

<https://doi.org/10.31891/csit-2026-2-12>

UDC 004.6.056.3:004.77

### Vitalii TKACHOV

Candidate of Technical Sciences, Doctoral  
Candidate of Department of Electronic  
Computers,  
Kharkiv National University of Radio Electronics  
<https://orcid.org/0000-0002-6524-9937>  
[vitalii.tkachov@nure.ua](mailto:vitalii.tkachov@nure.ua)

### Ihor RUBAN

Doctor of Technical Sciences, Professor,  
Professor of Department of Electronic  
Computers,  
Kharkiv National University of Radio Electronics  
<https://orcid.org/0000-0002-4738-3286>  
[ihor.ruban@nure.ua](mailto:ihor.ruban@nure.ua)

Received: 09/04/2026

Accepted: 06/05/2026

Published: 31/05/2026

© Copyright  
2026 by the author(s)



This is an Open Access article distributed  
under the terms of the [Creative Commons  
CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

## METHOD OF ACCELERATED DATA RECOVERY IN INFORMATION SYSTEM ON MOBILE PLATFORM IN POST-FAILURE MODE

*The article considers the problem of accelerated data recovery in an information system on a mobile platform (ISMP) in post-failure mode. The relevance of the study stems from the fact that, following an emergency data failure in an ISMP, there is a need to restore the critical part of the data to proper use as quickly as possible, given limited computational, network, and power resources. The aim of the work is to develop a method of accelerated data recovery, which after a data failure provides the selection of the smallest sufficient recovery action based on the control copy and the change log and reduces the duration of the return of the critical data portion to the correct use under current ISMP's resource constraints. The work formalizes post-failure data recovery through a set of data elements, indicators of a failure, a critical data portion, a set of control copies and a change log. A set of permissible recovery actions is constructed, within which each action is determined by the selection of a control copy, a fragment of the change log, the composition of immediate recovery data and the composition of data for deferred recovery. A rule for selecting the smallest sufficient recovery action and an algorithm for implementing the method have been developed, which takes into account the duration of recovery, the level of residual disruption of the data critical part, the energy resource consumption, and the number of elements for which immediate recovery is provided in the current action. According to the results of the computational study, it was established that the proposed method has an advantage over the basic recovery methods in terms of the recovery duration, the volume of data transmission and reading, the energy resource consumption, the level of residual disruption of the data critical part, and the proportion of successful completion of recovery within the acceptable time interval. The practical significance of the results obtained lies in the suitability of the method for use in onboard and distributed ISMPs, where after an emergency data failure it is necessary to restore the critical part of information base for operation within a short time interval without proceeding to the full recovery of the entire dataset.*

*Key words: information system on a mobile platform; post-failure mode; data recovery; control copy; survivability.*

### Introduction

There exists a class of information systems operating on the basis of mobile platforms [1]. In this case, mobile platforms are an edge computing environment characterized by variable availability of computing and network resources, mobility of nodes, intermittent data exchange, short decision-making intervals, and limited duration of autonomous operation due to energy constraints. Modern research on ISMP is devoted to issues of operation stability and reduction of decision-making time under conditions of limited resources [2–4].

For information systems on mobile platforms maintaining data integrity, completeness, timeliness, and consistency is essential, as data serve as the base for the execution of critical functions. In the event of an emergency failure of ISMP components, individual fragments of data may be lost, local records may be distorted, discrepancies may be accumulated between control copies and change logs, and incomplete cycles of processing and recording the results of information registration may occur. Under such conditions, ISMP loses the ability to continue working correctly even when part of the software modules and hardware remains operational [4–7].

Related research in this field focuses on data preservation and reconstruction in technical systems, where new values, operational records, and version changes are continuously

accumulated during their operation. In particular, adaptive data reconciliation modes, data replication taking into account temporal dependencies, localized recovery of individual parts of the computational graph, partial control copies, and the construction of a control copy during the recovery process are considered [8–11]. The obtained results confirm that recovery performance significantly depends on the volume of data that must be regenerated, transmitted, or recovered from the change log [9–11].

Existing solutions are mainly focused on distributed systems, streaming data processing and systems with a more stable resource mode of operation. For an ISMP, the post-failure mode has a different application complexity. In such a mode, full recovery of the entire data set often requires excessive time, a significant amount of service data exchange and additional load on computing nodes, which increases the energy costs of the mobile platform. Under such conditions, the applied significance lies in selecting the minimally sufficient recovery action which, using available control copies, change logs, and the available resources, ensures rapid return of the critical portion of data to correct usability [5], [7], [10]–[12].

Thus, there arises a scientific and applied task of developing a method for accelerated data recovery in ISMPs in post-failure mode. It aims to determine, after detecting a data integrity violation, which control copy is suitable as a recovery base, which fragment of the change log must be used, what amount of data is subject to urgent recovery, and which can be transferred to the next stage without disrupting the main function of an ISMP. The practical complexity of this task is associated with the need to reduce the duration of the post-failure period, reduce the load on platform resources, and limit data loss [2], [4]–[7], [10]–[12].

Accordingly, accelerated data recovery should be considered as a separate mechanism for increasing the survivability of an ISMP. The scientific complexity of this task is associated with the need to simultaneously take into account the nature of the accidental data violation, the composition of available records for recovery, the criticality of individual data groups, the processing and transmission time, as well as the limited energy resource of the mobile platform. This set of conditions justifies the feasibility of constructing a special method focused on rapid data recovery in the post-failure mode of ISMP operation.

#### **Analysis of recent research and publications**

The issue of control storage and recovery of cloud containers with dynamic data is considered in [13]. This work proposes an architectural solution for organizing control copies and recovering containerized applications. Its significance for this study is to confirm the suitability of the infrastructure-based approach to recovering a computing environment with changing data. At the same time, the subject of the study is related to containers as units of infrastructure recovery, while this article considers accelerated data recovery in an ISMP after a data integrity failure.

An approach to transferring microservices with changing data between edge infrastructure nodes is presented in [14]. In this work, a mechanism for migrating a service instance together with its associated data is formalized. For the problem posed in the article, this result creates a basis for estimating the time costs for recovery. However, [14] considers the transfer of a functioning service between nodes, while in this study the subject is the choice of an action for accelerated data recovery after a failure.

A related problem is discussed in [15], where the transfer of multi-container structures in multi-cluster edge systems is investigated. It is shown that as the structural complexity of the executed configuration increases, the role of organizing local data and the time required to bring it back into operation after migration also grows. For this study, this is important in terms of the dependence between the structure of the stored data and the time costs of restoring operability.

In [16], the location of a permanent storage in the computing space for robotic applications is analyzed. The significance of this work is that it confirms the impact of the storage location on the time characteristics of data access and on the resistance of the application system to failures. This provides grounds for considering data recovery in ISMP with account taken of the time required to access the control copy and to reintroduce the data into operational use.

For stream processing systems, a combination of control copies, backup executors, and logging is proposed in [17], taking into account the temporal and causal ordering of recovery. The applied significance of this result is that logging is considered as a means of reducing losses during recovery and returning to the consistent processing flow. For this study, it is the combination of a control copy with subsequent restoration of changes from the log that is important. At the same time, the results of [17] concern stream processing systems with a preconfigured backup execution structure. In the case of an ISMP after a failure, the selection of a part of the data for priority recovery and a fragment of the change log that provides the fastest return to the execution of a critical function becomes decisive.

A comparative evaluation of control saving protocols for streaming processing systems is given in [18]. This work investigates several classes of control saving protocols and shows the dependence of their effectiveness on the nature of the load and processing structure. The conclusions obtained confirm the absence of a universally best control saving mode for all situations. This directly leads to the need to choose a recovery action depending on the type of a failure and the available data for recovery.

Thus, in the works [13]–[18], the solution of individual parts of the general problem related to the organization of control copies, the transfer of services with changing data, taking into account the temporal properties of the storage, the combination of control copy with logging, and the comparison of control saving modes were

initiated. The combination of these results provides a scientific foundation for developing data recovery methods in complex distributed environments. At the same time, the problem of forming the smallest sufficient action for accelerated data recovery, which, based on the available control copies, change log and current resource of the mobile platform, ensures the shortest time for returning the critical part of the data to correct use, remains unsolved for ISMPs.

#### Formulation of article objectives

The purpose of this article is to develop a method for accelerated data recovery in an information system on a mobile platform in post-failure mode, which after a data failure provides the selection of the smallest sufficient recovery action based on the control copy and the change log and reduces the duration of returning the critical part of the data to correct use under the current resource limitations of the ISMP.

To achieve the goal, it is necessary to solve the following tasks:

- to formalize post-failure data disruption in the ISMP, submission of control copies, change log and critical part of the data, which will allow setting the initial conditions for constructing the accelerated recovery method;
- to construct a set of permissible recovery actions, each of which determines the reference control copy, the limit of use of the change log and the composition of the immediate recovery data, which will allow taking into account the current ISMP resource and the conditions of the post-failure mode;
- to develop a rule for selecting the minimal sufficient recovery action and an algorithm for its implementation, which will allow to ensure the shortest time to restore the critical part of data for correct use.

#### Formalization of post-failure data disruption in ISMP, control copies, change log and critical part of data

We consider an ISMP in which the data used during the execution of the main function is given by a finite set of elements:

$$A = \{a_i\}_{i=1}^N, \quad (1)$$

where  $a_i$  is the  $i$ th data element;  $N$  is the number of data elements used during the ISMP operation.

At a time  $t$ , each element  $a_i$  is described by a local vector:

$$\vec{y}_i(t) = (u_i(t), \theta_i(t), \rho_i(t)), \quad (2)$$

where  $u_i(t)$  is the value of the data element;  $\theta_i(t)$  is its version number;  $\rho_i(t)$  is the integrity indicator of the data element.

Formula (2) allows us to separate the value of a data element, its version number, and its integrity state. Then the complete current representation of the ISMP data at a time  $t$  is given by the vector:

$$\vec{Y}(t) = (\vec{y}_1(t), \vec{y}_2(t), \dots, \vec{y}_N(t)). \quad (3)$$

Formula (3) is used as a generalized representation of the current state of the ISMP data in the post-failure mode. After a data failure, a vector of failure indicators is introduced for each element  $a_i$ :

$$\vec{b}_i(t) = (\mu_i(t), \nu_i(t), \xi_i(t), \zeta_i(t)), \quad (4)$$

where  $\mu_i(t) \in \{0, 1\}$  is the indicator of value  $u_i(t)$  loss or unavailability;  $\nu_i(t) \in \{0, 1\}$  is an indicator of integrity failure;  $\xi_i(t) \in \{0, 1\}$  is an indicator of version consistency failure;  $\zeta_i(t) \in \{0, 1\}$  is an indicator of non-recovery of the current state of the data element according to the change log.

The presence of a post-failure disruption for an element  $a_i$  is determined by the indicator:

$$\gamma_i(t) = \begin{cases} 1, & \mu_i(t) + \nu_i(t) + \xi_i(t) + \zeta_i(t) > 0, \\ 0, & \mu_i(t) + \nu_i(t) + \xi_i(t) + \zeta_i(t) = 0. \end{cases} \quad (5)$$

Formula (5) specifies a post-failure disruption of a data element. If  $\gamma_i(t) = 1$ , then the element at a time  $t$  cannot be considered as suitable for use during the execution of the main function of the ISMP. To specify the critical part of the data, on which further execution of the ISMP's main function directly depends, an indicator of the belonging of the data element to the critical part is introduced:

$$\chi_i = \begin{cases} 1, & a_i \in K, \\ 0, & a_i \notin K, \end{cases} \quad (6)$$

where  $K \subseteq A$  is the critical part of the ISMP data.

The critical part of the data is defined by the set

$$K = \{a_i \in A \mid \chi_i = 1\}. \quad (7)$$

Formula (7) is further used as a basis for constructing an accelerated recovery method, since it is the return of elements of the set  $K$  to correct use that determines the possibility of continuing the execution of the ISMP's critical function. We specify the integral assessment of the failure level of the data critical part by the formula:

$$\Gamma(t) = \sum_{i=1}^N \chi_i \gamma_i(t). \quad (8)$$

The quantity  $\Gamma(t)$  (8) shows the number of failed elements in the critical part of the data at a time  $t$ . If  $\Gamma(t) = 0$ , then the critical part of the data does not contain post-failure disruption. If  $\Gamma(t) > 0$ , then the ISMP requires data recovery in the post-failure mode.

For data recovery in the ISMP, control copies are used. The set of available control copies is given as follows:

$$C = \{c_m\}_{m=1}^M, \quad (9)$$

where  $c_m$  is the  $m$ th control copy;  $M$  is the number of available control copies.

Each control copy  $c_m$  is provided by a tuple:

$$c_m = (\tau_m, A_m, Y_m), \quad (10)$$

where  $\tau_m$  is the moment of generating the control copy;  $A_m \subseteq A$  is the set of data elements covered by this copy;  $Y_m$  is the representation of the elements of the set  $A_m$ , recorded at the moment  $\tau_m$ .

The control copy  $c_m$  (10) is generally partial, since one copy can cover only a subset of  $A_m$ . This necessitates the selection of such a control copy that provides the best basis for accelerated recovery of the critical part of the data.

Changes that occur after the generation of the control copy are recorded in the change log. The set of log entries is:

$$J = \{j_n\}_{n=1}^L, \quad (11)$$

where  $j_n$  is the  $n$ th change log entry;  $L$  is the number of available entries.

Each change log entry is given by a tuple:

$$j_n = (\tau_n, B_n, o_n), \quad (12)$$

where  $\tau_n$  is the time of record fixation;  $B_n \subseteq A$  is the set of data elements affected by the change;  $o_n$  is the change operation that transfers the corresponding data elements to a new state.

For the control copy  $c_m$ , the change log corresponding to time  $t$  is given by a subset:

$$J_m(t) = \{j_n \in J \mid \tau_m < \tau_n < t\}. \quad (13)$$

The subset  $J_m(t)$  (13) contains all the change log records generated after the control copy  $c_m$  until time  $t$ .  $J_m(t)$  will be used to regenerate the changes after selecting the reference control copy.

For the accelerated recovery problem, the part of  $J_m(t)$ , which concerns the critical part of the data, is decisive. The corresponding fragment of the log is given as:

$$J_m^K(t) = \{j_n \in J_m(t) \mid B_n \cap K \neq \emptyset\}. \quad (14)$$

Formula (14) defines a fragment of the change log that directly affects the recovery of a critical part of the data. Using this fragment allows to limit the amount of recoveries to only those records that are necessary to return the ISMP to perform a critical function.

Since not every control copy is suitable for recovering a critical part of the data, a requirement is introduced for the completeness of the critical part of the data to be reproduced for the selected control copy and the corresponding fragment of the change log:

$$K \subseteq A_m \cup \bigcup_{j_n \in J_m^K(t)} B_n. \quad (15)$$

Formula (15) means that for a control copy  $c_m$  and a fragment of the change log  $J_m^K(t)$ , the entire critical part of the data  $K$  must be recovered from the elements present in the control copy and the change log records formed after it.

Thus, the set of admissible set of  $c_m$  and  $J_m^K(t)$  at a time  $t$  is defined as follows:

$$S(t) = \left\{ \left( c_m, J_m^K(t) \right) \mid K \subseteq A_m \cup \bigcup_{j_n \in J_m^K(t)} B_n \right\}. \quad (16)$$

### Construction of permissible recovery actions set

To construct a method for accelerated data recovery, it is advisable to proceed to a formal representation of the recovery action through selecting a control copy, a fragment of the change log, a set of immediate recovery elements and a set of deferred recovery elements. Such a transition is consistent with modern approaches to localized recovery and partial control storage [10], [17], [19], with approaches to reducing the overhead of logging [20], as well as with solutions in which a shared log is used to preserve the integrity and recoverability of data and ISMP [21], [22].

Let at time  $t$  for each log entry  $j_n \in J$  an indicator of its belonging to the current time interval be entered:

$$I_n(t) = \begin{cases} 1, & \tau_n \leq t, \\ 0, & \tau_n > t. \end{cases} \quad (17)$$

Formula (17) is used to separate those change log entries that can be involved in recovery at a time  $t$ . For each pair  $(c_m, j_n)$ , where  $c_m \in C$ ,  $j_n \in J$ , an indicator of the compatibility of the control copy and the change log entry is introduced:

$$\omega_{mn}(t) = \begin{cases} 1, & \tau_m < \tau_n < t, B_n \cap K \neq \emptyset, \\ 0, & \text{в іншому разі.} \end{cases} \quad (18)$$

Formula (18) means that a log record  $j_n$  can be used together with a control copy  $c_m$  only if it is generated after it and concerns a critical part of the data. To formally specify the belonging of data elements to control copies and log records, matrices are introduced:

$$\alpha_{im} = \begin{cases} 1, & a_i \in A_m, \\ 0, & a_i \notin A_m, \end{cases} \quad \beta_{in} = \begin{cases} 1, & a_i \in B_n, \\ 0, & a_i \notin B_n, \end{cases} \quad (19)$$

where  $\alpha_{im}$  indicates whether the element  $a_i$  is part of the control copy  $c_m$ , and  $\beta_{in}$  – whether the element  $a_i$  belongs to the set  $B_n$  to which the record  $j_n$  refers.

To maintain the correct order of re-application of change log records, it is advisable to take into account the dependence between records belonging to the same data element or to interconnected elements. For this, the following matrix is introduced:

$$\zeta_{nk}(t) = \begin{cases} 1, & \tau_n < \tau_k, B_n \cap B_k \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases} \quad (20)$$

Formula (20) specifies the order in which change log entries are re-applied in cases where selecting a later entry would require using a previous entry associated with the same piece of data.

The recovery action at a time  $t$  is given by four groups of variables:

$$x_m(t) \in \{0, 1\}, \quad y_n(t) \in \{0, 1\}, \quad h_i(t) \in \{0, 1\}, \quad d_i(t) \in \{0, 1\}, \quad (21)$$

where  $x_m(t) = 1$  means selecting a control copy  $c_m$  for the current recovery action;  $y_n(t) = 1$  denotes using a log entry  $j_n$  in the current recovery action;  $h_i(t) = 1$  means that an element  $a_i$  from the critical part of the data is subject to immediate recovery;  $d_i(t) = 1$  means that the recovery of the element  $a_i$  is postponed to the next stage.

Since one control copy is selected within a single recovery action, the condition must be met:

$$\sum_{m=1}^M x_m(t) = 1. \quad (22)$$

For each log entry  $j_n$ , the variable  $y_n(t)$  is subject to the compatibility condition with the selected control copy:

$$y_n(t) \leq \sum_{m=1}^M \omega_{mn}(t)x_m(t), \quad n = 1, \dots, L. \quad (23)$$

Formula (23) means that a log entry cannot be used outside the selected control copy or in the absence of the compatibility conditions specified by formula (18).

To maintain the correct order of using the change log, restrictions are introduced:

$$y_k(t) \leq y_n(t), \quad \forall n, k : \zeta_{nk} = 1. \quad (24)$$

Formula (24) specifies the sequential nature of the use of the change log for related records – selecting a later record implies selecting a previous record related to the same part of the data.

To divide the failed elements of the critical part of the data into immediate and deferred recovery elements, the equality is used:

$$h_i(t) + d_i(t) = \chi_i \gamma_i(t), \quad i = 1, \dots, N. \quad (25)$$

Formula (25) specifies a complete division of the failed elements of the set  $K$  into two groups: those included in the current recovery action and those whose recovery is postponed to the next stage.

However, immediate recovery of an element  $a_i$  is possible only if there is sufficient data for its recovery. Therefore, the following restriction is set for all  $i = 1, \dots, N$ :

$$h_i(t) \leq \sum_{m=1}^M \alpha_{im} x_m(t) + \sum_{n=1}^L \beta_{in} y_n(t). \quad (26)$$

Formula (26) denotes that a critical data element can be included in the immediate recovery only if it is covered by either the selected control copy or the change log records used in the current recovery action.

To evaluate the result of a recovery action by the data element version number, the recovered version of the element  $a_i$  is entered:

$$\hat{\theta}_i(t) = \max \left\{ \sum_{m=1}^M \alpha_{im} \theta_{im} x_m(t), \max_{1 \leq n \leq L} (\beta_{in} \vartheta_{in} y_n(t)) \right\}, \quad i = 1, \dots, N. \quad (27)$$

where  $\theta_{im}$  is the version number of the element  $a_i$  contained in the control copy  $c_m$ ;  $\vartheta_{in}$  is the version number of the element  $a_i$  corresponding to the log entry  $j_n$ .

Formula (27) specifies the version number obtained for the element  $a_i$  after the current recovery action is performed. For the same element, the largest recoverable version number at a time  $t$  is entered:

$$\theta_i^*(t) = \max \left\{ \max_{1 \leq m \leq M} (\alpha_{im} \theta_{im}), \max_{1 \leq n \leq L} (\beta_{in} \vartheta_{in} I_n(t)) \right\}, \quad i = 1, \dots, N. \quad (28)$$

The quantity  $\theta_i^*(t)$  (28) characterizes the largest version number that can be recovered for an element  $a_i$  based on all available control copies and change log entries at a time  $t$ .

Then the relative version number lag for the element  $a_i$  is defined as:

$$\delta_i(t) = \frac{\max \{0, \theta_i^*(t) - \hat{\theta}_i(t)\}}{\max \{1, \theta_i^*(t)\}}, \quad i = 1, \dots, N. \quad (29)$$

Formula (29) is used to quantify the part of critical data that is returned to correct use after a recovery action, but retains an incomplete recovery by version number.

To assess the load that a recovery action creates on the ISMP subsystems, three components are introduced. The first component characterizes the volume of data transfer and reading during recovery:

$$g^b(t) = \sum_{m=1}^M q_m x_m(t) + \sum_{n=1}^L l_n y_n(t), \quad (30)$$

where  $q_m$  is the amount of data of the control copy  $c_m$  to be read or transferred during recovery;  $l_n$  is the amount of data associated with the use of the log entry  $j_n$ .

The second component characterizes the load on the storage subsystem when using the control copy and the change log:

$$g^s(t) = \sum_{m=1}^M \varphi_m x_m(t) + \sum_{n=1}^L \nu_n y_n(t), \quad (31)$$

where  $\varrho_m$  is the load on the storage subsystem associated with the use of a control copy  $c_m$ ;  $\nu_n$  is the load on the storage subsystem associated with the use of a log entry  $j_n$ . The values  $\varrho_m$  and  $\nu_n$  are measured in normalized units of the storage subsystem load, which reflect the set of operations for reading, positioning, restoring the structure of records and fixing the updated state of the data.

The third component characterizes the amount of computational work required to return the critical part of the data to correct use:

$$g^c(t) = \sum_{n=1}^L \kappa_n y_n(t) + \sum_{i=1}^N \psi_i h_i(t), \quad (32)$$

where  $\kappa_n$  denotes computational costs for re-application of the log record  $j_n$ ;  $\psi_i$  denotes computational costs for forming the restored state of the element  $a_i$  and its return to correct use as part of the critical part of the data. These values characterize the load on the computing resources of the ISMP and are given in normalized units.

Since the ISMP operates on a mobile platform with a limited energy reserve, the execution of the procedures for reading, transmitting, re-applying changes and forming the restored state of the data is accompanied by the costs of the energy resource of the platform. Therefore, the costs of the energy resource of the mobile platform for performing the current restoration action are given by the formula:

$$\Xi(t) = \sum_{m=1}^M p_m x_m(t) + \sum_{n=1}^L s_n y_n(t) + \sum_{i=1}^N v_i h_i(t), \quad (33)$$

where  $p_m$ ,  $s_n$ ,  $v_i$  are the energy resource costs of the mobile platform, respectively, for using the control copy, applying the change log, and returning the critical data element to correct use.

Under the current values of the available ISMP resources:  $r^b(t)$  is the bandwidth available for recovery;  $r^s(t)$  is the storage subsystem performance;  $r^c(t)$  is the available computing performance;  $r^e(t)$  is the available energy reserve.

The generalized recovery action execution time is calculated as:

$$\Theta(t) = \frac{g^b(t)}{r^b(t)} + \frac{g^s(t)}{r^s(t)} + \frac{g^c(t)}{r^c(t)}. \quad (34)$$

To quantify the incompleteness of the recovery of a critical part of the data after the current recovery action is performed, an integral indicator is introduced:

$$\Omega(t) = \sum_{i=1}^N \chi_i (d_i(t) + h_i(t) \delta_i(t)). \quad (35)$$

Then the set of permissible recovery actions at a time  $t$  is given as:

$$W(t) = \left\{ (x(t), y(t), h(t), d(t)) \left\{ \begin{array}{l} \sum_{m=1}^M x_m(t) = 1 \\ y_n(t) \leq \sum_{m=1}^M \omega_{mn}(t) x_m(t), \quad n = 1, \dots, L, \\ y_k(t) \leq y_n(t), \quad \forall n, k : \zeta_{nk} = 1, \\ h_i(t) + d_i(t) = \chi_i \gamma_i(t), \quad i = 1, \dots, N, \\ h_i(t) \leq \sum_{m=1}^M \alpha_{im} x_m(t) + \sum_{n=1}^L \beta_{in} y_n(t), \quad i = 1, \dots, N, \\ g^b(t) \leq r^b(t), \\ g^s(t) \leq r^s(t), \\ g^c(t) \leq r^c(t), \\ \Xi(t) \leq r^e(t), \\ \Theta(t) \leq \Lambda(t), \end{array} \right. \right\} \quad (36)$$

where  $\Lambda(t)$  is the available time interval within which the ISMP must complete the current recovery action.

Thus, formula (36) specifies the set of permissible recovery actions for the current state of post- failure data disruption, available control copies, available change log, and current ISMP resource constraints.

For further selection of the recovery action, it is advisable to introduce the criterion:

$$\Phi(t) = \eta_1 \frac{\Theta(t)}{\Lambda(t)} + \eta_2 \frac{\Omega(t)}{\sum_{i=1}^N \chi_i \gamma_i(t) + \varepsilon} + \eta_3 \frac{\Xi(t)}{r^e(t) + \varepsilon} + \eta_4 \frac{\sum_{i=1}^N d_i(t)}{\sum_{i=1}^N \chi_i \gamma_i(t) + \varepsilon} + \eta_5 \frac{\sum_{m=1}^M (t - \tau_m) x_m(t)}{1 + \max_{1 \leq m \leq M} (t - \tau_m)}, \quad (37)$$

In formula (37)  $\eta_1, \eta_2, \eta_3, \eta_4, \eta_5$  are non-negative coefficients of the criterion setting, which set the relative priority of its components. In this case, it is advisable to assume that  $\eta_1 + \eta_2 + \eta_3 + \eta_4 + \eta_5 = 1$ . The parameter  $\varepsilon > 0$  is a regularization constant, which is introduced into the denominators of the normalizing fractions to preserve the correctness of the criterion calculation in cases of zero or close to zero value of the corresponding normalizing quantity. The value  $\varepsilon$  is selected significantly less than one after normalizing the components of the criterion, so that it does not affect the preference order of permissible restoration actions.

Criterion  $\Phi(t)$  (37) combines five components: the relative duration of recovery, the level of the recovery incompleteness in the critical part of the data, the energy resource consumption of the mobile platform, the proportion of elements of deferred recovery and the time distance of the selected control copy. This construction is consistent with modern approaches, within which the effectiveness of data recovery is assessed taking into account time costs, the load caused by logging, the localization of the recovery area and the completeness degree of data return to use [10], [17], [19], [20].

Then the problem of constructing the smallest sufficient recovery action takes the form:

$$(x^\circ(t), y^\circ(t), h^\circ(t), d^\circ(t)) = \arg \min_{(x, y, h, d) \in W(t)} \Phi(t). \quad (38)$$

The solution to problem (38) determines the recovery action that, given the current state of post- failure data disruption, available control copies, available change log records, and current resource constraints of the ISMP, ensures the shortest relative duration of returning the critical part of the data to correct use with the selected balance between time, energy, and data version indicators.

Thus, a set of admissible recovery actions is constructed through the formal specification of the choice of the control copy, change log records, immediate and deferred recovery elements according to formula (21), a system of constraints according to formulas (22)–(36), and a selection criterion according to formulas (37)–(38).

#### Rule for selecting the smallest sufficient restorative action and the algorithm for method implementation

The constructed set of admissible recovery actions  $W(t)$  determines the area of possible solutions in the post-failure mode of the ISMP. To complete the construction of the method, it is necessary to specify a rule for choosing such a recovery action that ensures the shortest duration of returning the critical part of the data to correct use under current resource constraints. In modern studies devoted to partial control storage, use of the change log and data recovery in distributed environments, such a choice is associated with the simultaneous consideration of time costs, the volume of recoverable changes and incompleteness of recovery after the completion of the current recovery action [17]–[22]. This logic is the basis for further formalization.

Let an arbitrary admissible recovery action at a time  $t$  be represented by a tuple:

$$\pi(t) = (x(t), y(t), h(t), d(t)). \quad (39)$$

To select a recovery action, the following numerical indicators are assigned to each permissible action  $\pi(t)$  (39):

– indicator of the recovery action duration:  $T(\pi(t)) = \Theta(t)$ ;

– indicator of incomplete recovery of the critical part of the data after the current recovery action:

$$L(\pi(t)) = \Omega(t);$$

– indicator of the energy resource consumption of the mobile platform for the recovery action:

$$E(\pi(t)) = \Xi(t);$$

– indicator of the number of damaged elements in the critical part of the data for which immediate recovery

is provided in the current recovery action:  $U(\pi(t)) = \sum_{i=1}^N \chi_i h_i(t)$ .

The indicator  $U(\pi(t))$  determines the number of failed elements in the critical part of the data that are included in the current recovery action as elements of immediate recovery.

To take into account the age of the selected control copy, the indicator is entered:

$$A(\pi(t)) = \sum_{m=1}^M (t - \tau_m) x_m(t). \quad (40)$$

Formula (40) determines the time elapsed from the moment of generating the selected control copy to the current moment. A smaller value  $A(\pi(t))$  corresponds to the use of a newer control copy.

For the post-failure mode of the ISMP, the determining requirement is that after the completion of the current recovery action, the state of the critical part of the data remains within the permissible level of failure. For this purpose, a threshold value  $L_0(t) \geq 0$ , is introduced that sets the largest permissible value of the indicator  $L(\pi(t))$  for the current recovery cycle.

Then the set of recovery actions for which the residual disruption level in the critical part of the data does not exceed the permissible value is defined as

$$W_0(t) = \{ \pi(t) \in W(t) \mid L(\pi(t)) \leq L_0(t) \}. \quad (41)$$

Formula (41) identifies those permissible actions after which the residual level of disruption in the critical part of the data does not exceed the set threshold.

Since in the post-failure mode individual indicators have different priorities, it is advisable to select the smallest sufficient recovery action by sequential selection. This approach reduces the dependence of the result on the precise adjustment of the coefficients and corresponds to the practice of constructing stable selection rules in recovery and control storage problems [10], [18], [19], [21].

First, a set of actions with minimal residual failure is formed:

$$W_1(t) = \left\{ \pi(t) \in W_0(t) \mid L(\pi(t)) = \min_{\pi'(t) \in W_0(t)} L(\pi'(t)) \right\}. \quad (42)$$

Among the actions of the set  $W_1(t)$ , those with the minimum execution duration are selected:

$$W_2(t) = \left\{ \pi(t) \in W_1(t) \mid T(\pi(t)) = \min_{\pi'(t) \in W_1(t)} T(\pi'(t)) \right\}. \quad (43)$$

Next, if there is more than one action left in  $W_2(t)$ , selection is performed based on the minimum energy resource costs of the mobile platform:

$$W_3(t) = \left\{ \pi(t) \in W_2(t) \mid E(\pi(t)) = \min_{\pi'(t) \in W_2(t)} E(\pi'(t)) \right\}. \quad (44)$$

The next selection step takes into account the number of failed elements in the critical part of the data for which the current action provides for immediate recovery:

$$W_4(t) = \left\{ \pi(t) \in W_3(t) \mid U(\pi(t)) = \max_{\pi'(t) \in W_3(t)} U(\pi'(t)) \right\}. \quad (45)$$

If after this there is more than one action left in the set  $W_4(t)$ , the final selection is made based on the minimum time elapsed from the moment the used control copy was formed to the current moment:

$$\pi^\circ(t) = \arg \min_{\pi(t) \in W_4(t)} A(\pi(t)). \quad (46)$$

Formula (46) means that, under other equal conditions, preference is given to the action that uses a control copy with a shorter time elapsed from the moment of its formation to the current moment.

Separately, it is necessary to consider the case when none of the permissible actions of the set  $W(t)$  ensures the fulfillment of the condition  $L(\pi(t)) \leq L_0(t)$ . In such a situation, the ISMP proceeds to the selection of the action that, under the current constraints, ensures the lowest residual level of disruption in the critical part of the data. For this, the rule of reserve selection is introduced:

$$\pi^\diamond(t) = \arg \min_{\pi(t) \in W(t)} \left[ \lambda_1 L(\pi(t)) + \lambda_2 T(\pi(t)) + \lambda_3 E(\pi(t)) - \lambda_4 U(\pi(t)) \right], \quad (47)$$

where  $\lambda_1, \lambda_2, \lambda_3, \lambda_4$  are the non-negative coefficients of the backup selection rule setting, which specify the relative priority of its components. It is advisable to set the ratio  $\lambda_1 ? \lambda_2 ? \lambda_3$  and  $\lambda_1 ? \lambda_4$  so that the minimization of the residual level of disruption in the critical part of the data retains priority even when there is no action satisfying the condition  $L(\pi(t)) \leq L_0(t)$ .

Therefore, the actual selection rule at a time  $t$  takes the form:

$$\pi^*(t) = \begin{cases} \pi^\circ(t), & W_0(t) \neq \emptyset, \\ \pi^\diamond(t), & W_0(t) = \emptyset. \end{cases} \quad (48)$$

The selected action  $\pi^*(t)$  (48) completely determines the structure of the current recovery cycle. The selected control copy number is determined by the formula:

$$m^*(t) = \sum_{m=1}^M mx_m^*(t), \quad (49)$$

where  $x_m^*(t)$  are the components of the selected action  $\pi^*(t)$ .

The set of change log entries used in the current recovery cycle is defined as:

$$J^*(t) = \{j_n \in J \mid y_n^* = 1\}. \quad (50)$$

Let  $J^*(t)$  contain  $Z(t)$  records and after sorting by fixation time it has the form

$$J^*(t) = \{j_{n_1}, j_{n_2}, \dots, j_{n_{Z(t)}}\}, \quad \tau_{n_1} < \tau_{n_2} < \dots < \tau_{n_{Z(t)}}. \quad (51)$$

Ordering (51) is required for correct re-application of the change log records accumulated after the selected control copy was created.

The sets of elements in the critical part of the data for immediate and deferred recovery are given as follows, respectively:

$$\begin{aligned} H^*(t) &= \{a_i \in K \mid h_i^*(t) = 1\}, \\ D^*(t) &= \{a_i \in K \mid d_i^*(t) = 1\}. \end{aligned} \quad (52)$$

For each element  $a_i \in H^*(t)$ , an initial representation for restoration is first formed based on the selected control copy:

$$\bar{y}_i(t) = \begin{cases} y_{im^*(t)}, & \alpha_{im^*(t)} = 1, \\ y_i(t), & \alpha_{im^*(t)} = 0, \end{cases} \quad (53)$$

where  $y_{im^*(t)}$  is the representation of the element  $a_i$  in the control copy  $c_{m^*(t)}$ .

Next, for each log entry  $j_{n_z} \in J^*(t)$ ,  $z = 1, \dots, Z(t)$ , a re-application operator is introduced to the element  $a_i$ :

$$P_{in_z}(w) = \begin{cases} o_{n_z}(w), & \beta_{in_z} = 1, \\ w, & \beta_{in_z} = 0, \end{cases} \quad (54)$$

where  $w$  is the current representation of the data element;  $o_{n_z}$  is the operation recorded in the change log.

Then the result of restoring the element  $a_i \in H^*(t)$  is determined by the composition:

$$R_i(t) = P_{in_{Z(t)}} \circ P_{in_{Z(t)-1}} \circ \dots \circ P_{in_1}(\bar{y}_i(t)). \quad (55)$$

Formula (55) defines the restored representation of a data element after sequentially applying all change log records included in the current restore action.

Then the updated representation of the ISMP data after performing the current restore action is given by the elements:

$$y_i^+(t) = \begin{cases} R_i(t), & a_i \in H^*(t), \\ y_i(t), & a_i \in D^*(t), \\ y_i(t), & \chi_i \gamma_i(t) = 0. \end{cases} \quad (56)$$

Accordingly, the complete data representation after the current recovery cycle is complete has the form:

$$Y_i^+(t) = (y_1^+(t), y_2^+(t), \dots, y_N^+(t)). \quad (57)$$

To evaluate the result of the performed action, the vector of failure indicators is recalculated for each element  $a_i$ :

$$b_i^+(t) = \left( \mu_i^+(t), v_i^+(t), \xi_i^+(t), \zeta_i^+(t) \right). \quad (58)$$

where  $\mu_i^+(t)$ ,  $v_i^+(t)$ ,  $\xi_i^+(t)$ ,  $\zeta_i^+(t)$  are determined by rules (4), but for the updated  $y_i^+(t)$ .

Data element failure indicator after performing the current recovery action:

$$\gamma_i^+(t) = \begin{cases} 1, & \mu_i^+(t) + v_i^+(t) + \xi_i^+(t) + \zeta_i^+(t) > 0, \\ 0, & \mu_i^+(t) + v_i^+(t) + \xi_i^+(t) + \zeta_i^+(t) = 0. \end{cases} \quad (59)$$

Integral assessment of the residual level of disruption in the critical part of the data:

$$\Gamma^+(t) = \sum_{i=1}^N \chi_i \gamma_i^+(t). \quad (60)$$

The quantity  $\Gamma^+(t)$  (60) directly characterizes the result of the current recovery cycle. If  $\Gamma^+(t) = 0$ , then the critical part of the data is returned to correct use within the current cycle. If  $\Gamma^+(t) > 0$ , then the recovery should be continued in the next cycle.

In the general case, the accelerated recovery method is implemented iteratively. Let  $r = 0, 1, \dots$  be the recovery cycle number, and  $Y^{(r)}(t)$  be the ISMP data representation at the beginning of the  $r$ th cycle. Then the rule for selecting the next recovery action:

$$\pi^{(r)}(t) = \Pi \left( Y^{(r)}(t), C, J, r^b(t), r^s(t), r^c(t), r^e(t), \Lambda(t) \right), \quad (61)$$

where  $\Pi(\cdot)$  is the operator for selecting the recovery action according to the rules given in (41)–(48).

The update of the data representation after the  $r$ th cycle is given as:

$$Y^{(r+1)}(t) = F \left( Y^{(r)}(t), \pi^{(r)}(t) \right), \quad (62)$$

where  $F(\cdot)$  is the data view update operator, implemented by formulas (52)–(57). The evaluation of the result of the performed action after updating the data view is carried out by formulas (58)–(60).

The accelerated recovery process is completed when one of the conditions is met:

$$\Gamma^{(r+1)}(t) = 0, \quad \text{or} \quad r = r_{\max}, \quad \text{or} \quad W \left( Y^{(r)}(t), t \right) = \emptyset, \quad (63)$$

where  $r_{\max}$  is the maximum allowable number of recovery cycles;  $W \left( Y^{(r)}(t), t \right)$  is the set of allowable recovery actions formed for the current data representation  $Y^{(r)}(t)$ .

Therefore, the algorithm for implementing the accelerated data recovery method in ISMP in the post-failure mode includes the following stages:

- construction of the current data representation  $Y^{(r)}(t)$  and determination of the vector of failure indicators  $b_i^{(r)}(t)$  and integral assessment  $\Gamma^{(r)}(t)$  (4), (5), (8);
- formation of the set of allowable recovery actions  $W(t)$  (36);
- formation of the set  $W_0(t)$  (41);
- selection of the current recovery action  $\pi^{(r)}(t)$  (42)–(58);
- determination of the index of the selected control copy  $c_{m^*(t)}$ , the set of log entries  $J^*(t)$ , the sets  $H^*(t)$  and  $D^*(t)$  (49)–(52);
- calculation of the updated data representation  $Y^{(r+1)}(t)$  (53)–(57);
- evaluation of the result of the performed recovery action and checking the completion condition according to formulas (58)–(63).

Thus, the construction of a method for accelerated data recovery in ISMP in post-failure mode has been completed. The results obtained define a complete apparatus for selecting and performing a recovery action aimed at quickly returning the critical part of ISMP data to correct use.

**Evaluating the developed method effectiveness**

The evaluation of the method effectiveness is performed in discrete time on a finite set of post- failure data disruption scenarios:

$$\Sigma = \left\{ \sigma_q \right\}_{q=1}^Q,$$

where  $\sigma_q$  is the  $q$ th scenario;  $Q$  is the number of scenarios.

Each scenario  $\sigma_q$  is given by a tuple:

$$\sigma_q = \left( N_q, M_q, L_q, P_q, R_q, \Lambda_q \right),$$

where  $N_q$  is the number of data elements;  $M_q$  is the number of available control copies;  $L_q$  is the number of available change log entries;  $P_q$  is parameters of post- failure data disruption;  $R_q$  is the vector of available ISMP resources;  $\Lambda_q$  is allowable recovery duration.

For each scenario  $\sigma_q$ , the initial level of critical data failure is determined by the formula:

$$\Gamma_q^0 = \Gamma(t_q^0),$$

where  $t_q^0$  is the initial moment of recovery in the scenario  $\sigma_q$ .

The result of executing the method in the scenario  $\sigma_q$  is given by the sequence of recovery actions:

$$\Pi_q = \left\{ \pi_q^{(r)} \right\}_{r=0}^{r_q^*},$$

where  $\pi_q^{(r)}$  is recovery action on the  $r$ th cycle;  $r_q^*$  is the final cycle number in the scenario  $\sigma_q$ .

Such a representation allows to perform a single comparative assessment for all considered recovery methods under the same conditions of post- failure data disruption.

To confirm the effectiveness of the proposed method, it is advisable to consider four recovery methods applied to each scenario  $\sigma_q \in \Sigma$ :

- $Y_1$  is the recovery of the critical part of the data using the closest control copy in time without re-application of the change log;
- $Y_2$  is the recovery using the control copy with subsequent re-application of the entire available fragment of the change log related to the critical part of the data;
- $Y_3$  is the recovery of only the failed elements in the critical part of the data without optimizing the choice of the recovery action;
- $Y_4$  is the recovery using the proposed method of accelerated data recovery, in which the recovery action is selected according to formulas (42)–(48).

Comparison  $Y_1 - Y_4$  allows to assess the impact of choosing the smallest sufficient recovery action on the recovery result.

For each scenario  $\sigma_q$  and for each recovery method  $Y_u$ ,  $u = 1, 2, 3, 4$  the recovery duration is defined as

$$T_{qu} = \sum_{r=0}^{r_{qu}^*} \Theta_{qu}^{(r)}, \tag{64}$$

where  $\Theta_{qu}^{(r)}$  is the duration of the  $r$ th recovery cycle;  $r_{qu}^*$  is the number of the final cycle.

The sign of successful recovery in the permissible interval:

$$S_{qu} = \begin{cases} 1, & \Gamma_{qu}^+ = 0 \wedge T_{qu} \leq \Lambda_q, \\ 0, & \text{otherwise.} \end{cases} \tag{65}$$

where  $\Gamma_{qu}^+$  is the residual level of disruption in the critical part of the data after the recovery is complete.

The amount of data read and transferred during recovery is determined as:

$$V_{qu} = \sum_{r=0}^{r_{qu}^*} g_{qu}^{b(r)}, \tag{66}$$

where  $g_{qu}^{b(r)}$  is the amount of data transmission and reading on the  $r$ th cycle.

The integral energy costs of recovery are determined explicitly

$$E_{qu} = \sum_{r=0}^{r_{qu}^*} \Xi_{qu}^{(r)}, \tag{67}$$

where  $\Xi_{qu}^{(r)}$  is the energy consumption in the  $r$ th cycle.

To compare the recovery methods across the entire set of scenarios, the average values of the indicators are used:

$$\bar{T}_u = \frac{1}{Q} \sum_{q=1}^Q T_{qu}, \quad \bar{S}_u = \frac{1}{Q} \sum_{q=1}^Q S_{qu}, \quad \bar{V}_u = \frac{1}{Q} \sum_{q=1}^Q V_{qu}, \quad \bar{E}_u = \frac{1}{Q} \sum_{q=1}^Q E_{qu}. \tag{68}$$

The computational study was performed in a simulation environment [23], the values obtained by the indicators (64)–(68) are given in [24] and in Fig. 1–4. The comparison was performed for four recovery methods  $\Upsilon_1$  –  $\Upsilon_4$ .

The best results were obtained for the proposed method  $\Upsilon_4$ : the average recovery duration is 9.85, the proportion of successful recoveries is 0.620, the average amount of returned data is 627.84, the integral energy costs are 349.02, and the residual disruption of the critical part of the data is 0.131.

Compared with the method  $\Upsilon_3$ , the method  $\Upsilon_4$  reduces the average recovery time by 28.6%, the energy resource consumption by 20.5%, the residual level of critical data disruption by 72.3%, and increases the proportion of successful recovery completion from 0.229 to 0.620. Compared with the method  $\Upsilon_2$ , the reduction in the average recovery time is 62.1%, the volume of data transmission and reading is 50.6%, the energy resource consumption is 66.0%, and the residual level of critical data disruption is 54.4%. Compared with the method  $\Upsilon_1$ , the residual level of disruption is reduced by 85.2%, and the average recovery time is 27.8%.

Fig. 1 shows the dependence of the average recovery time on the proportion of critical data failed elements. For all methods, with an increase in the fraction of the disruption, the duration of recovery increases, but in the entire studied range, the method  $\Upsilon_4$  retains the lowest values. For a fraction of 0.10, the average duration of recovery for  $\Upsilon_4$  is 6.62, while for  $\Upsilon_3$  it is 9.95, for  $\Upsilon_1$  it is 9.37, for  $\Upsilon_2$  it is 18.49. For a fraction of 0.58, the corresponding values are 13.27, 17.66, 17.92 and 33.49. Therefore, with an increase in the scale of the post-failure disruption, the advantage of the proposed method is preserved.

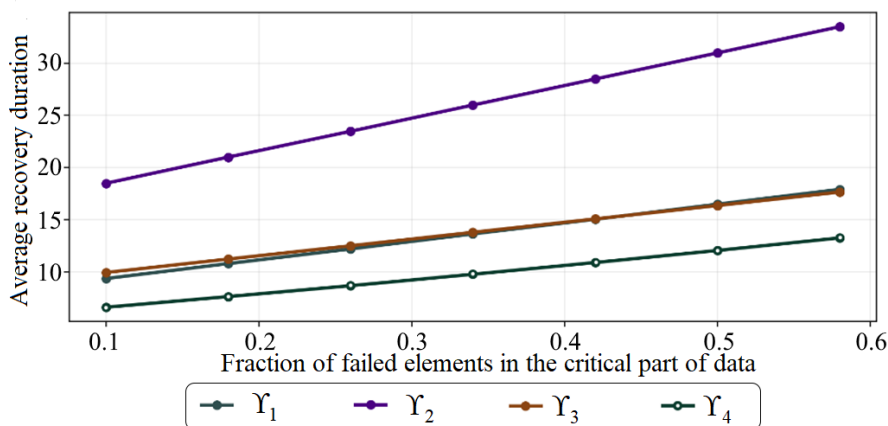


Fig. 1. Dependence of the average recovery duration on the fraction of failure

Fig. 2 shows the change in the average residual disruption in the critical part of the data depending on the depth of the change log. For all methods, with increasing log depth, an increase in residual disruption is observed, which is associated with an increase in the volume of changes that must be returned into the working state after the control copy. At the same time, the curve for  $\Upsilon_4$  remains the lowest in the entire range: for a log depth of 20, the

average residual disruption is 0.117, and for a depth of 120 it is 0.145. For the method  $\Upsilon_2$ , these values are 0.259 and 0.315, for  $\Upsilon_3$  they are 0.424 and 0.520, for  $\Upsilon_1$  they are 0.801 and 0.963. This confirms that with an increase in the volume of post-control changes, the proposed method better keeps the critical part of the data within the limits of correct use.

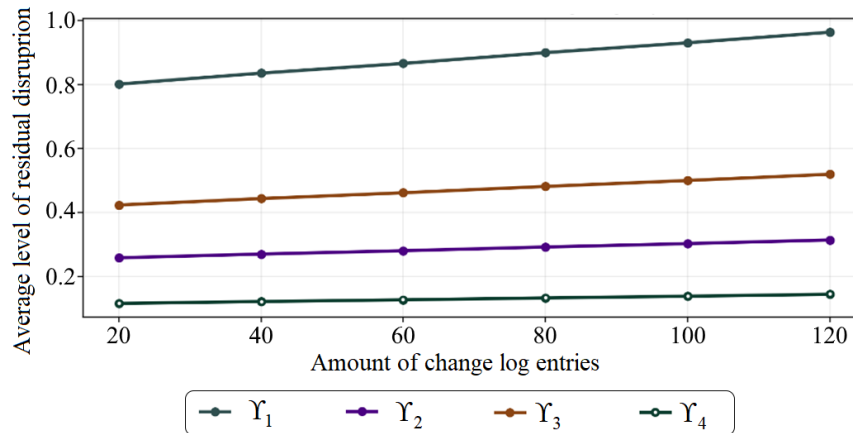


Fig. 2. Dependence of the average level of residual disruption on the number of change log entries

Fig. 3 shows the dependence of the proportion of successful completion of recovery on the level of available ISMP resource. For all methods, the value of this indicator increases with an increase in the available resource. This is most clearly observed for the method  $\Upsilon_4$ : from 0.159 at a resource level of 0.35 to 0.976 at a level of 1.00. For the method  $\Upsilon_3$ , the corresponding values are 0.063 and 0.317, for  $\Upsilon_2$  it is 0.000 and 0.349, and for  $\Upsilon_1$  within the studied scenarios, successful completion of recovery was not recorded. Therefore, with an increase in the available resource, the proposed method provides the greatest increase in the proportion of successful completion of recovery.

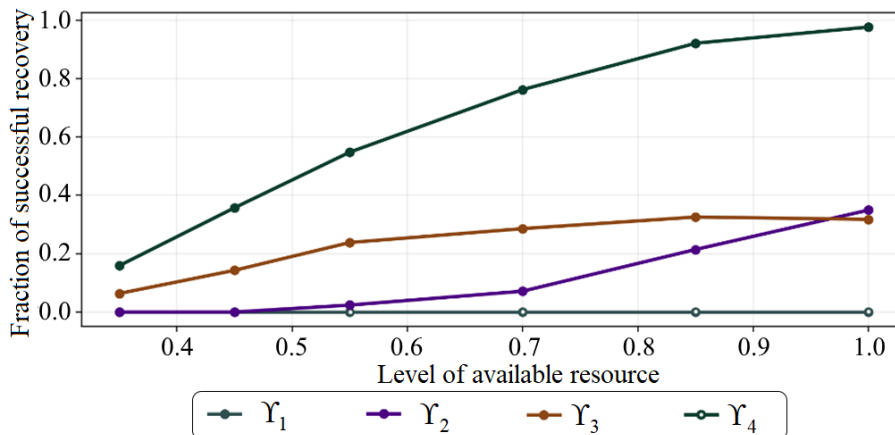


Fig. 3. Dependence of successful recovery completion fraction on the level of available resources

The results show that the proposed method has an advantage over the basic recovery methods in terms of a set of key indicators. This advantage is conditioned by the choice of the smallest sufficient recovery action, which is formed taking into account the available control copies, change log records and current resource limitations of the ISMP. In methods that use only a control copy or the entire available fragment of the change log, recovery covers the excess amount of data and service records. In the proposed method, recovery is limited to only those data and change log records that are required to return the critical part of the data to correct use. This is what explains the reduction in the duration of recovery, the amount of data transfer and reading, energy resource consumption and the level of residual disruption of the critical part of the data.

Comparison with  $\Upsilon_1$  shows that using only a control copy does not provide an acceptable level of performance in the post-failure mode of the ISMP. This method is characterized by a moderate recovery duration, but leaves the greatest level of disruption in the critical part of the data, since it does not compensate for the changes accumulated after the formation of the control copy. Therefore, even with a relatively short recovery duration, it does not ensure the successful completion of the recovery in an acceptable time interval.

Comparison with  $\Upsilon_3$  is the most indicative, since this method is already focused on the critical part of the data. However,  $\Upsilon_3$  lacks an optimized choice of recovery action. As a result, recovery is performed only for the failed elements without consistent consideration of the selected control copy, change log entries, and the time costs of the current cycle. This provides better results than in  $\Upsilon_1$  and  $\Upsilon_2$ , but is inferior  $\Upsilon_4$  in all key indicators.

It is worth noting the stability of the proposed method to the complexity of the scenarios. As shown in Fig. 1 and Fig. 2, with an increase in the share of failed elements in the critical part of the data and the number of change log entries, the values of all indicators change naturally for each of the methods. However, the variant  $\Upsilon_4$  retains the best or closest to the best values of the indicators in the entire studied range. This indicates that the construction of the smallest sufficient recovery action provides better adaptability to the growth of the scale in post-failure data disruption.

The dependence given in Fig. 3 shows another important property of the method. With an increase in the available ISMP resource, the share of successful recovery completion for  $\Upsilon_4$  grows significantly faster than for the basic methods. This means that the proposed method more effectively converts the additional resource into an increase in recovery efficiency. In the applied aspect, this property is important for ISMP, where the resource often changes abruptly and must be used within a short interval with maximum performance.

Therefore, the results of the computational study confirm that the proposed method of accelerated data recovery in ISMP in post-failure mode provides shorter recovery duration, smaller amount of data transmission and reading, lower energy resource consumption and lower level of residual disruption in the critical part of the data, and also increases the proportion of successful completion of recovery in the permissible time interval compared to the basic methods  $\Upsilon_1 - \Upsilon_3$ .

### Conclusions

The article solves the problem of developing a method for accelerated data recovery in ISMP in post-failure mode, focused on the rapid return of the critical part of the data to correct use based on the available control copies, change log records and current ISMP resource limitations. A formal representation of post-failure data disruption is constructed, a critical part of the data, a set of control copies and a change log are specified, and a set of permissible recovery actions is formed, each of which is determined by the choice of a control copy, a change log fragment, the composition of immediate recovery data and the composition of deferred recovery data.

A rule for selecting the smallest sufficient recovery action and an algorithm for implementing a method, which in a post-failure mode ensures a reduction in the duration of returning the critical part of the data to correct use under the current resource limitations of the mobile platform, has been developed. Model studies have shown that the proposed method has an advantage over basic recovery methods in terms of recovery duration, data transfer and reading volume, mobile platform energy resource consumption, the level of residual disruption to the critical part of the data, and the proportion of successful recovery completion within the permissible time interval. This confirms the feasibility of using the developed method as a separate means of maintaining the survivability of the ISMP at the data level.

The scientific novelty of the results obtained lies in the fact that the approach to data recovery in ISMP in the post-failure mode has been further developed by formalizing the smallest sufficient recovery action, which is determined by the joint choice of a control copy, a fragment of the change log, the composition of the data for immediate recovery and the composition of the data for deferred recovery, taking into account the current resource of the mobile platform and the requirement of a quick return of the critical part of the data to correct use, as well as by developing a formal representation of a post-failure data disruption through failure indicators, a critical part of the data, a set of control copies and a change log and developing a rule for selecting a recovery action, which ensures a consistent consideration of the recovery duration, the residual level of the critical part of the data failure, energy resource consumption and the number of elements for which immediate recovery is provided in the current action.

The practical significance of the results obtained lies in the possibility of using the developed method in on-board and distributed ISMPs, where after a data failure it is necessary to restore a critical part of the information basis of functioning in a short time interval without going to a full recovery of the entire data set.

It is advisable to associate the prospects of further research with the extension of the method to multi-node data restoration in a distributed ISMP, taking into account several simultaneous sources of post-failure disruption and developing adaptive rules for selecting recovery actions under varying rates of change-log entries and variable availability of mobile platform resources.

### ADDITIONAL INFORMATION

#### AUTHOR CONTRIBUTIONS

Conceptualization, I.R.; methodology, V.T.; validation, I.R.; formal analysis, V.T.; investigation, V.T.; data curation, I.R.; writing-original draft preparation, V.T.; writing-review and editing, I.R.; visualization, V.T.; supervision, I.R.; project administration, I.R. All authors have read and agreed to the published version of the manuscript.

## DECLARATION ON THE USE OF GENERATIVE ARTIFICIAL INTELLIGENCE TOOLS

In preparing this work, the author used DeepL Translate and Grammarly for: grammar and spelling checks, paraphrasing, and rephrasing. After using these tools/services, the author reviewed and edited the content and takes full responsibility for the content of this publication.

## References

1. Tkachov V. Method of restoring critical information system services on a mobile platform under conditions of controlled degradation. *Management Information System and Devices*. 2026. No. 188. P. 78–97. URL: <https://doi.org/10.30837/0135-1710.2026.188.078>.
2. Edge computing in future wireless networks: A comprehensive evaluation and vision for 6G and beyond / M. Ergen et al. *ICT Express*. 2024. URL: <https://doi.org/10.1016/j.icte.2024.08.007>.
3. Task offloading strategies for mobile edge computing: A survey / S. Dong et al. *Computer Networks*. 2024. P. 110791. URL: <https://doi.org/10.1016/j.comnet.2024.110791>.
4. Resilient edge-to-cloud architecture with self-healing and self-correcting mechanisms for industrial data continuity / M. Boluda-Prieto et al. *Computers & Industrial Engineering*. 2026. Vol. 213. P. 111795. URL: <https://doi.org/10.1016/j.cie.2025.111795>.
5. Real-Time Service Migration in Edge Networks: A Survey / Y. Zhang et al. *Journal of Sensor and Actuator Networks*. 2025. Vol. 14, no. 4. P. 79. URL: <https://doi.org/10.3390/jsan14040079>.
6. Service migration with edge collaboration: Multi-agent deep reinforcement learning approach combined with user preference adaptation / S. Chen et al. *Future Generation Computer Systems*. 2024. P. 107612. URL: <https://doi.org/10.1016/j.future.2024.107612>.
7. Colarusso C., Falco I., Zimeo E. Business continuity of Cloud-based IoT applications through a seamless continuum. *Internet of Things*. 2025. P. 101723. URL: <https://doi.org/10.1016/j.iot.2025.101723>.
8. Khelaifa A., Benharzallah S., Kahloul L. A new adaptive causal consistency approach in Edge Computing Environment. *International Journal of Computing and Digital Systems*. 2022. Vol. 12, no. 4. P. 945–960. URL: <https://doi.org/10.12785/ijcids/120177>.
9. Fouto P., Prego N., Leitão J. Large-Scale Causal Data Replication for Stateful Edge Applications. 2024 *IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*, Jersey City, NJ, USA, 23–26 July 2024. P. 209–220. URL: <https://doi.org/10.1109/icdc60910.2024.00028>.
10. Takdir, Kitagawa H., Amagasa T. Local recovery and partial snapshot in distributed stateful stream processing. *Knowledge and Information Systems*. 2025. URL: <https://doi.org/10.1007/s10115-025-02509-z>.
11. Dynamic Scaling with Just-in-Time Snapshot in Distributed Stream Processing Systems / Takdir et al. *IEEE Access*. 2025. P. 1. URL: <https://doi.org/10.1109/access.2025.3638805>.
12. Efficient Fault Tolerance for Stateful Serverless Computing with Asymmetric Logging / S. Qi et al. *ACM Transactions on Computer Systems*. 2025. URL: <https://doi.org/10.1145/3725985>.
13. Müller R. H., Meinhardt C., Mendizabal O. M. An architecture proposal for checkpoint/restore on stateful containers. *SAC '22: The 37th ACM/SIGAPP Symposium on Applied Computing*, Virtual Event. New York, NY, USA, 2022. URL: <https://doi.org/10.1145/3477314.3507221>.
14. Design, Modeling, and Implementation of Robust Migration of Stateful Edge Microservices / A. Calagna et al. *IEEE Transactions on Network and Service Management*. 2023. P. 1. URL: <https://doi.org/10.1109/tnsm.2023.3331750>.
15. Live Migration of Multi-Container Kubernetes Pods in Multi-Cluster Serverless Edge Systems / L. Poggiani et al. *SEATED '24: 1st Workshop on Serverless at the Edge*, Pisa Italy. New York, NY, USA, 2024. P. 9–16. URL: <https://doi.org/10.1145/3660319.3660330>.
16. Evaluation of Storage Placement in Computing Continuum for a Robotic Application / Z. Bakhshi et al. *Journal of Grid Computing*. 2024. Vol. 22, no. 2. URL: <https://doi.org/10.1007/s10723-024-09758-2>.
17. Clonos / P. F. Silvestre et al. *SIGMOD/PODS '21: International Conference on Management of Data*, Virtual Event China. New York, NY, USA, 2021. URL: <https://doi.org/10.1145/3448016.3457320>.
18. CheckMate: Evaluating Checkpointing Protocols for Streaming Dataflows / G. Siachamis et al. 2024 *IEEE 40th International Conference on Data Engineering (ICDE)*, Utrecht, Netherlands, 13–16 May 2024. URL: <https://doi.org/10.1109/icde60146.2024.00309>.
19. Technology for ensuring the survivability of territorially distributed information computer systems in a unified information space / O. Dodonov et al. *Data Recording, Storage & Processing*. 2024. Vol. 26, no. 1. P. 121–143. URL: <https://doi.org/10.35681/1560-9189.2024.26.1.308659>.
20. Combining Markov and Semi-Markov Modelling for Assessing Availability and Cybersecurity of Cloud and IoT Systems / V. Kharchenko et al. *Cryptography*. 2022. Vol. 6, no. 3. P. 44. URL: <https://doi.org/10.3390/cryptography6030044>.

21. FlexLog: A Shared Log for Stateful Serverless Computing / D. Giantsidi et al. *HPDC '23: The 32nd International Symposium on High-Performance Parallel and Distributed Computing*, Orlando FL USA. New York, NY, USA, 2023. URL: <https://doi.org/10.1145/3588195.3592993>.

22. Styx: Transactional Stateful Functions on Streaming Dataflows / K. Psarakis et al. *Proceedings of the ACM on Management of Data*. 2025. Vol. 3, no. 3. P. 1–28. URL: <https://doi.org/10.1145/3725363>.

23. Tkachov V. GitHub - tikey/recovery\_study. *GitHub*. URL: [https://github.com/tikey/recovery\\_study](https://github.com/tikey/recovery_study).

24. Tkachov, V. (2026). Source figures and tables for the study "Method of Accelerated Data Recovery in Information System on Mobile Platform in Post-Failure Mode" [Data set]. *Zenodo*. URL: <https://doi.org/10.5281/zenodo.19915451>.

Віталій ТКАЧОВ, Ігор РУБАН  
Харківський національний університет радіоелектроніки

## МЕТОД ПРИСКОРЕНОГО ВІДНОВЛЕННЯ ДАНИХ В ІНФОРМАЦІЙНІЙ СИСТЕМІ НА МОБІЛЬНІЙ ПЛАТФОРМІ В ПІСЛЯВІДМОВНОМУ РЕЖИМІ

У статті розглядається проблема прискореного відновлення даних в інформаційній системі на мобільній платформі (ІСМП) у післяаварійному режимі. Актуальність дослідження випливає з того, що після аварійного збою даних в ІСМП виникає необхідність якомога швидше відновити критичну частину даних до належного використання, враховуючи обмежені обчислювальні, мережеві та енергетичні ресурси. Метою роботи є розробка методу прискореного відновлення даних, який після збою даних забезпечує вибір найменшої достатньої дії відновлення на основі контрольної копії та журналу змін і скорочує тривалість повернення критичної частини даних до правильного використання за поточних ресурсних обмежень ІСМП. У роботі формалізовано післяаварійне відновлення даних за допомогою набору елементів даних, індикаторів збою, критичної частини даних, набору контрольних копій та журналу змін. Побудовано набір допустимих дій відновлення, в межах якого кожна дія визначається вибором контрольної копії, фрагмента журналу змін, складом даних негайного відновлення та складом даних для відкладеного відновлення. Розроблено правило вибору найменшої достатньої дії відновлення та алгоритм реалізації методу, який враховує тривалість відновлення, рівень залишкових порушень критичної частини даних, споживання енергоресурсів та кількість елементів, для яких передбачено негайне відновлення в поточній дії. За результатами обчислювального дослідження встановлено, що запропонований метод має перевагу над базовими методами відновлення за тривалістю відновлення, обсягом передачі та зчитування даних, споживанням енергоресурсів, рівнем залишкових порушень критичної частини даних та часткою успішного завершення відновлення протягом прийнятного інтервалу часу. Практичне значення отриманих результатів полягає в придатності методу для використання в бортових та розподілених ІСМП, де після аварійного збою даних необхідно відновити критичну частину інформаційної бази для роботи протягом короткого інтервалу часу, не переходячи до повного відновлення всього набору даних..

Ключові слова: інформаційна система на мобільній платформі; післяаварійний режим; відновлення даних; контрольна копія; живучість.