DOI: https://doi.org/10.36910/6775-2524-0560-2025-61-09
УДК 004.72
Бортник Катерина Яківна[1], к.т.н., доцент
https://orcid.org/0000-0001-5282-099X
Кардашук Володимир Сергійович[2], к.т.н., доцент
https://orcid.org/0000-0002-7940-6753
Чепіль Микола Анатолійович[1], студент
[1] Луцький національний технічний університет, м. Луцьк, Україна
[2] Східноукраїнський національний університет імені Володимира Даля, м. Сєвєродонецьк, Україна

МЕТОД ПОБІТОВОГО ПРЕДСТАВЛЕННЯ ЛОГІЧНИХ СТАНІВ ДЛЯ ЗМЕНШЕННЯ ОБСЯГУ ДАНИХ У СИСТЕМАХ IOT

Бортник К. Я., Кардашук В. С., Чепіль М. А. Метод побітового представлення логічних станів для зменшення обсягу даних у системах IOT. У статті представлено підхід до ефективного кодування логічних (булевих) станів у системах Інтернету речей (IoT) шляхом використання побітового представлення цілого числа. Запропонований метод ґрунтується на ідеї об'єднання множини незалежних булевих значень у єдину цілочисельну змінну типу ulong, що дозволяє зберігати до 64 логічних станів у межах одного цілого числа. Такий спосіб кодування істотно скорочує обсяг переданої та збереженої інформації, що є критично важливим для пристроїв з обмеженими обчислювальними ресурсами, обмеженою пропускною здатністю каналів зв'язку та підвищеними вимогами до енергоефективності. У роботі наведено програмну реалізацію алгоритмів пакування та розпакування даних на мові C#, продемонстровано особливості використання побітових операцій, а також описано принципи відновлення початкових булевих масивів. Проведено аналітичне порівняння побітового кодування зі стандартним байтовим представленням, яке передбачає зберігання кожного логічного значення у вигляді окремого байта, а також з іншими поширеними методами представлення булевих даних, наприклад представлення у вигляді списку строк. Результати дослідження свідчать про можливість зменшення обсягу даних на 87–88 %, що позитивно впливає на тривалість роботи автономних пристроїв та пришвидшує передачу даних у мережах із низькою швидкістю. Запропонований метод рекомендовано застосовувати у вбудованих IoT-рішеннях, зокрема у системах безпеки та моніторингу. Спеціальну увагу приділено можливості використання такого підходу у «розумних сейфах» та інших проектах, де необхідно компактно передавати стани замка, сенсорів та тривожних сигналів з мінімальним навантаженням на мережу.
Ключові слова: побітове кодування, логічні стани, IoT, оптимізація передачі даних, вбудовані системи, енергоефективність.

Bortnyk K., Kardashuk V., Chepil M. Method of bitwise representation of logical states for data size reduction in iot systems. The article presents an approach to efficient encoding of logical (boolean) states in Internet of Things (IoT) systems by using bitwise representation of an integer value. The proposed method is based on the idea of combining multiple independent boolean values into a single integer variable of type ulong, which allows storing up to 64 logical states within one integer. This encoding method significantly reduces the volume of transmitted and stored data, which is critically important for devices with limited computational resources, constrained communication bandwidth, and increased requirements for energy efficiency. The paper provides a software implementation of packing and unpacking algorithms in C#, demonstrating the use of bitwise operations, as well as describing the principles of reconstructing the original boolean arrays. An analytical comparison between bitwise encoding and standard byte-based representation—where each logical value is stored as a separate byte—is conducted, along with a comparison to other common approaches to representing boolean data, such as lists of string values. The results indicate that bitwise encoding enables reducing data volume by 87–88%, which positively affects the autonomy of battery-powered devices and accelerates data transmission in low-bandwidth networks. The proposed method is recommended for use in embedded IoT systems, particularly in security and monitoring applications. Special attention is given to its applicability in "smart safe" solutions and other projects where compact transmission of lock states, sensor readings, and alarm signals is required while maintaining minimal network load.
Keywords: bitwise encoding, logical states, IoT, data transmission optimization, embedded systems, energy efficiency.

Постановка наукової проблеми. Сучасні вбудовані системи, зокрема пристрої безпеки, дедалі частіше базуються на технологіях Інтернету речей (IoT), що передбачає постійний обмін даними між сенсорами, контролерами та серверними додатками. Одним із ключових викликів у таких системах є ефективність передачі інформації за умов обмеженої пропускної здатності каналів зв'язку, малих обсягів пам'яті мікроконтролерів і потреби у швидкодії в реальному часі.

У пристроях безпеки – таких як «розумні» сейфи, сигналізації чи системи контролю доступу – часто передаються логічні (булеві) стани, що відображають прості події: «замкнено/відкрито», «рух виявлено/не виявлено», «тривога/норма» тощо. На практиці такі стани можуть кодуватися по-різному – як числові, логічні або навіть текстові значення. Найменш ефективним є передавання текстових позначень ("true", "false", "opened", "closed"), що вимагає 4-7 байтів для кожного елемента, тоді як для передачі одного логічного стану достатньо лише одного біта. Навіть коли булеві значення передаються у вигляді чисел (1 / 0) або байтів типу bool, мережеве навантаження

залишається високим, особливо якщо йдеться про передачу масивів станів декількох сенсорів або виконавчих елементів. У системах із бездротовим зв'язком (Wi-Fi, Bluetooth Low Energy) така надлишковість призводить до зайвого енергоспоживання, затримок передачі та нестабільності комунікаційного каналу.

Аналіз останніх досліджень і публікацій. Дослідження в галузі оптимізації передавання даних для IoT-пристроїв показують, що зменшення обсягу переданої інформації є одним із найефективніших способів скорочення енергоспоживання та підвищення швидкодії комунікацій [1-3]. У низці праць підкреслюється доцільність використання побітового або двійкового кодування для ущільнення логічних даних без втрати точності, що є особливо актуальним для сенсорних мереж та систем із великим обсягом булевих сигналів [4].

Ідея полягає в тому, що кожен біт байта відповідає одному логічному елементу, що дає змогу упакувати до восьми станів у межах одного байта. Це забезпечує скорочення обсягу переданої інформації в порівнянні зі звичайним байтовим представленням.

Виділення невирішених раніше частин загальної проблеми. Попри активний розвиток технологій передавання даних та енергоефективних протоколів у сфері IoT питання оптимізації структури переданих повідомлень залишається актуальним. Більшість пристроїв використовують надлишкові формати даних, коли булеві стани передаються у вигляді текстових або байтових змінних, що збільшує обсяг трафіку, затримки та споживання енергії. Недостатньо дослідженою залишається можливість зменшення обсягу переданої інформації за рахунок побітового подання логічних даних без втрати точності.

Мета дослідження: Основною метою дослідження є розробка та експериментальна перевірка методу побітового представлення логічних станів, який дозволяє зменшити обсяг даних, що передаються між IoT-пристроями, підвищити швидкодію обміну та знизити енергоспоживання.

Завдання дослідження. Завдання дослідження полягає у проведенні аналізу сучасних підходів до оптимізації передавання даних у вбудованих системах IoT, розробленні алгоритму побітового пакування та розпакування булевих значень засобами мови C#, а також у здійсненні аналітичного та експериментального порівняння ефективності побітового представлення з традиційним байтовим способом. Крім того, дослідження передбачає оцінку можливостей практичного застосування запропонованого методу у ресурсно обмежених IoT-пристроях для підвищення ефективності обміну даними та зниження енергоспоживання.

Основна частина дослідження. Передавання інформації у вбудованих системах безпеки передбачає часту комунікацію між мікроконтролером, сенсорами та зовнішнім сервером або мобільним застосунком. Більшість таких пристроїв оперують дискретними станами, що можуть приймати лише два значення – «так» або «ні», «увімкнено» або «вимкнено», «замкнено» або «відкрито». Такі змінні називаються булевими, і для їхнього зберігання теоретично достатньо одного біта інформації.

Попри те, що булеві змінні містять лише 1 біт інформації, у практичних реалізаціях вони часто зберігаються у вигляді окремих байтів або навіть текстових значень. Тобто, тип bool або uint8_t у мові програмування C, C++, Python, C# зазвичай займає 1 байт (8 бітів), а якщо ж система передає значення у форматі тексту ("true", "false", "open", "closed"), то кожне слово потребує від 4 до 7 байтів, не враховуючи службових символів JSON або XML.

Таким чином, при передачі масиву з 8 булевих станів витрачається 8 байтів у випадку типу bool та від 32 до 56 байтів у випадку текстових значень.

У той час, як теоретично достатньо 1 байта, у якому кожен біт може представляти один логічний стан. Це означає, що звичайне представлення є у 8-50 разів менш ефективним, ніж мінімально можливе.

Згідно з теорією інформації, одиниця даних, що має лише два можливих стани, містить один біт ентропії. Отже, якщо маємо n булевих значень $b_0, b_1, \dots, b_{\{n-1\}}$, то кожне можна записати як окремий розряд у двійковому числі:

$$N = \sum_{i=0}^{n-1} b_i \cdot 2^i, \tag{1}$$

де N – це десяткове представлення всього масиву булевих змінних. Наприклад, для послідовності $1, 0, 1, 1, 0, 0, 1, 0$ отримаємо:

$$N = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 0 \cdot 2^5 + 1 \cdot 2^6 + 0 \cdot 2^7 = 77. \tag{2}$$

Число 77 у двійковому вигляді (01001101) фактично містить усі 8 початкових логічних значень, упакованих у 1 байт.

Для усунення цієї надлишкових витрат пам'яті при передачі одного булевого значення у байті – було розроблено метод, який перетворює набір булевих значень у ціле число, де кожен біт відповідає одному елементу. Таким чином, до 64 логічних змінних можуть бути записані в межах одного числа типу ulong. Коли значення дорівнює true, відповідний біт встановлюється у 1; коли false – у 0, приклад реалізації зображений на рисунку 1.

```
public static ulong PackBools(IReadOnlyList<bool> values){
  if (values == null || values.Count == 0)
    return 0;
  if (values.Count > 64)
    throw new ArgumentException("Максимум 64 булевих значення для пакування у 64 біти.");
  ulong result = 0;
  for (int i = 0; i < values.Count; i++) {
    if (values[i])
      result |= 1UL << i; // встановлюємо біт i, якщо значення true
  }
  return result;
}
```

Рис. 1. Приклад реалізації методу упакування даних

Зворотна операція – розпакування – здійснюється зчитуванням кожного біта числа і відтворенням початкового списку логічних значень продемонстрована на рисунку 2.

```
public static List<bool> UnpackBools(ulong packed, int count){
  if (count < 0 || count > 64)
    throw new ArgumentOutOfRangeException(nameof(count), "count має бути у межах 0–64.");
  var result = new List<bool>(count);
  for (int i = 0; i < count; i++) {
    bool bit = ((packed >> i) & 1UL) != 0;
    result.Add(bit);
  }
  return result;
}
```

Рис. 2. Приклад реалізації методу розпакування даних

Нехай n – кількість булевих значень, які необхідно передати. У стандартному варіанті кожен елемент займає один байт, тому:

$$B1 = n \tag{3}$$

У побітовому представленні, де кожен байт містить вісім логічних станів:

$$B2 = \left\lceil \frac{n}{8} \right\rceil \tag{4}$$

Відносна економія обсягу даних обчислюється за формулою:

$$E = 1 - \frac{B2}{B1} \tag{5}$$

Для прикладу, якщо n = 24, то у звичайному представленні потрібно 24 байти, а після побітового кодування – лише 3 байти, що дає 87,5 % економії.

Для перевірки коректності реалізації було проведено тестування з довільним набором булевих значень, представлених у прикладі на рисунку 3.

```
bool[] values = { true, false, true, true, false, true, false, true, true, false };

ulong packed = PackBools(values);

List<bool> unpacked = UnpackBools(packed, values.Length);


Console.WriteLine($"Початкові значення: {string.Join(", ", values)}");

Console.WriteLine($"Паковане число: {packed}");

Console.WriteLine($"Розпаковані значення: {string.Join(", ", unpacked)}");
```

Рис. 3. Приклад програми для перевірки цілісності даних після упакування та розпакування

Результат виконання програми:

```
Початкові значення: True, False, True, True, False, True, False, True, True, Fals

Паковане число: 469

Розпаковані значення: True, False, True, True, False, True, False, True, True, False
```

Рис. 4. Приклад виконання програми для перевірки цілісності даних після упакування та розпакування

Як видно на рисунку 4, усі логічні значення успішно відновлюються, що підтверджує коректність роботи алгоритму.

Також варто згадати один з найпоширеніших форматів передачі даних – JSON. Він забезпечує хорошу читабельність, зручність інтеграції з REST API, підтримується практично всіма мовами програмування та мережевими протоколами. Однак формат JSON є суттєво надлишковим для передачі простих логічних даних, оскільки кожне булеве значення кодується текстовим представленням "true" або "false", що займає значно більше місця, ніж один біт. Крім цього, JSON містить структуру керуючих символів – квадратні дужки, коми, лапки – які ще більше збільшують загальний розмір повідомлення.

Для ілюстрації недоліків JSON розглянемо масив із 64 логічних станів. У текстовому представленні "true" складається з 4 байтів, а "false" – з 5. У середньому, використання текстових булевих значень займає близько 4,5 байта на один елемент. Додатково додається 63 коми між елементами та пара символів [і]. Таким чином, розмір повідомлення дорівнює приблизно 64 × 4,5 = 288 байтів лише булевих значень, крім того присутні 63 байти (коми) + 2 байти (дужки) = 65 байтів, що в результаті становить близько 353 байтів. У деяких випадках цей обсяг може бути ще більшим, якщо логічні значення передаються у вигляді рядків "true"/"false" з лапками, або якщо повідомлення формується відладочними інструментами, що додають пробіли та розриви рядків. У будь-якому випадку, JSON є одним із найменш ефективних форматів для передачі булевих масивів у ресурсно обмежених системах, оскільки фактичний інформаційний вміст – один біт на стан – збільшується у сотні разів.

Ще один формат, що часто використовується для компактного та транспортабельного представлення даних, – Base64. На відміну від JSON, Base64 не додає структурних символів для опису масиву, а перетворює байтовий набір у текстовий вигляд, без зміни структури даних. Проте важливо зазначити, що Base64 не стискає інформацію, а лише перекодовує її у формат, який може безпечно передаватися через текстові канали (HTTP, WebSocket, MQTT тощо). Це означає, що розмір даних після кодування Base64 збільшується приблизно на 33 % через механізм перетворення 3 байтів у 4 символи ASCII. У випадку передачі булевих даних у класичному байтовому вигляді (1 логічний стан = 1 байт) масив із 64 елементів займає 64 байти. Після кодування у Base64 його розмір визначається за формулою:

$$B_{base64} = 4 \cdot \left\lceil \frac{n}{3} \right\rceil \quad (6)$$

Порахуємо для 64 байтів:

$$B_{base64} = 4 \cdot \left\lceil \frac{64}{3} \right\rceil = 4 \cdot 22 = 88 \text{ байтів} \quad (7)$$

Таким чином, масив логічних станів у вигляді Base64 займає 88 байтів, що значно менше за JSON, але все одно в 11 разів більше за обсяг, який потрібен для представлення тих самих даних у вигляді простого бітового набору (8 байтів). Base64 є прийнятним у ситуаціях, коли необхідно передавати дані у форматі, сумісному з текстовими протоколами, однак цей метод не призначений для оптимізації обсягу інформації.

Порівняння двох форматів показує, що JSON є найбільш надлишковим способом передачі булевих масивів через текстову природу представлення та наявність допоміжних символів. Base64 є компактнішим, але все ж не дозволяє досягти мінімального можливого розміру даних, оскільки оперує байтовими одиницями, а не бітами. У той час як JSON може збільшувати обсяг даних у понад 40–50 разів, Base64 – у 1,33 раза, а побітове представлення – у десятки разів ефективніше за обоє.

Висновки та перспективи подальшого дослідження. Результати проведеного дослідження підтверджують ефективність побітового представлення логічних станів для зменшення обсягу переданої інформації у системах IoT. Застосування такого методу дозволяє скоротити кількість переданих байтів у середньому на 85-88 %, що позитивно впливає на швидкодію комунікацій і знижує енергоспоживання пристроїв із обмеженими ресурсами. Реалізований алгоритм пакування та розпакування булевих даних у мові C# є простим у впровадженні та може бути інтегрований у будь-які вбудовані системи, де необхідна передача великої кількості логічних станів. Перспективи подальших досліджень полягають у розширенні методу для роботи з багаторівневими логічними сигналами, поєднанні побітового кодування з методами стиснення та шифрування даних, а також у тестуванні запропонованого підходу на реальних бездротових протоколах зв'язку, таких як LoRaWAN, Zigbee чи MQTT, з метою підвищення ефективності обміну даними у масштабних IoT-мережах.

Список бібліографічного опису:

1. Меккі К., Баджич Е., Шаксел Ф., Мейєр Ф. Порівняльне дослідження технологій LPWAN для масштабних розгортань IoT. *ICT Express (Вираження ІКТ).* 2019. Т. 5, № 1. С. 1–7. URL: https://doi.org/10.1016/j.icte.2017.12.005 (дата звернення: 01.11.2025).

2. Перейра Ф., Родрігес Ж., Пінто А., Соліч П. Проблеми енергоефективності та комунікацій у ресурсно-обмежених IoT-пристроях: ключові фактори успішного впровадження. *Sensors (Датчики).* 2020. Т. 20, № 22. С. 6420. URL: https://doi.org/10.3390/s20226420 (дата звернення: 01.11.2025).

3. Занаї Е., Дака А., Бербері П., Кулла Е., Ю С. Огляд енергоефективності короткодіапазонних і широкодіапазонних технологій IoT. *Technologies (Технології).* 2021. Т. 9, № 1. С. 22. URL: https://doi.org/10.3390/technologies9010022 (дата звернення: 01.11.2025).

4. Алмудейні З., Аль-Весабі Ф. Н., Алькатірі А., Альшамрані А., Альзахрані А. Д. Енергонеефективність мереж IoT: причини, наслідки та стратегічна модель сталого вдосконалення. *Electronics (Електроніка).* 2025. Т. 14, № 1. С. 159. URL: https://doi.org/10.3390/electronics14010159 (дата звернення: 01.11.2025).

References:

1. Mekki K., Bajic E., Chaxel F., Meyer F. A comparative study of LPWAN technologies for large-scale IoT deployment. ICT Express. 2019. Vol. 5, No. 1. P. 1–7. Available at: https://doi.org/10.1016/j.icte.2017.12.005 (accessed: 01.11.2025).

2. Pereira F., Rodrigues J., Pinto A., Solic P. Challenges in Resource-Constrained IoT Devices: Energy and Communication as Critical Success Factors for Future IoT Deployment. Sensors. 2020. Vol. 20, No. 22. P. 6420. Available at: https://doi.org/10.3390/s20226420 (accessed: 01.11.2025).

3. Zanaj E., Daka A., Berberi P., Kulla E., Yoo S. Energy Efficiency in Short and Wide-Area IoT Technologies – A Survey. Technologies. 2021. Vol. 9, No. 1. P. 22. Available at: https://doi.org/10.3390/technologies9010022 (accessed: 01.11.2025).

4. Almudayni Z., Al-Wesabi F. N., Alkathiri A., Alshamrani A., Alzahrani A. J. Energy Inefficiency in IoT Networks: Causes, Impact, and a Strategic Framework for Sustainable Optimisation. Electronics. 2025. Vol. 14, No. 1. P. 159. Available at: https://doi.org/10.3390/electronics14010159 (accessed: 01.11.2025).