DOI: https://doi.org/10.36910/6775-2524-0560-2021-42-27
УДК 681.3.01
[1]**Мороз Сергій Анатолійович,** доцент
https://orcid.org/0000-0003-4677-5170,
[2]**Гузюк Георгій Олександрович,** учень
[1] Луцький національний технічний університет,
[2] Волинський науковий ліцей-інтернат.

# АНАЛІЗ ПРОТОКОЛІВ ПЕРЕДАЧІ ІНФОРМАЦІЇ ДЛЯ ТЕХНОЛОГІЇ ДИСТАНЦІЙНОГО УПРАВЛІННЯ ЗАСОБАМИ МЕРЕЖІ ІНТЕРНЕТ

**Мороз С.А., Гузюк Г.О. Аналіз протоколів передачі інформації для технології дистанційного управління засобами мережі Інтернет.** В статті розглянуті технології дистанційного управління різноманітним обладнанням за допомогою засобів мережі Інтернет. Зокрема проаналізовані різні протоколи передачі даних (інформації), які використовуються для технології управління, виявлено їх переваги та недоліки. Визначено, що одним з найбільш раціональним є протокол передачі даних MQTT.
**Ключові слова:** дистанційне управління, протоколи передачі даних, протокол MQTT, обмін повідомленнями.

**Мороз С.А., Гузюк Г.О. Анализ протоколов передачи информации для технологии дистанционного управления средствами сети Интернет.** В статье рассмотрены технологии дистанционного управления разнообразным оборудованием с помощью средств сети Интернет. В частности проанализированы различные протоколы передачи данных (информации), которые используются для технологии управления, выявлены их преимущества и недостатки. Определено, что одним из наиболее рациональным является протокол передачи данных MQTT.
**Ключевые слова:** дистанционное управление, протоколы передачи данных, протокол MQTT, обмен сообщениями.

**Moroz S., Huziuk H. Analysis of information transfer protocols for the technology of remote control by means of the Internet.** The article discusses technologies for remote control of various equipment using the Internet. In particular, various protocols of data (information) transmission that are used for control technology are analyzed, their advantages and disadvantages are revealed. It has been determined that one of the most rational is the MQTT data transfer protocol.
**Keywords:** remote control, data transfer protocols, MQTT protocol, messaging.

**Постановка наукової проблеми.**

Ідея дистанційного керування і моніторингу різноманітного обладнання та систем через мережу Інтернет є важливою науково-технічною проблемою. Вирішенню цієї проблеми сприяє бурхливий розвиток електроніки, а саме поява нових датчиків та способів передачі керуючої інформації.

Можливість в будь-який момент часу отримати або відправити потрібну керуючу інформацію, не залежно від свого місця розташування, дає широкі можливості. Саме тому актуальною є проблема вибору раціонального протоколу передачі даних, який би забезпечував достатню швидкість передачі даних, не навантажував інтернет-канал, був зручним у використанні та запобігав втратам даних. Крім того, важливим чинником, який впливає на вибір протоколу передачі даних є тип обладнання, яким необхідно керувати.

Дистанційне керування реалізується різними способами:

1) Передача сигналу по проводах - використовується там, де немає можливості застосувати бездротову передачу (наприклад, через відсутність прямої видимості, наявності екранування, міркувань секретності тощо) або з міркувань вартості і перешкодозахищеності. Такий канал використовується, головним чином, для управління системами мобільних об'єктів, обладнанням виробничих об'єктів, лабораторій, або спеціальних об'єктів (військового і іншого призначення);

2) Радіо-дистанційне управління - використовується, головним чином, для управління рухомими об'єктами - радіокерованими спортивними моделями і іграшками, обладнанням для надзвичайних ситуацій (роботи і т. д.), безпілотними літальними апаратами (БПЛА), військовими мобільними об'єктами; або в ситуаціях, коли передавач і приймач не можуть перебувати в зоні прямої видимості (системи освітлення або опалення, підйомники гаражних дверей тощо);

3) Ультразвуковий канал - використовується рідко, для управління мобільними і стаціонарними об'єктами на порівняно невеликій відстані;

4) Інфрачервоний канал - використовується, як правило, для побутової електроніки. Інфрачервоне випромінювання сильно розсіюється, тому дальність дії обладнання, що використовує такий спосіб передачі даних обмежена кількома десятками метрів.

Радіо-дистанційне управління (RF-дистанційне керування) використовується для управління віддаленими об'єктами за допомогою різних радіосигналів, що передаються пристроєм дистанційного

керування. Більшість пультів дистанційного керування використовують власне кодування, передаючи від 8 до 100 або більше імпульсів, фіксований або змінний код.

Реалізація бездротових технологій дозволяє створювати мобільні системи. Дистанційне управління зазвичай складається з двох частин: передачі і прийому. Передавач розділений на дві частини: радіочастотний пульт дистанційного керування і модуль передавача При цьому мобільним може блок на стороні оператора і (або) на стороні об'єкта управління. Це дозволяє використовувати модуль передавача в якості компонента в більшому додатку.

Для забезпечення великої дальності потрібні значні потужності радіопередавачів. Без дозволу можна працювати на малих потужностях в певних діапазонах частот.

Як різновид радіоканального варто відзначити WiFi обладнання, яке застосовується для створення IoT-обладнання, що застосовуються в системах "Розумний будинок".

**Аналіз останніх досліджень.**

Мережевим протоколом називають пакет правил і приписів, що здійснює обмін інформацією між кількома пристроями, які є включені в кабельну або бездротову мережу. Найпоширенішою класифікацією мережевих протоколів, є OSI (Open System Interconnection) [1] - модель Взаємодії Відкритих Систем. Відповідно до неї, протоколи поділяються на 7 рівнів за призначенням. Від фізичного рівня, що відповідає за створення і визначення сигналів, до прикладного, який призначений для передачі інформації додатками (API). Найпоширенішим протоколом є протокол TCP / IP. Це протокол нижнього рівня, що по суті є платформою зв'язку в Internet. TCP або Transmission Control Protocol - розбиває передані дані на частини і нумерує їх. IP або ж Internet Protocol передає всі частини одержувачу. Потім, за допомогою TCP, виконується перевірка, чи всі компоненти отримані. При отриманні всіх частин, протокол TCP розподіляє їх в необхідному порядку і монтує в єдине ціле.

Розглянемо детальніше протоколи які часто використовуються глобальній мережі.

« HTTP - широко поширений протокол передачі даних, спочатку призначений для передачі гіпертекстових документів (тобто документів, які можуть містити посилання, що дозволяють організувати перехід до інших документів). Абревіатура HTTP розшифровується як HyperText Transfer Protocol, «протокол передачі гіпертексту». Відповідно до специфікації OSI, HTTP є протоколом прикладного (верхнього, 7-го) рівня. Актуальна на даний момент версія протоколу, HTTP 1.1, описана в специфікації RFC 2616. Протокол HTTP припускає використання клієнт-серверної структури передачі даних. Клієнтську програму формує запит і відправляє його на сервер, після чого серверне програмне забезпечення обробляє цей запит, формує відповідь і передає його назад клієнтові. Після цього клієнтську програму може продовжити відправляти інші запити, які будуть оброблені аналогічним чином. Завдання, яке традиційно вирішується за допомогою протоколу HTTP - обмін даними між призначеним для користувача додатком, що здійснює доступ до веб-ресурсів (зазвичай це веб-браузер) і веб-сервером. На даний момент саме завдяки протоколу HTTP забезпечується робота Всесвітньої павутини » [2]. Недоліком даного протоколу є висока затримка сигналу та складність створення зворотних запитів.

Для обміну даними багато сучасних дронів, що збираються користувачами особисто, або комерційні і навіть промислові пристрої, використовують протокол MAVLink. MAVLink (англ. Micro Air Vehicle Link) — протокол створений для комунікації безпілотного апарату із наземною станцією. MAVLink розроблений у вигляді бібліотеки маршалінгу повідомлень, яка складається виключно з файлів-заголовків. MAVLink був уперше випущений 2009 року Лоренцом Майєром. « Протокол описує інформаційну взаємодію між системами, такими як MAV і GCS (Ground control station) - станція наземного управління, а також їх складовими частинами - компонентами. Базовою сутністю MAVLink є пакет, який має такий вигляд: Перший байт пакету (STX) - це символ початку повідомлення: 0xFD для версії v2.0, 0xFE для версії v1.0, 0x55 для версії v0.9. LEN - довжина корисного навантаження (повідомлення). SEQ - містить лічильник пакета (0-255), який допоможе нам виявити втрату повідомлення. SYS (System ID) - ідентифікатор системи, яка відправляє повідомлення, а COMP (Component ID) - ідентифікатор компонента, який відправляє повідомлення. MSG (Message ID) - тип повідомлення, від нього залежить, які дані будуть лежати в корисного навантаження пакета. PAYLOAD - корисне навантаження пакета, повідомлення, розміром від 0 до 255 байт. Два останніх байта пакету - CKA і CKB, нижній і верхній байт, відповідно, містять контрольну суму пакета. Бібліотека MAVLink дозволяє кодувати і розкодувати пакети згідно з протоколом, але вона не регламентує, якими апаратними та програмними засобами дані будуть відправлені - це можуть бути TCP / UDP повідомлення, обмін через послідовний порт або інший спосіб, який забезпечує

двосторонній обмін. Бібліотека обробляє вхідні дані побайтово, додаючи їх в буфер і сама збирає з них пакет. Кожна система або компонент, може одночасно обмінюватися даними з різних джерел, тоді для кожного джерела призначається спеціальний ідентифікатор, який називають channel (канал). MAVLink містить буфер на кожний канал » [3]. Недоліком даного протоколу є повна відсутність підтримки його розробниками

UDP (англ. User Datagram Protocol) - один з мережевих протоколів для Інтернету в стеку TCP/IP. « Відмінності даного протоколу з протоколом передачі даних TCP в тому, що він працює без встановлення з'єднання. UDP — один з найпростіших протоколів транспортного рівня OSI, який обмінюється повідомленнями, не гарантуючи їхньої доставки. Використовуючи UDP, відповідальність за обробку помилок і повторної передачі даних лежить на протоколі рівнем вище. Проте UDP є доцільним та ефективним для серверів, котрі надсилають невеликі за обсягом відповіді великій кількості клієнтів. Назва UDP-конверту містить у собі чотири поля, з яких два є опціональними. «Порт відправника» і «контрольна сума» — це 16-бітні поля, які ідентифікують відправляючий та отримуючий процеси. «Порт відправника» є необов'язковим, через те що, UDP працює без встановлення з'єднання та відправник даних може не потребувати відповіді. В такій ситуації «порт відправника» мусить бути рівним нулю. Номери портів зазвичай є зарезервованими за службами » [4].

Поле «Розмір» є обов'язковим, бо воно визначає довжину усього пакету в байтах, включно із полем «Дані». Мінімальне значення даного поля рівне восьми байтам.

Остаточне поле теми UDP-конверту довжиною 16 біт складається з контрольної суми теми і поля даних. «Контрольна сума» також є необов'язковим полем, але на практиці його використовується майже у всіх випадках.

Програми, котрі використовують UDP у ролі транспортного протоколу, повинні бути готові до помилок та втрат деяких даних і їх повторної передачі. Певні програми, наприклад TFTP, використовують додаткові програмні механізми для підвищення надійності передачі. Проте у більшості випадків для таких програм надійність не є вимогою і може завадити уповільненням зв'язку. Прикладами програм, які часто використовують UDP є потокове відео та VoIP (голос поверх IP). Коли ж програма потребує досить високого рівня надійності, то вона може використовувати протокол TCP або код, надлишковість якого допоможе знаходити помилки при передачі даних.

Через те, що у UDP відсутні будь-які контрольні механізми запобігання перенавантаженням, мережеві механізми мусять мати засоби для зменшення ефекту потенційних перевантажень від завеликого, неконтрольованого потоку UDP-трафіку. Тобто, через те, що UDP-відправники не можуть виявляти перевантаженість, єдиним інструментом для призупинки надмірного UDP-трафіку залишаються такі мережеві елементи, як роутери, які використовують «черги конвертів» та «відкидання конвертів». DCCP (Datagram Congestion Control Protocol, протокол, який використовується для контролю навантаженості датаграм) створений як часткове вирішення цієї проблеми, він може контролювати навантаження на кінцевих вузлах високошвидкісних потоків UDP-трафіку, наприклад, потокового відео або VoIP. Недоліками даного протоколу в тому, що оскільки послідовність і підтвердження під час передачі даних відсутні, UDP вважається ненадійною і небезпечною. Пошкоджені пакети видаляються, але не запитуються для повторної передачі, після того як вони загублені.

WebSocket — протокол, який призначений для обміну інформацією між веб-сервером та браузером в режимі реального часу. Він убезпечує двонаправлений повнодуплексний канал зв'язку через один TCP-сокет[5]. WebSocket створено для застосування у веб-серверах та веб-браузерах, але може також використовуватись будь-яким застосунком, який працює за клієнт-серверною архітектурою. Прикладний програмний інтерфейс WebSocket був стандартизований W3C, крім того протокол WebSocket стандартизований IETF як RFC 6455. Для встановлення WebSocket-з'єднання, клієнт мусить надіслати handshake-запит — так званий запит для встановлення довіри, своєрідне, «цифрове рукостискання». Клієнт також надсилає свій відкритий ключ Sec-WebSocket-Key для шифрування його повідомлень. Відкритий ключ в розділі параметрів HTTP-запиту кодується в форматі base64. У разі встановлення з'єднання, клієнтові надходить відповідь надіслана сервером. Сервер, через коректне заповнення параметра Sec-WebSocket-Accept надає підтвердження, що він дійсно має право встановлювати WebSocket-з'єднання. Недоліками даного протоколу є те, що він є несумісніший з деякими браузерами, немає захисту від проблем зі з'єднанням, гарантії доставки всіх повідомлень в потрібному порядку та потреба постійно тримати з'єднання відкритим.

**Виклад основного матеріалу.**

В результаті аналізу інформаційних джерел для технології дистанційного управління різним обладнанням раціональним є протокол передачі даних MQTT.

MQTT (Message Queue Telemetry Transport) — це спрощений мережевий протокол, який працює на прикладному рівні поверх TCP / IP. Використовується для обміну повідомленнями між пристроями за принципом публікач – підписник – публікач, клієнт, який передає повідомлення, називається видавцем (publisher); клієнт, який отримує повідомлення - підписником (subscriber) (рис. 1.). У центрі знаходиться MQTT-брокер – центральний вузол MQTT, що забезпечує взаємодію клієнтів. Обмін даними між клієнтами відбувається тільки через брокера. В якості брокера може виступати серверне програмне забезпечення або контролер. В обов'язки його функціонування входить отримання даних від клієнтів, обробка і збереження даних, доставка даних клієнтам, і контроль за доставкою повідомлень.

MQTT-клієнти не мають безпосереднього зв'язку один з одним, і не взаємодіють безпосередньо. Брокер може отримувати дані з різних джерел, проводити над ними маніпуляції, наприклад, розраховувати середнє значення від декількох датчиків, і вже оброблені дані повертати підписнику.

При цьому, асинхронність протоколу MQTT передбачає, що підписник та публікач можуть бути онлайн в різний час, втрачати пакети, і бути недоступні. Брокер подбає про те, щоб зберегти в пам'яті останні дані, отримані від публікача, і забезпечити їх доставку до підписника.

Пристрої MQTT використовують певні типи повідомлень для взаємодії з брокером, зокрема:

- Connect - встановити з'єднання з брокером;
- Disconnect - розірвати з'єднання з брокером;
- Publish - опублікувати дані в топік на брокера;
- Subscribe - підписатися на топік на брокера;
- Unsubscribe - відписатися від топіка.


Рис .1. Схема взаємодії між підписником, публікачем і брокером

База даних брокера окремо містить таблицю з усіма отриманими пакетами з індексацією по топіках цих пакетів, котрі мають можливість хешуватись.[6] Отримавши пакет, брокер надсилає його усім клієнтам в мережі згідно їх підписці. Щоб пристрій щось отримав від брокера він мусить бути підписаним на топік. Топіки складаються з UTF8-символів, і мають деревоподібну структуру, схожу на файлову систему в UNIX. Це зручний механізм, що дозволяє представити дані у зрозумілому вигляді. Топіки виникають динамічно за фактом підписника або за фактом приходу пакету з даними топіком. Таким чином топіки є зручним механізмом організації зв'язків різних видів: один до багатьох, багато до одного і багато до багатьох. Якщо у пакета немає підписників, він відкидається. В разі, коли підписника немає на зв'язку, то пакет або відразу стирається в базі брокера, або чекає на підключення абонентів вказаний в конфігурації час. Варіант поведінки визначається атрибутом QoS пакету. MQTT не залежить від формату даних. Корисне навантаження (payload) може містити будь-який тип даних, тому і видавці, і підписники повинні розуміти і узгоджувати формат даних. У корисне навантаження можна надсилати текстові повідомлення, дані зображення, звукові дані, зашифровані дані, двійкові дані, об'єкти JSON або практично будь-яку іншу структуру. Однак текстові і двійкові дані JSON є найбільш поширеними типами даних корисного навантаження.

Для контролю доступу в MQTT передбачена аутентифікація клієнтів, на відміну від протоколу Modbus, який не має такої функції. Для контролю доступу використовуються такі поля: ClientId - (обов'язкове поле) унікальний ідентифікатор клієнта. Повинен бути унікальним для кожного клієнта. Поточна версія стандарту MQTT 3.1.1 дозволяє використовувати порожнє поле ClientId, якщо не потрібно збереження стану підключення. Username - (опціональне поле) логін для аутентифікації, в форматі UTF-8. Може бути не унікальним. Наприклад, група клієнтів може авторизуватися з одним і тим же логіном / паролем. Password - (опціональне поле) може надсилатися тільки разом з полем Username, при цьому Username може передаватися без поля Password. Максимум 65535 байт. Важливо

знати, що ім'я та пароль передаються у відкритому вигляді, тому, якщо дані передаються по публічних мереж, необхідно використовувати SSL для шифрування підключення.

Гнучкість протоколу MQTT дозволяє клієнту передавати дані, заздалегідь не визначені на брокера. Тобто немає необхідності попередньо створювати потрібні топіки, в які зможе записати дані Publisher. Використовуючи дані, отримані з особистого кабінету можливо вручну скласти запит для публікації даних в необхідний топік та читання з нього.

Максимально допустимий розмір пакета в MQTT становить 256 Мб, що дозволяє отримати надзвичайно велику кількість даних. Наприклад, IBM Watson може обробляти дані розміром до 128 Кб, а Google підтримує 256 Кб. З іншого погляду, опубліковане повідомлення може включати корисне навантаження нульової довжини. Поле корисного навантаження є не обов'язковим. Необхідно звірити відповідність розмірів корисного навантаження з обраним хмарним провайдером. Недотримання цієї вимоги може призвести до помилок і відмови у доступі до хмарного брокера.

Для забезпечення безпеки в протоколі MQTT реалізовані наступні методи захисту:

- аутентифікація клієнтів. Пакет CONNECT може містити в собі поля USERNAME і PASSWORD. При реалізації брокера можна використовувати ці поля для аутентифікації клієнта;
- контроль доступу клієнтів через Client ID;
- підключення до брокера через TLS / SSL.

На даний момент MQTT є ефективним і одним із найбільш популярних протоколів передачі даних між окремими пристроями в рамках систем IoT. Він має декілька переваг відносно інших протоколів передачі даних[7]:

- працює за NAT (Network Address Translation) - клієнти можуть перебувати за NAT, тільки сервер (брокер) повинен мати реальний IP. Дозволяє не використовувати VPN і прокидання портів;
- динамічна конфігурація - не вимагає попередньо узгодження полів і форматів даних, може конфігуроватися «при потребі».;
- підтримує QoS - можливість управляти пріоритетом повідомлень і гарантувати доставку повідомлення адресату.;
- мізерне споживання трафіку;
- асинхронний - дозволяє обслуговувати велику кількість пристроїв, і не залежить від мережевих затримок;
- повна відсутність втрат даних;
- зручна адресація - поля даних мають текстові назви, зрозумілі для користувача. Не потрібно запам'ятовувати цифрові адреси та бітові зсуви.

Всі ці переваги дають можливість управління та моніторингу в режимі реального часу. Але MQTT потребує наявності свого особистого сервера, який відіграє роль посередника між усіма клієнтами мережі. Тут є два варіанти: використовувати сторонні сервіси по підписці, або створювати свій сервер. Існує багато хмарних провайдерів, які надають послуги MQTT-брокера, наприклад Microsoft Azure IoT Hub, Amazon AWS IoT, Cloudmqtt.com і інші.

Описана система управління побудована з двох основних частин: сервера MQTT і клієнтів, яких може бути безліч.

**Висновки.**

MQTT - сучасний протокол, позбавлений багатьох недоліків попередників. Його гнучкість дозволяє додавати клієнтські пристрої без налаштувань брокера, що суттєво економить час. Поріг входження для розуміння і настройки протоколу досить низький, а наявність бібліотек для великої кількості мов програмування дозволяє вибрати будь-який стек технологій для розробки. Гарантія доставки повідомлень істотно відрізняє MQTT від його попередників, і дозволяє не витрачати час на зайву розробку власних механізмів контролю цілісності на мережевому рівні.

**Список бібліографічного опису**

1. Мережева модель OSI [Електронний ресурс] // Вільна енциклопедія «Вікіпедія» – Режим доступу до ресурсу: URL: https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%80%D0%B5%D0%B6%D0%B5%D0%B2%D0%B0_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI.
2. Простым языком об HTTP [Електронний ресурс] // Информационная система «Хабр» – Режим доступу до ресурсу: URL: https://habr.com/ru/post/215117/.
3. MAVLink [Електронний ресурс] // Вільна енциклопедія «Вікіпедія» – Режим доступу до ресурсу: URL: https://uk.wikipedia.org/wiki/MAVLink.
4. UDP [Електронний ресурс] // Свободная энциклопедия «Википедия» – Режим доступу до ресурсу: URL:

https://ru.wikipedia.org/wiki/UDP.
5. WebSocket [Електронний ресурс] // Вільна енциклопедія «Вікіпедія» – Режим доступу до ресурсу: URL: https://clck.ru/SmteF
6. Как общаются машины — протокол MQTT [Електронний ресурс] // Информационная система «Хабр» – Режим доступу до ресурсу: URL: https://bit.ly/39rIDSB.
7. Климаш М. М. Технології безпровідного зв'язку / М. М. Климаш, В. О. Пелішок, П. М. Михайленич. – Львів: НВЕД УАД, 2007. – 818 с.
8. Погорілий С. Д. Комп'ютерні мережі. Апаратні засоби та протоколи передачі даних / С. Д. Погорілий, Д. М. Калита. – Київ: ВПЦ "Київський ун-т", 2007. – 455 с.

**References**

1. OSI Network Model [Electronic resource] // Free encyclopedia "Wikipedia" - Resource access mode: URL: https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%80%D0 % B5% D0% B6% D0% B5% D0% B2% D0% B0_% D0% BC% D0% BE% D0% B4% D0% B5% D0% BB% D1% 8C_OSI.
2. In simple language about HTTP [Electronic resource] // Information system "Habr" - Access mode to the resource: URL: https://habr.com/ru/post/215117/.
3. MAVLink [Electronic resource] // Free encyclopedia "Wikipedia" - Mode of access to the resource: URL: https://uk.wikipedia.org/wiki/MAVLink.
4. UDP [Electronic resource] // Free encyclopedia "Wikipedia" - Access mode to the resource: URL: https://ru.wikipedia.org/wiki/UDP
5. WebSocket [Electronic resource] // Free encyclopedia "Wikipedia" - Resource access mode: URL: https://clck.ru/SmteF.
6. How machines communicate - MQTT protocol [Electronic resource] // Information system "Habr" - Mode of access to the resource: URL: https://bit.ly/39rIDSB.
7. Klimash M.M. Wireless communication technologies / M.M. Klimash, V.O. Pelishok, P.M. Mykhaylenych. - Lviv: NVED UAD, 2007. - 818 p.
8. Pogoriliy SD Computer networks. Hardware and data transmission protocols / SD Pogoriliy, DM Kalita. - Kyiv: VPTs "Kyiv University", 2007. - 455 p.