

ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ ПОВІТРЯНИМ РУХОМ НА ОСНОВІ ВЗАЄМНОГО ІНФОРМАЦІЙНОГО УЗГОДЖЕННЯ

У статті пропонується методика діагностування несправностей у автоматизованій системі управління повітряним рухом на основі взаємного інформаційного узгодження між елементами системи.

Постановка проблеми. В умовах постійного підвищення інтенсивності повітряного руху на тлі існуючих вимог щодо якості наземного радіолокаційного забезпечення польотів повітряних суден особливого значення набуває надійність автоматизованих систем управління повітряним рухом (АСУПР). Забезпечення надійності таких систем здійснюється на основі застосування принципів відмовостійкості, тобто правильного виконання алгоритмів і завдань при виникненні несправностей. Реалізація принципів відмовостійкості передбачає своєчасне виявлення виникаючих несправностей і швидке відновлення АСУПР. Завдання виявлення несправностей АСУПР покладається на систему її діагностування.

Розподілені АСУПР характеризуються не тільки розосередженням апаратури безпосередньо в місцях обробки даних, але й розділенням усіх функцій обробки та управління на низку слабозв'язаних, їх поділом між обчислювачами (ЕОМ). Скоординована робота ЕОМ в розподіленій АСУПР можлива тільки при здійсненні між ними інформаційного узгодження для підтримки служби єдиного часу, забезпечення цілісності розподіленої бази даних, синхронізації розподілених обчислювальних процесів та ін.

Існуючі методи забезпечення взаємного інформаційного узгодження (ВІУ) ґрунтуються на ідентичності прийнятих справними обчислювачами повідомлень при заданій допустимій кількості несправних ЕОМ. Такий підхід відповідає методу маскування несправностей. Однак він є недостатнім для послідовного застосування ідеї відмовостійкості як інтегрованого комплексу механізмів, який включає і діагностику. Отже, необхідним є узгоджене виявлення та ідентифікація виявлених несправностей, яка полягає у визначенні їх місця і типу (збій або відмова). Це дозволить здійснити реконфігурацію і подальше відновлення обчислювального процесу розподіленої АСУПР.

Огляд останніх досліджень. У розподілених АСУПР, на відміну від централізованих обчислювальних систем в цілому, з'являються відмови особливого виду – «ворожі» [1, 2]. Компоненти, що відмовили, при взаємодії передають іншим компонентам суперечливу інформацію про свої дії та стан. При цьому вони не можуть бути виявлені звичайними діагностичними процедурами. При таких відмовах непридатною є модель із глобальним спостерігачем [3, 4].

При розробці засобів забезпечення відмовостійкості розподілених АСУПР доцільно застосувати ієрархічний підхід [5], який широко використовується при аналізі й синтезі обчислювальних систем.

Формулювання завдання дослідження. Метою даної статті є розробка методики діагностування розподіленої обчислювальної системи АСУПР на основі взаємного інформаційного узгодження для забезпечення її відмовостійкості.

Постановка завдання. Розглянемо розподілену систему, що складається з $N \geq 2t + 2$ ЕОМ (вузлів), у якій виникає не більше t відмов. Нехай довільний вузол k ($k=1...N$) є «генералом». Номер «генерала» у системі змінюється при кожному новому запуску процедури перевірки за певним правилом, закладеним у програму її функціонування. Кожному з решти вузлів («лейтенантам») вузол k повинен передати відповідне повідомлення (терміни «генерал» і «лейтенант» запозичені з роботи [6]). Після цього «лейтенанти» обмінюються повідомленнями про отриману від «генерала» інформацію, при цьому кожен формує початковий набір інформації (чисел), за яким визначаються «нелояльні лейтенанти».

Нелояльність («ворожість») обчислювача полягає в довільному спотворенні інформації, яка передається. При цьому в різних повідомленнях нелояльний «лейтенант» передає суперечливі повідомлення іншим «лейтенантам». Якщо діагностика розподіленої обчислювальної системи (ОС) приводить до однозначного визначення нелояльних (несправних) вузлів, то угода між лояльними «лейтенантами» і лояльним «генералом» буде досягнута.

Завдання полягає у розробці алгоритмів обміну інформацією між елементами системи, за якою можна було б однозначно визначити несправні вузли.

Виклад основного матеріалу досліджень. Складовими початкового набору є повідомлення з множини $\{a, \bar{a}, \emptyset\}$, які можуть визначати справний, несправний та невизначений стани вузла системи відповідно. Припустимо, що $a=0$, $\bar{a}=1$ і кожному вузлу в системі привласнений індивідуальний номер $n=1...N$, який відомий усім вузлам системи.

Початковий набір, що формується в n -му вузлі, має вид матриці:

$$A_n = \begin{bmatrix} a_{11}^n & \cdots & a_{1j}^n & \cdots & a_{1L}^n \\ a_{21}^n & \cdots & a_{2j}^n & \cdots & a_{2L}^n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i1}^n & \cdots & a_{ij}^n & \cdots & a_{iL}^n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{L1}^n & \cdots & a_{Lj}^n & \cdots & a_{LL}^n \end{bmatrix},$$

де $L=2t+1$ – кількість вузлів, які беруть участь в обміні повідомленнями (усі, крім «генерала»);

у i -му рядку містяться значення повідомлень усіх j -х вузлів, прийняті в першому раунді i -м вузлом і передані ним в другому раунді в n -й вузол;

у j -му стовпці містяться значення повідомлень j -го вузла, передані ним усім i -м вузлам;

на головній діагоналі розташовані елементи, що є значеннями повідомлень j -х вузлів, передані ними самим собі в першому раунді.

В алгоритмі пошуку несправних вузлів, який пропонується, використовується функція *majority*, яку можна визначити таким чином [6]. Нехай задано вектор $e = \{e_i\}$ ($i=1,2,\dots,n$), елементи якого групуються за ознакою тотожності їх значень у групи: M_1, M_2, \dots, M_m , які не перетинаються (де m – кількість таких груп). Очевидно, що $m \leq n$. Нехай $|M_1| > |M_2| > \dots > |M_m|$, де через $|M|$ позначена потужність множини M , тобто кількість елементів множини. Якщо елементи $\{e_i\}$, що входять в M_1 , мають значення $v \in \{a, \bar{a}, \emptyset\}$, то функція *majority* визначається виразом $maj \{e_i\} = v$.

Якщо $|M_1| = |M_2| = \dots = |M_m|$, то значення функції *majority* визначається випадковим вибором індексу групи серед груп найбільшого розміру.

Діагностика починається з виконання алгоритму А1, переваги якого в порівнянні з відомими алгоритмами ВГУ полягають в тому, що він вимагає меншої надмірності системи (усього два раунди обміну повідомленнями між вузлами розподіленої АСУПР) і забезпечує її діагностування при відмові майже половини вузлів. При однозначному визначенні вузлів, що відмовили, алгоритм дозволяє закінчити діагностику з мінімальними часовими витратами.

Алгоритм А1 діагностики розподіленої АСУПР на основі алгоритму взаємного інформаційного узгодження

Розглянемо розподілену АСУПР, яка складається з N вузлів (ЕОМ) з номерами $1 \dots N$. ВГУ здійснюється шляхом взаємообміну повідомленнями при припущеннях про її синхронність і можливість отримувачем повідомлення визначити його відправника.

Алгоритм А1 складається з двох етапів: етапу пересилок повідомлень (кроки 1–3) і етапу аналізу отриманих повідомлень (кроки 4–7), який виконується у кожному вузлі автономно на основі інформації, отриманої ним на попередньому етапі.

Крок 1. k -й вузол «генерал» надсилає решті n -их вузлів ($n \neq k$) повідомлення Z_k з множини $Z = \{a, \bar{a}, \emptyset\}$. Для зручності вважатимемо номер k рівним $2t+2$.

Крок 2. Решта n -х вузлів обмінюються повідомленнями, отриманими від k -го вузла на кроці 1. При цьому вузли, у яких має місце «ворожа» відмова, можуть передавати суперечливі повідомлення іншим вузлам.

Кожен n -й вузол ($n=1 \dots N-1$; $n \neq k$) формує з отриманих повідомлень вектор $STR(n)$, що містить $2t+1$ елемент, для розміщення інформації, отриманої від усіх обчислювачів, включаючи і себе на даному кроці: $STR(n) = (Z_1, Z_2, \dots, Z_{N-1})$.

Крок 3. Усі вузли (крім «генерала») обмінюються векторами $STR(n)$ ($n=1 \dots N-1$), з яких кожна формує початковий набір у вигляді матриці A_n , у котрій вектори $STR(n)$ ($n=1 \dots N-1$) розташовуються як рядки в порядку зростання номерів вузлів-відправників, які можна завжди однозначно визначити.

Крок 4. Кожен 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 $PRS_n(j) = majority\{a_{ij}^n \mid i,j=1 \dots N\}$.

Крок 5. Кожен n -й вузол визначає елементи a_{ij}^{n*} , $i,j=1 \dots N-1$ власної матриці A_n , що знаходяться на перетині стовпця j з рядком i , для яких

$$PRSn(j) \neq a_{ij}^{n*}. \tag{1}$$

Позначимо загальну кількість виявлених елементів через L_{max} . Якщо $L_{max} = 0$, то перейти до кроку 7.

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

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

де i, j – номери вузлів з елементів матриці, що відповідають номеру рядка та стовпця j елементу матриці A_n , що відрізняється від $PRSn(j)$.

Далі вираз (2) приводиться до диз'юнктивної нормальної форми та перетворюється до виду диз'юнкції кон'юнкцій. При цьому для урахування обмеження щодо наявності не більше ніж t несправних вузлів виключаються з розгляду терми з більш ніж t елементами. Кожен з термів, що залишилися, визначає припустиме поєднання несправностей, які можуть привести до всіх виявлених несправностей за умови, що їх у системі є не більш ніж t . Якщо у виразі $\sum \Pi$ залишається більшою за один терм, то результат діагностування неоднозначний і формується ознака $DIAGNOZ=1$, інакше $DIAGNOZ=0$.

Крок 7. У кожному n -му вузлі формується матриця B_n , отримана з A_n шляхом викреслювання рядків і стовпців, номери вузлів яких присутні в термах, отриманих після перетворень виразу $\sum \Pi$ на кроці 6. За отриманими B_n n -і вузли визначають стан (справність) k -ого вузла таким чином:

- 1) якщо матриця B_n містить ідентичні елементи, то k -й вузол справний ($DIAGNOZ=0$);
- 2) якщо матриця B_n містить неідентичні елементи, то k -й вузол несправний ($DIAGNOZ=0$);
- 3) якщо частина матриць B_n містить неідентичні елементи, а частина – ідентичні, то справність k -ого вузла не визначена і формується $DIAGNOZ=1$.

Крок 8. Кінець алгоритму.

Аналіз алгоритму А1 показує, що він може привести до неоднозначного вирішення задачі діагностики.

У такому випадку алгоритм А1 формує сигнал про свою неспроможність і діагностика розподіленої АСУП продовжується за алгоритмом А2, який за критерій використовує тривалість виконання фаз алгоритму.

Алгоритм А2 діагностики розподілених АСУП з «ворожими» відмовами складається з певних етапів, кожен з яких може включати декілька фаз (рис.1). Передбачається, що існує механізм, який призначає для кожного етапу вузол (ЕОМ), який здійснює діагностику, далі називатимемо його тестером. Протягом етапу тестер не змінюється. Алгоритм визначає дії тестера і решти вузлів для кожного етапу.

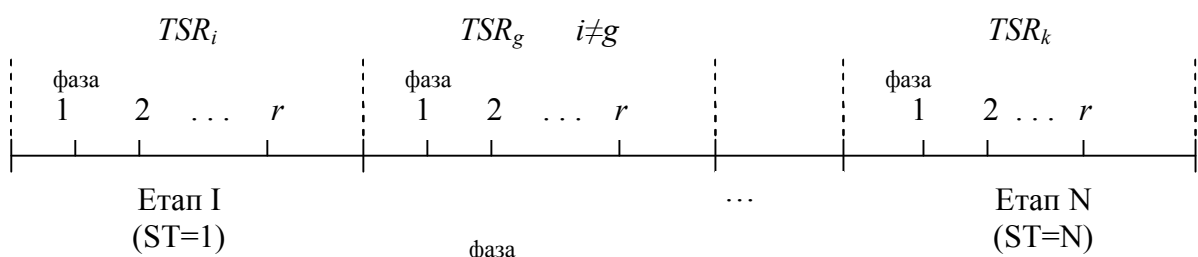


Рис. 1

Кожна фаза етапу включає окремі кроки.

Крок 1. Після надходження сигналу з нижнього рівня засобів забезпечення відмовостійкості про неоднозначне рішення алгоритму А1 щодо стану обчислювальної системи тестер TSR_i (i -й вузол) надсилає решті j -х вузлів ($j \neq i$) повідомлення, які залежно від однозначності визначення стану АСУПР можуть бути:

1а – «Я ВИЗНАЧИВ НЕСПРАВНІ ЕОМ». Посилається у випадку, якщо тестер в результаті виконання алгоритму А1 або попередніх етапів чи фаз поточного етапу алгоритму А2 однозначно визначив несправні вузли. Перехід до кроку б алгоритму А2.

1б – «ВИКОНАТИ ТЕСТ T_i^r ». Призначається тест з множини внутрішніх тестів системи, який слід виконати у фазі r . При цьому передбачається, що тести для i -х вузлів, що задаються, вже доступні всім j -м вузлам. Крім того, тестеру відомо, який час необхідний кожному j -му вузлу ($j \neq i$) для обробки чергового тесту (строго синхронного).

Позначимо цей час через $t_T^r(j) = DELAY(j)T_i^r$.

Після закінчення часу $t(j) = DELAY(j)$ тестер направляє j -м вузлам повідомлення: «ВИКОНАТИ ТЕСТ T_i^{r+1} » і т.д.

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. Усі j -і вузли обробляють ТЕСТ (j) T_i^r .

Крок 4. Усі j -і вузли передають результати тестування $REZ(j) T_i^r$ 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 -им вузлам ($l \neq k, l \neq j, l \neq i$), а отримане повідомлення – тестеру. Кожен j -й вузол накопичує історію етапів (для кожного етапу ST матриця $G(r,n)$, у якій фіксується номер фази r етапу ST , номер вузла n і момент отримання результату тестування від нього). Ведення цієї історії триває до настання власного етапу, перед початком якого обробляються дані попередніх етапів. Після завершення власного етапу знову ведеться історія, яка обробляється по закінченню всієї діагностики.

Крок 5. Тестер обробляє отримані повідомлення.

5а: якщо протягом часу

$$t^r(j) = t_T^r(j) + t_{\Pi}^r(i, j) = DELAY T_i^r(j) + (N - 1) DELAY_MES(i, j) \quad (3)$$

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

У (3) позначено: $t_T^r(j)$ – час, який витрачається j -м вузлом на обробку тесту, отриманого від тестера i в r -й фазі; $t_{\Pi}^r(i, j)$ – час, який витрачається тестером на отримання повідомлення $MES(i, j)$ від j -го вузла.

5б: якщо протягом $t^r(j)$ тестер отримає повідомлення у вигляді

$$\{REZ(j)T_i^r, REZ T_i^r(j, l) \mid l \neq i, l \neq j\}, \quad (4)$$

то перевіряється:

– правильність результатів виконання тесту:

$$\{REZ(j)T_i^r, REZ T_i^r(j, l) \mid l = 1 \dots N, l \neq i, l \neq j\} = REZ T_{емал}^r; \quad (5)$$

– час надходження відповіді:

$$t_T^r(j) = t_{емал}. \quad (6)$$

У (4-6) позначено: $REZ(j)T_i^r$ – результат обробки j -м вузлом тесту, заданого у фазі r тестером i поточного етапу діагностики, а вираз $REZ(j)T_i^r(j, l)$ – повідомлення про результати обробки l -ми вузлами тесту, заданого у фазі r тестером i поточного етапу діагностики, отримані j -ми вузлами від l -х вузлів у фазі r і передані тестеру.

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

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

Крок 6. По закінченню певної кількості фаз поточного етапу i -й вузол після незначної 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.

Аналіз методики діагностування на основі алгоритмів А1 і А2

Запропонована методика діагностування розподілених АСУПР забезпечує діагностування несправностей при будь-якій формі їх прояву, які можуть виникнути в обчислювальній системі. У процесі виконання процедури діагностування за даною методикою накопичується інформація про форму прояву несправності, достатня для того, щоб визначити її належність до однієї з трьох груп. Кожній групі відповідає певний алгоритм її діагностування. Таким чином, процедура містить елементи самонавчання.

Достовірність результатів теоретичних досліджень підтверджується математичним моделюванням процедури діагностування розподілених обчислювальних систем, що складаються з $N = 7, 8, 9$ обчислювачів.

Мета експериментальних досліджень передбачає:

– аналіз правильності функціонування алгоритму А1 діагностування розподілених АСУПР з «ворожими» відмовами;

– оцінку показників діагностування: достовірності (ймовірності безпомилкового визначення вузлів, які відмовили) та часу діагностування.

У процесі моделювання за допомогою генератора випадкових чисел задавався технічний стан обчислювальної системи $SZ(N)$ для таких її параметрів: $t = 1 \dots (N-2)/2$, де t – кількість вузлів, що відмовили, N – кількість усіх вузлів системи. Після чого імітувалася робота алгоритму А1, за яким визначався технічний стан системи $SN(N)$. Якщо в результаті виконання алгоритму А1 усі справні вузли було узгоджено і однозначно визначено несправні, то діагностика розподіленої ОС з «ворожими» відмовами завершувалася успішно і $DIAGNOZ=0$, інакше $DIAGNOZ=1$. При $DIAGNOZ=0$ отриманий стан обчислювальної системи $SN(N)$ порівнювався із заданим. За наслідками порівняння

заданого і отриманого розподілу відмов у системі визначалася правильність результатів діагностування. При проведенні великої кількості дослідів (процедур діагностування) за результатами можна статистично визначити достовірність діагностування і середній час діагностування. Результати моделювання зведені в таблицю.

Таблиця
Результати розрахунку достовірності діагностування D , часу діагностування t_0 залежно від кількості вузлів, що відмовили, і кількості вузлів N у системі

Кількість вузлів у системі	Показники діагностування	Кількість відмов			
		1	2	3	4
$N=7$	D	0.98	0.97	0.40	-
	t_0, c	4.5	5.8	8.4	-
$N=8$	D	0.99	0.97	0.94	0.40
	t_0, c	7.2	8.9	11.7	15.3
$N=9$	D	0.99	0.98	0.96	0.4
	t_0, c	12.3	15.2	18.6	21.8

Математичне моделювання підтвердило ефективність розробленої дворівневої системи діагностування «ворожих» відмов на основі ВІУ. У результаті моделювання визначено його достовірність, час і трудомісткість.

Висновки

1. Діагностування на основі ВІУ ефективне для діагностування ситуацій з кількістю обчислювачів $t < (N-2)/2$, що відмовили, де N – кількість усіх обчислювачів системи. Максимально досяжна оцінка достовірності при одному модулі, що відмовив, наближається до величини, рівної 0,99. При ситуаціях, коли кількість несправних обчислювачів перевищує допустиму кількість відмов, достовірність діагностування знижується до 0,35...0,4.

2. Для реалізації діагностування на основі ВІУ необхідно, щоб кожен обчислювач мав достатній об'єм пам'яті ($2N \times N$ чарунок) для зберігання проміжних значень.

3. Результати моделювання показали перевагу застосування діагностування на основі взаємного інформаційного узгодження над використанням маскуванню несправностей. Застосування алгоритмів ВІУ з узгодженим виявленням та ідентифікацією несправностей дозволяє покращити показники відмовостійкості АСУПР у 3...5 разів у порівнянні з алгоритмами, що маскують несправності.

СПИСОК ЛІТЕРАТУРИ

1. Никитин А. И. Отказоустойчивость распределенных систем / А. И. Никитин // Управляющие системы и машины. – 1987. – № 6. – С. 77–83.
2. Лобанов А. В. Взаимное информационное согласование с обнаружением и идентификацией враждебных неисправностей в неполносвязных многомашинных вычислительных системах / А. В. Лобанов // Автоматика и телемеханика. – 2003. – № 6. – С. 175–186.

3. Preparata F. P. On the Connection Assignment Problem at Diagnosable Systems / F. P. Preparata, G. Metze, R. J. Chien // IEEE Trans. – 1967. – Vol. EC–16. – P. 848–854.
4. Барабаш О. В. Методология построения функционально устойчивых распределенных информационных систем / О. В. Барабаш. – К. : НАОУ, 2004. – 216 с.
5. Таненбаум Э. Многоуровневая организация ЭВМ / Э. Таненбаум. – М. : Мир, 1979. – 574 с.
6. Lamport L. The Byzantine Generals Problem / L. Lamport, R. Shostak, M. Pease // ACM Trans. on Progr. Lang. and Systems. – 1982. – Vol. 4–3. – P. 382–402.

Подано 07.07.09

О. В. Барабаш, В. А. Савченко, А. В. Чмут

**ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ
АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ВОЗДУШНЫМ
ДВИЖЕНИЕМ НА ОСНОВЕ ВЗАИМНОГО ИНФОРМАЦИОННОГО
СОГЛАСОВАНИЯ**

В статье предложена методика диагностирования неисправностей в автоматизированной системе управления воздушным движением на основе взаимного информационного согласования между элементами системы.

O. V. Barabash, V. A. Savchenko, O. V. Chmut

**THE INFORMATION SECURITY SUPPORT OF AUTOMATIC FLIGHT CONTROL
SYSTEM WITH MUTUAL INFORMATION AGREEMENT**

The article highlits the faults diagnosys methodology in automatic flight control system with mutual information agreement between its elements.