МІЖНАРОДНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, програмної інженерії та комп'ютерних наук
Кафедра комп'ютерних наук

Пояснювальна записка

до кваліфікаційної роботи
першого (бакалаврського) рівня

на тему **Оцінка ефективності роботи модемів у комп'ютерних мережах**

Виконав: студент 4 курсу, групи КІ-4
спеціальності
123 Комп'ютерна інженерія

Топчанюк Б. М.

Керівник Розенвассер Д.М.

Рецензент Григор'єва П. І.

Одеса – 2023 р

ДОВІДКА

кафедри КН про виконану бакалаврську роботу

студента 4 курсу ФКПІтаКН групи КІ-4

Топчанюк Б. М.

на Оцінка ефективності роботи модемів у комп'ютерних мережах

Висновок нормоконтролера пояснювальна записка до кваліфікаційної роботи виконано з незначними порушеннями ДСТУ та вимог внутрішніх нормативних документів КПІ

Нормоконтролер викл. каф ІТП (науковий ступінь, вчене звання, посада) 29.06.23 (підпис, дата) І.В. Клімішина (і. б. прізвище)

Висновок відповідального за наявність плагіату згідно з сертифікатом ID 1015327938 унікальність підтверджено

Відповідальна особа викл. каф ІТ (науковий ступінь, вчене звання, посада) 29.06.23 (підпис, дата) І.В. Клімішина (і. б. прізвище)

Попередня експертиза (захист) бакалаврської роботи (бакалаврської роботи чи магістерської роботи)

студ. Топчанюк Б. М. (прізвище і.б.) проведена "____" _____ 20__ р.

Висновки студент Топчанюк Б.М. добре порівався з усіма поставленими використання завадостійкого кодування у комп'ютерних мережах і основи аналізу ефективності обраного кодування завдання на ВКР виконано. Необхідні для цього розрахунки проведено

Виконана робота бакалавра відповідає вимогам стандарту та рекомендується до захисту в ЕК

Члени комісії (підпис) к.т.н., доц. Соловська І.М. (науковий ступінь, вчене звання, посада, прізвище і.б.)

(підпис) к.т.н. доц. Григор'єва Т.І. (науковий ступінь, вчене звання, посада, прізвище і.б.)

(підпис) к.т.н., доц. Чорна Л.Г. (науковий ступінь, вчене звання, посада, прізвище і.б.)

МІЖНАРОДНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, програмної інженерії та комп'ютерних наук

Кафедра комп'ютерних наук

Освітній ступінь бакалавр

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри КН
к.т.н., доц.
_____ І.М.Соловська
" 7 " 04 2023 року

З А В Д А Н Н Я

НА БАКАЛАВРСЬКУ РОБОТУ

Топчанюку Б. М.

1. Тема роботи: Оцінка ефективності роботи модемів у комп'ютерних мережах

керівник роботи Розенвассер Денис Михайлович, доцент каф. КН

затверджені наказом закладу вищої освіти від від 7 квітня 2023 р. № 587

2. Строк подання студентом роботи 10.06.2023 р.

3. Вихідні дані до роботи: Комп'ютерна мережа з такими параметрами: коефіцієнт бітових помилок 10^{-6}; ймовірність стирання пакетів (0,3,5,8,15)%. Необхідно проаналізувати можливість та ефективність використання модемів з різними типами модуляції та завадостійкого кодування у комп'ютерних мережах.

4. Зміст розрахунково-пояснювальної записки

Розділ 1: Комп'ютерні мережі.

Розділ 2: Пакетна передача даних.

Розділ 3: Дослідження ефективності модемів.

5. Перелік графічного матеріалу (з зазначенням обов'язкових креслень)

Слайд 1 – Методи побудови інформаційних каналів

Слайд 2 – Структурна схема сучасного модему

Слайд 3 – Завадостійкість розглянутих кодових конструкцій

Слайд 4 – Відсоток відновлення пакетів для LT

Слайд 5 – Висновки та рекомендації

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	завдання видав	завдання прийняв

7. Дата видачі завдання 25.11.2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Вступ	18.01.2023 – 11.02.2023	вик
2	Вивчення та огляд класичних методів завадостійкого кодування у пакетних мережах (код Ріда-Соломона, каскадний код, турбо-код)	12.02.2023 – 28.02.2023	вик
3	Дослідження технології фонтанного кодування та їх характеристик	01.03.2023 – 19.03.2023	вик
4	Дослідження ефективності модемів	20.03.2023 – 14.04.2023	вик
5	Висновки та рекомендації	15.04.2023 – 09.05.2023	вик
6	Перелік посилань	10.05.2023 – 31.05.2023	вик
7	Оформлення презентації	01.06.2023 – 10.06.2023	вик

Студент _____ Топчанюк Б. М.
(підпис)

Керівник роботи _____ Д.М. Розенвассер
(підпис)

РЕЦЕНЗІЯ

на бакараврську роботу студента Топчанюка Б. М.

на тему: «Оцінка ефективності роботи модемів у комп'ютерних мережах»

Бакалаврська робота містить 3 розділи текстової частини, демонстраційні слайди та виконана згідно з завданням на бакалаврську роботу.

У роботі розглядаються класичні та новітні методи застосування завадостійкого кодування у модемах комп'ютерних мереж та проведено їх порівняння.

Актуальність теми полягає в тому, що у сучасному світі досі стоїть питання про покращення передавання даних через пакетні мережі. Бакалаврська робота виконана відповідно до завдання. Демонстраційні матеріали й пояснювальна записка виконані охайно й відповідно до вимог ЄСКД.

Прийняті рішення не завжди обґрунтовано, для деяких розрахунків приведено тільки результати.

Автором показана достатня теоретична підготовка. Робота виконана грамотно, текст її послідовний та зрозумілий, оформлення роботи та демонстраційних аркушів якісне.

До недоліків роботи варто віднести:

- не проведено реальний експеримент, який би досконало показав переваги використання кода типу «Raptor» в комп'ютерних мережах;
- відсутнє економічне обґрунтування переваги вибору фонтанного коду в комп'ютерних мережах.

Зазначені недоліки суттєво не знижують якості виконаної роботи. Бакалаврська робота відповідає вимогам до кваліфікаційних робіт бакалаврів.

Робота студента Топчанюка Б. М. заслуговує оцінки «задовільно». Студент Топчанюк Б. М. заслуговує присвоєння за заявленою спеціальністю 123 «Комп'ютерна інженерія» кваліфікації бакалавр з комп'ютерної інженерії.

Рецензент
Зав. кафедри ІТ,
к.т.н., доцент

Т.І.Григор'єва

ВІДГУК КЕРІВНИКА

бакалаврської роботи студента Топчанюка Б. М.
на тему: «Оцінка ефективності роботи модемів у комп'ютерних мережах»

Завдання вивчення класичних та новітніх методів завадостійкого кодування для використання у мереж з пакетною передачею даних є актуальною темою сьогодення, оскільки на даний момент такі мережі все частіше застосовуються у сучасному світі.

У роботі розглядаються методи, технологі та стандарти, що передбачають використання завадостійкого кодування. Наведено приклади використання класичних та новітніх кодів у комп'ютерних системах та виконано порівняння їх ефективності.

Студент Топчанюк Б. М. добре розібрався з усіма проблемами використання завадостійкого кодування у пакетних мережах і основну увагу приділив докладному аналізу ефективності фонтанного кодування. Робота проводилася значною мірою самостійно. Графік консультацій не порушувався.

Завдання на ВКР виконано. Необхідні для цього розрахунки проведені.

При оформленні пояснювальної записки та демонстраційних слайдів використовувались комп'ютерні технології.

Під час виконання бакалаврської роботи студент Топчанюк Б. М. вивчив питання щодо використання фонтанного кодування, показав уміння користуватись навчальною та технічною літературою, розв'язувати інженерні задачі.

Бакалаврська робота відповідає вимогам до кваліфікаційних робіт бакалаврів.

Робота студента Топчанюка Б. М. заслуговує оцінки «задовільно». Студент Топчанюк Б. М. заслуговує присвоєння за заявленою спеціальністю 123 «Комп'ютерна інженерія» кваліфікації бакалавр з комп'ютерної інженерії.

Керівник
доцент кафедри КН

Д.М.Розенвассер

UNICHECK
by Turnitin

Ім'я користувача:
Анна Серединко

Дата перевірки:
23.06.2023 14:25:34 EEST

Дата звіту:
23.06.2023 14:26:30 EEST

ID перевірки:
1015683844

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100001433

Назва документа: Топчанюк

Кількість сторінок: 42 Кількість слів: 7949 Кількість символів: 61040 Розмір файлу: 517.27 KB ID файлу: 1015327938

16.1%
Схожість

Найбільша схожість: 4.01% з Інтернет-джерелом (https://journal.tusur.ru/storage/46284/185-192.pdf?1467099979=)

14.8% Джерела з Інтернету	871	Сторінка 44
3.67% Джерела з Бібліотеки	51	Сторінка 47

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи	3

РЕФЕРАТ

Текстова частина дипломної роботи: 44 с., 13 рис., 4 табл., 22 джерела.

МОДЕМИ, ЗАВАДОСТІЙКЕ КОДУВАННЯ, ПАКЕТНІ МЕРЕЖІ, КАНАЛИ ЗІ СТИРАННЯМИ, КОД РІДА-СОЛОМОНА, КАСКАДНИЙ КОД, ТУРБО-КОД, LT КОД, РАПТОР КОД, КАМ-16

Об'єкт дослідження – використання модемів та завадостійкого кодування в комп'ютерних мережах.

Мета роботи – дослідження завадостійких кодів, які використовуються у сучасних комп'ютерних мережах.

Метод дослідження – аналітичний з використанням комп'ютерних технологій.

У дипломній роботі проведено дослідження застосування завадостійкого кодування у пакетних мережах за критеріями завадостійкості та ефективності доставки пакетів до приймача. Проаналізовані види завадостійкого кодування, які допомагають досягти бажаного результату з найкращими показниками.

Зроблено порівняння різних видів кодування та модуляції. Дослідження ефективності кодування виконано з використанням комп'ютерного моделювання.

ABSTRACT

The text part of the thesis: 44 pp., 13 figures, 4 tables, 22 sources.

MODEMS, ERROR-CONTROL CODING, PACKET NETWORKS, CHANNELS WITH ERASURES, REED-SOLOMON CODE, CASCADE CODE, TURBO-CODE, LT CODE, RAPTOR CODE, 16QAM

The object of research is the use of modems and interference-resistant coding in computer networks.

The purpose of the work is to study interference-resistant codes used in modern computer networks.

The research method is analytical with the use of computer technologies.

An application study was carried out in the thesis interference-resistant coding in packet networks according to the criteria of immunity and efficiency of packet delivery to the receiver. The types of interference-resistant coding that help to achieve the desired result with the best indicators are analyzed.

A comparison of different types of coding and modulations is made. The study of coding efficiency was carried out using computer simulation.

ЗМІСТ

ВСТУП

Через збільшення обсягу переданих даних поступово виникли проблеми під час їх передачі. Дані зазвичай містяться у великих файлах. Однак мережі не працюватимуть належним чином, якщо надсилати всі блоки даних на адресу одночасно. Є дві причини, які сповільнюють мережу, коли ви передаєте великі блоки даних по кабелю. По-перше, такий блок заповнює кабель і гальмує всю мережу, тобто заважає взаємодії інших компонентів мережі. По-друге, виникнення помилок при передачі великих блоків призведе до повторної передачі всього блоку. І якщо невеликий фрагмент даних пошкоджений, то необхідно повторно передати цей невеликий блок, що значно економить час.

Теорія завадостійкого кодування базується на результатах досліджень, проведених Клодом Шенноном. Він сформулював теорему для дискретного каналу з шумом: при будь-якій швидкості передачі двійкового символу, меншій пропускної здатності каналу, існує такий код, що ймовірність помилкового декодування буде як завгодно мала.

Побудова такого коду досягається ціною введення надмірності. Тобто, застосовуючи для передачі інформації код, що використовує не всі можливі комбінації, а лише деякі з них, можна підвищити завадостійкість прийому. Такі коди називають надлишковими або коригуючими. Коригуючі властивості надлишкових кодів залежать від правил побудови цих кодів і параметрів коду (довжини символів, кількості розрядів, надлишковості тощо).

У цій роботі будемо досліджувати різні коди контролю помилок. Порівняємо їх один з одним та дамо рекомендації, які з них краще підходять для комп’ютерних мереж.

1 КОМП'ЮТЕРНІ МЕРЕЖІ

1.1 Комутація пакетів

За останнє десятиліття передача даних зазнала революції завдяки радикально новій технології під назвою комутація пакетів. ***Комутація пакетів*** - це метод групування даних, які передаються через цифрову мережу, у пакети, які складаються із заголовка та корисного навантаження. Дані в заголовку використовуються мережевим обладнанням для спрямування пакета до місця призначення, де корисне навантаження витягується та використовується прикладним програмним забезпеченням . Комутація пакетів є основною основою для передачі даних у комп’ютерних мережах у всьому світі.

Просте визначення комутації пакетів:

«Маршрутизація та передача даних за допомогою адресованих пакетів таким чином, що канал зайнятий лише під час передачі пакету, а після завершення передачі канал стає доступним для передачі іншого трафіку» [1].

Комутація пакетів — це метод групування даних, які передаються через цифрову мережу, у пакети , які складаються із заголовка та корисного навантаження . Дані в заголовку використовуються мережевим обладнанням для спрямування пакета до місця призначення, де корисне навантаження витягується та використовується прикладним програмним забезпеченням . Комутація пакетів є основною основою для передачі даних у комп’ютерних мережах у всьому світі [2].

Переваги комутації пакетів:

- Ефективність використання пропускної здатності
- При перевантаженні мережі ніхто не "викидає" з повідомленням "мережа зайнята", мережа просто знижує всім або кільком абонентам швидкість передачі

- Канал не повністю зайнятий, фактично дає пропускну здатність мережі решті
- Менші витрати.

Недоліки комутації пакетів:

- Складна структура; без мікропроцесорної техніки практично неможливо створити пакетну мережу.
- Пропускна здатність використовується для передачі технічних даних (сервісної інформації).
- Пакети можуть бути втрачені

У комп'ютерних мережах виникають шуми, що призводять до втрати і спотворення даних. Однією з актуальних проблем є проблема втрати та затримки пакетів, яка виникає через перевантаження вузлів (комутаторів, маршрутизаторів тощо), колізії (перекриття пакетів від різних абонентів), «просідання» пропускної здатності каналів передачі даних. через надто багато одночасно підключених користувачів, спотворення в пакеті, переповнення вхідних буферів мережевих пристроїв тощо. Традиційні програми комп’ютерного зв’язку, такі як передача файлів або електронна пошта, зазвичай вимагають надійної передачі даних. У мережах з комутацією пакетів це досягається за допомогою транспортних протоколів, які реалізують певну техніку автоматичного повторного запиту (ARQ). В Інтернеті протокол керування передачею (TCP) надає ці функції прозоро для програм. Однак для чутливих до затримки додатків, таких як аудіо- та відеоінструменти в реальному часі, схеми ARQ не підходять, оскільки затримку, викликану повторними передачами, неможливо допустити. Більше того, обмежена кількість втрат пакетів, як правило, не є катастрофою для таких типів додатків, а скоріше призводить до плавного зниження продуктивності. Однак у разі постійної втрати якість звуку та відео буде постійно низькою . У цій роботі ми дослідимо, які коди ми можемо використовувати для передачі

інформації з використанням * *зворотного каналу* і без нього (Erasure channels). (Рис.1.1)

* Зворотний канал зазвичай використовується для передачі сигналів запиту, контролю, підтвердження або контролю помилок .

Рисунок 1.1- Методи побудови інформаційних каналів, стійких до завад

1.2 Канали зі стираннями

Велике значення мають канали зі стираннями. Наприклад, файли, надіслані через Інтернет, поділяються на пакети (які мають кінцевий розмір і представлені як файли даних) і надсилаються відправником у вигляді послідовності пакетів. Розмір пакетів обмежений стандартами або існуючою

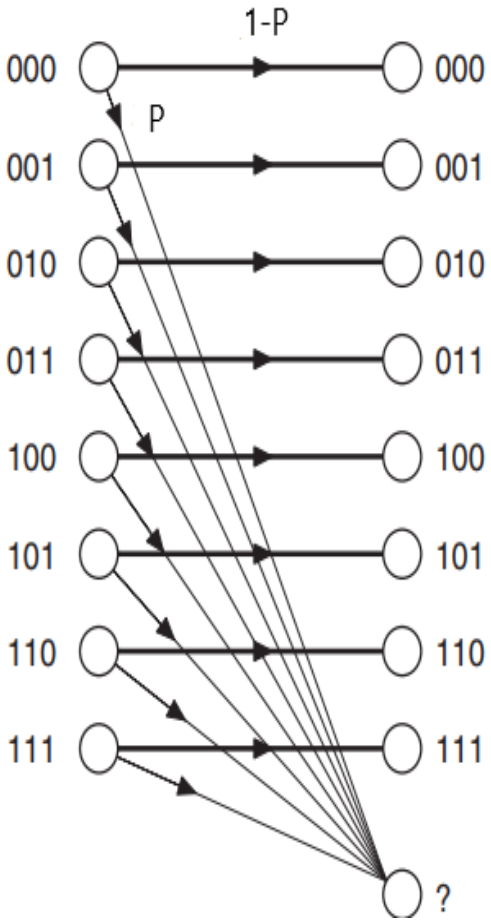

Рисунок 1.2 - Канал стирання – 8-ми канал стирання Вісім можливих входів {0; 1; 2; ... ; 7} тут показано двійковими пакетами 000; 001; 010; ... ; 111

мережевою архітектурою мережі. В ідеальному випадку кожен пакет повинен бути отриманий без помилок на стороні одержувача.

Але деяка кількість пакетів може бути не доставлена одержувачу ("втрачена"), а деякі з них доставлені з бітами помилок, які виявляє одержувач. Тому такі пакети ми повинні розглянути «загублений». Канали стирання настільки важливі, тому що багато каналів зв'язку з втратами або шумами можна змоделювати або спростити, щоб наблизити канал стирання. Еліас представив концепцію каналів стирання в 1955 році [3].

Простою моделлю каналу, що описує цю ситуацію, є канал зі стираннями (рис. 1.2).

Модель цього каналу має q – входів і q+1 виходів. Його розмір алфавіту q=2k, де k — кількість бітів у пакеті. Вихід зі знаком «?» ми позначили як стирання. The ймовірність P доставки результату '?' і 1-P ймовірність передачі вхідних даних без помилок.

Інтернет є хорошим прикладом каналів стирання пакетів і каналів стирання бітових помилок. У мережах зв’язку вихідні дані містяться в окремих пакетах, а потім пересилаються до адресатів. З кількох причин деякі з переданих пакетів випадково втрачаються в дорозі з ненульовою ймовірністю. Коли пакет надходить до місця призначення, він перевіряється з

виявленням помилок. Якщо виявлені помилки неможливо виправити, цей пакет буде відкинуто та/або позначено як стирання. Для зв’язку «точка-точка» традиційна послуга найкращих зусиль може компенсувати всі втрачені дані, якщо дозволено достатній час передачі. Однак для сучасних мультимедійних послуг типу «точка-багато точок», які користуються високим попитом, послуга найкращих зусиль більше не задовольняє велику потребу в корекції стирання з точки зору пропускної здатності, надійності та затримки тощо.

Для виявлення помилок у пакетах ми зазвичай використовуємо код. У таких мережах ми можемо зіткнутися з двома проблемами: 1) Втрата бітів 2) Втрата пакетів. Рішення цих двох проблем ми спробуємо показати тут.

1.3 Модеми

Під поняттям «модем» ми розуміємо пристрій, який перетворює цифрову інформацію, що поступає з одного комп’ютера, у аналогові сигнали, які можуть передаватися по телефонним лініям, та виконує зворотну операцію на приймальному кінці системи передавання. Модем об’єднує у собі два поняття: модулятор та демодулятор. Модулятор перетворює інформацію, яка надходить з комп’ютера у двійковому вигляді, в аналогову форму, що дозволяє передавати її через звичайні телефоні лінії. Демодулятор з сигналу, що надходить, отримує закодовану двійкову інформацію та передає її до комп’ютера на приймальному боці. При цьому дуже важливо використовувати однакові методи кодування/декодування і на прийомі, і на передачі.

Якщо уявити це графічно, то аналогові сигнали є синусоїдальними та косинусоїдальними хвилями, а цифрові сигнали представляються у вигляді прямокутних хвиль. Наприклад, звук є аналоговим сигналом, оскільки звук завжди змінюється. Аналоговий сигнал є тією інформацією, яка передається безперервно, у той час як цифровий сигнал передає тільки ті дані, які визначені у конкретний момент часу. Головною перевагою аналогових

сигналів є можливість повністю представити потік інформації. Проте аналоговий сигнал, у порівняні з цифровим, набагато більше схильний до впливу різних шумів та завад.

Однією з причин такого різновиду модемів у наш час є те, що не існує ніяких стандартів на конструкцію модемів. Це призводить до того, що в модемах одного виду одні і ті самі протоколи та методи можуть бути реалізовані по різному. Проте майже усі модеми мають схожі схеми, до яких входять універсальний (PU), центральний сигнальний (DSP) та модемний процесори, адаптери портів канального і DTE-DCE інтерфейсів, постійний (ПЗП, ROM), постійний енергонезалежний перепрограмувальний (ППЗП, ERPROM) оперативний (ОЗП, RAM) запам'ятовуючі пристрої і схеми індикаторів стану модему.

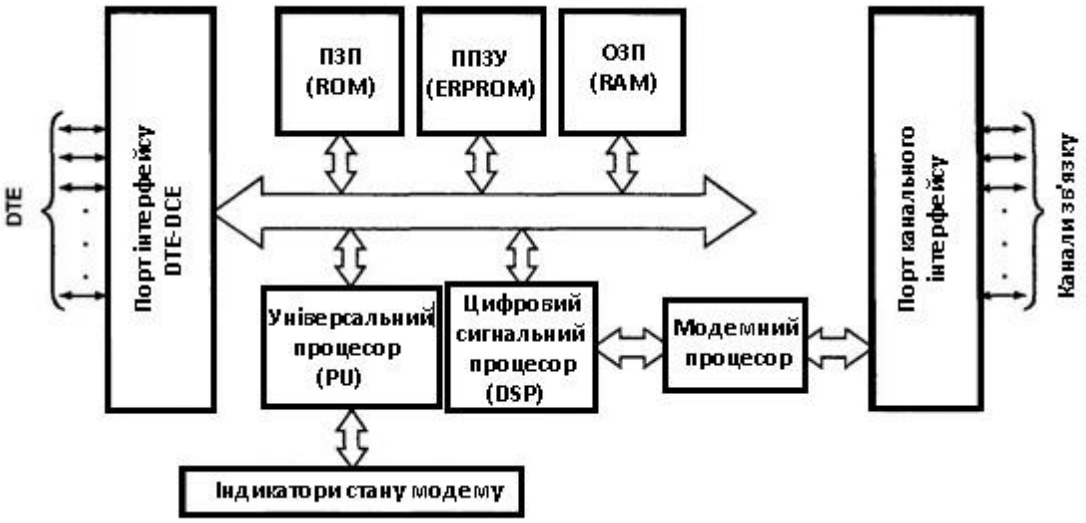

Рисунок 1.3 – Структурна схема сучасного модему

Універсальний процесор забезпечує управління взаємодію з DTE та схемами індикації станів модему. Також він відповідає за прийом та виконування команд , обробку даних, управління сигнальним процесором та буферизацію. Він виконує AT- команди (які потрібні для конфігурації та

управління роботою модему), що надсилає DTE та керує режимами роботи інших складових частин.

Сигнальний процесор реалізує основні функції протоколів модуляції. Модемний процесор, у свою чергу, виконує модуляцію/демодуляцію та розділяє спектр частот

DTE (data terminal equipment) – обладнання, яке формує чи приймає інформацію. DCE (data communication equipment) – це і є модеми. Порт інтерфейсу DTE-DCE забезпечує взаємодію з DTE. Порт канального інтерфейсу забезпечує узгодження електричних параметрів з використовуваним каналом зв'язку.

ПЗП зберігає програми для універсального та центрального сигнального процесорів. Крім цього, також у ПЗП зберігається мікропрограма управління – набір даних та команд, за допомогою яких здійснюється управління роботою модему. Програмний запам’ятовуючий пристрій може бути різних видів у залежності від кількості можливих операцій перепрограмування.

ППЗП забезпечує зберігання налаштувань модему у спеціальних файлах (профайлах) від час відключення. Як правило, існує основний та додатковий профайли, які використовуються для ініціалізації.

ОЗП використовується для тимчасового зберігання інформації, виконання обчислень процесорами (як універсальним, так і цифровим сигнальним). Також в ОЗП зберігається поточний набір налаштувань модему.

Модуляція – процес перетворення сигналу, що надходить з джерела інформації, у таку форму, яка дає можливість передавання по лінії або, кажучи іншими словами, це процес узгодження характеристик сигналу з характеристиками каналу зв’язку.

Як правило, повідомлення, які надходять з джерела, неможливо безпосередньо сприймати та передавати, тому у процесі передавання інформація зазнає неодноразового кодування. За призначенням коди можна

розділити на коди джерела, коди аналого-цифрового перетворювача, коди лінії, модуляційні коди та коректувальні коди.

Кодування – операція ідентифікації символів або груп символів з одного коду символами або групами символів в інший код. Виникає необхідність кодування. Перш за все, від вимоги адаптувати форму повідомлення до даного каналу зв'язку або до будь-якого іншого пристрою, призначеного для передачі чи зберігання інформації [4]. Типова блок-схема цифрової телекомунікаційної системи наведена нижче.

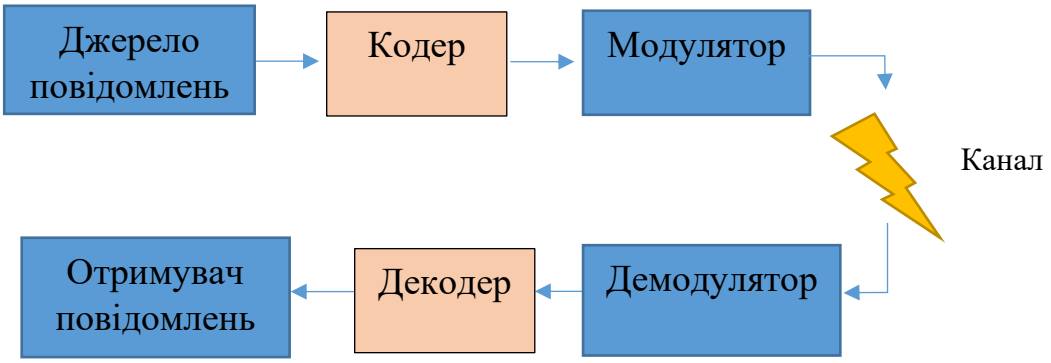

Рисунок 1.4 - Типова структурна схема цифрової телекомунікаційної системи.

Кодування контролю помилок стосується методів доставки інформації від джерела до пункту призначення з мінімальною кількістю помилок. Як таку її можна розглядати як галузь теорії інформації та бере свій початок від роботи Шеннона наприкінці 1940-х років. Рання теоретична робота вказує на те, що можливо, і дає деяке уявлення про загальні принципи контролю помилок. З іншого боку, проблеми, пов’язані з пошуком і впровадженням кодів, призвели до того, що практичні ефекти від використання кодування часто дещо відрізняються від очікуваних спочатку.

У реальних умовах довжина цього коду обмежена допустимою складністю пристрою кодування або декодування. Таким чином, успішний результат помилково-контрольного кодування залежить від параметрів коду

та обмежень на реалізацію канального кодека. У наш час теорія пропонує велику кількість кодів контролю помилок. Різні принципи структури та побудови та їх можливості (виявлення та виправлення помилок).

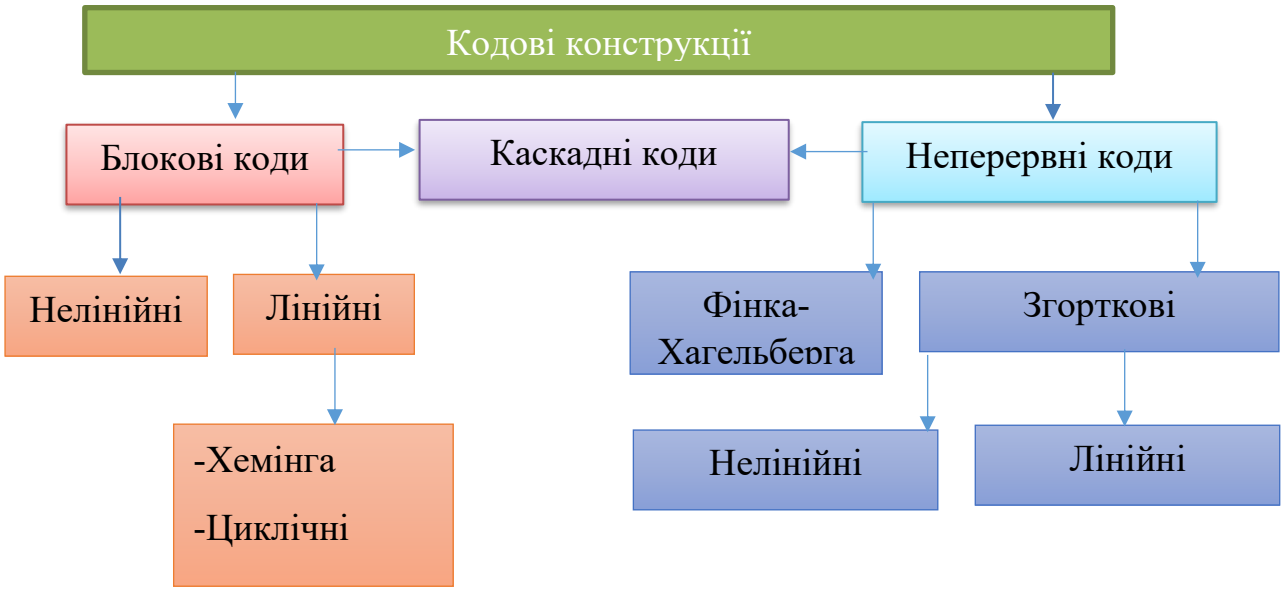

Рисунок 1.5 – Стандартна структура класифікації кодів

В основному існує два типи методів кодування з прямим виправленням помилок (FEC): лінійне блокове кодування та згорткове кодування [5]. Але тепер для сучасних телекомунікацій (у нашому випадку мереж з комутацією пакетів) можна використовувати нову класифікацію, яка представляє собою такі конструкції кодування. В сучасних телекомунікаційних технологіях класичні коди використовуються лише у складі більш ефективних турбоподібних зчеплених структур. Також, враховуючи специфіку розвитку телекомунікаційних технологій, запропоновано нові базові конструкції, що застосовуються як окремо, так і як елементи каскадних структур. Еволюція підходів до декодування структур коду полягає в переході від алгебраїчного жорсткого декодування та імовірнісного м'якого декодування до ітераційного декодування, що дає можливість використовувати як жорсткі, так і м'які рішення.

У цій роботі дослідимо найбільш розповсюджені коди контролю помилок:

1) Коди, які допомагають знаходити бітові помилки: коди Ріда - Соломона, каскадні коди, турбо-коди.

2) Коди, які допомагають доставляти пакети в заданих мережах: фонтанні коди LT і Raptor.

1.4 Блокові коди

Кодер блокового коду розділяє інформаційну послідовність на блоки повідомлень по k інформаційних біт кожен. Блок повідомлення представлено двійковим k-кортежем $u = (u_1, u_2 \ldots u_n)$, який називається повідомленням. (Тут ми використовуємо u для позначення k-бітного повідомлення, а не всього джерела інформації). Всього існує 2 тисячі різних можливих повідомлень. Кодер перетворює кожне повідомлення u незалежно в n-кортеж $v = (v_1, v_2 \ldots v_n)$ дискретних символів, який називається кодовим словом. (Тут ми використовуємо v для позначення n-символьних блоків, а не всієї закодованої послідовності).

Таблиця 2.1- Параметри блокового кодування контролю помилок

Параметр	Визначення	Формула
n	Це кількість бітів на виході кодера (кодове слово)	
k	Кількість бітів на вході інформаційної послідовності кодера	
r	Кількість додаткових бітів	r=n-k
R	Швидкість коду. Можна інтерпретувати як кількість інформаційних бітів, що надходять у кодер на переданий символ.	k/n
χ	Надмірність	1-R
M	Кількість кодових слів на вході декодера	2^n
$M_{доз}$	Кількість дозволених кодових слів	$2^{тис}$
$M_з$	Кількість заборонених кодових слів	$M - M_{доз}$
d_{min}	Мінімальна відстань між двома сусідніми кодовими словами	

$q_{вип}$	Кількість виправлених помилок	
$q_{вияв}$	Кількість виявлених помилок	d_{min}-1

1.4.1 Коди Ріда-Соломона (RS-коди)

Коди Ріда-Соломона — це блокові коди для виправлення помилок із широким спектром застосувань у цифровому зв’язку та зберіганні. Він вразливий до випадкових помилок, але сильний до пакетних помилок. Код Ріда Соломона — це окремий випадок коду БЧХ, у якому довжина коду на одиницю менша за розмір поля, у якому визначені символи.

Існує принаймні два типи кодерів Ріда-Соломона: *несистематичні* та *систематичні*. Розрахунок несистематичних *коригувальних* кодів Ріда-Соломона здійснюється шляхом множення інформаційного слова на згенерований поліном, в результаті чого утворюється кодове слово, повністю відмінне від вихідного інформаційного слова, і тому для прямого використання воно повністю непридатний. Щоб привести отримані дані в початковий вигляд, ми обов'язково повинні виконати ресурсомістку операцію декодування, навіть якщо дані не спотворені і не потребують відновлення. При *систематичному* кодуванні вихідне інформаційне слово буде незмінним, а в його кінець додаються коригуючі коди, так що до операції декодування потрібно вдаватися тільки в тому випадку, якщо дані фактично знищені. Розрахунок несистематичних коригуючих кодів Ріда-Соломона виконується шляхом ділення інформаційного слова на згенерований поліном. У цьому випадку всі символи інформаційного слова зсуваються на n - k байт вліво, а 2t байт залишку записуються на вільне місце.

Кодер Ріда-Соломона приймає блок цифрових даних і додає додаткові «зайві» біти. Помилки виникають під час передавання або зберігання з кількох причин (наприклад, шум або перешкоди, подряпини на компакт-диску тощо). Декодер Ріда-Соломона обробляє кожен блок і намагається виправити

помилки та відновити вихідні дані. Кількість і тип помилок, які можна виправити, залежить від характеристик коду Ріда-Соломона.

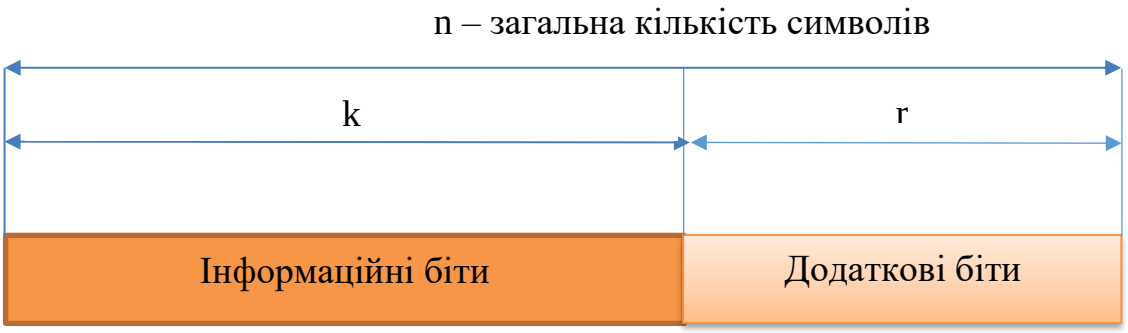

Рисунок 1.6 - Структура систематичного кодового слова Ріда-Соломона

Коди Ріда-Соломона використовуються для виправлення помилок у багатьох системах, зокрема:

- Пристрої зберігання даних (зокрема стрічки, компакт-диски, DVD, штрих-коди тощо)
- Бездротовий або мобільний зв'язок (включаючи стільникові телефони, мікрохвильові канали тощо)
- Супутниковий зв'язок
- Цифрове телебачення / DVB
- Високошвидкісні модеми, такі як ADSL, xDSL тощо.

Процес кодування:

крок 1) Обчислити кількість додаткових бітів r

крок 2) Виберіть генераторний поліном g(x) (Де максимальний порядок g (x) = r)

крок 3) Перетворення вхідної послідовності на поліноміальну форму a(x)

крок 4) Помножте a(x) * x^2

крок 5) Розділити результат g(x)

крок 6) Знайдіть нагадування про цей розрахунок

крок 7) Щоб додати нагадування до кроку 4

Процес декодування:

крок 1) Перетворіть заданий сигнал у поліноміальну форму

крок 2) Розділіть крок 1 на вибраний поліном генератора

крок 3) Отримайте синдром (це нагадування про крок 2)

крок 4) Виберіть синдром із таблиці синдромів

крок 5) Виправте відповідні біти

крок 6) Виберіть перші k бітів із даного повідомлення

Перевага використання кодів Ріда-Соломона полягає в тому, що ймовірність помилки, що залишається в декодованих даних, (зазвичай) набагато нижча, ніж ймовірність помилки, якщо Ріда-Соломона не використовується. Це часто описують як виграш кодування.

1.4.2 Каскадні коди

Конкатенація (каскадування) — це метод побудови довгих кодів із коротших; він намагається вирішити проблему складності декодування, розбиваючи необхідні обчислення на керовані сегменти. Ми представляємо теоретичні погляди на ефективність і складність конкатенованих кодів; основні теоретичні результати такі:

1. Конкатенація як завгодно великої кількості кодів може призвести до ймовірності помилки, яка експоненціально зменшується із загальною довжиною блоку, тоді як складність декодування зростає лише алгебраично; і

2. Конкатенація кінцевої кількості кодів дає показник похибки, який є нижчим від того, що досягається за допомогою одного етапу, але відмінний від нуля на всіх швидкостях, нижчих за пропускну здатність. Обчислення підтверджують ці теоретичні результати, а також дають зрозуміти взаємозв’язок між модуляцією та кодуванням [7].

Каскадне кодування вперше було запропоновано Форнеєм як метод для досягнення великих переваг кодування шляхом поєднання двох або більше відносно простих будівельних блоків або компонентних кодів**.** Конкатенований код був запропонований як практичний метод побудови типу

коду з більшою довжиною кодового слова та кращою продуктивністю виправлення помилок, і це спеціальний метод, за яким довгий код може складатися з кількох коротких кодів. По суті, конкатенований код є окремим випадком коду продукту, і найбільш застосовуваний конкатенований код складається з двох кодів [7].

Приклад того, як це працює. Інформація для передачі спочатку кодується за допомогою (N, K) кодера, який на малюнку позначений як зовнішній код. Потім символи в (N, K) кодовому блоці розглядаються як інформаційний потік, який кодується як послідовність (n, k) коду внутрішнього блоку. Поєднання внутрішнього кодера, каналу та внутрішнього декодера можна вважати утворенням нового каналу, який називається «суперканал», який передає двійкові k-кортежі. Часто в якості зовнішнього коду використовується код RS. Загальний код є двійковим кодом довжини nN, розмірності kK. Зрозуміло, що якщо алфавіти символів внутрішнього та зовнішнього кодів не збігаються, необхідно переформатувати дані між кодерами для внутрішнього та зовнішнього кодів. Кодування виконується наступним чином. KK двійкових інформаційних бітів поділено на K k-кортежів, які вважаються елементами $GF(2^k)$. Потім вони кодуються зовнішнім кодувальником у кодове слово $a_0 a_1 \ldots a_{N-1}$ з $a_i \in GF(2^k)$. Кожен a_i тепер кодується внутрішнім кодувальником у двійковий n-кортеж b_i. Тоді $b_0 b_1 \ldots b_{N-1}$ - кодове слово , яке буде передано по каналу. Таким чином, ми отримуємо зчеплений блоковий код, який має довжину блоку Nn біт і містить kK інформаційних біт. Тобто ми створили еквівалентний (Nn, Kk) двійковий код. Швидкість отриманого каскадного коду дорівнює (k n) (KN), що дорівнює добутку двох швидкостей коду. Крім того, мінімальна відстань зчепленого коду становить $d_{min}D_{min}$, де D_{min} — мінімальна відстань зовнішнього коду, а d_{min} — мінімальна відстань внутрішнього коду.

Зрозуміло, що роль зовнішнього коду полягає у виправленні помилок, які залишаються після декодування внутрішнього коду, помилок, які можуть

демонструвати статистику незалежних або пакетних помилок. Крім того, швидкість зовнішнього коду має бути підібрана ретельно з наступних причин. Якщо зовнішня кодова швидкість дуже висока, вона може не мати належної можливості виправлення помилок, і тому помилки, що виходять із внутрішнього декодера, не можуть бути надійно виправлені. Однак, якщо E_b/N_0 і швидкість внутрішнього коду є фіксованими, тоді, коли ми зменшуємо швидкість зовнішнього коду (для покращення його можливостей виправлення помилок), SNR на вході внутрішнього коду зменшується. Якщо швидкість зовнішнього коду знижується занадто сильно, SNR може знизитися до точки, де буде втрата кодування, а не посилення, пов'язане з внутрішнім кодом. Таким чином, ми можемо очікувати існування оптимальної швидкості для зовнішнього коду. Проблема розробки для вибору внутрішнього коду зчепленого коду дещо відрізняється від тих, що стосуються інших програм. Небажано, щоб внутрішній код мав дуже круту характеристику, тобто криву водоспаду з яскраво вираженим коліном. Якщо такий код було обрано як внутрішній код, а номінальна проектна точка була обрана трохи правіше коліна, результатом буде неефективна та нестабільна конструкція. Це пояснюється тим, що якщо якість каналу хоча б незначно погіршиться, частота помилок на вході зовнішнього декодера різко зросте, а зовнішній декодер буде перевантажений помилками. З іншого боку, якщо SNR трохи збільшиться, внутрішній код сам по собі зможе забезпечити бажану якість обслуговування. У цьому випадку зовнішній код спричиняє загальну втрату ефективності зв'язку.

1.4.3 Турбо коди

Турбо-коди Клас турбо-кодів, представлений у 1993 році, обіцяє знизити рівні BER до раніше недосяжних рівнів для заданої складності коду [8]. Завдяки їхній чудовій продуктивності поблизу межі Шеннона, вони знайшли застосування в Консультативному комітеті систем космічних даних

(CCSDS), стандарті 3GPP/UMTS, супутниковому та наземному зворотному каналі цифрового відеомовлення (DVB-RCS та DVB-RCT), системи бездротового зв'язку 3GPP2/cdma2000 та стандарти IEEE.802.16 WiMAX [9].

Типовий турбокодер складається з двох кодерів RSC, розділених перемежувачем. Стандарт турбокоду UMTS вимагає, щоб турбокодери, що входять до його складу, починалися та закінчувалися у відомому стані (все нульовий стан). Ця відома інформація про стан використовується на стороні приймача для початку процесу декодування. Завершення обох кодерів RSC у нульовому стані може бути досягнуто лише шляхом надсилання певної послідовності кінцевих бітів для кожного кодера через цикл зворотного зв'язку в кодерах RSC. Кінцеві біти для кожного кодера RSC залежать від полінома генератора, а також послідовності вхідних даних. Таким чином, необхідні кінцеві біти для завершення кожного з кодерів RSC будуть різними. Інформацію про кінцеві біти також слід надсилати разом із кодовими словами в турбо кодуванні. У літературі існує ряд різних методів завершення решітки, таких як завершення складових кодерів RSC, завершення лише одного з кодувальників або без використання завершення взагалі, а також хвостові турбо-коди, які визначають початковий і кінцевий стани шляхом вибору перший стан як функція вхідної послідовності та решітки [10]. Дві основні проблеми, пов'язані з завершенням решітки, – це погіршення продуктивності декодування біля кінця послідовності даних та вплив завершення решітки на спектр відстані коду [10]. Іншим фактом про методи завершення решітки турбо-коду є те, що їх продуктивність сильно залежить від перемежувач використовується турбокодером. Ця залежність є результатом так званих краєвих ефектів перемежувача, які погіршують спектр відстані коду [10], [11].

Декодери засновані на техніці soft-in-soft-out (SISO). На стороні декодера систематичний вхід і дві закодовані послідовності даних з двох кодерів подаються як вхідні дані. Спочатку декодер намагається декодувати різні вхідні дані в порядку. Потім дані повертаються через канал зворотного

зв’язку. Декодер ітеративно декодує надані йому вхідні дані; після кількох ітерацій ми можемо зробити досить хорошу оцінку біта даних, який було передано, і, як результат цього механізму зворотного зв’язку, турбо-коди ефективно досягають пропускної здатності Шеннона.

Використання зворотного зв'язку вимагає існування алгоритмів декодування SISO для кодів обох компонентів. Оскільки алгоритм м’якого виведення Вітербі **(**SOVA) вже був доступний на момент винаходу, застосування згорткових кодів виглядало природним для обох кодів. З міркувань ефективності пропускної здатності послідовне об’єднання замінено на паралельне об’єднання. Насправді, паралельне об’єднання двох кодів зі швидкістю R_1 і R_2 дає глобальну швидкість Rp , що дорівнює:

$$Rp = \frac{R1R2}{1-(1-R1)(1-R2)}$$

Ця швидкість вища, ніж у послідовно з’єднаного коду, який:

$$Rs = R_1R_2$$

для однакових значень R_1 і R_2, і чим нижче ці показники, тим більше різниця [12].

2 ПАКЕТНА ПЕРЕДАЧА ДАНИХ

2.1 Проблема втрат пакетів

Коли комп’ютер під’єднується до інтернету або іншої мережі, відбувається обмін маленьких блоків даних, які називаються пакетами. Коли передача одного або декількох пакетів не відбувається, це називають втратою пакетів. Для користувача це проявляється у вигляді повільного завантаження будь-яких даних, низької якості підключення або повної втрати з’єднання з мережею.

У сучасному житті ми не уявляємо своє життя без мобільних систем і комп'ютерних мереж. З кожним роком підвищуються вимоги до швидкості та достовірності інформації, що передається. Такі системи передачі даних піддаються різним завадам і шумам, що призводить до спотворення або зникнення частини інформації.

Є кілька причин втрати пакетів:

- Помилки програмного забезпечення,
- Проблеми обладнання (заліза),
- Перевантаженість мережі,
- Бездротові та дротові мережі,
- Кібернетична атака.

Деякі пакети можна відновити за допомогою спеціальних завадостійких кодів.

Для вирішення цієї проблеми використовуються різні методи, засновані на кодуванні інформації з контролем помилок. У цій роботі розглядаються методи кодування, які використовуються в каналах помилок, які називаються «фонтанними кодами» [13]. Головна особливість полягає в тому, що коди з цього класу можуть передавати кінцеве повідомлення з нескінченною кількістю пакетів.

2.2 Фонтанні коди

Фонтанні коди - це рекордні коди скопійованих графів для каналів зі стираннями. Такі як Інтернет. Коли файли передаються кількома невеликими пакетами, кожен з яких приймається без помилок або не приймається взагалі. Як ми вже знаємо, у мережі з комутацією пакетів стандартний файл розрізається на шматки розміром K пакетів. Потім повторно передає їх, поки не буде успішно отримано [14]. Але що, якщо у нас немає можливості чекати повторної передачі втрачених файлів? Наприклад, в IP-телефонії. У такому випадку нам краще використовувати коди фонтанів, щоб вирішити цю проблему. Навпаки, фонтанні коди створюють пакети, які є випадковими функціями всього файлу.

Що таке коди фонтану? Ось загальна картина теоретичного цифрового фонтану над каналом без стирання пам’яті. Запитуваний файл ділиться на k блоків однакового розміру, і кожен блок інкапсулюється в пакет.

Враховуючи ці k пакетів, фонтанний кодер потенційно може генерувати необмежений потік незалежних і однаково розподілених пакетів кодування. Ці пакети кодування передаються по каналу стирання, але лише частина пакетів отримується без помилок, а інші втрачаються або відкидаються.

Потім фонтанний декодер з високою ймовірністю відновлює вихідні пакети з будь-якої підмножини отриманих пакетів кодування з числом n, рівним або більшим за k. Цей процес схожий на те, що ми використовуємо чашку, щоб навмання зібрати кілька крапель із фонтану, і коли в чашці буде достатньо крапель, ми зможемо втамувати спрагу [14].

Фонтанні коди є кодами з меншою швидкістю, оскільки як на стороні передавача, так і на стороні приймача кодова швидкість не є фіксованою і потенційно може наближатися до нуля. Однак на практиці потрібно просто враховувати усічені коди фонтанів. Відповідно до вимог програми, як-от рівень успішного декодування та вартість обчислень, може бути призначена мінімальна кількість отриманих пакетів для задоволення вимог. Наприклад, це

мінімальне число дорівнює n, і декодер відновлює дані зі швидкістю kn, усічений фонтанний код. Для ідеального або оптимального фонтанного коду кодова швидкість має бути 1. Коли k трохи менше n, фонтанний код є субоптимальним або близьким до оптимального. Сьогодні існує два основних класи практичних фонтанних кодів: коди Лабі (Luby Transform, LT) і коди Raptor. Обидва вони майже оптимальні для кінцевої довжини даних.

2.2.1 Коди LT

Винайдені Лабі у 1998 році коди LT є першим класом практичних фонтанних кодів. Більш детально ви можете прочитати в [15]. Кодування та декодування в цьому випадку досить тривіальні, але дають хороший результат. Для кращого розуміння LT-кодів опишемо, як відбувається процес його кодування.

Припустимо, що у нас є початкове повідомлення, що складається з K початкових символів або пакетів. Довжини всіх символів еквівалентні і дорівнюють L. Введемо поняття ступеня i-го кодового символу d_i , де i = 0,…,N, де N – кількість кодових символів, згенерованих кодувальником . N потенційно нескінченна. Ступінь кодового символу d - це кількість вихідних символів, які беруть участь в операції кодування для генерації i -го кодового символу. Початкові символи, які були використані для генерації деякого символу коду i, тут будуть називатися «сусідами» [15],[16].

Введемо також щільність розподілу f(d): для всіх d, f(d) існує ймовірність того, що кодовий символ має ступінь d. Щільність розподілу потужностей кодових пакетів є дискретною функцією і може бути задана як аналітично, так і як набір значень f(1), f(2), ..., f(M-1), f(M), де M – максимальний ступінь кодових символів.

Нижче наведено опис процесу генерації кодового символу, запропонованого М. Лабі [15]:

- Випадково обирають ступінь d кодового символу з щільності розподілу ступенів символів f (d);
- Виберіть навмання d різних вихідних символів як сусідів із кодовим символом;
- Значення символу коду встановлюється як результат операції "виключаюче або" (XOR) над d обраними сусідами.

Процес генерації кодових символів повторюється до тих пір, поки початкове повідомлення не буде повністю декодовано, або на певному кроці подальше декодування стане неможливим через відсутність кодових символів потужністю 1 [15],[16] .

Ключовим аспектом розробки кодів RT є розробка щільності ймовірності f(d). Зарубіжними авторами запропоновані різні методи побудови щільності розподілу ймовірностей [15],[16].

Параметрами робастного розподілу є: c - параметр розподілу; подається як позитивне число, рекомендовано, щоб $0<c<1$; δ - параметр, що визначає ймовірність успішного декодування; Процес декодування буде успішно завершений з імовірністю (1-δ) [16].

Значення $K \cdot \sum p(d) + \tau(d)$ визначає кількість кодових символів, необхідних для успішного декодування послідовності з імовірністю (1 - δ). Робастне розподіл призначене для вирішення наступних завдань: визначення кількості пакетів з ступеня 1, необхідного для успішного повного декодування послідовності, а також, завдяки піку в точці K / S, збільшення ймовірності того, що кожен з вихідних символів буде прийнятий як сусід хоча б один раз.

2.2.2 Структура та визначення коду Raptor

Двійковий код Raptor будується як послідовна конкатенація зовнішнього блокового коду та внутрішнього коду LT. Зовнішній код зазвичай є блоковим кодом, який кодує k-бітний інформаційний вектор у k-бітне кодове слово, де біти кодового слова називаються вхідними символами.

Потім за допомогою коду LT потенційно необмежена кількість кодованих Raptor символів, які називаються вихідними символами, генерується та передається по каналу, де кожен вихідний символ є XOR (сумою за модулем 2) випадково вибраної частини вхідних символів. Точніше, для генерації кожного вихідного символу спочатку ціле число d, яке називається ступенем, отримується із попередньо визначеної масової функції ймовірності, званої функцією розподілу ступеню. Вихідний символ потім генерується за допомогою операції XOR на d різних вхідних символах, рівномірно вибраних випадковим чином.

Граф декодування повинен мати порядок ребер, щоб переконатися, що всі вхідні вузли охоплені з високою ймовірністю. Ідея кодування Raptor послаблює цю умову та вимагає, щоб лише постійна частка вхідних символів була відновлюваною. Тоді той самий інформаційно-теоретичний аргумент, що й раніше, показує лише лінійну нижню межу для кількості ребер у графі декодування. Існує дві потенційні проблеми з цим підходом: 1) Теоретико-інформаційна нижня межа може не співставлятися з алгоритмом, і 2) нам потрібно відновити всі вхідні символи, а не лише постійну частку.

3 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МОДЕМІВ

У частині 2 ми представляємо теоретичні відомості про кодування з контролем помилок. Тут ми наводимо деякі експериментальні результати надання кодів. Для виправлення бітових помилок ми обрали блочні коди, каскадні коди та турбо-коди.

З блокових кодів ми взяли код Ріда – Соломона. Добре відомо, що коди Ріда-Соломона, які використовуються як коди виправлення помилок, здатні виправляти будь-яку комбінацію (n − k)/2 або менше помилок. Навпаки, коди Ріда-Соломона, коли вони використовуються як коди стирання, здатні виправляти (n−k) стирань з будь-якого успішно отриманого набору з k символів. Коди Ріда-Соломона є потужними лінійними блоковими кодами виправлення помилок, але неефективність і обмеження можуть бути очевидними при їх використанні як кодів стирання на рівні пакетів. Фундаментальне визначення коду Ріда-Соломона накладає практичне обмеження: алгоритм Ріда-Соломона вимагає виконання операцій над розширеним полем Галуа, але для того, щоб бути доступним, розмір елемента поля зазвичай обмежується 8 бітами (одним байтом), особливо якщо реалізація має бути на основі програмного забезпечення або якщо потужність обробки обмежена.

У нашому першому експерименті ми використовуємо модуляцію КАМ-16 і блокуємо код Ріда-Соломона для дослідження ймовірності виправлення помилок. Для візуалізації можливостей кодів Ріда-Соломона беремо наступні параметри, w довжина повідомлення k=247,120,502,1013 і довжина блоку n=247,120,502,1013.

Імовірність помилки в системі без кодування розраховується за формулою:

$$p = \frac{k_1}{k_2} Q(k_3 * h_b)$$

де Q-функція:

$$Q(z) = \frac{1}{\sqrt{2\pi}} \int_z^{\infty} \exp\left(-\frac{t^2}{2}\right) dt$$

Але для розрахунків доречно використовувати апроксимацію цієї функції:

$$Q(x) = 0{,}65 * e^{-0{,}44*(x+0{,}75)^2}$$

Отже, для КАМ-16 ми маємо

$$p = Q(0{,}89 * h_b)$$

Підставляючи ці формули в подальші розрахунки, ми повинні враховувати незначні зміни, внесені кодовою швидкістю:

$$p = \frac{k_1}{k_2} Q\left(k_3 * \sqrt{R_{cod}} * h_b\right)$$

Щоб розрахувати ймовірність помилки декодування в кодах RS, нам необхідно знати мінімальну кодову відстань і кількість виправлених помилок, яка становить:

$$q_{corr\ RS} = \left\lfloor \frac{n-k}{2} \right\rfloor$$

$$d_{\min\ RS} = 2 * q_{corr} + 1$$

Імовірність помилки декодування кодів RS розраховується за формулою:

$$p_{dec\ RS}(h_b) = \frac{d_{min}}{n} * \sum_{q=q_{corr}+1}^{n} C_n^q * p^q * (1-p)^{n-q}$$

Отже, будуємо графіки в MathCad коду Ріда-Соломона з обраними параметрами.

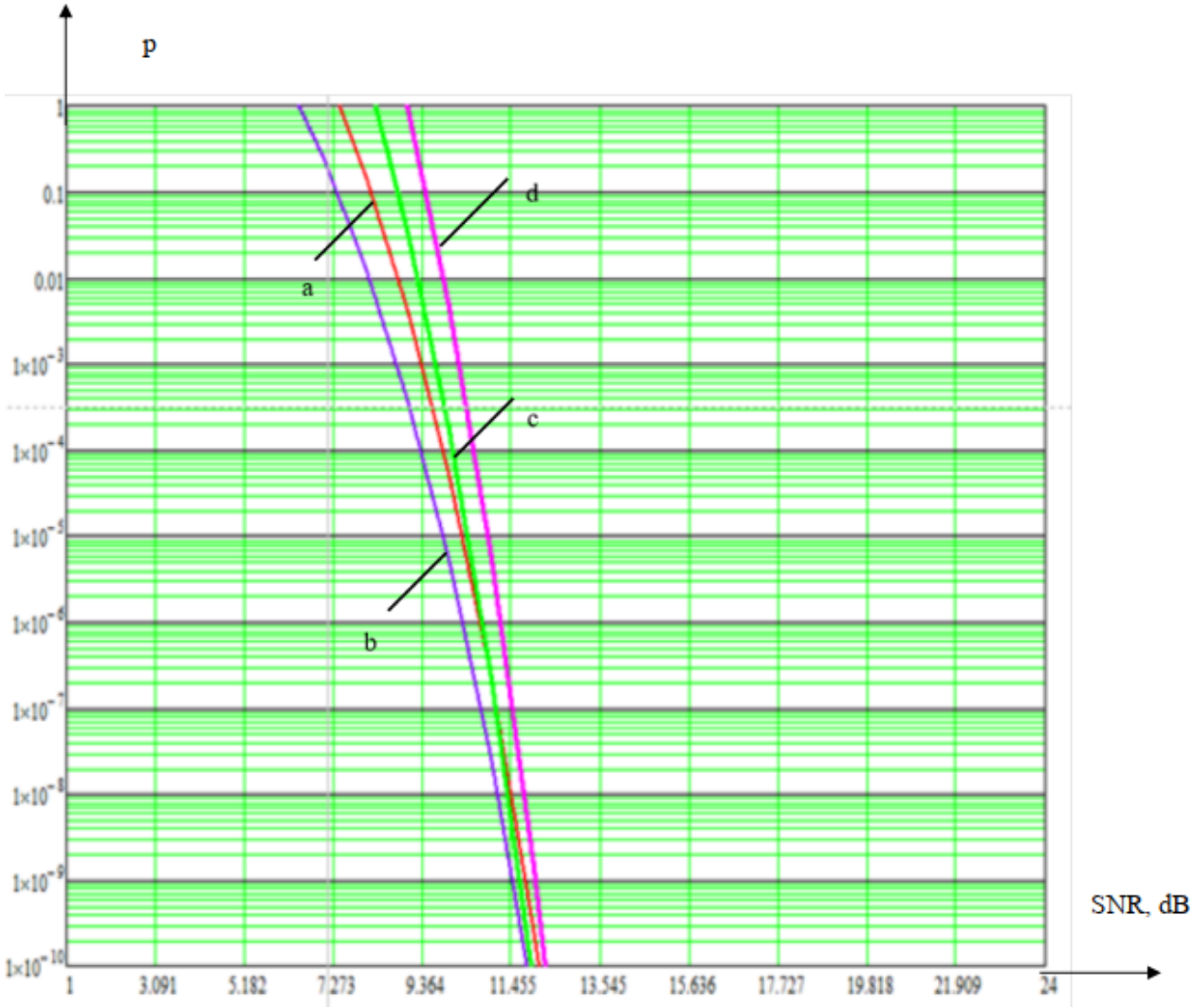

Рисунок 3.1 – Завадостійкість кодів Ріда-Соломона

a) PC (255;247) b) PC (127;120) c) PC (511;502) d) PC (1023;1013).

Потім ми проводимо експеримент із каскадними кодами та турбо кодами з однаковою модуляцією.

Імовірність помилки декодування турбо кодів розраховується за формулою:

$$p_{turbo}(h_b) = e^{R_{cod}*df*h_b^2} * \frac{k_1}{k_2} Q\left(k_3\sqrt{d_f R_{cod}} * h_b\right) * w_{df} * \left(e^{-R_{cod}*h_b^2}\right)^{d_f+1}$$

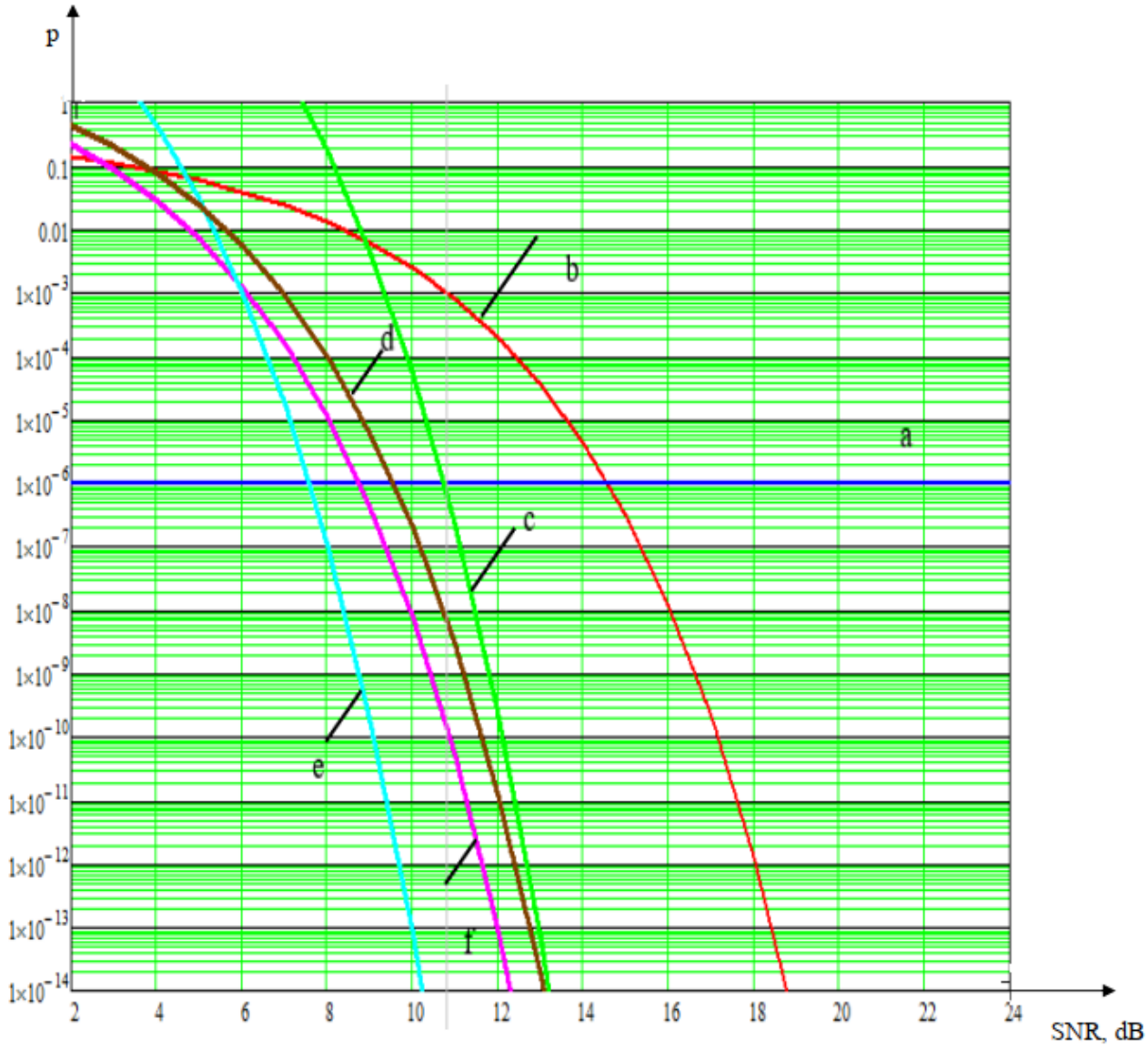

Рисунок 3.2 – Завадостійкість розглянутих кодових конструкцій:

a) допустима ймовірність помилки, b) КАМ-16, c) блоковий код Ріда-Соломона (255;247) d) турбо-код e) каскадний код f) згортковий код

Імовірність помилки в цьому випадку каскадного декодування така:

$$p_{dec\ cas} = \frac{d}{n} \sum_{q=q_{corr}+1}^{n} C_n^q * p_x^q * (1 - p_x)^{n-q}$$

Як ми бачимо, ці коди дають нам чудове виправлення помилок. Зауважимо, що класичні завадостійкі коди не здатні вирішити названу

проблему. Згорткові коди виправляють окремі бітові помилки. Блокові коди, включаючи коди Ріда Соломона, можуть виправляти купу помилок в одному пакеті. Але в разі втрати цілої упаковки вони безсилі. В даний час можна говорити про створення нового класу кодування каналів зі стираннями - кодів стирання. Коди цього класу можуть кодувати повідомлення кінцевого розміру з потенційно необмеженим потоком незалежних пакетів. Ця властивість нового класу кодів принципово відрізняє його від класичних блокових або згорткових, завадостійких кодів з фіксованою швидкістю. Для кодів з нового класу з'явився термін «rate less» (нефіксована швидкість). Їх також називають фонтанними кодами (цифровими кодами фонтанів). Найголовнішою відмінністю стирання кодів з блокових і згорткових кодів є можливість відновлення всього пакета в разі його втрати. В даний час коди стирання вже знайшли застосування в комерційних продуктах для комп'ютерних мереж.
Для експерименту вибираємо кодування LT. Для представлення властивостей фонтанних кодів.

Експериментальне дослідження ефективності кодів Лабі. Найважливішими характеристиками коду контролю помилок є здатність відновлювати дані з помилками та реалізувати кодування за величиною надмірності. Не менш важливою властивістю коду є надійність. Тому що можливості сучасних комп'ютерів повинні дати нам можливість реалізації кодування та декодування. У разі кодування контролю помилок у реальному часі реалізуються додаткові вимоги, оскільки кодування та декодування виконуються в реальному часі.

Дослідження кодів Лабі присвячені теоретичним питанням розробки символьних розподілів коду [15],[16] Моделювання коду Lab проводилось за допомогою програми Erasure Simulator [17], наведені результати були взяті з вимог до кодів LT, наведених Шинкаренко К.В., Коріковим А.М. [18].
Наступні кроки реалізації експерименту. Кодування виконано за алгоритмом LT з використанням робастного розподілу (див. частину 2.2.1). Значення

параметрів K, c та δ змінювалися в інтервалах: K = {100, 500, 1000, 5000}, c = {0.05, 0.2}, δ = {0.05, 0.2}. Відсоток втрати пакетів (PLR-Packet Loss Ratio) змінювався, приймаючи значення {0, 3, 5, 8, 15}.

В якості критерію ефективності LT-коду в експерименті використовувався відсоток успішно декодованих пакетів від загальної кількості вихідних пакетів. Результати експерименту є середнім відсотком успішно декодованих пакетів для R = 1000 стартів для кожного набору змінних параметрів коду.

Результати експерименту наведені в таблицях 3.1-3.3.

Таблиця 3.1 - Відсоток відновлення пакетів для LT: c = 0,05, δ = 0,05

Надмірність пакетів *N/K*, %	PLR, %	Довжина вихідної послідовності K, у пакетах			
		100	500	1000	5000
5	0	20	14.4	20	28.94
	3	3.8	9,88	5.47	7.36
	5	5.64	6.68	14.67	12,89
	8	18.28	16.4	20.97	4,95
	15	7,88	9.24	6.8	1.16
10	0	32	23.8	8.8	30,68
	3	21.24	18.68	6.06	2.17
	5	9.56	9.74	4.4	12.45
	8	3.68	5.58	6.27	15.4
	15	20.46	2.18	10.83	6.39
20	0	99	100	98.2	100
	3	2.88	90,69	99.02	100
	5	1	75,62	53,52	100
	8	42,98	23.96	27,89	100
	15	22.4	12.91	12.11	8.56
	0	100	100	100	100
	3	62,52	100	100	100

30	5	28.64	100	99,99	100
	8	59,58	86,78	96,74	100
	15	84,28	98.14	38.06	100

Таблиця 3.2 - Відсоток відновлення пакетів для LT: $c = 0{,}2$, $\delta = 0{,}05$

Надмірність пакетів N/K, %	PLR, %	Довжина вихідної послідовності K, у пакетах			
		100	500	1000	5000
5	0	100	100	100	99.04
	3	97,6	98.08	98.12	98,78
	5	93,66	95,99	96,41	98,67
	8	89.18	89.13	88.07	97,32
	15	77.7	74,53	77,41	83.16
10	0	100	100	100	99,36
	3	98,54	98,85	98,8	98.3
	5	97,36	97,72	98,79	98.01
	8	95.1	96,53	98,52	98.16
	15	82,96	82,65	81.06	81.9
20	0	100	100	100	99,84
	3	99,46	99,25	99.3	99.14
	5	99.08	98,82	98,96	99.01
	8	98.16	98,62	99,38	98,8
	15	89,74	98.12	98,6	98,41
30	0	100	100	100	100
	3	99,78	99,9	99,92	99,61
	5	99.06	99,8	99,83	99.14
	8	97,44	99,64	99,76	98,87
	15	94,76	99,78	99,91	98,46

Таблиця 3.3 - Відсоток відновлення пакетів для LT: $c = 0,2$, $\delta = 0,2$

Надмірність пакетів N/K, %	PLR, %	Довжина вихідної послідовності K, у пакетах			
		100	500	1000	5000
5	0	100	100	100	100
	3	97.7	97,78	97,65	97,32
	5	95.04	95,69	96.31	96.15
	8	92.32	89,98	90.7	85,34
	15	80,76	80.4	80,97	75.7
10	0	100	100	100	100
	3	98.12	98,97	98,78	98,43
	5	97.1	97,28	98,26	97.12
	8	95,76	96,42	96,81	92,92
	15	84,64	78,86	84,56	82,44
20	0	100	100	100	99,99
	3	99.22	99,72	99,69	98.13
	5	98,52	99,47	99,29	93,64
	8	97,98	98,9	99.21	97,88
	15	92.7	96.08	98,36	95,82
30	0	100	100	100	99,9
	3	99,86	99,61	99,55	85,58
	5	99,5	99,47	99,48	86,77
	8	99,66	99,27	99,34	85,86
	15	98.1	99,64	99,7	90,61

Як бачимо з результатів експерименту, для параметрів (c = 0,05, δ = 0,05) декодувати повідомлення з невеликою надлишковістю пакетів практично неможливо. У цьому випадку ми можемо декодувати тільки велику довжину вихідної послідовності і тільки з надмірністю, яка дорівнює 20% або більше. Навіть із рівнем втрати пакетів близько 8 %. І зі збільшенням швидкості втрати пакетів ми бачимо, що наша можливість декодування пакетів стає неможливою: для PLR 15% ми спостерігаємо, що лише 8% оригінальних пакетів відновлюються, тобто спадкоємність практично не підлягає відновленню. Також ми бачимо, що зі збільшенням надмірності також можливо відновити майже всю інформацію з K=500, K=1000 з рівнем PLR 5%.

Для параметрів (c = 0,2, δ = 0,05) та параметрів (c = 0,2, δ = 0,2) можна побачити , що ми відзначаємо зсув ефективності в бік малих значень K. За відсутності втрат послідовність повністю декодується для значень K=100…1000, що зумовлено достатньо великою кількістю кодових символів зі степенем 1 у кодовій послідовності. У таблиці 3.2 для K = 5000 повне відновлення може бути здійснено лише при N/K = 30%. Для розглянутих комбінацій параметрів за однакових умов пропускання та рівного об'єму надлишку ефективність відновлення незначно змінюється в залежності від значення K. Слід зазначити, що для табл. 3 для K = 5000 найгірший результат дорівнює для дорівнює PLR і N/K порівняно з K = 100 ... 1000. Очевидно, це пояснюється тим, що пік функції щільності робастного розподілу відповідає значенню аргументу (ступеня кодових пакетів d) дорівнює відношенню K / S.

Природно, що ефективність відновлення зростає зі збільшенням надлишковості кодових символів, але для таблиць 3.2 і 3.3 це не має такого вражаючого ефекту, на відміну від таблиці 3.1. Аналіз таблиць 3.2 і 3.3 показує, що кодування LT має сенс при обсяг надлишковості не менше 20 %, оскільки при менших значеннях виграш, досягнутий кодуванням, мінімальний.

Отримані експериментальні результати дозволяють зробити такі висновки:

- Використання значень c = 0,05, δ = 0,05 дає можливість досягти максимального ступеня безпеки даних, однак тільки при великих значеннях кількості вихідних пакетів K в кодованій послідовності.

- Збільшення значень c і δ дозволяє досягти більш впевненого результату при менших значеннях K.

- Мінімальна кількість надлишкової інформації, необхідної для впровадження ефективного кодування Лабі, становить близько 20%.

З огляду на те, що в основі теорії LT-кодів лежить статистична проблема м'ячів і кошиків, ефективність LT-кодів проявляється при саме великих значеннях параметра K, тобто великій кількості початкових символів [16]. Винахідники кодів LT рекомендують значення K порядку 10000 [15]. Перераховані властивості LT-кодів є дуже цінними і дозволяють використовувати LT-коди на практиці для будь-яких типів даних, що передаються. Важлива, зокрема, якість фонтану кодів для завдання «багато до багатьох». Вони досить хороші для систем, які не залежать від доставки в реальному часі. Основною проблемою LT-кодів є тривала затримка сигналу між приймачем і передавачем, якщо ми використовуємо мережі комутації пакетів, які працюють з доставкою інформації в реальному часі (IP-телефонія або IPTV), викликана необхідністю буферизації великої кількості джерел, пакети (більше K =10000) і LT-коди не забезпечують захист бітових помилок.

Для вирішення цієї проблеми «Відновлення відсутніх/втрачених пакетів і виправлення бітових помилок» були надані коди Raptor. Воні побудовані як послідовне з’єднання зовнішнього блокового коду та внутрішнього коду LT [19],[20]. Коди Raptor складніші, але кращі для використання [21],[22] в комп’ютерних мережах.

ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ

У даній роботі ми досліджували кодування контролю помилок для комутації пакетів мережі. Ми досліджуємо, що використання класичних кодів недостатньо для якісної роботи канали з втратами. Де важливо не тільки виправляти бітові помилки. Але також запобігти втратам пакетів, які можуть виникнути в мережі з комутацією пакетів. Ми можемо зробити такі висновки.

1. Якщо ми використовуємо зворотний канал і можемо повторно надіслати втрачені пакети до приймач. Тоді краще використовувати класичні коди контролю помилок. Це реалізує високий ймовірність відновлення бітових помилок у даних пакетах.

2. Ми представляємо новий вид кодування контролю помилок. Так званий фонтанний код. Це досить нове і може відкрити нові можливості для комутації пакетів.

3. Ми досліджуємо коди LT, які є першим представленням фонтану коди. І покажіть, як можна відновити втрачені пакети під час комутації пакетів мережі без використання зворотного каналу.

4. Код LT добре працює з надлишковістю пакетів понад 20%, тобто може бути не добре. Через навантаження на канал. Це може бути під час надсилання суми створення пакетів.

5. Код LT хороший для відновлення надсилання інформації, але дуже поганий для запобігання бітовим помилкам. Це може застосовуватися під час передачі інформації.

6. Хороша здатність декодування кодів LT також залежить від ступеня кодування пакети. Це забезпечується ідеальним розподілом.

7. Через погані результати коду LT у виправленні бітових помилок ми представив новий, побудований як послідовна конкатенація зовнішнього блокового коду та внутрішній код LT – коди Raptor. Вони складніші та краще працюють, ніж класичний код LT.

ПЕРЕЛІК ПОСИЛАНЬ

1. **National** Telecommunication Information Administration - Telecommunications: Glossary of Telecommunications Termspublished by Government Institutes 1 Apr 1997, 480 pages, ISBN 1461732328, Volume 1037, Part 3 of Federal Standard [Retrieved 2015-08-04]

2. **Roberts**, Dr. Lawrence G. (November 1978). "The Evolution of Packet Switching".

3. **P. Elias**, “Coding for two noisy channels,” in Proc. 3rd London Symp. Inform. Theory, London, England, 1955, pp. 61–76

4. **Charles** Wang; Dean Sklar; Diana Johnson (Winter 2001–2002). "Forward Error-Correction Coding"

5. **Daniel J.,** Costello, JR., Error Control Coding, Fundamentals and Applications, Prentice Hall, Inc. Englewood Cliffs, New Jersey, 1983

6. **W. Rao** and S. Chen, "A Rateless 16QAM Scheme for IoT Uplink Communications," in *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21712-21722, 1 Nov.1, 2022, doi: 10.1109/JIOT.2022.3183592.

7. **G. D. Forney Jr.,** “Performance of concatenated codes,” Concatenated Codes, pp. 1–6, 82–90, 1966

8. **C. Berrou,** A. Glavieux, and P. Thitimasjshima, “Near Shannon Limit ErrorCorrecting Coding and Decoding: Turbo-Codes,” in Proc., IEEE Int. Conf. On Communications, (Geneva, Switzerland, May 1993), pp. 1064-1070.

9. **E. Boutillon,** C. Douillard, and G. Montorsi, Iterative Decoding of Concatenated Convolutional Codes: Implementation Issues. Proceedings of the IEEE Custom Integrated Circuits Conference (CICC), 2007. 95(6): p. 1201 - 1227.

10. **J. Hokfelt,** O. Edfors, and T. Maseng, A survey on trellis termination alternatives for turbo codes. IEEE 49th Vehicular Technology Conference, 1999. 3: p. 2225 - 2229.

11. **J. Hokfelt,** O. Edfors, and T. Maseng, On the theory and performance of trellis termination methods for turbo codes. IEEE Journal on Selected Areas in Communications, 2001. 19(5): p. 838 - 847.

12. **Z. He, P. Fortier,** and S. Roy, Highly-Parallel Decoding Architectures for Convolutional Turbo Codes. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2006: p. 1147 - 1151.]

13. **Штомпель М. А.** Метод оптимізації фонтанних кодів на основі популяційних процедур пошукової оптимізації. Політ. Сучасні проблеми науки: тези доповідей Міжнародної науково-практичної конференції молодих учених і студентів (м. Київ, 6–8 квітня 2016 р.). Київ, 2016. – С. 126.

14. **R. Palankiand,** J.Yedidia**,**“Rateless codes on noisy channels”, in Proc. IEEE Int.Symp.Inform. Th., Chicago, IL, Jun. 2004, p. 37.

15. **M. Luby,** "LT codes," The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings., Vancouver, BC, Canada, 2002, pp. 271-280, doi: 10.1109/SFCS.2002.1181950.

16. **MacKay** D.J.C. Fountain codes // IEE Proc.Commun., 2005. – Vol.152. –№6 (December). – P. 1062–1068.,

17. **J. D. Brown**, S. Pasupathy and K. N. Plataniotis, "Adaptive demodulation using rateless erasure codes," in IEEE Transactions on Communications, vol. 54, no. 9, pp. 1574-1585, Sept. 2006, doi: 10.1109/TCOMM.2006.881236.

18. **X. Li** and Q. Huang, "Online Analog Fountain Codes," in IEEE Communications Letters, vol. 27, no. 1, pp. 50-54, Jan. 2023, doi: 10.1109/LCOMM.2022.3219600.

19. **A. Shokrollahi.** Raptor codes. IEEE Transactions on Information Theory, 52:2551–2567, June 2006.

20. **Weizheng Huang,** Huanlin Li, Jeffrey Dill, ―Fountain Codes with Message Passing and Maximum Likelihood Decoding over Erasure Channels‖, IEEE 2011.

21. **Zhu H P, Zhang** G X, Xie Z D. Suboptimal degree distribution of LT codes. Journal of Applied Sciences-Electronics and Information Engineering. Jan 2009, 27(1):6-11.

22. **Omid Etesami** and Amin Shokrollahi, Senior Member, IEEE, “Raptor Codes on Binary Memoryless Symmetric Channels”, IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 52, NO. 5, MAY 2006.