda_2 + \lambda_3)(\lambda_4 + \lambda_5)}{\lambda_3} B + (\lambda_2 \frac{\lambda_4 + \lambda_5}{\lambda_3} + \lambda_5 + \lambda_4) B = \\ &= -\frac{(\lambda_2+\lambda_3)(\lambda_4+\lambda_5)}{\lambda_3} B + \frac{(\lambda_2+\lambda_3)(\lambda_4+\lambda_5)}{\lambda_3} B = 0. \end{aligned} \quad (16)$$

As expected, the right-hand side of the tested equation is equal to zero, confirming the proposed model's correctness. So, the obtained analytical expressions (11)-(15) make it possible to estimate the probability of the computer game server being in each possible state depending on the intensities of transitions between states.

Conclusions

Computer game server is considered a stochastic system, the operation of which depends on many random factors, both internal and external. An example of external factors is delays in the delivery of packets caused by the data transmission network. Exceeding the delay above the permissible norms can lead even to the inoperability of the application. To analyze the operation of the server, a representation of its functioning in the form of a state transitions diagram is proposed. Due to this, the operation of the server is described by a system of differential equations, as a result of which analytical expressions for the probabilities of the server being in all its states have been obtained. The resulting formulas can be used for further analysis of the server's operation in various scenarios. Based on them, recommendations for improving data exchange algorithms in the system can be developed.

Acknowledgment. The research was carried out within the framework of the Erasmus+ Jean Monnet Module «European Data Strategy: Data Governance for New Opportunities», 101127839 — Data4EU — ERASMUS-JMO-2023-HEI-TCH-RSCH.

References

1. Multiplayer Networking Challenges**. (**2022). **https://www.argentics.io/multiplayer-networking-challenges**
2. Kevin Glass. (2023). Modern Game Networking Models. https://developers.rune.ai/blog/modern-game-networking-models
3. Client Side Prediction. https://mirror-networking.gitbook.io/docs/manual/general/client-side-prediction
4. Paik, Doowon & Yun, Chung-Ha & Hwang, Jooyeon. (2008). Effective message synchronization methods for multiplayer online games with maps. Computers in Human Behavior. 24. 2477-2485. 10.1016/j.chb.2008.03.004.
5. Ferretti, Stefano. (2014). Synchronization in Multiplayer Online Games. 10.1002/9781118796443.ch7.
6. Smerdov A, Somov A, Burnaev E, Stepanov A. AI-enabled prediction of video game player performance using the data from heterogeneous sensors. Multimed Tools Appl. 2023;82(7):11021-11046. doi: 10.1007/s11042-022-13464-0. Epub 2022 Aug 23. PMID: 36035326; PMCID: PMC9395877.
7. Kovtun V, Izonin I, Gregus M. (2022) Modeling a session of subject-system interaction in a wireless communication infrastructure with a mixed resource. PLoS ONE 17(7): e0271536. https://doi.org/10.1371/journal.pone.0271536
8. Viacheslav Kovtun, Ivan Izonin, Michal Gregus. Model of functioning of the centralized wireless information ecosystem focused on multimedia streaming // Egyptian Informatics Journal. – 2022. – (in press). https://doi.org/10.1016/j.eij.2022.06.009
9. LUTSYK, I., & FEDASYUK, D. (2024). ANALYSIS OF APPROACHES TO DESIGN ONTOLOGICAL MODELS OF AN ADAPTIVE SOFTWARE SYSTEM. *Computer Systems and Information Technologies*, (3), 13–20. https://doi.org/10.31891/csit-2024-3-2
10. Franciny M. Barreto and Stéphane Julia. 2020. A Possibilistic Simulation Model for Multiplayer Game Scenarios Using CPN Tools. In Proceedings of the XXXIV Brazilian Symposium on Software Engineering (SBES '20). Association for Computing Machinery, New York, NY, USA, 114–119. https://doi.org/10.1145/3422392.3422472
11. Barreto, F. M., de Freitas, J. C. J., & Julia, S. (2018). A timed Petri net model to specify scenarios of video games. In *Information Technology-New Generations: 15th International Conference on Information Technology* (pp. 467-473). Springer International Publishing.
12. FIELDER, G. (2024). Choosing the right network model for your multiplayer game. https://mas-bandwidth.com/choosing-the-right-network-model-for-your-multiplayer-game/
13. OBELOVSKA, K., PELEKH, K., PELEKH, Y., SNAICHUK, Y. (2022). ANALYSIS OF THE CSMA/CA SCHEME FOR WIRELESS LOCAL AREA NETWORKS. *Herald of Khmelnytskyi National University Technical Sciences*, 1(6), 148-152 http://journals.khnu.km.ua/vestnik/?cat=74
14. Auzinger, W., Obelovska, K., Dronyuk, I., Pelekh, K., Stolyarchuk, R.A Continuous Model for States in CSMA/CA-Based Wireless Local Networks Derived from State Transition Diagrams. In: Saraswat, M., Roy, S., Chowdhury, C., Gandomi, A.H. (eds) Proceedings of International Conference on Data Science and Applications. Lecture Notes in Networks and Systems, vol 287. Springer, Singapore. (2022).

Kvitoslava Obelovska **Квітослава Обельовська**	Ph.D., Associated Professor at the Department of Artificial Intelligence, Lviv Polytechnic National University https://orcid.org/0000-0002-8714-460X kvitoslava.m.obelovska@lpnu.ua	Кандидат технічних наук, доцент, доцент кафедри автоматизованих систем управління Національного університету «Львівська політехніка»
Artur Hrytsiv **Артур Гриців**	Student of master degree at the Computer Science Department, Lviv Polytechnic National University artur.hrytsiv.mknus.2023@lpnu.ua	Магістрант кафедри автоматизованих систем управління Національного університету «Львівська політехніка»
Oleh Liskevych **Олег Ліскевич**	Ph.D., senior lecturer at the Automated Control Systems Department, Lviv Polytechnic National University https://orcid.org/0009-0007-0624-6828 oliskevych@gmail.com	Кандидат технічних наук, старший викладач кафедри автоматизованих систем управління Національного університету «Львівська політехніка»
Andriy Abzyatov **Андрій Абзятов**	Postgraduate student at the Automated Control Systems Department, Lviv Polytechnic National University https://orcid.org/0007-0375-5141 andrii.z.abziatov@lpnu.ua	Аспірант кафедри автоматизованих систем управління Національного університету «Львівська політехніка»
Rostyslav Liskevych **Ростислав Ліскевич**	Ph.D., doctoral student at Interregional Academy of Personnel Management, Kyiv, Ukraine https://orcid.org/0009-0001-1335-4436 rl@uptc.com.ua	Кандидат технічних наук, докторант Міжрегіональної академії управління персоналом, Київ, Україна.