

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна інженерія»

Група: 2БКС-28

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

здобувача освіти денної форми навчання
БКС.28.06.000.КРБ

ГОНЧАРЕНКА
ОЛЕКСАНДРА
ВОЛОДИМИРОВИЧА

м. Одеса
2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Комп'ютерна інженерія»

Група: 2БКС-28

ПОЯСНЮВАЛЬНА ЗАПИСКА

До кваліфікаційної роботи бакалавра на тему: «Аналіз ефективності
відновлення даних при збереженні на віддалених носіях»

Проектний матеріал складається з пояснювальної записки на 84 сторінках та графічного (презентаційного) матеріалу на 19 аркушах (слайдах)

Виконавець _____ (Гончаренко О.В.)

Керівник проекту _____ (Кривченко Ю.В.)

Консультанти:

з розділу охорони праці та техніки безпеки _____ (Чорновол Н.І.)

з нормоконтролю _____ (Петрашова В.І.)

старший консультант _____ (Кривченко Ю.В.)

До захисту допущений

Завідувач кафедри _____ (Іванова Л.В.)

Завідувач відділення _____ (Скорнякова О.В.)

Захист «26» 06 2024 р. Протокол ЕК № 2

Оцінка ЕК 4 (добре) 850

Секретар ЕК _____

АНОТАЦІЯ

У даній роботі розглянуто проблему виникнення помилок та втрати інформації при її передачі через глобальну мережу та при збереженні на віддалених носіях за використання хмарних технологій.

Виконано аналіз ефективності відновлення даних при збереженні на віддалених носіях з використанням методу відновлення втрачених, пошкоджених або не отриманих вчасно пакетів даних за рахунок формування резервних пакетів, що представляються у вигляді лінійних відновлюючих кодів.

Ефективність методу відновлення даних досліджено шляхом формування резервних пакетів за допомогою реалізованого мовою програмування C++ програмного забезпечення. Створено блок-схему алгоритму, який дозволяє визначити ймовірність відновлення втрачених даних за такими вхідними параметрами: кількість основних, кількість резервних пакетів та кількість помилок, які виникли при передачі.

У програмній моделі реалізовано два підходи: формування резервних пакетів випадковим чином та за допомогою лінійних відновлюючих кодів.

Підтверджено ефективність реалізованого у роботі методу відновлення даних шляхом формування резервних пакетів.

Описано заходи з охорони праці та техніки безпеки.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення Комп'ютерних систем Кафедра Комп'ютерної інженерії
Спеціальність 123 «Комп'ютерна інженерія»
Освітньо-професійна програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР Беркань І.В.

« 16 » 01 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра

здобувачеві освіти Гончаренку Олександр Володимировичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Аналіз ефективності відновлення даних при збереженні на віддалених носіях

затверджена наказом по коледжу від «02» листопада 2023 р. № 244-А2-08

2. Термін здачі студентом кваліфікаційної роботи 13.06.24 р.

3. Вихідні дані до роботи 1. Математична модель лінійних відновлюючих кодів; 2. Алгоритм має визначати ймовірність відновлення втрачених даних за кількістю основних, резервних пакетів та помилок передачі; 3. Алгоритм має передбачати формування резервних пакетів випадковим чином та з використанням лінійних відновлюючих кодів; 4. Реалізувати алгоритм засобами мови C++; 5. Порівняти ймовірності відновлення за допомогою додаткових пакетів, сформованих з використанням лінійних відновлюючих кодів та випадковим чином

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1. Аналіз методів відновлення даних при передачі у мережі

2. Реалізація математичної моделі відновлення даних на базі НЛК



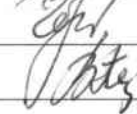
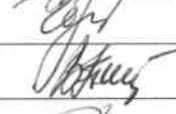
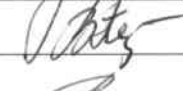


3. Програмна реалізація методу формування надлишкових лінійних кодів

4. Реалізація функціональної схеми та БСА контролю помилок передачі даних

5. Питання охорони праці та техніки безпеки

5. Перелік графічного матеріалу (слайдів мультимедійної презентації) Організація розподіленого віддаленого зберігання інформації; Приклади розщеплення дисків за технологією RAID; Структурна схема системи відновлення даних; Графіки залежностей ймовірності відновлення втрачених пакетів від кількості основних, резервних та втрачених пакетів; Графіки порівняння ймовірностей відновлення втрачених пакетів за допомогою додаткових пакетів, сформованих з використанням лінійних відновлюючих кодів та випадковим чином; Функціональна схема системи контролю помилок передачі даних; БСА контролю та корекції помилок передачі даних



6. Консультанти по кваліфікаційній роботі, із зазначенням розділів, що їх стосуються

Розділ	Консультант	ПІДПИС	
		Завдання видав	Завдання прийняв
Основний розділ	Кривченко Ю.В.		
Розділ охорони праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Кривченко Ю.В.		

7. Дата видачі завдання 15.01.24

Керівник роботи Кривченко Ю.В.

Завдання прийняв до виконання


(підпис)

(підпис)

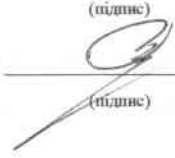
КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Аналіз проблеми забезпечення надійності передачі даних у глобальній мережі	8.05.24р	Виконав
2.	Аналіз технічного завдання та пошук літератури	10.05.24р.	Виконав
3.	Огляд існуючих методів та засобів відновлення даних	13.05.24р.	Виконав
4.	Огляд технологій відновлення даних при їх віддаленому зберіганні	16.05.24р	Виконав
5.	Аналіз математичної моделі відновлення даних	18.05.24р.	Виконав
6.	Формування ефективної системи резервних лінійних кодів	20.05.24р.	Виконав
7.	Оцінка ефективності реалізованого методу	23.06.24р.	Виконав
8.	Опис організації даних та змінних у програмі	27.05.24р.	Виконав
9.	Опис процедур та функцій програми	29.05.24р.	Виконав
10.	Створення функціон. схеми та БСА контролю пакетів	2.06.24р.	Виконав
11.	Розробка ПЗ, тестування, аналіз результатів роботи	5.06.24р.	Виконав
12.	Побудова графіків залежностей	7.06.24р.	Виконав
13.	Розробка питань з охорони праці	9.06.24р.	Виконав
14.	Оформлення слайдів презентації	13.06.24р.	Виконав

Здобувач освіти


(підпис)

Керівник роботи


(підпис)

ЗМІСТ

Вступ.....	6
1 Основний розділ.....	7
1.1 Аналіз проблеми забезпечення надійності передачі даних у глобальній мережі.....	7
1.2 Огляд існуючих методів та засобів відновлення даних	6
1.3 Огляд технологій відновлення даних при їх віддаленому зберіганні.....	6
1.4 Реалізація методу відновлення даних на базі НЛК.....	24
1.5 Формування ефективної системи резервних лінійних кодів	33
1.6 Програмна реалізація методу формування надлишкових лінійних кодів.....	6
1.6.1 Організація даних та змінних у програмі.....	37
1.6.2 Процедури та функції програми.....	38
1.7 Реалізація функціональної схеми та БСА контролю помилок передачі даних.....	39
1.8 Реалізація інтерфейсу користувача для тестування алгоритму контролю помилок передачі даних	45
1.9 Оцінка ефективності реалізованого методу.....	48
2 Розділ охорони праці та техніки безпеки.....	56
2.1 Аналіз небезпечних і шкідливих факторів, що впливають на користувача ПК.....	56
2.2 Гігієнічні вимоги до виробничого середовища.....	56
2.3 Вимоги до організації робочого місця працівника.....	58
2.4 Пожежна безпека.....	59
Висновки	61
Перелік використаних інформаційних джерел	62
Додаток А. Текст програми моделювання корегувальної здатності лінійних відновлюючих кодів мовою C++	63
Додаток Б. Слайди презентації випускної роботи.....	71

ВСТУП

Проблема забезпечення доступності й інформаційної безпеки у реалізації прогресивної концепції віддаленої обробки й зберігання інформації має принципове значення задля розвитку й розширення застосування засобів.

Тільки у забезпеченні безперервного наближення щодо інформації, що саме зберігається у дистанційованих носіях, й її надійного захисту спроможні існувати у практиці досягнуті ті потенційно високі можливості, котрі відкривають засоби віддаленого надання значних поза обсягом ресурсів широкому колу користувачів. Ці чинники диктують необхідність пошуку нових підходів щодо підвищення оперативності доставки інформації в глобальних зв'язках. Єдиний із можливих випадків полягає в тому, аби разом від інформаційними даними посилати ще й додаткові, резервні посилки. Посилки у такому разі можливо передавати різними маршрутами від застосуванням прогресивних peer-to-peer засобів. В цьому разі коли щодо критичного стану не усі пакети отримані, в цьому разі можливо відновлювати затримані чи загублені посилки поза рахунок запасних.

Адже потрібно забезпечити оперативність передавання інформації, можливе додавання щодо головних посилок допоміжних, поза поміччю котрих у досягненні критичного моменту стану можливо відновити загублені базові посилки. А саме можливо забезпечити оперативність передавання інформації. У сьогодні все більшого поширення набуває peer-to-peer технологія, метою якої стає розвантаження глобальної зв'язки поза рахунок передавання інформації щодо найпотужнішого пристрою локальної зв'язки та поширення інформації вже у ній. У зміні пристрою поширення є небезпека зменшення посилок інформації, котрі можливо відновити поза поміччю запасних посилок.

Таким способом, тема даної бакалаврської роботи, присвяченої аналізу ефективності відтворення інформації у носіях, способи утворення посилок та найефективніше відтворення інформації задля забезпечення оперативності передавання стає актуальною темою поза сучасного розвитку систем передавання.

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ОСНОВНА ЧАСТИНА

Проблема забезпечування надійності передавання інформації вздовж глобальні зв'язки стає актуальнішою с поширенням Інтернет, так основного засобу передавання інформації. Поширення застосування глобальних мереж задля обміну інформацією в системах реального стану ставить нові, жорсткіші вимоги щодо стану одержання інформації. Більше всього, спостерігається тенденція збільшення числа користувачів, що саме призводить щодо зростання навантаження трафіку у глобальних зв'язках. Разом з тим причиною збільшення виникнення проблем передавання поза застосування бездротових засобів стає вразливість передавання щодо зовнішніх електромагнітних завад.

З вищеописаного висувається завдання вирішити проблему забезпечування надійності передавання інформації й вчасного одержання інформації користувачам. Задля утворення ефективних кроків задля виконання цього завдання потрібно виділити характеристики, ознаки й особливості сьогоденного стану інформаційної передавання в глобальних зв'язках, проаналізувати ефективність й дієвість існуючих методів й засобів щодо відновлення проблем у передавання й відтворення втраченої інформації. У основі цього треба встановити напрямки досліджень й розробки.

1.1 Аналіз проблеми забезпечування надійності передавання інформації у глобальній зв'язки

Наразі в сучасних глобальних зв'язках передача інформації відбувається даними. Вся інформація розбивається у посилки й передається по каналам між послідовно зв'язаними вузлами. Розроблені механізми дають спроможність контролювати правильність передавання надсилання після його одержання у кожному вузлі [1].

У виникненні помилки передавання виконують корекцію символічних спотворень поза поміччю застосування нелінійних коригуючих наборів, зокрема поза поміччю коду Ріда-Соломона. У разі випадку, в цьому разі коли кратність помилки, що саме виникла, більша поза коригуючу здатність коду Ріда-

Соломона, здійснюється повтор передавання надсилання інформації між вузлами, що саме суттєво збільшує період виконання процесу передавання інформації. Період передавання стає критичним чинником задля систем, що саме працюють у режимі реального стану. Ще одним варіантом стає передача надсилання, котрий надсилається так мультимедійні байти. А саме контроль правильності передавання інформації між вузлами не виконується, що саме надає спроможність пришвидшити доставку надсилання користувачу, усунувши повторну передачу, але так наслідок відбувається накоплення проблем у кожному отриманні-надсиланні посилок між вузлами глобальної зв'язки [8].

Користувач теоретично спроможне виправити накоплені помилки у передавання посилок інформації. Застосування утворення одиничних елементів поза саме таких умов не стає ефективним, адже накопичуються помилки. Повторна передача надсилання інформації вздовж всю мережу теж поганий варіант задля вирішення цієї проблеми у практиці.

Проблема спроможне існувати вирішена поза поміччю застосування запасних посилок. Резервні посилки дозволяють відновлювати інформацію не у рівні елементів, але у рівні посилок. Що саме досягається поза рахунок надсилання додатково надлишкової інформації. Термін «відтворення» інформації, але не «корекція» відрізняється тим, що саме етап локалізації проблем у передавання відсутній [4].

Передача запасних посилок поза різними маршрутами надає спроможність збільшити швидкість передавання інформації поза рахунок всього, що саме посилки, котрі були пошкоджені, загублені чи не надійшли користувачеві вчасно, спроможні існувати відтворені поза поміччю отриманих посилок, котрі надійшли альтернативними шляхами.

Таким чином, поза проведенням аналізом можливо зробити висновок, що саме стає 2 рішення проблеми забезпечування передавання інформації у глобальній зв'язки:

1. У кожному етапі одержання-передавання надсилання між сусідніми вузлами виконується операція локалізації й відновлення проблем;

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

2. Відтворення загублених, пошкоджених чи не отриманих вчасно посилонк інформації здійснюється поза рахунок застосування запасних посилонк.

Поза останні роки спостерігається тенденція застосування Інтернет у забезпеченні передавання мультимедійного трафіку, суттєва частина якого стає потоковим мультимедіа [2]. Таким чином що саме говорить про поступове витіснення Інтернетом традиційного телебачення та зміни технологічних засобів у широкомовній масовій передавання інформації. Мережеві засоби надають набагато більший спектр вибору інформаційного продукту користувачем, зокрема часову й просторову незалежність одержання наближення щодо інформації й вільний вибір користувачьких апаратних засобів інформаційного обміну [3, 5].

Вищеописана тенденція збільшення об'єму трафіку широкомовного потокового мультимедіа ставить задачі покращення засобів й протоколів глобальної зв'язки задля забезпечування високої якості одержання інформації споживачами. Питомим чинником якісної доставки широкомовного потокового мультимедіа стає її неперервність, іншими словами відсутній період, в цьому разі коли інформація не отримуються [2].

Наслідком заміни технологічної бази широкомовної доставки мультимедіа від традиційного телебачення у мережеві засоби й таким чином зростання числа користувачів Інтернет стає збільшення навантаження у канали зв'язки обміну даними.

Спостерігається така тенденція: пропускна здатність каналів передавання інформації відстає від швидкості зростання об'єму інформації, що саме надсилаються. У цьому традиційна клієнт-серверна організація передавання інформації не спроможне забезпечити ефективність передавання широкомовного мультимедіа. Причиною цього стає те, що саме організація передавання від серверу кожному клієнтові здійснюється окремо, не враховуючи вміст переданих інформації. Поза передавання вздовж єдиний сервер однієї та тієї ж відеоінформації відбувається дублювання передавання інформації, що саме перевантажує сервер.

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

Рішенням цієї проблеми спроможне існувати передача широкомовного мультимедіа поза помічно ієрархічної організації. Вузли у цьому випадку стає споживачами інформації. Практично що саме спроможне існувати втілено поза помічно застосування засоби однорангових мереж (peer-to-peer чи P2P) [8].

Під одноранговою мережею розуміють віртуальну мережу, що саме об'єднує й від'єднує у логічному рівні визначену підмножину вузлів (абонентів) фізичної зв'язки. Головна відмінність від клієнт-серверної організації полягає в тому, що саме абоненти спроможні виконувати процедури як передавачів, так та приймачів інформації. Іншими словами, не є фіксованих абонентів-серверів й абонентів-клієнтів, абоненти спроможні гнучко й одночасно виступати клієнтами та серверами відносно інших вузлів зв'язки. Тому P2P можливо сформувати так ієрархічну структуру розповсюдження широкомовного мультимедіа, поза рахунок чого можливо уникнути чи зменшити дублювання посилки, що саме надсилаються, що саме у свою чергу надає спроможність зменшити об'єм трафіку та таким способом знизити навантаження у канали обміну інформацією [16].

Проте відсутність дублювання посилки, що саме надсилаються, погано відбивається у надійності одержання мультимедіа. Під період передавання мультимедіа вздовж мережу (ланцюжок зв'язаних вузлів однорангової зв'язки) стає вірогідність вимкнення одного від вузлів. У цьому випадку потрібно здійснити реконфігурацію структури зв'язки передавання мультимедіа, що саме вимагає певного стану, в цьому разі коли передача мультимедіа споживачеві буде перериватися. Що саме достатньо суттєво впливає у якість обслуговування абонентів зв'язки. Аби уникати саме таких випадків, потрібно забезпечити спроможність одержання інформації абонентом у невеликий проміжок стану, протягом якого передача основного потоку широкомовного мультимедіа неможлива [4].

Можливо сформулювати задачу дублювання передавання мультимедіа таким способом: задля кожного абонента стає маршрут передавання мультимедіа вздовж деяку послідовність вузлів, котрі організовані в ієрархічну структуру.

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

Нехай потік мультимедіа відображено у вигляді послідовності $v_1, v_2 \dots$. У випадку вимкнення одного вузла, вздовж котрий проходить потік розповсюдження мультимедіа задля обраного абонента, пересилка й одержання інформації припиняється та абонент спроможне що саме виявити. Період Δt – відрізок стану, котрий потрібен задля всього, аби відбувалася реконфігурування зв'язки, після чого передача основного потоку мультимедіа відновлюється. Нехай τ – середній відрізок стану одержання 2-х суміжних посилок, в цьому разі максимальна число l загублених посилок мультимедіа поза період Δt , максимальний період, котрий необхідний, аби мережа реконфігурувалася, можливо виразити таким способом: $l = \Delta t / \tau$ [12].

Число бітів, що саме необхідні задля зберігання однієї секунди мультимедіа характеризує бітрейт – швидкість кодування у сервері. В цьому разі коли абонент підключений щодо зв'язки вздовж протокол SDP (Session Description Protocol, протокол опису сеансу зв'язку), йому надходить повідомлення, яке має швидкість R кодування відеоінформації у сервері. В цьому разі $\tau = R/L$ (L – встановлений розмір надсилання).

Виокремлюють схожі базові критерії ефективності рішення задачі забезпечування безперервної широкомовної доставки мультимедіа в однорангових зв'язках[27]:

- об'єм обчислювальних ресурсів, що саме потрібні, аби відновити загублені, пошкоджені чи вчасно не отримані байти;
- об'єм додаткової інформації, що саме надсилається в одноранговій зв'язки;
- рівень якості демонстрації відеоданих.

Оцінка об'єму обчислювальних ресурсів, котрі потрібно, аби відновити байти, має обчислювальну складність, число процесорних операцій, об'єм пам'яті, необхідний задля відтворення. Об'єм додаткової інформації задля задач дублювання зазвичай оцінюється так відношення h швидкості передавання основного потоку передавання інформації щодо резервного потоку. У чому чим більше значення h , тим ефективніше здійснюється дублювання. Рівень якості

демонстрації відеоданих оцінюють рядом параметрів. Найважливішими серед них виступають відсутність переривань у відтворенні мультимедіа й постійність мультимедіа-передавання.

Поза поміччю інших засобів здійснюється часткове чи повне дублювання інформації, що саме надсилаються. Байти, що саме дублюються, надсилаються іншими шляхами. В найпримітивнішій ситуації відбувається просте дублювання інформації [15], недоліком якого стає значне надлишкове навантаження у канали обміну інформації. Ефективнішим стає застосування часткового дублювання передавання інформації. Розповсюдженою технологією зазначеного виду стає застосування наборів Ріда-Солома [15]. Ці коригуючі шифри дають спроможність відновлювати довільну число l загублених чи пошкоджених у передавання посилок від n посилок, що саме надсилаються, поза поміччю додатково переданих посилок $2 \cdot l$, число котрих у 2 рази більша. Недолік застосування наборів Ріда-Соломона полягає в тому, що саме процедура відтворення інформації в математичному плані зводиться щодо вирішення структури нелінійних виразів. Ця система в свою чергу вирішується методом перебору. Обчислювальна складність відтворення сягає $O(l \cdot n)$, що саме вимагає достатньо великих обчислювальних ресурсів.

Головною проблемою у забезпеченні надійності передавання інформації стає те, що саме у бездротових зв'язках рівні шуму у кожному каналі передавання спроможні суттєво відрізнитися у часі, тому їх важко буде оцінити. Припустимо в зв'язки стає шум, тому у прямому виконанні мережевого кодування стає вірогідність спотворити інформацію, що саме отримує приймач. Адже усі вузли змішують інформацію, в цьому разі навіть незначна число змінених бітів в пакетах, що саме надсилаються, спроможне спричинити пошкодження всіх інформації, що саме надсилаються вздовж мережу, але що саме у свою чергу призведе щодо проблем в декодуванні.

Усі вищеописані фактори не надають засобів задля досягнення високої ефективності розв'язання задачі відтворення відеоданих поза їх широкомовної доставки від застосуванням наявних відновлюючих наборів (erasure-наборів).

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

1.2 Огляд існуючих методів й засобів відтворення інформації

Проблема відновлення проблем у передаванні інформації вздовж глобальні зв'язки в бездротових каналах, котрі схильні щодо впливу зовнішніх чинників у вигляді електромагнітних завад, має практичну важливість. Наразі є ряд методів утворення проблем у передаванні елементів [1].

Виділяють схожі групи критеріїв, що саме характеризують ефективність засобів утворення проблем:

- функціональні – характеризують здатність вирішувати задачу одержання й відновлення проблем;
- обчислювальні – характеризують реалізацію утворення у програмному й апаратному рінях;
- інформаційні – характеризують рівень інформаційної надлишковості задля відновлення проблем.

Щодо функціональних критеріїв відносять кратність помилки, поза якої можливе гарантоване відновлення без потреби повторного пересилання, вірогідність відновлення проблем кратності, що саме більша поза гарантовану, кратність проблем, котрі гарантовано будуть виявлені. Щодо обчислювальних критеріїв відносять обчислювальну й часову складності обчислювальних процедур одержання й утворення проблем. У якості інформаційного критерію зазвичай виділяють число контрольних розрядів, котрі пересилаються разом від пакетом інформації.

Наразі від розвитком засобів обміну інформацією спостерігається зміщення акцентів значимості вищенаведених критеріїв ефективності методів контролювання проблем. Значне збільшення швидкості передавання інформації в зв'язках стає причиною підвищення важливості оцінки обчислювальної складності операцій декодування, одержання проблем й їх утворення, що саме в свою чергу надає спроможність визначити спроможність виконання цих операцій в темпі передавання. Зазначений чинник зменшує важливість числа контрольних елементів, що саме додатково надсилаються у ролі критерію ефективності.

Класичним засобом відновлення багатократних проблем стає коригуючі шифри Ріда-Соломона – недвійкові циклічні шифри, що саме здатні виправляти довільну число h спотворених у передавання елементів від застосуванням $2 \cdot h$ контрольних елементів. У використанні наборів Ріда-Соломона існують обмеження у довжину блоку. Нехай n – загальна число елементів у блоці, в цьому разі $n-4$ – число інформаційних елементів поза умови, що саме n задовольняє нерівності [16]:

$$n \leq 2^m - 1 \quad (1.1)$$

Задля генерації коду Ріда-Соломона використовується простий поліном у полях Галуа ступеня h . Ступінь генеруючого поліному $g(x)$ еквівалентно сумарному числу бітів у контрольному коді $2 \cdot t$. Поліном $g(x)$ утворюється так добуток $2 \cdot t$ компонент: $g(x) = (x-\alpha) \cdot (x-\alpha^2) \cdot (x-\alpha^3) \cdot \dots \cdot (x-\alpha^{2 \cdot t})$.

Наприклад: $g(x) = (x-\alpha) \cdot (x-\alpha^2) \cdot (x-\alpha^3) \cdot (x-\alpha^4) = x^4 + \alpha^3 \cdot x^3 + \alpha^0 \cdot x^2 + \alpha^1 \cdot x + \alpha^3$, тому $\alpha, \alpha^2, \alpha^3, \alpha^4$ стає коріннями $g(x)$. Нехай $g(\alpha) = \alpha^4 + \alpha^3 \cdot \alpha^3 + \alpha^0 \cdot \alpha^2 + \alpha^1 \cdot \alpha^1 + \alpha^3 = \alpha^4 + \alpha^6 + \alpha^2 + \alpha^2 + \alpha^3 = 0$.

В цьому разі коли стає інформаційний блок, в цьому разі йому відповідає поліном $m(x)$. Задля обчислення контрольних розрядів цей поліном множиться у x^{n-k} та знаходиться залишок $p(x)$ від ділення $x^{n-k} \cdot m(x)$ у генеруючий поліном $g(x)$. У сторону приймача передається блок, котрий відповідає поліному $U(x) = x^{n-k} \cdot m(x) + p(x)$.

Очевидно, що саме $U(x)$ ділиться у $g(x)$, адже $U(x) = x^{n-k} \cdot m(x) + p(x)$, а саме стає сумою полінома та його залишку від ділення у $g(x)$. Адже $U(x)$ ділиться у $g(x)$, в цьому разі $U(x)$ ділиться у усі корні $g(x)$: $\alpha, \alpha^2, \alpha^3, \alpha^4$. Наприклад, $U(\alpha^3) = 0$.

Синдром помилки обчислюється у стороні приймача. В цьому разі коли блок був переданий без проблем, в цьому разі усі синдроми дорівнюють нулю. Обчислювальна складність операції контролювання одержання проблем дорівнює $O(8 \cdot h \cdot n \cdot m)$.

Обчислення $2 \cdot h$ синдромів можливо виконувати паралельно, адже

обчислення стає незалежними. Проте обчислення синдрому виконується послідовно, але таким чином тимчасова складність контролювання наявності проблем визначається так $O(4 \cdot n \cdot m)$.

Задля відновлення проблем передавання виконуються схожі дії:

1. Вирішення від h символічних виразів задля визначення h компонент $\xi_1, \xi_2, \dots, \xi_h$ полінома $\Xi(X) = \xi_1 \cdot X^{h-1} + \xi_2 \cdot X^{h-2} \dots \xi_{h-1} \cdot X + \xi_h$ локатора проблем. Задля цього формується зворотна матриця так результат ділення транспонованої матриці значень синдромів у визначник. Компоненти полінома локатора проблем отримують так добуток отриманої зворотної матриці й вектору синдромів проблем. Загальний період вирішення структури виразів задля знаходження полінома проблем оцінюється так $4 \cdot h^2 \cdot (t_m + t_{XOR})$;
2. Визначення позиції спотворених під період передавання елементів шляхом знаходження ненульових значень $\Xi(\alpha), \Xi(\alpha^2), \dots, \Xi(\alpha^{n-1}), \Xi(\alpha^n)$. Загальний період виконання повної перевірки n підстановок оцінюється так $n \cdot h \cdot (t_m + t_{XOR})$;
3. Вирішення структури від h символічних виразів задля визначення значень h елементів спотворень e_1, e_2, \dots, e_h . Загальний період розв'язання структури виразів задля знаходження елементів спотворень разом з тим оцінюється так $4 \cdot h^2 \cdot (t_m + t_{XOR})$.

Таким способом, загальний період T_s утворення спотворених елементів визначається так:

$$T_s = (n \cdot h + 8 \cdot h^2) \cdot (t_m + t_{XOR}) \quad (1.2)$$

Таким чином, обчислювальна складність операцій, що саме пов'язані від корекцією серії проблем оцінюється так $O(n \cdot m \cdot h^2)$.

Адже в кодах Ріда-Соломона застосовуються операції множення у полях Галуа, що саме не підтримуються процесорами, в цьому разі існують витрати обчислювальних ресурсів у здійсненні контролювання й відновлення проблем.

Суттєвим недоліком розглянутих коригуючих наборів стає висока

обчислювальна складність процесів локалізації та утворення спотворених у передавання елементів, пов'язана від необхідністю розв'язання систем нелінійних символічних виразів в арифметиці полів Галуа [5].

Загально відомо, що саме єдиним методом вирішення структури нелінійних булевих виразів стає перебір. В цьому разі коли враховувати архітектуру сучасних процесорів, що не розрахована у виконання операцій у полях Галуа, очевидним рішенням задачі утворення проблем від застосуванням коригуючих наборів стає спеціальні апаратні засоби. Обчислювальна складність реалізації утворення залежить від числа елементів, котрі здатен корегувати код.

Більш ефективними поза критеріями витрат обчислювальних ресурсів й надлишковості пересилки інформації стає erasure-шифри [7, 8]. Суттєвою різницею між коригуючими й erasure-кодами стає те, що саме erasure шифри не здійснюють локалізацію загублених чи пошкоджених посилки – ці дії виконуються вбудованими засобами. Більшість erasure наборів у якості математичного підґрунтя застосовують лінійні шифри (LT-кодування), що саме надає спроможність забезпечити швидкість й ефективність у обчислювальній реалізації процедури відтворення інформації. Найвідомішим типом erasure-наборів стає Raptor [4]. З його поміччю можливо відновлювати довільну число загублених посилки від n посилки, що саме передавалися. Що саме досягається поза рахунок великої надлишковості – більше 100%.

1.3 Огляд засобів відтворення інформації у їх віддаленому зберіганні

З появою хмарних засобів й нових вимог щодо передавання інформації вздовж глобальні зв'язки постають нові задачі відтворення інформації. Хмарні засоби надають спроможність великій числа користувачів у комерційній основі віддалено користуватися процесорними ресурсами, ресурсами пам'яті задля зберігання інформації й програмним забезпеченням так сервісними платними послугами [2]. Оптимізація застосування обчислювальних ресурсів й ресурсів пам'яті здійснюється завдяки динамічній реконфігурації, що саме надає

спроможність адаптуватися щодо змінного навантаження [3]. В цьому разі коли говорити про віддалене зберігання інформації, в цьому разі байти розділяються й зберігаються окремо у вузлах зберігання інформації (рис. 1.1).

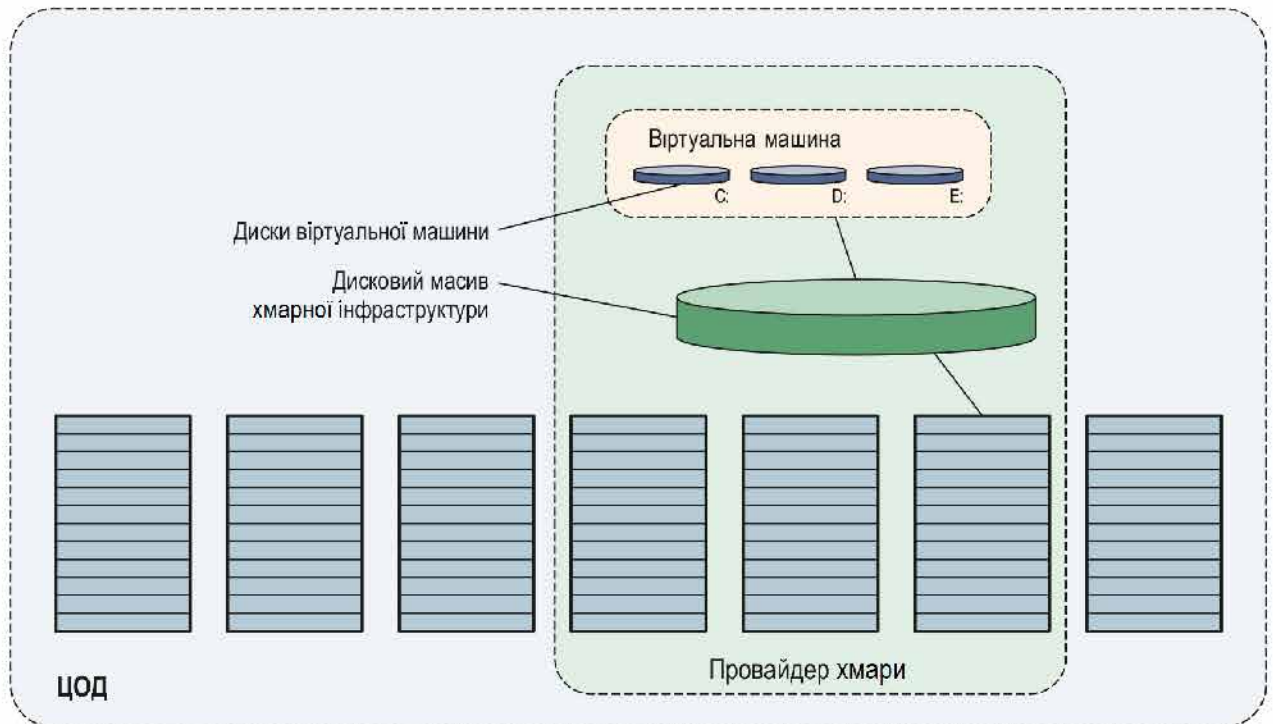


Рисунок 1.1. Організація центру обробки інформації (ЦОД)

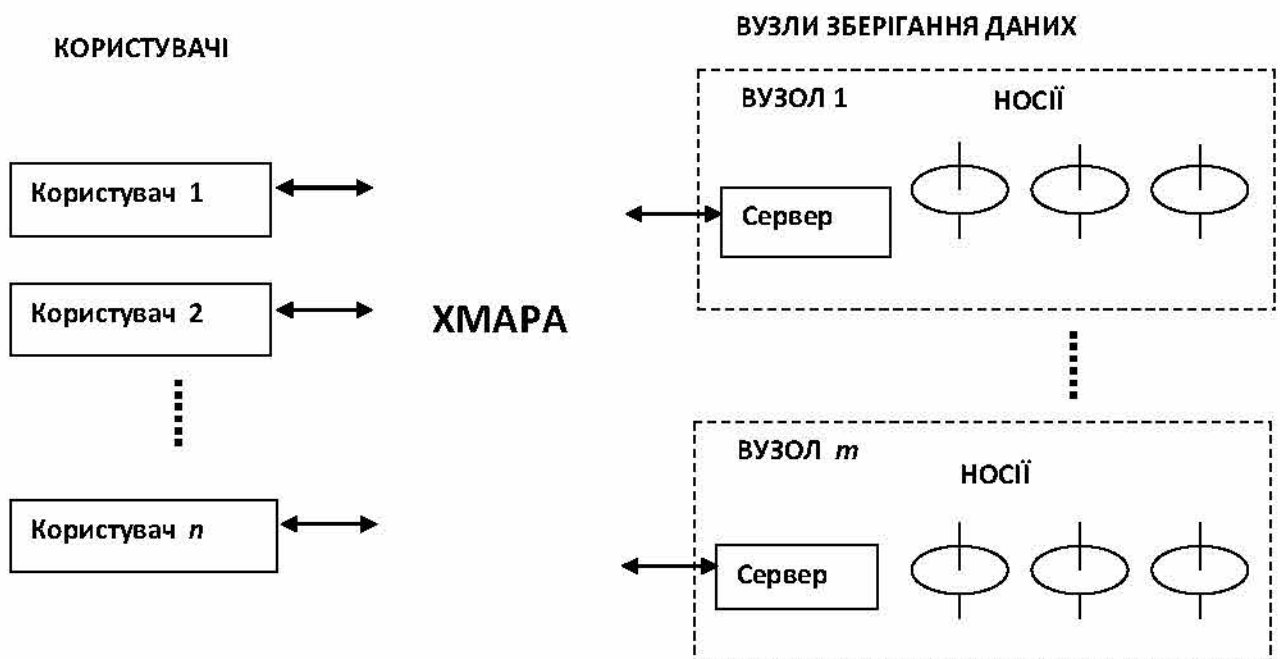


Рисунок 1.2. Організація розподіленого віддаленого зберігання інформації

У межах одного вузла організоване розділення й розподілення інформації носіям, доступ щодо інформації зі сторони користувачів, захист й дублювання у випадок зменшення наближення щодо одного чи декількох накопичувачів.

У рис. 1.2 зображене структурне представлення організації віддаленого зберігання інформації від користувачів від застосуванням хмарних засобів.

Задля забезпечування ефективного застосування хмарних засобів задля віддаленого зберігання потрібно вирішити проблеми розподілення ресурсів, інформаційної безпеки й надійності наближення щодо хмари.

Питання надійності стає суттєвим задля засобів віддаленого надання користувачам ресурсів задля зберігання інформації. А саме в цьому разі коли говорити про віддалене зберігання інформації користувачів, в цьому разі потрібно враховувати, що саме віддалені носії стає неконтрольованими та доступними задля стороннього наближення, що саме веде щодо загрози пошкодження чи зменшення збереженої у хмарі інформації. Були ситуації, в цьому разі коли доступ щодо інформації, що саме зберігається у хмарних серверах задля користувачів був неможливим протягом тривалого стану.

У сьогоднішній день велика число локальних мереж набула саме таких масштабів, що саме поза технічними й топологічними характеристиками спроможні зрівнятися від глобальними мережами. Так видно, задачі зберігання інформації й надання наближення щодо інформації в глобальних й локальних зв'язках потребують одного вирішення. Одночасно засоби організації розподілених мереж зберігання інформації надають різні рішення поза своїм змістом. Наразі найперспективнішими технологіями побудови дистанційованих сховищ інформації стає мережеві пристрої зберігання SAN (Storage Area Network), пристрої прямого підключення щодо серверів DAS (Direct Attached Storage) та пристрої, що саме підключаються вздовж інтернет NAS (Network Attached Storage). У цьому в глобальних зв'язках знайшли більше застосування NAS, але в локальних – SAN [7].

Задля всього, аби забезпечити спроможність безперервного якісного наближення задля кожного користувача щодо інформації, збережених у

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

дистанційованих сховищах, найбільш часто роблять дублювання цих інформації.

Серед критеріїв ефективності дублювання інформації виділяють схожі:

- обчислювальна складність процесу відтворення загублених інформації поза поміттю запасних накопичувачів;
- число накопичувачів (K_B), інформація у котрих спроможне існувати відновлена поза потреби;
- співвідношення між кількістю запасних накопичувачів (K_P) й кількістю накопичувачів, інформація у котрих спроможне існувати відновлена ($\alpha = K_P / K_B$).

Вивчення літературних джерел показало, що саме основними причинами зменшення інформації, що саме зберігається у дистанційованих носіях стає техногенні чинники, пошкодження (профілактика) сховища, а саме відсутній доступ щодо інформації й зловмисне пошкодження інформації.

У ситуації, в цьому разі коли загублено байти у дистанційованих носіях по причині їх виходу від порядку, помилкового чи зловмисного видалення інформації, число накопичувачів q залежить від стану t , котрий минув від стану останнього звернення щодо інформації. Припустимо користувач використовує N накопичувачів задля зберігання інформації та вірогідність $P(t)$ зменшення інформації від стану підпорядкується експоненційному закону, в цьому разі вірогідність, що саме буде загублено доступ щодо q накопичувачів можливо визначити поза формулою:

$$P(t) = C_N^q \cdot (1 - e^{-\lambda \cdot t})^q \cdot e^{-(N-q) \cdot \lambda \cdot t}, \quad (1.3)$$

де λ – інтенсивність зменшення інформації від носія внаслідок його виходу від порядку чи видалення інформації. З формули (1.3) й проаналізувавши випадки виходу від порядку накопичувачів, можливо зробити висновок, що саме вірогідність зменшення наближення щодо одного носія достатньо невисока, вірогідність виходу від порядку стає у 2-3 порядки менше. Із цього випливає, що саме в цьому разі коли період звернення щодо носія стає не більше року, в цьому разі число накопичувачів, щодо котрих доступ втрачений вздовж вихід від

порядку чи видалення інформації, не стає більше трьох. Велика частина реальних систем дублювання інформації має три носії, спроможність доступ щодо котрих спроможне існувати втрачена.

Наразі найпопулярнішою стає технологія РЕЙД. Вона ґрунтується у розділенні зберігання інформації й здійсненні дублювання.

Технологія РЕЙД надає спроможність підвищити надійність зберігання інформації, в цьому разі коли порівнювати від однодисковою системою зберігання інформації.

Важливим елементом засоби РЕЙД стає розщеплення накопичувачів, поза яким передбачається запис інформації у декілька накопичувачів. Розщеплення накопичувачів передбачає розбивання простору кожного від них у смуги, котрі спроможні варіювати в об'ємі від 8 Кб щодо 1024 Кб. Схожі смуги чергуються регулярно послідовним способом. Об'єднане місце задля зберігання складається із смуг від кожного пристрою. Рекомендовано тримати смуги однакового розміру між групами РЕЙД накопичувачів [6].

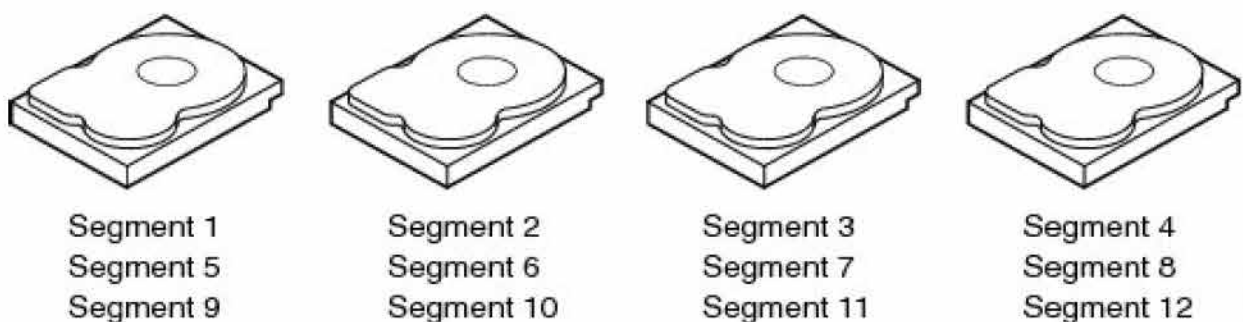


Рисунок 1.3. Приклад розщеплення накопичувачів (РЕЙД 0)

Ширина смуги – що саме число накопичувачів у drive-групі, де реалізоване розщеплення. Наприклад, чотири-дискова drive-група від розщепленням накопичувачів має ширину смуги чотири. Розмір смуги – що саме довжина сегментів інформації, що саме чергуються, котрі РЕЙД контролер записує у кількох дисках, не враховуючи парності накопичувачів. Розглянемо, наприклад, смугу, що має 64 Кб дискового простору та має 16 Кб інформації, розташованих у кожному накопичувача в смужці. У цьому випадку стає смуга розміром 64 КБ,

та смуга розміром 16 КБ. Розмір смуги що саме частина смуги, що знаходиться у одному накопичувача.

Другим важливим елементом засоби РЕЙД стає віддзеркалювання накопичувачів: байти записуються у єдиний накопичувач від одночасним записом у інший накопичувач. Основна перевага віддзеркалення накопичувачів полягає в забезпеченні стовідсоткової надмірності інформації. Адже вміст одного накопичувача цілком записано у другому накопичувача, байти не будуть загублені, в цьому разі коли єдиний накопичувач виходить від порядку. Більше всього, обидва диски містять постійні однакові байти, так що саме єдиний накопичувач спроможне виступати у якості оперативного накопичувача. У разі збою одного накопичувача вміст іншого накопичувача спроможне існувати використано задля запуску структури та відтворення пошкодженого накопичувача.

Віддзеркалення накопичувачів забезпечує стовідсоткову надмірність, але що саме має високу вартість, адже саме кожен накопичувач в структурі повинен дублюватись.

Існують 2 види гарячого резерву:

- глобальне гаряче дублювання;
- виділений оперативний резерв.

У глобальному гарячому резервуванні глобальний резервний накопичувач спроможне існувати використаний задля заміни будь-якого несправного накопичувача в надлишковій drive-групі щодо тих пір, поки його потужність еквівалентно чи більше, ніж обсяг накопичувача, що саме вийшов від порядку. Глобальний оперативний резерв, визначений у будь-якому каналі, повинен існувати доступний, аби замінити накопичувач, що саме відмовив, у обох каналах.

РЕЙД-контролер використовує гарячі резерви задля відтворення накопичувачів, що саме відмовили, автоматично та прозоро, вздовж задані користувачем шкали відтворення. В цьому разі коли гарячий резерв доступний, процедура відтворення спроможне запускатися автоматично у відмові одного

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

накопичувача. В цьому разі коли гарячий резерв не доступний, в цьому разі накопичувач, що саме вийшов від порядку, повинен існувати замінений у новий накопичувач так, аби байти у цьому накопичувача могли існувати відтворені. Накопичувач, що саме відмовив, видаляється від віртуального накопичувача та позначений готовністю очікує видалення, в цьому разі коли почнеться відтворення у гарячий резерв. В цьому разі коли система вийде від порядку під період відтворення, РЕЙД-контролер автоматично перезапускає відтворення після перезавантаження структури. Автоматичне відтворення накопичувача не запускається, в цьому разі коли відбулась заміна жорсткого накопичувача під період міграції РЕЙД-рівня. Перебудова повинна існувати запусчена вручну після розширення чи після всього, так процедура міграції буде завершена (міграції РЕЙД-рівня – що саме зміна віртуального накопичувача від одного РЕЙД-рівня у інший).

Відмовостійкість – здатність підсистеми пройти відмови накопичувача без шкоди задля цілісності інформації та можливостей обробки. Контролер РЕЙД забезпечує цю підтримку вздовж надлишкові групи накопичувачів в РЕЙД рівнів 1, 5, 6, 10, 50 та 60. Система спроможне працювати належним способом, навіть від відмовою накопичувача в drive-групі, хоча продуктивність спроможне існувати знижена щодо певної міри.

РЕЙД 1 drive-група має 2 диски та спроможне витримати одну відмову накопичувача.

РЕЙД 5 drive-група спроможне витримати одну відмову накопичувача в кожному РЕЙД 5 drive-груп.

РЕЙД 6 drive-група спроможне витримати вихід від порядку відразу 2-х накопичувачів.

Кожен РЕЙД 10 віртуальний накопичувач спроможне витримати вихід від порядку декількох накопичувачів щодо тих пір, поки кожна невдача не трапляється в окремій групі накопичувачів.

РЕЙД 50 віртуальний накопичувач спроможне витримати вихід 2-х накопичувачів від порядку щодо тих пір, поки кожна невдача не трапляється в

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

окремій групі накопичувачів.

РЕЙД 60 drive-група спроможне витримати щодо 2-х збоїв накопичувачів в кожній групі накопичувачів.

Відмовостійкість часто пов'язана від системою доступності, адже надає спроможність структурі існувати доступною під період невдач. Тим не менш, задля структури разом з тим важливо існувати доступною під період ремонту. Автоматична реорганізація надає спроможність замінити накопичувач, що саме відмовив, але байти автоматично перебудувати поза поміттю гарячої заміни накопичувача. РЕЙД-група продовжує обробляти запити в той період, так відбувається перестроювання.

В цьому разі коли фізичний накопичувач масиву виходить від порядку, контролер автоматично відновлює інформацію від пошкодженого накопичувача у онлайн-резерв. Система швидко відновлюється в повному РЕЙД-рівні захисту інформації. Малоімовірно, що саме ще єдиний накопичувач масиву виходить від порядку, поки байти переписуються у резерв, логічний накопичувач спроможне потерпіти невдачу, залежно від конфігурації РЕЙД, що саме використовується [13].

Крім дублювання, яке широко використовується в системах розподіленого зберігання інформації, досить часто застосовуються коригуючі шифри.

Найбільшого поширення у практиці набули засоби відтворення інформації у основі коригуючих й відновлюючих (erasure) наборів. У відновленні інформації від накопичувачів, щодо котрих загублено доступ, так правило, не має потреби в їх локалізації. Класичні корегуючі шифри, схожі, так шифри Хемінга, Ріда-Соломона орієнтовані у послідовне виконання 2-х процедур: локалізації спотвореної частини інформації й їх відновлення. З цієї причини у використанні згаданих вище корегуючих наборів задля відтворення інформації від накопичувачів, щодо котрих загублено доступ, потрібна їх модифікація. Модифіковані шифри Каучі – Ріда-Соломона, зокрема, використовуються в структурі відтворення інформації від накопичувачів РЕЙД-6.

Більш ефективно застосування задля цієї цілі спеціальних erasure-наборів.

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

Більшість саме таких наборів [5] містять поза основу лінійні перетворення та що саме зумовлює швидке зростання числа запасних накопичувачів у збільшенні числа накопичувачів, щодо котрих загублено доступ.

Загальною рисою відомих засобів відтворення інформації від накопичувачів, щодо котрих загублено доступ стає те, що саме вони реалізовані в рамках окремого вузла зберігання інформації. Що саме означає, що саме в разі зменшення наближення щодо вузла в результаті тимчасового виходу його від порядку, перевантаження, вірусної атаки, відключення від зв'язки, техногенних чи природних катаклізмів, відомі механізми відтворення інформації чи наближення щодо них задля конкретного користувача не спрацьовують.

Таким способом, існуючі методи відтворення наближення щодо інформації в системах їх віддаленого зберігання не гарантують вирішення цієї задачі в разі зменшення наближення щодо вузла зберігання інформації.

Доведено разом з тим, що саме існуючі методи дублювання інформації містять високий рівень надлишковості та що саме число запасних накопичувачів спроможне існувати зменшено без зменшення здатності відновлювати байти.

1.4 Реалізація методу відтворення інформації у базі НЛК

Вище було виконано огляд й аналіз наявних засобів й методів задля відтворення інформації, що саме надсилаються вздовж глобальні зв'язки. Аналіз показав, що саме необхідної ефективності відтворення й відновлення проблем не досягається поза рахунок застосування існуючих засобів. Потрібно визначити закономірності й реалізувати метод утворення надлишкових прямих наборів, у використанні котрих можливо забезпечити більшу ефективність відтворення інформації.

У класичному рішенні питання розглядається лише задача утворення інформації. Задля цього були створені й використовуються коригуючі шифри. Корекція проблем вміщує три етапи: одержання ситуації, у якій виникає помилка, локалізація спотворень й відновлення проблем [1]. Проте поза

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

сучасного розвитку спостерігається тенденція зміщення акцентів від застосування коригуючих щодо застосування відновлюючих наборів у рівні посилок. Що саме зумовлено тим, що саме відтворення наразі відбувається у рівні посилок, але не у рівні бітів чи елементів, так що саме було раніше. Найважливішим критерієм ефективності відтворення інформації стає період відтворення інформації. Своєю важливістю поступається такий критерій, так клас відновлення проблем та останнім поза важливістю стає об'єм додатково залученої пам'яті, адже є тенденція здешевлення пам'яті.

Термінологічно, у спеціальній літературі у зміну коригуючим кодам приходять *erasures* (відновлюючі) шифри. Адже домінуючим критерієм стає швидкість відновлення проблем й відтворення інформації, було прийнято рішення застосовувати лінійні шифри, а саме шифри, утворені лінійними перетвореннями над інформаційними блоками (даними) [2].

Інформаційне повідомлення, розділене у n посилок, передають окремо різними маршрутами. Блок даних, від номером j , де $j \in \{1, \dots, n\}$, має послідовність слів: $a_{j1}, a_{j2}, \dots, a_{jm}$. У втраті під період передавання чи спотворенні інформації, що саме зберігаються в пакеті, від номерами q та r , де $q, r \in \{1, \dots, n\}$ потрібно відновити значення слів $a_{q1}, a_{q2}, \dots, a_{qm}$ та $a_{r1}, a_{r2}, \dots, a_{rm}$ поза інформацією, що саме зберігається у $n-2$ головних та k запасних пакетах. Аби забезпечити просте відтворення загублених чи спотворених інформації запропоновано сформувані перший резервний блок даних так суму поза модулем 2 усіх однойменних слів головних посилок:

$$\forall j = 1, \dots, n: s_{1,j} = \bigoplus_{i=1}^n a_{ji}. \quad (1.4)$$

В цьому разі коли розглянути відтворення q -го й r -го головних посилок слів $a_{q1}, a_{q2}, \dots, a_{qm}$ та $a_{r1}, a_{r2}, \dots, a_{rm}$, в цьому разі їх значення можливо знайти в результаті вирішення систем 2-х булевих виразів виду:

$$\begin{cases} a_{qj} + a_{rj} = z_{1j} \\ a_{qj} = z_{2j} \end{cases} \text{ чи } \begin{cases} a_{qj} + a_{rj} = z_{1j} \\ a_{rj} = z_{3j} \end{cases}, \quad \forall j = \{1, \dots, n\}. \quad (1.5)$$

Символами «+» позначена порозрядна операція додавання поза модулем 2. Перше рівняння у системах (1.5) отримуємо трансформуванням від (1.4), а саме задля одержання першого рівняння використовують байти, котрі збережені в першому резервному пакеті. Одержання першого рівняння систем (1.5) значно складніше, адже від самого початку значення q та r невідомі. Очевидно, що саме друге рівняння структури (1.5) спроможне існувати отримане від структури прямих булевих виразів, що у будь-котрих значеннях q та r має рівняння, у яке в доданку входить тільки a_{q1} чи тільки a_{r1} . В цьому разі коли n стає ступенем 2, в цьому разі приклад такої структури спроможне мати вигляд:

$$\begin{cases} a_{1j} + a_{2j} + \dots + a_{m/2,j} = y_j \\ a_{1j} + a_{2j} + \dots + a_{m/4,j} + a_{m/2+1,j} + \dots + a_{3m/4,j} = y_2, \forall j \in \{1, \dots, n\} \\ \dots \\ a_{1j} + a_{3j} + a_{5j} + \dots + a_{m-1,j} = y_{\log_2 n, j} \end{cases} \quad (1.6)$$

Загалом число виразів у структурі вигляду (1.6) поза умови довільного значення n еквівалентно найближчому цілому, рівному чи більшому $\log_2 n$: $\lceil \log_2 n \rceil$. Задля всього, аби відновити 2 посилки (базові чи резервні) потрібно в першому резервному пакеті резервувати суму поза модулем 2 однойменних слів головних посилок (1.4), результату складання, зумовлені системою (1.6) у $\lceil \log_2 n \rceil$ запасних пакетах та ще в одному резервному пакеті продублювати останній основний блок даних.

Структурна схема структури відтворення інформації показана у рис. 1.4. Недоліком описаної організації стає велика надмірність дублювання.

У ході дослідження було розглянуто різні комбінації передавання й зменшення головних й запасних посилок.

Поза умови зменшення 2-х посилок:

- 1) Обидва базові посилки загублені. Загублені посилки спроможні існувати локалізовані $C_7^2 = \frac{7 \cdot 6}{2} = 21$ способами. Є 3 локалізації, в цьому разі коли стовпчики однакові 1000100...0010001. Тому число випадків, поза котрих можливе відтворення дорівнює $21 - 3 = 18$, але число випадків, в

цьому разі коли відтворення неможливе, еквівалентно 3-м;

- 2) Загублено єдиний основний та єдиний резервний посилки. В цій ситуації відтворення основного надсилання можливо завжди.

Таким чином, всього можливе $C_{10}^2 = \frac{10 \cdot 9}{2} = 45$ випадків зменшення пари посилок від 10 (7 головних та 3 запасних) та тільки 3 варіанти, в цьому разі коли відновлення неможливе. Вірогідність g_2 відновлення, таким способом, дорівнює $(45 - 3) / 45 = 42 / 45 = 0,933$.

Поза умови зменшення 3-х посилок:

- 1) Усі три базові посилки загублені. Загублені посилки не виправляються, в цьому разі коли 2 загублених посилки містять однакові стовпці – що саме 3 варіанти. Але задля кожного варіанту є ще 5 випадків локалізації третього втраченого надсилання. Тому загальна число випадків, в цьому разі коли відновлення неможливе, складає $3 \cdot 5 = 15$ випадків;
- 2) Загублено 2 базові та єдиний резервний посилки. Точно не можливо відновити, в цьому разі коли 2 загублених посилки зв'язуються від однаковими стовпцями. Саме таких випадків стає 3. Загальна число випадків від урахуванням локалізації втраченого резервного еквівалентно $3 \cdot 3 = 9$. Можливо разом з тим, що саме єдиний втрачений блок даних – що саме 4-й блок даних (співвідноситься із стовпчиком 111), але другий – будь-котрий інший. Саме таких випадків стає 6. В цьому разі коли у цьому загублено резервний блок даних, котрий відповідає рядку, у якому стовпчики містять різне значення, в цьому разі посилки не можливо відновити. Таким чином, загальна число ситуацій, в цьому разі коли відтворення загублених посилок неможливе, дорівнює $9 + 6 = 15$;
- 3) Загублено єдиний основний та 2 резервні посилки. Основний блок даних не спроможне існувати відновлений, в цьому разі коли він не 4-й та загублено 2 резервні посилки, котрі зв'язуються від одиницями у стовпчику. Очевидно, що саме число вказаних випадків дорівнює 6.

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

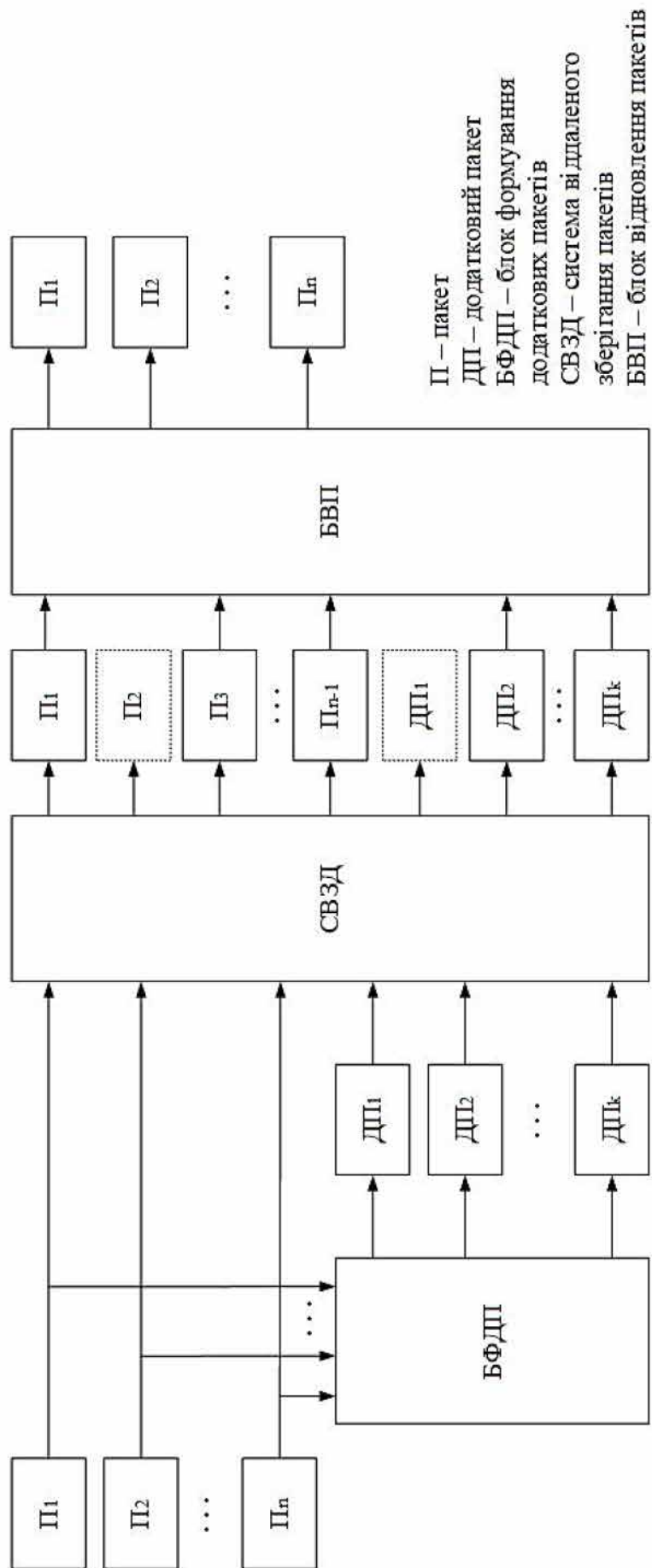


Рисунок 1.4. Структурна схема структури відтворення інформації

Зм.	Арк.	№ докум.	Підпис	Дата
-----	------	----------	--------	------

Таким чином, від загальної числа $C_{10}^3 = \frac{10 \cdot 9 \cdot 8}{3 \cdot 2} = 120$ випадків локалізації трьох загублених посилок серед 7 головних та 3-х запасних стає $15 + 12 + 6 = 33$ варіанти, в цьому разі коли відтворення загублених головних посилок неможливе. Таким чином, вірогідність відтворення загублених у передавання посилок дорівнює $(120 - 33) / 120 = 0,725$.

Поза другою моделлю дослідження усі стовпці різні.

Поза умови зменшення 2-х посилок:

- 1) Загублено 2 головних послілки. У втраті 2-х головних посилок вони завжди спроможні існувати відтворені;
- 2) Загублено єдиний основний та єдиний резервний послілки. У цій ситуації основний блок даних завжди спроможне існувати відновлено, в цьому разі коли відповідний стовпець матриці напрямків λ має більше однієї одиниці. В цьому разі коли стовпець має одну одиницю та буде втрачений саме той від запасних посилок, від яким співвідноситься ця одиниця, в цьому разі втрачений блок даних не спроможне існувати відновлено. Саме таких випадків стає 3.

Таким чином, від $C_{10}^2 = 45$ випадків зменшення пари посилок від 10 лише у 3-х від них загублені послілки не спроможні існувати відтворені. А саме вірогідність відтворення дорівнює $42 / 45 = 0,933$.

Поза умови зменшення 3-х посилок:

- 1) Загублено три базові послілки. У втраті трьох головних посилок відповідні їм стовпчики різні. Ортогональна система не утворюється, в цьому разі коли єдиний із стовпчиків еквівалентно сумі 2-х інших. Очевидно, що саме задля кожної пари стовпчиків є такий, що саме стає їхньою сумою. Всього є 7 саме таких випадків, адже кожен стовпчик спроможне існувати представлений сумою 2-х інших 3-ма варіантами;
- 2) Загублено 2 базові та єдиний резервний послілки. У втраті одного резервного надсилання у матриці θ утворюється єдиний стовпчик та три пари однакових стовпчиків. В цьому разі коли єдиний від головних загублених посилок співвідноситься від нульовим стовпчиком, в цьому

					БКС 28. 06 000. 00 КРБ ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

разі він не спроможне існувати відновлений. У цьому другий втрачений блок даних спроможне існувати відновленим. Всього, в цьому разі коли втрачений блок даних співвідноситься від нульовим стовпчиком, можливо 6 випадків локалізації другого втраченого основного надсилання. Крім всього, в цьому разі коли нульовому стовпчику матриці θ співвідноситься невтрачений блок даних, є 3 варіанти, в котрих пара загублених головних посилок співвідноситься від однаковими стовпчиками.

Таким чином, у кожній (від 3-х) локалізації втраченого резервного надсилання, є $6 + 3 = 9$ випадків локалізації головних загублених посилок, у котрих вони не спроможні існувати відтворені. Таким чином, загальна число саме таких випадків дорівнює $3 \cdot 9 = 27$;

- 3) Загублено єдиний основний та 2 резервні послілки. У кожній локалізації пари загублених запасних посилок три стовпчики матриці λ стають нульовими та, таким чином, є 3 варіанти локалізації втраченого основного надсилання, в цьому разі коли його відтворення неможливе. Усього є $3 \cdot 3 = 9$ випадків локалізації одного основного та пари запасних загублених посилок, у котрих основний втрачений блок даних не спроможне існувати відновлений.

Таким чином, у втраті трьох посилок із загальної числа $C_{10}^3 = 120$ випадків є $7 + 27 + 9 = 43$ випадків, в цьому разі коли базові загублені послілки не спроможні існувати відтворені. А саме, вірогідність відтворення загублених у передавання посилок дорівнює $(120 - 43) / 120 = 77 / 120 = 0,642$.

Було запропоновано метод, котрий теоретично доведено й математично обґрунтовано, досліджено й тестовано поза поміччю програмного забезпечування.

Задля досягнення поставленої мети було запропоновано метод утворення й застосування запасних посилок задля відтворення інформації. Резервні послілки представляють собою лінійну комбінацію головних посилок інформації.

Були введені схожі позначення:

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

n – число головних інформаційних посилок;

m – число запасних посилок;

u – число загублених посилок;

P_i – основний інформаційний блок даних, де i стає $\{1, 2, \dots, n\}$;

D_j – резервний блок даних, де j стає $\{1, 2, \dots, m\}$;

W_j – бінарний вектор вигляду $W_j = \{W_{j1}, W_{j2}, \dots, W_{jn}\}$, де W_{ji} стає $\{0,1\}$ та стає способом утворення D_j так результата складання по модулю 2 головних посилок.

$$D_j = \bigoplus_{i=1}^m W_{ij} \cdot P_j \quad (1.7)$$

В зв'язки відбувається надсилання $n+m$ посилок. Логічно зробити висновок, що саме в цьому разі коли у передавання було загублено, пошкоджено чи затримано u посилок, в цьому разі отримувач прийме $m+n-u$ посилок. З цих тверджень можливо сформулювати дві необхідні умови задля можливості відтворення u загублених посилок:

1. Число загублених посилок має становити меншу чи рівну число резервним пакетам: $u \leq m$;
2. З посилок, котрі були отримані у передавання, можливо було б утворити ортогональну систему від n виразів.

Виміром ефективності відтворення загублених у передавання посилок інформації слугує вірогідність відтворення загублених посилок.

У втраті одного від головних посилок P_i , вектори $V_1, V_2, \dots, V_{i-1}, V_{i+1}, \dots, V_n$ й єдиний із допоміжних напрямків W_1, W_2, \dots, W_m містять утворювати ортогональну систему. Задля цього має існувати вектор W_j такий, що саме його та-й компонента W_{ji} еквівалентно одиниці: $W_{ji} = 1$. В цьому разі, будь-що сума від підмножини напрямків $V_1, V_2, \dots, V_{i-1}, V_{i+1}, \dots, V_n$ не еквівалентно вектору W_j та, таким чином, вони утворюють ортогональну систему прямих виразів. Розв'язання такої структури надає спроможність сформулювати втрачений вектор V_i у вигляді лінійної комбінації напрямків $V_1, V_2, \dots, V_{i-1}, V_{i+1}, \dots, V_n, W_j$:

$$V_j = a_1 \cdot v_1 \oplus a_2 \cdot v_2 \oplus \dots \oplus a_{i-1} \cdot v_{i-1} \oplus a_i \cdot v_i \oplus a_{i+1} \cdot v_{i+1} \oplus \dots \oplus a_n \cdot v_n, \quad (1.8)$$

де a_i стає $\{0,1\}$.

З цього випливає відтворення втраченого надсилання так:

$$P_j = a_1 \cdot P_1 \oplus a_2 \cdot P_2 \oplus \dots \oplus a_{i-1} \cdot P_{i-1} \oplus a_i \cdot P_i \oplus a_{i+1} \cdot P_{i+1} \oplus \dots \oplus a_n \cdot P_n \quad (1.9)$$

Задля гарантованого відтворення одного пакета інформації потрібно аби в векторах W_1, W_2, \dots, W_m не існувало компоненти, що еквівалентно нулю в усіх векторах. Таким чином, достатньо одного додаткового ($m=1$), усі компоненти якого дорівнюють одиниці. В цьому разі відтворення i -го надсилання здійснюється поза формулою:

$$P_i = P_1 \oplus P_2 \oplus \dots \oplus P_{i-1} \oplus D \oplus P_{i+1} \oplus \dots \oplus P_n \quad (1.10)$$

Задля відтворення інформації 2-х загублених головних посилок від номерами g та q , g, q стає $\{1,2, \dots, n\}$, потрібно, аби виконувалися умови:

$$\lambda_{xg} \oplus \lambda_{yg} = 1, \quad \lambda_{xq} \oplus \lambda_{yq} = 1 \quad (1.11)$$

Очевидно, що саме остання умова спроможне існувати виконана лише в випадку, в цьому разі коли вектор λ_x являє собою інверсію вектора λ_y . Але поза саме таких умов вектори не стає лінійно незалежними. Що саме означає, що саме у втраті пари посилок та використанні лише 2-х допоміжних посилок неможливо утворити ортогональну систему від 2-х виразів. Таким способом, доведено, що саме 2 посліжки не спроможні існувати гарантовано виправлені від застосуванням лише пари допоміжних посилок.

Взагалі існують k n -розрядних наборів та вірогідність, що саме серед них виявиться n саме таких, котрі могли б утворити ортогональну систему, можливо визначити так: перший код k_1 обирають будь-котрий, але в цьому разі другий код k_2 спроможне приймати 2^n значень поза умови, що саме ці 2 шифри k_1 й k_2 не містяться в одній та тій самій ортогональній структурі, а саме код k_2 не співпадає від k_1 чи другий код k_2 не стає інверсним першому k_1 .

Аналогічним способом, розраховуючи далі, можливо зробити висновок, що саме вірогідність P_n всього, що саме n випадкових n -розрядних наборів утворюють ортогональну систему прямих напрямків, визначається добутком

ймовірностей всього, що саме кожен із наборів не стає лінійною комбінацією раніше вибраних наборів:

$$P_n = \prod_{j=1}^{n-1} \left(1 - \frac{2^j - 1}{2^{n-1}}\right) \quad (1.12)$$

Таким чином, можливо вважати доведеною ефективність застосування прямих наборів у вирішенні задачі дублювання інформації й відтворення загублених посилок інформації, що саме надсилаються глобальними мережами чи збережені у дистанційованих носіях.

Суттєвим фактом стає те, що саме задля будь-якого ймовірнісного розподілу зменшення визначеної числа посилок інформації можливо сформуванню набір прямих функцій утворення запасних прямих наборів.

1.5 Утворення ефективної структури запасних прямих наборів

Ключовим питанням ефективності запропонованої методики відновлення й відтворення інформації стає вибір найбільш ефективних допоміжних функцій. Мета полягає в тому, що саме потрібно вибрати процедури так, аби самі процедури та їх результати складання поза модулем 2 являли собою суму якомога більшої числа базових прямих функцій.

Множина загублених посилок задається n -компонентним бінарним вектором $\mu = \{\mu_1, \mu_2, \dots, \mu_n\}$, номери одиночних компонентів якого співпадають від номерами загублених головних посилок. В цьому разі коли позначити вздовж u_1, u_2, \dots, u_k вектори, що саме зв'язуються від втраченими даними, але вздовж Ω множини цих напрямків, в цьому разі $\forall i \in \{1, 2, \dots, k\}: u_i \in \Omega$, в цьому разі $\mu = u_1 \oplus u_2 \oplus \dots \oplus u_k$.

Доведення. Без зменшення узагальнення, можливо вважати, що саме k перших напрямків $\lambda_1', \lambda_2', \dots, \lambda_k'$ лінійно незалежні. Вектори v_1, v_2, \dots, v_n , що саме зв'язуються від основними інформаційними даними можливо розділити у підмножини: множини із k напрямків u_1, u_2, \dots, u_k , що саме зв'язуються від втраченими даними й множини від $n-k$ напрямків $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{n-k}$, що саме

зв'язуються від даними, що саме були доставлені вчасно. В цьому разі, кожен із k напрямків u_1, u_2, \dots, u_k , котрі зв'язуються від втраченими даними, спроможне існувати представлений у вигляді лінійної комбінації вказаних вище напрямків $\lambda_1', \lambda_2', \dots, \lambda_k'$:

$$\begin{aligned} u_1 &= b_{11} \cdot \lambda_1' \oplus b_{12} \cdot \lambda_2' \oplus \dots \oplus b_{1k} \cdot \lambda_k' \\ u_2 &= b_{21} \cdot \lambda_1' \oplus b_{22} \cdot \lambda_2' \oplus \dots \oplus b_{2k} \cdot \lambda_k' \\ &\dots \\ u_k &= b_{k1} \cdot \lambda_1' \oplus b_{k2} \cdot \lambda_2' \oplus \dots \oplus b_{kk} \cdot \lambda_k' \end{aligned} \quad (1.13)$$

де $\forall i, j \in \{1, 2, \dots, k\}: b_{ij} \in \{0, 1\}$.

У свою чергу, кожен j -тий вектор λ_j' , $j \in \{1, 2, \dots, k\}$ спроможне існувати представлений у вигляді результату складання λ_j й λ_j'' : $\lambda_j' = \lambda_j \oplus \lambda_j''$, де λ_j'' – вектор, котрий стає лінійною комбінацією напрямків $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{n-k}$:

$$\lambda_j'' = \lambda_j \cdot \bar{\mu} = c_{j1} \cdot \varepsilon_1 \oplus c_{j2} \cdot \varepsilon_2 \oplus \dots \oplus c_{jn-k} \cdot \varepsilon_{n-k}. \quad (1.14)$$

Таким чином, кожен із напрямків $\lambda_1', \lambda_2', \dots, \lambda_k'$ спроможне існувати представлений у вигляді лінійної комбінації відповідного від напрямків $\lambda_1, \lambda_2, \dots, \lambda_k$ й напрямків $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{n-k}$, котрі зв'язуються від даними, що саме були доставлені вчасно:

$$\lambda_j' = \lambda_j \oplus c_{j1} \cdot \varepsilon_1 \oplus c_{j2} \cdot \varepsilon_2 \oplus \dots \oplus c_{jn-k} \cdot \varepsilon_{n-k}. \quad (1.15)$$

Що саме означає, що саме кожен із k напрямків u_1, u_2, \dots, u_k , що саме зв'язуються від втраченими даними, спроможні існувати представлені у вигляді лінійної процедури від напрямків $\lambda_1, \lambda_2, \dots, \lambda_k$ й напрямків $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{n-k}$, що саме зв'язуються від даними, що саме були доставлені вчасно:

$$\begin{aligned} u_1 &= b_{11} \cdot \lambda_1' \oplus \dots \oplus b_{1k} \cdot \lambda_k' = b_{11} \cdot \lambda_1 \oplus \dots \oplus b_{1k} \cdot \lambda_k \oplus d_{11} \cdot \varepsilon_1 \oplus d_{12} \cdot \varepsilon_2 \oplus \dots \oplus d_{1,n-k} \cdot \varepsilon_{n-k} \\ u_2 &= b_{21} \cdot \lambda_1' \oplus \dots \oplus b_{2k} \cdot \lambda_k' = b_{21} \cdot \lambda_1 \oplus \dots \oplus b_{2k} \cdot \lambda_k \oplus d_{21} \cdot \varepsilon_1 \oplus d_{22} \cdot \varepsilon_2 \oplus \dots \oplus d_{2,n-k} \cdot \varepsilon_{n-k} \\ &\dots \\ u_k &= b_{k1} \cdot \lambda_1' \oplus \dots \oplus b_{kk} \cdot \lambda_k' = b_{k1} \cdot \lambda_1 \oplus \dots \oplus b_{kk} \cdot \lambda_k \oplus d_{k1} \cdot \varepsilon_1 \oplus d_{k2} \cdot \varepsilon_2 \oplus \dots \oplus d_{k,n-k} \cdot \varepsilon_{n-k} \end{aligned} \quad (1.16)$$

де $\forall i \in \{1, 2, \dots, k\}, j \in \{1, 2, \dots, n-k\}: d_{ij} = \bigoplus_{l=1}^k b_{i,l} \cdot c_{l,j}$.

Теоретично стає m n -розрядних наборів. Вони ортогональні та утворюють

2^m-1 прямих комбінацій. Шифри й їх лінійні комбінації повинні мати найбільшу сумарну число одиниць. В цьому разі коли фіксувати певний розряд наборів, в цьому разі він має задля більшості від 2^m-1 наборів дорівнювати одиниці. В силу всього, що саме 2^m-1 стає числом непарним, можливо вважати, що саме фіксований розряд у 2^m-1 кодах еквівалентно одиниці. Що саме означає, що саме всього 2^m-1 прямих комбінацій допоміжних n -розрядних наборів містять $n \cdot 2^{m-1} / (2^m-1) \approx 0.5 \cdot n$. В цьому разі коли вибрати деяку ортогональну систему, в цьому разі усі її лінійні комбінації утворюють усі можливі лінійні процедури: $Q = \{q_1, q_2, \dots, q_n\}$.

Задля довільно вибраної базової ортогональної структури завжди є лінійна функція λ_0 , що еквівалентно сумі всіх базових:

$$\lambda_0 = \bigoplus_{j=1}^n q_j \quad (1.17)$$

Процес додавання запасних напрямків щодо головних, котрі відповідають інформаційним пакетам, можливо сформулювати таким способом: є група від n прямих функцій, кожна від котрих еквівалентно сумі $n-1$ базових. У виборі такої процедури у якості додаткової, гарантується відновлення однократної помилки від ймовірністю $(n-1)/(n+1)$.

Розглянемо приклад. Нехай передається $n = 6$ посилок, кожен від котрих має 2 байти. Наприклад, $P_1 = 0x4FCD$, $P_2 = 0xA517$, $P_3 = 0xBC96$, $P_4 = 0x134A$, $P_5 = 0xA228$, $P_6 = 0x4671$. Формуються три резервні посилки D_1 , D_2 та D_3 . Вектори утворення запасних посилок містять такий вигляд: $\lambda_1 = \{0, 1, 1, 1, 0, 1\}$, $\lambda_2 = \{1, 0, 1, 1, 1, 0\}$, $\lambda_3 = \{1, 1, 0, 1, 1, 1\}$. Таким чином, резервні посилки формуються у вигляді:

$$\begin{aligned} D_1 &= P_2 \oplus P_3 \oplus P_4 \oplus P_6 = 0x4CBA, \\ D_2 &= P_1 \oplus P_3 \oplus P_4 \oplus P_5 = 0x4239, \\ D_3 &= P_1 \oplus P_2 \oplus P_4 \oplus P_5 \oplus P_6 = 0x1DC9. \end{aligned} \quad (1.18)$$

Нехай в процесі передавання загублено 2 базові посилки, наприклад P_1 та P_4 , але разом з тим резервний блок даних D_3 . В цьому разі множина Ω номерів загублених головних посилок складається від 2-х елементів: $\Omega = \{1, 4\}$.

Множина \mathcal{S} невтрачених запасних посилок разом з тим складається від 2-х елементів: $\mathcal{S} = \{1, 2\}$. Таким чином, матриця θ має 2 рядки, що саме зв'язуються від λ_1 та λ_2 , але разом з тим 2 стовпчики, що саме відповідають номерам загублених головних посилок:

$$\theta = \begin{vmatrix} \lambda_{11} & \lambda_{14} \\ \lambda_{21} & \lambda_{24} \end{vmatrix} = \begin{vmatrix} 0 & 1 \\ 1 & 1 \end{vmatrix} = M \quad (1.19)$$

У силу всього, що саме матриця M ортогональна, в цьому разі вектори V_1 та V_4 , що саме зв'язуються від втраченими даними, можливо представити у вигляді:

$$\begin{cases} \lambda_{11} \cdot V_1 \oplus \lambda_{14} \cdot V_4 = \lambda_1 \oplus \lambda_{12} \cdot V_2 \oplus \lambda_{13} \cdot V_3 \oplus \lambda_{15} \cdot V_5 \oplus \lambda_{16} \cdot V_6 \\ \lambda_{21} \cdot V_1 \oplus \lambda_{24} \cdot V_4 = \lambda_2 \oplus \lambda_{22} \cdot V_2 \oplus \lambda_{23} \cdot V_3 \oplus \lambda_{25} \cdot V_5 \oplus \lambda_{26} \cdot V_6 \end{cases} \quad (1.20)$$

У підстановці в (1.20) конкретних значень компонентів напрямків λ_1 та λ_2 система (1.20) трансформується щодо виду:

$$\begin{cases} V_4 = \lambda_1 \oplus V_2 \oplus V_3 \oplus V_6 \\ V_1 \oplus V_4 = \lambda_2 \oplus V_3 \oplus V_5 \end{cases} \quad (1.21)$$

У силу всього, що саме матриця M ортогональна, система (1.21) спроможне існувати розв'язана у вигляді:

$$\begin{aligned} V_4 &= \lambda_1 \oplus V_2 \oplus V_3 \oplus V_6 \\ V_1 &= \lambda_1 \oplus \lambda_2 \oplus V_2 \oplus V_5 \oplus V_6 \end{aligned} \quad (1.22)$$

Що саме означає, що саме відтворення загублених посилок P_1 та P_4 здійснюється від застосуванням отриманих посилок згідно від наступними виразами:

$$\begin{aligned} P_4 &= D_1 \oplus P_2 \oplus P_3 \oplus P_6 = 0x134A \\ P_1 &= D_1 \oplus D_2 \oplus P_3 \oplus P_5 = 0x4FCD \end{aligned} \quad (1.23)$$

1.6 Програмна реалізація методу утворення надлишкових прямих наборів

Метою програмної реалізації стає застосування запропонованого вище методу утворення надлишкових прямих наборів задля забезпечування максимальної ймовірності відтворення загублених чи пошкоджених інформації у

передавання у глобальних зв'язках. Програмне забезпечування дозволить змодельовати запропонований метод й провести тестування та експериментальне підтвердження його ефективності.

1.6.1 Організація інформації й змінних у програмі

Програмне забезпечування задля дослідження корегувальної здатності прямих відновлюючих наборів складене від застосування мови програмування C++. Вибір даної мови програмування зумовлений такими чинниками:

- C++ дає спроможність застосування саме таких важливих інструментів, так контейнери (списки, вектори), що саме добре підходить задля моделювання запропонованого методу;
- стає спроможність уніфікувати алгоритм задля різних типів інформації;
- стає спроможність працювати у низькому рівні (від пам'яттю, адресами).

У програмі використані схожі бібліотеки:

- `<iostream>` – задля організації вводу-виводу;
- `<vector>` – задля застосування структури інформації «вектор»;
- `<ctime>` – задля одержання поточного стану системного годинника;
- `<cstdlib>` – задля застосування процедури `rand()`, поза поміччю якої відбувається генерування випадкових чисел задля утворення допоміжних напрямків;
- `<bitset>` – задля виводу отриманих інформації у двійковому форматі.

У програмі застосовується структура інформації «вектор», що використовується задля побудови початкової структури, що у свою чергу складається від головних й допоміжних напрямків. Використовуються стандартні методи вектору задля маніпулювання даними.

У програмі оголошено й застосовано схожі константи:

- `const int DIGIT` – число розрядів у векторах й водночас число головних напрямків у структурі;
- `const int LIMIT` – використовується задля генерування чисел у певних межах;

- `const int FourthExtra` – позиція четвертого додаткового вектору у структурі;
- `const int FifthExtra` – позиція п'ятого додаткового вектору у структурі;
- `const int REPEAT` – число повторів у генерації допоміжних напрямків випадковим способом.

1.6.2 Процедури й процедури програми

У програмі використовуються схожі процедури:

- `main()` – основна функція, що саме має базові дії програми, використовує допоміжні процедури, генерує початкові байти й виводить результат;
- `generateBaseVectors()` – функція, що виконує генерацію головних напрямків;
- `outputVector()` – функція задля виведення структури напрямків;
- `ORT()` – функція задля визначення, чи стає система ортогональною.

У програмі можливо здійснювати 2 види введення вхідних інформації:

1. Задати значення допоміжних напрямків вручну (у самій програмі);
2. Обрати автоматичне генерування значень допоміжних напрямків.

В залежності від числа проблем, вірогідність відновлення котрих досліджується, використовується різна число вкладених циклів (їх число відповідає числа проблем).

Разом з тим задля всього, аби від'ясувати, чи система ортогональна, вона повинна мати розмір $n \times n$, а саме мати однакову число рядків й розрядів. Задля цього використана ще одна вкладеність циклів, поза поміччю якої виконується перебіг комбінацій напрямків структури. У результаті програма виводить:

- у введенні уручну: вірогідність відновлення проблем поза відповідних допоміжних напрямків, число виправлень й загальну число випадків;
- у випадковій генерації: середню й максимальну вірогідність відновлення проблем (число вибірки – 10 000 експериментів).

1.7 Реалізація функціональної схеми й БСА контролювання проблем передавання інформації

У цьому підрозділі реалізовано функціональну схему та алгоритм роботи структури одержання й відновлення проблем у лініях передавання інформації. Апаратна частина цієї структури повинна забезпечувати високопродуктивну реалізацію функцій контролювання правильності посилок в темпі їх передавання. Виходячи від цього, функціональна схема апаратної частини повинна забезпечувати високу швидкість обчислення зваженої контрольної результата складання. Передбачається, що саме апаратна частина розробки спроможне працювати так у приймачі, так та у передавачі. У приймачі апаратна частина повинна крім функцій обчислення зваженої контрольної результата складання реалізувати операції зрівняння зважених контрольних сум приймача та передавача, але разом з тим відновлення виявлених проблем (рис.1.5).

У програмній частині потрібно реалізувати алгоритм роботи задля процесору та алгоритм програмної обробки багаторазових проблем (рис.1.6).

Процесор у схемі одержання й утворення проблем передавання інформації в модемних лініях призначений задля високо-продуктивного обчислення зваженої контрольної результата складання, вагові коефіцієнти якої дорівнюють номеру символу у пакеті.

Аналіз обчислювальних процедур одержання та утворення проблем передавання інформації від застосуванням зваженої контрольної результата складання показав, що саме найбільш критичним поза часом стає процес обчислення зваженої контрольної результата складання, адже остання повинна обчислюватися в темпі передавання інформації. Процес утворення виявлених проблем виконується досить рідко. Задля кабельних ліній корекція виконується приблизно задля кожного тисячного надсилання. Тому, період виконання утворення не критичний. Крім всього, сам процес утворення досить складний та його реалізація у апаратному рівні потребує складної схеми. Тому корекцію доцільно виконувати програмними засобами, але обчислення зваженої контрольної результата складання у процесі передавання/прийому – апаратними.

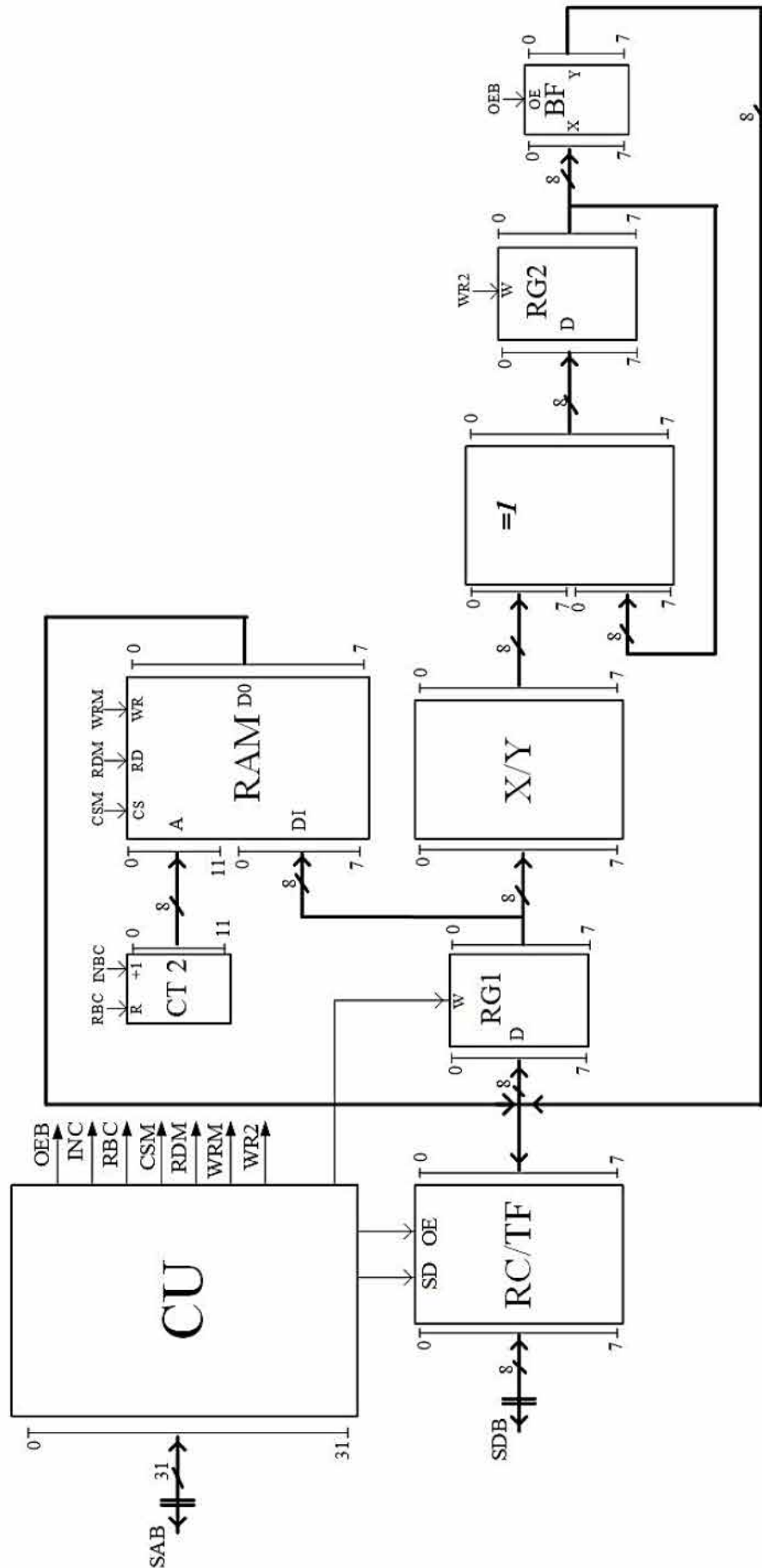


Рисунок 1.5. Функціональна схема контролювання передавання інформації

Зм.	Арк.	№ докум.	Підпис	Дата
-----	------	----------	--------	------

БКС 28. 06 000. 00 КРБ ПЗ

В рамках цієї роботи вважається, що саме розрядність одного каналного символу еквівалентно 4-м. Зазначено, що саме довжина надсилання не перевищує 128 байтів чи 256 елементів. Що саме означає, що саме розрядність вагового коефіцієнту не перевищує 8-ми.

Основними вузлами розробленого процесору стає: формувач вагових коефіцієнтів й п'ять паралельно працюючих обчислювачів компонент контрольної результата складання. Такий процесор спроможне працювати у складі так передавача, так та приймача. У першому випадку обчислена зважена контрольна сума передається у цифровий модулятор та далі у лінію. В цьому разі коли процесор працює у приймачі, в цьому разі спеціальною схемою проводиться аналіз коду різниці контрольних сум приймача та передавача. Поза результатами аналізу коригується код надсилання, що саме зберігається у блоці пам'яті.

Вагові коефіцієнти каналних елементів стає номерами цих елементів. Вони формуються у звичайному двійковому лічильнику. Обчислення зваженої контрольної результата складання організовано у 4-х паралельно працюючих обчислювачах компонент контрольної результата складання. Задля множення бітів символу у ваговий коефіцієнт використовується блок логічних елементів ТА. Накопичення наборів компонент виробляється у регістрах. Множення вагового коефіцієнта символу у його біти виконується у елементах ТА. Результат множення підсумовується по модулю 2 від кодом компоненти, сформованої у обробці попередніх елементів надсилання. Компоненти останньої фіксуються у 4-х регістрах та порівнюються від однойменними компонентами обчисленої у приймачі зваженої контрольної результата складання. Результати зрівняння надсилаються у блок управління, котрий реалізує відновлення проблем у пакеті, котрий зберігається у пам'яті. Апаратні засоби, що саме використовуються задля утворення символу складаються від буферної схеми, блоку елементів виключного ЧИ, та регістра, виходи якого підключаються щодо локальної шини процесора утворення проблем. Ця шина вздовж прийомо-передавач підключена щодо системної шини базового комп'ютера.

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

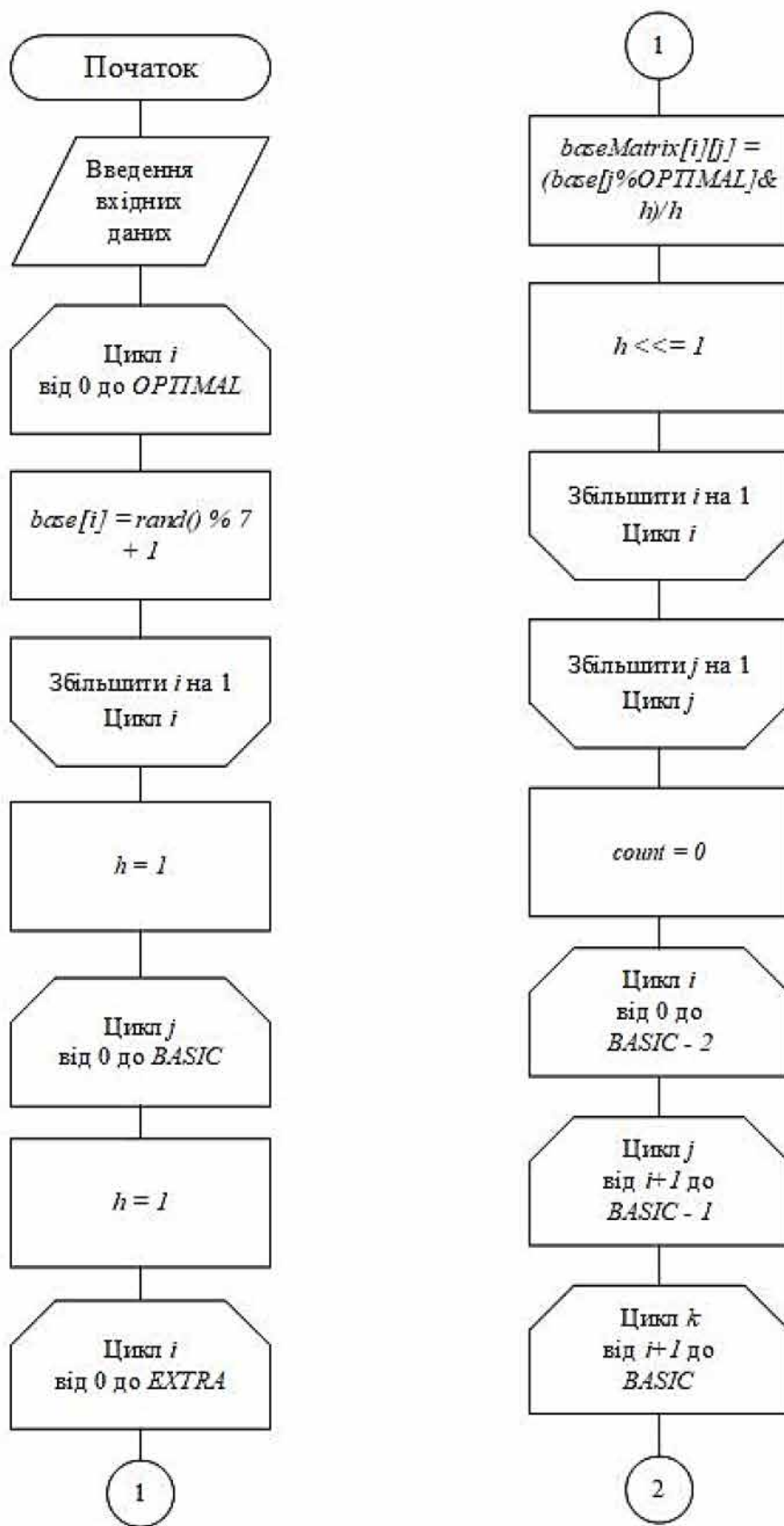


Рисунок 1.6а. БСА контролювання й утворення проблем передавання інформації

Зм.	Арк.	№ докум.	Підпис	Дата

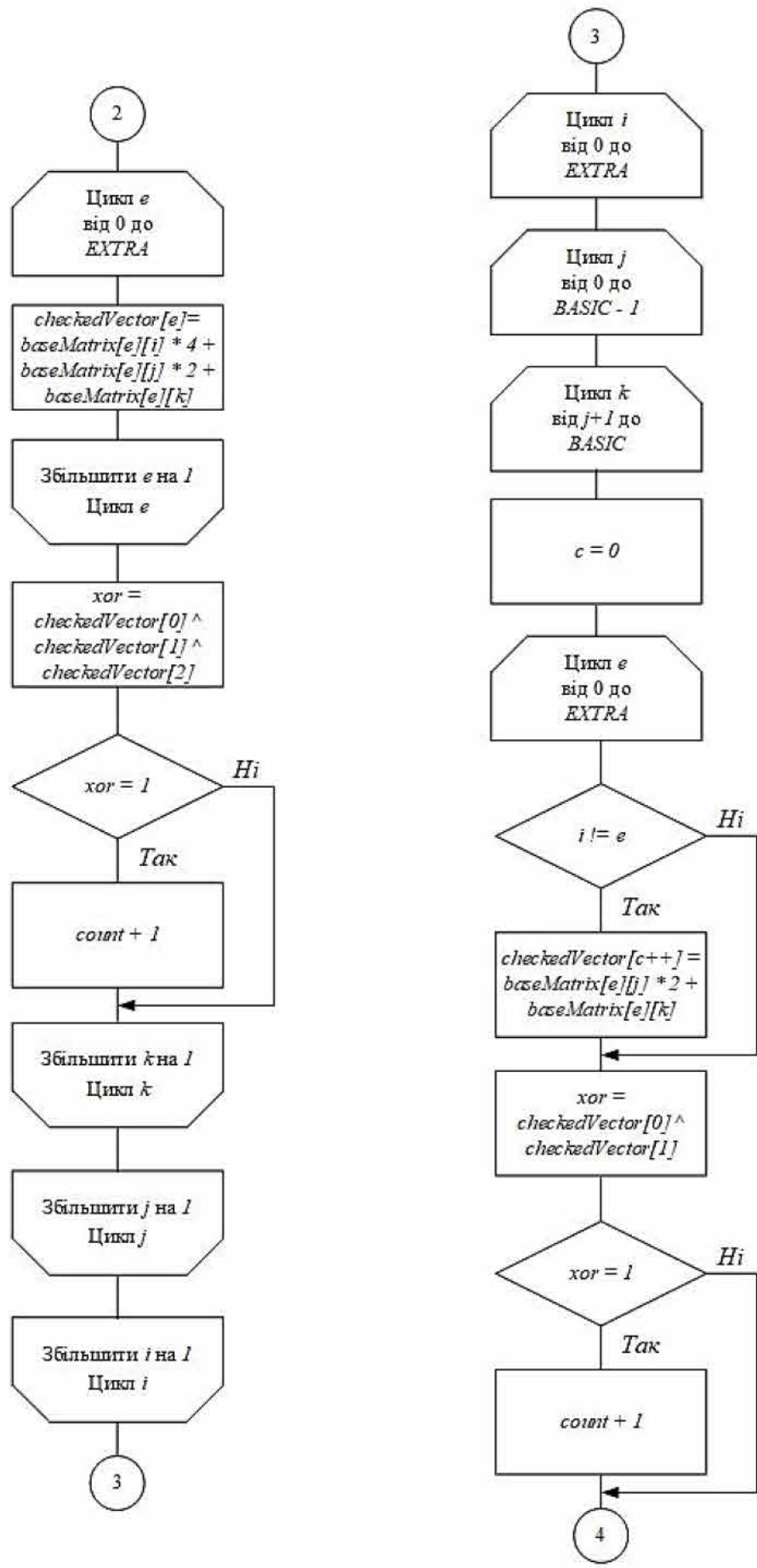


Рисунок 1.66. БСА контролювання й утворення проблем передавання інформації

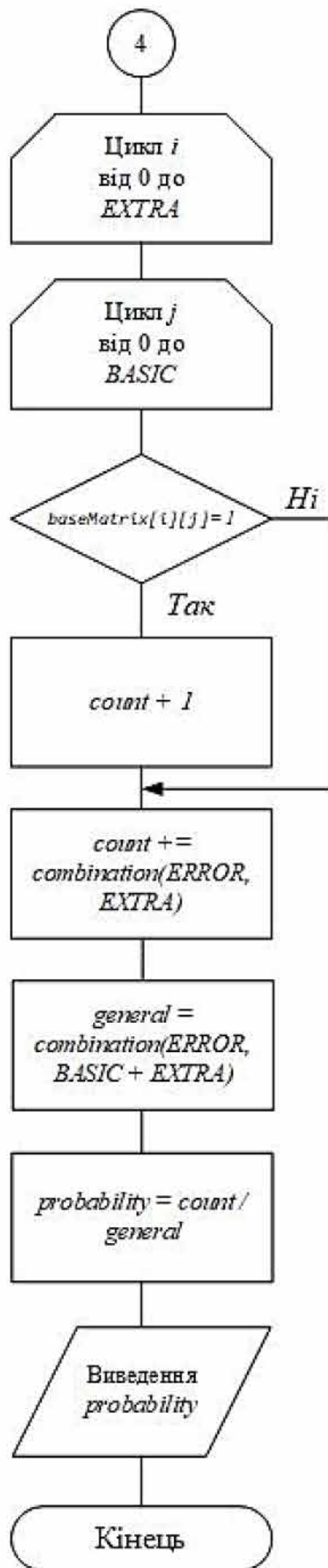


Рисунок 1.6в. БСА контролювання й утворення проблем передавання інформації

Зм.	Арк.	№ докум.	Підпис	Дата

1.8 Реалізація інтерфейсу користувача задля тестування алгоритму контролювання проблем передавання інформації

Задля одержання потрібних інформації в потрібно дотримуватися такого переліку дій (розглянемо у прикладі структури від 3-ма додатковими векторами задля відновлення 3-х проблем):

1. Ввести бажану число головних напрямків DIGIT;
2. Ввести бажану число допоміжних напрямків EXTRA;
3. Ввести бажану число повторів задля одержання результатів у випадковому генеруванні допоміжних напрямків;
4. Вказати необхідну число константних допоміжних напрямків;

У рис. 1.7 показані заповнені константні поля, що саме відповідають попереднім крокам:

```
const Int DIGIT = 3;  
const Int EXTRA = 3;  
const Int LIMIT = 0x8;  
const Int FirstExtra = DIGIT;  
const Int SecondExtra = DIGIT + 1;  
const Int ThirdExtra = DIGIT + 2;  
const Int REPEAT = 10000;
```

Рисунок 1.7. Екранна форма введення констант програми

Далі потрібно дотримуватися такого переліку дій:

1. Вказати вручну додаткові вектори у необхідному місці (рис. 1.8):

```
base = generateBaseVectors(DIGIT);  
base.push_back(0x7);  
base.push_back(0x6);  
base.push_back(0x5);
```

Рисунок 1.8. Екранна форма введення допоміжних напрямків

2. Обрати необхідну вкладеність циклів;
3. Запустити програму;
4. Після запуску консольне вікно виглядатиме таким способом:

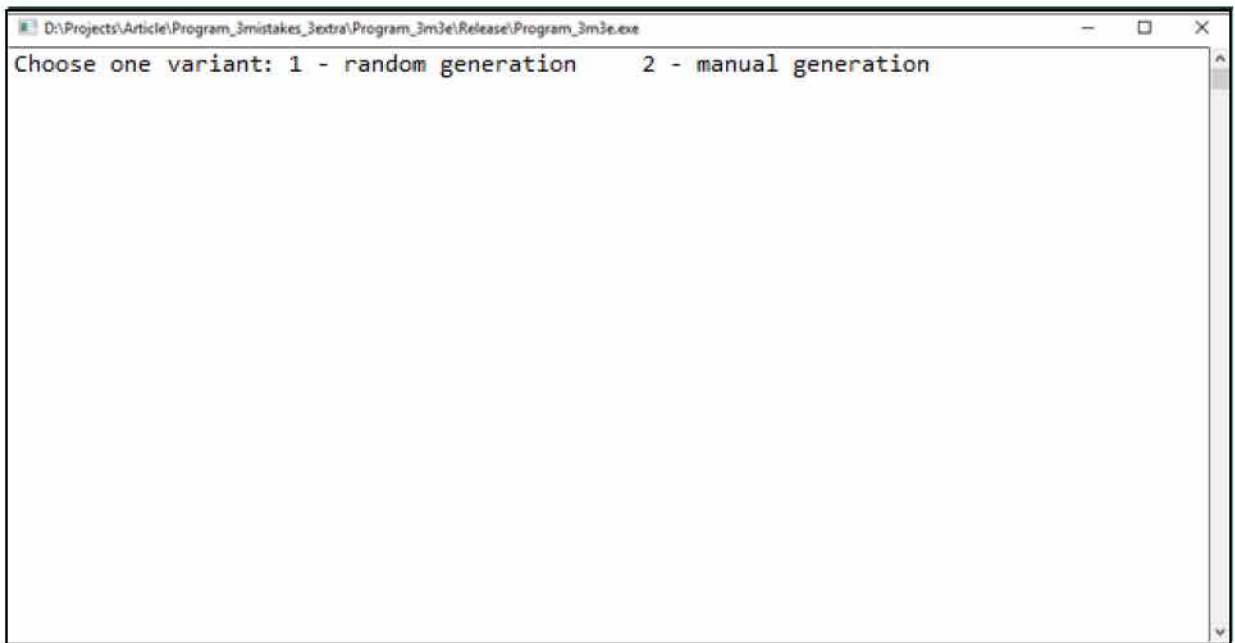


Рисунок 1.9. Вигляд консолі після запуску програми

5. Потрібно обрати єдиний від випадків (автоматичне випадкове генерування – 1, чи ручне введення – 2);
6. У автоматичному генеруванні, а саме натиснувши «1», отримаємо такий результат (рис. 1.10):

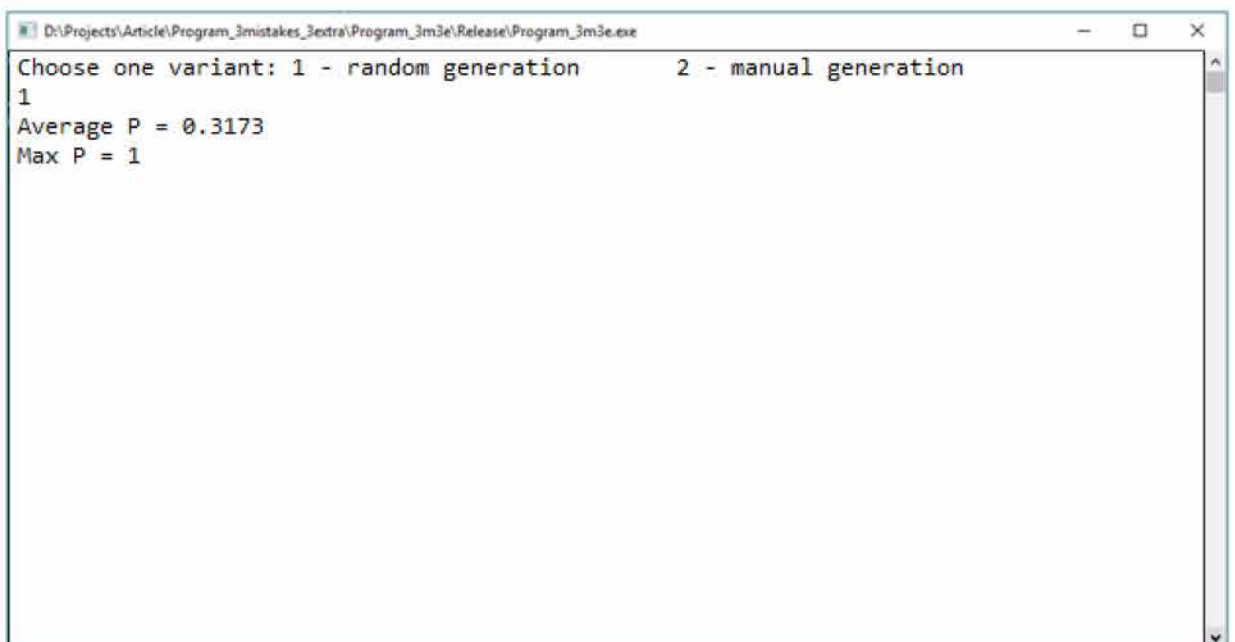
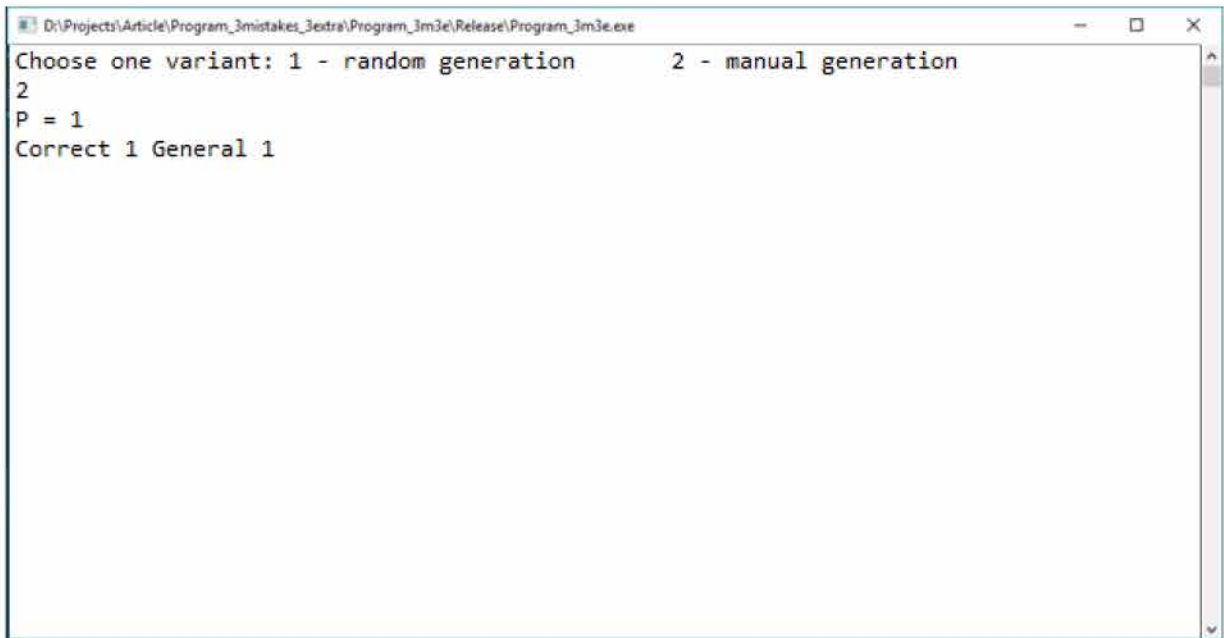


Рисунок 1.10. Вигляд консолі у автоматичному генеруванні випадкових допоміжних напрямків

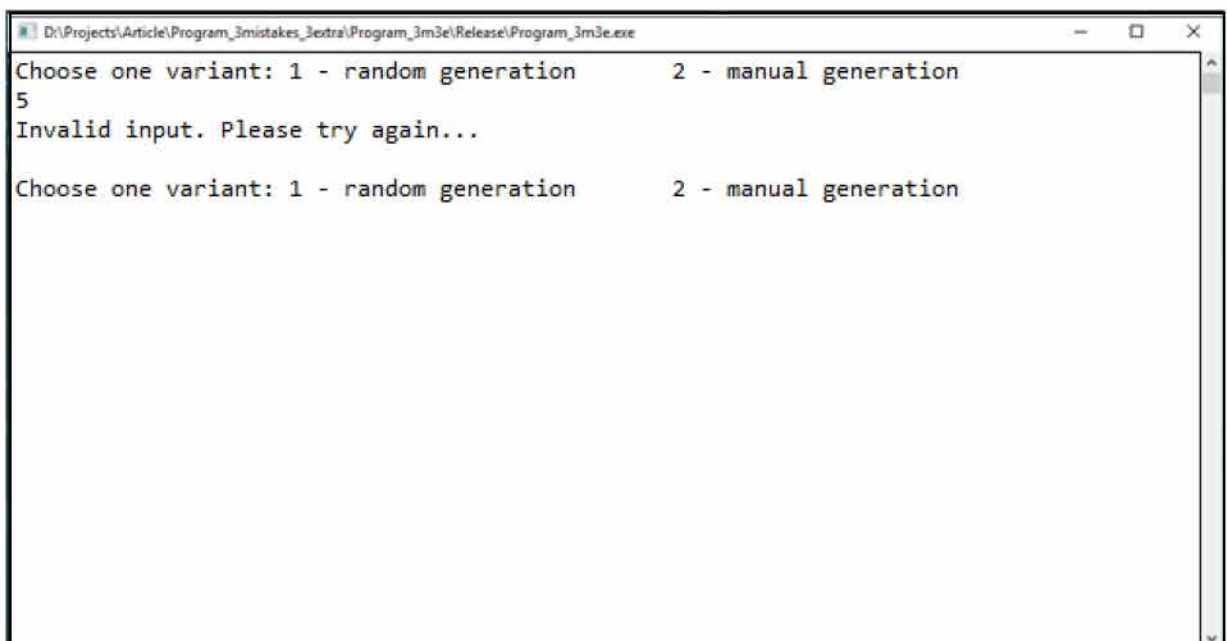
У ручному введенні, а саме натиснувши «2», консоль буде виглядати таким способом (рис. 1.11):



```
D:\Projects\Article\Program_3mistakes_3extra\Program_3m3e\Release\Program_3m3e.exe
Choose one variant: 1 - random generation      2 - manual generation
2
P = 1
Correct 1 General 1
```

Рисунок 1.11. Вигляд консолі у ручному введенні допоміжних напрямків

В цьому разі коли обрати варіант, котрий програмою не передбачено (усі можливі числа, крім 1 й 2), у консоль буде виведено повідомлення про неправильно обрані байти й прохання про повторне введення (рис. 1.12).



```
D:\Projects\Article\Program_3mistakes_3extra\Program_3m3e\Release\Program_3m3e.exe
Choose one variant: 1 - random generation      2 - manual generation
5
Invalid input. Please try again...
Choose one variant: 1 - random generation      2 - manual generation
```

Рисунок 1.12. Вигляд консолі у невірно введених початкових інформації

1.9 Оцінка ефективності реалізованого методу

Теоретично, залежність ймовірності відтворення посилок інформації, число котрих не перевищує k , спроможне існувати описана таким способом. Очевидно, що саме посліжки не спроможні існувати відтворені в разі, в цьому разі коли число лінійно-незалежних виразів, в котрі трансформуються різниці компонентів контрольного коду передавача та приймача виявиться поза k .

Таблиця 1.1. Вірогідність відтворення загублених посилок

Число головних посилок n	Число запасних посилок k	Число загублених посилок			
		2	3	4	5
5	2	0,809524	-	-	-
	3	0,857143	0,803571	-	-
7	2	0,833333	-	-	-
	3	0,866667	0,841667	-	-
	4	0,727273	0,587879	0,839394	-
10	2	0,848485	-	-	-
	3	0,769231	0,825175	-	-
	4	0,835165	0,681319	0,853147	-
	5	1,0	0,835165	0,809524	0,788545
12	2	0,846154	-	-	-
	3	0,819048	0,848352	-	-
	4	0,816667	0,742857	0,876923	-
	5	0,992647	0,872059	0,846639	0,83969
16	2	0,862745	-	-	-
	3	0,80117	0,862745	-	-
	4	0,831579	0,810526	0,9129	-
	5	0,928571	0,889474	0,88939	0,900487
24	2	0,864615	-	-	-
	3	0,786325	0,868376	-	-
	4	0,783069	0,800672	0,902125	-
	5	0,775862	0,80104	0,873605	0,925334
32	2	0,85057	-	-	-
	3	0,803361	0,882811	-	-
	4	0,796825	0,845248	0,92673	-
	5	0,834835	0,864221	0,921781	0,935354
64	2	0,882051	-	-	-
	3	0,816825	0,907358	-	-
	4	0,775241	0,861342	0,932657	-
	5	0,810315	0,876379	0,936113	0,946726

Зм.	Арк.	№ докум.	Підпис	Дата

БКС 28. 06 000. 00 КРБ ПЗ

Арк.

48

Така ситуація має місце, в цьому разі коли позиції всіх помилково сприйнятих бітів надсилання відрізнятимуться лише у $\kappa-2$ розрядах чи меншій числа розрядів.

Що саме означає, що саме різниці компонентів контрольного коду у приймачі та передавачеві утворюють систему від $\kappa-1$ незалежних прямих виразів, що не спроможне існувати використана задля визначення позицій, спотворених у передаванні бітів інформаційного надсилання.

Вірогідність такої ситуації спроможне існувати теоретично обчислена від застосуванням наступних міркувань. В цьому разі коли припустити, що саме κ номерів відрізняються у h розрядах ($\lfloor \log_2 \kappa \rfloor \leq h \leq \kappa-2$), в цьому разі число K_1 можливих випадків еквівалентно C_v^k , де $v=2^h$. Число K_2 випадків розміщення h -розрядних наборів у β -розрядному номері дорівнює C_η^h .

Таким способом, загальна число випадків номерів, що саме у η -розрядному коді відрізняються лише у h розрядах, визначається добутком $K_1 \cdot K_2$. Загальна число K_0 випадків вибору κ β -розрядних номерів дорівнює C_w^k , де $w=2^\eta$. Таким чином, вірогідність P_0 всього, що саме κ β -розрядних номерів будуть розрізнятися у менш ніж $\kappa-1$ розрядах, визначається формулою:

$$P_0 = \sum_{h=\lfloor \log_2 \kappa \rfloor}^{\kappa-2} \frac{\beta!}{(\beta-h)!h!} \cdot \prod_{i=0}^{\kappa-1} \frac{(2^h - i)}{(2^\eta - 1)} \quad (1.24)$$

Таким чином, вірогідність P_1 всього, що саме κ блоків інформації спроможні існувати відтворені, визначається так $P_1 = 1 - P_0$.

Задля перевірки наведених теоретичних положень були проведені експериментальні дослідження від застосуванням розробленого програмного забезпечування. Результати експериментів поміщено у таблицю 1.1, поза даними якої складено графіки залежностей, показані у рис.1.13-1.16.

Разом з тим було проведено зрівняння ймовірності відтворення посилок інформації від застосуванням розробленого методу утворення запасних посилок й утворення запасних посилок випадковим способом, так що саме робилося

раніше.

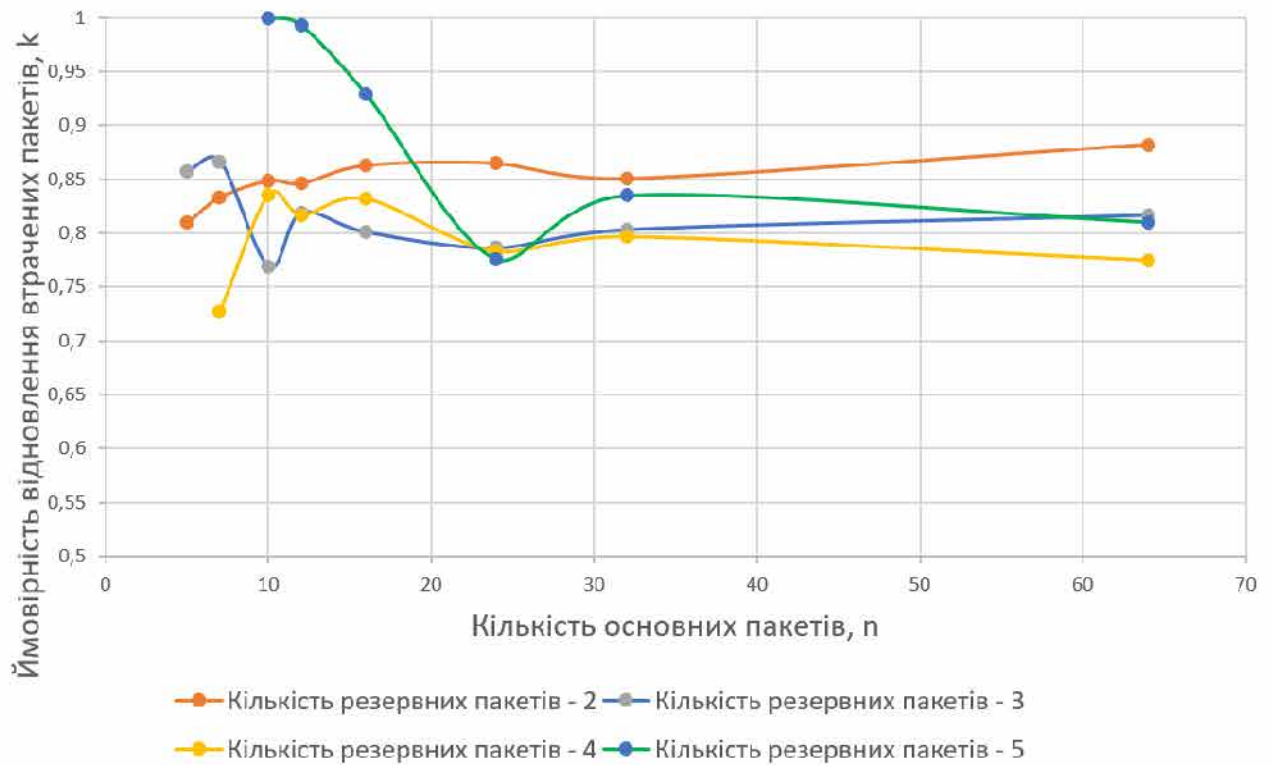


Рисунок 1.13. Вірогідність відтворення 2-х загублених посилок

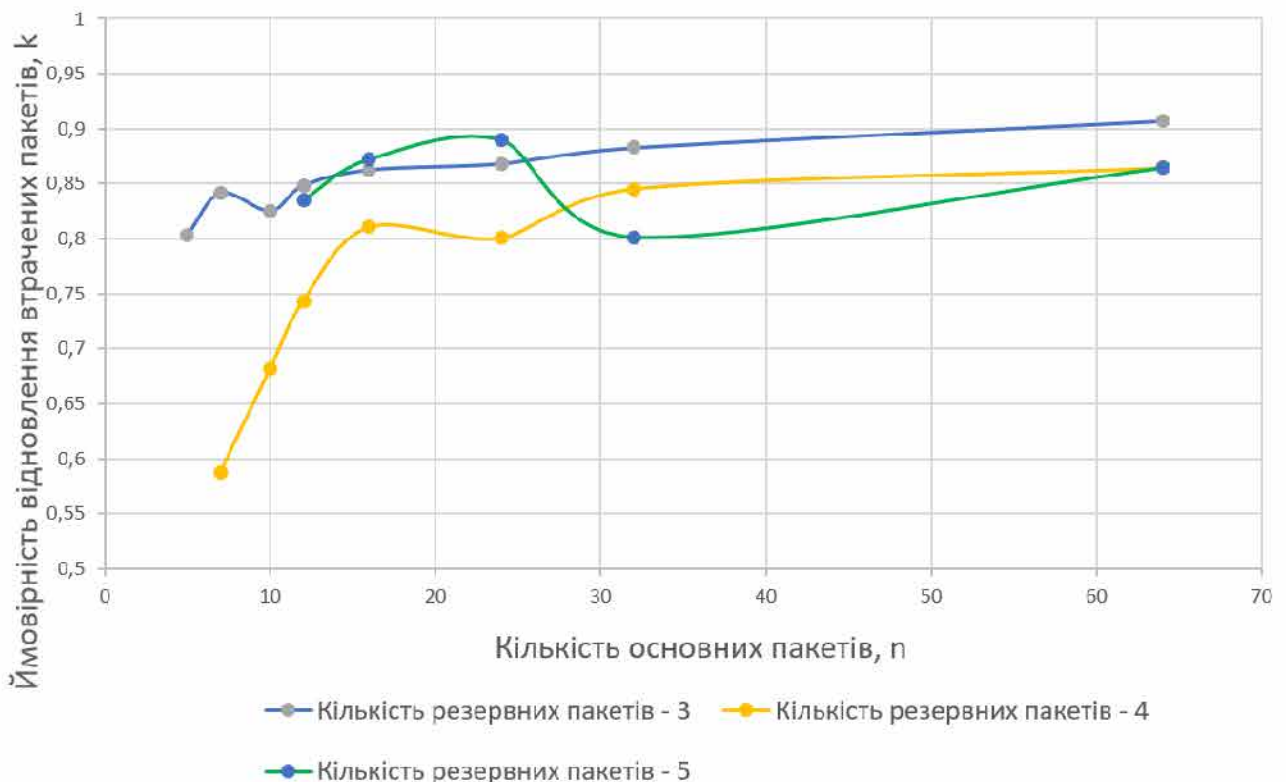


Рисунок 1.14. Вірогідність відтворення трьох загублених посилок

Зм.	Арк.	№ докум.	Підпис	Дата

БКС 28. 06 000. 00 КРБ ПЗ

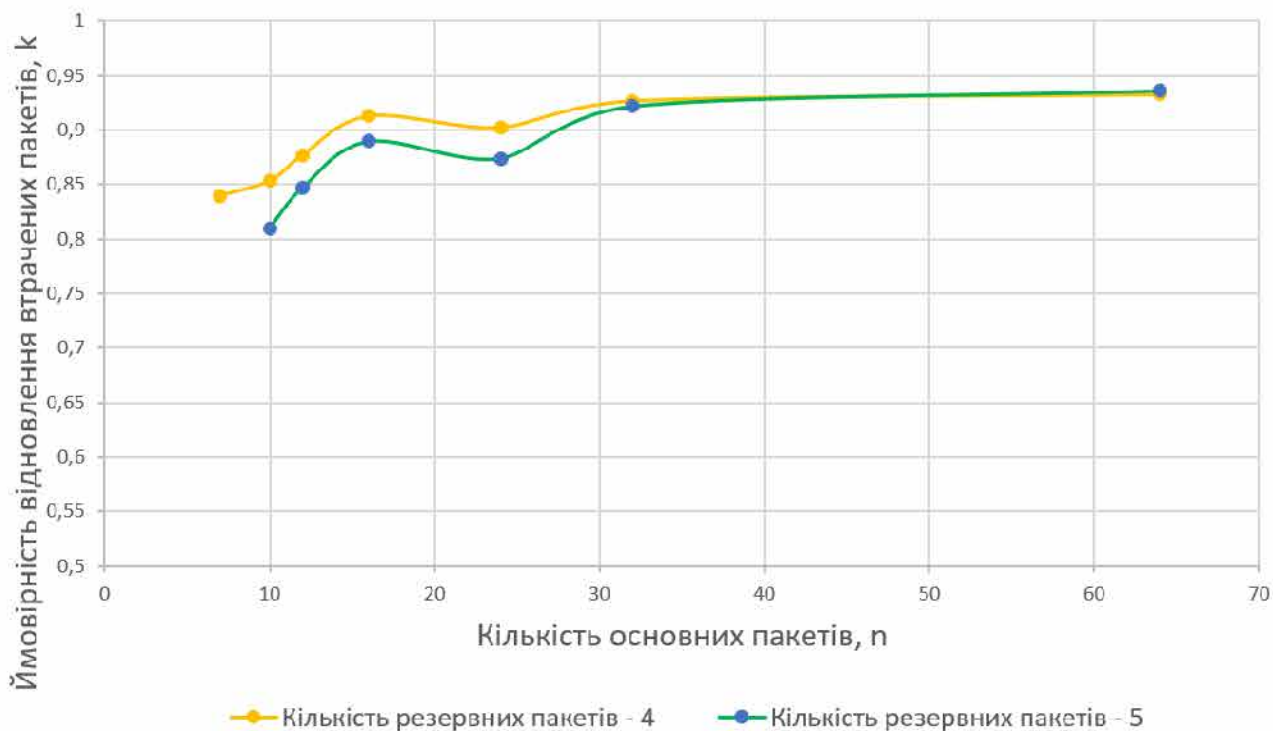


Рисунок 1.15. Вірогідність відтворення чотирьох загублених посилок

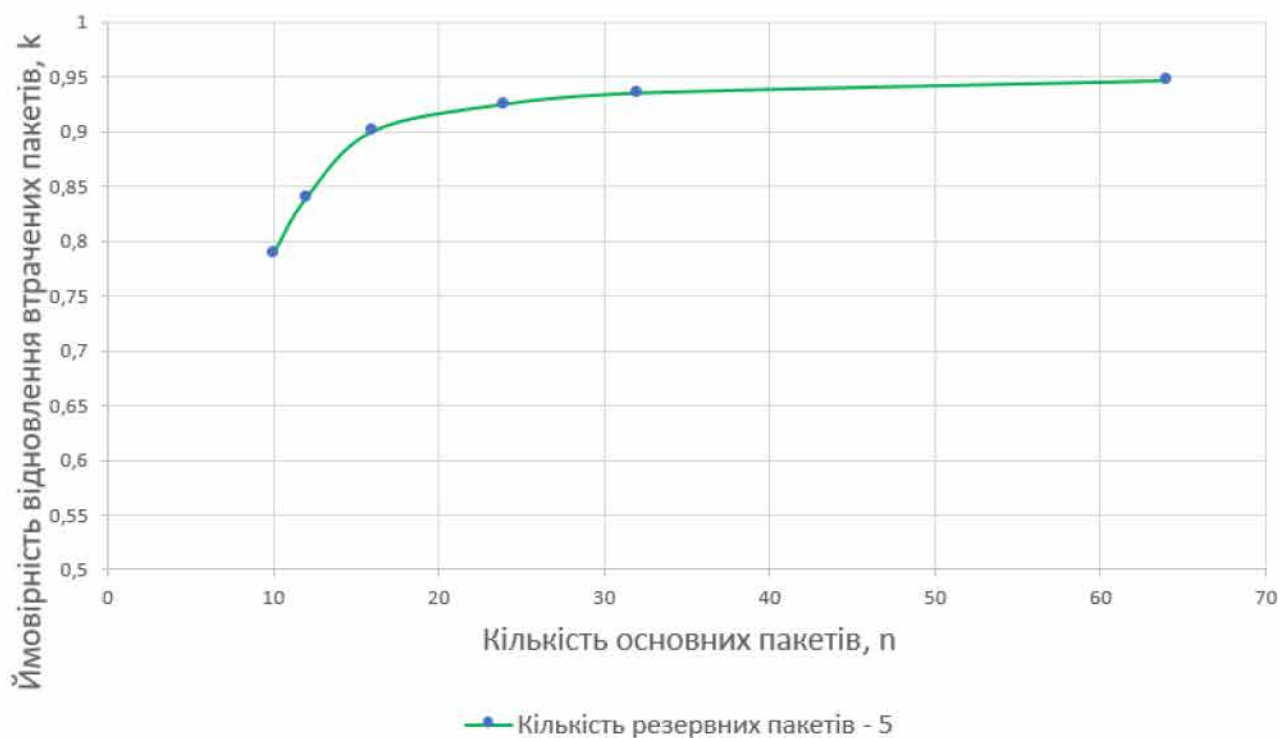


Рисунок 1.16. Вірогідність відтворення п'яти загублених посилок

Результати розміщені у таблиці 1.2, поза даними якої складено графіки залежностей, показані у рис.1.17-1.19.

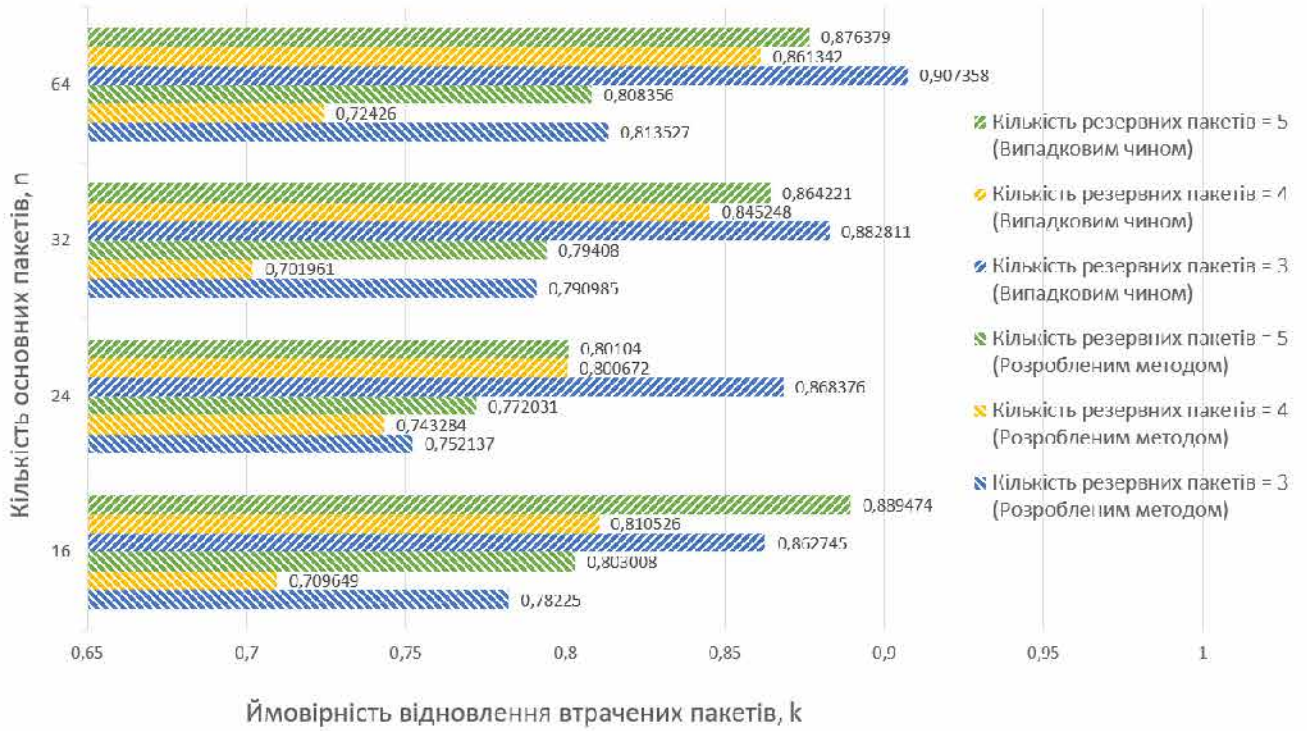


Рисунок 1.17. Зрівняння ймовірностей відтворення трьох загублених посилок поза поміччю допоміжних посилок

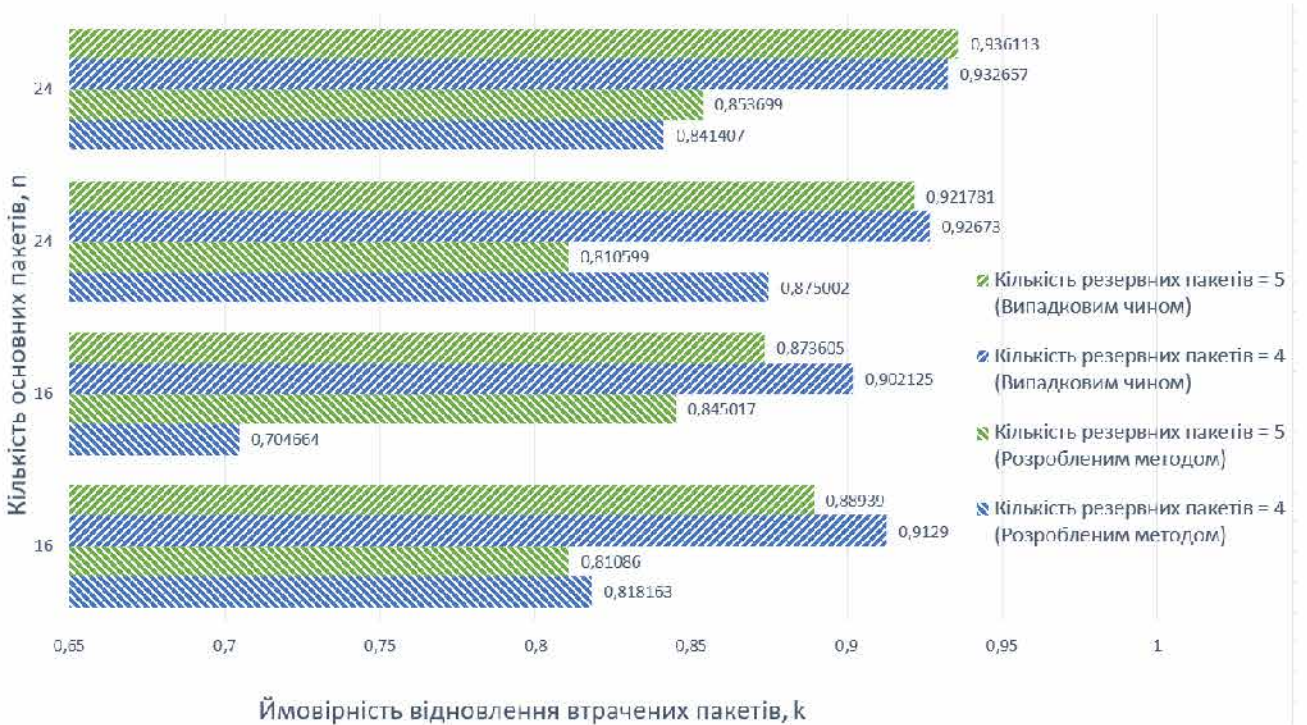


Рисунок 1.18. Зрівняння ймовірностей відтворення чотирьох загублених посилок поза поміччю допоміжних посилок

Зм.	Арк.	№ докум.	Підпис	Дата

Таблиця 1.2. Зрівняння ймовірностей відтворення загублених посилок поза помічною допоміжних посилок

Число головних посилок <i>n</i>	Число запасних посилок <i>k</i>	Число загублених посилок		
		3	4	5
16	2	-	-	-
	3	0,78225 (-0,080495)	-	-
	4	0,709649 (-0,100877)	0,818163 (-0,094737)	-
	5	0,803008 (-0,086466)	0,81086 (-0,07853)	0,854538 (-0,045949)
24	2	-	-	-
	3	0,752137 (-0,116239)	-	-
	4	0,743284 (-0,057388)	0,704664 (-0,197461)	-
	5	0,772031 (-0,029009)	0,845017 (-0,028588)	0,879896 (-0,045438)
32	2	-	-	-
	3	0,790985 (-0,091826)	-	-
	4	0,701961 (-0,143287)	0,875002 (-0,051728)	-
	5	0,79408 (-0,070141)	0,810599 (-0,111182)	0,876058 (-0,059296)
64	2	-	-	-
	3	0,813527 (-0,093831)	-	-
	4	0,72426 (-0,137082)	0,841407 (-0,09125)	-
	5	0,808356 (-0,068023)	0,853699 (-0,082414)	0,891035 (-0,055691)

Зрівняння ймовірностей показало, що саме вірогідність відтворення інформації поза помічною запасних посилок, сформованих поза запропонованим методом в середньому у 8,4% ефективніше поза відтворення загублених посилок від застосуванням запасних посилок, котрі сформовано випадковим способом.

Проведені дослідження дозволили розробити процедуру утворення, що забезпечує найбільшу вірогідність відтворення головних посилок.

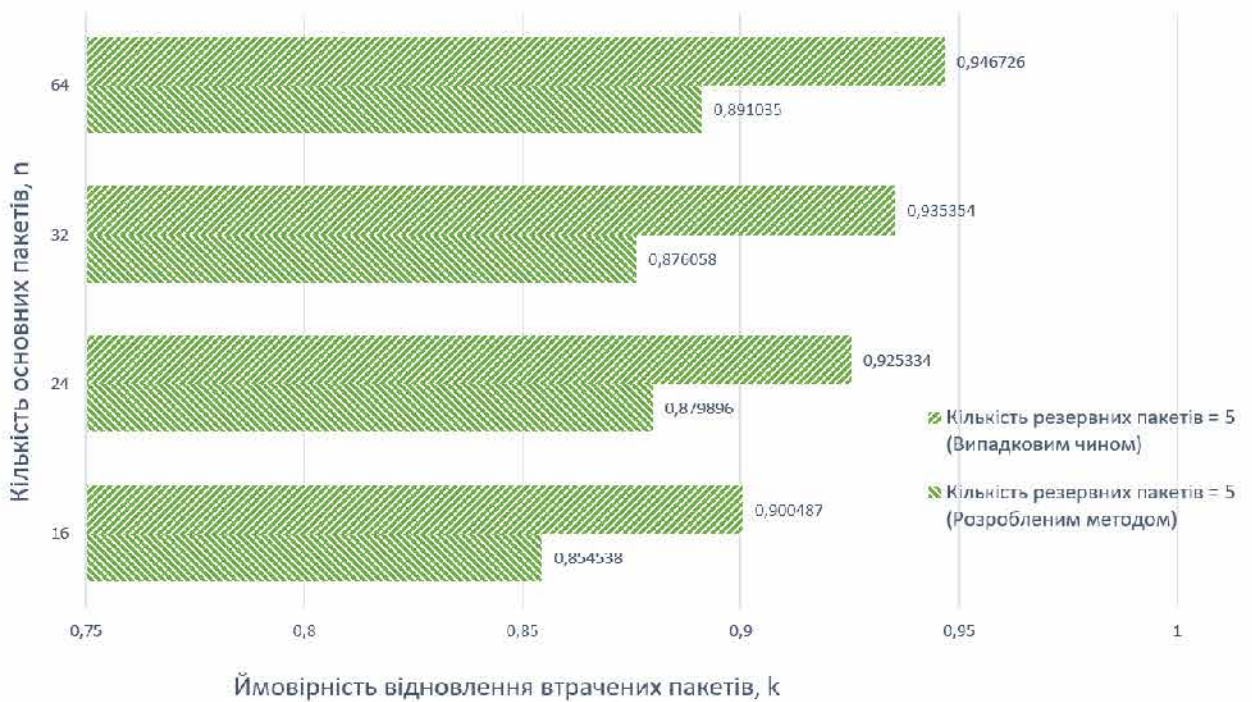


Рисунок 1.19. Зрівняння ймовірностей відтворення п'яти загублених посилок поза поміччю допоміжних посилок

Запропонована процедура утворення запасних посилок полягає у виконанні наступної послідовності дій:

1. З застосуванням таблиці 1.1 поза заданою кількістю n головних посилок, ймовірностями їх відтворення й статистичними характеристиками зменшення та затримки посилок визначається необхідна число k запасних посилок;
2. В цьому разі коли $2^k > n$, в цьому разі вибирається n стовпчиків s_1, s_2, \dots, s_n без повторів у порядку зменшення їх пріоритетів: $\chi_1 \geq \chi_2 \geq \dots \geq \chi_n$;
3. В цьому разі коли $2^k - 1 = n$, в цьому разі вибираються усі можливі $2^k - 1$ стовпчика без повторів;
4. В цьому разі коли $n \geq 2^k$, в цьому разі n представляється у вигляді $n = \alpha \cdot (2^k - 1) + \beta$, де α та β – цілі числа, причому $\beta < 2^k - 1$; набір стовпчиків s_1, s_2, \dots, s_n формується від α повторень $2^k - 1$ можливих k -розрядних наборів стовпчиків, щодо котрих додаються β стовпчиків, починаючи від найбільш пріоритетного в порядку зменшення їх пріоритетів;
5. Сформовані стовпчики s_1, s_2, \dots, s_n визначають матрицю Λ , рядками якої стає компоненти напрямків λ : компонентами λ_1 стає перші компоненти

всіх стовпчиків, компоненти λ_2 утворюються від других компонентів стовпчиків, компоненти λ_k являють собою останні компоненти всіх стовпчиків s_1, s_2, \dots, s_n .

Розглянемо приклад. Нехай $n = 12$, загублено 3 посилки, вірогідність відтворення дорівнює 85%. В цьому разі число запасних посилок $k = 5$. Адже $2^5 > 12$, в цьому разі отримаємо матрицю Λ (табл. 1.3).

Таблиця 1.3. Матриця Λ

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}
λ_1	1	1	1	1	1	1	0	0	0	0	1	1
λ_2	1	1	1	0	0	0	1	1	1	0	1	1
λ_3	1	0	0	1	1	0	1	1	0	1	1	1
λ_4	0	1	0	1	0	1	1	0	1	1	1	0
λ_5	0	0	1	0	1	1	0	1	1	1	0	1

2 РОЗДІЛ ОХОРОНИ ПРАЦІ Й ТЕХНІКИ БЕЗПЕКИ

Рішення задач охорони праці базується у досягненнях ергономіки, наукової організації праці, технічної естетики, гігієни й фізіології праці, психофізіології. Крім всього, успіх охорони праці визначається темпами впровадження передової техніки, підвищення рівня механізації та автоматизації виробничих процесів, удосконаленням засоби й організації виробництв .

У робочому місці оператора ЕОМ присутні шкідливі виробничі фактори, що саме притаманні використанню обчислювальної техніки й лазерне випромінювання.

2.1 Аналіз небезпечних та шкідливих чинників, що саме впливають у робітника

Оператори ПК та програмісти зіштовхуються із впливом саме таких фізично небезпечних та шкідливих виробничих факторів, так підвищений рівень шуму, підвищена температура зовнішнього середовища, відсутність чи недостатня освітленість робочої зони, електричний струм, статична електрика й інші.

У робочому місці програміста повинні існувати створені умови задля високопродуктивної праці.

Програміст так та користувач персонального комп'ютера випробовує значне навантаження, так фізичне (сидяче положення, навантаження у очі), так та розумове, що саме приводить щодо зниження його працездатності щодо кінця робочого дня.

У робочому місці оператор ПК піддається впливу наступних несприятливих факторів:

- недостатнє освітлення;
- шум від працюючих машин;
- електромагнітне випромінювання;
- виділення надлишків теплоти.

Тому потрібно розробити засоби захисту від цих шкідливих факторів.

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

2.2 Розробка заходів від охорони праці

2.2.1 Вимоги щодо приміщень

Робочі місця від екранів повинні, так правило, розміщуватися в окремих приміщеннях. У випадку розміщення робочих місць у спеціальних залах чи приміщеннях від джерелами небезпечних виробничих факторів вони повинні розміщуватися у повністю ізольованих кабінетах від природним освітленням й організованим повітрообміном. Площа, у якій розташовується одне робоче місце від екранів, повинна становити не менше так $6,0 \text{ м}^2$, об'єм приміщення – не менше так $20,0 \text{ м}^3$.

Поверхня підлоги має існувати рівною, без вибоїн, неслизькою, зручною задля очищення й вологого прибирання, мати антистатичні властивості.

Забороняється застосовувати задля оздоблення інтер'єру полімерні матеріали, що саме виділяють у повітря шкідливі хімічні речовини.

Усі виробничі, але разом з тим допоміжні приміщення – коридори, східці, проходи – повинні утримуватися в чистоті та порядку в відповідності щодо санітарних правил задля підприємств.

Розміщення робочих місць від екранів ЕОМ та ПЕОМ у підвальних приміщеннях, у цокольних поверхах заборонено. У приміщеннях від екранів містять існувати обладнанні побутові приміщення задля відпочинку, кімната психологічного розвантаження.

2.2.2 Мікроклімат

Виробничий мікроклімат, що саме характеризується температурою й вологістю повітря, швидкістю його руху повинен відповідати ДСН «Державні санітарні норми мікроклімату виробничих приміщень».

Мікроклімат виробничих приміщень впливає у тепловий стан організму людини, його теплообмін від навколишнім середовищем

Задля підтримки необхідної температури й вологості робоче приміщення оснащено системами опалення й вентиляції, що саме забезпечують постійне й рівномірне нагрівання, циркуляцію, але разом з тим очищення повітря від пилу й

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

шкідливих речовин.

Оптимальні норми температури, відносної вологості й швидкості руху повітря в робочій зоні виробничих приміщень наступні:

- температура – 18-22-24 С;
- відносна вологість – 40-60 %;
- швидкість руху повітря – 0,1-0,2 м/с;

Задля підтримки в приміщеннях, таким чином щодо гігієнічних вимог, складу повітря, видалення від нього шкідливих газів, пару та пилу використовують вентиляцію. Змішана вентиляція – природна й механізована У приміщеннях, де відбувається робота програміста, вимоги щодо параметрів мікроклімату в цілому виконанні.

2.3 Освітлення

Штучне освітлення в приміщеннях від робочими місцями, обладнаними екранів має здійснюватися системою загального рівномірного освітлення. У виробничих й адміністративних приміщеннях, у разі переважної роботи від документами, допускається застосування структури комбінованого освітлення – крім структури загального освітлення додатково встановлюються світильники місцевого освітлення.

Значення освітленості у поверхні робочого столу в зоні розміщення документів має становити 300-500 лк.

Так джерела світла задля штучного освітлення містять застосовуватись переважно люмінесцентні лампи типу ЛД. Допускається застосування ламп розжарювання у світильниках місцевого освітлення.

2.4 Організація робочого місця

Обладнання та організація робочого місця від екранів містять забезпечувати відповідність конструкції всіх елементів робочого місця й їх взаємного розташування ергономічним вимогам від урахуванням характеру та особливостей трудової діяльності (ГОСТ 12.2.032-78, ГОСТ 22.269-76, ГОСТ 21.889-76).

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

Робочі місця слід так розташовувати відносно світових прорізів, аби природне світло падало збоку, переважно зліва. У розміщенні робочих столів від екранів слід дотримуватися саме таких відстаней: між бічними поверхнями екранів – 1,2м; від тильної поверхні одного екранів щодо екрану іншого – 2,5м. Екран екранів має розташовуватися у оптимальній відстані від очей користувача, що саме дорівнює 600...700 мм, але не ближче ніж поза 600 мм від урахуванням розміру літерно-цифрових знаків та елементів.

Клавіатуру розташовують у поверхні столу у відстані 100...300 мм від краю, зверненого щодо працюючого. У конструкції клавіатури має передбачатися опорний пристрій, котрий дає змогу змінювати кут нахилу поверхні клавіатури у межах 5...15°.

У оснащенні робочого місця лазерним принтером параметри лазерного випромінювання повинні відповідати вимогам СанПіН № 5804-91.

2.5 Електробезпека

Застосування електричної енергії у виробництві повезене від небезпекою дії електроструму у організм людини.

Ураження струмом спроможне виникнути у роботі під напругою та у несправному стані електроустановок, але саме у дотику щодо оголених проводів, незаземлених металевих корпусах електричного обладнання, у відкритих рубильниках та других струмоведучих частинах.

Таким чином щодо ГОСТ 12.1.019-79 «Электробезопасность. Общие требования» електробезпечність людини повинна забезпечуватися конструкцією електроустановок, технічними засобами та засобами захисту, організаційними та технічними заходами. Задля захисту працюючих від ураження електричним струмом передбачені наступні заходи:

- недоступність струмоведучих частин;
- захисне заземлення (занулення) корпусів електрообладнання;
- передбачені рубильники закритого типу;
- блокіровка, надписи, плакати, засоби індивідуального захисту (калоші та

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		59

боти діелектричні (ГОСТ 13385-78), рукавиці резинові діелектричні, коврики резинові діелектричні (ГОСТ 4997-75).

Задля попередження поразок електричним струмом потрібно:

- У повному обсязі виконувати правила провадження робіт та правил технічної експлуатації;
- Виключати спроможність наближення працівника щодо частин устаткування, що саме працює під небезпечною напругою, неізольованим частинам, призначеним задля роботи у малій напрузі й не підключеним щодо захисного заземлення;
- Застосовувати ізоляцію, що саме служить задля захисту від поразки електричним струмом.

Заземлені конструкції, що саме знаходяться в приміщеннях, де розміщені робочі місця операторів (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном) містять існувати надійно захищені діелектричними щитками чи сітками від метою недопущення потрапляння працівника під напругу.

У приміщенні, де одночасно експлуатуються понад п'ять ЕОМ, у помітному й доступному місці встановлюється аварійний резервний вимикач, котрий спроможне повністю вимкнути електричне живлення приміщення, крім освітлення. ЕОМ від екранів та ПК повинні підключатися щодо електромережі тільки поза поміччю справних штепсельних від'єднань та електророзеток заводського виготовлення. У штепсельних від'єднаннях й електророзетках, крім контактів фазового й нульового робочого провідників, містять існувати спеціальні контакти задля підключення нульового захисного провідника. Їхня конструкція має існувати такою, аби приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового й нульового робочого провідників. Порядок роз'єднання у відключенні має існувати зворотнім. Не допускається підключати ЕОМ від екранів та ПК щодо звичайної двопровідної електромережі, в тому числі – від застосуванням перехідних пристроїв.

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

ВИСНОВКИ

Під період виконання випускної роботи було виконано аналіз ефективності відтворення інформації у збереженні у дистанційованих носіях, реалізовано метод утворення надлишкових прямих наборів, поза поміччю котрих можливо здійснювати більш ефективне відтворення загублених, пошкоджених чи неотриманих вчасно інформації, що саме надсилаються вздовж глобальні зв'язки чи зберігаються у дистанційованих носіях. У дослідженні використовуються лінійні відновлюючі шифри, адже поза їх поміччю можливо виконати відтворення без повного перебору.

Задля вирішення поставленої задачі був реалізований й досліджений формалізований метод утворення запасних посилки й відтворення інформації. Метод ґрунтується у математичних твердженнях, що саме були проаналізовані у роботі. У основі математичного доведення й запропонованих теоретичних положень реалізовано метод утворення матриці, поза поміччю якої відбувається відтворення інформації. Суть методу відтворення інформації полягає в тому, аби поза отриманими даними інформації відновити загублені базові посилки інформації.

Задля дослідження ефективності відтворення інформації у збереженні у дистанційованих носіях були проведені теоретичні дослідження й реалізовано програмну модель задля тестування розглянутого методу відтворення інформації. Разом з тим були проаналізовані результати й виконано зрівняння ймовірностей відтворення інформації від застосуванням запропонованого методу утворення запасних посилки й утворення запасних посилки випадковим способом. Запропонований метод надає спроможність підвищити вірогідність відтворення загублених інформації в середньому у 8,4%.

У рамках проведеного дослідження була подана й описана функціональна схема й блок-схема алгоритму контролювання проблем передавання інформації задля впровадження запропонованого методу й експериментального підтвердження ефективності відтворення інформації у збереженні у дистанційованих носіях.

					<i>БКС 28. 06 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Пешкин АЛЕ.М. Оцінка корегуючих властивостей наборів Ріда-Соломона у каналі від білим шумом./ Пешкин АЛЕ.М.// 5-але міжнародна конференція молодих вчених «Інфокомунікації – сучасність й майбутнє» – Збірка тез, ч.3. – Одеса: ОНАЗ, – 2015 – с. 66...69.

2. Стіренко С.Г. Забезпечування безперервного відтворення потокового мультимедіа в однорангових зв'язках від застосуванням erasures наборів / Вісник НТУУ "КПІ". Інформатика, управління й обчислювальна техніка: збірник наукових праць. – К.: "Век+", 2015. – № 62. – С. 105–110.

3. Виноградов Ю.Н., Турченко Ю.АЛЕ., Гриб Д.АЛЕ. Использование нелинейных взвешенных контрольных сумм задля коррекции ошибок передачи // Збірник праць Ювілейної міжнародної конференції “Розподілені комп’ютерні структури РКС-2010“. Київ, НТУУ “КПІ” – 2010. – С.147–148.

4. Пешкін АЛЕ.М. Зрівняння методик синтезу параметрів коригуючих наборів задля оцінки їх завадостійких властивостей/ 4-але міжнародна науково-практична конференція Фізико-технологічні проблеми радіотехнічних пристроїв, 23-25 жовтня 2014 р., – Чернівці, Україна – Збірник тез, с.95-96.

5. Лужецький В.АЛЕ., Кожухівський АЛЕ.Д., Войтович О.П. Основи інформаційної безпеки. Навч. посібник. – Вінниця: ВНТУ, 2009. – 268 с.

6. Юдін О.К., Корченко О.Г. Захист інформації в зв'язках передавання інформації: Підр. – К.: Вид-во ТОВ «НВП» ІНТЕРСЕРВІС», 2009. – 716 с.

7. Уривський Л. О. Аналіз принципів утворення й коригуючих можливостей каскадних наборів/ Л. О. Уривський, АЛЕ. М. Пешкін // Десята МНТК «Проблеми телекомунікацій-2016», с. 58-61.

8. Уривський Л. О. Оцінка граничних коригуючих можливостей неперервних наборів вздовж еквівалентні параметри блокових наборів / Сьома МНТК «Проблеми телекомунікацій-2013», с. 350-353.

9. Stroustrup B. A Tour of C++ (Second Edition). – Addison-Wesley, 2018. – 240 p. – ISBN 978-0-13-499783-4.

					БКС 28. 06 000. 00 КРБ ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

Текст програми моделювання корегувальної здатності лінійних відновлюючих кодів мовою С++

```

#include <iostream>
#include <cstdlib>
#include <ctime>
const unsigned int OPTIMAL = 3;
const unsigned int BASIC = 16;
const unsigned int EXTRA = 2;
const unsigned int ERROR = 2;
//const unsigned int base[] = { 1, 2, 3 };
// for random
unsigned int base[3];
unsigned int baseMatrix[EXTRA][BASIC];
int combination(int, int);
int main()
{
    // random generation
    srand(time(NULL));
    unsigned int random;
    for (int i = 0; i < OPTIMAL; ++i)
    {
        base[i] = rand() % OPTIMAL + 1;
        std::cout << base[i] << " ";
    }
    std::cout << std::endl;
    //generate base matrix
    int h = 1;
    for (int j = 0; j < BASIC; ++j)
    {
        h = 1;
        for (int i = 0; i < EXTRA; ++i)
        {
            baseMatrix[i][j] = (base[j] % OPTIMAL] & h) / h;
            h <<= 1;
        }
    }

    // output matrix
    for (int i = 0; i < EXTRA; ++i)
    {
        for (int j = 0; j < BASIC; ++j)
        {
            std::cout << baseMatrix[i][j] << " ";
        }
        std::cout << std::endl;
    }

    // 2 mistakes
    int count = 0;
    // 2 in basic packets
    for (int i = 0; i < BASIC - 1; ++i)
        for (int j = i + 1; j < BASIC; ++j)
        {
            unsigned int checkedVector[2];
            for (int e = 0; e < EXTRA; ++e)
                checkedVector[e] = baseMatrix[e][i] * 2 + baseMatrix[e][j];
            unsigned int xor = checkedVector[0] ^ checkedVector[1];
            if (xor)
                ++count;
        }
}

```

```

    }
    // 1 in basic and 1 in reserve packets
    for (int i = 0; i < EXTRA; ++i)
        for (int j = 0; j < BASIC; ++j)
            if (baseMatrix[i][j])
                ++count;
    // 2 in reserve packets, calculate as combination
    count += combination(ERROR, EXTRA);
    std::cout << "\ncount " << count;
    int general = combination(ERROR, BASIC + EXTRA);
    float probability = (float)count / general;
    std::cout << "\nprobability " << probability;
    getchar();
    getchar();
    return 0;
}
int combination(int upper, int lower)
{
    int copyUpper = upper;
    int copyLower = lower;
    int dividend = 1;
    int divisor = 1;
    while (copyUpper > 0)
    {
        dividend *= copyLower;
        divisor *= copyUpper;
        --copyLower;
        --copyUpper;
    }
    return dividend / divisor;
}

```

```

#include <iostream>
#include <cstdlib>
#include <ctime>
const unsigned int OPTIMAL = 7;
const unsigned int BASIC = 64;
const unsigned int EXTRA = 3;
const unsigned int ERROR = 3;
//const unsigned int base[] = { 3, 5, 6, 7, 4, 2, 1 };
//for random
unsigned int base[7];
unsigned int baseMatrix[EXTRA][BASIC];
int combination(int, int);
int main()
{
    // random generation
    srand(time(NULL));
    unsigned int random;
    for (int i = 0; i < 7; ++i)
    {
        base[i] = rand() % 7 + 1;
        std::cout << base[i] << " ";
    }
    std::cout << std::endl;
    //generate base matrix
    int h = 1;
    for (int j = 0; j < BASIC; ++j)
    {
        h = 1;
        for (int i = 0; i < EXTRA; ++i)
        {

```

```

        baseMatrix[i][j] = (base[j % OPTIMAL] & h) / h;
        h <<= 1;
    }
}
// output matrix
for (int i = 0; i < EXTRA; ++i)
{
    for (int j = 0; j < BASIC; ++j)
    {
        std::cout << baseMatrix[i][j] << " ";
    }
    std::cout << std::endl;
}
// 3 mistakes
int count = 0;
// 3 in basic packets
for (int i = 0; i < BASIC - 2; ++i)
    for (int j = i + 1; j < BASIC - 1; ++j)
        for (int k = j + 1; k < BASIC; ++k)
        {
            unsigned int checkedVector[3];
            for (int e = 0; e < EXTRA; ++e)
                checkedVector[e] = baseMatrix[e][i] * 4 + baseMatrix[e][j] * 2 + baseMatrix[e][k];
            unsigned int xor = checkedVector[0] ^ checkedVector[1] ^ checkedVector[2];
            if (xor)
                ++count;
        }
std::cout << "\ncount " << count;
// 2 in basic and 1 in reserve packets
unsigned int c;
for (int i = 0; i < EXTRA; ++i)
    for (int j = 0; j < BASIC - 1; ++j)
        for (int k = j + 1; k < BASIC; ++k)
        {
            unsigned int checkedVector[2];
            c = 0;
            for (int e = 0; e < EXTRA; ++e)
                if (i != e)
                    checkedVector[c++] = baseMatrix[e][j] * 2 + baseMatrix[e][k];
            unsigned int xor = checkedVector[0] ^ checkedVector[1];
            if (xor)
                ++count;
        }
std::cout << "\ncount " << count;
// 1 in basic and 2 in reserve packets
for (int i = 0; i < EXTRA; ++i)
    for (int j = 0; j < BASIC; ++j)
        if (baseMatrix[i][j])
            ++count;
std::cout << "\ncount " << count;
// 3 in reserve packets, calculate as combination
count += combination(ERROR, EXTRA);
std::cout << "\ncount " << count;
int general = combination(ERROR, BASIC + EXTRA);
float probability = (float)count / general;
std::cout << "\nprobability " << probability;
getchar();
getchar();
return 0;
}
int combination(int upper, int lower)
{

```

```

    int copyUpper = upper;
    int copyLower = lower;
    int dividend = 1;
    int divisor = 1;
    while (copyUpper > 0)
    {
        dividend *= copyLower;
        divisor *= copyUpper;
        --copyLower;
        --copyUpper;
    }
    return dividend / divisor;
}
#include <iostream>
#include <stdlib.h>
#include <time.h>
const unsigned int OPTIMAL = 15;
const unsigned int BASIC = 64;
const unsigned int EXTRA = 4;
const unsigned int ERROR = 3;
//const unsigned int base[] = { 0x3, 0x5, 0x9, 0x6, 0xA, 0xC, 0X7, 0xE, 0xD, 0xB, 0XF, 0x1, 0x2, 0x4, 0x8 };
// for random
unsigned int base[OPTIMAL];
unsigned int baseMatrix[EXTRA][BASIC];
int combination(int, int);
int main()
{
    // random generation
    srand(time(NULL));
    unsigned int random;
    for (int i = 0; i < OPTIMAL; ++i)
    {
        base[i] = rand() % OPTIMAL + 1;
        std::cout << base[i] << " ";
    }
    std::cout << std::endl;
    //generate base matrix
    int h = 1;
    for (int j = 0; j < BASIC; ++j)
    {
        h = 1;
        for (int i = 0; i < EXTRA; ++i)
        {
            baseMatrix[i][j] = (base[j] % OPTIMAL] & h) / h;
            h <<= 1;
        }
    }
    // output matrix
    for (int i = 0; i < EXTRA; ++i)
    {
        for (int j = 0; j < BASIC; ++j)
        {
            std::cout << baseMatrix[i][j] << " ";
        }
        std::cout << std::endl;
    }
    // 3 mistakes
    int count = 0;
    // 3 in basic packets
    for (int i = 0; i < BASIC - 2; ++i)
        for (int j = i + 1; j < BASIC - 1; ++j)
            for (int k = j + 1; k < BASIC; ++k)

```

```

        {
            unsigned int checkedVector[4];
            for (int e = 0; e < EXTRA; ++e)
                checkedVector[e] = baseMatrix[e][i] * 4 + baseMatrix[e][j] * 2 + baseMatrix[e][k];
            unsigned int xor = checkedVector[0] ^ checkedVector[1] ^ checkedVector[2] ^ checkedVector[3];
            if (xor)
                ++count;
        }
        std::cout << "\ncount " << count;
        // 2 in basic and 1 in reserve packets
        unsigned int c;
        for (int i = 0; i < EXTRA; ++i)
            for (int j = 0; j < BASIC - 1; ++j)
                for (int k = j + 1; k < BASIC; ++k)
                    {
                        unsigned int checkedVector[3];
                        c = 0;
                        for (int e = 0; e < EXTRA; ++e)
                            if (i != e)
                                checkedVector[c++] = baseMatrix[e][j] * 2 + baseMatrix[e][k];

                        unsigned int xor = checkedVector[0] ^ checkedVector[1] ^ checkedVector[2];
                        if (xor)
                            ++count;
                    }
        std::cout << "\ncount " << count;
        // 1 in basic and 2 in reserve packets
        for (int i = 0; i < EXTRA; ++i)
            for (int j = 0; j < BASIC; ++j)
                if (baseMatrix[i][j])
                    ++count;
        std::cout << "\ncount " << count;
        // 3 in reserve packets, calculate as combination
        count += combination(ERROR, EXTRA);
        std::cout << "\ncount " << count;
        int general = combination(ERROR, BASIC + EXTRA);
        float probability = (float)count / general;
        std::cout << "\nprobability " << probability;
        getchar();
        getchar();
        return 0;
    }
    int combination(int upper, int lower)
    {
        int copyUpper = upper;
        int copyLower = lower;
        int dividend = 1;
        int divisor = 1;
        while (copyUpper > 0)
        {
            dividend *= copyLower;
            divisor *= copyUpper;
            --copyLower;
            --copyUpper;
        }
        return dividend / divisor;
    }
}

#include <iostream>
#include <cstdlib>
#include <ctime>

```

```

const unsigned int OPTIMAL = 31;
const unsigned int BASIC = 64;
const unsigned int EXTRA = 5;
const unsigned int ERROR = 5;
//const unsigned int base[] = { 0x7, 0xB, 0x13, 0xD, 0x15, 0x19, 0xE, 0x16, 0x1A, 0x1C, 0xF, 0x17, 0x1B,
0x1D, 0x1E,
//0x3, 0x5, 0x9, 0x11, 0x6, 0xA, 0x12, 0xC, 0x14, 0x18, 0x1F, 0x1, 0x2, 0x4, 0x8, 0x10 };
// for random
unsigned int base[OPTIMAL];
unsigned int baseMatrix[EXTRA][BASIC];
int combination(int, int);
int main()
{
    // random generation
    srand(time(NULL));
    unsigned int random;
    for (int i = 0; i < OPTIMAL; ++i)
    {
        base[i] = rand() % OPTIMAL + 1;
        std::cout << base[i] << " ";
    }
    std::cout << std::endl;
    //generate base matrix
    int h = 1;
    for (int j = 0; j < BASIC; ++j)
    {
        h = 1;
        for (int i = 0; i < EXTRA; ++i)
        {
            baseMatrix[i][j] = (base[j] % OPTIMAL] & h) / h;
            h <<= 1;
        }
    }
    // output matrix
    for (int i = 0; i < EXTRA; ++i)
    {
        for (int j = 0; j < BASIC; ++j)
        {
            std::cout << baseMatrix[i][j] << " ";
        }
        std::cout << std::endl;
    }
    // 5 mistakes
    int count = 0;
    // 5 in basic packets
    for (int i = 0; i < BASIC - 4; ++i)
        for (int j = i + 1; j < BASIC - 3; ++j)
            for (int k = j + 1; k < BASIC - 2; ++k)
                for (int g = k + 1; g < BASIC - 1; ++g)
                    for (int h = g + 1; h < BASIC; ++h)
                        {
                            unsigned int checkedVector[5];
                            for (int e = 0; e < EXTRA; ++e)
                                checkedVector[e] = baseMatrix[e][i] * 8 + baseMatrix[e][j] * 4 + baseMatrix[e][k] * 2 + baseMatrix[e][g];

                            unsigned int xor = checkedVector[0] ^ checkedVector[1] ^
checkedVector[2] ^ checkedVector[3] ^ checkedVector[4];
                            if (xor)
                                ++count;
                        }
    std::cout << "\ncount " << count;
    // 4 in basic and 1 in reserve packets

```

```

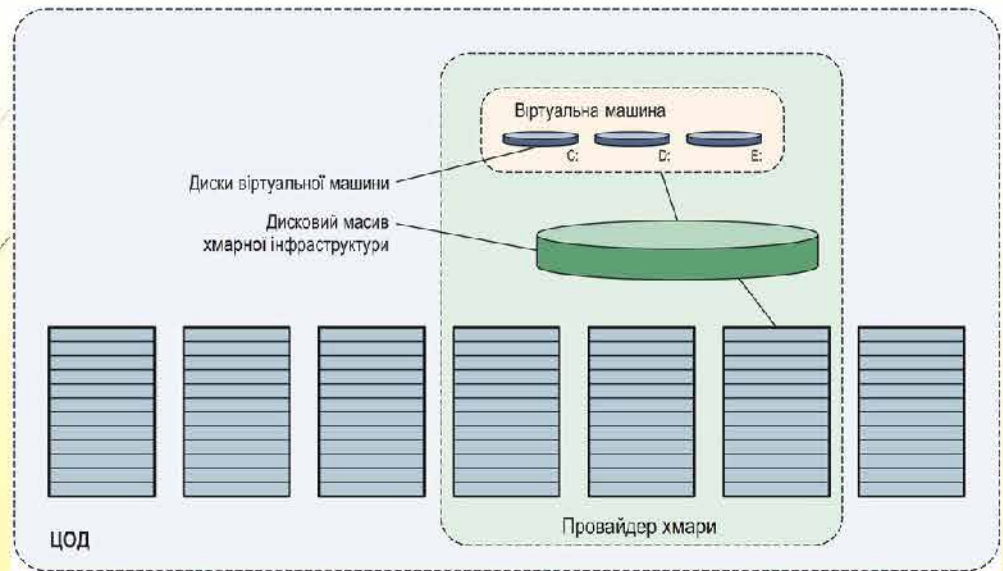
unsigned int c;
for (int i = 0; i < EXTRA; ++i)
    for (int j = 0; j < BASIC - 3; ++j)
        for (int k = j + 1; k < BASIC - 2; ++k)
            for (int g = k + 1; g < BASIC - 1; ++g)
                for (int h = g + 1; h < BASIC; ++h)
                    {
                        unsigned int checkedVector[4];
                        c = 0;
                        for (int e = 0; e < EXTRA; ++e)
                            if (i != e)
                                checkedVector[c++] = baseMatrix[e][j] * 8 + baseMatrix[e][k] * 4 + baseMatrix[e][g] * 2 + baseMatrix[e][h];
                        unsigned int xor = checkedVector[0] ^ checkedVector[1] ^ checkedVector[2] ^ checkedVector[3];
                        if (xor)
                            ++count;
                    }
std::cout << "\ncount " << count;
// 3 in basic and 2 in reserve packets
for (int i = 0; i < EXTRA - 1; ++i)
    for (int f = i + 1; f < EXTRA; ++f)
        for (int j = 0; j < BASIC - 2; ++j)
            for (int k = j + 1; k < BASIC - 1; ++k)
                for (int g = k + 1; g < BASIC; ++g)
                    {
                        unsigned int checkedVector[3];
                        c = 0;
                        for (int e = 0; e < EXTRA; ++e)
                            if (i != e && f != e)
                                checkedVector[c++] = baseMatrix[e][j] * 4 + baseMatrix[e][k] * 2 + baseMatrix[e][g];
                        unsigned int xor = checkedVector[0] ^ checkedVector[1] ^ checkedVector[2];
                        if (xor)
                            ++count;
                    }
std::cout << "\ncount " << count;
// 2 in basic and 3 in reserve packets
for (int i = 0; i < EXTRA - 2; ++i)
    for (int f = i + 1; f < EXTRA - 1; ++f)
        for (int j = f + 1; j < EXTRA; ++j)
            for (int k = j + 1; k < BASIC - 1; ++k)
                for (int g = k + 1; g < BASIC; ++g)
                    {
                        unsigned int checkedVector[2];
                        c = 0;
                        for (int e = 0; e < EXTRA; ++e)
                            if (i != e && f != e && j != e)
                                checkedVector[c++] = baseMatrix[e][k] * 2 + baseMatrix[e][g];
                        unsigned int xor = checkedVector[0] ^ checkedVector[1];
                        if (xor)
                            ++count;
                    }
std::cout << "\ncount " << count;
// 1 in basic and 4 in reserve packets
for (int i = 0; i < EXTRA; ++i)
    for (int j = 0; j < BASIC; ++j)
        if (baseMatrix[i][j])
            ++count;
std::cout << "\ncount " << count;
// 4 in reserve packets, calculate as combination
count += combination(ERROR, EXTRA);
std::cout << "\ncount " << count;
int general = combination(ERROR, BASIC + EXTRA);

```

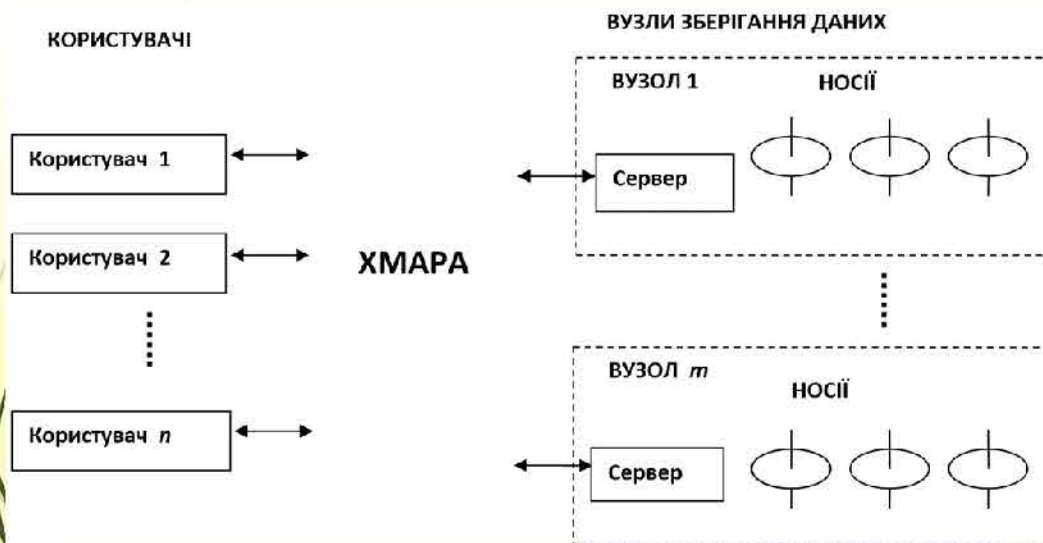
```
float probability = (float)count / general;
std::cout << "\nprobability " << probability;
getchar();
getchar();
return 0;
}
int combination(int upper, int lower)
{
    int copyUpper = upper;
    int copyLower = lower;
    int dividend = 1;
    int divisor = 1;
    while (copyUpper > 0)
    {
        dividend *= copyLower;
        divisor *= copyUpper;
        --copyLower;
        --copyUpper;
    }
    return dividend / divisor;
}
```

Слайди презентації випускної роботи

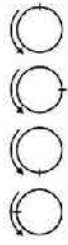
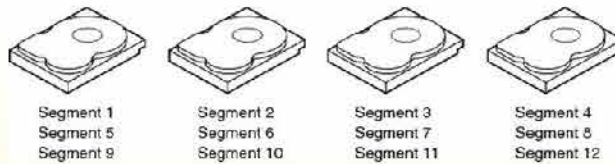
Організація центру обробки даних (ЦОД)



Організація розподіленого віддаленого зберігання інформації

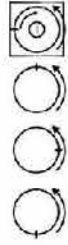


Запис даних на RAID-масив з паралельним доступом



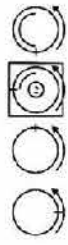
$t=0$
Дані записуються в буфер 1-го пристрою

буфер 1-го пристрою повний
буфер 2-го пристрою готовий
буфер 3-го пристрою готовий
буфер 4-го пристрою готовий



$t=1$
Дані записуються в буфер 2-го пристрою

буфер 1-го пристрою порожній
буфер 2-го пристрою повний
буфер 3-го пристрою готовий
буфер 4-го пристрою готовий



$t=2$
Дані записуються в буфер 3-го пристрою

буфер 1-го пристрою закінчений
буфер 2-го пристрою порожній
буфер 3-го пристрою повний
буфер 4-го пристрою готовий



$t=3$
Дані записуються в буфер 4-го пристрою

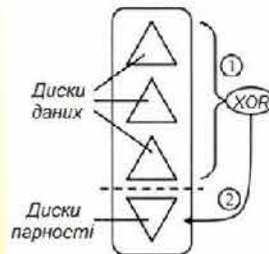
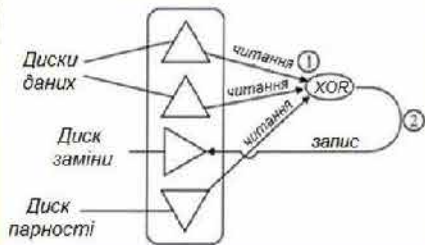
буфер 1-го пристрою готовий
буфер 2-го пристрою закінчений
буфер 3-го пристрою порожній
буфер 4-го пристрою повний



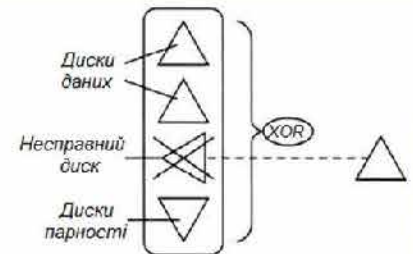
$t=4$
Дані записуються в буфер 1-го пристрою

буфер 1-го пристрою повний
буфер 2-го пристрою готовий
буфер 3-го пристрою закінчений
буфер 4-го пристрою порожній

Процес відтворення даних на новому диску з використанням функції XOR



Функція XOR створює дані перевірки на дисках парності



Функція XOR використовує дані перевірки для відновлення даних з несправних дисків

Створення перевірочних даних і відновлення за допомогою функції XOR

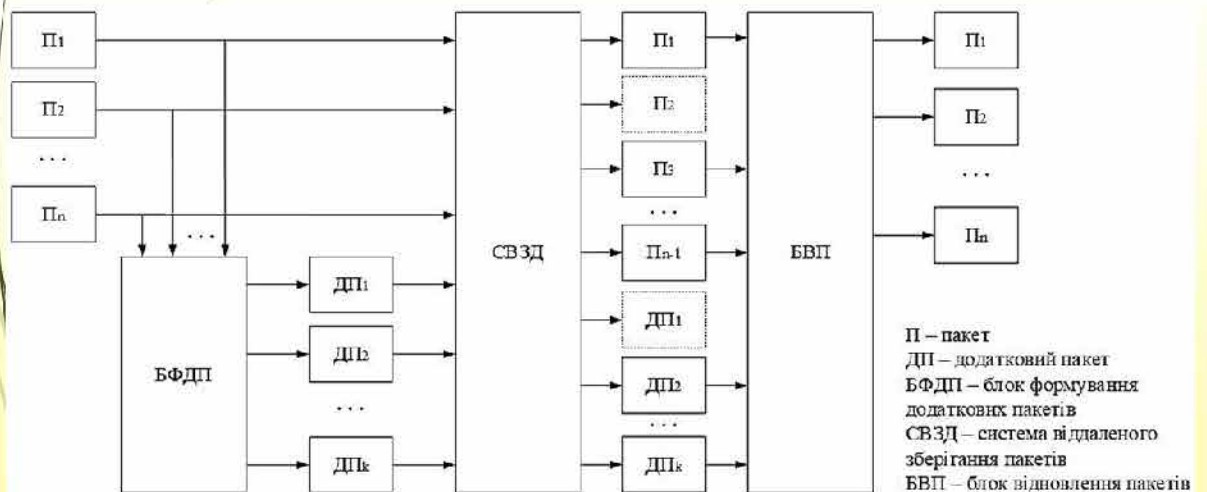
Математична модель методу відновлення даних

$$\forall j = 1, \dots, n: s_{1,j} = \bigoplus_{i=1}^n a_{ji}. \quad (1)$$

$$\begin{cases} a_{qj} + a_{rj} = z_{1j} \\ a_{qj} = z_{2j} \end{cases} \text{ або } \begin{cases} a_{qj} + a_{rj} = z_{1j} \\ a_{rj} = z_{3j} \end{cases}, \forall j = \{1, \dots, n\}. \quad (2)$$

$$\begin{cases} a_{1j} + a_{2j} + \dots + a_{m/2,j} = y_j \\ a_{1j} + a_{2j} + \dots + a_{m/4,j} + a_{m/2+1,j} + \dots + a_{3m/4,j} = y_2, \forall j \in \{1, \dots, n\} \\ \dots \\ a_{1j} + a_{3j} + a_{5j} + \dots + a_{m-1,j} = y_{\log_2 n, j} \end{cases} \quad (3)$$

Структурна схема системи відновлення даних



Формування резервних пакетів

n – число основних інформаційних пакетів;

m – число резервних пакетів;

u – число втрачених пакетів;

P_i – основний інформаційний пакет, де $i \in \{1, 2, \dots, n\}$;

D_j – резервний пакет, де $j \in \{1, 2, \dots, m\}$;

W_j – бінарний вектор вигляду $W_j = \{W_{j1}, W_{j2}, \dots, W_{jm}\}$, де $W_{ji} \in \{0,1\}$ і є способом формування D_j як суми по модулю 2 основних пакетів.

$$D_j = \bigoplus_{i=1}^m W_{ji} \cdot P_i \quad (4)$$

Відновлення втрачених пакетів

$$V_j = a_1 \cdot v_1 \oplus a_2 \cdot v_2 \oplus \dots \oplus a_{i-1} \cdot v_{i-1} \oplus a_i \cdot v_i \oplus a_{i+1} \cdot v_{i+1} \oplus \dots \oplus a_n \cdot v_n, \quad (5)$$

де $a_i \in \{0,1\}$.

$$P_j = a_1 \cdot P_1 \oplus a_2 \cdot P_2 \oplus \dots \oplus a_{i-1} \cdot P_{i-1} \oplus a_i \cdot P_i \oplus a_{i+1} \cdot P_{i+1} \oplus \dots \oplus a_n \cdot P_n \quad (6)$$

$$P_i = P_1 \oplus P_2 \oplus \dots \oplus P_{i-1} \oplus D \oplus P_{i+1} \oplus \dots \oplus P_n \quad (7)$$

$$\lambda_{xg} \oplus \lambda_{yg} = 1, \quad \lambda_{xq} \oplus \lambda_{yq} = 1 \quad (8)$$

$$P_n = \prod_{j=1}^{n-1} \left(1 - \frac{2^j - 1}{2^{n-1}}\right) \quad (9)$$

Формування ефективної системи резервних лінійних кодів

$$\lambda_{ij}' = \lambda_{ij} \oplus c_{j1} \cdot \varepsilon_1 \oplus c_{j2} \cdot \varepsilon_2 \oplus \dots \oplus c_{j,n-k} \cdot \varepsilon_{n-k} \quad (12)$$

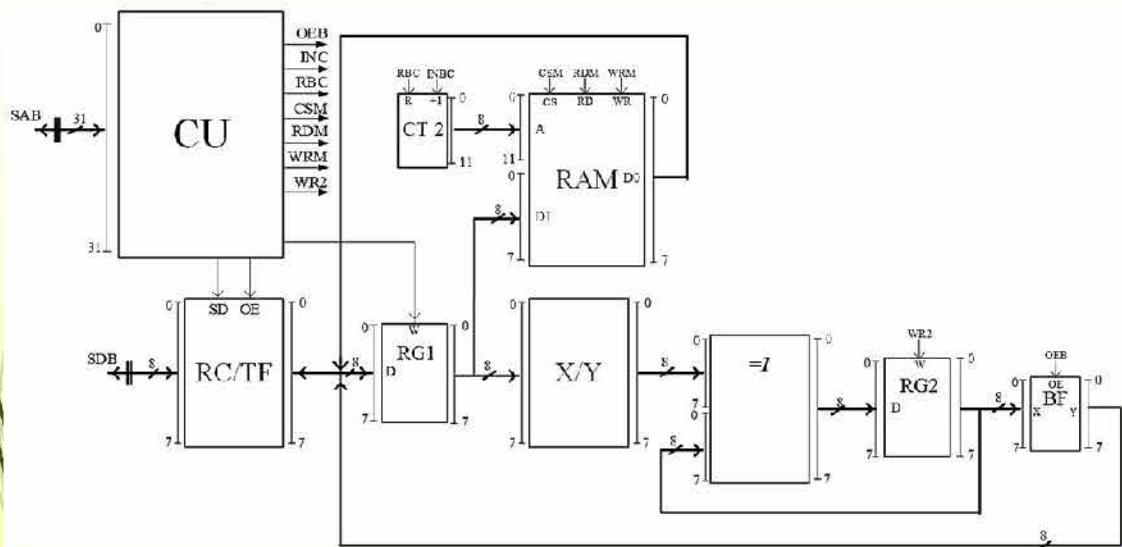
$$\begin{aligned} v_1 &= b_{11} \cdot \lambda_1 \oplus \dots \oplus b_{1k} \cdot \lambda_k = b_{11} \cdot \lambda_1 \oplus \dots \oplus b_{1k} \cdot \lambda_k \oplus d_{11} \cdot \varepsilon_1 \oplus d_{12} \cdot \varepsilon_2 \oplus \dots \oplus d_{1,n-k} \cdot \varepsilon_{n-k} \\ v_2 &= b_{21} \cdot \lambda_1 \oplus \dots \oplus b_{2k} \cdot \lambda_k = b_{21} \cdot \lambda_1 \oplus \dots \oplus b_{2k} \cdot \lambda_k \oplus d_{21} \cdot \varepsilon_1 \oplus d_{22} \cdot \varepsilon_2 \oplus \dots \oplus d_{2,n-k} \cdot \varepsilon_{n-k} \\ &\dots \\ v_k &= b_{k1} \cdot \lambda_1 \oplus \dots \oplus b_{kk} \cdot \lambda_k = b_{k1} \cdot \lambda_1 \oplus \dots \oplus b_{kk} \cdot \lambda_k \oplus d_{k1} \cdot \varepsilon_1 \oplus d_{k2} \cdot \varepsilon_2 \oplus \dots \oplus d_{k,n-k} \cdot \varepsilon_{n-k} \end{aligned} \quad (13)$$

де $\forall i \in \{1, 2, \dots, k\}, j \in \{1, 2, \dots, n-k\}$: $d_{ij} = \bigoplus_{l=1}^k b_{il} \cdot c_{lj}$.

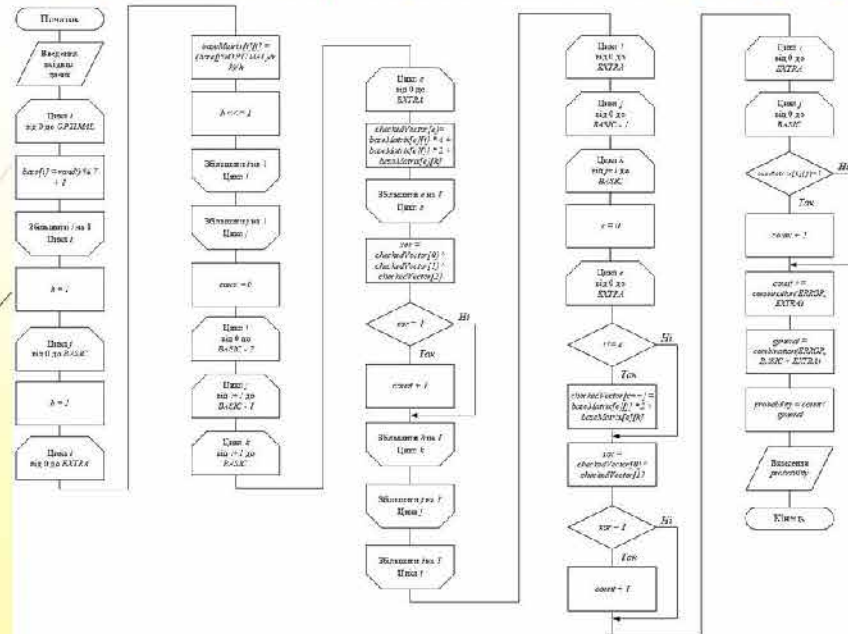
$$\lambda_0 = \bigoplus_{j=1}^n q_j \quad (14)$$

$$\begin{aligned} P_4 &= D_1 \oplus P_2 \oplus P_3 \oplus P_6 = 0x134A \\ P_1 &= D_1 \oplus D_2 \oplus P_3 \oplus P_5 = 0x4FCD \end{aligned} \quad (20)$$

Функціональна схема системи контролю помилок передачі даних



БСА контролю та корекції помилок передачі даних



Інтерфейс додатку для дослідження корегувальної здатності лінійних відновлюючих кодів

```

const int DIGIT = 3;
const int EXTRA = 3;
const int LIMIT = 8x8;
const int FirstExtra = DIGIT;
const int SecondExtra = DIGIT + 1;
const int ThirdExtra = DIGIT + 2;
const int REPEAT = 10000;
  
```

```

base = generateBaseVectors(DIGIT);
base.push_back(0x7);
base.push_back(0x6);
base.push_back(0x5);
  
```

```

C:\Project\code\Program_Imitate_SomeProgram_Imit\SomeProgram_Imit.exe
Choose one variant: 1 - random generation 2 - manual generation
1
Average P = 0.3273
Max P = 1
  
```

```

C:\Project\code\Program_Imitate_SomeProgram_Imit\SomeProgram_Imit.exe
Choose one variant: 1 - random generation 2 - manual generation
2
P = 1
Correct 1 General 1
  
```

```

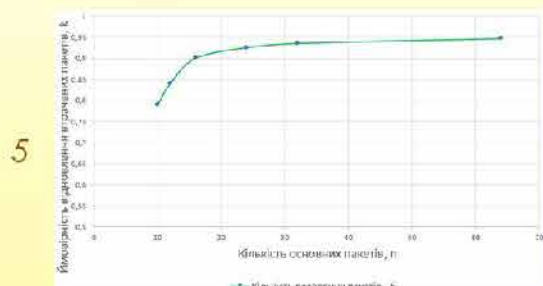
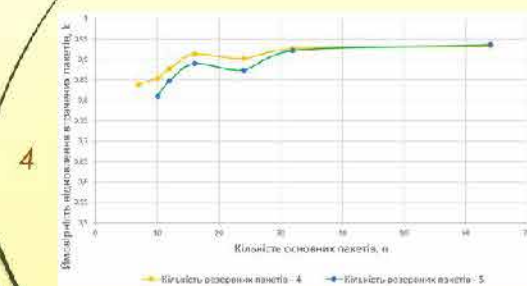
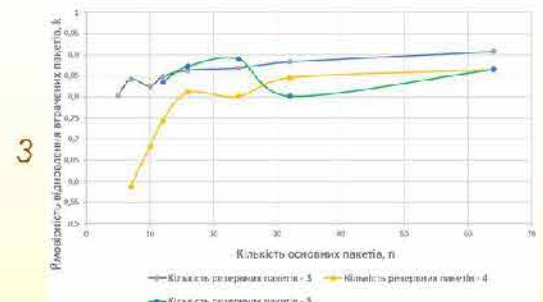
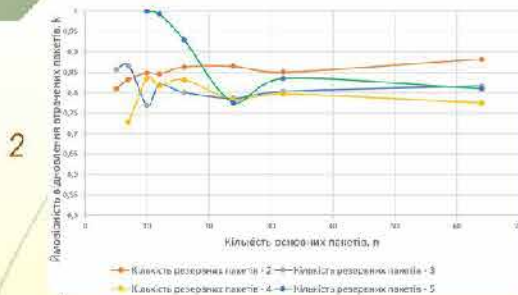
C:\Project\code\Program_Imitate_SomeProgram_Imit\SomeProgram_Imit.exe
Choose one variant: 1 - random generation 2 - manual generation
5
Invalid input. Please try again...
Choose one variant: 1 - random generation 2 - manual generation
  
```

Кількість основних пакетів n	Кількість резервних пакетів k	Кількість втрачених пакетів			
		2	3	4	5
5	2	0,809524	-	-	-
	3	0,857143	0,803571	-	-
7	2	0,833333	-	-	-
	3	0,866667	0,841667	-	-
	4	0,727273	0,587879	0,839394	-
10	2	0,848485	-	-	-
	3	0,769231	0,825175	-	-
	4	0,835165	0,681319	0,853147	-
	5	1,0	0,835165	0,809524	0,788545
12	2	0,846154	-	-	-
	3	0,819048	0,848352	-	-
	4	0,816667	0,742857	0,876923	-
16	5	0,992647	0,872059	0,846639	0,83969
	2	0,862745	-	-	-
	3	0,80117	0,862745	-	-
	4	0,831579	0,810526	0,9129	-
24	5	0,928571	0,889474	0,88939	0,900487
	2	0,864615	-	-	-
	3	0,786325	0,858376	-	-
32	4	0,783069	0,800672	0,902125	-
	5	0,775862	0,80104	0,873605	0,925334
	2	0,85057	-	-	-
	3	0,803361	0,882811	-	-
64	4	0,796825	0,845248	0,92673	-
	5	0,834835	0,864221	0,921781	0,935354
	2	0,882051	-	-	-
64	3	0,816825	0,907358	-	-
	4	0,775241	0,861342	0,932657	-
	5	0,810315	0,876379	0,936113	0,946726

Ймовірність відновлення втрачених пакетів

$$P_0 = \sum_{h=\lfloor \log_2 k \rfloor}^{k-2} \frac{\beta!}{(\beta-h)! \cdot h!} \cdot \prod_{i=0}^{h-1} \frac{(2^h - i)}{(2^n - 1)} \quad (21)$$

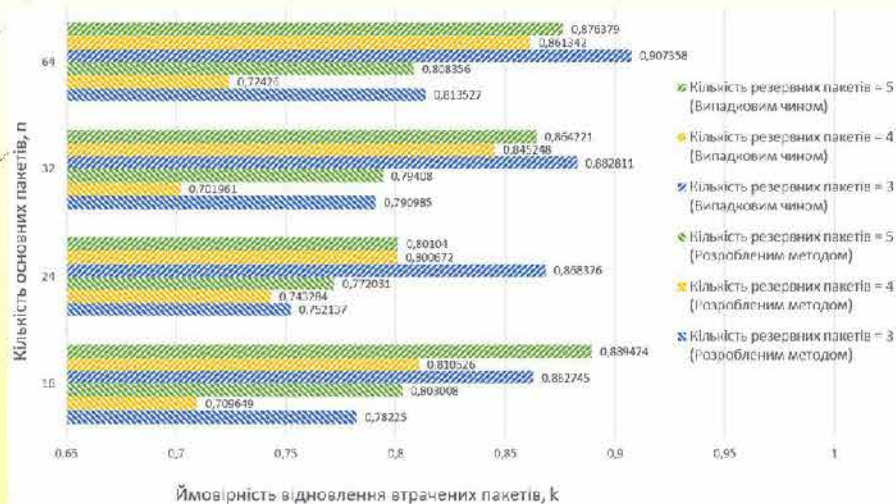
Ймовірність відновлення 2,3,4,5 втрачених пакетів



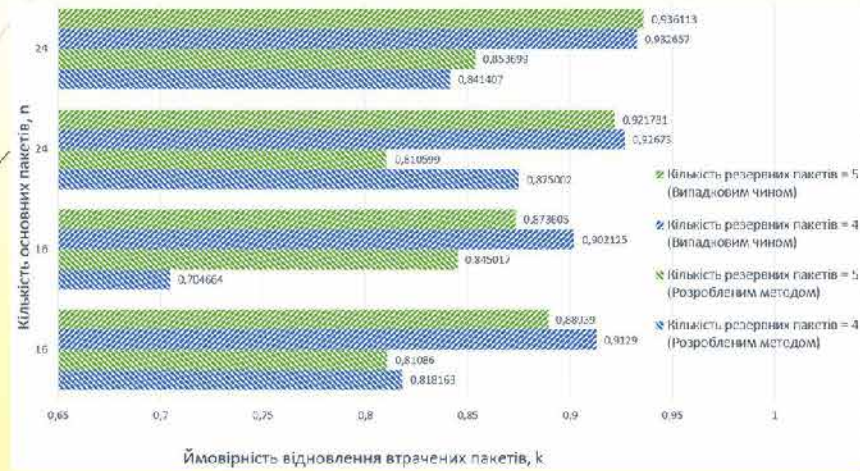
Кількість основних пакетів n	Кількість резервних пакетів k	Кількість втрачених пакетів		
		3	4	5
16	2	-	-	-
	3	0,78225 (-0,080495)	-	-
	4	0,709649 (-0,100877)	0,818163 (-0,094737)	-
	5	0,803008 (-0,086466)	0,81086 (-0,07853)	0,854538 (-0,045949)
24	2	-	-	-
	3	0,752137 (-0,116239)	-	-
	4	0,745284 (-0,057388)	0,704664 (-0,197461)	-
	5	0,772031 (-0,029009)	0,845017 (-0,028588)	0,879896 (-0,045438)
32	2	-	-	-
	3	0,790985 (-0,091826)	-	-
	4	0,701961 (-0,143287)	0,875002 (-0,051728)	-
	5	0,79408 (-0,070141)	0,810599 (-0,11182)	0,876058 (-0,059296)
64	2	-	-	-
	3	0,813527 (-0,093831)	-	-
	4	0,72426 (-0,137082)	0,841407 (-0,09125)	-
	5	0,808356 (-0,068023)	0,853699 (-0,082414)	0,891035 (-0,055691)

Порівняння ймовірностей відновлення втрачених пакетів за допомогою додаткових пакетів

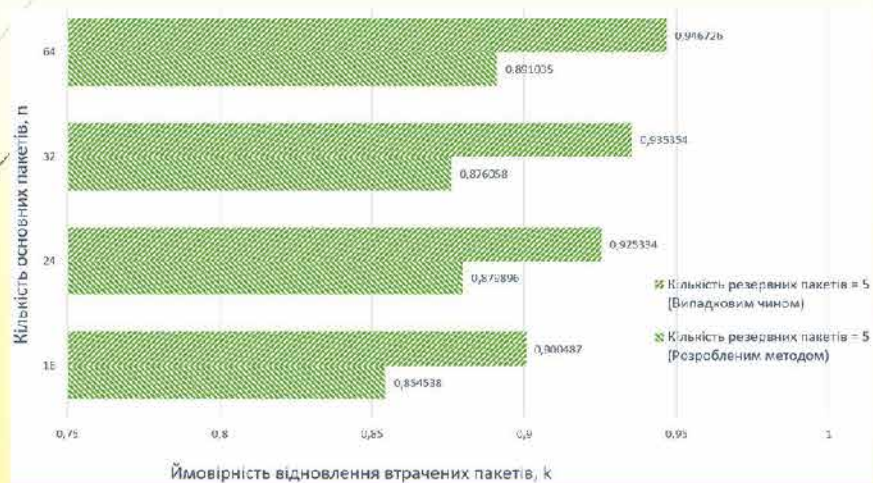
Порівняння ймовірностей відновлення 3 втрачених пакетів за допомогою додаткових пакетів



Порівняння ймовірностей відновлення 4 втрачених пакетів за допомогою додаткових пакетів



Порівняння ймовірностей відновлення 5 втрачених пакетів за допомогою додаткових пакетів



ВІДГУК

керівника про кваліфікаційну роботу бакалавра

Гончаренка Олександра Володимировича

(прізвище, ім'я та по батькові)

Спеціальність 123 "Комп'ютерна інженерія"

Тема кваліфікаційної роботи Аналіз ефективності відновлення даних при збереженні на віддалених носіях

ХАРАКТЕРИСТИКА КВАЛІФІКАЦІЙНОЇ РОБОТИ

а) Обсяг і якість виконання роботи (графічного матеріалу і розрахунково-пояснювальної записки) Випускна робота виконана відповідно технічному завданню. Пояснювальна записка до випускної роботи містить 84 сторінок. У пояснювальній записці розглянуто проблему виникнення помилок та втрати інформації при її передачі через глобальну мережу та при збереженні на віддалених носіях за використання хмарних технологій. Графічна частина складається з 19 слайдів, оформлених у вигляді презентації, передбачених технічним завданням. Якість виконання пояснювальної записки та слайдів добра, розробку виконано у повному обсязі.

б) Самостійність роботи

Протягом виконання випускної бакалаврської роботи Гончаренко Олександр поступово та послідовно виконував всі етапи, проявив ініціативу у створенні загальної концепції та реалізації випускної роботи. Всі роботи він виконував самостійно, з оглядом на рекомендації керівника.

в) Теоретична підготовка здобувача освіти _____

Гончаренко Олександр під час роботи над випускною бакалаврською роботою вивчив і опрацював достатню кількість літературних джерел за даною тематикою.

Вважаю, що теоретична підготовка здобувача освіти достатня і він готовий до захисту роботи.

г) Вміння розв'язувати виробничі і конструкторські питання на базі останніх досліджень науки і техніки, передових методів виробництва _____

Під час виконання роботи Гончаренко Олександр мав змогу самостійно приймати окремі рішення з виконання програмної частини роботи та показав вміння організовано працювати над поставленою задачею, складати та оформлювати презентацію проекту, користуючись сучасними комп'ютерними програмними засобами, такими як Microsoft Visual Studio, Microsoft PowerPoint, Microsoft Visio.

Оцінка розрахункової частини Добре

Оцінка графічної частини Добре

Загальна оцінка Добре

Прізвище, ім'я, по батькові Кривченко Юрій Вікторович

Місце роботи і посада керівника роботи ВСП "Одеський технічний фаховий коледж ОНТУ", викладач кафедри комп'ютерної інженерії, голова циклової комісії комп'ютерних технологій та програмної інженерії

Підпис _____

«13» Червень 2024р.

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра здобувача освіти
відділення комп'ютерних систем

Гончаренка Олександра Володимировича

(прізвище, ім'я та по батькові)

Спеціальність 123 "Комп'ютерна інженерія"

Освітня програма «Комп'ютерна інженерія»

Керівник дипломного проекту (роботи) Кривченко Юрій Вікторович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Аналіз ефективності відновлення даних при збереженні на віддалених носіях

Обсяг розрахунково-пояснювальної записки 84 сторінок

Обсяг графічної (презентаційної) частини 19 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту (роботи) завданню Представлена на рецензію кваліфікаційна робота бакалавра повністю відповідає меті проектування та технічному завданню. Тематика кваліфікаційної роботи є актуальною для своєї галузі та присвячена аналізу ефективності відновлення даних при збереженні на віддалених носіях.

б) характеристика виконання кожного розділу дипломного проекту (роботи) Кваліфікаційна робота складається зі вступу, двох розділів, висновків, переліку використаних джерел. У основному розділі розглянуті проблеми забезпечення надійності передачі даних у глобальній мережі, виконано огляд існуючих методів та засобів відновлення даних, технологій відновлення даних при їх віддаленому зберіганні, виконано програмну реалізацію методу формування надлишкових лінійних кодів, реалізовано функціональну схему та БСА контролю помилок передачі даних.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту (роботи) Графічна частина виконана на достатньо високому рівні у вигляді презентації із використанням офісного пакету Microsoft PowerPoint та Visio. Пояснювальна записка виконана охайно та у відповідності до норм оформлення документів із використанням офісного пакету Microsoft Word. Загальна якість виконання документації – добра, академічного плагіату у роботі не виявлено

г) перелік позитивних якостей дипломного проекту (роботи) _____

1. Детально описано мету та цілі аналізу;
2. Проведено серію експериментів для порівняння ймовірностей відновлення втрачених пакетів за допомогою додаткових пакетів;
3. Виконано побудову графіків, що візуально відображують результати роботи

д) основні недоліки дипломного проекту (роботи) _____

1. Для зручності зняття значень корегувальної здатності лінійних відновлюючих кодів у застосунку варто було передбачити візуальний інтерфейс;
2. Варто було описати результати експериментів більш докладно.

Оцінка розрахункової частини _____ Відмінно

Оцінка графічної частини _____ Добре

Загальна оцінка _____ Відмінно

Прізвище, ім'я, по батькові рецензента _____ Стайкуца Сергій Володимирович

Місце роботи і посада рецензента _____ Державний університет інтелектуальних технологій і зв'язку, к.ф.н., доцент кафедри КБ та ТЗІ



Підпис: _____

18 червня 2024 р.

Ім'я користувача:
Катерина Григоріївна Краснокутська

ID перевірки:
1016214028

Дата перевірки:
27.04.2024 16:41:21 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
27.04.2024 16:47:56 EEST

ID користувача:
100011688

Назва документа: **ЗБКС-28_Олександр_Гончаренко**

Кількість сторінок: **58** Кількість слів: **11736** Кількість символів: **84966** Розмір файлу: **1.63 MB** ID файлу: **1015987289**

4.63% Схожість

Найбільша схожість: **0.99%** з Інтернет-джерелом (<https://card-file.ontu.edu.ua/server/api/core/bitstreams/2bedf2c4-39b..>)

4.63% Джерела з Інтернету 520

Сторінка 60

Не знайдено джерел з Бібліотеки

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 52

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Гончаренко Олександр Володимирович,
здобувач освіти гр. 2БКС-28, та

Кривченко Юрій Вікторович,
керівник випускної кваліфікаційної роботи,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до випускної кваліфікаційної роботи бакалавра на тему:

«Аналіз ефективності відновлення даних при збереженні на віддалених носіях» (автор роботи – Гончаренко О.В., керівник роботи – Кривченко Ю.В.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2024 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець



/ Гончаренко О.В. /

Керівник



/ Кривченко Ю.В. /

«10» червня 2024 р.