МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТА КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ В СУЧАСНОМУ ЦИФРОВОМУ СУСПІЛЬСТВІ

КОЛЕКТИВНА МОНОГРАФІЯ

ХЕРСОН, 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТА КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ В СУЧАСНОМУ ЦИФРОВОМУ СУСПІЛЬСТВІ

Колективна монографія
за загальною редакцією

кандидата технічних наук, доцента

Райко Галини Олександрівни

Херсон, 2020

УДК 004:316.324.8:316.776
В 43

Рекомендовано до друку
Вченою Радою Херсонського національного технічного університету
(протокол №2 від 24.09.2020)

Рецензенти:

Тригуба А.М. д.т.н., професор, завідувач кафедри інформаційних систем та технологій Львівського національного аграрного університету

Бараненко Р.В. к.т.н., доцент, професор кафедри професійних та спеціальних дисциплін Херсонського факультету Одеського державного університету внутрішніх справ Міністерства внутрішніх справ України

Авторський колектив: Ходаков В.Є., Кругла Н.А., Веселовська Г.В., Романюк О.Н., Катєльніков Д.І., Шмалюх В.А., Пирог М.В., Борсук О.С., Райко Г.О., Чебукін Ю.В., Федорова М.С., Данилець Є.В., Сидорук М.В., Козел В.М., Дроздова Є.А., Іванчук О.В., Лєпа Є.В., Хапов Д.В., Ігнатенко І.П., Карамушка М.В., Григорова А.А.

В 43 **Використання інформаційних та комунікаційних технологій в сучасному цифровому суспільстві** : колективна монографія / За загальною редакцією Райко Г.О. – Херсон: Книжкове видавництво ФОП Вишемирський В.С., 2020. – 148 с.

ISBN 978-617-7783-91-5

Колективна монографія присвячена застосуванню інформатичних та комунікаційних технологій в сучасному цифровому суспільстві.

Колективна монографія розрахована на фахівців у галузі економіки, інформаційних технологій, фінансів, банківництва, державного управління, науковців, викладачів, аспірантів, магістрів та студентів.

Матеріали монографії представлено у авторській редакції.

УДК 004:316.324.8:316.776

ISBN 978-617-7783-91-5

© Херсонський національний технічний університет, 2020
© ФОП Вишемирський В.С., 2020

ЗМІСТ

1.	Khodakov V.Ye., Krugla N.A., Veselovskaya G.V.	The Research In Approaches To Improving The Efficiency Of Information Technologies As Components Of Computer Systems To Support The Teaching A Course On Basics Of Internet Technologies And The Web Design On The Basis Of Concepts Application In Expert Systems Of The Artificial Intelligence	5
2.	Романюк О.Н., Катєльніков Д.І., Шмалюх В.А.	Комп'ютерне моделювання протезів	19
3.	Пирог М.В., Борсук О.С.	Кроссплатформний застосунок для об’єднання співвласників багатоквартирних будинків в умовах інформатизації суспільства	32
4.	Райко Г.О., Чебукін Ю.В., Федорова М.С.	Моделювання складних систем в задачах нечіткого управління	43
5.	Данилець Є.В.	Використання адаптивного дизайну в сучасному сайтобудівництві	66
6.	Сидорук М.В.	Автоматизована система оцінки інтелектуального потенціалу працівника	78

7.	Козел В.М., Дроздова Є.А., Іванчук О.В.	Керування системою інтернету речей за допомогою зчитування електроміограми з вибором оптимального протоколу обміну даними	86
8.	Лєпа Є.В.	Класифікація клієнтів банку	98
9.	Хапов Д.В., Ігнатенко І.П.	Оцінка сталого розвитку регіонів за допомогою інформаційної системи	107
10.	Карамушка М.В.	Оцінка ефективності інноваційного управління підприємством на базі інформаційних систем	118
11.	Григорова А.А.	Інформаційна технологія аналізу туристичного потенціалу території	131
	АНОТАЦІЇ		141

РОЗДІЛ 7

КЕРУВАННЯ СИСТЕМОЮ ІНТЕРНЕТУ РЕЧЕЙ ЗА ДОПОМОГОЮ ЗЧИТУВАННЯ ЕЛЕКТРОМІОГРАМИ З ВИБОРОМ ОПТИМАЛЬНОГО ПРОТОКОЛУ ОБМІНУ ДАНИМИ

Козел В.М.
к.т.н., доцент, доцент кафедри інформаційних технологій,
Херсонський національний технічний університет

Дроздова Є.А.
старший викладач кафедри інформаційних технологій,
Херсонський національний технічний університет

Іванчук О.В.
студент кафедри інформаційних технологій,
Херсонський національний технічний університет

Постановка проблеми. Стрімкий розвиток Інтернету речей (Internet of Things, IoT) призвів до появи безлічі прикладних протоколів, необхідних для його реалізації.

Архітектура Інтернету речей передбачає наявність таких функціональних рівнів: мережа датчиків, шлюз, управління, додаток [2]. Оскільки нижній рівень складається з датчиків і сенсорів, то одразу ж виникає необхідність в "особливих" протоколах для забезпечення взаємодії цих пристроїв з сервером. Стандартні прикладні протоколи не можуть бути використані через їх непристосованість до умов мережі Інтернету речей. Датчик, зазвичай мініатюрний, з невеликою пам'яттю, вимірює фізичні параметри в режимі реального часу, найчастіше в умовах низького енергозабезпечення. Результати вимірювань обробляються вузлом і передаються на сервер. Обсяг інформації, що формується одним вузлом, порівняно невеликий, проте більшість сервісів Інтернету речей побудовано на принципі обробки інформації від безлічі вузлів, що принципово відрізняється від архітектур, прийнятих в класичних мережах, типу абонент - вузол зв'язку для телефонії, клієнт-сервер для передачі даних .

Таким чином, спостерігається новий тип архітектури: багато джерел - один одержувач, крім того, обсяг трафіку від вузла може бути як дуже маленьким, так і дуже великим.

Аналіз останніх досліджень і публікацій.

IoT розвивається швидкими темпами. На рисунку 1 приведено найважливіші придбання, інвестиції, результати досліджень та анонси, пов’язані з IoT за 2019 рік, і їх вплив на пошукові запити. IoT збирається увійти в усі сфери життєдіяльності людини: домашнє використання, малий бізнес, або великі компанії.

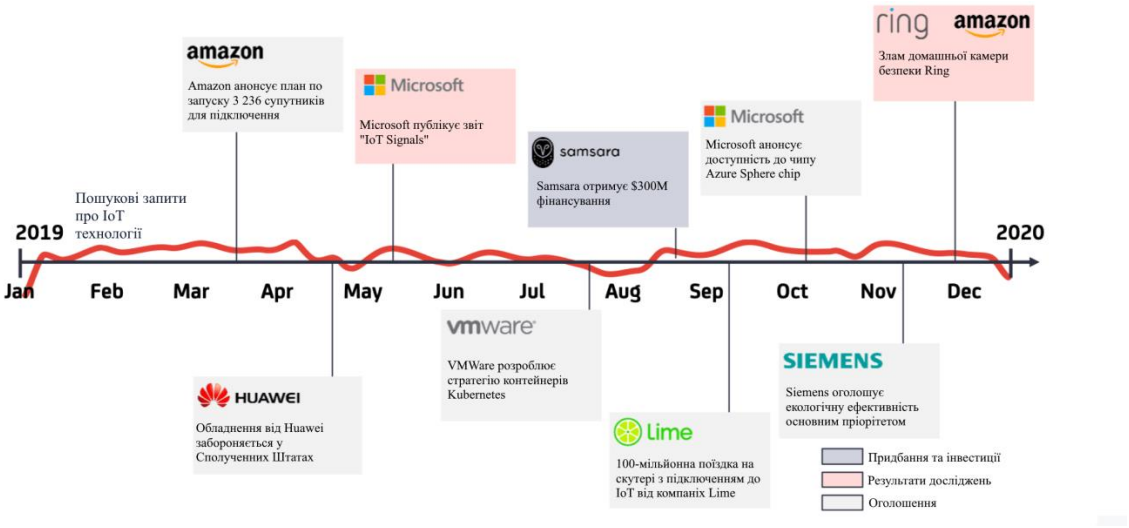

Рис. 1. Вплив інвестицій у IoT технології, досліджень та оголошень на пошукові запити у 2019 році [1]

Постановка завдання. Необхідно визначити протокол обміну даними у системі Інтернету речей, який має матиме мінімальний розмір пакету для забезпечення максимальної енергоефективності та мінімального часу обробки пакету IoT-пристроєм.

Виклад основного матеріалу дослідження.

В XXI столітті ідея виконувати смарт-керування будинком з Інтернет-речами набирає все більшої популярності. Збільшується попит на технології, які дозволяють керувати все більшою кількістю пристроїв [1]. Зростає прагнення зробити свій будинок місцем відпочинку без необхідності виконувати рутинні дії з прибирання, керування безпекою, перевіркою наявності їжі і т.д.

Для реалізації «Розумного будинку» використовують Інтернет речей - це пристрої, які увійшли в мережу та взаємодіють між собою, без участі людини. Приклад «розумного будинку» з використанням Інтернет речей представлено на рисунку 2.

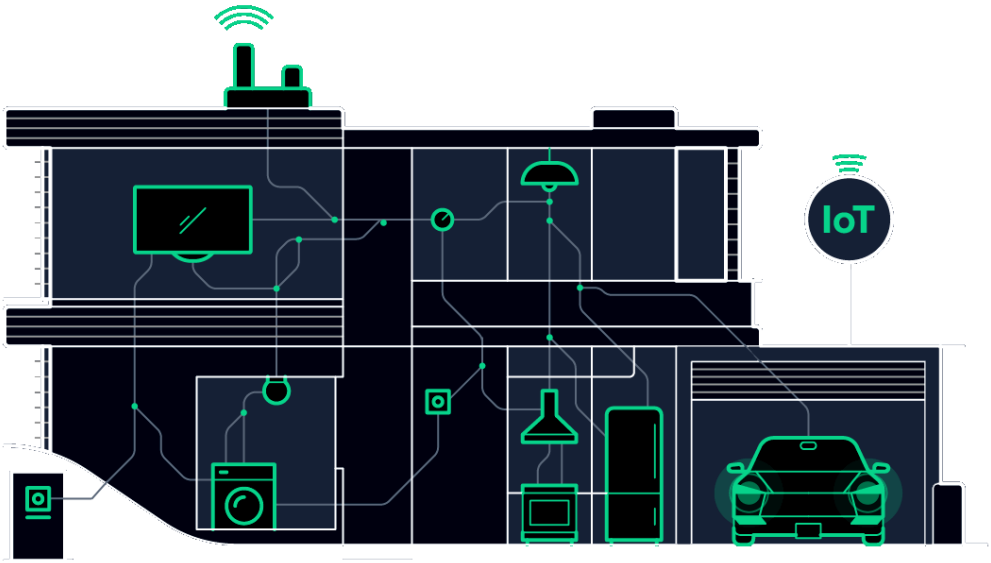

Рис. 2. Приклад «розумного будинку» з використанням Інтернет речей

Один із способів керування системами Інтернету речей – це використання жестів. Для застосування цього виду керування необхідно зчитувати електричні сигнали з м’язів – електроміограму. Електроди можуть підключатися через введення голок до м’язів або накладанням на поверхню шкіри.

Зчитувати електроміограму м’язів здатний спеціальний браслет MYO (рис. 3) від компанії Thalmic Labs.

Рис. 3. Браслет MYO

На рисунку 4 представлена структурна схема системи, в якій керування виконується за допомогою жестів.

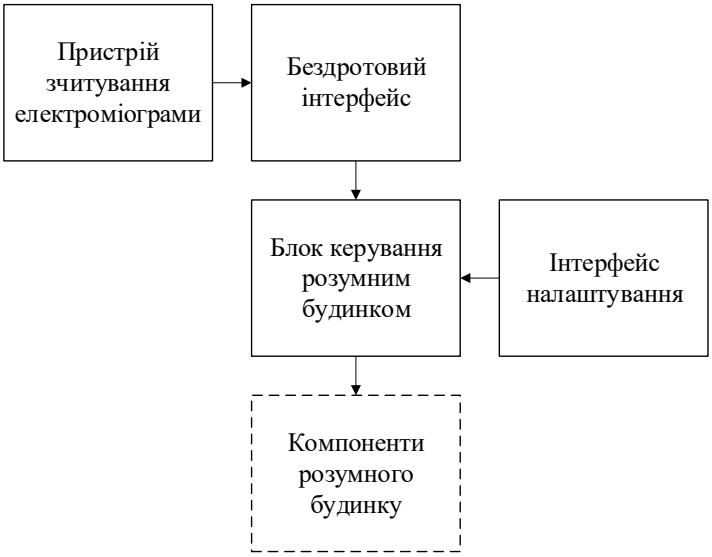

Рис. 4. Структурна схема системи Інтернету речей з керуванням за допомогою зчитування електроміограми

Блок керування будинком - це центральний контролер системи розумного будинку. Він буде використовувати інтерпретацію зчитаних електричних сигналів для керуваннями інтернет речами будинку.

У якості компонентів Інтернету речей можуть виступати системи освітлення, опалення, контроль мікрокліматом, датчики температури, вологості, безпеки, побутові електричні прилади, такі як телевізор, чайник, мікрохвильова піч та інші.

Зчитані електроміограмою електричні сигнали відповідають жестам, що їх виконує людина.

Перед початком використання пристрою для зчитування електроміограми використовується інтерфейс налаштування. За його допомогою виконується налаштування жестів та відповідних їм дій, пов'язаних з керуванням системи Інтернет речей будинку.

Пристрій зчитування електроміограми закріплюється на руці людини. Для передачі зчитаних сигналів використовується мережа, в якій пристрій збору електроміограми виконує зчитування інформації з сенсорів та виконує передачу на сервер обробки даних. Після обробки даних сервер визначає дію, яка має виконатися, та передає команду на пристрої виконання.

На рисунку 5 зображено схему передачі даних.

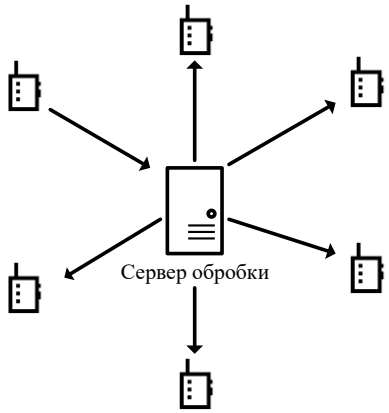

Рис. 5. Передача даних у мережах Інтернету речей

Для побудови мережі обирається спосіб передачі, протокол, за яким виконуватиметься підключення пристроїв у мережі, та протокол обміну даними. Оптимальним способом передачі даних є радіопередача, оскільки це дозволяє вносити зміни в мережу без додаткових дій, виконуючи лише зміну положення вузла у просторі.

Основою усіх протоколів Інтернету речей є технологія M2M (Machine-to-Machine). Згідно з цією технологією обмін даними виконується між машинами у двох- або односторонньому порядку.

У мережах Інтернету речей використовується шаблон обміну даними «видавець-підписник». У цьому шаблоні сервер обробки даних зберігає інформацію про наявних видавців та підписників. Коли видавець передає інформацію про подію, сервер визначає підписників, які мають отримати інформацію про подію, після чого виконує передачу даних.

Такий спосіб обміну даними дозволяє виконувати просте масштабування системи. Це виконується за рахунок відсутності необхідності у зберіганні «видавцем» інформації про усіх «підписників», що дозволить додавати або вилучати вузли з системи у реальному часі.

Для обміну даними у системах Інтернету речей використовуються такі протоколи: HTTP, SOAP, XMPP, STOMP, CoAP, MQTT, MQTT-SN [2,3].

На рисунках 6 – 11 відображено структуру пакетів протоколів.

Тип запиту	Ресурс	Версія протоколу	Параметри обміну	Дані
Заголовок				Тіло повідомлення
Пакет HTTP				

Рис. 6. Структура пакету у протоколі HTTP

У пакеті протоколу HTTP тип запиту визначає, що клієнт бажає виконати з даними. Основними діями є:

– GET – отримати дані;
– POST – відправити дані на сервер;
– PUT – замінити дані на сервері;
– DELETE – видалити дані з серверу [4].

Ресурс у пакеті визначає ідентифікатор об’єкту, з яким відбувається взаємодія.

Версія протоколу допомагає серверу визначити, чи зможе він обробити пакет. В нових версіях протоколу у параметри обміну були додані нові обов’язкові поля, які старі версії серверу не зможуть правильно обробити. Вказана версія дозволить одразу відхилити пакет.

У параметрах обміну вказується інформація про розмір даних, адреса, за якою відбувається підключення до серверу, необхідність підтримки відкритого каналу та інші.

Тіло повідомлення містить в собі дані, що необхідно обробити. Оскільки протокол передбачає передачу текстових даних, то будь-які дані форматуються у текст.

Заголовок	Тіло повідомлення
Пакет SOAP	

Рис. 7. Структура пакету у протоколі SOAP

У протоколі SOAP заголовок містить в собі параметри обміну даними. Тіло повідомлення містить дані для обробки сервером [5]. Оскільки у протоколі відсутнє обов’язкове зазначення ресурсу для взаємодії у заголовку, то його розміщують у тілі повідомлення.

Усі дані форматуються у текст для передачі за протоколом SOAP.

Відправник	Отримувач	Дані
Заголовок		Тіло повідомлення
Пакет XMPP		

Рис. 8. Структура пакету у протоколі XMPP

Протокол XMPP вимагає вказувати у заголовку відправника та отримувача через відсутність постійного з’єднання між клієнтом та сервером [6]. У протоколі можуть використовуватися шлюзи для об’єднання мереж з різними протоколами, що не дозволяє постійно підтримувати відкритий канал зв’язку.

Шлюз дозволяє виконати переадресацію згідно даних, вказаних у заголовку пакету.

Тіло повідомлення у протоколі також використовує текстове форматування, так само як HTTP та SOAP.

<table>
<tr><td>Тип повідомлення</td><td>Параметри</td><td>Дані</td></tr>
<tr><td colspan="2">Заголовок</td><td>Тіло повідомлення</td></tr>
<tr><td colspan="3">Пакет STOMP</td></tr>
</table>

Рис. 9. Структура пакету у протоколі STOMP

Основними типами повідомлення у протоколі STOMP є:

– SUBSCRIBE – клієнт повідомляє, з якого ресурсу бажає отримувати дані, при їх оновлені;

– SEND – клієнт передає нові дані;

– MESSAGE – сервер відправляє клієнту оновлені дані з ресурсу, що був вказаний через повідомлення типу SUBSCRIBE;

– ERROR – це повідомлення відправляє сервер при неможливості обробити дані [7].

У параметри додаються обов’язкові поля, що необхідно відправити в залежності від типу повідомлення. Також можна додати власні поля.

Тіло повідомлення містить дані для обробки сервером.

<table>
<tr><td>Версія протоколу</td><td>Тип повідомлення</td><td>Довжина ідентифікатор пакету</td><td>Ідентифікатор повідомлення</td><td>Ідентифікатор пакету</td><td>Дані</td></tr>
<tr><td colspan="5">Заголовок</td><td>Тіло повідомлення</td></tr>
<tr><td colspan="6">Пакет CoAP</td></tr>
</table>

Рис. 10. Структура пакету у протоколі CoAP

У пакету CoAP версія протоколу необхідна для швидкої відмови у обробці даних, через неможливість обробки нових типів повідомлення при отриманні сервером пакету, у якому версія протоколу старіша за версію клієнта [8].

Тип повідомлення визначає дію, що необхідно виконати над даними. Основними діями є:

– GET – отримання даних з вказаного ресурсу;

– PUT – оновлення даних у вказаному ресурсі;

– POST – перевіряє наявність ресурсу на сервері, та створює ресурс у разі його відсутності;

– DELETE – видалення ресурсу з серверу [8].

Для кожного пакету задається унікальний номер, що зазначається у ідентифікаторі пакету. Це необхідно, оскільки протокол використовується поверх протоколу UDP, який не має алгоритмів підтвердження передачі даних. Наявність унікального ідентифікатора дозволяє відправити повідомлення у відповідь з підтвердженням вдалої передачі, вказуючи ідентифікатор пакету, що було отримано. Якщо сервер отримає пакет з тим же ідентифікатором, то це означає, що відповідь сервера не досягла клієнта.

Довжина ідентифікатора пакету визначає кількість байт, що займатиме ідентифікатор пакету.

Ідентифікатор повідомлення зберігає унікальний номер, що відповідає даним у тілі повідомлення. Це необхідно для перевірки наявності даних у системі. Якщо вони вже присутні, або застарілі, то вони ігноруються.

Згідно з протоколом ресурс, до якого йде звернення, задається у URL адресі сервера.

<table>
<tr><td>Тип повідомлення</td><td>Параметри для типу</td><td>Параметри повідомлення</td><td>Дані</td></tr>
<tr><td colspan="3">Заголовок</td><td>Тіло повідомлення</td></tr>
<tr><td colspan="4">Пакет MQTT</td></tr>
</table>

Рис. 11. Структура пакету у протоколі MQTT та MQTT-SN

Основними типами повідомлення у пакетах протоколів MQTT та MQTT-SN є:

– CONNECT – запит клієнта на підключення до серверу;
– SUBSCRIBE – передача клієнтом до сервера інформації про ресурс, з якого клієнт бажає отримувати дані;
– PUBLISH – оновлення інформації у ресурсі [10].

Параметри для типу займають 4 біта для визначення повідомлень, що відправляються повторно, необхідності підтвердження передачі пакету та необхідності збереження даних після успішною передачі їх клієнтам.

Параметри повідомлення зберігають інформацію про розмір тіла повідомлення, ідентифікатор повідомлення, назву протоколу, його версію та параметри, що необхідні при ініціалізації підключення.

Ідентифікатор ресурсу вказується у тілі повідомлення.

У MQTT-SN для обміну даними використовується протокол UDP замість TCP для зменшення розміру пакету. Також це дозволяє передати дані, не очікуючи підтвердження. Для пристроїв, дані яких не мають бути обов’язково передані, це дає можливість зменшити енерговитрати за рахунок зменшення передач, що підтверджують успішну передачу пакету.

Для порівняння розміру пакетів були обрані такі дані для передачі:

– Число 15 з ідентифікатором int1;
– Число 11 з ідентифікатором int2;
– Число 2004 з ідентифікатором int3;
– Число 1959 з ідентифікатором int4;
– Текст Khersonal National Technical University з ідентифікатором text;
– Ідентифікатор ресурсу /send/info;
– Ідентифікатор серверу iot.server;
– Ідентифікатор клієнту iot.client.

Кожен рядок закінчується послідовністю ASCII символів ‘\r\n’. Послідовність займає 2 байти. Для кодування даних у двійковій формі використовується система кодування ASCII, що визначає розмір 1 байт для кожного символу.

Для протоколів XMPP та SOAP тіло повідомлення має вигляд:

<int1>15</int1>
<int2>11</int2>
<int3>2004</int3>
<int4>1959</int4>
<text>Khersonal National Technical University</text>

Для протоколів HTTP, STOMP, CoAP, MQTT та MQTT-SN тіло повідомлення має вигляд:

int1:15
int2:11
int3:2004
int4:1959
text:Khersonal National Technical University

Через використання мови розмітки XML у тілі пакету XMPP та SOAP для зазначення будь-якого параметру необхідно двічі зазначати ідентифікатор параметру. Це викликає надмірність інформації для передачі. У заданому прикладі кожен параметр потребує додаткову передачу 7 байт.

На рисунку 12 відображено гістограму розмірів тіла пакету для протоколів з текстовим форматуванням та XML форматуванням.

Рис. 12. Гістограма розміру тіла повідомлення

Для кожного з протоколів було складено мінімальний пакет. Кожен пакет використовується для оновлення інформації на серверів.

Пакет для протоколу HTTP:

HTTP:
PUT /send/info HTTP/1.1
Host: iot.server

int1:15
int2:11
int3:2004
int4:1959
text:Khersonal National Technical University

Пакет для протоколу SOAP:

<soap:Envelope>

```
<soap:Body>
<put resource="/send/info">
<int1>15</int1>
<int2>11</int2>
<int3>2004</int3>
<int4>1959</int4>
<text>Khersonal National Technical University</text>
</put>
</soap:Body>
</soap:Envelope>
```


Пакет для протоколу XMPP:


```
<message from="iot.client@domain" to="iot.server@domain/send/info">
<body>
<int1>15</int1>
<int2>11</int2>
<int3>2004</int3>
<int4>1959</int4>
<text>Khersonal National Technical University</text>
</body>
</message>
```


Пакет для протоколу STOMP:


```
SEND
destination:/send/info
content-type:text/plain

int1:15
int2:11
int3:2004
int4:1959
text:Khersonal National Technical University
```


Протоколи CoAP, MQTT та MQTT-SN використовують бінарне кодування для заголовків. Протоколи MQTT та MQTT-SN використовують бінарне кодування для даних. Бінарні дані представлені за допомогою шістнадцятирічних чисел.

Пакет для протоколу CoAP:


```
48 03 7B 7E 00 00 00 00
Uri-Path="/send/info"
FF
int1:15
int2:11
int3:2004
int4:1959
```

text:Khersonal National Technical University

Пакет для протоколу MQTT та MQTT-SN:

34 64 00 08 2F 73 65 6E 64 2F 69 6E 66 6F 00 56 69 6E 74 31 3A 31 35 0D 0A 69 6E 74 31 3A 31 31 0D 0A 69 6E 74 31 3A 32 30 30 34 0D 0A 69 6E 74 31 3A 31 39 35 39 0D 0A 74 65 78 74 3A 4B 68 65 72 73 6F 6E 61 6C 20 4E 61 74 69 6F 6E 61 6C 20 54 65 63 68 6E 69 63 61 6C 20 55 6E 69 76 65 72 73 69 74 79 0D 0A

На рисунку 13 відображено гістограму розмірів пакетів у байтах за протоколами.

Згідно з отриманими результатами мінімальний розмір пакету виявлений у протоколі MQTT. Але цей протокол потребує додатково відправляти пакет з'єднання до передачі пакету з новими даними, що збільшує об'єми даних.

Пакет для з'єднання у протоколі MQTT має вигляд:

10 12 00 04 4D 51 54 54 00 0A 69 6F 74 2E 63 6C 69 65 6E 74

На рисунку 14 відображена гістограма розміру пакетів даних у протоколах з урахуванням пакету з'єднання для протоколу MQTT.

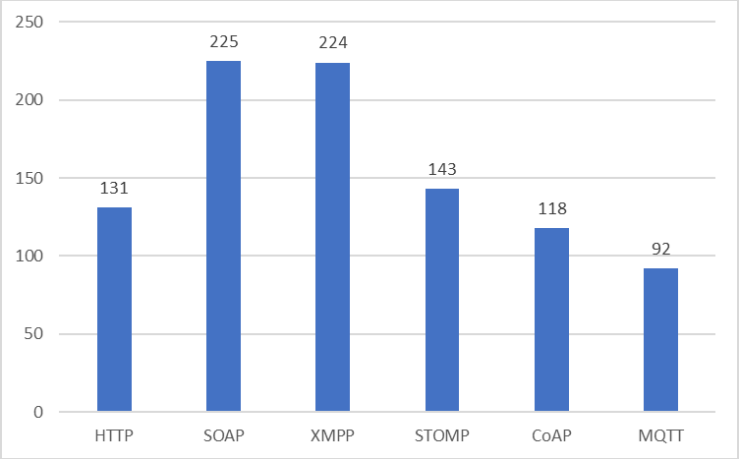

Рис. 13. Розмір пакету даних у протоколах

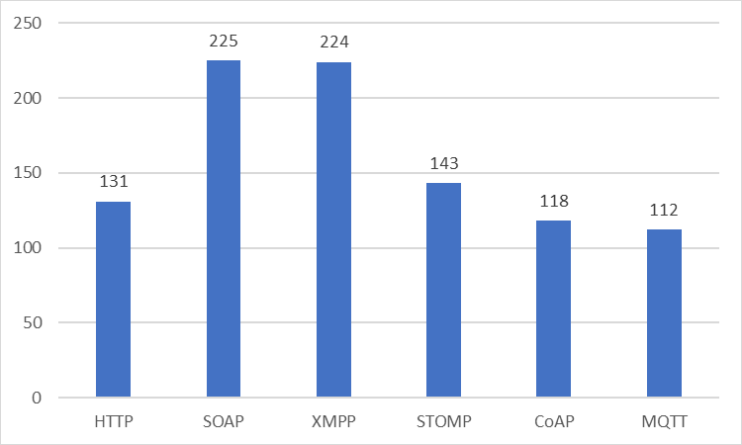

Рис. 14. Розмір пакетів у протоколах з урахуванням пакету з'єднання MQTT

Після додавання пакету з'єднання кількість даних, що передаються за протоколом MQTT, залишається найменшою. Це досягається за рахунок

двійкового кодування даних, що дозволило зменшити розміри заголовку пакету до 2 байт.

Використання двійкового кодування даних дозволяє зменшити розміри пакету при передачі числових значень. Можна використовувати числові ідентифікатори параметрів, що зменшить їх розмір. Можна отримати 256 ідентифікаторів, використовуючи лише один байт даних. Також можливо закодувати числа від 0 до 255 одним байтом. При передачі тризначних чисел це дозволить зменшити розміри пакету на 2 байти з кожного числа.

При передачі 256 тризначних чисел, двійкове кодування дозволить зменшити розміри пакету на: 256 * 2 = 512 байт.

Оскільки більшість даних з пристроїв мають числові значення, то двійкове кодування є оптимальним в протоколах систем Інтернету речей.

Заголовок пакету у протоколі CoAP також кодується у двійковій формі, але він має більше параметрів. Також протокол вимагає розміщення ідентифікатора ресурсу в окремий параметр одразу після заголовку. Опис параметру вимагає вказувати ідентифікатор параметру, що не кодується двійковим кодом, а використовує текстовий опис. Опис параметру потребує 9 байт даних. В протоколі MQTT використовується додатково 2 байти перед описом ідентифікатору ресурсу, для зазначення довжини ідентифікатору.

Перевагою протоколу CoAP є схожість з протоколом HTTP, що дозволяє використовувати архітектуру REST при побудові серверів.

Також перевагою протоколу, відносно протоколу MQTT, є миттєве оновлення даних. У протоколі MQTT для оновлення даних на пристрої, для якого ці дані потребуються, пристрій має виконати підключення до серверу та отримати нові дані. Проміжки часу, через які виконується оновлення даних, залежить від конфігурації пристрою, через що отримання команди для виконання може коливатися від декількох секунд до декількох годин, що не дозволить використовувати протокол MQTT у системах швидкого реагування. Протокол CoAP вимагає постійного очікування пристроєм підключення від серверу для оновлення даних.

Висновки.

Використання пристроїв зчитування електроміограми дозволяє виконувати керування системами Інтернету речей за допомогою жестів.

Для обміну даними у системах Інтернету речей було розглянуто пакети даних існуючих протоколів, та були отримані такі результати:

– Протоколи, що використовують мову розмітки XML, для опису тіла повідомлення мають надмірність інформації у повідомленні.

– Протоколи з двійковим кодуванням даних дозволяють зменшити розміри повідомлення при передачі чисельних значень.

– Протоколи MQTT та MQTT-SN мають мінімальний розмір пакету за рахунок двійкового кодування даних та мінімального розміру заголовку. Мінімальний розмір пакету зменшує час передачі даних. Це зменшує енерговитрати у автономних системах.

– Протокол CoAP за рахунок двійкового кодування заголовку та схожістю з HTTP протоколом дозволяє розробляти системи з REST архітектурою.

– Оптимальним протоколом для керування системами Інтернету речей є CoAP, через найменший розмір пакету даних серед протоколів, що одразу виконують оновлення даних на пристроях.

Список використаних джерел.

1. Knud Lasse Lueth. IoT 2019 in Review: The 10 Most Relevant IoT Developments of the Year. URL: https://iot-analytics.com/iot-2019-in-review/ (дата звернення 26.02.2020).
2. В. Гойхман, А. Савельєва. Аналитический обзор протоколов Интернета вещей. *Технологии и средства связи*. 2016. (№4). С. 32-37 URL: http://lib.tssonline.ru/articles2/reviews/analiticheskiy-obzor-protokolov-interneta-veschey (дата звернення 26.02.2020).
3. Протоколы передачи данных. URL: https://iot.ru/wiki/protokoly-peredachi-dannykh-iot (дата звернення 26.02.2020).
4. R. Fielding, J. Reschke. Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing. URL: https://httpwg.org/specs/rfc7230.html#http.message (дата звернення 26.02.2020).
5. XML Soap. URL: https://www.w3schools.com/xml/xml_soap.asp (дата звернення 26.02.2020).
6. XML Stanzas. Extensible Messaging and Presence Protocol (XMPP): Core. URL: https://xmpp.org/rfcs/rfc6120.html#stanzas (дата звернення 26.02.2020).
7. STOMP Protocol Specification, Version 1.2. URL: https://stomp.github.io/stomp-specification-1.2.html (дата звернення 26.02.2020).
8. RFC 7252. The Constrained Application Protocol: Method Definitions. URL: https://tools.ietf.org/html/rfc7252#section-5.8 (дата звернення 26.02.2020).
9. Francesco Azzola. CoAP Protocol: Step-by-Step Guide. URL: https://dzone.com/articles/coap-protocol-step-by-step-guide (дата звернення 26.02.2020).
10. MQTT Packet Structure. URL: http://www.bytesofgigabytes.com/mqtt/mqtt-protocol-packet-structure/ (дата звернення 26.02.2020).
11. Іванчук О.В., Завгородній В.В., Козел В.М., Дроздова Є.А. Аналіз протоколів обміну даними для керування системами інтернету речей // Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки. - 2020. - Т. 31 (70), № 2. - С. 99-104.