

**Покришка С.А.**

## **УДОСКОНАЛЕННЯ ТА ПІДВИЩЕННЯ БЕЗПЕКИ UDP ПРОТОКОЛУ**

*У статті розглянуто питання підвищення безпеки та перевірка відправки повідомлень через мережу. Також розглянуто спосіб шифрування повідомлень, а саме CRC (циклічний надлишковий код). Розглянуто як це працює на прикладі простого набору символів та як це впровадити в додаток.*

**Ключові слова:** UDP, TCP/IP, протоколи, CRC.

**Вступ.** Зв'язок відіграє ключову роль в більшості сучасних додатків. Особливо в інтернеті. Велика кількість систем використовують UDP (User Datagram Protocol) [1] в якості основного протоколу. Протокол UDP є ненадійним протоколом без встановлення з'єднання. Є необхідність покращити отримку повідомлень, методом виявлення помилок і методом корекції.

**Постановка проблеми.** Набір протоколів TCP-IP[2] був розроблений по моделі OSI. Він складається з чотирьох рівнів. Прикладний рівень відповідає за надання послуг користувачеві. Багато суб-протоколів використовуються на рівні додатків, таких як Hyper Text Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), сервер доменних імен (DNS), (ICQ) та інші. Один з цих протоколів використовуються для надання послуг чату. Найостанніші програми для спілкування використовують протокол TCP або безпосередньо, або за допомогою протоколу HTTP. TCP має певний недолік такий як труднощі у використанні, повільність і низьку безпеку. Для подолання цих недоліків використовується протокол UDP, який швидше, ніж протокол TCP і простіший. Протокол UDP є ненадійним (немає гарантій про отримання відправлених даних), тому існує необхідність використовувати один з методів виявлення помилок.

User Datagram Protocol (UDP) є більш простим з двох стандартних протоколів TCP-IP транспорту. Крім того, це швидше і легше, ніж TCP. UDP не створює віртуальний канал, як TCP, а також не вимагає підтвердження прийому. Він просто посилає повідомлення. Заголовки UDP всі по 8 байт, в той час як заголовки TCP можуть бути 20-60 байт довжиною. Подібно TCP, UDP забезпечує доставку сегментів з використанням IP.

**Мета роботи.** Для збільшення безпеки та удосконалення системи відправки повідомлень в мережі потрібно додати контрольну суму та циклічні коди.

**Основний текст.** Основні User Datagram Protocol (UDP) шари: рівень прикладної програми, транспортний рівень, інтернет рівень та фізичний рівень даних.

UDP є ненадійним і для того, щоб подолати це обмеження, необхідно використовувати метод виявлення помилок. Є два ефективних методи: контрольна сума і контроль циклічного надлишкового коду (CRC) [3] (рис. 1).

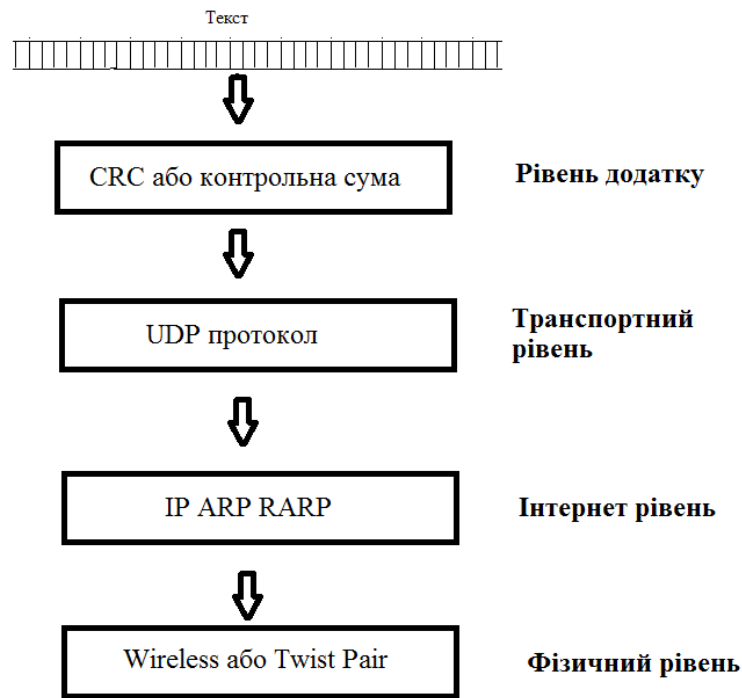


Рис. 1. Структура протоколу

User Datagram Protocol (UDP) використовує просту модель передачі без неявних рукописних для забезпечення надійності, упорядкування, або цілісності даних. Таким чином, UDP забезпечує ненадійну службу, і датаграми можуть виходити з ладу, дублюватися, або зникати без попереднього повідомлення. UDP передбачає, що перевірка помилок і корекція або не потрібна або виконана в заявці, уникаючи такої обробки на рівні мережевого інтерфейсу. Якщо корекція помилок необхідна на рівні мережевого інтерфейсу, додаток може використовувати протокол управління передачею (TCP) або протокол управління передачею потоку (SCTP), які призначені для цієї мети.

UDP-х також підходить для серверів, які відповідають на невеликі запити від величезного числа клієнтів. На відміну від TCP, UDP сумісний з широкомовним пакетом (відправлення всім по локальній мережі) і мультикастингом (опублікувати всім абонентам) [4].

Додатки, що використовують UDP: система доменних імен (DNS), потокове мультимедіа, таке як IPTV, передача голосу по IP (VoIP), тривіальний протокол передачі файлу (TFTP).

UDP не надає ніяких гарантій для протоколів верхнього рівня щодо доставки повідомлень. З цієї причини, UDP іноді називають ненадійний протокол дейтаграм.

UDP забезпечує застосування мультиплексування (через номери портів) і перевірку цілісності (за допомогою контрольної суми) в заголовку. Якщо потрібна надійність, вона має бути реалізована на рівні додатку.

Заголовок UDP складається з 4-х полів. Використання двох з них є необов'язковим (в IPv4 - порт джерела та контрольна сума). В IPv6 тільки порт джерела є необов'язковим.

Порт джерела ідентифікує порт відправки. Порт призначення - це поле ідентифікує порт призначення. Довжина - це 16-бітове поле, яке визначає довжину в байтах всієї дейтаграми: заголовка і даних. Мінімальна довжина становить 8 байт, оскільки це довжина заголовка. Контрольна сума - це 16-розрядне поле, яке використовується для перевірки помилок заголовка і даних.

Управління Winsock поставляється з Visual Basic 6.0 (VB6) і використовується для створення додатків, які мають доступ до низькорівневих функцій передачі управління протоколом Internet Protocol (TCP/IP).

TCP/IP є специфікація, яка визначає ряд протоколів, використовуваних для стандартизації, як комп'ютери обмінюються інформацією один з одним. TCP/IP забезпечує зв'язок через взаємопов'язані мережі, що використовують різні апаратні архітектури та різні операційні системи. Протоколи в TCP/IP розташовані у вигляді ряду шарів, відомих як стек протоколів. Кожен шар має свою власну функціональність.

Winsock є стандартом, який підтримується корпорацією Майкрософт. Цей стандарт в основному є набором процедур, який описує зв'язок із стека TCP/IP. Ці процедури знаходяться в бібліотеці динамічного компонування (DLL), яка працює під Windows. Winsock DLL сполучається з TCP/IP, а звідти через Інтернет.

Порт є спеціальне місце в пам'яті, яке існує, коли два комп'ютера знаходяться в системі зв'язку через TCP/IP. Додатки використовують номер порту в якості ідентифікатора до інших комп'ютерів. І відправник, і комп'ютери використовують цей порт для обміну даними.

Winsock вище стека протоколу TCP/IP в моделі ISO/OSI. TCP/IP є стандартним протоколом зв'язку, який визначає методи для упаковки даних в пакети для передачі між обчислювальними пристроями в мережі. TCP/IP є стандартом для передачі даних по мережах, включаючи Інтернет. TCP встановлює з'єднання для передачі даних, а IP визначає спосіб передачі пакетів даних [5].

Транспортний рівень (також відомий як транспортний рівень хост-хост) несе відповідальність за забезпечення прикладного рівня з сесією і датаграмою послуг зв'язку. Основні протоколи шару транспорту TCP і протокол призначені для користувача дейтаграм (UDP). Управління Winsock підтримує наступні два режими роботи: `sktTCPProtocol` та `sktUDPProtocol`.

CRCs засновані на теорії про циклічні коди корекції. Використовуються систематичні циклічні коди, які кодують повідомлення, додаючи значення перевірки фіксованої довжини, з метою виявлення помилок в мережах зв'язку. Вперше метод був запропонований В. Уеслі Петерсоном в 1961 році. Циклічні коди не тільки прості в реалізації, але й добре підходять для виявлення помилок.

Для обчислення N-біт двійкового CRC вирівнюють біти, що представляють вхід в ряд, і положення  $(n + 1)$ -бітний шаблон, який представляє дільник CRC під лівий кінець ряду.

Повідомлення, що буде закодовано, спочатку доповнюється нулями відповідної бітової довжини  $n$  CRC. Вхід зліва зсувається на 3 біти.

Множинна контрольна сума є простим методом. Для цього використовують рівняння модуль XOR 2 між послідовними бітами повідомлення.

Наприклад, повідомлення = [ABCD]

$g = (A + B + D)$  по модулю 2

$S = (A + B + C)$  по модулю 2

$t = (B + C + D)$  по модулю 2

Код = [rsatbcd]

Властивості контрольної суми: невелика кількість надлишкових бітів, простота впровадження, не дуже надійний результат.

Код помилки повинен бути обчислений за допомогою двох методів CRC, а також кількох контрольних сум, і ці коди повинні бути відправлені на іншу сторону через певний номер порту. Після цього текст надсилається в протокол нижнього рівня (Transport Protocol). Це може бути зроблено за допомогою управління WinSock. За допомогою керуючого входом повідомлення WinSock може бути надіслано повідомлення. В іншій стороні приймач отримує повідомлення, і обчислює код CRC для кожного повідомлення. Якщо CRC код невірний, приймач просить відправника послати текстовий блок знову. Також є ще один метод для перевірки отриманого повідомлення, контрольної суми. Ця методика була використана з деякими текстовими блоками в тому випадку, коли відправник використовує метод контрольної суми.

**Висновки.** Мета роботи полягає в удосконаленні протоколу для більшої безпеки. Завдяки UDP протоколу повідомлення надходять швидко, але немає ніяких гарантій, що повідомлення буде отримане приймачем. Для рішення цієї проблеми в роботі були використані методи виявлення помилки та циклічний надлишковий код (CRC). Також в роботі використовується контрольна сума. У випадку, коли повідомлення прийшло з помилкою, або не все, приймач може запросити відправку ще раз.

### Література

1. UDP протокол [Електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/UDP>
2. TCP протокол [Електронний ресурс]. – Режим доступу: [https://ru.wikipedia.org/wiki/Transmission\\_Control\\_Protocol](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol)
3. CRC (циклічний надлишковий код) [Електронний ресурс]. – Режим доступу: [https://ru.wikipedia.org/wiki/Циклический\\_избыточный\\_код](https://ru.wikipedia.org/wiki/Циклический_избыточный_код)
4. Будущее интернет - протоколов [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/post/345472/>
5. Надстройка протокола TCP [Електронний ресурс]. – Режим доступу: <http://samag.ru/archive/article/3812>

### References

1. UDP protocol [Electronic resource]. - Access mode: <https://ru.wikipedia.org/wiki/UDP>
2. TCP protocol [Electronic resource]. - Access mode: [https://ru.wikipedia.org/wiki/Transmission\\_Control\\_Protocol](https://ru.wikipedia.org/wiki/Transmission_Control_Protocol)
3. CRC [Electronic resource]. - Access mode: [https://ru.wikipedia.org/wiki/Циклический\\_избыточный\\_код](https://ru.wikipedia.org/wiki/Циклический_избыточный_код)
4. The Future of Internet Protocols [Electronic resource]. - Access mode: <https://habr.com/ru/post/345472/>
5. TCP add-in [Electronic resource]. - Access mode: <http://samag.ru/archive/article/3812>

В статье были рассмотрены методы проверки ошибок при отправке сообщений через сеть. Так же рассмотрен способ шифрования сообщений, а именно CRC (циклический избыточный код). Рассмотрено как это работает на примере и как это внедрить в приложение.

**Ключевые слова:** UDP, TCP/IP, протоколы, CRC.

The article examined methods for checking errors when sending messages over a network. The method of message encryption, namely CRC (cyclic redundancy code), is also considered. It is examined how this works with an example and how to implement it in an application.

**Keywords:** UDP, TCP / IP, protocols, CRC.

### **Відомості про автора**

Покришка С.А. – магістр групи КН-18дм кафедри комп'ютерних наук та інженерії Східноукраїнського національного університету імені Володимира Даля, e-mail: [hakermans@gmail.com](mailto:hakermans@gmail.com)