

## ОБРОБКА ІНФОРМАЦІЇ В СЕНСОРНИХ МЕРЕЖАХ

*Анотація.* В роботі наведено особливості структури, архітектури та застосування сенсорних мереж, відмінності бездротових та сенсорних мереж, проведено аналіз існуючих алгоритмів стиснення та показана їх неефективність, запропоновано використання методів стиснення для передачі.

*Ключові слова.* Бездротові сенсорні мережі, сенсори, алгоритми стиснення, методи стиснення при передачі, коефіцієнт стиснення.

### Вступ

У багатьох сферах людської життєдіяльності часто стоїть завдання збору даних, які надходять від великого числа джерел, їх обробки, і забезпечення реакції, відповідної зафіксованим подіям. Наприклад, якщо розглядати квартиру або офісне приміщення, то для створення найбільш комфортних умов необхідно контролювати вологість повітря, температуру, концентрацію кисню, освітленість і, виходячи з цих даних, регулювати потужність витяжки та обігрівача, вмикати освітлення (або регулювати його інтенсивність), включати зволожувач повітря і т.д. Якщо розглядати сільське господарство, то для досягнення хороших результатів необхідно контролювати стан вологості ґрунту, концентрації речовин, що містяться в ґрунті і т.п., і вже, виходячи з цих даних, приймати рішення про необхідність поливу, або внесення добрив.

Щоб здійснювати подібну діяльність автоматизовано, необхідно забезпечити обмін інформацією між усіма цими пристроями, тобто створити єдину мережу датчиків (чутливих елементів, сенсорів) і виконавчих пристроїв (обігрівач, зволожувач повітря, вентилятор в системі кондиціонування, поливальна машина, дозатор добрив і т.д.). Вочевидь, що провідна мережа прийнятна тільки у вузькому колі завдань в силу очевидних причин: вона була б занадто масивна, дорога, вимагала б складного монтажу, налагодження та обслуговування, непридатності для мініатюрних або мобільних об'єктів.

В силу всіх вищенаведених причин і з'явилося напромак, що набирає силу: бездротові сенсорні мережі. Бездротові сенсорні мережі можна визначити наступним чином - це масштабна, за кількістю вузлів і займаної площі, організована безліч окремих чутливих елементів і виконавчих пристроїв, об'єднаних за допомогою технології бездротового обміну інформацією, яка володіє можливостями самоорганізації, самовідновлення і здатна функціонувати автономно. Кожен ву-

Міжвідомчий науково-технічний збірник «Адаптивні системи автоматичного управління» № 1' (32) 2018  
зол в такій мережі повинен мати вкрай низьку вартість і володіти мінімально можливими розмірами.

### **1. Постановка задачі дослідження.**

Багато протоколів і алгоритмів були розроблені для традиційних бездротових мереж, тому вони не дуже добре підходять для унікальних особливостей і вимог сенсорних мереж. Основні відмінності сенсорних і бездротових мереж:

- кількість вузлів сенсорної мережі може бути на кілька порядків вище, ніж вузлів в бездротовій мережі;
- вузли щільно розташовані;
- вузли схильні до збоїв;
- топологія сенсорних мереж може часто змінюватися;
- вузли в основному використовують ширококомунікаційні повідомлення, в той час як більшість бездротових мереж засновані на зв'язку "точка-точка";
- вузли обмежені в живленні, обчислювальних потужностях, і пам'яті;
- вузли не можуть мати глобальний ідентифікаційний номер через великої кількості накладних витрат і великої кількості датчиків.

Основні проблеми сучасних бездротових сенсорних мереж:

- стиснення даних і контроль трафіку, що передається;
- вибір стратегії при обчисленнях, в яких задіяно кілька вузлів (з одного боку потрібно обмін даними, з іншого - потрібно витрати батарей на пересилку).

В роботі пропонується використання методів стиснення даних при передачі замість широко відомих алгоритмів стиснення.

### **2. Особливості сенсорних мереж**

Основні області, де сенсорні системи вкрай необхідні сьогодні:

- системи спостереження і моніторингу, охорони периметрів;
- системи домашньої і промислової автоматизації;
- модернізація та автоматизація сфери ЖКГ;
- військове застосування;
- охорону здоров'я;
- агропромисловий комплекс;
- охорона навколишнього середовища.

Це і контроль популяцій видів тварин, моніторинг лісових пожеж, охорона територій різної протяжності (зокрема, державного кордону), комплексний контроль кліматичних умов при вирощуванні сільськогосподарських культур. Особливий інтерес представляють сфери домашньої і промислової автоматизації. Зо-

крема, це системи - інтелектуальної будівлі або цифрового будинку, системи регулювання та контролю за витрачанням енергоресурсів, автоматизований контроль активів під час перевезення вантажів різних типів і т.д.

Одним з перспективних напрямків у використанні технології бездротового зв'язку є «розумний» або «цифровий» будинок -автоматизоване система, що забезпечує узгоджену роботу всіх інженерних систем будинку, грамотно розподіляє ресурси, забезпечує зрозумілий інтерфейс контролю і управління. Використання бездротових рішень зменшує термін будівництва і введення в експлуатацію будівлі, а також зменшує витрати на підтримку вже існуючої інфраструктури, економлячи кошти на прокладання кабелів і можливої перепланування будівлі.

Бездротові сенсорні мережі можуть бути невід'ємною частиною військового управління, зв'язку, розвідки, спостереження і систем орієнтування. Швидке розгортання, самоорганізації і відмовостійкість - це характеристики сенсорних мереж, які роблять їх перспективним інструментом для вирішення поставлених завдань.

Одним із застосувань в медицині є пристрої для інвалідів; моніторинг пацієнтів; діагностика; моніторинг використання медикаментів у лікарнях; збір фізіологічних даних людини; і моніторингу лікарів і пацієнтів в лікарнях.

Сьогодні існує багато ситуацій, де застосування бездротових технологій виправдано і навіть переважно: важкодоступні для прокладки проводів місця в будівлі, або ситуації, коли прокладка проводів тягне за собою підвищені витрати, зв'язок між собою декількох будівель, необхідність в мобільності. В кінцевому підсумку розглядається економічна виправданість застосування того чи іншого рішення, при задоволенні їм за всіма технічними параметрами.

Рішення бездротової автоматизації будівлі можуть використовуватися для кліматичного контролю, управління освітленням, а також пожежної та охоронної сигналізації,широкомасштабного моніторингу землі і дослідження планет, виявлення лісових пожеж, метеорологічних або геофізичних досліджень, виявлення повеней і дослідження забруднення.

До вузлів, що входять до складу сенсорних мереж пред'являються досить жорсткі і специфічні вимоги. Перш за все, це тривале функціонування від автономного джерела живлення, наявність бездротового інтерфейсу широкі мережеві можливості, включаючи самоорганізацію, динамічну аутентифікацію, гнучкі механізми маршрутизації, і т.п. Не варто забувати, що мережа, крім усього іншого, може бути вельми і вельми масштабною, нараховуючи сотні тисяч вузлів, маючи при цьому складну топологічну структуру. І при всьому цьому кожен подібний вузол повинен мати вкрай низьку вартість і володіти мінімально можливими розмірами.

**Структура сенсорних мереж.** Сенсорна мережа складається з програмного і апаратного компонента. Апаратний компонент являє собою безліч оснащених приймачами мікропроцесорних пристроїв, за допомогою яких можлива передача інформації між цими пристроями в мережі. Програмний компонент агрегує каналний, мережевий рівень і рівень додатків, що дозволяють підтримувати передачу даних від всіх вузлів мережі до деяких вузлів (базових станцій) [3].

**Архітектура сенсорних мереж.** Вузли, як правило, розташовані випадковим чином по всій території спостереження. Кожен з них може здійснювати збір даних і знає маршрут передачі даних назад в центральний вузол, кінцевому користувачеві. Дані передаються за допомогою гнучкої архітектури мережі. Центральний вузол може спілкуватися з менеджером завдань через Інтернет або супутник. Стек протоколів включають в себе інформацію про потужності і інформації про маршрути, містить дані про мережевих протоколах, допомагає ефективно спілкуватися посередством бездротової середовища, і сприяє спільній роботі вузлів[1].

**Застосування бездротових сенсорних мереж.** Сенсорні мережі можуть складатися з різних типів датчиків: сейсмічних, датчиків визначення магнітного поля, теплових, інфрачервоних, акустичних, і т.д. Використовуючи сенсорні мережі можна здійснювати найрізноманітніші вимірювання умов навколишнього середовища. Сенсори можуть використовуватися для безперервного зондування, виявлення та ідентифікації подій. Концепція мікрозондування і бездротове з'єднання відкривають багато нових областей застосування для таких мереж.

### 3. Алгоритми стиснення інформації в мережах

Існує досить багато різних алгоритмів стиснення інформації без втрат інформації. [5] Найбільш поширеними і широко відомими є:

- Алгоритм Хаффмана;
- Алгоритм арифметичного стиснення;
- Алгоритм Лемпеля-Зива.

#### **Алгоритм стиснення Хаффмена**

В основі оригінального алгоритму Хаффмана лежить наступна ідея. Символам повідомлень ставиться у відповідність код змінної довжини, що складається з цілої кількості біт. Довжина коду визначається частотою появи символу в даних що підлягають стисненню, чим рідше зустрічається символ, тим довше його код. Більш того, процедура побудови кодів гарантує однозначність їх наступного декодування завдяки властивості унікальності префікса.

Для того, щоб побудувати код символу необхідно знати таблицю частот появи символів в вхідних даних, або імовірності появи символів, із алфавіту

що має джерело повідомлень. На основі цієї таблиці будується дерево Хаффмана (H - дерево) за наступним алгоритмом:

- із символів, які зустрічаються в вхідних даних, створюють список вільних джерел. Кожен вузол характеризується вагою, під якою розуміють імовірність появи символу, чи кількість його появ в даних;
- вибираємо два вільних вузли, вага котрих мінімальна і створюємо для них батьківський вузол. При цьому вага батьківського вузла буде дорівнювати додатку ваги його дітей;
- додаємо батьківський вузол в список вільних вузлів, а його дітей, звідти, викреслюємо. Дугам, які з'єднують батьків з дітьми, ставимо у відповідність 0 і 1.
- два попередніх кроки повторюємо до тих пір, поки в списку вільних вузлів не залишиться тільки один вузол, котрий називають коренем дерева.

Тепер, знаючи дерево Хаффмана, ми можемо легко визначити код будь-якого символу. Для цього необхідно пройти шлях від кореня дерева до вузла відповідного цьому символу. Отримана послідовність нулів та одиниць і буде, представляти собою код Хаффмана.

Оригінальний алгоритм досить простий, але він має два значні недоліки. По-перше, для коректної роботи розпаковник повинен знати таблицю кодів, створену запаковником. Отже, її необхідно зберігати в архівному матеріалі, або передати декодеру іншими каналами, тим самим збільшуючи об'єм інформації, що передається. По-друге, так як нам необхідна статистика символів що містяться в початкових даних, то потрібно два проходи по ним - один для збору статистика, а другий вже для стиснення.

#### **Алгоритм стиснення за допомогою арифметичного кодування**

Арифметичне кодування є методом, який дозволяє стиснути послідовність вхідного алфавіту без втрат при умові, що відомий розподіл частот цих символів. Концепція методу була розроблена ще в роботах Еліаса в 60-х роках. В подальшому цей метод був істотно розвинутий і значно вдосконалений.

Арифметичне кодування є оптимальним, досягаючи теоретичної межі стиснення - ентропії вхідного потоку [4].

Текст, стиснутий арифметичним кодером, розглядається як деяка двійкова дріб із інтервалу  $[0, 1)$ . Результат стиснення можливо представити як послідовність двійкових цифр із цього дробу.

Ідея методу полягає в наступному: початковий текст розглядається як запис цього дробу, де кожний вхідний символ є цифрою з вагою, пропорційною імовірності його появи [6].

Нехай, алфавіт має два символи:  $a$  і  $b$  з імовірностями відповідно  $3/4$  і  $1/4$ . Алгоритм стиснення Хаффмана не може зменшити об'єм слова в даному алфавіті.

Розглянемо інтервал  $[0, 1)$ . Розіб'єм його на частини, довжина котрих пропорційна імовірності елементів. В нашому випадку це  $[0, 3/4)$  і  $[3/4, 1)$ . Суть алгоритму полягає в наступному: кожному символу в вхідному алгоритмі відповідає деякий підінтервал із  $[0, 1)$ . Пустому слову відповідає весь інтервал  $[0, 1)$ . Після отримання кожного наступного символу арифметичний кодер зменшує інтервал, вибираючи ту його частину, котра відповідає знову надійшовшому символу. Кодом ланцюжку є інтервал, виділений після всіх символів з ланцюжку, точніше кажучи, двійковий запис координати довільної точки із цього інтервалу.

Таким чином, довжина отриманого інтервалу пропорційна імовірності появи кодуємого ланцюжку.

Виконаємо наведений алгоритм для ланцюжку aaba, який представлено в таблиці 1

В якості коду можливо взяти довільне число із інтервалу, отриманого на кроці 4, наприклад 0.1.

Таблиця 1.

**Особливості роботи алгоритму арифметичного стиснення.**

Крок	Ланцюжок що підлягає розгляду	Інтервал
0	""	$[0, 1)=[0, 1)$
1	a	$[0, 3/4)=(0, 0.11)$
2	aa	$[0, 9/16)=[0, 0.1001)$
3	aab	$[27/64, 36/64)=[0.011011, 0.100100)$
4	aaba	$[108/256, 135/256)=[0.01101100, 0.10000111)$

Арифметичний декодер працює синхронно з кодером: почав з інтервалу  $[0, 1)$ , він послідовно визначає символи вхідного ланцюжка. В особливості, в нашому випадку він спочатку розділить (пропорційно частотам символів) інтервал  $[0, 1)$  на  $[0, 0.11)$  і  $[0.11, 1)$ . Так як, число 0.1 (переданий кодером код ланцюжку aaba) знаходиться в першому з них, можливо отримати перший символ a. Потім ділимо перший підінтервал  $[0, 0.11)$  на  $[0, 0.1001)$  і  $[0.1001, 0.1100)$  пропорційно частотам символів). Знову вибираємо перший, так як  $0 < 0.1 < 0.1001$ . Продовжуючи цей процес, ми однозначно декодуємо всі чотири символи. Для того, щоб декодер міг визначити кінець ланцюжка, ми можемо передавати її довжину окремо, або додати до алфавіту додатковий символ "кінець ланцюжка".

При реалізації цього методу виникають дві проблеми: по-перше, необхідна арифметика з нескінченною точністю, і по-друге, результат кодування стає відомим лише при закінченні вхідного потоку.

Крім того, арифметический кодер як і алгоритм Хаффмана, також є двухпрохідним та потребує передачі разом з закодованим текстом ще й таблиці частот символів.

### **Алгоритм стиснення LZ**

Всі розглянуті вище методи і моделі кодування розглядали в якості вхідних даних ланцюжки символів (тексту) в деякому кінцевому алфавіті. При цьому залишається відкритим питання про зв'язок цього вхідного алфавіту кодера з даними, що підлягають стисненню (представленими також у вигляді ланцюжків в (іншому) алфавіті, звичайно маючим 256 символів - біт).

В найпростішому випадку можливо використовувати в якості вхідного алфавіту кодеру саме ці символи (байти) вхідного потоку. Ступінь стиснення при цьому відносно невелика - порядку 50% для текстової інформації.

Значно більшого ступеня стиснення можливо досягти при виділенні із вхідного потоку ланцюжків що повторюються і кодування звернень на ці ланцюжки.

Алгоритм, про який буде йти мова, належить Лемпелю і Зіву, звичайно, його називають LZ77 compression (по року публікації методу стиснення).

В його основі лежить наступна концепція: запаковник постійно зберігає деяку кількість останніх оброблених символів в деякому буфері (який також має назву словника що ковзає - slidingdictionary). Назва - словник що ковзає, обумовлена тим, що довжина його постійна; кожний раз, коли компресор кодує наступний ланцюжок, він дописує його в кінець словника і "обрізає" відповідну кількість символів на початку буфера. По мірі обробки вхідного потоку знову надійшовші символи потрапляють в кінець буфера, здвигаючи попередні символи і витискаючи самі старі. Розміри цього буфера можуть бути різні, вони змінюються в різних реалізаціях. Є методи які використовують буфер розміром 4 кбайт, а є моделі маючі значно більший буфер, наприклад 32 кбайт.

Алгоритм виділяє (шляхом пошуку у словнику) самий довгий початковий підрядок вхідного потоку, співпадаючий з одним з підрядків в словнику, і видає на вихід пару (length, distance), де length - довжина знайденого в словнику підрядку, distance - відстань від нього до вхідного підрядку (тобто, фактично індекс підрядку в буфері, віднятий з його розміру). В випадку, якщо такий підрядок не знайдено, в вхідний потік просто копіюється черговий символ вхідного потоку [8].

В першій версії алгоритму пропонувалося використовувати найпростіший пошук по всьому словнику. Час стиснення, при такій реалізації, був пропорційний добутку довжини вхідного потоку на розмір буфера, що не придатно для практичного використання. Однак, в подальшому було запропоновано використовувати двійкове дерево для швидкого пошуку в словнику, що дозволили на порядок збільшити швидкість роботи.

Таким чином, алгоритм Лемпеля - Зіва перетворює один потік початкових символів в два паралельних потоки length і distance. Очевидно, що ці потоки є потоками символів в нових алфавітах  $L$  і  $D$ , і до них можливо застосувати один з розглянутих вище методів (кодування Хаффмена, арифметичне кодування). Таким чином, ми підходимо до схеми двохступінчатого кодування, найбільш ефективного з практичного використовуємих в теперішній час. При реалізації цього методу необхідно досягнути узгодженого виводу обох потоків в один. Ця проблема звичайно розв'язується шляхом почергового запису кодів символів із обох потоків.

Передача інформації в сенсорній мережі показує ряд специфічних обмежень, що накладаються на передану інформацію:

1. Одностороння спрямованість передачі даних в сенсорній мережі.
2. Відсутність статистичних та імовірнісних характеристик прийнятої і переданої інформації між сенсорами сенсорної мережі.
3. Розподілений характер переданих даних. Кожен з сенсорів є окремим і незалежним від інших сенсорів джерелом інформації. Доступ до всієї інформації в сенсорній мережі можна отримати тільки на базовій станції.
4. Аналіз, обробка і перетворення інформації, отриманої від усіх датчиків, виконується тільки на базовій станції або ж безпосередньо на крайовому пристрої за участю людини.
5. Відсутність обміну статистичними характеристиками інформації між сенсорами сенсорної мережі [2].

Наявність зазначених обмежень в значній мірі знижує ефективність роботи найбільш поширених методів стиснення інформації в сенсорній мережі. З урахуванням обмеження на відсутність втрат при обробці і передачі даних, а так само через відсутність певного типу одержуваної інформації (наприклад текст, графіка, відео і т.д.), в огляді були розглянуті універсальні і найбільш часто використовувані методи стиснення без втрат. До таких методів належать: стиснення Хаффмана, LZ стиснення і арифметичне стиснення. Крім специфічних особливостей реалізації зазначених методів, існує ще дві основні підходи при моделюванні джерела повідомлень: статичний і динамічний.

Статична модель має на увазі наявність заздалегідь відомих статистичних характеристик джерела інформації або ж можливість попереднього перегляду всієї інформації з метою отримання даних характеристик.

Динамічна модель має на увазі поступове накопичення статистичної інформації паралельно з її стисненням. При цьому метод стиснення перебудовується відповідно до знову отриманою інформацією. Розглянемо обмеження і недоліки використання в сенсорних мережах зазначених методів стиснення з урахуванням динамічного і статичного підходів при їх реалізації:



1. Реалізація статичного алгоритму Хаффмана, статичного LZ стиснення і статичного арифметичного стискання неможлива через відсутність необхідних значень ймовірності появи повідомлень для побудови кодових дерев, створення словників і кодових таблиць при реалізації зазначених методів стиснення. Крім того, наявність великої кількості марнотратів в сенсорній мережі призводить до виникнення великої кількості джерел інформації. Оскільки джерела не пов'язані між собою, то необхідно створювати кодові структури для кожного марнотратника окремо, що призводить до значного зростання апаратно-програмних витрат і складності реалізації методів стиснення.

2. Використання динамічного алгоритму Хаффмана, LZ стиснення і арифметичного стискання в сенсорних мережах є малоефективним. Динамічний підхід до стиснення інформації розрахований на досить великі обсяги інформації, що передається. Чим більше розмір переданого повідомлення - тим точніше буде побудована модель джерела і краще виконається стиснення. У сенсорних мережах джерелом інформації є датчики, підключені до сенсора. Кількість цих датчиків не велика, отже, і кількість переданої інформації є недостатнім для ефективного використання адаптивних методів стиснення. Складнощі виникають і при використанні інформації для стиснення від попередніх сенсорів. В цьому випадку, для додавання до отриманої інформації в стислому вигляді нових даних необхідно виконати відновлення прийнятої інформації з метою отримання статистичної характеристики джерела. Процес відновлення інформації в кожному з послідовності сенсорів при передачі повідомлення призводить до значного уповільнення процесу передачі інформації в сенсорній мережі.[2]

#### **4. Методи стиснення даних при передачі.**

Існуючі на цей момент методи стиснення даних в більшості випадків мають достатньо складні алгоритми, реалізація яких потребує значних обчислювальних ресурсів і суттєвих апаратних затрат при схемотехнічній реалізації. Крім того, специфічні особливості побудови і функціонування сенсорних мереж в значному ступені знижує ефективність використання найбільш розповсюджених методів стиснення.

Для стиснення інформації в сенсорних мережах з вищенаведеними характеристиками пропонується використовувати методи стиснення для передачі: лінійні, матричні, каскадні та комбіновані.

**Стиснення даних із використанням замість повторень додаткових символів.**

Ці способи ґрунтуються на заміні повторюваних елементів деякими умовними символами. Вони є ефективними в тому разі, коли масиви інформації, які

подаються у вигляді рядків або стовпців, розташованих у зростаючому порядку, мають однакові значення елементів в одних і тих самих розрядах, що характерно для інформації, що передається від сенсорів. Таке стиснення даних дає змогу скоротити масив у кілька разів.

Так, якщо елементи повторюються на початку рядків (стовпців) відносно попередніх, то замість виключених розрядів у масив уводиться знак поділу, який дає можливість відокремити елементи в згорнутому масиві. При розгортанні замість знака поділу поновлюють всі пропущені розряди, які були до елемента, що знаходився безпосередньо за знаком поділу в стисненому тексті. Запис знаків, які знаходяться після знаку поділу, виконується з кінця рядка (стовпця) [5].

Після заповнення пропущених цифр за аналогічними розрядами попереднього рядка дістаємо масив інформації, який був до стиснення.

Недоліком цього методу стиснення інформації є неможливість його застосування до впорядкованих масивів, у яких повторювані розряди зустрічаються не на початку рядків (стовпців).

Для згортання масивів, у яких в одному рядку (стовпці) є тільки одна повторювана ділянка, можна використати вище згаданий спосіб з введенням додаткового символу (кінець рядка, стовпця). При цьому розгортання масиву ведеться від одного додаткового символу до іншого. В разі фіксованої довжини рядка (стовпця) всі розряди, які знаходяться між додатковими символами, разом із пропущеними розрядами мають утворювати повний рядок (стовпець).

Поновлення масиву може виконуватися з початку або з кінця. За наявності в рядку (стовпці) кількох повторюваних ділянок вводять спеціальні символи, що вказують необхідну кількість пропусків [5].

Розгортання масивів при цьому методі стиснення виконується з їх початку або з кінця, а заповнення відповідної кількості пропусків замість додаткових символів А, В, С - перенесенням відповідної кількості символів, які знаходяться на однойменних розрядах попереднього рядка.

### **Стиснення інформації з типовими матрицями**

При цьому способі в пам'яті ПК центральної станції зберігаються матриці деяких найпоширеніших наборів символів, які зустрічаються в інформаційному масиві, що стискається. Для цього заздалегідь аналізують інформаційний масив, утворюють типові матриці і при стисненні замість наборів символів, які відповідають цій типовій матриці, в послідовність символів інформаційного масиву вводять адреси цих матриць [5].

За цим алгоритмом сканується масив вибірки вхідного потоку і після того, як знаходиться будь-яка матриця, у вихідний потік записуються: символ матриці, один з рядків в матриці, кількість рядків матриці. Нестиснуті символи вхідного по-

току записуються у вихідний інформаційний потік без будь-яких змін. Розгортання повідомлення у приймальному пристрої виконується за аналогією з описаними вище алгоритмами двох матричних способів стиснення. Як перевагу цього алгоритму можна зазначити високу швидкість розгортання і досить високий коефіцієнт стиснення, з огляду на адаптацію до кожного конкретного масиву вибірки [7].

Дамо оцінку матричних способів стиснення.

Основною умовною стиснення без втрат інформації, як відомо, є рівність кількості інформації у вхідному (первинному) і стисненому вихідному (вторинному) інформаційних потоках, тобто  $I_{BX} = I_{BUX}$ . Якщо інформаційний потік складається з множини  $N_1$  елементів,  $N_1 = \{x_1, x_2, \dots, x_i, \dots, x_{N_1}\}$ , алфавіту  $q_1$ , і з урахуванням того, що кожний елемент (символ) вхідного інформаційного потоку несе  $I(x_i) = \log_2 q_1$  біт інформації, тоді кількість інформації, яка міститься у вхідному інформаційному потоці, біт

$$I_{BX} = \sum_{i=1}^{N_1} I(x_i), \quad (1)$$

За аналогією з (1), кількість інформації, що міститься у вихідному інформаційному потоці, який складається з множини  $N_2$  елементів,  $N_2 = \{y_1, y_2, \dots, y_{N_1}\}$  алфавіту  $q_2$ ,

$$I_{BUX} = \sum_{j=1}^{N_2} I(y_j), \quad (2)$$

Для визначення коефіцієнта стиснення необхідно одержати значення швидкостей передачі інформації у вхідному і вихідному потоках. При цьому швидкість передачі інформації у вхідному потоці, біт/с,

$$R_{BX} = \frac{I_{BX}}{T_{BX}}, \quad (3)$$

де  $T_{BX}$  - час передачі інформаційного масиву вхідного потоку, який з урахуванням однотипності інформаційних елементів, визначиться як

$$T_{BX} = N_1 t_1, \quad (4)$$

де  $t_1$  - тривалість передачі одного елемента вхідного інформаційного потоку.

За аналогією з (3) і урахуванням (4) для вихідного потоку, маємо, біт/с

$$R_{BUX} = \frac{I_{BUX}}{T_{BUX}}, \quad (5)$$

де  $T_{BUX}$  - час передачі інформаційного масиву вихідного потоку [3].

Виконаємо більш детальний аналіз вихідного інформаційного потоку після його стиснення матричним способом.

Позначимо через  $I_M(y_j)$  кількість інформації, що міститься у  $j$ -й матриці і

$I(y_{j_0})$  - кількість інформації, що міститься в одиничних елементах, які не стискаються. Тоді вираз (2) буде мати вигляд, біт,

$$I_{BUX} = \sum_{j_0=1}^{N_2^{j_0}} I(y_{j_0}) + \sum_{j=1}^{N_2^j} I_M(y_j), \quad (6)$$

де  $N_2^{j_0}, N_2^j$  - число одиничних елементів (символів) та матриць різної розмірності у вихідному інформаційному потоці, відповідно.

У свою чергу, матриці можуть мати різні розміри, однак їх набір обмежений, що викликано обмеженнями, які накладаються алфавітом, що використовується для передачі інформації, та обсягом інформаційного потоку, що стискається. Таким чином, друга складова виразу (6) може бути записана як

$$\sum_{j=1}^{N_2^j} I_M(y_j) = \sum_{j_1=1}^{n_1} I_M(y_{j_1}) + \sum_{j_2=1}^{n_2} I_M(y_{j_2}) + \dots + \sum_{j_k=1}^{n_k} I_M(y_{j_k}), \quad (7)$$

де  $n_1, n_2, \dots, n_k$  - число матриць різної розмірності;  $K = \{1, 2, \dots, q\}$ , тут  $q = q_2 - q_1$ ,  $q_2$  - алфавіт, що використовується для кодування вихідного потоку,  $q_1$  - частина алфавіту, яка використовується у вихідному потоці для кодування одиничних символів (алфавіт вхідного потоку).

Отже, не дивлячись на рівність кількості інформації, яка міститься у вхідному і вихідному потоках, число елементів  $N_1$  і  $N_2$ , якими передаються ці потоки, буде різним. При цьому, як правило  $N_1 > N_2$ . Звичайно, що при матричному стисненні розмір матриць, їх правильний вибір відіграють значну роль, і чим більший буде розмір матриці, тим більшу кількість інформації вона містить [3].

З урахуванням (6) і (7) число елементів вихідного потоку

$$N_2 = N_2^{j_0} + N_2^j = N_2^{j_0} + (n_1 + n_2 + \dots + n_k), \quad (8)$$

а час передачі інформаційного масиву вихідного потоку, с

$$T_{BUX} = [N_2^{j_0} + n_1 + n_2 + \dots + n_k] t_j, \quad (9)$$

де  $t_j$  - тривалість передачі одного елемента (символу) вихідного інформаційного потоку. Якщо у (5) підставити значення  $I_{BUX}$  і  $T_{BUX}$  (6), (7) і (8), одержимо швидкість передачі інформації  $R_{BUX}$ , і, знаючи, що коефіцієнт стиснення визначається

як  $K_{CT_M} = \frac{R_{BUX}}{R_{BX}}$ , а також виходячи з рівності кількості інформації у вхідному і

вихідному інформаційних потоках, маємо

$$K_{CT_M} = \frac{I_{BUX} T_{BX}}{T_{BUX} I_{BX}} = \frac{N_1 t_i}{(N_2^{j_0} + n_1 + n_2 + \dots + n_k) t_j}, \quad (10)$$

### Каскадні методи

Застосовуються для збільшення коефіцієнта стиснення. При цьому, як правило, використовуються комбінації лінійних і матричних методів. Можна утворити такі каскадні способи стиснення, як кодо-зонний, кодо- та зонно-матричні тощо.

При використанні послідовного кодового і матричного способів стиснення отримують каскадний кодо-матричний метод.

В даному випадку вхідний інформаційний потік підлягає кодовому стисненню, а потім - матричному. При цьому коефіцієнт стиснення буде дорівнювати добутку:

$$K_{cm_{км}} = K_{cm_k} * K_{cm_m} = \frac{N_1 * \log_2 q_2}{(N_2^{j_0} + n_1 + n_2 + \dots + n_k) * \log_2 q_1}$$

де  $N_1$  - загальна кількість елементів (символів) вхідного інформаційного потоку;  $N_2$  - число одиночних елементів (символів) в вхідному інформаційному потоці;  $n_1, n_2, \dots, n_k$  - число матриць різної розмірності;  $k = \{1, 2, \dots, q\}$ , тут  $q = q_2 - q_1$ ;  $q_2$  - алфавіт, що використовується для кодування вихідного (вторинного) інформаційного потоку;  $q_1$  - частина алфавіту  $q_2$ , що використовується у вторинному потоці для кодування одиночних символів (а також, алфавіт вхідного інформаційного потоку) [4].

### Каскадно-комбіновані методи стиснення

Кодо-поздовжньо-матричне і кодо-поперечно-матричне стиснення відносяться до каскадно-комбінованих способів стиснення, так як такі способи передбачають на першому етапі кодове стиснення, а на другому комбіноване - поздовжньо-матричне (або поперечно-матричне) стиснення. Коефіцієнт стиснення визначається добутком коефіцієнтів стиснення і поздовжнього-матричного (поперечно-матричного), тобто:

$$K_{cm_{кпм}} = K_{cm_k} K_{cm_{пм}}, \quad (11)$$

Або з урахуванням (11) отримаємо вираз:

$$K_{cm_{кпм}} = \frac{N_1 \log q_2}{(N_2^{j_0} + n_n + n_1 + n_2 + \dots + n_k) * \log_2 q_1}, \quad (12)$$

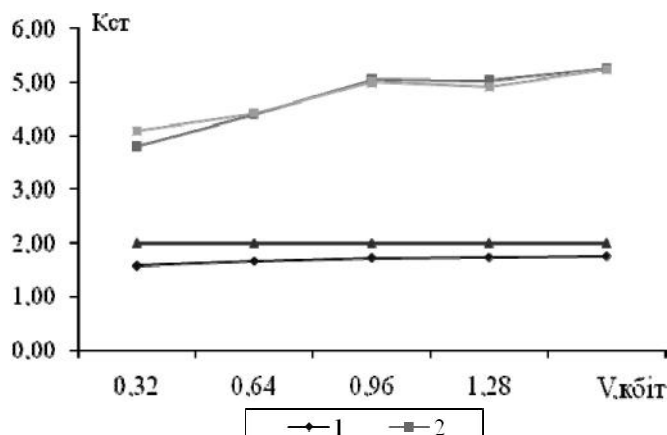
де  $N_1$  - загальна кількість елементів (символів) вхідного інформаційного потоку;  $N_2^{j_0}$  - число одиночних елементів (символів) в вхідному інформаційному потоці;  $k = \{1, 2, \dots, q\}$ , тут  $q = q_2 - q_1$ ;  $q_2$  - алфавіт, що використовується для кодування вихідного (вторинного) інформаційного потоку;  $q_1$  - частина алфавіту  $q_2$ , що використовується в вихідному інформаційному потоці для кодування одиночних символів;  $n_n$  - число матриць розмірності  $1 \times m_2$  (при кодо-поздовжньому-матричному стисненні), або розмірності  $m_1 \times 1$  (при кодо- поперечно-матричному стисненні) [3].

Для того, щоб вибрати найкращий спосіб стиснення скористаємося відомими залежностями коефіцієнта стиснення  $K_{ст}$  від довжини інформаційного масиву при різних алфавітах  $q$ . Представимо ці залежності у вигляді таблиць та графіків при  $q = 4$  та  $q = 8$  [3].

Таблиця 2.

**Залежність коефіцієнта стиснення  $K_{ст}$  від довжини інформаційного масиву при  $q = 4$ .**

Метод стиснення	$K_{ст}$ при інформаційному масиві, кбіт				
	0.32	0.64	0.96	1.28	1.6
1. Лінійний з $\rho$	1.58	1.66	1.72	1.73	1.75
2. Лінійний з $a, b, \dots, c$	3.81	4.41	5.05	5.04	5.26
3. Кодовий	2	2	2	2	2
4. Матричний	4.10	4.42	5.0	4.92	5.23



**Рис. 1.** Графік залежності  $K_{ст}$  від довжини інформаційного масиву при алфавіті  $q = 4$ .

Таблиця 2.

**Залежність коефіцієнта стиснення  $K_{ст}$  від довжини інформаційного масиву при  $q=8$ .**

Метод стиснення	$K_{ст}$ при інформаційному масиві, кбіт				
	0.32	0.64	0.96	1.28	1.6
1. Лінійний з $\rho$	1.58	1.66	1.72	1.73	1.75
2. Лінійний з $a, b, \dots, c$	3.90	6.10	7.56	8.53	9.30
3. Кодовий	3.00	3.00	3.00	3.00	3.00

Закінчення табл. 2

Метод стиснення	$K_{ст}$ при інформаційному масиві, кбіт				
	0.32	0.64	0.96	1.28	1.6
4. Матричний	4.10	6.21	8.07	9.62	10.53
5.Кодо-матричний	7.60	11.85	12.63	14.06	14.68
6.Лінійно-матричний	4.10	6.40	8.65	9.77	10.50
7.Кодо-лінійно-матричний	7.70	12.30	14.12	16.62	16.80

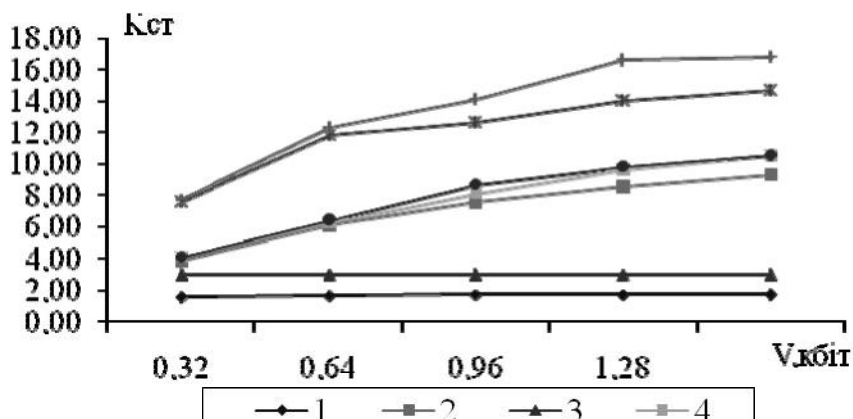


Рис. 2. Графік залежності  $K_{ст}$  від довжини інформаційного масиву при алфавіті  $q=8$ .

З таблиць та графіків можна зробити висновок, що для алфавіту  $q=4$  найкращими способами стиснення є лінійний з  $a, b, \dots, c$  та матричний, а для алфавіту  $q=8$  - кодо-матричний та кодо-лінійно-матричний способи, тобто каскадні та комбіновані способи стиснення.

### Висновки

Розглянуті методи стиснення при передачі мають деякі переваги на відміну від існуючих методів при використанні їх в сенсорних мережах:

1. Запропоновані методи стиснення можуть ефективно працювати з невеликими по обсягу даними, на відміну від існуючих методів.
2. Запропоновані методи стиснення перетворюють двійкові послідовності в числа, що дозволяє проводити обробку вихідних даних в стиснутому вигляді.
3. Крім того, випадкова помилкова зміна символу не є критичним для прийому всього потоку даних. Можливо використання завдостійкого кодування разом з запропонованими методами стиснення для запобігання виникнення помилок при передачі.

### Список використаних джерел

1. Akyildiz I.F., Su W., Sankarasubramaniam Y., Cayirci E. Wireless sensor networks: a survey // *Computer Networks*, Volume 38, 2002. – pp. 393–422.
2. Костель С. В. Рева В. В. Биномиальное сжатие информации в сенсорных сетях // *Науково-технічна конференція "Фізика, електроніка, електротехніка (ФЕЕ-2012)"* – Суми: СумДУ, 2012. - С.154
3. Жураковський Б. Ю. Матричні та комбіновані способи стиснення даних при передачі / Жураковський Б. Ю. // *Наукові записки Українського науково-дослідного інституту зв'язку № 2(26)*, 2013. – С.28-32.
4. Жураковський Б. Ю., Жураковський Я. Ю. Каскадне стиснення інформації під час обробки в автоматизованих системах управління // *Зв'язок №2*, 2001 р. – стр.44-46.
5. Жураковський Ю. П. Полторак В. П. Теорія інформації та кодування./ Жураковський Ю. П. Полторак В. П – К.: «Вища школа», 2001. – 256 с.
6. Кохманюк Д. Сжатие данных: как это делается. / Кохманюк Д – *IndexPro*, 1992, №1, с.18-29; 1993, №2, с.30-49.
7. Жураковський Б.Ю. Матричне стиснення при передаванні даних / Герасимчук Д.В., Жураковський Ю.П., Жураковський Б.Ю. // *Зв'язок №5(25)*, 2000. – С.58-59
8. Кричевский Р. Е. Сжатие и поиск информации./ Кричевский Р. Е. — М.: Радио и связь, 1989. —168с.