УДК 004.65 DOI: 10.31673/2412-4338.2019.012230

Собчук В.В.[1], **Коваль М.О.**[2], **Мусієнко А.П.**[3], **Мацько О.Й.**[4]
[1]*Східноєвропейський національний університет ім. Лесі Українки, Луцьк*
[2]*Київський національний університет імені Тараса Шевченка, Київ*
[3]*Державний університет телекомунікацій, Київ*
[4]*Національний університет оборони України ім. І. Черняховського, Київ*

МЕТОД ДІАГНОСТУВАННЯ ПРИХОВАНИХ ВІДМОВ В ІНФОРМАЦІЙНІЙ СИСТЕМІ НА ОСНОВІ ЗАСТОСУВАННЯ ДВОРІВНЕВОЇ СИСТЕМИ ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНАЛЬНОЇ СТІЙКОСТІ

В роботі на основі використання ієрархічної концепції організації засобів забезпечення функціональної стійкості інформаційної системи підприємства, розроблено два алгоритми, які утворюють дворівневу систему діагностування прихованих відмов. Діагностування розпочинається з виконання першого алгоритму, переваги якого в порівнянні з відомими алгоритмами полягають в тому, що він вимагає меншої надмірності системи, всього два раунди обміну повідомленнями між вузлами інформаційної системи і забезпечує діагностування інформаційної системи підтриємства при відмові майже половини її вузлів.

У разі неоднозначного рішення задачі діагностування, алгоритм формує сигнал про свою неспроможність і діагностування інформаційної системи триває по другому алгоритму, який в якості критерію використовує тривалість виконання фаз.

Ключові слова: *функціональна стійкість, автоматизована система управління підприємством, зовнішні та внутрішні дестабілізуючі фактори.*

Sobchuk V.V.[1], **Koval M.O.**[2], **Musienko A.P.**[3], **Matsko O.Y.**[4]
[1]*Lesya Ukrainka Eastern European National University, Lutsk*
[2]*Kyiv National Taras Shevchenko University, Kyiv*
[3]*State University of Telecommunications, Kyiv*
[4]*Ivan Cherniahovsky National Defense University of Ukraine, Kyiv*

DIAGNOSTIC METHOD OF OBSERVED DISCONTINUES IN THE INFORMATION SYSTEM BASED ON THE USE OF THE TWO-LEVEL SYSTEM FOR PROVISION OF FUNCTIONAL SUSTAINABILITY

In this paper one of the main properties of enterprise information systems is considered - functional stability. Under functional stability refers to the property of an enterprise information system to store, during a given time, the performance of its main functions within the limits set by regulatory requirements, under conditions of influence of flows of failures, malfunctions, failures. This property is closely related to the properties of stability, reliability, durability and fault-tolerance.

In accordance with the concept of hierarchical organization of means of ensuring functional stability, for the diagnosis of information system with hidden failures, several levels should be used. Hidden refusal is a refusal which is not revealed visually or by staff methods and means of control and diagnostics, but is revealed during maintenance or special methods of diagnosis. At the first level, simple diagnostic procedures are used. If they can identify the rejected sites, then this diagnosis ends with the least waste of time. In the event of uncertainty in the definition of the components of the rejected system, the next level of the hierarchy must run diagnostic procedures that can handle more complex failures.

Based on the use of the hierarchical concept of organization of means for ensuring the functional stability of the enterprise information system, two algorithms have been developed that form a two-level system for diagnosing hidden failures. Diagnosis begins with the implementation of the first algorithm, the advantages of which compared to known algorithms are that it requires less redundancy of the system, only two rounds of communication between the nodes of the enterprise information system and provides diagnosis of the information system in the rejection of almost half of its nodes.

In the case of an ambiguous solution to the diagnostic problem, the algorithm generates a signal about its failure and the diagnosis of the information system proceeds according to a second algorithm, which uses the duration of the phases as a criterion.

Keywords: *functional stability, automated enterprise management system, external and internal destabilizing factors.*

Собчук В.В.[1], **Коваль Н.А.**[2], **Мусиенко А.П.**[3], **Мацько О.И.**[4]

[1]*Восточноевропейский национальный университет им. Леси Украинский, Луцк*

[2]*Киевский национальный университет имени Тараса Шевченко, Киев*

[3]*Государственный университет телекоммуникаций, Киев*

[4]*Национальный университет обороны Украины им. И. Черняховского, Киев*

МЕТОД ДИАГНОСТИКИ СКРЫТЫХ ОТКАЗОВ В ИНФОРМАЦИОННОЙ СИСТЕМЕ НА ОСНОВЕ ПРИМЕНЕНИЯ ДВУХУРОВНЕВОЙ СИСТЕМЫ ОБЕСПЕЧЕНИЕ ФУНКЦИОНАЛЬНОЙ УСТОЙЧИВОСТИ

В работе на основе использования иерархической концепции организации средств обеспечения функциональной устойчивости информационной системы предприятия, разработаны два алгоритма, которые образуют двухуровневую систему диагностирования скрытых отказов. Диагностирования начинается с выполнения первого алгоритма, преимущества которого по сравнению с известными алгоритмами заключаются в том, что он требует меньшей избыточности системы, всего два раунда обмена сообщениями между узлами информационной системы и обеспечивает диагностирование информационной системы предприятия при отказе почти половины ее узлов.

В случае неоднозначного решения задачи диагностирования, алгоритм формирует сигнал о своей несостоятельности и диагностирования информационной системы продолжается по второму алгоритму, который в качестве критерия использует продолжительность выполнения фаз.

Ключевые слова: *функциональная устойчивость, автоматизированная система управления предприятием, внешние и внутренние дестабилизирующие факторы.*

Вступ. Інформаційні системи широко використовуються в усіх сферах людської діяльності. На даний час не можливо уявити жодного підприємства, яке б не використовувало інформаційні системи автоматизованого управління. Аналіз функціонування таких систем, показав, що їх характеризують такі властивості як: відмовостійкість, надійність, стійкість, живучість. Проте, дані властивості не зовсім повно обписують поведінку інформаційної системи автоматизованого управління при дії різного типу дестабілізуючих чинників, які в свою чергу викликають відмови і збої. Тому, необхідно розглянути ще одну властивість інформаційних систем автоматизованого управління – функціональну стійкість.

Вперше поняття функціональної стійкості було введене в роботах О.А. Машкова [1]. Функціональна стійкістю інформаційної системи – це властивість системи зберігати упродовж визначеного часу виконання своїх основних функцій в межах, встановлених нормативними вимогами, в умовах впливу потоків відмов, несправностей, збоїв [1]. Дана властивість тісно пов'язана з властивостями стійкості, надійності, живучості і відмовостійкості. Терміни надійність, живучість і відмовостійкість визначені в [1, 2].

Аналіз літературних джерел. В роботах Ю.П. Зайченка, В.К. Попкова, В.С. Семеніхіна, О.А. Машкова, О.В. Барабаша, Д.М. Обідіна, Ю.В. Кравченка, Г.А. Кучука розглянуто методи побудови оптимальних складних технічних систем.

Властивість відмовостійкості складних технічних систем досліджувалось в роботах А.А. Авіжиєніса, В.А. Машкова, О.Ю. Ільїна та інших вчених. Питання стійкості систем щодо зовнішніх дестабілізуючих факторів досліджувалось О.Г. Додоновим, І.В. Рубаном, І.Ю. Субачем, Ю.В. Журавським.

Після проведеного аналізу можна зробити висновок, що в основному в роботах розглядаються питання проектування та оптимізації складних технічних систем. В досліджуваних системах ефективність функціонування залежить від обраного показника якості. До таких показників можна віднести: вартість проектування і експлуатації системи, середній час затримки повідомлення в мережі та надійність елементів системи. Крім того багато уваги приділено задачі синтезу живучих і надійних мереж. Проте в цих роботах недостатньо уваги приділено діагностуванню інформаційних систем автоматизованого управління. В проаналізованих роботах приділяється основна увага побудові систем діагностування на принципах функціонального та тестового діагностування постійних відмов. Але інтелектуальні системи, що мають в своєму складі обчислювальні системи, характеризуються впливом на них потоку відмов. При чому інтенсивність потоку збоїв та нестійких відмов не менше інтенсивності постійних відмов. Тому питання діагностування інформаційних систем на основі взаємного інформаційного узгодження на сьогоднішній день є актуальним.

Постановка завдання в загальному вигляді. Відповідно до концепції про ієрархічну організацію засобів забезпечення функціональної стійкості, для проведення діагностування інформаційної системи з прихованими відмовами слід використати декілька рівнів [3, 4]. Прихована відмова – це відмова, що не виявляється візуально чи штатними методами й засобами контролю та діагностики, але виявляється під час проведення технічного обслуговування чи спеціальними методами діагностування [5]. На першому рівні використовуються прості діагностичні процедури. Якщо вдається з їх допомогою визначити вузли, що відмовили, то на цьому діагностування завершується з найменшими витратами часу. У разі неоднозначності визначення компонентів системи, що відмовили, наступний рівень ієрархії повинен запустити діагностичні процедури, здатні впоратися із складнішими відмовами. Даний підхід ґрунтується на припущенні того, що простіші відмови проявляються частіше, чим складніші [6, 7]. Тому не потрібно відразу застосовувати потужні, діагностичні методи, які вимагають великих часових витрат.

Розробка алгоритму діагностування інформаційної системи на основі взаємного інформаційного узгодження. Розглянемо інформаційну систему, що складається з N вузлів з номерами $1 \ldots N$. Перевірка здійснюється шляхом взаємного обміну повідомленнями при припущеннях про синхронність інформаційної системи і про можливість одержувачем повідомлення визначити його відправника.

Алгоритм А1 складається з двох етапів:

– етапу пересилок повідомлень (кроки 1-3);

– етапу аналізу отриманих повідомлень (кроки 4-7).

Етап аналізу виконується кожним вузлом автономно на основі повідомлень, отриманих ним на етапі пересилок.

Алгоритм А1 полягає в наступному.

Крок 1. *k*-ий вузол посилає іншим *n*-м вузлам ($n \neq k$) повідомлення Z_k з множини $Z = \left\{a, \overline{a}, \varnothing\right\}$. Для зручності вважатимемо номер *k* рівним $2t+2$.

Крок 2. *n* вузлів обмінюються повідомленнями, отриманими від *k*-го вузла на кроці 1. При цьому вузли, в яких має місце прихована відмова, можуть передавати суперечливі повідомлення різним вузлам.

Кожний *n*-й вузол ($n = 1, ..., N-1; \quad n \neq k$) формує з отриманих повідомлень вектор $STR(n)$, що містить $2t+1$ елементи, для розміщення повідомлень, отриманих від всіх вузлів, включаючи і себе на цьому кроці $STR(n) = (Z_1, Z_2, ..., Z_{N-1})$.

Крок 3. *n* вузлів обмінюються векторами *STR(n)* ($n = 1, ..., N-1$), з яких кожний формує початковий набір у вигляді матриці A_n, в якій вектори *STR(n)* ($n = 1, ..., N-1$)

розташовуються в вигляді рядків в порядку зростання номерів вузлів-посилачів, яких можна завжди однозначно визначити згідно припущення 1.

Крок 4. *n*-й вузол, застосовуючи функцію *majority* до стовпців матриці A_n, отримує по одному значенню цієї функції для кожного стовпця, з яких формує вектор PRS_n, елементи якого рівні $PRS_n(j) = majority\{ a_{ij}^n \mid i = 1...N\}$.

Крок 5. *n*-й вузол визначає елементи a_{ij}^{n*} *i, j*= 1...*N* - 1 власної матриці A_n, які знаходяться на перетині стовпця *j* з рядком *i,* якщо

$$PRS_n(j) \neq a_{ij}^{n*} . \quad (1)$$

Позначимо загальне число помічених елементів через $L_{\max}$. Якщо $L_{\max} = 0$, то перейти до кроку 7.

Крок 6. *n*-й вузол визначає підозрювану область у вигляді логічного виразу

$$\sum \Pi = \bigwedge_{L=1}^{L_{\max}} (i_l \vee j_l), \quad (2)$$

де *i, j* – номери вузлів з елементів матриці a_{ij}^{n*}, що відповідають номеру рядка *i* і стовпця *j* елементу матриці A_n що відрізняється від $PRS_n(j)$.

Далі рівність (2) зводиться до диз'юнктивної нормальної форми, для цього розкриваються дужки і виконуються перетворення, тобто вираження з кон'юнкції зводиться до виду диз'юнкції кон'юнкцій. При цьому для обліку обмеження в не більше ніж в *t* несправних вузлах виключаються з розгляду терми з рангом більше *t*, тобто з більш ніж *t* елементами. Кожен з термів, що залишилися, визначає допустиме поєднання несправностей, які можуть привести до усіх виявлених несправностей за умови, що їх в системі є не більше ніж *t.* Якщо у рівності $\sum \Pi$ залишається більше ніж одна терма, то результат діагностування неоднозначний і $DIAGNOZ = 1$, інакше $DIAGNOZ = 0$.

Крок 7. *n*-ті вузли формують матриці B_n отримані з A_n шляхом викреслювання рядків і стовпців, відповідних вузлів, номери яких є присутніми в термах, отриманих після перетворень виразу $\sum \Pi$ на кроці 6. По отриманим B_n *n*-ті вузли визначають стан (справність) *k*-го вузла за наступними правилами:

1) якщо матриця B_n містить ідентичні елементи, то *k*-й вузол справний (*DIAGNOZ*=0);

2) якщо матрица B_n містить неідентичні елементи, то *k*-й вузол несправний (*DIAGNOZ*=0);

3) якщо частина матриці B_n містить неідентичні елементи, а частина – ідентичні, то справність *k*-го вузла не визначена і *DIAGNOZ*=1.

Крок 8. Кінець алгоритму.

Якщо після закінчення алгоритму *DIAGNOZ*=1, то діагностування не виконане успішно, що служить сигналом у верхній рівень ієрархії про неспроможність Алгоритму А1. Якщо *DIAGNOZ*=0, то діагностування виконане успішно.

Для обґрунтування коректності алгоритму введемо наступні визначення і положення.

Означення 1. Якщо в матриці A_n вузла n ($n = 1, ..., N-1$) є помічений елемент a_{ij}^{n*} ($i \vee j$) (тобто на перетині рядка *i* із стовпцем *j*) (*i, j* ∈ {1,2,...,2t+1}), то вважатимемо, що вузли *i* і *j* взаємно «оточують по несправності» один одного, а сам елемент a_{ij}^{n*} ($i \vee j$) називатимемо елементом «оточення по несправності» вузлів *i* і *j*.

Означення 2. Сукупність усіх елементів «оточення по несправності» вузла *i* для даної A_n називатимемо «несправним оточенням» вузла *i* і позначимо через $ENVIR\ (i)_n$.

Слід також зауважити, що вузол i є несправним, якщо він на кроках 2 або 3 передає різні повідомлення справним вузлам. Якщо несправність проявляється ще на кроці 2.

Для подальшого викладу присвоїмо повідомленням наступну семантику:

$a \equiv$ FF (немає несправності, самодіагностування пройшло успішно);

$\overline{a} \equiv$ FAIL (виявлена несправність);

$\varnothing \equiv$ FAIL.

При цих припущеннях алгоритм А1 призводить до наступного:

1. Якщо в результаті самодіагностування вузол n виявив свою несправність і в змозі проінформувати інші вузли шляхом передачі повідомлення FAIL (у разі комунікаційної несправності теж буде передано FAIL, в силу семантичного визначення $\varnothing$), то алгоритм А1 вирішує задачу діагностування. Тут крок 1 еквівалентний проведенню самодіагностування кожного вузла. Після кроку 3 всі A_n міститимуть вичерпну інформацію про діагностування інформаційної системи: розряди PRS_n, що містять FF, відповідатимуть справним вузлам, а ті, які містять FAIL – несправним.

2. Якщо в результаті виконання алгоритму А1 справний вузол приходить до неоднозначного рішення задачі діагностування, то це свідчить про більше число вузлів, що відмовили, або про наявність складних прихованих відмов з якими алгоритм А1 не в змозі впоратися.

Введемо в розгляд визначення деяких властивостей тестів.

Означення 3. Абсолютно надійними називатимемо тести, що не містять програмних помилок, тобто які поводяться однаково, незалежно від початкових даних, до яких вони застосовуються.

Означення 4. Строго синхронними називатимемо тести, час обробки яких в кожному вузлі відомий усім вузлам інформаційної системи.

У разі, коли алгоритм А1 призводить до неєдиного рішення задачі діагностування, то вузол n переходить до виконання іншого алгоритму, який дістав назву алгоритм А2.

Розробка алгоритму діагностування бездротових сенсорних мереж з прихованими відмовами. Алгоритм складається з декількох етапів, кожен з яких може включати декілька фаз (рис. 1). Передбачається, що існує механізм, який призначає для кожного етапу вузол, що здійснює діагностування, який далі називатимемо «тестером». Впродовж етапу тестер не міняється. Алгоритм пропонує дії тестера і дії вузлів, що не є тестерами для кожного етапу.

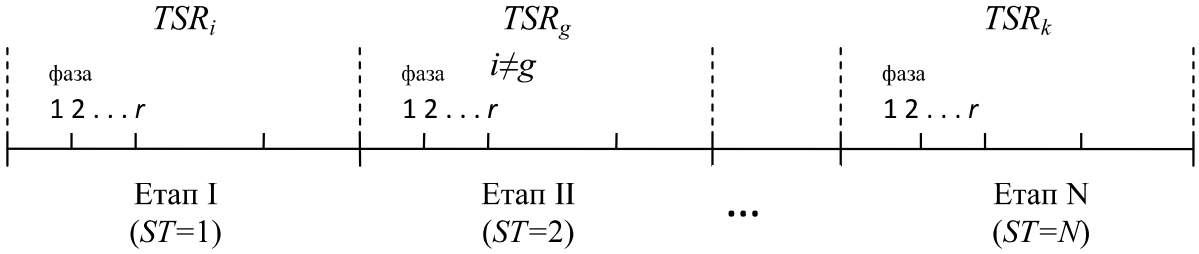

Рис 1. Часова діаграма діагностування

Кожна фаза етапу включає наступні кроки:

Крок 1. Після вступу сигналу з нижнього рівня інформаційної системи про неспроможність алгоритму А1 (тобто про його неоднозначне рішення про стан вузлів) тестер TSR_i (i-й вузол) посилає іншим j-м вузлу ($j \neq i$) повідомлення, які залежно від однозначності визначення стану інформаційної системи можуть бути:

1а: «Я ВИЗНАЧИВ НЕСПРАВНІ ВУЗОЛ». Посилається у разі, якщо тестер в результаті виконання алгоритму А1 або попередніх етапів або фаз справжнього етапу алгоритму А2 визначив однозначно несправні вузли. Перехід до п.6.

1б: «ВИКОНАТИ ТЕСТ T_i^r». Вказується заданий деяким чином тест який слід виконати у фазі *r*. При цьому передбачається, що тести, вузли, що задаються, *i*, вже доступні усім *j*-м вузлам. При цьому тестеру (і-й вузол) відомо, який час потрібний кожній з *j*-х вузлів ($j \neq i$) для обробки чергового тесту (строго синхронні тести). Позначимо цей час через $t_T^r(j) = \text{DELAY } T_i^r(j)$.

Після закінчення часу $t(j) = \text{DELAY } T_i^r(j)$ *i*-й вузол направляє *j*-му вузлу повідомлення: «ВИКОНАТИ ТЕСТ T_i^{r+1}» і так далі.

Крок 2. Обмін інформацією між *j*-ми вузлами ($j \neq i$) про отримане завдання від *i*-го вузла (тестера) виконати ТЕСТ T_i^r.

Крок 3. *j*-і вузли обробляють ТЕСТ $T_i^r(j)$.

Крок 4. *j*-і вузли передають результати тестування $REZ\ T_i^r(j)$ *i*-у вузлу (тестеру) і повідомляють про них один одного. При цьому, *j*-й вузол, отримавши від *k*-го вузла результат тестування, передає його номер *k* іншим *l*-м вузлам ($l \neq k,\ l \neq j,\ l \neq i$), а отримане повідомлення тестеру. Кожний *j*-й вузол накопичує історію етапів (для кожного етапу *ST* матриця *G*(*r,n*), в якій фіксується номер фази *r* етапу *ST*, номер вузла *n* і момент отримання результату тестування від цього вузла. Ведення цієї історії триває до настання власного етапу, перед початком якого обробляється історія попередніх етапів. Після завершення власного етапу знову ведеться історія етапів, яка обробляється після завершення усіх етапів, тобто після завершення усього діагностування.

Крок 5. Тестер обробляє отримані повідомлення.

5а: Якщо впродовж часу

$$t^r(j) = t_T^r(j) + t_\Pi^r(i,j) = DELAY\,T_i^r(j) + (N-1)DELAY\_MES(i,j)$$

тестер не отримає ніякого повідомлення від *j*-го вузла, те рішення про стан даної «мовчазного» вузла відкладається до завершення усіх етапів. Якщо і після закінчення останнього етапу стан *j*-й вузол ще не визначений, то він вважається таким, що відмовив.

$t_T^r(j)$ – час витрачається *j*-м вузлом на обробку тесту T_i^r отриманого від тестера *i* в *r*-й фазі; $t_\Pi^r(i,j)$ – час витрачається тестером *i* на отримання повідомлення *MES*(*i, j*) від *j*-го вузла.

5 б: Якщо на протязі $t^r(j)$ тестер отримає повідомлення у вигляді:

$$\{REZ\ T_i^r(j),\ REZ\ T_i^r(j,l) \mid l \neq i,\ l \neq j\},$$

то перевіряється наступне:

– правильність результатів виконання тесту

$$\{REZ\ T_i^r(j),\ REZ\ T_i^r(j,l) \mid l=1...N,\ l \neq i,\ l \neq j\} = REZ\ T_{етал}^r;$$

– час вступу відповіді

$$t_T^r(j) = t_{етал}.$$

$REZ\ T_i^r(j)$ – результат обробки *j*-м вузлом тесту, заданого у фазі *r* тестером *i* поточного етапу діагностування, а вираз $REZ\ T_i^r(j,\ l)$ – повідомлення про результати обробки *l*-ми вузлом тесту, заданого у фазі *r* тестером і поточного етапу діагностування, отримані *j*-м вузлом від *l*-х вузлів у фазі *r* і передані тестеру.

Мають місце жорсткі часові обмеження, визначені строго фіксованим часом обробки тесту і відомим часом, що витрачається на обробку кожного отриманого повідомлення.

Якщо результат обробки тесту неправильний або час вступу не відповідає тимчасовому ліміту, то вузол вважається таким, що відмовив і виключається з розгляду.

Крок 6. Після закінчення заздалегідь визначеного числа фаз поточного етапу *i*- й вузол після невеликої затримки, необхідної у разі різної тривалості обробки тестів на різних вузлах, передає повідомлення: «МІЙ ЕТАП END». Після чого призначається новий тестер етапу TSR_g(*g*-й вузол) ($g \neq i$) і відбувається повторення кроків 1-6.

Кінець алгоритму.

Слід зауважити, що *i*-й вузол може закінчити свій етап раніше, не чекаючи завершення обробки усіх передбачених на цьому етапі фаз, якщо прийде до однозначного рішення задачі діагностування інформаційної системи (див. крок 1а). У такому разі на завершення етапу ініціюється повідомлення: «Я ВИЗНАЧИВ НЕСПРАВНІ ВУЗЛИ».

Доведемо коректність алгоритму А2. Спершу відмітимо, що якщо останньою фазою кожного етапу буде самодіагностування, то усі відмови, що знову з'явилися, будуть відомі в системі. Якщо вузол має деяку функціональну стійкість, то про появу відмов, з якими вузол може сам впоратися, не повідомлятиметься іншому вузлу. Таким чином, у кінці кожного етапу нові відмови, що з'явилися впродовж етапу, будуть виявлені усіма справними вузлами інформаційної системи. Залишаються вузли, які навмисно поводилися так, щоб заплутати справні вузли при спробі виявити відмови. Алгоритм А2 є коректним в тому сенсі, що вузли з прихованими відмовами виявляються кожним справним вузлом.

Можна виділити наступні три випадки:

1. Якщо *j*-й вузол несправний, то він передасть суперечливу інформацію хоч б один раз впродовж усіх етапів діагностування. Інакше при виконанні алгоритму А2 станеться збій. Якщо несправність *j*-й вузол проявився, то при обробці наступного тесту, він запізниться на $\Delta = t_T^r(j) - t_{Tетал}^r$ і тестер виявить це. Передача ж другого повідомлення в поточній фазі буде зафіксована в історії етапів тих вузлів, яким це повідомлення було спрямоване.

А починаючи з наступної фази вузол, що запізнився один раз, вже не в змозі «надолужити» і він знову і знову проявлятиме себе як нелояльний.

Існує і друга можливість – передача невірного результату чергового тесту. Тоді по неспівпадінню вузла, що передав невірний результат, оголошується несправним.

2. Несправність *j*-го вузла може проявлятися і в приховуванні повідомлення, отриманого від справного вузла. Але в цьому випадку буде невідповідність між часом, витраченим на отримання результату тесту T_i^r і обробку повідомлень, які передаються тестеру і лімітом, який на це відводиться. Згідно допущення 6, якщо *j*-й вузол посилає *i*-му вузлу повідомлення, то *i*-й вузол витратить на його отримання час $DELAY\_MES^r(i,j)$, тобто несправний вузол знову запізниться і це дозволить усім вузлам виявити її несправність.

3. Особливий інтерес викликає випадок, коли вузол «мовчить», тобто передає порожнє повідомлення ($\varnothing$). Це не призводить до втрати часу, на чому ґрунтується суть алгоритму А2 і що відрізняє його від алгоритму А1 і від інших підходів, наприклад [8]. Якщо упродовж усіх етапів *i*-й вузол не отримав жодного повідомлення від *j*-го вузла, то після завершення останнього етапу *i*-й вузол оголошує *j*-й вузол несправній. При цьому, якщо *j*-й вузол дійсно несправний, то це стало відомо усім, кому він передавав свої повідомлення (згідно допущення 6 він спізнюватиметься з відповіддю, якщо укриватиме, що отримує повідомлення від *i*-го або ж його несправність проявиться при спробі передати суперечливі повідомлення).

Якщо ж *j*-й вузол справний, але зник комунікаційний шлях між двома вузлами, то матиме місце ситуація, коли кожний із справних вузлів рахуватиме свого сусіда по цьому комунікаційному шляху несправним. Вважатимемо, що ця ситуація допустима і, що її можна дозволити за рахунок додаткових витрат часу або використанням механізмів нижчих рівнів, що дозволяють виявити несправність комунікаційного шляху.

Тривалість алгоритму А2 визначається наступною теоремою.

Теорема. Якщо несправність *j*-го вузла уперше проявилася під час обробки тесту T_i^r, то про неї буде відомо усім справним вузлам інформаційної системи не пізніше, ніж в (*r*+1)-ій фазі етапу *i*.

Доведення. Розглянемо окремі прояви несправності *j*-го вузла.

1) Видача більше одного результату на тест T_i^r. Тестер отримає повідомлення про результати обробки тесту від *j*-го вузла через час $t^r(j) > t^r_{Тетал}$ оскільки на *j*-й вузол буде витрачено часу більше, ніж на отримання повідомлень від усіх вузлів і передачу одного результату виконання тесту. Таким чином, в роботі *j*-го вузла виходить запізнення, яке буде виявлено усіма вузлами, які отримують повідомлення про результати обробки тесту T_i^r *j*-м с вузлом.

2) Замовчування про отримання якого-небудь повідомлення. Очевидно, що з'явиться невідповідність між фактично витраченим часом на передачу свого повідомлення і отримання повідомлень від інших вузлів і лімітом, який відводиться на передачу одного результату і на отримання тієї кількості результатів, яка вказується в повідомленні поточної фази. Кожен результат, про який замовчується, збільшує фактично витрачений час на $DELAY\_MES^r(i,j)$. Це запізнення з'явиться в (r+1)-ій фазі при видачі результату тесту цієї фази і буде кратним $DELAY\_MES^r(i, j)$, що і потрібно було довести.

Висновки. Таким чином, відповідно до ієрархічної концепції організації засобів забезпечення функціональної стійкості інформаційної системи підприємства, розроблено два алгоритми, які утворюють дворівневу систему діагностування прихованих відмов. Діагностування розпочинається з виконання алгоритму А1, переваги якого в порівнянні з відомими алгоритмами полягають в тому, що він вимагає меншої надмірності системи, всього два раунди обміну повідомленнями між вузлами інформаційної системи і забезпечує діагностування інформаційної системи підприємства при відмові майже половини її вузлів.

У разі неоднозначного рішення задачі діагностування, алгоритм А1 формує сигнал про свою неспроможність і діагностування інформаційної системи триває по алгоритму А2, який в якості критерію використовує тривалість виконання фаз.

Список використаної літератури

1. V.A. Mashkov, O.V. Barabash "Self-checking and Self-diagnosis of Module Systems on the Principle of Walking Diagnostic Kernel" Engineering Simulation. – Amsterdam: OPA, 1998. Vol. 15. pp. 43-51.

2. Саланда І. П. Система показників та критеріїв формалізації процесів забезпечення локальної функціональної стійкості розгалужених інформаційних мереж / І. П. Саланда, О. В. Барабаш, А. П. Мусієнко // Системи управління, навігації та зв'язку». – 2017. – Вип. 1 (41). – С. 122-126.

3. Мусієнко А.П. Методи пошуку оптимальних маршрутів графа структури розгалуженої інформаційної мережі за заданим критерієм оптимальності при різних обмеженнях / І.П. Саланда, О.В. Барабаш, А.П. Мусієнко // Наукові записки Українського науково-дослідного інституту зв'язку. – К.: УНДІЗ, 2016. – №2 (42). – С. 99 – 106.

4. N. Pashynska, V. Snytyuk, V. Putrenko, A. Musienko "A decision tree in a classification of fire hazard factors ", Eastern-European Journal of Enterprise Technologies. – Kharkov, 2016. – № 5/10(83). – P. 32-37.

5. Мусієнко А.П. Забезпечення функціональної стійкості інформаційних мереж на основі розробки методу протидії DDoS-атакам / О. В. Барабаш, Н. В. Лукова-Чуйко, А. П. Мусієнко, В. В. Собчук // Сучасні інформаційні системи. – Харків: НТУ «ХПІ», 2018. – Т 2. – № 1. – С. 56-64.

6. Мусієнко А.П. Аналіз застосування мереж Петрі для підтримки функціональної стійкості інформаційних систем / О.В. Барабаш, Н.В. Лукова-Чуйко, А.П. Мусієнко, О.Ю. Ільїн // Телекомунікаційні та інформаційні технології. – 2018. – № 1 (58). – С. 11-18.

7. Musienko A.P. Diagnostic model of wireless sensor network based on the random test of checks / A.P. Musienko, O.V. Barabash, N.V. Lukova-Chuiko, I.P. Salanda // Science and Education a New Dimension. Natural and Technical Sciences, 2018. – VI (18), Issue 158, Budapest, Hungary, pp. 25-28.

8. Musienko A. Information Technology of Targeting: Optimization of Decision Making Process in a Competitive Environment / O. Barabash, G. Shevchenko, N. Dakhno, O. Neshcheret, A. Musienko // International Journal of Intelligent Systems and Applications. – Vol. 9. – № 12. – Hong Kong: MECS Publisher, 2017. – P. 1-9.

References

1. V.A. Mashkov, O.V. Barabash. "Self-checking and Self-diagnosis of Module Systems on the Principle of Walking Diagnostic Kernel Engineering Simulation." *Amsterdam: OPA* 15. (1998): 43-51. Print

2. I.P. Salanda, O.V. Barabash, A.P. Musiienko "The system of indicators and criteria for formalizing the processes of ensuring the local functional stability of the branched information networks" *Systemy upravlinnia, navihatsii ta zviazku* 1(41) (2017): 122-126. Print

3. I.P. Salanda, O.V. Barabash, A.P Musienko "Methods of searching for optimal routes of the graph of the structure of the branched information network by the given optimality criterion under different constraints." *Scientific notes of the Ukrainian Research Institute of Communication,* Kiev, 2 (42) (2016): 99-106. Print

4. N. Pashynska, V. Snytyuk, V. Putrenko, A. Musienko "A decision tree in a classification of fire hazard factors ", *Eastern-European Journal of Enterprise Technologies,* Kharkov 5/10(83) (2016): 32-37. Print

5. O.V. Barabash, N.V. Lukov-Chuiko, A.P. Musienko, V.V. Sobchuk "Providing of functional stability of information networks on the basis of development of a method of counteraction to DDoS-attacks." *Modern information systems*, Kharkiv: NTU "KPI", Vol.2, 1 (2018): 56-64. Print

6. O.V. Barabash, N.V. Lukova-Chuiko, A.P Musienko, O.Yu. Ilyin "Analysis of the use of Petri Networks to support the functional stability of information systems. " *Telecommunication and information technologies* 1 (58) (2018): 11-18. Print

7. A.P. Musienko, O.V. Barabash, N.V. Lukova-Chuiko, I.P. Salanda "Diagnostic model of wireless sensor network based on the random test of checks." – *Science and Education a New Dimension. Natural and Technical Sciences*, Budapest, Hungary. Issue 158, VI (18) (2018): 25-28. Print

8. O. Barabash, G. Shevchenko, N. Dakhno, O. Neshcheret, A. Musienko "Information Technology of Targeting: Optimization of Decision Making Process in a Competitive Environment" *International Journal of Intelligent Systems and Applications*. Vol.9, Hong Kong: MECS Publisher 12, (2017): 1-9. Print

Автори статті (Authors of the article)

Собчук Валентин Володимирович – к. ф.-м. н., доцент, доцент кафедри диференціальних рівнянь і математичної фізики, факультету інформаційних систем, фізики та математики (Sobchuk Valentin Volodymyrovych – candidate of physical and mathematical sciences, associate professor, associate professor of the department of differential equations and mathematical physics, faculty of information systems, physics and mathematics) Phone.: +380 (50) 339 81 13. E-mail: v.sobchuk@ugmk.kiev.ua

Коваль Мирослав Олександрович – аспірант Військового інституту (Koval Myroslav Oleksandrovych – postgraduate student at the Military Institute). Phone.: +380638352199. E-mail: teckkill-a@live.com

Мусієнко Андрій Петрович – д.т.н., доцент кафедри вищої математики (Doctor of Technical Sciences, Associate Professor of the Department of Higher Mathematics). Phone.: +380 (95) 315 69 17. E-mail: mysienkoandrey@gmail.com

Мацько Олександр Йосипович – к.військ.н., професор, начальник інституту Національного університету оборони України ім. І. Черняховського (Matsko Oleksander – Candidate of Military Sciences, Professor, Chief of Institute of Ivan CherniahovskyNational Defense University of Ukraine).

Рецензент: доктор техн. наук, професор **В. В. Вишнівський**, Державний університет телекомунікацій, Київ.