ЮГОЗАПАДЕН УНИВЕРСИТЕТ
„Неофит Рилски“
Технически факултет

Катедра „Комуникационна и компютърна техника и технологии“

маг. инж. Мартин Пандурски

АВТОРЕФЕРАТ

ИЗСЛЕДВАНЕ НА ВЪЗМОЖНОСТИТЕ ЗА ИНТЕГРАЦИЯ НА СЕНЗОРНИТЕ МРЕЖИ В ОБЛАЧНИ СТРУКТУРИ

за присъждане на научна и образователна степен ”доктор”
в ПН 5.3. „Комуникационна и компютърна техника“,
по научната специалност „Компютърни системи, комплекси и мрежи “

Научен ръководител:
Гл. ас. д-р инж. Филип Цветанов
2022

Дисертационният труд е обсъден на заседание на катедрен съвет на катедра “Комуникационна и компютърна техника и технологии” при Технически Факултет на Югозападен университет „Неофит Рилски“ на 10.02.2022 г. и е насрочен за официална защита пред

Научно жури в състав: проф. дтн инж. Петър Апостолов - ЮЗУ, проф. дтн. инж. Галина Чернева -ЮЗУ, проф. д-р инж. Георги Любенов Илиев - ТУ София, доц. д-р инж. Агата Христова Манолова -ТУ София, доц. д-р инж. Иван Динков Иванов -ВУТП София.

Резервни членове на научното жури: доц. д-р инж. Иван Недялков - ЮЗУ, доц. д-р инж. Валентина Илиева Маркова - ТУ Варна

Дисертационният труд съдържа: брой страници -182, брой фигури – 119, брой таблици – 10, брой формули - 25, брой литературни източници -155.

По темата на дисертацията са направени 9 публикации.

Номерацията на фигурите, таблиците и формулите в автореферата отговаря на тези в дисертацията.

Материалите по защитата се намират в секретариата на катедра „Комуникационна и компютърна техника и технологии“ и на Интернет страницата на Югозападен Университет.

Автор: маг. инж. Мартин Пандурски

Заглавие: ИЗСЛЕДВАНЕ НА ВЪЗМОЖНОСТИТЕ ЗА ИНТЕГРАЦИЯ НА СЕНЗОРНИТЕ МРЕЖИ В ОБЛАЧНИ СТРУКТУРИ

Тираж: 20 бр.

Проект, оформление и предпечатна подготовка: авторът

I. ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД

Актуалност на проблема

Съвременният технологичен напредък в миниатюризацията на устройства и безжичните сензорни технологии, разширява възможностите за приложения на безжичните сензорни мрежи (БСМ). Сензорната мрежа е самоорганизираща се и съставена от голям брой, различен тип сензори, които са разполагат в определена зона за мониторинг и предават данни помежду си чрез безжична комуникация.

Приложенията на БСМ са многобройни, като наблюдение на околната среда, индустрията и т.н. Количеството данни, които генерират сензорните мрежи е огромно и разнородно.

Специфична особеност на БСМ са ограничените ресурси. Ограниченията включват изисквания за малък разход на енергия, периметър на комуникация, ниска честотна лента, ограничена обработка на сензорните данни и др.

За ефективното използване на големите обеми от данни, събирани от сензорните мрежи е нужна мощна, мащабируема високопроизводителна изчислителна инфраструктура за обработка и съхранение на данни в реално време, както и за анализ на обработената информация.

Облачните структури предоставят огромна изчислителна мощност и място за съхранение на данни. Интеграцията между облаците и сензорните мрежи е чудесно решение на проблема с ограничената изчислителна мощност на сензорните мрежи, за съхранение и обработката на данните. За осъществяване на тази интеграция е формулирана нова парадигма наречена Sensor Cloud Computing.

Актуалността на работата се обуславя и от факта, че БСМ предоставят възможност за разработване на иновативни подходи за събиране, обработка на данни и интеграцията им в облачни системи за дистанционно управление в много области на индустриалната автоматизация, мониторинга на околната среда и др.

Настоящата работа е насочена към изследване на някой възможности и алгоритми за ефективна работа на сензорните мрежи при интеграцията на сензорните данни в облачни структури.

НАУЧНА ОПРЕДЕЛЕНОСТ НА ИЗСЛЕДВАНЕТО

Обект на изследването

Обект на изследване на дисертационния труд са сензорни мрежи, базирани на стандарта IEEE 802.15.4.

Предмет на изследване са модели с клъстерна и меш топологии и предаване на данни за стандартите IEEE 802.15.4, модели за интеграция на сензорни данни към облачни структури.

Цел на дисертацията:

Да се изследва работата на безжични сензорни мрежи и предложат модели и алгоритми за интегриране на сензорните данни към облачни структури.

Задачи:

1. Изследване и анализ на съвременни методи за създаване и управление на сензорни мрежи.
2. Анализ на протоколите и проблемите на интеграцията на сензорните мрежи с облачни структури.
3. Физическо изграждане на структура сензорна мрежа-облак и разработване на алгоритъм за ефективно интегриране на сензорни данни чрез приемане, предаване, визуализиране и анализиране на данни от сензорни устройства.
4. Да се изследват, симулационно и физически сигурността на предаваните сензорни данни в облачна структура.
5. Да се оцени влиянието на протоколите и механизмите предоставящи услуга за интеграция в системата сензорна мрежа – облак, върху ефективността за предоставяне на пакетите данни.

Методи на изследване

За решаване на поставените задачи в дисертационната работа се използват методите на системния анализ, синтез, имитационно моделиране, компютърна симулация и програмиране, емпирични методи като наблюдение, сравнение и практически експерименти.

Апробация на изследването

В процеса на изследване са реализирани симулационни и експериментални опитни постановки, позволяващи провеждане на редица изследвания на протоколи за комуникация и пренос на сензорни данни, както и разработване на управляващ софтуер за интеграция на данните към облак. Междинните резултати са отразени в *девет публикации*, които са докладвани на международни конференции, в списания, от които 5 са в реферирани бази данни Scopus и Web of Science, 2 са в базата данни на Нацид, 2 в международни вторични бази от данни Google Scholar, EBSCO, Crossref, Publons, DOAJ с др.

Структура и обем на дисертационния труд

Дисертационният труд е в обем от 182 страници, като включва увод, три глави за решаване на формулираните основни задачи, списък на основните приноси, списък на публикациите по дисертацията, използваната литература и приложения. Цитирани са общо 155 литературни източници, като 144 са на латиница и 11 на кирилица. Работата включва общо 119 фигури и 10 таблици. Номерата на фигурите и таблиците в автореферата съответстват на тези в дисертационния труд.

Практическа приложимост: Практическата приложимост е

голяма, поради факта че броят на интернет свързаните устройства нараства непрекъснато. Обема на информацията непрекъснато се увеличава и са нужни нови решения за интеграция на сензорните мрежи към облачни структури за повишаване на ефективността и свързаността.

II. СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД

ПЪРВА ГЛАВА

Анализ на съвременните подходи за създаване и управление на безжични сензорни мрежи

В резултат на анализа на съвременните методи за изграждане и управление БСМ, са:

- ✓ Проучени и систематизирани основните направления, върху които са провеждани изследвания за специфичните особености, изисквания и предизвикателства, стоящи пред безжичните сензори, основен компонент при изграждане на сензорните мрежи;
- ✓ Изследвани и систематизирани са архитектурата на БСМ, моделите за комуникация и управление на захранването, мобилността и задачите в сензорните мрежи;
- ✓ Анализирана е структурата на сензорните мрежи и определени спецификата на приложение на топологиите звезда, дървовидна, меш, хибридна и клъстерна топология при изграждането и приложението на сензорните мрежи;
- ✓ Изследвани и анализирани са основните проблеми на сигурността, предизвикателствата, стоящи пред сензорните мрежи като ограничените ресурси на сензорите, радиокомуникационната среда за предаване на данните, управление на енергийната консумация от сензора, лимитирана мощност на процесора и радиообсег;
- ✓ Ограничения изчислителен капацитет на сензорните мрежи, води до проблем при анализа и съхранението на генерираните огромно количество сензорни данни. Интеграцията между облачните структури и сензорните мрежи дава възможност за решение на този проблем, както и за съхранени и обработка на събраните данни, без излишно оскъпяване себестойността на сензорните мрежи;
- ✓ Формулирана е тезата, че създаването на модели за интеграция на сензорните данни към облак, позволява да се съкрати времето за разработване и оцени ефективността на сензорните мрежи, да се осигури по-висока надеждност и енергийна ефективност на проектираните системи, което е актуален проблем в областта на сензорните мрежи.
- ✓ Проведени са експериментални изследвания върху ефективността на архитектурата на БСМ.

1.3.2. Архитектура на БСМ и модел на комуникация и управление

Архитектурата на БСМ е изградена от различни типове възли, използвани за наблюдение на различни параметри в околната среда. Тези възли могат да се използват в различни приложения в реално време и ефективно маршрутизиране между БС и възлите. Архитектурата на БСМ бива два 2 типа: Многослойна мрежова архитектура и Клъстерна архитектура. *Мнослойна мрежова архитектура,* използва множество сензорни възли и една мощна БС.

Клъстерна мрежова архитектура, тук отделни сензорни възли се добавят в групи (клъстери), чрез LEACH Protocol „Протокол за извличане“. Предимството на LEACH е в намаляване консумацията на енергия чрез създаване на клъстери и йерархично маршрутизиране като по този начин се увеличава живота на БСМ, защото скъсява времето за предаване на данни на всеки сензорен възел. Всички възли в мрежата са организирани в локални клъстери, с една клъстерна "глава". Функциите на клъстерната глава изискват тя да има висока енергийна ефективност. За да се избере клъстерна глава с най-висока енергийна ефективност са създадени различни варианти на LEACH.

Алгоритъмът за формиране на клъстери се проектира така, че всеки възел в мрежата последователно да играе ролята на глава на клъстер за еднакъв отрязък от време, фиг.1.10, при условие че всички възли започват с едно и също количество енергия. Праговата стойност се определя съгласно израза 1.1.

$$Tn(LEACH) = \begin{cases} \frac{p}{1-p\times\left(r\times mod\frac{1}{p}\right)} & if\ n \in G \\ 0 & otherwise \end{cases} \qquad 1.1.$$

Където:

P - предварително определен процент за възможния брой главни възли;

R - текущия интервал за функциониране;

G - брой сензорни възли, които не са избрани за главни през последните $1/p$ интервали.

Ако това случайно число е по-малко от праговата стойност, T (n), възелът става клъстерна глава за текущия кръг (цикъл). Стойността на прага се изчислява въз основа на уравнение (1.1). Когато даден възел е избран да бъде клъстерна глава, той излъчва съобщение ADV. Това съобщение съдържа идентификатора на възела и заглавие. Всеки възел, който не е клъстер, определя към кой клъстер ще принадлежи, като избира клъстер, който изисква минималната комуникационна енергия, въз основа на силата на получения сигнал от ADV от всички клъстерни глави.

Фиг.1.10. Алгоритъм на LEACH

Всеки възел предава съобщение за заявка за присъединяване *Join-REQ* обратно към избраната глава на клъстера. Клъстерите в LEACH действат като локални контролни центрове за координиране на предаването на данни. Възелът на клъстерната глава създава TDMA график и предава този график на възлите в клъстера. Това гарантира, че няма да има конфликти между предаваните съобщения и също така позволява радиокомпонентите на всеки краен възел да работят само, при предаване на данни, като минимизират използването на енергия.

1.3.2.2. Експериментално изследване енергийната ефективност на сензорна мрежа с клъстерна топология [A7].

Планувано е експериментално стимулационно изследване работата на LEACH, H-LEACH и N-LEACH протоколите. H-LEACH минимизира енергийните разходи като в първия кръг работи като LEACH. В следващите кръгове, използва метод за оптимизиране на разстоянието за предаване на данни до базова станция, за да се избере кои възли ще станат клъстерни глави. N-LEACH поддържа информация за оставащата енергия на всички възли и я изпраща до БС при всеки цикъл. Разработен е код с Matlab за реализация на алгоритъма.

Алгоритъм

// n е множеството от всички възли в мрежата, r е броят на кръговете, p е вероятността сензор да бъде глава на клъстер.

Стъпка 1: Разположение на всички n възли на случаен принцип

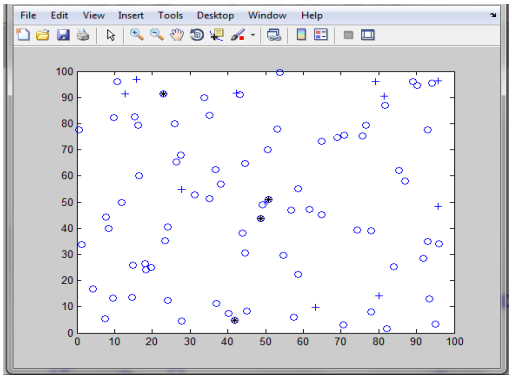
фиг.1.13.

***Стъпка 2*:** Задаване на първоначална еднаква енергия на всички възли.

Стъпка 3: Избор на клъстерна глава в над 0,5 процента от възлите в мрежата. В LEACH-N, възли с по-високи енергийни нива са избрани за клъстерни глави. В LEACH-H за клъстерни глави се избират възли, които са най-близо до БС.

Стъпка 4: Всяка не-СН възел се присъединява към най-близката глава на клъстера.

Стъпка 5: Всяка СН предава обобщените данни на БС по най - краткия път.

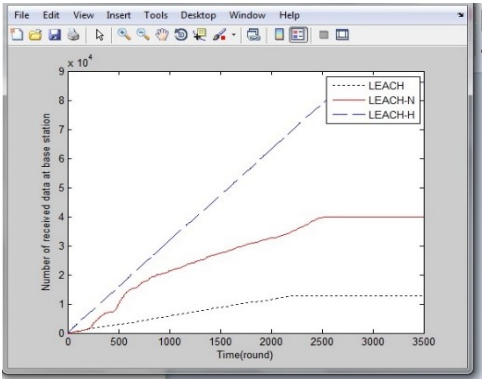


Фиг.1.11.Рандомизирано разположение на сензорите в сензорното поле

Резултати от симулационното изследване

А) Брой на получените данни на базовата станция, фиг.1.14

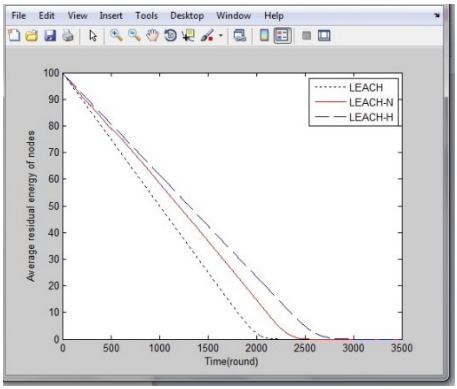
Резултатите показват, че броя на предадените пакети данни, изпратен от LEACH-H, бавно се увеличават и стават по-големи от тези на LEACH и LEACH-N.



Фиг.1.12. Брой получени данни до БС в LEACH, LEACH-N и LEACH-H

Б) Средна остатъчна енергия на възлите, фиг.1.15

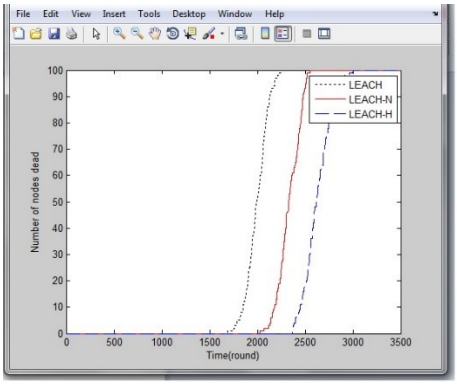
Експериментите показват, че средната остатъчна енергия на възлите при LEACH е по-малка от тази при работа с LEACH-N и LEACH-H.



Фиг.1.13. Средна остатъчна енергия на възлите при LEACH. LEACH-N и LEACH-H

В) Брой изтощени възли, фиг.1.16

При вероятност 0,5 за 100 възли и 3500 кръга, от фиг.1.16 е видно, че при около 1500 кръга възлите започват да умират в LEACH. При LEACH-N - около 2000, а при LEACH-H - около 2500. Резултатите показват, че в LEACH-H показва най-добрите резултати от гледна точка на подобряване на живота на мрежата в сравнение другите два протокола.



Фиг.1.16. Брой изтощени възли в LEACH, LEACH-N и LEACH-H

Обикновено най-енерго ефективният път се определя като оптимален път за предаване на данни. По този начин всеки възел трябва да е запознат с възможностите на своите съседи, за да избере най-добрия съсед, по който да изпраща данни. Въпреки че е постигнато значително подобрение на жизнеспособността на мрежата, този тип методи изискват често актуализиране на

информацията за енергията на пътя в таблиците за маршрутизиране, което води до допълнително натрупване на самоорганизирани безжични сензорни мрежи.

1.5.5. Експериментално изследване влиянието на топологията върху ефективността на комуникационния обхват на мрежата [A 5]

За изследване влиянието на топологията е плануван и проведен е експеримент, чиято методология обхваща:

1) *Хардуерен дизайн* на XBee сензорна мрежа със седем XBee модула, които имат функции на координатор, рутери и крайни устройства, с три различни топологии: звезда, клъстер и меш.

2) Мрежов дизайн, конфигуриране са три вида мрежи тип звезда, меш и клъстерна.

3) ***Алгоритъм*** *за изследване влиянието топологията върху ефективността на комуникационния обхват на сензорната мрежа.*

Стъпка 1. *Избор на параметрите на хардуерните модули*

Стъпка 2. *Инсталиране на специализирания софтуер XCTU.* Софтуерът включва инструменти, улесняващи настройката на XBee модули..

Стъпка 3. *Конфигуриране режима на работа на модулите и актуализация на фърмуера в два режима. XBee модули в AT transparent режим.* Всички серийни данни, получени от модула XBee се изпраща безжично до отдалечените модули. *Отдалечени модули в API режим.* За да тества обхвата на безжичната мрежа, XCTU трябва има достъп до поне едно отдалечено устройство XBee3 в мрежата. В *API режим.* Изпраща се API *Tx* рамка от един XBee модул към друг модул API *Rx.*

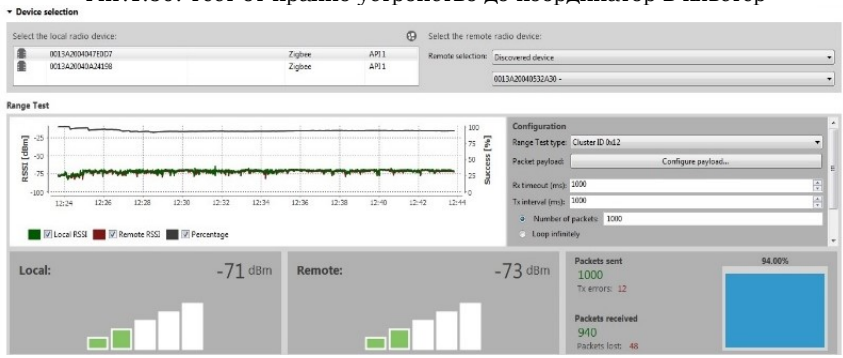
Стъпка 4. *Конфигуриране функциите на устройствата*, като крайно устройство, рутер и координатор.

Стъпка 5. *Провеждане на теста.* При стартирането на теста, XCTU изпраща пакети данни от локалния модул XBee към отдалечения и изчаква ехото да се върне. XCTU отчита броя на предадените пакети и оценява силата на сигнала от двете страни чрез RSSI.

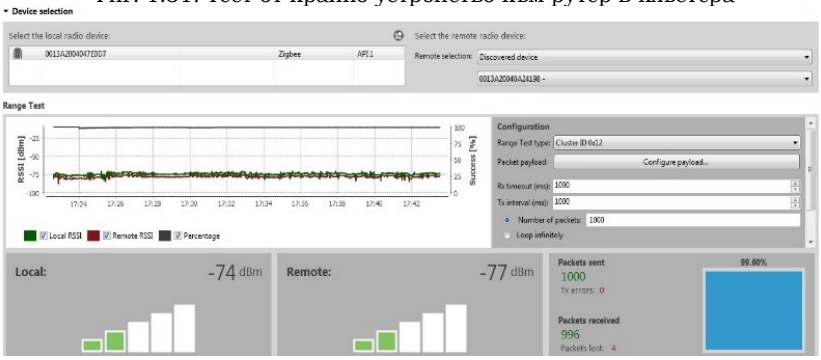
Стъпка 6. *Анализ на получените резултати.* Загубата на пакет от данни, която е над 5%, може да доведе до влошаване на качеството на работата на мрежата. Анализът на получените параметри в резултат на експеримента дава отговор на въпроса за най-ефективната топология. На фиг. 1.30. са показани резултатите при предаване на данни от крайно устройство до координатор в клъстера. На фиг.1.31. са показани пакетите, грешките при закъснение, RSSI диаграма при предаване на данни от крайно устройство към рутер в клъстера. Резултатите от теста за предаване

на данни от рутер до координатор в клъстера е показано на фиг.1.32. Показани са процентите на получените пакети, загубените пакети, грешки при закъснение, RSSI диаграма и за Router to Coordinator в клъстера.

Фиг.1.30. Тест от крайно устройство до координатор в клъстер



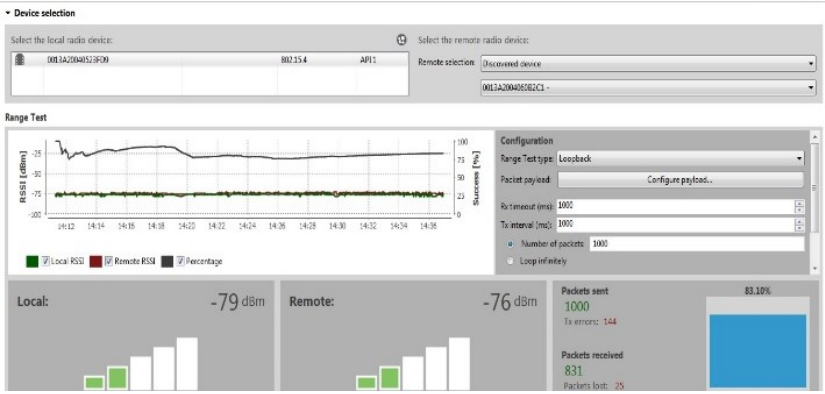
Фиг. 1.31. Тест от крайно устройство към рутер в клъстера



Фиг. 1.32. Тест от рутер до координатор в клъстера

Резултатите от теста на крайното устройство до координатора при топология звезда е показан на фиг.1.33, тестът за обхват

показва процента на приетите пакети.



Фиг. 1.33. Тест от крайно устройство до координатор в звезда

Резултатите от теста на крайно устройство до крайно устройство в звезда е показано на фиг.1.34.

Фиг. 1.34. Тест от крайно устройство до крайно устройство в звезда

Резултатите от теста на рутер към рутер при Меш топология на фиг.1.35. Анализът на резултатите за комуникационна ефективност на сензорните мрежи при различни топологии е извършен съгласно указанията в работи [57] и показва че:

1) При Топология звезда данните, изпратени от крайните устройства до координатора, нямат загуба на пакети от разстояние под 10 м. Но има загуба на пакети от разстояние над 10 м. Тестовете са извършени с 1000 пакета за предаване. Получените пакети са средно 82% .Тази топология показва най-ниското ниво на резултати по отношение на разстоянието на обхвата, приетите пакети и забавянията.

2) При Меш Топология, с увеличаването на разстоянието се увеличава и загубата на пакети и забавянето, тъй като

комуникацията на дълги разстояния отнема време в процеса на предаване на данни през рутера.

Фиг. 1.35. Тест от рутер към рутер в Меш топология

Тестовете са извършени с 1000 пакета за предаване. Получените пакети са 90,22%. Тази топология показва по-добро качество на резултатите, но може да доведе до намалено качество на производителността на мрежата.

3) При Клъстерна Топология, тестовете са направени с три различни типа устройства - координатор, рутер и крайно устройство. Извършени са тестове между координатор-маршрутизатор, крайно устройство до координатора и крайно устройство до маршрутизатор. При тестовете са предавани по 1000 пакета. Получените пакети са средно 97,30%, което не води до влошаване на качеството на мрежата. Резултатите показват, че най-голям успех има предаването на данни между координатор-маршрутизатор, последван от маршрутизатор-координатор, маршрутизатор-маршрутизатор, крайно устройство-координатор крайно устройство- маршрутизатор и крайно устройство-крайно устройство. Резултатите показват, че клъстерната топология има значително по-висок процент на успех от другите две топологии.

В резултат на извършеното литературно изследване и направените експериментални изследвания **може да обобщим**:

- Установено е, че в последните години към проблемите на сензорните мрежи има засилен интерес за внедряване в индустрията и се провеждат интензивни научни изследвания върху проектирането и изграждането на безжични сензори с характеристики, които да преодоляват чест от маркираните по горе конструктивни дефицити. Проведените изследвания засягат също и въпросите, свързани със сигурността на мрежите, енергоемкостта на сензорите и живота на сензорните мрежи.

- Няма универсална методика и инструментариум за изследване и оценка на параметрите на сензорните мрежи, поради което е необходимо разработване на нови модели, подходи и алгоритми за провеждане на експериментални и симулационни изследвания.

- Важен проблем при сензорните мрежи е генерирането на огромно количество сензорни данни, ограничената изчислителна мощност и ограничения капацитет на сензорните мрежи, а от там друг проблем свързан с живота на мрежата. Интеграцията между облачните структури и сензорните мрежи дава възможност за решение на този проблем, както и за съхранени и обработка на събраните данни, без излишно оскъпяване себестойността на сензорните мрежи.

- Формулирана е тезата, че създаването на модели и алгоритми за интеграция на сензорните данни към облак, позволява да се съкрати времето за разработване и оцени ефективността на сензорните мрежи, да се осигури по-висока надеждност и енергийна ефективност на проектираните системи, което е актуален проблем в областта на сензорните мрежи.

ВТОРА ГЛАВА

Модели и алгоритми за интеграция на сензорните данни в облачни структури

2.1. Анализ на съвременните решения за съхранение и обработка сензорните данни

Данните, които се извличат от сензорите и се представят като прости времеви редове се наричат *сензорни данни*. Често за дълъг период от време данните от сензора може да не променят, а в други случаи да нарастват бързо, няколко отчета в секунда за един или повече сензори от една мрежа. Основните модели за съхранение и обработка на сензорните данни са: 1)Съхранение и обработка на данни извън сензорната мрежа и 2) Разпределено съхранение и обработка на сензорните данни в мрежата.

Базите данни с времеви редове TSBD (time series database) са специално проектирани за съхраняване на сензорни данни. TSDB са софтуерна система, оптимизирана за обработка на данни от времеви редове, масиви от числа, индексирани по време (дата -час). Характерните свойства на TSDB са управлението на жизнения цикъл на данните, обобщаването и сканирането на голям обхват на много записи.

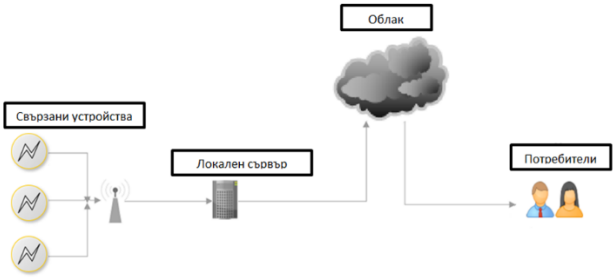
2.2. Структури за интеграция на сензорните данни в облачни структури

2.2.2. Характеристики на облачните структури

Облачните структури са едновременно платформа и вид приложение.

Облачните структури предоставят: 1) Услуга при поискване, 2) Измерване на услугата, 3)Обединяване на ресурсите.

2.2.3. Модели облаци, според ниво на достъпност са частен, публичен, обществен и хибридна облачна инфраструктура. Функционалната структура на свързване на сензорните данни с облаците е показана на фиг.2.4.



Фиг.2.4. Функционална структура на предаване на сензорните данни към облаците

Комбинирайки различните технологии, може да се създават различни системи за решаване на необходимите задачи. Например, Mathworks предоставя готова облачна платформа за IoT приложения, *ThingSpeak*. Тази безплатна облачна услуга събира сензорни данни и с инструментите, предоставени от приложението, предоставя визуализация и анализ на различни типове данни.

2.3. Протоколи за комуникация и предаване на данните между сензорни мрежи и облачни платформи

2.3.1. Комуникационни протоколи за интегриране на сензорни данни към облак

Най-голямото предизвикателство при проектиране на IoT системи е установяването на *комуникационен канал между устройства, шлюз и облачни платформи.* Следователно, това изисква използването на различни протоколи. Пълният комуникационен стек се състои от четири различни нива - „слоеве“ [85] показани в табл.2.1.

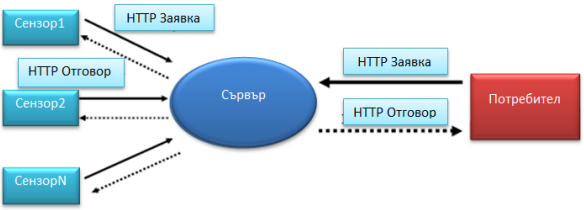
Таб.2.1. Протоколи за комуникационен слой при предаване на данни към облак

Слой	**Функции**	**Протоколи**
Приложен слой	Съхранява полезен товар	•*MQTT* (Message Queuing Telemetry Transport) • HTTP, REST (Representational state transfer), RESTful
Транспортен слой	Осигуряват комуникация от край	TCP и UDP

	до край за приложения слой.	
Интернет слой	Интернет протокол (IP), слой, определя прехвърлящи данни между хостове.	IPv4 и IPv6
Канален слой	Управлява връзка хост - мрежа	•802.11 WiFi •Zigbee •Ethernet

2.3.2. HTTP за комуникация в сензорни и IoT мрежи

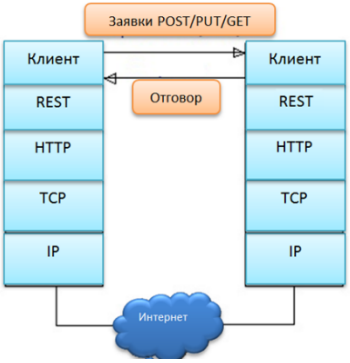
Протоколът HTTP определя начина, на предаване на съобщенията и форматират в Интернет. HTTP прехвърля голям брой малки пакети, когато комуникира с IoT. Връзките TCP се освобождават при всеки достъп и прехвърлят въз основа на IP и URL адрес и тяхната връзка се променя динамично. Следователно комуникацията в IoT причинява консумация на мрежови ресурси и големи закъснения.



Фиг. 2.7. Конфигурация на системата с помощта на HTTP

2.3.3. Rest и Rest full

Rest е софтуерна архитектура за реализация на уеб услуги, като се използва с HTTP. REST се състои от клиенти и сървъри и използва архитектура, базирана на шина, където не е необходим брокер и крайните устройства могат да комуникират директно.



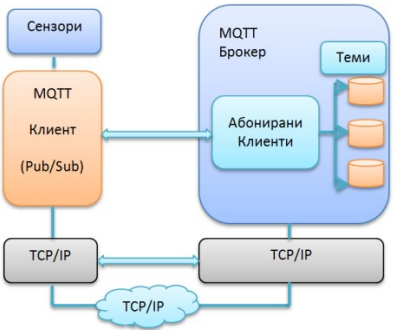
Фиг.2.9. Архитектура на RESTful

REST означава сървър, който споделя JSON файловете с клиент по HTTP. Клиентите инициират заявки към сървърите; сървърите

преработват заявките и връщат подходящи отговори. *Restful* е усъвършенствана форма на сървър за уеб обмен, който споделя всички други документи или JSON файлове за разработване на нови приложения.

2.3.4. Протокол MQTT

MQTT е транспортен протокол за съобщения, базиран на архитектура *publish/subscribe.* MQTT работи на TCP/IP, които осигуряват подредени, двупосочни връзки. *Publish/subscribe* се управлява от събития и позволява съобщенията да се придвижват между устройствата и клиентите, чрез брокер, който управлява *Publish/subscribe* операциите. Важен елемент на *MQTT* архитектурата е брокерът, който обработва съобщенията, отделя издателя от абоната и действа като рутер за съобщенията. Всеки клиент, който публикува съобщение до брокера, включва тема в съобщението.

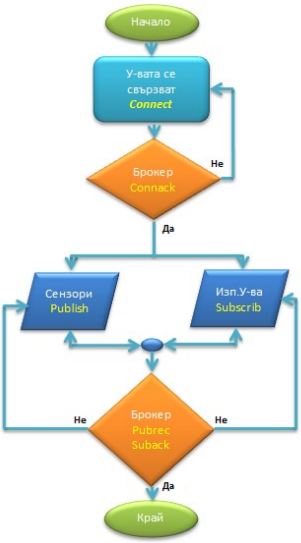


Фиг.2.11. *MQTT* архитектура, абониране и теми

Пакетната структура на MQTT , фиг.2.14 [94] се състои от: *Фиксирано заглавие (Fixed header)* - Съдържа командите за връзка (CONNECT, PUBLISH.) 2 байта. *Заглавие на променлива (Variable Header).* Съдържанието на заглавието варира в зависимост от типа на пакета (Информация за флаговете, пакетен идентификатор). *Полезен товар (Payload)* - Данните, които трябва да бъдат изпратени.

Фиг. 2.14. *MQTT* Пакетна структура

На фиг.2.15 е показана последователността за извършване на обмен на съобщенията.



Фиг.2.15. *MQTT* Обмен на заявки. Блок-схема

2.3.5. Модел на очакваното закъснение при предаване на данните при комуникации базирани на MQTT протокола.

Моделът [97] описва връзката между размера на данните, интервалите за събиране на данни, мрежовия трафик и забавянето, като е представен подход за моделиране на мрежов *MQTT* дизайн, базиран на експериментиране на поведението на ниво пакет. Закъснението от край до край в мрежовата връзка се състои от различни събития, които са изразени чрез уравнение 2.1. За всеки даден възел съществува забавяне поради обработка, опашка, предаване и разпространение.

$$D_{node} = D_{processing} + D_{queuing} + D_{transmission} + D_{propagation} \qquad (2.1)$$

Закъсненията в обработката и предаването възникват във възела, а забавянето в разпространението възниква между възлите. Специално внимание трябва да се обърне на забавянето на предаването (transmission delay), когато съвкупните данни от множество сензорни мрежи се разпространяват *през един възел***.** Забавянето на предаването може да се определи с уравнение 2.2.

$$D_{transmission\,(sec)} = Segment\ Length_{(bits)}/Rate_{(bits/\sec)}. \qquad (2.2)$$

2.3.6. Сравнение между MQTT и HTTP

MQTT е ориентиран към данните, а HTTP - към документи. HTTP е протокол за заявка-отговор, а *MQTT* използва модел публикуване-абониране. В сравнение с HTTP, *MQTT* е по-

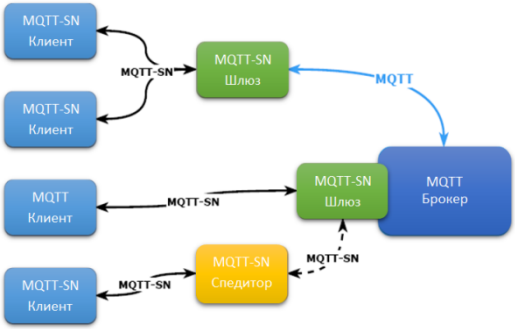
компактен (кратко заглавия на съобщението и най-малкият размер на пакетното съобщение от 2 байта) и позволява да се съставят дълги заглавия и съобщения. Основните разлики между протоколите са систематизирани в табл.2.5, [86]. [98], [99], [100].

Таб. 2.5 Разлики в *MQTT* и HTTP

Параметър	***MQTT***	**HTTP**
Име	Message Queuing Telemetry Transport	Hyper Text Transfer Protocol
Архитектура	Публикуване/абониране	Заявка/отговор.
Сложност	По-малка сложност	По-сложен
Работи върху	Transmission Control Protocol	User Datagram Protocol
Проектиране на протокола	Дизайнът на протокола е ориентиран към данните (Data centric).	Дизайнът на протокола е ориентиран към документи. (Document centric).
Размер на съобщението	Размера на съобщението е по-малък, поради двоичния формат. (binary format).	Създаденият размер на съобщението е по-голям, тъй като използва ASCII формат.
Размер на заглавната част	2 байта	8 байта.
Номер на порт	1883	порт 80 или 8080.
Сигурност на данните	Осигурява защита на данните със SSL/TLS.	HTTP без защита, но HTTPS има.

2.3.6. Протокол MQTT-SN

Протоколът *MQTT*-SN (Message Queue Telemetry Transport - Sensor Network) е версия на *MQTT*, адаптирана към *БСМ*, което го прави най-подходящ за сензорни устройства, поради техните компактни съобщения. *MQTT*-SN използва *UPD/IP,* защото е по-компактен в сравнение с TCP/IP. Архитектурата на протокола е показана на фиг.2.17. Най-важният MQTT-SN компонент е MQTT-SN шлюз (GW), позволяващ връзката на MQTT-SN клиентите с MQTT брокер.

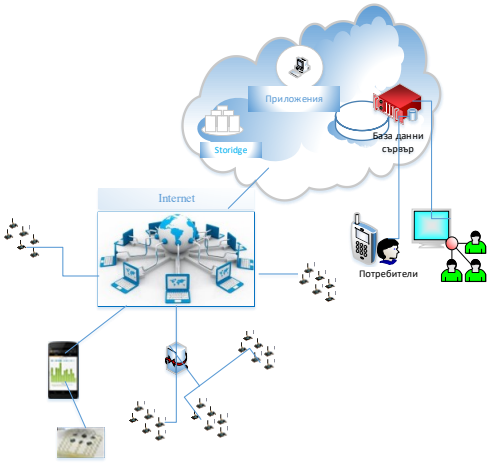


Фиг.2.17. Архитектура на *MQTT*-SN

2.4.Интеграция на сензорните данни в облачна структура

2.4.1.Същност на интеграцията [А6]

Свързването на сензорната мрежа с облачна структура решава проблема със съхранението и обработката на големите обеми данни от сензорните мрежи. Комуникацията между БСМ и облачни платформи се реализира чрез устройство като шлюз или координатор. На фиг.2.21 е представена блок-схема, илюстрираща интеграция на безжична сензорна мрежа с облачна структура.



Фиг.2.21. Интеграция на СМ към облачна структура [А6]

2.4.2. Сценарии за предаване на данните до облака

Данните може да се предават чрез: координатор или шлюз, директно предаване, до сървъра на облака. Известни са няколко начина на организация на достъпа до споделени ресурси - комуникационен канал и сървърни изчислителни

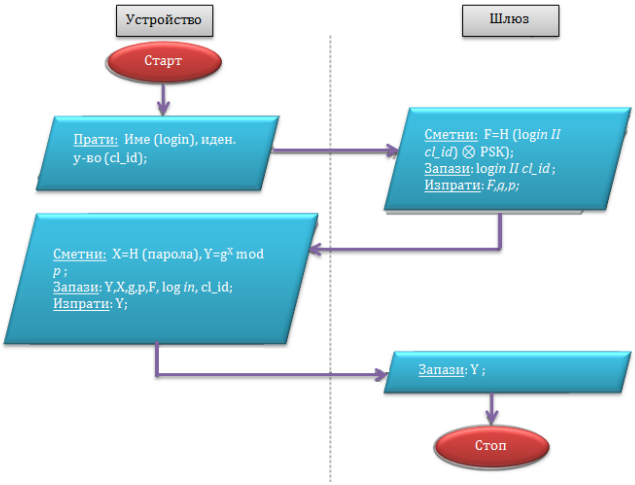
ресурси: запитване, прекъсвания и множествен достъп.

2.5. Подходи за управление сигурността на сензорните данни при интеграцията към облак

Към настоящият момент се прилагат различни подходи за управление сигурността на процеса на интегриране на сензорните данни към облак като: проектиране и разработване на мрежови контролери, разработване на международни стандарти за облачна сигурност, разработване на хардуерна защита.

2.5.5. Алгоритъм за удостоверяване достъпа до сензорни мрежи, базирани на MQTT протокол

Има два механизма за сигурност при *MQTT*. Единият е *удостоверяване* чрез потребителско име и парола, другият е *контрол на достъпа чрез ACL* (списък за контрол на достъпа/Access control list) файл или база данни. Паролата и потребителското име се прехвърлят в тялото на съобщението CONNECT (протоколът *MQTT* поддържа 16 типа съобщения). Ако удостоверяването е предадено, шлюзът отговаря съответно със съобщението CONNACK. В [123] се предлага използването на TLS за сигурност на удостоверяване между сензорите и шлюза. TLS се реализира на два етапа. Първият е генерирането на общ ключ за сесия на базата на сертификати от формат X509 чрез протокол DH (*Diffie-Hellman* или *ECDH* – Елиптична крива (*Elliptic curve) Diffie-Hellman*). Вторият е използването на алгоритъм за симетрично или асиметрично криптиране, напр. AES, RSA (*Rivest–Shamir–Adleman*). Най-важното предимство на този метод е пълното криптиране на връзката. За да се свърже чрез TLS протокол, устройството трябва да се свърже със шлюза по последния адрес, но с различен порт номер. Например, за незащитена връзка адресът ще бъде «TCP://127.0.0.1:1883», докато за връзка чрез TLS протокол - ще бъде «SSL://127.0.0.1:8883». По този начин се осъществява сигурна връзка чрез TLS. След това удостоверяването на шлюза се извършва чрез потребителско име и парола. Функционирането на алгоритъма за достъп има за цел да увери, че устройството знае паролата, без директно да я предава. Крайното устройство трябва да извърши процедурата по регистрация чрез защитен канал. Процедурата за регистрация е показана на фиг.2.26.



Фиг. 2.26. Процедура за регистрация на чрез защитен канал

Крайното устройство има стойности X, Y, F и публични g, p стойности. За да премине процедурата за удостоверяване, устройството генерира голямо произволно число R и изчисляваме:

$$T = gRmodp \quad (4) \qquad (2.17)$$

Използвайки стойностите на T и F, крайното устройство генерира ключ за сесия като:

$$K=F \otimes H(T) \qquad (2.18)$$

2.8. Изводи към втора глава

• Анализирани са основните модели за представяне и предаване на сензорни данни, специализирани бази от данни с времеви редове TSBD за сензорни данни и проблемите на интеграцията на сензорните данни към облачна структура стъпките за управление на сензорните данните

• Дефинирани са въпросите решавани в резултат на интеграцията между сензорната мрежа и -облачна структура.

• • Анализирани са моделите за съхранение на сензорни данни: Пет слоен модел базиран на OSI и три модела (файлово, блоково и обектно) за съхранение на данни в слоя за управление на данни.

• Разгледани са основните протоколи за комуникация и предаване на данните между сензорни мрежи и облачни платформи HTTP, HTTPS, REST, RESTful, MQTT и MQTT SN.

• Изяснена е същността на интеграцията между сензорна

мрежа и облака и определени подходите за управление сигурността на сензорните данни при интеграцията им към облак като: проектиране и разработване на специализирани мрежови контролери, разработване на международни стандарти, разработване на специализирана хардуерна защита.

• Дефинирани са основните режими на предаване на данните от сензорите към сървърите в облачната структура и анализирани възможностите за забавяне на предаването на данните.

• Представени са алгоритми за удостоверяване на комуникация, базирана на комуникационния протокол MQTT, който позволява да се провери автентичността на устройство без директно предаване на паролата, както и метода за защита на данните в MQTT чрез TLS протокол, чието приложение повишава нивото на сигурност на удостоверяване за сензорни мрежи.

• Систематизирани са методите за изследване на безжични сензорни мрежи, симулационни инструменти и предложена методика за провеждане на компютърно базирани симулационни изследвания.

ТРЕТА ГЛАВА

Изследване на някои възможности за интеграция на сензорните данни в облачни структури

3.1. Модел за изследване интегрирането на сензорни данни към облачна структура

Провеждането на експерименталните изследвания за изследване на възможностите за осъществяване на интеграция между сензорните мрежи и облачни структури е реализирано, съгласно предложения модел на фиг.3.1. Този модел е в съответствие със схемата за предаване на данни чрез гейтуей, подробно анализирана в работа [А4].

Физическият слой включва интелигентни сензори, които изпращат данните на микрокомпютъра. *Мрежовият слой* включва микрокомпютър, който играе ролята на шлюз и препраща данните към телекомуникационната базова станция. БС транспортира получените данни до облачната платформа, която съдържа сървърите за съхранение и обработка на данните. *Слоят за управление* на платформата осигурява функции за съхранение на данни и управление на устройства. *Слоят на услугата за приложения* е свързан с

облачната платформа чрез API, за да се реализира функцията за онлайн заявки за данни и дистанционно наблюдение.

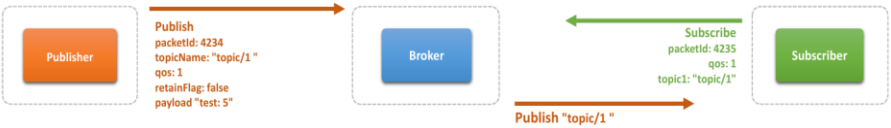
Фиг.3.1 Модел за интегриране на сензорни данни към облачна структура

3.2. Комуникационни модели на взаимодействие между сензорната мрежа и облачната структура

3.2.1. Комуникационният модел request-response, позволява на клиента да поиска информация от сървър, който получава съобщението за заявка, обработва го и връща съобщение за отговор. Двата най-известни протокола, базирани на модела са REST HTTP и CoAP. На фиг.3.2. е показано взаимодействие REST HTTP заявка/отговор между два клиента и един сървър.

Фиг.3.2. Комуникационен модел на взаимодействие отговор-заявка

3.2.2. Комуникационен модел публикуване-абониране publish/subscribe, осигурява комуникация между изпращачите на данни и дестинациите. Моделът на взаимодействие се състои от три страни: *издател, абонат и брокер*, фиг.3.3. Брокерът е софтуер, работещ на компютър и действа като пощенска станция, за устройство, изпълняващо роля на брокера, е необходимо да се инсталира MQTT брокерска библиотека.



Фиг.3.3. Модел на взаимодействие публикуване, брокер, абонат

3.3. Методология за интегрирането на сензорна мрежа

за събиране и споделяне на данни чрез използване на *Pub/Sub* метода *чрез REST базирани уеб услуги към облак ThingSpeak* [А3]

3.3.1. Мотивация

Целта е да се изгради физическа XBee сензорна мрежа с Digi XBee Cloud Kit и оцени възможността за интеграция на събраните сензорни данни и изпращането им в ThingSpeak облак, чрез REST *HTTP*.

3.3.2. Хардуерни компоненти

За интеграцията „сензор-облак“ се използва *Digi XBee Cloud Kit* [142]. Разработен е Python код в Digi XBee Gateway, позволяващ интеграцията на данни в ThingSpeak, позволяваща онлайн анализ и обработка на данните, чрез Matlab код.

Фиг.3.4. Digi XBee Cloud Kit

3.3.3.Архитектура на експерименталната мрежата

Фиг.3.6. Интегриране на сензорни данни в облак *ThingSpeak*

***3.3.4. Алгоритъм** за изграждане на сензорната мрежа и интегрирането и към облака*

***Първо:** Конфигурация на XBee модулите чрез* XBee ZigBee координатор.

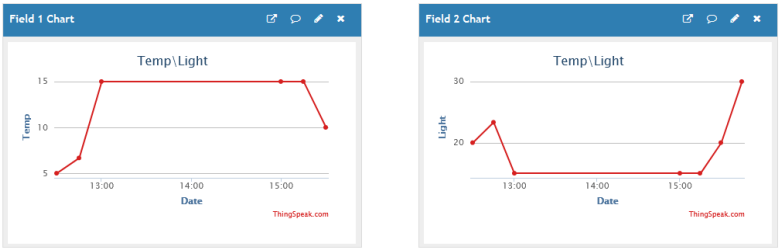
Второ: *Допълнителни мрежови и комуникационни настройки на устройствата* - тип на фърмуера, криптиране, PAN ID, присъединяване на възлите към мрежата и др.

Трето: *Изграждане на канал за предаване на сензорните данни в ThingSpeak, включва* дефиниране името на канала и API ключове на канала.

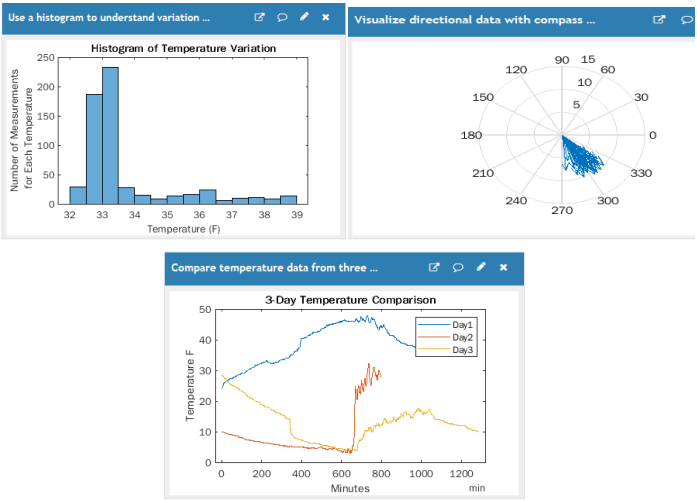
Четвърто: *Свързване на XBee мрежата към ThingSpeak чрез XBee Gateway и разработване код на Python за XBee Gateway за интегриране на данни в ThingSpeak. ThingSpeak* извлича данни от устройствата чрез *HTTP*.

Пето: *Събиране и изпращане на сензорните данни* към XBee Gateway чрез XCTU. От XBee Gateway те се изпращат по REST/HTTP до *ThingSpeak*.

Шесто: *Визуализация и анализ на данни,* с код на Matlab.



Фиг.3.10. Резултати от измерванията на данните в *ThingSpeak*



Фиг. 3.11. Опции за Matlab визуализация в *ThingSpeak*

Резултатите от визуализацията, показват, че предложения алгоритъм за интегриране на сензорните данни в облак,

осигурява онлайн достъп, мониторинг и анализ на постъпилите данни.

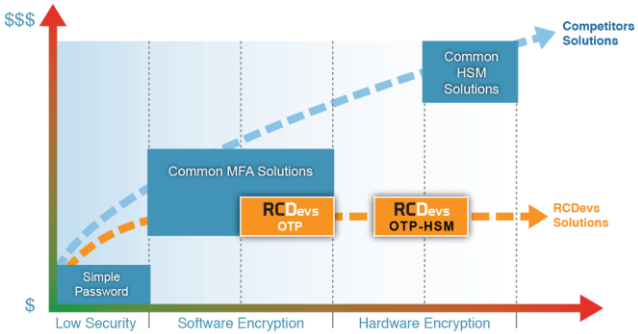
3.4. Експериментално изследване сигурността на предаваните сензорни данни в *ThingSpeak* облака [A2]

3.4.1. Анализ на проблема

Проблемите на сигурността на предаваните данни в сензорните мрежи са разгледани в т.2.6. Многобройните изследвания, обобщени в [137] показват, че сигурността на интеграцията чрез REST/HTTP и MQTT, които са обект на изследванията е най-целесъобразно да се извършва чрез TLS/SSL.

3.4.2. Избор на протокол за сигурност

Изборът на този протокол се определя от възможностите на избраната хардуерна реализация. На фиг.3.13 е показан резултатът от изследване на компанията *RF Desk security solution* [143].

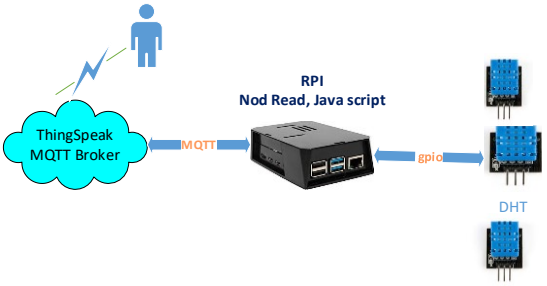


Фиг.3.13. Избор на решение за сигурност на данните [143]

Избираме *протокол TLS*, тъй като защитава данните, които се предават *онлайн между уеб браузър и уебсайт чрез HTTPS* и защото гарантира, че данните запазват първоначалната си цялост и осигурява конфиденциалност чрез криптиране.

3.4.3. Експериментална установка за изследване на сигурността

Провеждането на експеримента е реализирано по моделите съгласно фиг.3.1 и фиг.3.3. Експерименталната установка е изградена, съгласно блок схемата на фиг. 3.14.

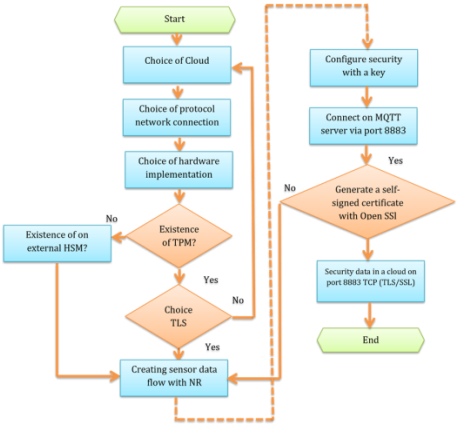


Фиг.3.14. Експериментална установка за изследване сигурността

3.4.4. ***Алгоритъм за изследване сигурността на предаваните данни от сензорна мрежа към облак***

Първо: *Избор на облак.* Данните се изпращат в реално време с *ThingSpeak* , което дава възможност за анализ и изпращане на данни в реално време с Open API (HTTP или MQTT).

Второ: *Избор на протокол за връзка на мрежата с облака и предаване на данни.* Избира се *MQTT*, тъй като е разработен за свързване на устройства с ограничена мощност през безжични мрежи, използва модел *публикуване/абониране* и брокер за предаване на кратки съобщения.



Фиг.3.15. Алгоритъм за проверка TLS сигурността на предаваните сензорните данни

Трето: *Избор на протокол за сигурност в зависимост от хардуерната реализация.* Съгласно съображенията в т.3.4.2 е избран протокол TLS.

Четвърто: *Създаването на поток за събиране на данни, става чрез MQTT* и *Node-RED.*

Пето: *Свързването към MQTT брокер/сървър и публикуване с помощта на възела за публикуване, се*

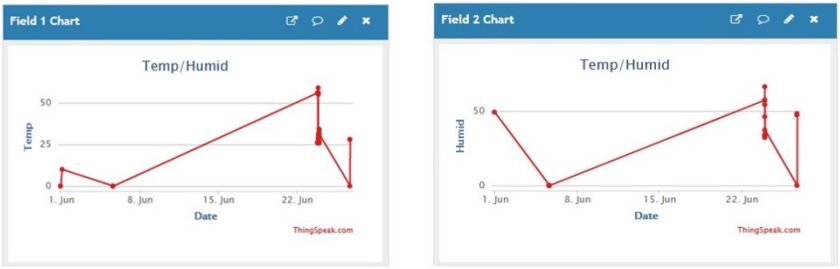
извършва чрез въвеждане на *IP адреса/името на Broker и порта, който се използва* (8883). *MQTT* брокерът предлага криптиране и удостоверяване на потребителско име/парола, фиг.3.17.

Шесто: *Настройка на сигурността*

- *Генериране на само подписан сертификат.* За целта може да се създаде собствен (*self-signed*) сертификат, използвайки *Open-SSL*.
- *Защита в облака:* За конфигурирането на *MQTT* клиента и осъществяването на комуникация с брокера *ThingSpeak MQTT*, е нужна една от опциите за връзка, показани в таблица 3.2.

Таб.3.2. *ThingSpeak MQTT* брокер, опции за комуникация

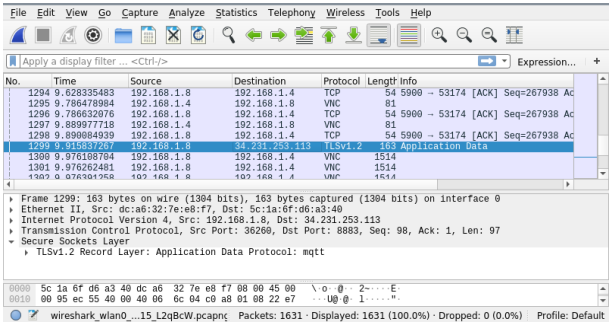
Port	**Connection Type**	**Encryption**
1883	*TCP*	*None*
8883	*TCP*	*TLS/SSL*
80	*WebSocket*	*None*
443	*WebSocket*	*TLS/SSL*



Фиг.3.18. Резултат от интегрирането на сензорните данни в *ThingSpeak*

Седмо: *Проверка работоспособността на алгоритъма за предаване на криптирани данни.*

Работоспособността на предложения алгоритъма и реално постигната сигурност на данните е проверена чрез тестване на производителността на мрежата Wireshark софтуера. Той улавя мрежовия трафик в локалната мрежа и съхранява данни за офлайн анализ от Ethernet, Bluetooth, Wireless (IEEE.802.11), Token Ring, Frame Relay връзки и други. Резултата от проверката е показана на фиг.3.19 и потвърждава, че пакетите данни, които се изпращат от сензорната мрежа до *ThingSpeak* чрез *MQTT* протокол за пренос на данни се свързват към комуникационен порт 8883, който е защитен/криптиран чрез *TLS/SSL* протокол.



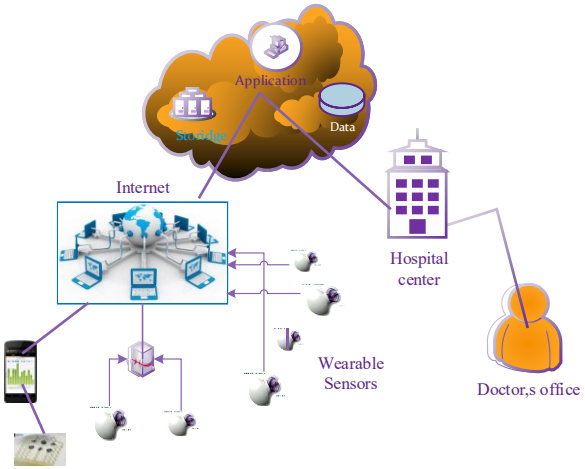
Фиг.3.19. Резултатите от анализирането на пакети с Wireshark

3.5 Моделиране и изследване на сензорна мрежа за измерване на инсулинови нива [A1]

3.5.4. Модел на сензорна мрежа за дистанционна здравна помощ, чрез мониторинг на диабетното състояние, базиран на технологията IoT

3.5.4.2. Архитектура на сензорна мрежа за измерване на глюкозния индекс

Предложената архитектура за сензорната мрежа е облачно базирана. Фиг.3.21 показва директно свързване на сензорите към облака или през шлюз. Генерираните данни от устройствата могат да бъдат транспортирани до регионален или глобален сървър за обработка.



Фиг.3.21. Архитектура на сензорна мрежа за измерване на глюкозен индекс

3.5.4.3. Модел на сензорна мрежа за измерване на глюкозен индекс

Здравният модел разглежда абстрактен случай с пациент с диабет. Пациентът използва здравна компания за мониторинг (health monitoring company/HMC), за да избегне диабетна кома. Той носи устройство за непрекъснато наблюдение на глюкозата (continuous glucose monitoring/CGM). Когато данните за здравето на пациента показват опасни модели (влошаване показателите на гликозен индекс), HMC изпраща сигнал на интелигентното устройство на пациента да се свърже със здравния център. В случаите, когато след изпълняване на указанията дадени от HMC няма промяна в състоянието на пациента, се предвижда изпращане на мобилен екип за спешна помощ. Моделирането на IoT прототипа на потенциалните взаимодействия и от типа M2M, M2P и P2P.

3.5.4.4. ***Алгоритъм за проектиране на модел за сензорна мрежа*** *за наблюдение на инсулиновите нива, базиран на IoT със симулационен софтуер*

Първа стъпка: Конфигуриране на основни мрежови настройки. IP адресиране на сензорите и шлюза.

Втора стъпка: Свързване на устройствата с микрокомпютър. Устройствата се свързват към домашен шлюз и към IoE Регистрационния Сървър.

Трета стъпка: Конфигуриране на шлюз за свързване на устройствата в мрежа. Свързване на IoT устройствата към изграждане на домашната мрежа за пациента.

Четвърта стъпка: Създаване на мрежа и Интернет доставчик. Създаване на Сървъри.

Пета стъпка: Създаване на математически обекти за дефиниране на гликозните нива в кръвта и интегриране в кода на приложението.

3.5.4.5. Тестване на системата

Проектираната система е изградена от интелигентни устройства с глюкозен монитор (CGM), отчитащ нивата на глюкозата. Устройствата изпращат безжично информация до домашния шлюз и след това до компания за здравни грижи. Компанията може да изпрати медицинска помощ, ако състоянието на пациента стане животозастрашаващо.

А) Симулиране на Хипергликемия. Приложението изпраща сигнал до IoT устройствата. Изпраща се също съобщение до сървъра на медицинския център, където данните се анализират и при нужда се изпраща бърза помощ.

Фиг.3.24. Симулиране на хипергрликемия

Б). Симулиране на Хипогликемия. Приложението изпраща сигнал до IoT устройствата, които показват нивата на глюкозата. Данни, получени при хипогликемия се наблюдават на фиг.3.25.

В). Симулиране на нормални нива. Симулира се нормално ниво на глюкозата, степента на дишане, ниво на физическо натоварване и др. са визуализиране на фиг.3.26.

Фиг.3.25. Симулиране на Хипогликемия

Фиг.3.26.Симулиране нормални нива на кръвната захар

Данните от глюкозния сензор се предават на „умните устройствата“ на пациента през MQTT протокол като за целта е разработен код на Python за *publisher/subscriber* модела,

чийто настройки брокера и клиента са показани на фиг.3.28 и 3.29 в дисертацията.

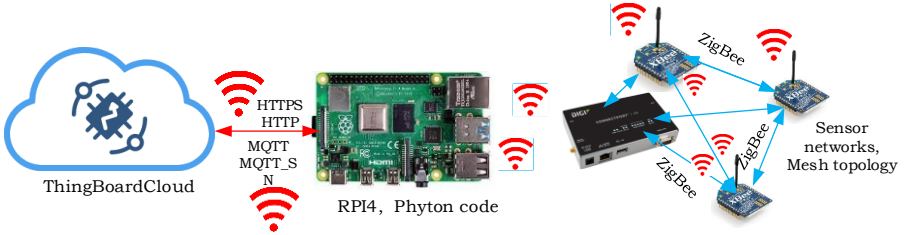
Резултатите от симулационните изследвания, показват, че в проектираната система могат да бъдат следени и управлявани: инсулиновите нива, дихателната честота, степента на натоварване и др. Предложеният алгоритъм и методиката за конфигурирането и проектирането на предложената симулационна система, дават възможност за придобиване на цялостната картина за състоянието на пациента. Внедряването на този модел дава възможност за наблюдаване здравословното състояние на пациентите в реално време, оценка за състоянието и вземане на решения относно здравния статус на пациента.

3.5.Експериментално изследване влиянието на протокола върху интеграцията на сензорните данни към облака

3.6.1.Целта на експеримента е да направи оценка на интеграцията, чрез изследване влиянието на протоколите HTTPS, HTTP, MQTT и MQTT-SN за интеграция на сензорните данни от мрежата към облак. Оценката на интеграцията може да се извърши по различни критерии като: поддръжка на приложения, работещи с различни комуникационни модели и осигуряващи различна скорост на данните. Общото изискване е *надеждно предаване на данните*. За параметър на ефективността на интеграцията *приемаме закъснението на предаваните данни*, *натоварването на CPU и RAM*, показващи степента на изразходваната енергия.

3.6.2. Променливите параметрите са: брой пакети, брой теми, байтова стойност за тема и тяхното влияние върху закъснението между генерирането на сензорните данни и получаването им в облака, породено от преноса на съобщението до брокера/сървър.

3.6.3. Дизайн на експеримента



Фиг.3.31. Дизайн на експеримента

Изградена е физически XBee сензорна мрежа. Събраните

данни се предават на RPI4, в който се зарежда разработения код за провеждане на експеримента. RPI4 предава сензорните данни към облака ThingBoard по протоколи MQTT, HTTP, HTTPS и MQTT-SN.

Облакът ThingBoard не е проектиран за достъп на данни от MQTT-SN. За предаването на сензорни данни по MQTT-SN се използва библиотека *MQTTSNClient Eclipse Paho*, отразяваща начина, по който MQTTClient работи. MQTT-SN изисква MQTT-SN Gateway, който действа като преобразувател на протоколи и преобразуване на MQTT-SN съобщения в MQTT съобщения. Тези дейности изискват преобразуване на данните. Директорията *MQTTSNPacket* съдържа библиотека, осигуряваща процедури за преобразуване на данни чрез сериализация и десериализация. Сериализацията е процес на преобразуване на структури от данни или обекти до поток от байтове, запазвайки състоянието на техните свойства.

3.6.4. Алгоритъм и код за провеждане на изследването

Първа стъпка: *Настройка на сензорните възли в „броудкаст“ режим за предаване на данните в меш мрежа и към RPI4.*

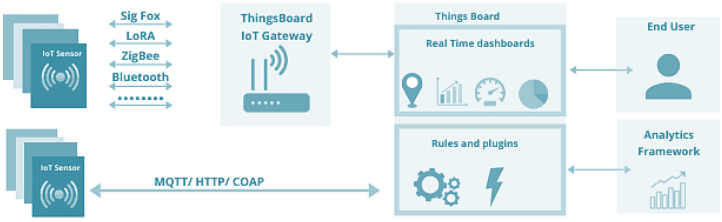
Втора стъпка: Разработване на кода за работа на RPI за събиране на сензорните данни.

Трета стъпка: Задаване на параметрите за провеждане на изследването по различни сценарии: брой предавани пакети, брой теми, байтова стойност на темите

Четвърта стъпка: връзка с облака, генериране на данните и постъпване в базата данни за съхранение.

Пета стъпка*:* анализ и визуализация на данните в облака.

3.6.5. Настройка на експеримента включва настройка на облака, настройка на XBee модулите, настройка на протоколите за интеграция MQTT и HTTP и внедряване на HTTP и MQTT клиент. За провеждане на експеримента е нужна регистрация в ThingBoard, който приема HTTP и MQTT връзки. Регистърът насочва съобщенията на сензорите към една тема Pub/Sub, която има една крайна точка на *Cloud Functions* като абонат. Cloud функцията просто записва полезния товар в журнала (*payload to log*). Крайното устройство работи с MQTT и HTTP клиенти, като измерва времето за реакция и проследява изпратените пакети и скоростта, фиг. 3.33.



Фиг.3.33. ThingBoard Архитектура

3.6.6. Експериментално изследване влиянието на параметрите на пакетите върху закъснението на предаваните данни.

Целта на изследването е да се определи влиянието на различни параметри като брой пакети, брой теми в пакетите и байтова стойност за всяка тема върху скоростта (закъснението) на предаваните данни към облака през различни протоколи, работещи по различен комуникационен модел като: HTTP, HTTPS, MQTT и MQTT-SN и оцени ефективността на интеграцията между мрежата и облак.

3.6.6.1.Сценарии за провеждане на експериментите

Табл. 3.3. Сценарии

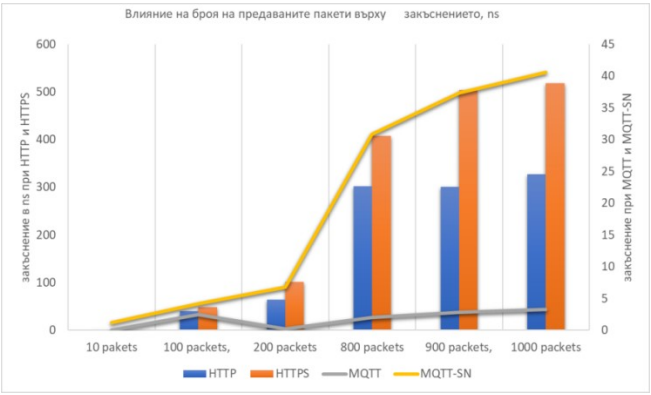
Сценарий	Изследвани протоколи	Брой пакети	Брой теми в пакетите	Байтове за тема
1.	HTTP, HTTPS,MQTT MQTT-SN	10	100	10
2.	HTTP, HTTPS,MQTT MQTT-SN	100	100	10
3.	HTTP, HTTPS,MQTT MQTT-SN	200	100	10
4.	HTTP, HTTPS,MQTT MQTT-SN	800	100	10
5.	HTTP, HTTPS,MQTT MQTT-SN	900	100	10
6.	HTTP, HTTPS,MQTT MQTT-SN	1000	100	10
7.	HTTP, HTTPS,MQTT MQTT-SN	200	200	1400
8.	HTTP, HTTPS,MQTT MQTT-SN	200	200	1800
9.	HTTP, HTTPS,MQTT MQTT-SN	200	200	2000
10.	HTTP, HTTPS,MQTT MQTT-SN	200	200	2500
11.	HTTP, HTTPS,MQTT MQTT-SN	200	400	2500
12.	HTTP, HTTPS,MQTT MQTT-SN	400	600	2500
13.	HTTP, HTTPS,MQTT	600	600	800

	MQTT-SN			
14.	HTTP, HTTPS,MQTT MQTT-SN	1000	100	100
15.	HTTP, HTTPS,MQTT MQTT-SN	200	1000	100
16.	HTTP, HTTPS,MQTT MQTT-SN	200	800	100
17.	HTTP, HTTPS,MQTT MQTT-SN	200	400	100

Експеримента включва 17 сценария. Предаването на данните се извършва по четири протокола HTTP, HTTPS, MQTT, MQTT-SN като параметрите на експерименталните данни са посочени в *приложение 3.4* на дисертацията.

6.6.6.2. *Изследване влиянието на броя на предаваните пакети върху закъснението на доставяните данни при различни протоколи.*

Изследванията се извършват съгласно сценарии №1 до №6, при 100 броя на темите в пакет и 100 байта за тема.

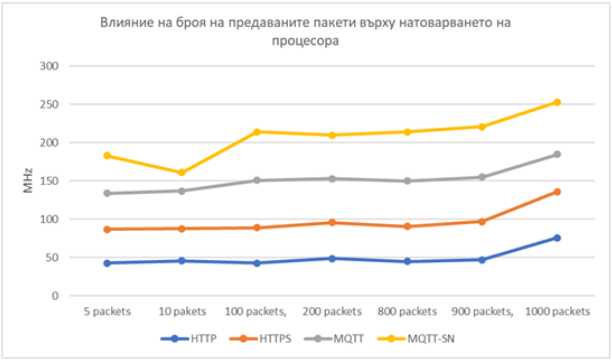


Фиг.3.34. Влияние на броя на предаваните пакети върху закъснението

Анализът на фиг.3.34 показва, че при увеличаване броя на пакетите, най-голямо е закъснението при HTTPS, а най-малко при MQTT. Закъснението на MQTT-SN (UDP) се дължи на това, че не предава данните директно на облака, а ги преобразува чрез MQTT-SN Gateway, така че да са съвместими с MQTT (TCP).

6.6.6.3.Изследване влиянието на броя на предаваните пакети върху CPU натоварването

Изследванията се извършват съгласно сценарии №1 до №6, при 100 броя на темите в пакет и 100 байта за тема.

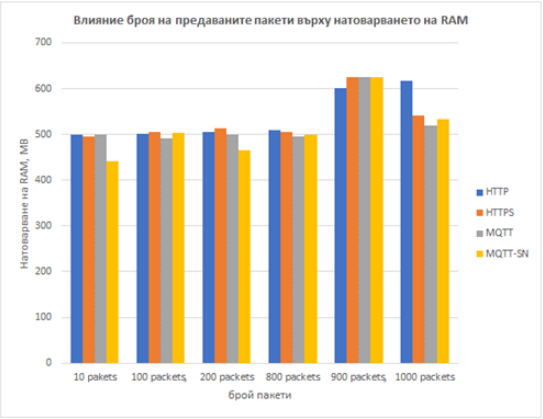


Фиг.3.35. Влиянието на броя на пакетите върху CPU натоварването

Графиката на фиг.3.35 показва, че натоварването на процесора при MQTT-SN е най-високо, поради описаните особености по-горе.

6.6.6.4. Изследване влиянието на броя на предаваните пакети върху RAM

Изследванията се извършват съгласно сценарии №1 до №6, при 100 броя на темите в пакет и 100 байта за тема.

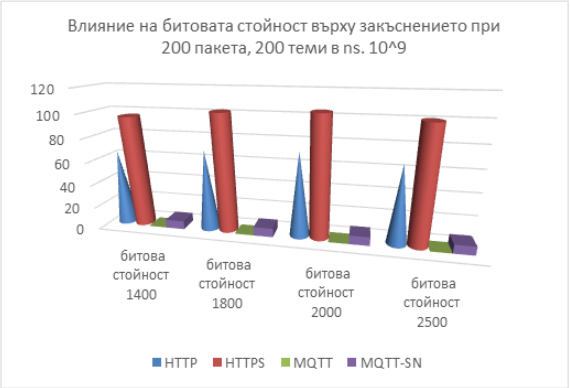


Фиг.3.36. Влиянието на броя на предаваните пакети върху RAM

Анализът на фиг.3.36 показва, че с нарастване броя на пакетите, нараства и натоварването върху RAM. Най-голямо натоварване се наблюдава при HTTPS протокола, поради използването на TLC защита.

6.6.6.5. Изследване влиянието на битовата стойност върху закъснението

Изследванията се извършват съгласно сценарии от №7 до №10, при 200 броя на темите в пакет и 200 байта за тема.

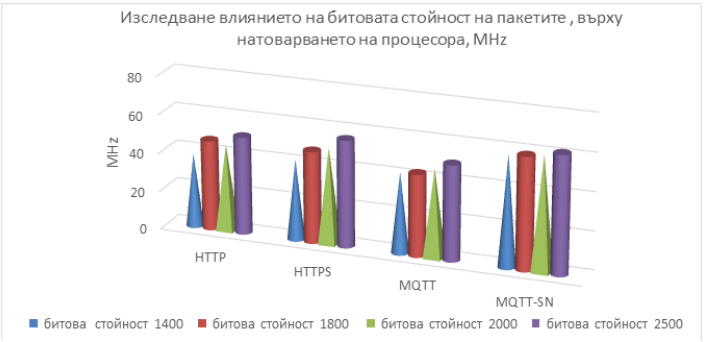


Фиг.3.37. Влиянието на байтовата стойност върху закъснението

Най голямо е влиянието при HTTPS поради осигуряването на TLC защита, а най малко при MQTT.

6.6.6.6. Изследване влиянието на битовата стойност върху натоварването на процесора

Изследванията се извършват съгласно сценарии от №7 до №10, при 200 броя на темите в пакет и 200 байта за тема.

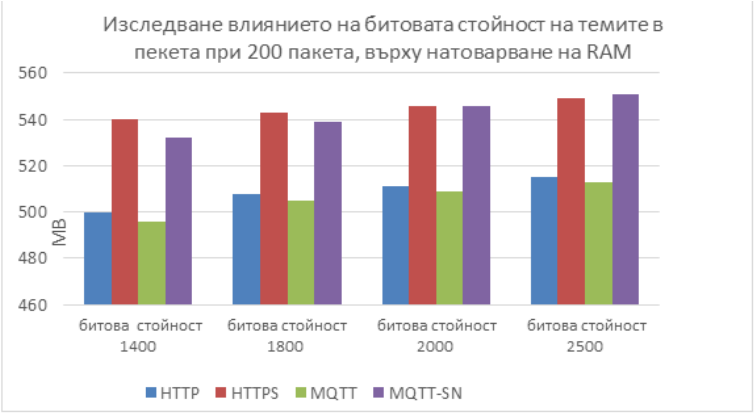


Фиг.3.38. Влиянието на байтовата стойност върху CPU натоварването

На фиг.3.38 Увеличената битова стойност на пакетите влияе най-много на MQTT-SN, поради вече описаните свойства.

6.6.6.7. Изследване влиянието на битовата стойност на темите в пакета върху натоварването на RAM

Изследванията се извършват съгласно сценарии от №7 до №10, при 200 броя на темите в пакет и 200 байта за тема.

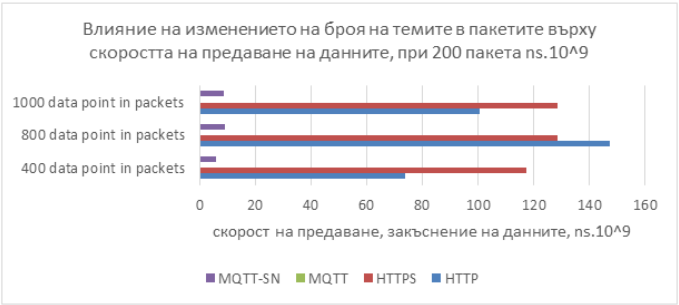


Фиг.3.39. Влиянието на байтовата стойност на темите върху RAM натоварването

На фиг.3.39 е видно, че най-много се натоварва паметта от работата на протокола MQTT-SN,а най-малко от MQTT.

6.6.6.8 Изследване влиянието на броя на темите в пакетите върху закъснението на предаваните данни.

Изследванията се извършват съгласно сценарии от №15, №16, №17 при 200 броя на темата в пакетите и 100 байтова стойност.

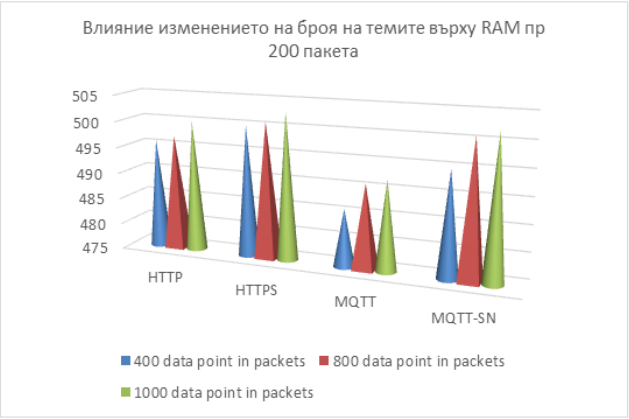


Фиг.3.40. Влиянието на броя на темите върху закъснението на предаваните данни

При увеличаване броя на темите в пакетите закъснението е най-голямо при HTTPS, а броя на темите при MQTT има пренебрежимо малко влияние на закъснението.

6.6.6.9. Изследване влиянието на броя на темите в пакетите върху натоварването на процесора.

Изследванията се извършват съгласно сценарии от №15, №16, №17 при 200 броя на темата в пакетите и 100 байтова стойност.

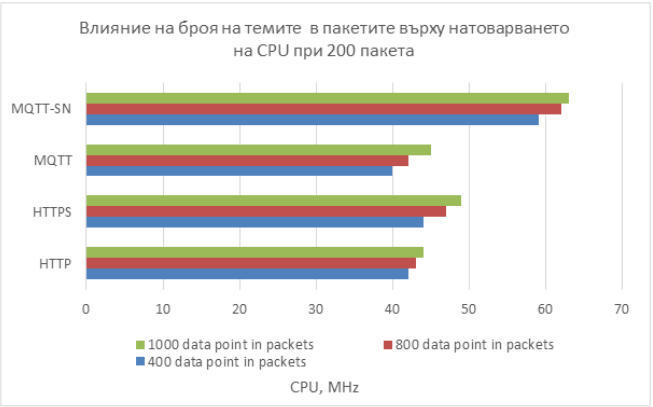


Фиг.3.41. Влиянието изменението на броя на темите върху RAM

На фиг.3.41 натоварването на паметта е по-високо при протоколите MQTT-SN и HTTPS, а най малко се натоварва паметта при MQTT.

6.6.6.10. Изследване влиянието на броя на темите в пакетите върху натоварването на RAM

Изследванията се извършват съгласно сценарии от №16, №17, №18 при 200 броя на темата в пакетите и 100 MB байтова стойност.



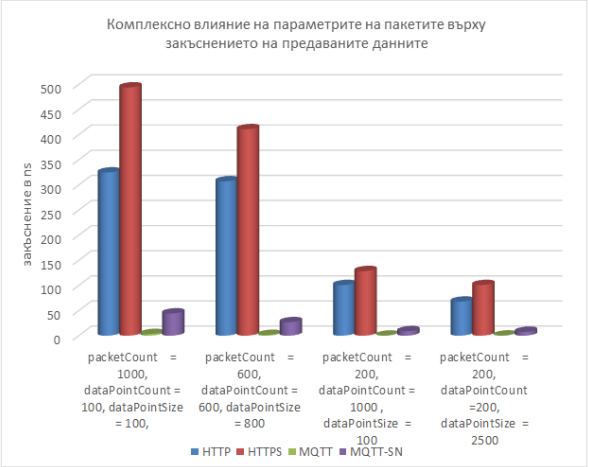
Фиг.3.42. Влиянието на броя на темите върху CPU натоварването

Натоварването на CPU е най-голямо при MQTT-SN, поради вече описаните свойства. Забелязва се че при HTTPS има по-голямо натоварване, поради TLC защита. Практически най-малко се натоварва CPU при MQTT.

6.6.6.11. Изследване на комплексното влияние на параметрите на пакетите върху закъснението на данните

Изследванията се извършват съгласно сценарии от №11, №14, №15 и №16.

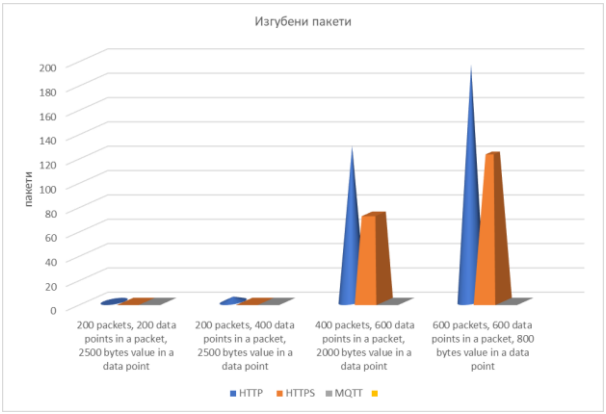
С увеличаване броя на пакетите се увеличава закъснението при предаването на данните, фиг.3.43. Отчитайки едновременното влияние на трите параметъра, се вижда че при MQTT закъснението почти не се променя, а MQTT-SN е на второ място с минимално влияние върху закъснението. Може да се обобщи, че най голямо е закъснението при HTTPS.



Фиг.3.43. Комплексното влияние на параметрите на пакетите върху закъснението на предаваните данните

6.6.6.12.Изследване влиянието на параметрите на данните върху загубените пакети при HTTP и HTTPS

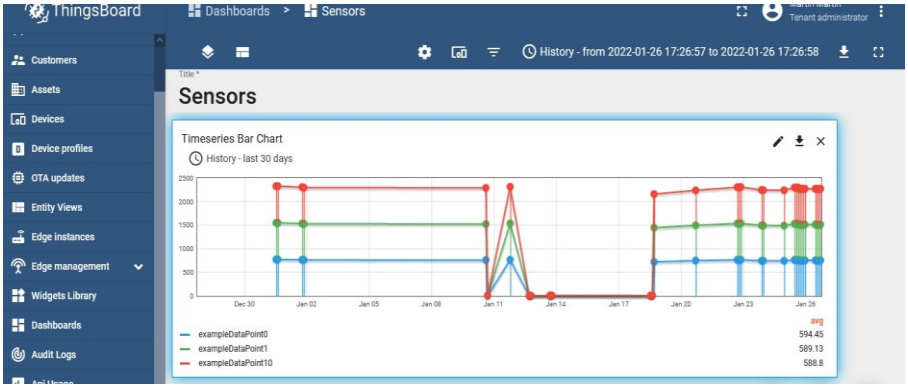
Изследванията се извършват съгласно сценарии от №11 до №14



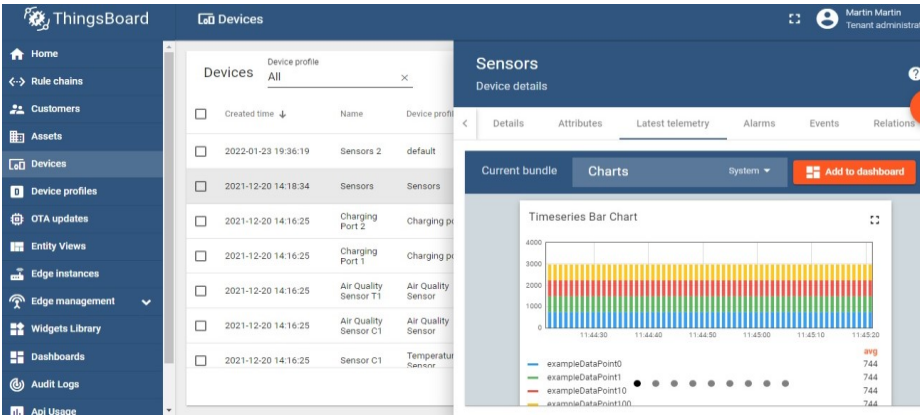
Фиг.3.44. Влиянието на параметрите на данните върху загубените пакети

Едновременното увеличаване на пакетите, броя на темите и битовата им стойност води до най-голяма загуба на пакети при HTTPS.

6.6.9. Визуализация на данните в облачната структура
Визуализирането на данните за потребителите се реализира чрез вградени възможности на облака. Една от възможности е исторически график на качваните данни, фиг.3.53. Друга възможност е наблюдаването в реално време на данните в облака, фиг.3.54.



Фиг.3.53. Исторически график на качените данни за определен период



Фиг.3.54. Наблюдение в реално време на качваните данни в облака

Сравнителният анализ от проведените експериментални изследвания показва, че MQTT осигурява най-малко закъснение при предаване на данните, най-малко натоварва процесора и паметта, респективно изисква най-малко консумация на енергия и най-кратко време за предаване на данните.

HTTPS и MQTT-SN, работят с по-голямо закъснение, по-

голям разход на енергия, поради допълнителните обработки, които извършват.

Оценявайки резултатите на проведените експерименти за интеграция, най-ефективен от гледна точка на скорост на предаване на данните и разход на енергия е MQTT.

3.7. Изводи към трета глава

- Изградена е реална сензорна мрежа от XBee модули. Данните се визуализират, анализират и изпращат по метода *request-response* чрез REST базирани уеб услуги и протокола HTTPS към облак ThingSpeak.
- Предложен и практически реализиран е алгоритъм за интегриране на сензорни данни в облак чрез REST базирани уеб услуги.
- Разработени са код за интегриране на данни в ThingSpeak чрез Phyton и код за визуализация и анализ на сензорните данни чрез Matlab в ThingSpeak.
- Аргументиран е избора на протокол TLS за гарантиране сигурността на предаваните данни. Предложен алгоритъм за предаване на сензорни данни чрез TLS в облак ThingSpeak. Интеграцията между сензорната мрежа и облака е чрез протокола MQTT, а потока за данни е създаден чрез софтуер Node-RED.
- Верифицирана е работоспособността на алгоритъма сигурност на предаваните данни чрез инструмента Wireshark, която потвърждава, че изпращаните пакети данни от сензорната мрежа до облака преминават през криптиран комуникационен порт и протокола TLS.
- Предложен модел на сензорна мрежа за дистанционна здравна помощ, чрез мониторинг на диабетното състояние, базиран на технологията IoT.
- Предложен и реализиран е алгоритъм за проектиране на модела и конфигуриране на симулационна система за интеграция на сензорните данни от биосензорите за наблюдение на инсулиновите нива със симулационен софтуер.
- Работата на предложения модел е верифицирана чрез симулиране на хипергликемия, хипогликемия и нормални нива на кръвната захар. Сензорните данни се предават успешно от умните устройства на пациента до симулационния сървър за анализ и вземане на решения.
- Проектирана и практически реализирана е измервателна система от XBee устройства и облачна структура ThingBoard Cloud за изследване влиянието на различните протоколи за интеграция на сензорна мрежа

към облак и направена оценка ефективността на предаване на сензорните данни.

- Предложен е модел за изследване на интеграцията сензорна мрежа- облак с различни протоколи. Анализирани са основни комуникационни модели за интеграция- *request-response и publish-subscribe* и разработена методология за осъществяване на интеграцията сензорна мрежа- облак
- Разработен и практически приложен е алгоритъм и код на Python за експериментални изследвания на протоколите HTTP, HTTPS, MQTT и MQTT-SN и верифицирана работоспособността им чрез Wireshark.
- Планувано и проведено е експериментално изследване, което включва 17 сценария, за определяне работоспособността HTTP, HTTPS, MQTT и MQTT-SN в зависимост от различни параметри като брой пакети, брой теми в пакетите и битове за всяка тема върху скоростта (закъснението) на предаваните данни към облака.
- Установено е, че протоколът MQTT осигурява най-малко закъснение при предаване на данните, най малко натоварва процесора и паметта, респективно изисква най-малко консумация на енергия и най кратко време за предаване на данните.
- Установено е, че HTTPS и MQTT-SN, работят с по-голямо закъснение, по-голям разход на енергия и натоварване на паметта, поради допълнителните обработки. HTTPS използва TLS протокол за сигурност, което забавя предаването на данните. MQTT-SN реализира процес на сериализация като използва MQTT-SN Gateway за преобразуване на структури от данни или обекти до поток от байтове, запазвайки състоянието на техните полета и свойства, което обяснява по-голямото закъснение.
- Оценявайки резултатите на проведените експерименти за интеграция между сензорна мрежа и облак най-ефективен от гледна точка на скорост на предаване на данните и разход на енергия е протокол MQTT.

- Получените резултати дават полезна и практически приложима информация, относно ефикасността на предаваните данни чрез четири от най-популярните протоколи за интеграция на сензорни данни HTTP, HTTPS , MQTT и MQTT-SN към облачна структура.

III. ЗАКЛЮЧЕНИЕ И ОСНОВНИ ПРИНОСИ

За изпълнение на поставената цел и задачи на дисертационната работа са проведени теоретични, симулационни и практически изследвания и са постигнати резултати, които могат да бъдат обобщени по следния начин:

- ✓ Проучени и систематизирани са основните направления, върху които са провеждани изследвания, дефинирани са изискванията и предизвикателствата, стоящи пред безжичните сензори и БСМ;
- ✓ Анализирана е структурата на сензорните мрежи и определени спецификата на приложение на топологиите звезда, дървовидна, меш, хибридна и клъстерна топология при изграждане и приложението на сензорните мрежи;
- ✓ Планувано и преведено е експериментално изследване на комуникационна ефективност за сензорни мрежи с различни топологии и XBee модули.
- ✓ Предложен е алгоритъм за изследване комуникационна ефективност на сензорни мрежи при различни топологии. Резултатите показват, че най ефективна комуникация има при клъстерна топология.
- ✓ Проведено е симулационно изследване за енергоефективността на сензорни мрежи с клъстерна топология и протоколи LEACH, LEACH-N и LEACH-H, което установява по-високата енергийна ефективност на LEACH-N.
- ✓ Няма универсална методика и инструментариум за изследване и оценка на параметрите на сензорните мрежи, поради което е необходимо разработване на нови модели, подходи и алгоритми за провеждане на експериментални изследвания.
- ✓ Важен проблем при сензорните мрежи е генерирането на огромно количество сензорни данни, ограничената изчислителна мощност и ограничения капацитет на сензорните мрежи, и живота на мрежата. Интеграцията между облачните структури и сензорните мрежи дава възможност за решение на този проблем, без излишно оскъпяване себестойността на сензорните мрежи.
- ✓ Формулирана е тезата, че създаването на модели и алгоритми за интеграция на сензорните данни към облак, позволява да се съкрати времето за разработване и оцени ефективността на сензорните

мрежи, да се осигури по-висока надеждност и енергийна ефективност на проектираните системи, което е актуален проблем в областта на сензорните мрежи.

✓ Анализирани са основните модели за представяне и предаване на сензорни данни, специализираните бази от данни с времеви редове TSBD за сензорни данни и проблемите на интеграцията на сензорните данни към облачна структура.

✓ Анализирани са моделите за съхранение на сензорни данни: Пет слоен модел базиран на OSI и три модела за файлово, блоково и обектно съхранение на данни в слоя за управление на данни.

✓ Анализирана е работата на основните протоколи интеграция на сензорни мрежи и облачни платформи, HTTP, REST, RESTful, MQTT и MQTT-SN и систематизирани разликите между тях.

✓ Изяснена е същността на интеграцията между сензорна мрежа и облака и определени подходите за управление сигурността на сензорните данни при интеграцията им към облак като: проектиране и разработване на специализирани мрежови контролери, разработване на международни стандарти, разработване на специализирана хардуерна защита.

✓ Дефинирани са основните режими на предаване на данните от сензорите към сървърите в облачната структура и анализирани възможностите за забавяне на предаването на данните.

✓ Представени са алгоритми за удостоверяване на комуникация, базирана на комуникационния протокол MQTT, който позволява да се провери автентичността на устройство без директно предаване на паролата, както и метода за защита на данните в MQTT чрез TLS протокол, чието приложение повишава нивото на сигурност на удостоверяване за сензорни мрежи.

✓ Изградена е реална сензорна мрежа от XBee модули по метода request-response чрез REST базирани уеб услуги и протокола HTTPS към облак Things Speak и разработен и реализиран алгоритъм за интегриране на сензорни данни в облак чрез REST базирани уеб услуги.

✓ Разработени са код за интегриране на данни в ThingSpeak чрез Python и код за визуализация и анализ на сензорните данни чрез Matlab;

✓ Изследвана и експериментирана е сигурността на

интеграцията на предаваните данни. Предложен е алгоритъм за проверка на сигурността на предаваните сензорни данни чрез протокола MQTT в облак ThingSpeak, чрез протокол TLS като потока данни е създаден чрез софтуер Node-RED.

✓ Верифицирана е работоспособността на алгоритъма за сигурност чрез инструмента Wireshark, която потвърждава, че изпращаните пакети данни от сензорната мрежа до облака преминават през криптиран комуникационен порт и протокола TLS.

✓ Предложен модел на сензорна мрежа за дистанционна здравна помощ, чрез мониторинг на диабетното състояние, базиран на технологията IoT.

✓ Предложен и реализиран е алгоритъм за проектиране на модела и конфигуриране на симулационна система за интеграция на сензорните данни от биосензорите за наблюдение на инсулиновите нива със симулационен софтуер.

✓ Работата на проектирания модел е тествана чрез симулиране на хипергликемия, хипогликемия и нормални нива на кръвната захар. Сензорните данни се предават успешно от умните устройства на пациента до симулационния сървър за анализ и вземане на решения.

✓ Проектирана и практически реализирана е експериментално измервателна система от XBee устройства и облачна структура ThingBoard Cloud за изследване влиянието на различните протоколи за интеграция на сензорна мрежа към облак и оценка ефективността на предаване на сензорните данни.

✓ Предложен е модел за изследване на интеграцията сензорна мрежа- облак с различни протоколи Анализирани са основни комуникационни модели за интеграция- request-response и publish-subscribe и разработена методология за осъществяване на интеграцията сензорна мрежа- облак

✓ Разработен и практически приложен е алгоритъм и код на Python за експериментални изследвания на протоколите HTTP, HTTPS, MQTT и MQTT-SN и верифицирана работоспособността им чрез Wireshark.

✓ Планувано и проведено е експериментално изследване, което включва 17 сценария, за определяне работоспособността HTTP, HTTPS, MQTT и MQTT-SN в зависимост от различни параметри като брой пакети,

брой теми в пакетите и битове за всяка тема върху скоростта (закъснението) на предаваните данни към облака.

✓ Установено е, че протоколът MQTT осигурява най-малко закъснение при предаване на данните, най малко натоварва процесора и паметта, респективно изисква най-малко консумация на енергия и най кратко време за предаване на данните.

✓ Установено е, че HTTPS и MQTT-SN, работят с по-голямо закъснение, по-голям разход на енергия и натоварване на паметта, поради допълнителните обработки. HTTPS използва TLS протокол за сигурност, което забавя предаването на данните. MQTT-SN реализира процес на сериализация като използва MQTT-SN Gateway за преобразуване на структури от данни или обекти до поток от байтове, запазвайки състоянието на техните полета и свойства, което обяснява по-голямото закъснение.

✓ Оценявайки резултатите на проведените експерименти за интеграция между сензорна мрежа и облак най-ефективен от гледна точка на скорост на предаване на данните и разход на енергия е протокол MQTT.

✓ Получените резултати дават полезна и практически приложима информация, относно ефикасността на предаваните данни чрез четири от най-популярните протоколи за интеграция на сензорни данни HTTP, HTTPS , MQTT и MQTT-SN към облачна структура

✓ Резултатите от проведените теоретични, практически и симулационни изследвания могат да служат като полезен инструмент на специалистите при изграждане на интеграция между сензорна мрежа и облак.

ПРИНОСИ НА ДИСЕРТАЦИОННИЯ ТРУД

Научно-приложни приноси

1. Предложен и практически реализиран е алгоритъм за изследване комуникационна ефективност на сензорни мрежи при различни топологии. Резултатите показват, че при клъстерната топология приети пакети и закъснения са с минимални загуби.
2. Предложен и изследван е алгоритъм за енергоефективността на сензорни мрежи с клъстерна топология и протоколи LEACH, LEACH-N и LEACH-H. Резултатите показват, че LEACH-H е най-енергийно ефективен.
3. Предложен и реализиран е алгоритъм за интегриране на сензорни данни в облак Things Speak чрез REST базирани уеб услуги и протокола HTTPS по метода request-response.
4. Предложен, експериментиран и верифициран е алгоритъм за проверка сигурността на предаваните сензорни данни с протокола MQTT в облак ThingSpeak, чрез протокол TLS и поток на данни реализиран с Node-RED.
5. Предложен, реализиран и верифициран е алгоритъм за изграждане и експериментиране на симулационна система за интеграция на сензорните данни от биосензорите в облак за мониторинг на инсулиновите нива на пациенти с диабет чрез хипергликемия, хипогликемия и нормални нива на кръвната захар.
6. Предложен, реализиран и верифициран е алгоритъм за изследване ефективността на интеграцията между сензорна мрежа и облак чрез протоколите HTTPS, MQTT и MQTT-SN.
7. Определена и оценена е работоспособността на протоколите за интеграция на сензорни данни към облак HTTP, HTTPS, MQTT и MQTT-SN в зависимост от различни параметри като брой пакети, брой теми в пакетите и битове за всяка тема върху скоростта (закъснението) на предаваните данни към облака.
8. Установено е, че протоколът MQTT осигурява най-малко закъснение при предаване на данните, най малко натоварва процесора и паметта и има най-малка енергийна консумация. Установено е, че HTTPS и MQTT-SN, работят с по-голямо закъснение, по-голям разход на енергия и натоварване на паметта, поради допълнителните обработки.

Приложни приноси

9. Предложен е модел и създадена тестова информационно-измервателна система за изследване интеграцията между сензорна мрежа и облак, чрез която е проведено експериментално изследване, включващо 17 сценария.

10 Разработени са кодове на Python за експериментални изследвания на протоколите HTTP, HTTPS, MQTT и MQTT-SN.

.

IV. ПУБЛИКАЦИИ ПО ТЕМАТА НА ДИСЕРТАЦИЯТА

[A1]. **Pandurski M.,** Tsvetanov F., Application of Sensor Networks for Measuring Insulin Levels, International Journal of Online and Biomedical Engineering (iJOE) Vol 16, No 14 (2020) pp.122-136, https://doi.org/10.3991/ijoe.v16i14.17185, https://www.learntechlib.org/p/218476/ *Scopus, Web of Science*

[A2]. F A Tsvetanov and **M N Pandurski**, 2021, Security of the Sensory Data in the Cloud, Journal IOP Conference Series: Materials Science and Engineering, International Scientific Conference of Communications, Information, Electronic and Energy Systems – CIEES 2020, vol. 1032, pp. 012005, *Web of Science*

[A3]. Tsvetanov F., **Pandurski M.**, 2020, Methodology for Integration of Sensors data in the cloud, Fifth International Scientific Conference "Telecommunications, Informatics, Energy and Management", pp 109-113, *Nacid*

[A4]. Tsvetanov F., **Pandurski M**., Some aspects for the integration of sensor networks in cloud structures, Proceedings of International Conference on High Technology for Sustainable Development HiTech 2018 p.p. 249-253.q *Scopus*

[A5]. Tsvetanov F., Georgieva I., **Pandurski M**., 2019, The influence of the topology on the radio communication range of sensor networks, Technium: Romanian Journal of Applied Sciences and Technology (ISSN: 2668-7798) Proceedings of the Technium International Conference Vol. 2 (2020): p.p. 17-24 https://techniumscience.com/index.php/technium/issue/view/3, *Ebsco, Google scholar, Crossref,* Publons, DOAJ с др.

[A6]. **Pandurski M.,** Storage of sensor data in cloud databases, XXIX International Conference for Young Scientists, 2020

[A7]. **Pandurski M.,** Tsvetanov F., (2020), Research of energy resources of the cluster sensor network, "Journal of Engineering Science and Technology Review", Special Issue on Conference in Telecommunications, Informatics, Energy and Management, (ISSN: 1791-2377) vol.13, pp.32-36, *Scopus, DOAJ, Google Scholar, EBSCO*, ACS Publications, CERN

[A8]. Tsvetanov F., **Pandurski M,** 2019, Multidisciplinary Journal of Science, Education and Art, Технически приложения на различни типове сензори в зависимост от измерваната величина, ISSN 1313 – 5236, p.p. 415-427, *Nacid*

[A9]. Filip Tsvetanov, **Martin Pandurski**, Ivanka Georgieva, Grigor Mihajlov, Simulation of the performance of wireless radio controlled fire protection system, 2020 7th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE), pp. DOI: 10.1109/EEAE49144.2020.9279048, *Scopus, Web of Science*

Пет от публикации са реферирани в Scopus, Web of Science, две в Nacid и две в международни вторични бази от данни Google Scholar, EBSCO, Crossref, Publons, DOAJ с др.

South West University "Neofit Rilski"
Faculty of Engineering

DEPARTMENT of „Communication and computer technique and technologies“

MARTIN NIKOLOV PANDURSKI, M.Sc,

INVESTIGATION OF THE POSSIBILITIES FOR INTEGRATION OF SENSOR NETWORKS IN CLOUD STRUCTURES

ABSTRACT of Ph.D. THESIS

The subject of the current PhD Thesis are wireless sensor networks (WSN), protocols, like Zigbee, HTTP, HTTPS, MQTT, MQTT-SN and cloud platforms. In the dissertation research tasks are solved, testing the performance of the protocols (HTTP, HTTPS, MQTT, MQTT-SN), regarding their ability to integrate sensor data on cloud platforms, offering a modification of the standard algorithms, offering new models for studying and evaluating the performance of protocols and conducting theoretical and practical research on the most popular representatives of the wireless sensor networks, protocols and cloud platforms.

Analysis of the results of the studies found significant differences between the performance of HTTP, HTTPS, MQTT and MQTT-SN, regarding the data speed transmitted from the sensor network to cloud platforms, as well as energy consumption. MQTT shows the highest-level performance, while HTTPS and MQTT-SN, work at a lower-level performance due to their specifications.

The results of the theoretical and practical studies can serve as a useful tool professional when designing WSN and choosing a protocol for integration of sensor data in cloud platforms.

SOUTH-WEST UNIVERSITY
"Neophyte Rilski"
Technical Faculty

DEPARTMENT of "Communication and computer technique and technologies"

MARTIN NIKOLOV PANDURSKI, M.Sc,

INVESTIGATION OF THE POSSIBILITIES FOR INTEGRATION OF SENSOR NETWORKS IN CLOUD STRUCTURES

ABSTRACT of PhD. THESIS

for awarding a scientific and educational degree "Doctor"
in PD 5.3. "Communication and computer technology",
in the scientific speciality "Computer Systems, Complexes and Networks"

Supervisor**:**
Ch. Assistant Professor Dr. Eng. Filip Tsvetanov
2022

The dissertation was discussed at a meeting of the Department of "Communication and Computer Engineering and Technology" at the Technical Faculty of South-western University "Neofit Rilski" on February 10, 2022, and is scheduled for official defence before:

Scientific jury composed of: Prof. Dr. Eng. Peter Apostolov - SWU, Prof. Dr. Eng. Galina Cherneva - SWU, Prof. Dr. Eng. Georgi Lyubenov Iliev - TU Sofia, Assoc. Prof. Dr. Eng. Agata Hristova Manolova - TU Sofia, Assoc. Prof. Dr. Eng. Ivan Dinkov Ivanov - VUTP Sofia.

Reserve members of the scientific jury: Assoc. Prof. Dr. Eng. Ivan Nedyalkov - SWU, Assoc. Prof. Dr. Eng. Valentina Ilieva Markova - TU Varna.

The *dissertation* contains: number of pages -182, number of figures - 119, number of tables - 10, number of formulas - 30, number of literary sources -155.

On the topic of the dissertation have been made 9 publications

The numbering of the figures, tables and formulas in the abstract corresponds to those in the dissertation.

The materials on the defence are available in the Secretariat of the Department of Communication and Computer Engineering and Technology and on the website of Southwestern University.

Author: Martin Pandurski, M.Sc,

Title: INVESTIGATION OF THE POSSIBILITIES FOR INTEGRATION OF SENSOR NETWORKS IN CLOUD STRUCTURES

Circulation: 20 pcs.

Design, layout and prepress: the author

I. GENERAL CHARACTERISTICS OF THE DISSERTATION

Relevance of the problem

Modern technological advances in the miniaturization of devices and wireless sensor technologies expand the possibilities for wireless sensor networks (WSN) applications. The sensor network is self-organizing and consists of a large number of different types of sensors that are located in a specific monitoring area and transmit data to each other via wireless communication.

The WSN applications are numerous, such as environmental monitoring, industry, etc. The amount of data generated by sensor networks is huge and diverse.

A specific feature of WSN is the limited resources. Limitations include low power consumption requirements, communication perimeter, low bandwidth, limited sensor data processing, etc.

The efficient usage of large volumes of data collected from sensor networks requires a powerful, scalable, high-performance computing infrastructure for real-time data processing and storage and analysis of processed information.

Cloud structures provide enormous computing power and storage space. Integration between clouds and sensor networks is an excellent solution to the problem of the limited computing power of sensor networks for data storage and processing. A new paradigm called Sensor Cloud Computing has been formulated to achieve this integration.

The *relevance* of the work is determined by the fact that WSN provides an opportunity to develop innovative approaches to data collection, processing and integration into cloud remote control systems in many areas of industrial automation, environmental monitoring, etc.

The present work explores some possibilities and algorithms for the efficient operation of sensor networks in integrating sensor data in cloud structures.

SCIENTIFIC DEFINITION OF THE RESEARCH

The object of the research

The dissertation's research object is sensor networks based on the IEEE 802.15.4 standard.

The research subject is models with cluster and mesh topologies and data transmission for the IEEE 802.15.4 standards, models for integrating sensor data to cloud structures.

The objective of the dissertation:

To study the operation of wireless sensor networks and propose models and algorithms for integrating sensor data into cloud structures.

Tasks:

1. Research and analyze modern methods for creating and managing

sensor networks.

2. Analysis of the protocols and problems for sensor networks - cloud structures integration.

3. Physical construction of a sensor network-cloud structure and development of an algorithm to effectively integrate sensor data by receiving, transmitting, visualizing and analyzing data from sensor devices.

4. To explore the physical security of the transmitted sensor data in the cloud structure.

5. To evaluate the influence of the protocols and mechanisms, providing services for sensor network-cloud integration.

Research methods

The dissertation uses methods, analysis, synthesis, modelling, computer simulation and programming, empirical methods such as observation, comparison, and practical experiments to solve the given objectives.

Approbation research

In the research process, simulation and experimental setups are realized, allowing several types of research of protocols for communication and transmission of sensor data and the development of control software to integrate data to the cloud. The intermediate results are reflected in *nine publications* reported at international conferences, in journals, 5 of which are in referenced databases Scopus and Web of Science, 2 are in the Nacid database, 2 in international secondary databases Google Scholar, EBSCO, Crossref, Publons, DOAJ, etc.

Structure and volume of the dissertation

The dissertation has 182 pages, including an introduction, three chapters for solving the formulated main tasks, a list of key contributions, a list of dissertation publications, references and applications. A total of 155 literary sources were cited, 144 in Latin and 11 in Cyrillic. The work includes a total of 119 figures and 10 tables. The numbers of the figures and tables in the abstract correspond to those in the dissertation.

Practical applicability: The practical applicability is enormous due to the fact that the number of Internet-connected devices is constantly growing. The volume of information is continuously increasing, and new solutions are needed to integrate sensor networks into cloud structures to increase efficiency and connectivity.

II. DISSERTATION CONTENT

FIRST CHAPTER

Modern approach analysis for creating and managing wireless sensor networks

As a result of modern methods, analysis for construction and management of WSN are performed:

- ✓ Researching and systematizing the main directions of the specific features, requirements and challenges facing wireless sensors, a major component in the construction of sensor networks;
- ✓ Studying and systematizing architecture of WSN, the models for communication and power management, mobility and operations in the sensor networks;
- ✓ Analyzing the structure of sensor networks and the specification of star, tree, mesh, hybrid and cluster topology in the construction and application of the sensor networks;
- ✓ Studying and analyzing the main security issues and challenges facing sensor networks, such as limited sensor resources, radio communication environment for data transmission, sensor energy consumption management, limited processor power and radio range;
- ✓ The limited computing capacity of sensor networks leads to a problem in analyzing and storing the generated huge amount of sensor data. The integration between cloud structures and sensor networks enables the solution for this problem, as well as the storage and processing of collected data, without unnecessarily costing increasing the sensor networks;
- ✓ Formulated are thesis that creating models for sensor data integration to clouds reduces the development time and efficiency evaluation of sensor networks. To ensure higher reliability and energy efficiency of designed systems, which is a current problem in the field of sensor networks:
- ✓ Experimental studies have been conducted on the effectiveness of the WSN architecture.

1.3.2. WSN architecture, communication and management model

The WSN architecture consists of different node types used to monitor different parameters in the environment. These nodes can be used in real-time applications and efficient routing between BS and nodes. There are two types of WSN architecture: *Multilayer network architecture* and *Cluster architecture*. Multilayer network architecture uses multiple sensor nodes and a single powerful BS.

In the *Cluster network architecture*, individual sensor nodes are added in groups (clusters) via LEACH Protocol "Retrieval Protocol". The advantage of LEACH is in reducing energy consumption by creating

clusters and hierarchical routing, thus increasing the life of WSN because it shortens the time for data transmission to each sensor node. All nodes in the network are organized in local clusters, with one cluster "head". The functions of the cluster head require high energy efficiency. Different LEACH variants have been created to select the cluster head with the highest energy efficiency.

The clustering algorithm is designed so that each node in the network sequentially acts as a cluster head for the same period of time, Fig. 1.10, provided that all nodes start with the same amount of energy. The threshold value is determined according to expression 1.1.

$$Tn(LEACH) = \begin{cases} \frac{p}{1-p\times\left(r\times mod\frac{1}{p}\right)} & if\ n \in G \\ 0 & otherwise \end{cases} \quad 1.1.$$

Where:

P - predetermined percentage for the possible number of head nodes;

R - the current operating interval;

G - number of sensor nodes that are not selected as heads during the last $1/p$ intervals.

If this random number is less than the threshold value, T(n), the node becomes a cluster head for the current cycle. The threshold value is calculated based on equation (1.1). When a node is selected as a cluster head, it transmits an ADV message.

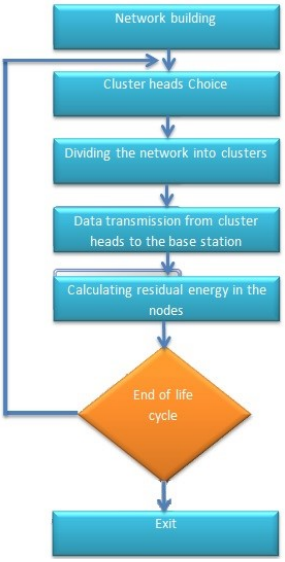


Fig.1.10. LEACH algorithm

This message contains the node ID and title. Each non-cluster node determines which cluster it will belong to by selecting a cluster,

requiring the minimum communication energy based on the strength of the ADV signal received from all cluster heads.

1.3.2.2. Experimental study of sensor network energy efficiency with a cluster topology [A7].

An experimental simulation study of the operation of the LEACH, H-LEACH and N-LEACH protocols is planned. H-LEACH minimizes energy costs by operating as LEACH in the first round. The following rounds use a method to optimize the distance for transmitting data to the base station (BS) to select which nodes will become cluster heads. N-LEACH maintains information for the remaining energy of all nodes and sends it to the BS in each cycle. Matlab code has been developed to implement the algorithm.

Algorithm

// *n* is the set of all nodes in the network, r is the number of circles, p is the probability that the sensor is a cluster head.

***Step 1*:** Random location of all *n* nodes fig.1.11.

***Step 2*:** Set the initial equal energy to all nodes.

Step 3: Select a cluster head in over 0.5 percent of the nodes in the network. In LEACH-N, nodes with higher energy levels are selected for cluster heads. The nodes closest to the BS are selected for cluster heads in LEACH-H.

Step 4:: Each non-CH node joins the nearest cluster head.

Step 5: Each CH transmits the data to the BS by the shortest path.

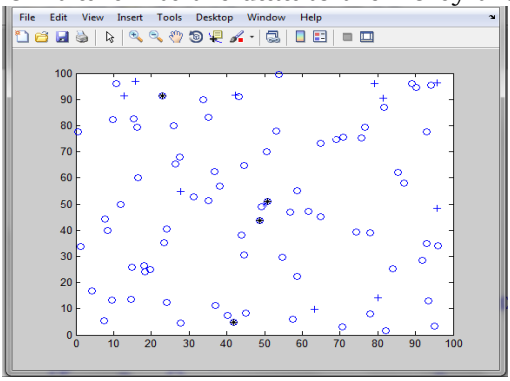


Fig.1.11. Random location of all nodes in the field

Results from the simulation study

A) Number of data received at the base station, Fig.1.12

The results show that the number of transmitted data packets sent by LEACH-H slowly increases and becomes bigger than LEACH and LEACH-N.

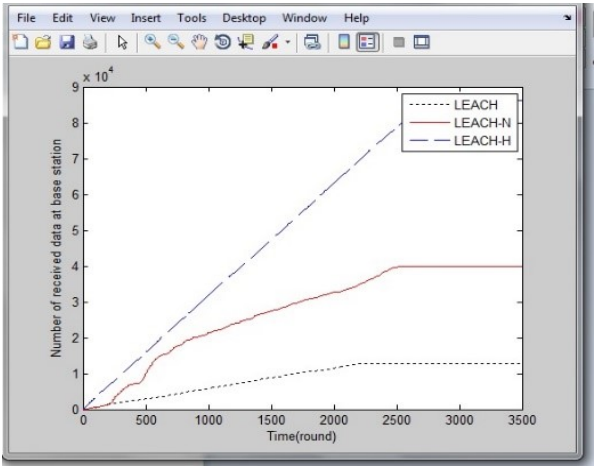


Fig.1.12.Number of data received at the base station in LEACH, LEACH-N, LEACH-H

B) Average residual nodes energy, fig.1.13

The experiments show that LEACH's average residual nodes energy is less than LEACH-N and LEACH-H.

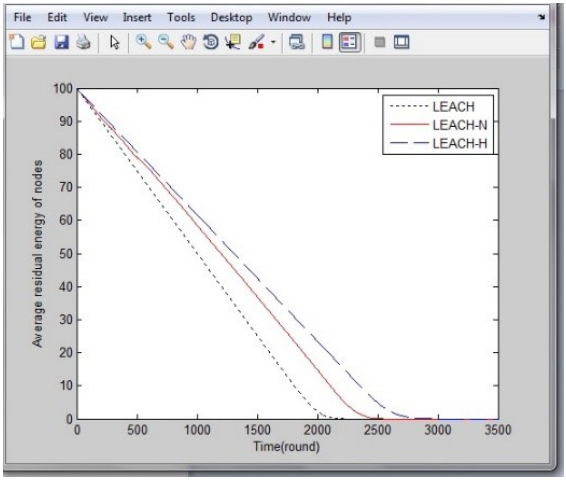


Fig.1.13. Average residual nodes energy in LEACH. LEACH-N and LEACH-H

C) Number of exhausted nodes, fig.1.16

With a probability of 0.5 per 100 nodes and 3500 rounds, from Fig.1.16, it is seen that at 1500 rounds, nodes start to die in LEACH. LEACH-N - about 2000, and LEACH-H - about 2500. The results show that LEACH-H gives the best results in improving network life compared to the other two protocols.

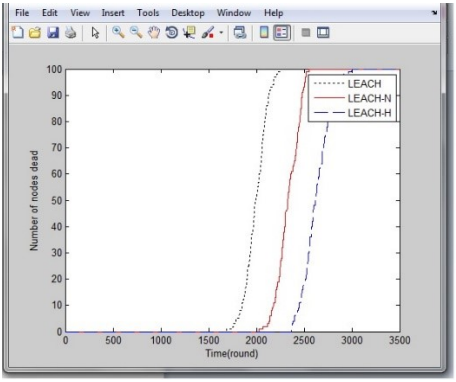


Fig.1.16. Number of exhausted nodes LEACH, LEACH-N и LEACH-H

Usually, the most energy-efficient path is defined as the optimal path for data transmission. In this way, each node must be aware of its neighbours' capabilities to choose the best neighbour to send data to. Although a significant improvement in network viability has been achieved, this type of method requires frequent updating of the road energy information in the routing tables, leading to an additional accumulation of self-organized WSN.

1.5.5. Experimental study of the topology influence on the network communication range efficiency [A5]

To study the network topology influence, an experiment was planned and conducted that includes:

1) *Hardware design* of XBee sensor network with seven XBee modules. We have coordinator, routers, and end devices functions with three different topologies: star, cluster, and mesh.

2) In the *Network design*, three types of networks are configured: star, mesh, and cluster.

3) ***Algorithm*** *for studying the topology influence on the communication efficiency range of the sensor network.*

Step 1. *Select the parameters of the hardware modules.*

Step 2. *Install XCTU software.* With the software, we set up the XBee modules.

Step 3. *Configure the module's operation mode and update the firmware in two modes. XBee modules in AT transparent mode.* All serial data received from the XBee module is sent wirelessly to remote modules. Remote modules are in API mode. To test the wireless network range, the XCTU must have access to at least one remote XBee3 device on the network in API mode. An API Tx frame is sent from one XBee module to another API Rx module.

Step 4. *Configuring device functions*, like terminal, router and coordinator.

Step 5. *Conduct the test.* When the test starts, the XCTU sends data

packets from the local XBee module to the remote Xbee and waits for the echo to return. XCTU records the number of packets transmitted and estimates the signal strength on both sides via RSSI.

Step 6. *Analysis of the obtained results*. The loss of more than 5% of the data packet can deteriorate network performance quality. The analysis of obtained parameters resulting from the experiment answers the question of the most efficient topology. Fig.1.30. shows the data transfer results from an end device to a coordinator in the cluster. Fig.1.31. shows the packets, delay errors, RSSI diagram when transmitting data from an end device to a router in the cluster. The test results for data transmission from the router to the coordinator in the cluster are shown in Fig.1.32. The percentages received are displayed as lost packets, latency errors, RSSI charts, etc.

Fig.1.30. Test from end device to coordinator in a cluster

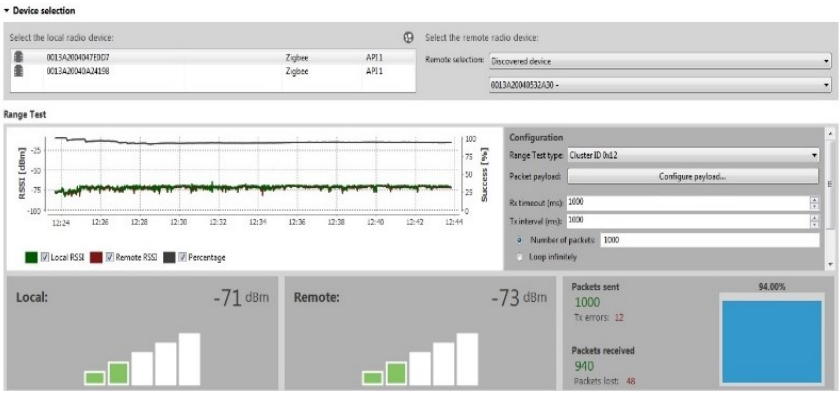


Fig. 1.31. Test from the end device to the router in the cluster

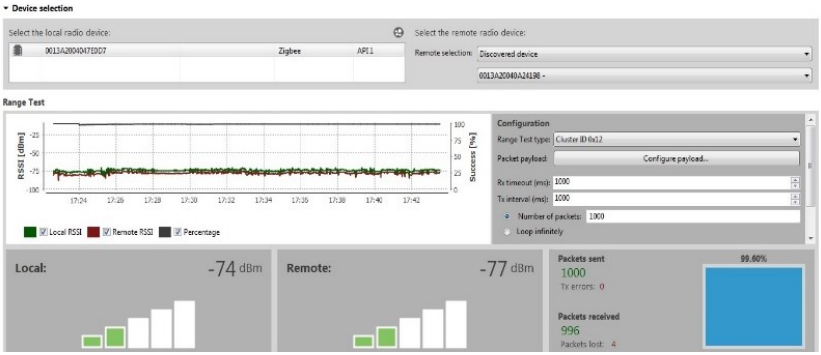


Fig. 1.32. Test from router to coordinator in a cluster

The test results from end device to coordinator in a star topology are shown in Fig. 1.33, and the range test shows the percentage of received packets.

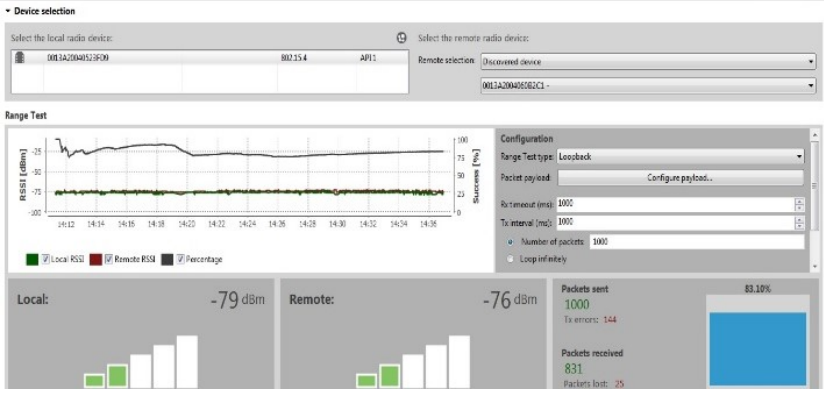


Fig. 1.33. Test from end device to coordinator in star

The test results from the end device-to-end device in the star are shown in Fig 1.34.

Fig. 1.34. Test from end device-to-end device in star

The router to router test results in Mesh topology is shown in Fig.1.35. The analysis of the results for sensor network's communication efficiency in different topologies was performed according to the instructions in works [57] and showed that:

1) In Star Topology, the data sent from the end devices to the coordinator does not lose packets from less than 10 m distance. But there is a loss of packages above 10 m. The tests were performed with 1000 packets. The received packets are, on average, 82%. This topology shows the lowest results in terms of range distance, received packages and delays.

2) In Mesh Topology, as the distance increases, so do packet loss and delay, as long-distance communication takes time to transmit data through the router.

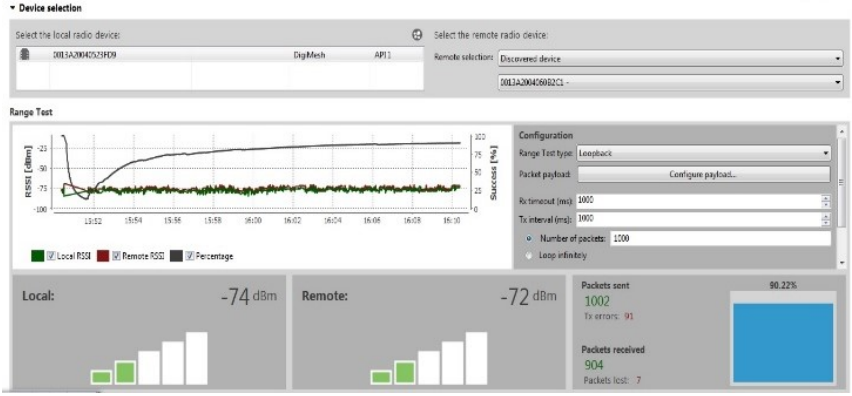


Fig. 1.35. Test from router to router in Mesh topology

The tests were performed with 1000 transmission packets. The received packages are 90.22%. This topology shows better quality results but can reduce network performance quality.

3) In Cluster Topology, the tests are performed with three different types of devices - coordinator, router and end device. Tests were performed between coordinator-to router, end device - to coordinator and end device - to the router. During the tests, 1000 packets were transmitted. The received packets are, on average, 97.30%, which does not lead to a deterioration in the network quality. The results show that the most successful data transfer is between coordinator-router, router-coordinator, router-router, end device-coordinator, end device-router and end device-end device. The results show that the cluster topology has a significantly higher success rate than the other two topologies.

As a result of the performed literature research and the performed experimental research, we can ***summarise***:

- In recent years, the industry's interest in integration problems of sensor networks has been established. Intensive research is being

conducted on the design and construction of wireless sensors, with features that overcome part of the above deficiencies. The study also addresses network security, energy consumption, and sensor networks' life issues.

- There is no universal methodology for research and evaluation of sensor networks parameters, so it is necessary to develop new models, approaches, and algorithms to improve experimental and simulation research.

- An essential problem with sensor networks is generating a huge amount of sensor data because the limited computing power and capacity of sensor networks make it difficult to process the data. The integration between cloud structures and sensor networks gives the solution to this problem and the storage and processing of the collected data without unnecessarily increasing the cost of the sensor networks.

- The thesis is formulated that creating models and algorithms for sensor data integration to the cloud allows for reducing the time for development. Evaluating the sensor network's efficiency to ensure higher reliability and energy efficiency of designed systems is a relevant problem in the sensor networks field.

CHAPTER TWO

Models and algorithms for sensor data integration in cloud structures

2.1. Analysis of modern solutions for storage and processing sensor data

The extracted sensors data, presented as simple time series, is called *sensor data*. Often, sensor data may not change over a long time, but in other cases, it grows rapidly, with several reports per second for one or more sensors from one network. The main models for storage and processing of sensor data are 1) Storage and processing of data outside the sensor network and 2) Distributed storage and processing of sensor data in the network.

TSBDs (time-series databases) are specially designed to store sensor data. TSDB is a software system optimized for processing data from time series, arrays of numbers indexed by time (date-time). Characteristic features of TSDB are data lifecycle management, aggregation and scanning of a wide range of many records.

2.2. Structures for sensor data integration in cloud structures

2.2.2. Characteristics of cloud structures

Cloud structures are both a platform and a type of application. Cloud structures provide 1) On-Demand Service, 2) Service Measurement, 3) Pooling of Resources

2.2.3. Cloud models, according to the accessibility level, are private, public, social and hybrid cloud infrastructures. The functional structure of the connection of the sensory data with the clouds is shown in Fig.2.4.

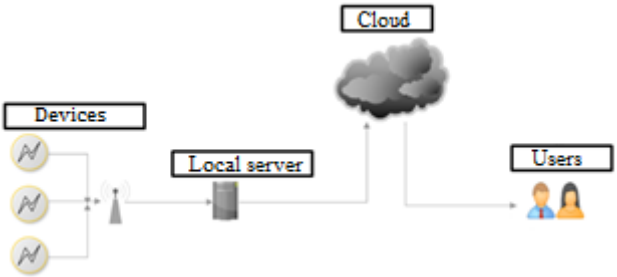


Fig.2.4.Functional structure of sensory data transmission to the clouds

By combining different technologies, different systems can be created to solve the necessary tasks. For example, Mathworks provides a ready-made cloud platform for IoT applications - *ThingSpeak*. This free cloud service collects sensor data and, with the tools provided by the platform, provides visualization and analysis of various types of data.

2.3. Protocols for communication and data transmission between sensor networks and cloud platforms

2.3.1.Communication protocols sensor data for integrating into clouds

The biggest challenge in designing IoT systems is establishing a communication channel between devices, gateways and cloud platforms. Therefore, this requires the use of different protocols. The complete communication stack consists of four different levels - "layers" [85] shown in Tab 2.1.

Tab.2.1. Communication layer protocols for data transmission to the cloud

Layer	**Functions**	**Protocols**
Application layer	Stores payload	•*MQTT* (Message Queuing Telemetry Transport) • HTTP, REST (Representational state transfer), RESTful
Transport layer	Provide end-to-end communication for the application layer.	TCP/UDP
Internet layer	Internet Protocol (IP) is a layer that defines transferring data	IPv4 and IPv6

	between hosts. IPv4 and IPv6	
Channel layer	Controls host-network connections	•802.11 WiFi •Zigbee •Ethernet

2.3.2. HTTP for communication in sensor and IoT networks

The HTTP protocol determines the way messages are transmitted and formatted on the Internet. HTTP transfers a large number of small packets when communicating with the IoT. TCP connections are released each time they are accessed and transferred based on IP and URL, and their connection changes dynamically. Therefore, IoT communication causes network resources consumption and long delays.

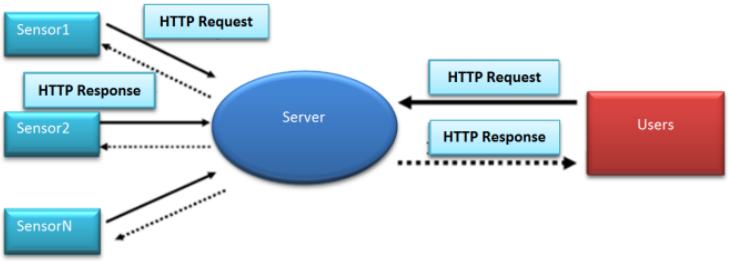


Fig. 2.7. HTTP System configuration

2.3.3. Rest and Rest full

Rest is a software architecture for implementing web services using HTTP. REST consists of clients and servers and uses a bus-based architecture where no broker is needed, and end devices can communicate directly.

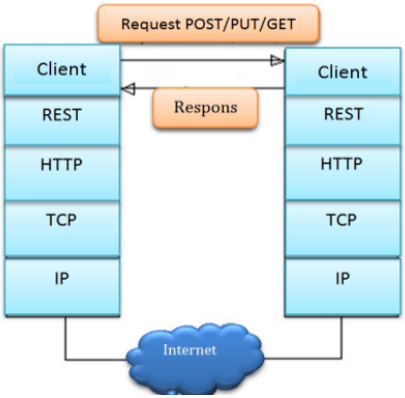


Fig.2.9. RESTful architecture

REST means a server that shares JSON files with an HTTP client. Clients initiate requests to servers; the servers process the requests and return appropriate responses. *Restful* is an advanced form of web

sharing server that shares all other documents or JSON files to develop new applications.

2.3.4. MQTT protocol

MQTT is a message application protocol based on publish/subscribe architecture. MQTT runs on TCP/IP, which provides orderly, two-way connections. Pub/sub is event-driven and allows messages to move between devices and clients through a broker who manages Pub/sub-operations. An essential element of the MQTT architecture is the broker, which processes the messages, separates the publisher from the subscriber and acts as a router for the messages. Each client who publishes a message to the broker includes a topic in the message.

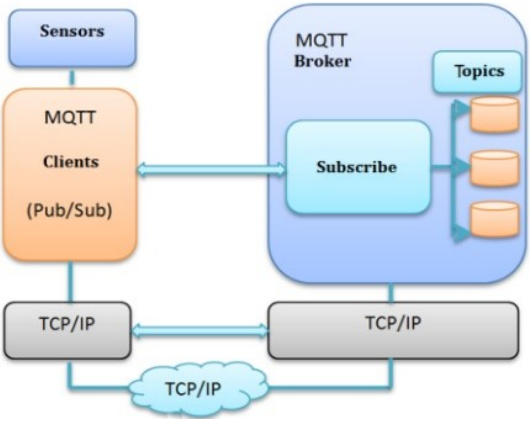


Fig.2.11. *MQTT* architecture, publisher and topics

Fig.2.14 [94], The MQTT packet structure consists of: *Fixed header* - Contains 2 bytes (CONNECT, PUBLISH.). *Variable Header*. The title's content varies depending on the type of package (Flag information, package identifier). *Payload* - The data to be sent.

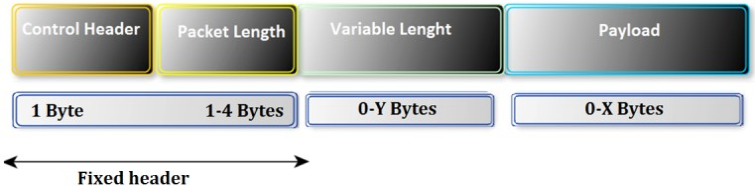


Fig. 2.14. *MQTT packet structure*

Fig.2.15 shows the sequence for exchanging messages.

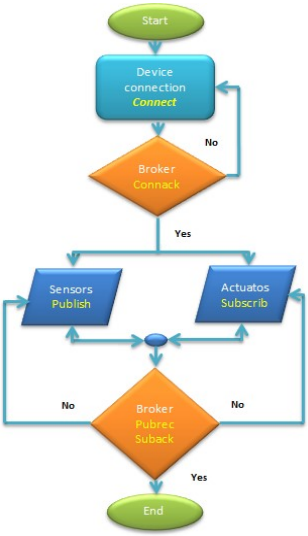


Fig.2.15. *MQTT* requests exchange. Block diagram

2.3.5. MQTT based communications expected delay model in data transmission.

The model [97] describes the relationship between data size, data collection intervals, network traffic and delay and presents an approach to modelling MQTT network design based on experimenting with packet-level behaviour. The end-to-end delay in the network connection consists of various events expressed by equation 2.1. For each node, there is a delay due to processing, queuing, transmission and propagation.

$$D_{node} = D_{processing} + D_{queuing} + D_{transmission} + D_{propagation} \quad (2.1)$$

Processing and transmission delays occur at individual nodes, and propagation delays occur between nodes. Particular attention should be paid to transmission delay when aggregate data from multiple sensor networks is propagated *through a single node*. The transmission delay can be determined by equation 2.2.

$$D_{transmission\,(sec)} = Segment\ Length_{(bits)}/Rate_{(bits/\sec)}. \quad (2.2)$$

2.3.6. Comparison between MQTT and HTTP

MQTT is data-oriented, and *HTTP* is document-oriented. HTTP uses the request-response model, and MQTT uses the pub-sub model. Compared to HTTP, MQTT is more compact (short message headers and smallest packet message size of 2 bytes) and composed of longer headers and messages. The main differences in the protocols are systematized in Tab 2.5,

[86].[98],[99],[100].

Table. 2.5 Differences in *MQTT* and HTTP

Parameter	***MQTT***	**HTTP**
Name	Message Queuing Telemetry Transport	Hyper Text Transfer Protocol
Architecture	Pub/Sub	Request/Response
Complexity	Less complexity	More complex
Works on	Transmission Control Protocol	User Datagram Protocol
Protocol design	The data design is data-centric.	The design of the protocol is document-oriented.
Message size	The message size is smaller due to the binary format.	The created message size is larger due to the ASCII format.
Header size	2 bytes	8 bytes
Port number	1883	port 80 or 8080.
Data Security	Provides data protection with SSL/TLS	HTTP without security, but there is HTTPS.

2.3.6. MQTT-SN protocol

The MQTT-SN (Message Queue Telemetry Transport -Sensor Network) protocol is a version of MQTT adapted to WSN, making it the most suitable for sensor devices due to its compact messaging. MQTT-SN uses UPD/IP because it is more compact than TCP/IP. The architecture of the protocol is shown in Fig.2.17. The MQTT-SN component's most important MQTT-SN Gateway (GW) allows MQTT-SN clients to connect to an MQTT broker.

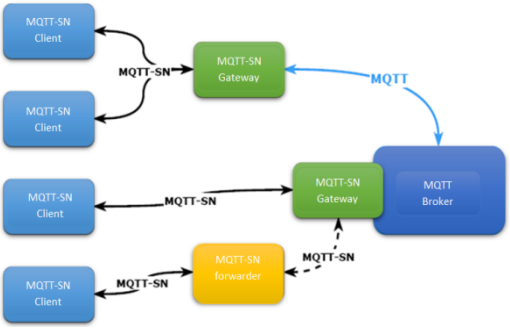


Fig.2.17. *MQTT*-SN Architecture

2.4.Sensor data integration into cloud structures

2.4.1.Basics of integration [A6]

Connecting sensor networks to cloud structures solve the problem of storing and processing large amounts of data from sensor networks. The communication between WSN and cloud platforms is realized through a gateway or a coordinator device. Fig.2.21 is a block diagram illustrating the integration of a WSN with a cloud structure.

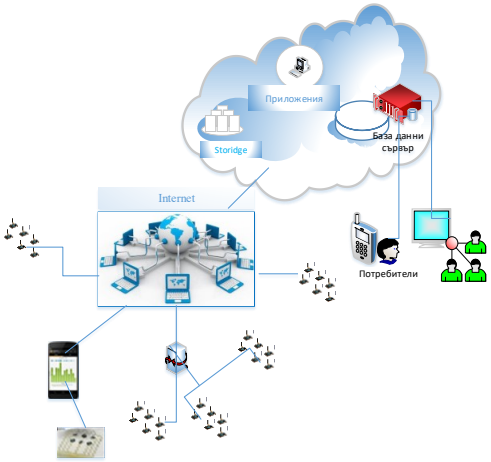


Fig.2.21. Sensor data integration into cloud structures [A6]

2.4.2. Scenarios for data transmissions to the cloud

Data can be transmitted via coordinator or gateway, direct transmission, to the cloud server. There are several ways to organize access to shared resources - communication channels and server computing resources: query, interruptions and multiple access.

2.5. Approaches to managing sensor data security in cloud integration

Currently, various approaches are applied to manage the security of the sensor data integration process to the cloud, such as the design and development of network controllers, international standards for cloud security, and hardware protection.

2.5.5. MQTT based algorithm for sensor networks authentication access

There are two MQTT security mechanisms. One is authentication with a username/password, and the other is access control via ACL (Access control list) file or database. The

username/password is transferred to the body of the CONNECT message. If authentication is passed, the gateway responds with CONNACK. In [123], the use of TLS for the security of authentication between sensors and gateway is proposed. TLS is implemented in two stages. The first generates a common session key based on X509 certificates using DH (Diffie-Hellman or ECDH - Elliptic curve (Diffie-Hellman)). The second is using an algorithm for symmetric or asymmetric encryption (AES, RSA (Rivest – Shamir – Adleman)). The advantage of this method is the complete encryption of the connection. Connecting via TLS, the device must connect to the gateway at the last address but with a different port number. For example, for an unsecured connection, the address will be «TCP://127.0.0.1:1883», while for a connection via TLS - it will be «SSL://127.0.0.1:8883». In this way, a secure TLS connection is established. The gateway is then authenticated with a username/password. The operation of the access algorithm aims to ensure that the device knows the password without transmitting it directly. The end device must perform the registration procedure through a secure channel. The registration procedure is visualized in Fig.2.26.

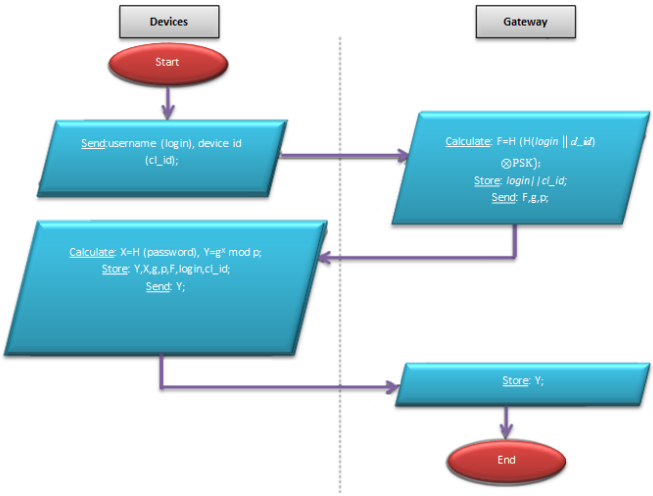


Fig. 2.26. Registration procedure

The end device performs the registration procedure through a secure channel. The device provides *login* and *cl_id* values to the gateway. The gateway then calculates token *F*:

$$F= H (H(login || cl\_id) \otimes PSK), \quad (2.17)$$

PSK is a secret key for the gateway. It sends to the device F,g,p. The parameters p and g are public.

Once the device receives p, g, F values from the gateway, it can calculate the X and Y values:

$$X = H(password), \quad (2.18)$$

$$Y=gX \bmod p. \quad (2.19)$$

The gateway stores the Y value. The procedure is complete

2.8. Conclusions from the second chapter

• Analyses are made for the main sensor data presentation and transmission models, specialized sensor data time series TSBD databases, problems of sensor data integration to cloud structures, and steps for sensor data management.

• The issues resolved for sensor network-cloud structure integration are defined.

• Sensor data storage models have been analyzed: a Five-layer OSI model and three models (file, block and object) for data storage in the data management layer.

• The main protocols for communication and data transmission between sensor networks and cloud platforms HTTP, HTTPS, REST, RESTful, MQTT and MQTT-SN are considered.

• The essence of integrating the sensor network and cloud is clarified. The approaches for managing the sensor data security for cloud integration are defined: design and development of specialized network controllers, international standards, and specialized hardware protection.

• The main data transmission modes from sensors to the servers in the cloud structure are defined, and the possibilities for delaying the data transmission are analyzed.

• Algorithms for authentication of communication based on the MQTT communication protocol are presented, which allows verifying the authenticity of a device without direct password transmission and the MQTT TLS protocol data protection method, whose application increases the level of authentication security for sensor networks.

• The methods for WSN research, simulation tools and proposed methodology for conducting computer-based simulation research are systematized.

CHAPTER THREE

Exploring some possibilities for integrating sensor data into cloud structures

3.1. Model for studying the integration of sensory data into cloud structure

Experimental research for possible integration models between sensor networks and cloud structures is realized, according to the proposed model of fig.3.1. This model aligns with the gateway data transmission scheme analyzed in [A4].

The *physical layer* includes intelligent sensors sending data to the microcomputer. The *network layer* includes a microcomputer that acts as a gateway and forwards data to the telecommunications base station (BS). BS transports the received data to the cloud platform, which contains data storage and processing servers. The *platform management layer* provides data storage and device management features. The *application service layer* is connected to the cloud platform via API to implement the online data request and remote monitoring feature.

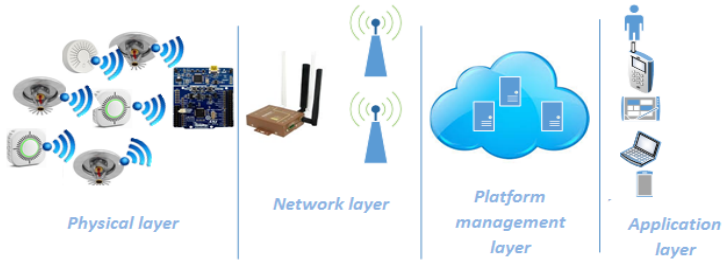


Fig.3.1 Sensor data - cloud structure integration model

3.2. Communication models of interaction between sensor networks and cloud structures

3.2.1. Communication model request-response allows clients to request information from a server that receives the request message, processes it, and returns a response message. The two best-known protocols from that model are REST HTTP and CoAP. In Fig.3.2. REST HTTP request/response interaction between two clients and one server is shown.

Fig.3.2. Communication model request-response

3.2.2. Communication model pub/sub provides communication between data senders and destinations. The model consists of three parties: publisher, subscriber and broker, fig.3.3. The broker is software running on a computer and acting as a "post office" for a device acting as a broker. It is necessary to install the MQTT broker library.

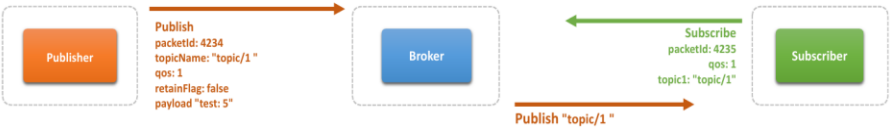


Fig.3.3. Communication model pub/sub, broker

3.3. *Methodology for integrating a sensor network data collection and sharing using the request-response method via REST-based web services to ThingSpeak cloud [A3]*

3.3.1. Motivation

The objective is to build a physical XBee sensor network with the Digi XBee Cloud Kit and evaluate the ability to integrate the collected sensor data and send it to the ThingSpeak cloud via *REST HTTP*.

3.3.2. Hardware components

Digi XBee Cloud Kit is used for cloud-sensor integration [142]. Python code is imported in Digi XBee Gateway, allowing data integration to ThingSpeak, allowing online data processing and analysis via Matlab code.

Fig.3.4. Digi XBee Cloud Kit

3.3.3.Architecture of the experimental network

Fig.3.6. Sensor data integrate into *ThingSpeak*

3.3.4. Algorithm *for building the sensor network and integrating it into the cloud*

First*: Configuration XBee modules network via XBee ZigBee coordinator.*

Second: *Additional network and communication devices settings* - firmware type, encryption, PAN ID, nodes connection to the network, etc.

Third*: Building a channel for transmitting sensor data in ThingSpeak, includes* defining the channel name and API keys.

Fourth: *Connect the XBee network data to ThingSpeak via XBee Gateway and develop Python code to integrate data into ThingSpeak. ThingSpeak* retrieves data from devices via *HTTP.*

Fifth: *Collecting and sending sensor data to the XBee Gateway via XCTU.* From the XBee Gateway, they are sent via REST/HTTP to *ThingSpeak.*

Sixth: *Using MatLab code visualization and analysis of the data.*

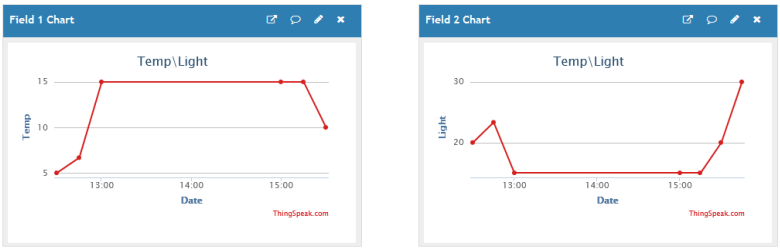


Fig.3.10. Results of data measurements in *ThingSpeak*

Fig. 3.11. Matlab preview options in *ThingSpeak*

The visualization results show that the proposed algorithm for integrating sensor data into the cloud provides online access, monitoring and analysis of incoming data.

3.4. Experimental study of the transmitted sensor data security in ThingSpeak cloud [A2]

3.4.1. Problem analysis

The security problems of the sensor networks transmitted data are considered in item 2.6. Numerous studies, summarised in [137], show that the integration security through REST/HTTP and MQTT, which are the subject of research, is most appropriate to be performed through TLS/SSL.

3.4.2. Choice of security protocol

The choice of this protocol is determined by the capabilities of the selected hardware implementation. Fig.3.1 shows the result of a study by the company *RF Desk security solution* [143].

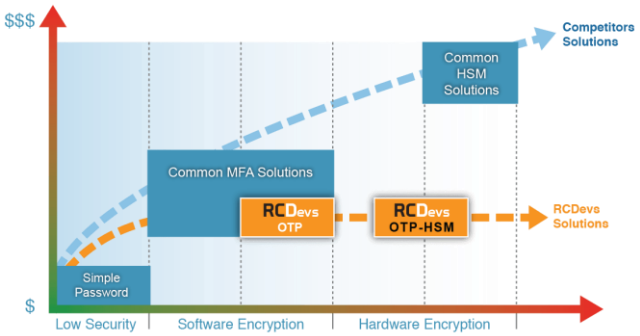


Fig.3.13. Choice of data security solution [143]

We choose the TLS protocol because it protects the transmitted data *online between a web browser and a website via HTTPS*. It ensures that the data retains its original integrity and confidentiality through encryption.

3.4.3. Experimental installation for security research

The experiment was performed according to the models from Fig.3.1 and Fig.3.3. The experimental setup is constructed according to the block diagram in fig. 3.14.

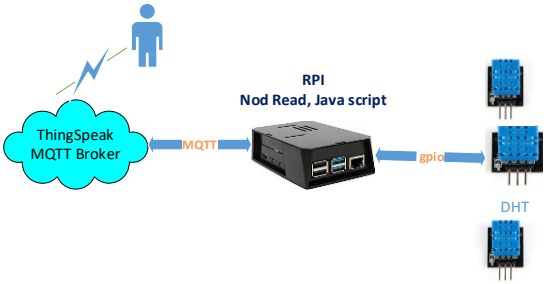


Fig.3.14. Experimental installation for security research

3.4.4. Algorithm for studying the security of the transmitted data from the sensor network to the cloud

First: *Choosing a cloud.* Data is sent to ThingSpeak in real-time, allowing analysis and transmitting of data in real-time with Open API (HTTP or MQTT).

Second: *Select a protocol for connecting the network to the cloud and data transmission.* We choose MQTT because it is designed to connect devices with limited power over wireless networks, uses a *pub/sub* model and a broker to send short messages

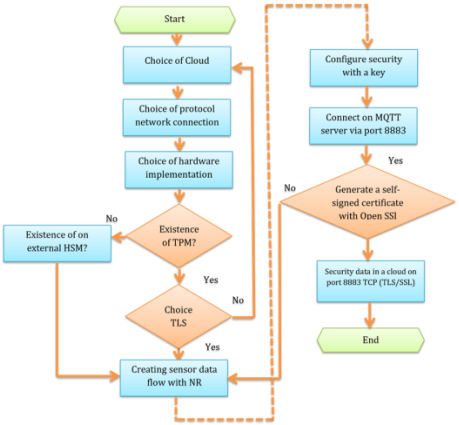


Fig.3.15. Algorithm for checking TLS security of transmitted sensor data

Third: *The choice of security protocol depends on the hardware implementation.* According to the considerations in item 3.4.2, the TLS protocol was chosen.

Fourth: *Creating a data collection stream is done through MQTT and Node-RED.*

Fifth: *The connection to the MQTT broker/server and publishing using the publishing node is made by entering the IP address/name of the Broker and the port (8883).* The MQTT broker offers encryption and authentication of username/ password, fig.3.17.

Sixth: *Security settings*

- *Generate a signed certificate.* For this purpose, we can create our own (self-signed) certificate using *Open-SSL.*
- *Cloud protection:* To configure the MQTT client and communicate with the ThingSpeak MQTT broker, we need one of the connection options shown in tab.3.2.

Tab.3.2. *ThingSpeak MQTT* broker, communication options

Port	**Connection Type**	**Encryption**
1883	*TCP*	*None*
8883	*TCP*	*TLS/SSL*
80	*WebSocket*	*None*
443	*WebSocket*	*TLS/SSL*

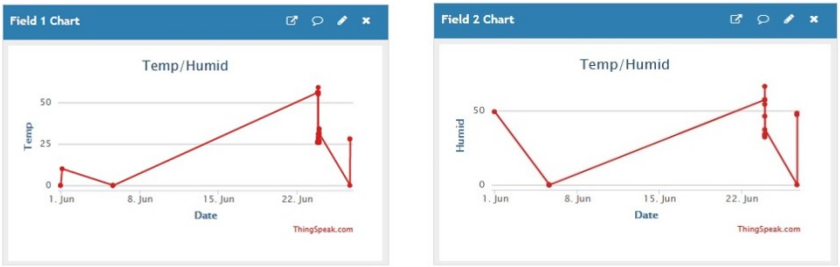


Fig.3.18. Result of sensory data integration in *ThingSpeak*

Seventh: *Check the operability of the algorithm for transmitting encrypted data.*

The operability of the proposed algorithm and the actually achieved data security were checked using Wireshark software. It captures network traffic on the local network and stores data for offline analysis from Ethernet, Wireless (IEEE.802.11), etc. The check result is shown in Fig.3.19 and confirms that the data packets sent from the sensor network to ThingSpeak via MQTT are connected to communication port 8883, which is encrypted via TLS/SSL.

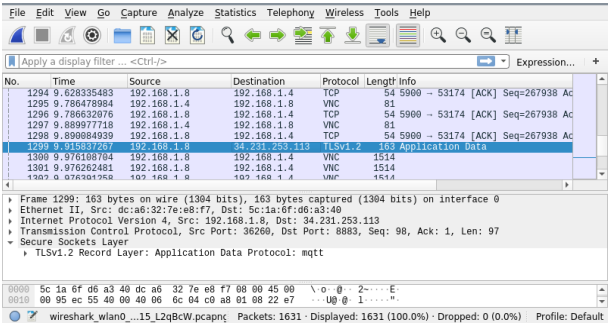


Fig.3.19. The results of the analysis of the package in Wireshark

3.5 Modelling and testing a sensor network for measuring insulin levels [A1]

3.5.4. IoT Diagnostic network model for remote health care through diabetes monitoring based on IoT technology

3.5.4.2. Sensor network architecture for glucose index measurement

The proposed sensor network architecture is cloud-based. Fig.3.21 shows the direct connection of the sensors to the cloud or via a gateway. The generated devices data is transported to a processing server.

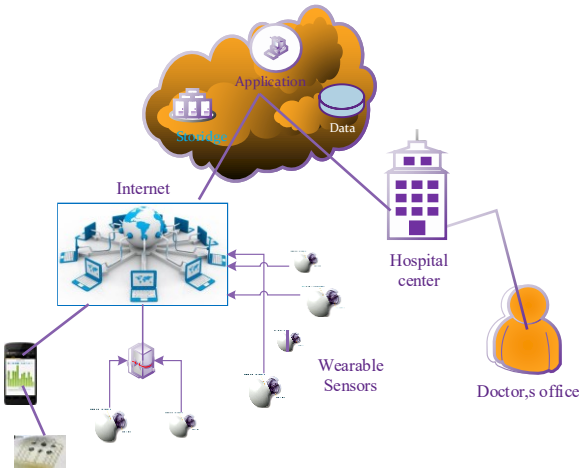


Fig.3.21. Sensor network architecture for glucose index measurement

3.5.4.3. Sensor network model for glucose index measurement

The health model considers an abstract case of a patient with diabetes. The patient uses a health monitoring company (HMC). It carries a device for continuous glucose monitoring (CGM). When patient health data shows dangerous patterns, the HMC sends a signal to the patient's smart device to contact the health centre. In case his condition worsens, an ambulance is dispatched.

3.5.4.4. Algorithm for designing a model for a sensor network *for monitoring insulin levels, based on IoT with simulation software*

First Step: Configure basic network settings—IP addressing of sensors and gateway.

Second step: Connecting the devices to a microcomputer. The devices connect to the home gateway and the IoE Registration Server.

Third Step: Configure a gateway to connect devices to a network. We are connecting IoT devices to build a home network for the patient.

Fourth Step: Create a network and Internet provider. Creating Servers.

Fifth Step: Create mathematical objects to define blood glucose levels and integrate them into the application code.

3.5.4.5. System testing

The designed system consists of intelligent devices with a glucose monitor (CGM). The devices wirelessly send information to the home gateway and the HMC.

A) Simulation of Hyperglycemia. The application sends a signal to the IoT devices, to the medical centre's server, where the data is analyzed and, if necessary, an ambulance is dispatched.

Fig.3.24. Simulation of hyperglycemia

B). Simulation of Hypoglycemia. The application sends a signal to IoT devices that display glucose levels. Data obtained in hypoglycemia are observed in Fig.3.25.

(C) Simulation of normal levels. Simulate normal glucose levels, respiration rate, etc. (Fig.3.26.)

Fig.3.25. Simulation of hypoglycemia

Fig.3.26. Simulation of normal blood sugar levels

The data from the glucose sensor are transmitted to the patient's "smart devices" via MQTT. And for this purpose, Python code has been developed. The settings for broker and client are shown in Fig.3.28, and 3.29 in the dissertation.

The simulation studies show that the designed system can be monitored and managed: insulin levels, respiratory rate, etc. The proposed algorithm, configuration and design of the simulation system allow obtaining a complete picture of the patient's condition. The implementation of this model makes it possible to monitor the health status of patients in real-time, assess their condition, and make decisions about the patient's health status.

3.5.Experimental study of the protocols integration influence from sensor data to the cloud

3.6.1.The experiment objective is to evaluate the integration by studying the influence of the HTTPS, HTTP, MQTT and MQTT-SN protocols from sensor data network integration to the cloud. The integration evaluation can be done by criteria such as support for applications running on different communication models and data rates. The general requirement is reliable data transmission. As a parameter of the integration efficiency, we accept the transmitted data, delay, CPU and RAM load, showing the degree of consumed energy.

3.6.2. The variable parameters are: the number of packets, number of topics per packet, bytes value per topic and their influence on the delay between the generating sensor data and their receipt in the cloud.

3.6.3. Experimental design

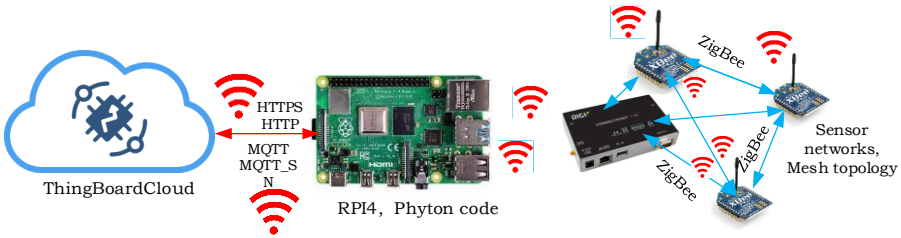


Fig.3.31. Experimental design

A physical XBee sensor network is built. The collected data is transmitted to RPI4, which loads the developed code for the experiment. RPI4 transmits sensor data to the ThingBoard cloud via MQTT, HTTP, HTTPS and MQTT-SN protocols.

The ThingBoard cloud is not designed to access MQTT-SN data. The MQTTSNClient Eclipse Paho library transmits MQTT-SN sensor data, reflecting how MQTTClient works. MQTT-SN requires the MQTT-SN Gateway, which acts as a protocol converter and converts MQTT-SN messages to MQTT messages. These activities require data transformation. The MQTTSNPacket directory contains a library that provides procedures for converting data through serialization and deserialization.

Serialization is the process of transforming data structures or objects into a stream of bytes while maintaining the state of their properties.

3.6.4. *Algorithm and code for conducting the research*

First step*: Setting the sensor nodes in "broadcast" mode for data transmission in the mesh network and RPI4.*

Second step: Develop code in RPI to collect sensory data.

Third step: Setting different parameters for conducting the research: number of transmitted packages, number of topics per packet, bytes value per topic.

Fourth step: Connect to the cloud, generate data and access the storage database.

Fifth step: Visualization and analysis of the data in the cloud.

*3.6.5. Experiment setup*s include cloud, XBee modules, and integration protocol setups. To run the experiment, we need to register in ThingBoard, which accepts HTTP and MQTT connections. The registry directs sensor messages to a Pub/Sub topic with a Cloud Functions endpoint as a subscriber. The cloud function simply writes the payload to the log. The end device works with MQTT and HTTP clients, measuring the response time and tracking sent packets and speed, fig. 3.33.

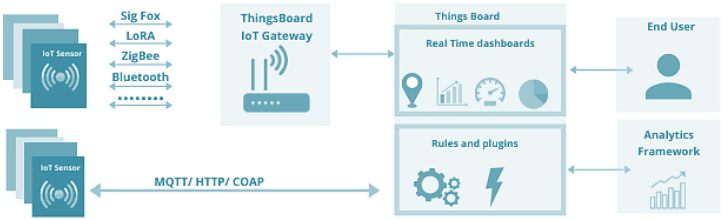


Fig.3.33. ThingBoard Architecture

3.6.6. Experimental study of the packet parameters influence on the transmitted data delay.

The objective of the study is to determine the influence of different parameters such as the number of packets, the number of topics per packet, and bytes value per topic on the speed (delay) of data transmitted to the cloud through different protocols working on different communication models such as HTTP, HTTPS, MQTT and MQTT-SN and evaluate the effectiveness of network - cloud integration.

3.6.6.1.Scenarios for conducting experiments

Tab. 3.3. Scenarios

Scenario	Research protocols	Number of packets	Number of topics per packets	Bytes per topic
1.	HTTP, HTTPS,MQTT MQTT-SN	10	100	10
2.	HTTP, HTTPS,MQTT MQTT-SN	100	100	10
3.	HTTP, HTTPS,MQTT MQTT-SN	200	100	10
4.	HTTP, HTTPS,MQTT MQTT-SN	800	100	10
5.	HTTP, HTTPS,MQTT MQTT-SN	900	100	10
6.	HTTP, HTTPS,MQTT MQTT-SN	1000	100	10
7.	HTTP, HTTPS,MQTT MQTT-SN	200	200	1400
8.	HTTP, HTTPS,MQTT MQTT-SN	200	200	1800
9.	HTTP, HTTPS,MQTT MQTT-SN	200	200	2000
10.	HTTP, HTTPS,MQTT MQTT-SN	200	200	2500
11.	HTTP, HTTPS,MQTT MQTT-SN	200	400	2500
12.	HTTP, HTTPS,MQTT MQTT-SN	400	600	2500
13.	HTTP, HTTPS,MQTT	600	600	800

	MQTT-SN			
14.	HTTP, HTTPS,MQTT MQTT-SN	1000	100	100
15.	HTTP, HTTPS,MQTT MQTT-SN	200	1000	100
16.	HTTP, HTTPS,MQTT MQTT-SN	200	800	100
17.	HTTP, HTTPS,MQTT MQTT-SN	200	400	100

The experiment included 17 scenarios. The data transmission is performed according to four protocols HTTP, HTTPS, MQTT, MQTT-SN and the parameters of the experimental data are given in *Appendix* 3.4 of the dissertation.

3.6.6.2. *study of the transmitted packets number influence delay for the delivered data in different protocols.*

The research is performed according to scenarios №1 to №6, with 100 topics per package and 100 bits per topic.

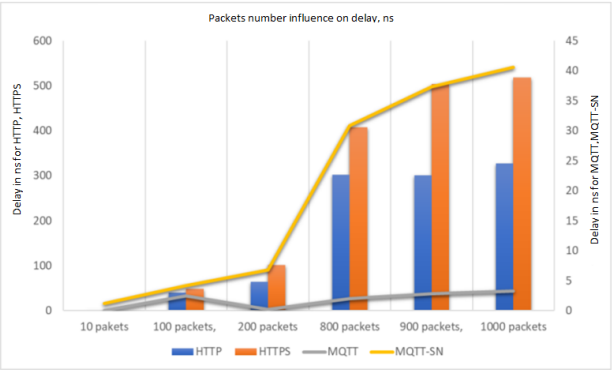


Fig.3.34. Number of transmitted packets influence on delay

The analysis of Fig.3.34 shows that the most significant delay is in HTTPS and the smallest in MQTT by increasing packets number. The delay in MQTT-SN (UDP) is that it does not transmit data directly to the cloud but converts it via the MQTT-SN Gateway to be compatible with MQTT (TCP).

3.6.6.3.Study of the number of transmitted packets influence CPU load

The research is performed according to scenarios №1 to №6, with 100 topics per package and 100 bits per topic.

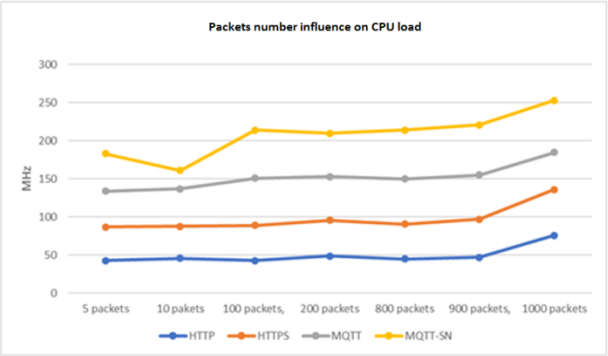


Fig.3.35. The number of packets influences on CPU load

The graphic in Fig.3.35 shows that the CPU load at MQTT-SN is the highest due to the features described above.

3.6.6.4. study of the transmitted packets number influence on RAM

The research is performed according to scenarios №1 to №6, with 100 topics per package and 100 bits per topic.

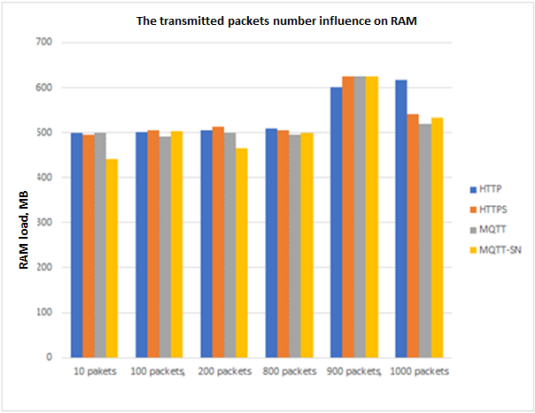


Fig.3.36. The transmitted packets number influences RAM

The analysis of Fig.3.36 shows that as the number of packets increases, so does the RAM load. The highest load is observed with HTTPS due to the use of TLC protection.

3.6.6.5. study of the bit value influence on the delay

The research is performed according to scenarios from №7 to №10, with 200 topics per package and 200 bits per topic.

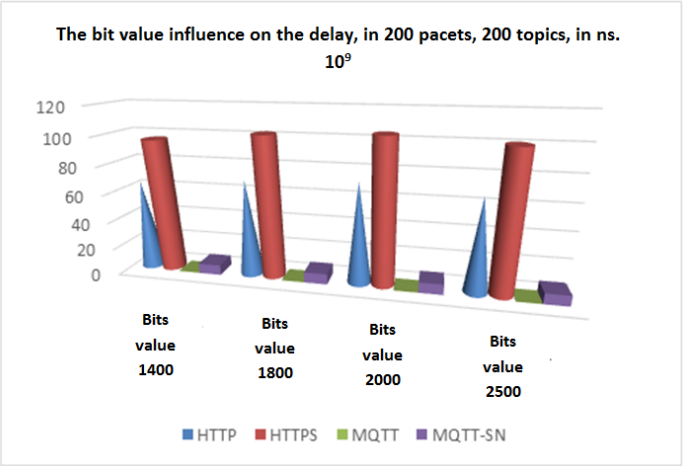


Fig.3.37. The bit value influences the delay

The greatest impact is with HTTPS due to the provision of TLC protection, and the smallest with MQTT.

3.6.6.6. study of the bit value influence on CPU load

The research is performed according to scenarios from №7 to №10, with 200 topics per package and 200 bits per topic.

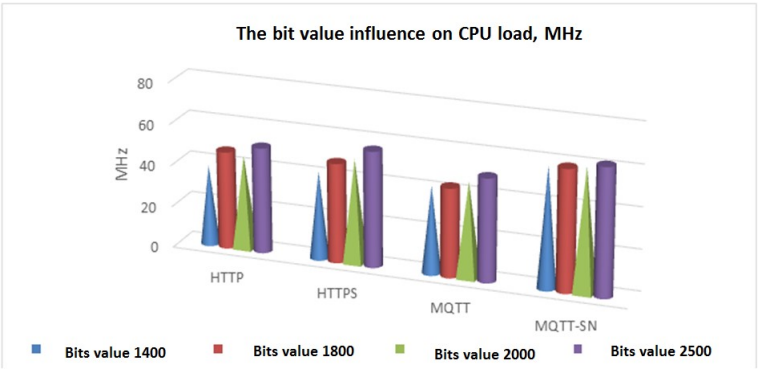


Fig.3.38. The bit value influences CPU load

In Fig.3.38, The increased bit value of the packets affects the MQTT-SN the most due to the properties already described.

3.6.6.7. study of the bit value topics in the package influence on RAM load

The research is performed according to scenarios from №7 to №10, with 200 topics per package and 200 bits per topic.

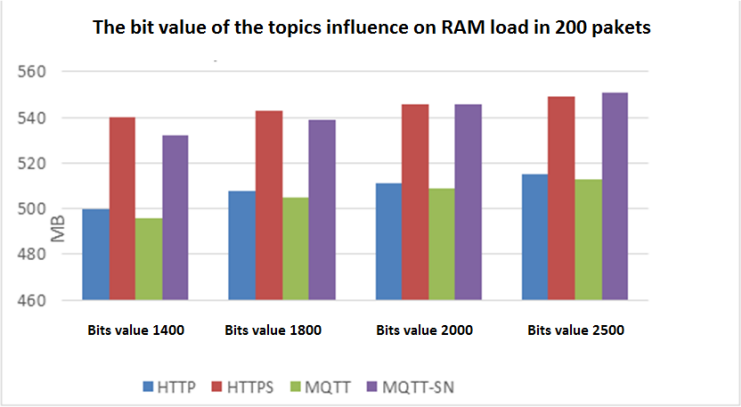


Fig.3.39. The bit value of the topic's influence on RAM load

Figure 3.39 shows that the most RAM load is in MQTT-SN and the smallest in MQTT.

3.6.6.8 Study of the number of topics in the influence of the packet on the delay of the transmitted data.

The research is carried out according to scenarios №15, №16, №17 with 200 topics number in the packages and 100-bit values.

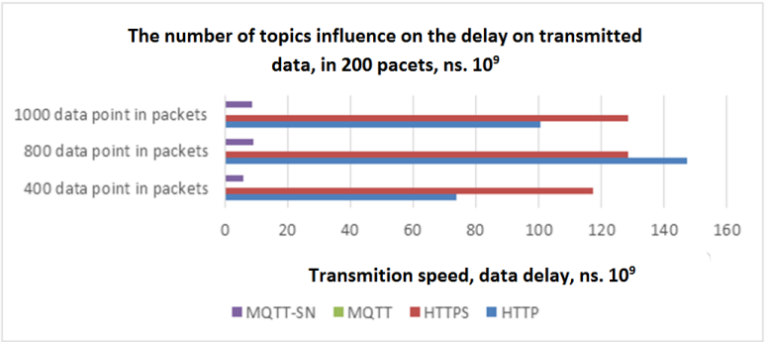


Fig.3.40.The number of topics influences the delay in transmitted data

As the packages number of topics increases, the delay is greatest in HTTPS, and the number of topics in MQTT has a negligible effect on the delay.

3.6.6.9. Study the number of topics in the packages that influence CPU load.

The research is carried out according to scenarios №15, №16, №17 with 200 topics numbers in the packages and 100-bit values.

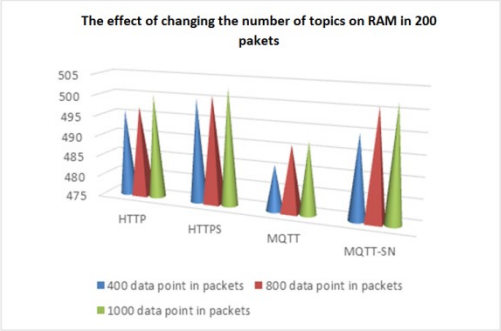


Fig.3.41.The effect of changing the number of topics on RAM

Fig. 3.4, the RAM load is higher for MQTT-SN and HTTPS. The smallest RAM load is for MQTT.

3.6.6.10. Study the number of topics packages that influence the RAM load.

The research is carried out according to scenarios №16, №17, №18 with 200 topics number in the packages and 100-bit values.

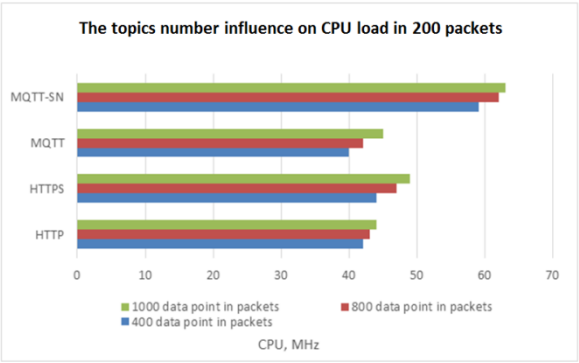


Fig.3.42. The topics number influences CPU load

CPU load is highest at MQTT-SN due to the properties already described. It is noticed that HTTPS has a higher load due to TLC protection. Virtually the smallest CPU load on the MQTT.

3.6.6.11. A study of the parameters of the complex packet influence data delay

The research is carried out according to scenarios №11, №14, №15 and №16.

As the number of packets increases, the delay in data transmission increases, Fig. 3.43. Considering the complex influence of the three parameters, it can be seen that in MQTT, the delay is almost unchanged MQTT-SN is in second place with

minimal influence on the delay. It can be summarised that the biggest delay is in HTTPS.

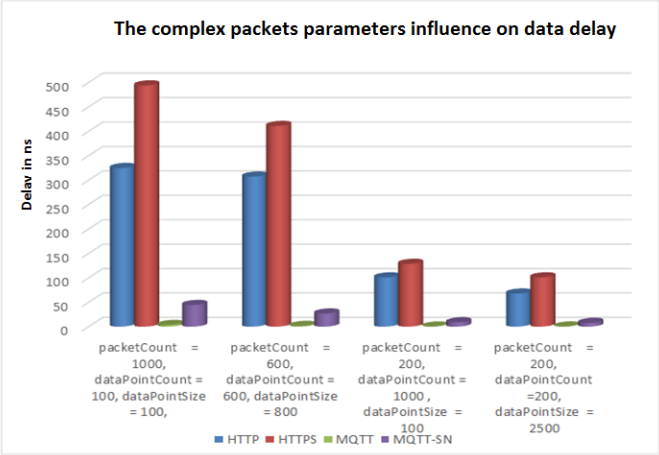


Fig.3.43. The complex packet parameters influence data delay

3.6.6.12. Study of the data parameter's influence on lost packets in HTTP and HTTPS

The research is carried out according to scenarios from №11 to №14.

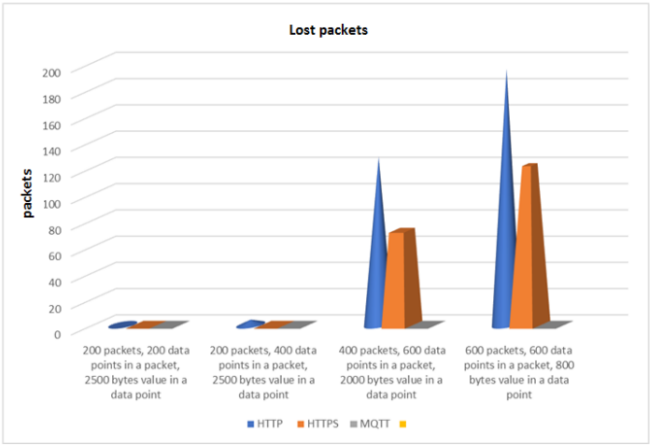


Fig.3.44. The influence of parameters on lost packets

Simultaneous increase of packages, number of topics, and bit value lead to the greatest packets loss in HTTPS.

3.6.6.13. Visualization of data in the cloud structure

Visualization is realized through built-in cloud capabilities. One of them is a historical graph of the uploaded data, Fig. 3.53.

Another is the real-time monitoring of data in the cloud, Fig.3.54.

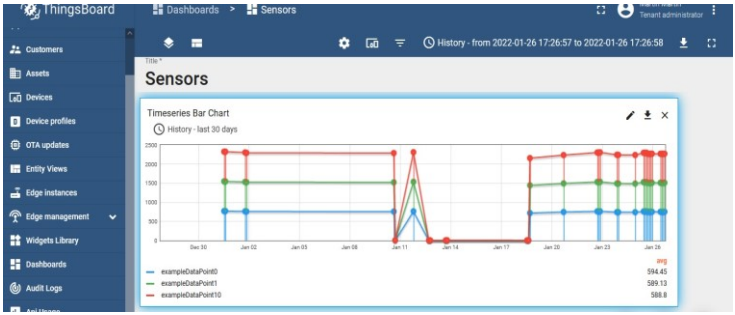


Fig.3.53.Historical schedule of uploaded data for a certain period

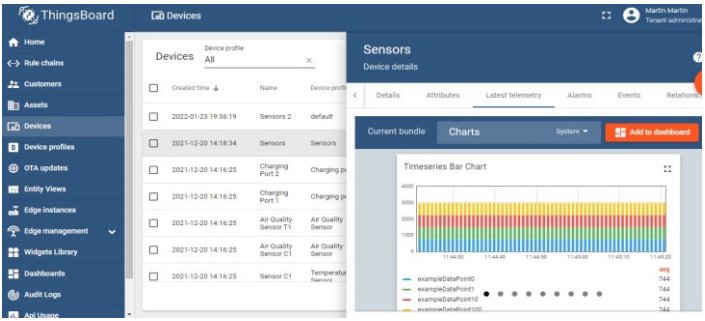


Fig.3.54. Uploaded real-time data monitoring in the cloud

Comparative analysis of experimental studies shows that MQTT provides the smallest delay in data transmission. The smallest CPU and RAM load require the smallest power consumption. Due to their additional processing, HTTPS and MQTT-SN work with more significant delays and higher energy consumption. Evaluating the results of integration experiments, the most efficient data rate and energy consumption are MQTT.

3.7. Conclusions to the third chapter

- A real XBee sensor network is built. The data is visualized, analyzed and sent by the request-response method via REST-based web services and the HTTPS protocol to the ThingSpeak cloud.
- An algorithm for integrating sensor data in the cloud through REST-based web services is proposed and practically implemented.
- Phyton code for data integration in ThingSpeak and Matlab code for visualization and analysis of sensory data in ThingSpeak are developed.
- The choice of TLS protocol for ensuring the security of the transmitted data is justified. Proposed algorithm for

transmitting sensor data via TLS in ThingSpeak. The sensor network-cloud integration is created through the MQTT, and the data stream is created through Node-RED software.

- The operability of the security algorithm of the transmitted data has been verified using the Wireshark tool, which confirms that the data packets sent from the sensor network to the cloud pass through an encrypted communication port and TLS.
- Proposed model of a sensor network for remote health care, through diabetic state monitoring, based on IoT technology.
- An algorithm for model design and configuration of a simulation system for sensory data integration from biosensors for monitoring insulin levels with simulation software has been proposed and implemented.
- The work of the proposed model was verified by simulating hypoglycemia and normal blood sugar levels. Sensory data is successfully transmitted from the patient's smart devices to the simulation server for analysis and decision making.
- A measuring system of XBee devices and ThingBoard has been designed and practically implemented to study the impact of the various protocols for sensor networks on cloud integration and evaluate the efficiency of sensor data transmissions.
- A model for sensor-cloud integration research with different protocols is proposed. Basic communication models for integration - request-response and publish-subscribe are analyzed, and a methodology for implementing the sensor network-cloud integration is developed.
- Python code and algorithm are developed and practically applied for experimental research of HTTP. Wireshark has verified HTTPS, MQTT and MQTT-SN protocols and their operability.
- An experimental study was planned and conducted, including 17 scenarios to determine the performance of HTTP, HTTPS, MQTT and MQTT-SN depending on various parameters such as number of packets, number of topics per packet and bits per topic on the speed (delay) of transmitted data to the cloud.
- It has been established that the MQTT protocol provides the smallest delay in data transmission. The smallest CPU and RAM load, respectively, requires the smallest power consumption and the shortest time for data transmission.

- HTTPS and MQTT-SN have been found to run with greater delay, higher power consumption and CPU load due to additional processing. HTTPS uses the TLS security protocol, which slows down data transmission. MQTT-SN implements a serialization process using the MQTT-SN Gateway to convert data structures or objects to a byte stream, preserving the state of their fields and properties, which explains the more significant delay.
- Evaluating the results of the experiments for sensor network-cloud integration, the MQTT is the most efficient in terms of data transfer rate and energy consumption.

- The obtained results provide useful and practically applicable information on the transmitted data efficiency through the most popular protocols for sensor data integration HTTP, HTTPS, MQTT and MQTT-SN to the cloud structure.

III. CONCLUSION AND MAIN CONTRIBUTIONS

In order to achieve the dissertation objectives, theoretical, simulation, and practical research have been conducted, and results have been achieved, which can be summarised as follows:

- ✓ The main research directions are conducted and systematized, the requirements and challenges facing the wireless sensors and WSN have been defined;
- ✓ The structure of the sensor networks and specifics application of star, tree, mesh, hybrid and cluster topologies are analyzed to improve sensor networks operability;
- ✓ An experimental study of the XBee sensor network's communication efficiency with different topologies has been planned and conducted;
- ✓ An algorithm for studying the communication sensor network's efficiency in different topologies is proposed. The results show that the most effective communication is cluster topology;
- ✓ A simulation study for the sensor network's energy efficiency with cluster topology and LEACH, LEACH-N and LEACH-H was conducted, which established the higher energy efficiency of LEACH-H.
- ✓ There is no universal research methodology of WSN parameters, so it's necessary to develop new models and algorithms for conducting experimental research;
- ✓ An important problem with sensor networks is the generation of huge amounts of sensor data, limited

computing power, capacity, and network life of sensor networks. The sensor networks - cloud structures integration makes it possible to solve this problem without unnecessarily increasing the sensor networks cost;

✓ The thesis is formulated that the creation of models and algorithms for sensor data - cloud integration allows reducing time for development and evaluation of sensor networks efficiency to ensure higher reliability and energy efficiency of designed systems, which is a relevant problem in the sensor networks field;

✓ The main models for sensor data presentation and transmissions, specialized time series TSBD databases for sensor data and problems of sensor data - cloud structure integration are analyzed;

✓ Sensor data storage models have been analyzed: Five-layer OSI-based model and three files, block and object storage models in the data management layer.

✓ The work of the main protocols for sensor networks - cloud platforms integration, HTTP, REST, RESTful, MQTT and MQTT-SN is analyzed, and the differences between them are systematized;

✓ The nature of the sensor network - cloud integration is clarified. The approaches for managing the sensor data security to cloud integration are defined, such as the design and development of specialized network controllers, development of international standards, specialized hardware protection.

✓ The primary data transmission modes from sensors to servers in the cloud structure are defined, and the possibilities for delaying the data transmission are analyzed;

✓ Algorithms for authentication of MQTT based communication are presented, which allows verifying the device authenticity without direct password transmission, as well as the MQTT data protection method, using TLS protocol, which increases the authentication security level for sensor networks.

✓ A real XBee sensor network has been built. An algorithm for sensor data - cloud integration has been implemented, using the request-response method via HTTPS web service protocol, sending data to ThingsSpeak;

✓ Python code for data integration in ThingSpeak and Matlab code for visualization and sensory data analysis in ThingSpeak have been developed;

✓ The integration security of transmitted data has been studied, and experiments were done. An algorithm for checking the transmitted sensor data security via MQTT in the ThingSpeak cloud has been proposed via TLS protocol, and the data stream has been created using the Node-RED software;

✓ The operability of the security algorithm has been verified via Wireshark, which confirms that sented sensor network data packets to the cloud pass through an encrypted TLS communication port;

✓ A sensor network model for remote health care is proposed, through diabetic state monitoring, based on IoT technology;

✓ A design model algorithm has been proposed and implemented for simulation system software configurations, designed to integrate biosensors data for monitoring insulin levels;

✓ The performance of the designed model was tested by simulating hypoglycemia and normal blood sugar levels. Sensory data is successfully transmitted from the patient's smart devices to the simulation server for analysis and decision making.

✓ An experimental XBee device and ThingBoard platform measurement system were designed and practically implemented to study the impact of various protocols for sensor network - cloud integration and sensor data transmission efficiency evaluation was tested.

✓ A model for research of sensor network-cloud integration with different protocols is proposed. Basic communication models for request-response and publish-subscribe integrations are analyzed, and a methodology for implementation of sensor network-cloud integration is developed;

✓ Python code and algorithm have been developed for experimental Research of HTTP, HTTPS, MQTT and MQTT-SN was practically applied. Also, their operability was verified via Wireshark;

✓ An experimental study was planned and conducted, which includes 17 scenarios, determining the operability of HTTP, HTTPS, MQTT and MQTT-SN, depending on various parameters such as number of packets, number of topics per packet and bits value per topic on the speed (delay) of transmissions data to the cloud.

✓ It has been established that MQTT provides the

smallest data transmission delay, the smallest CPU and RAM load, respectively shows the smallest power consumption and the shortest data transmission time;

✓ HTTPS and MQTT-SN have been found to run with higher delay, higher CPU and RAM load, power consumption, respectively, due to additional processing. HTTPS uses TLS security, which explains the higher delay. MQTT-SN implements a serialization process using the MQTT-SN Gateway, which explains the higher delay;

✓ Evaluating the experimental results for sensor network - cloud integration, the most efficient in terms of data transfer rate and the energy consumption is MQTT;

✓ The obtained results provide useful and applicable information on the transmitted data efficiency, through the most popular sensor data integration protocols, like HTTP, HTTPS, MQTT and MQTT-SN to the cloud;

✓ The results of theoretical, practical and simulation research can serve as a useful tool for professionals building sensor network - cloud integration;

CONTRIBUTIONS TO THE DISSERTATION

Scientific and applied contributions

1. An algorithm for studying the communication efficiency of sensor networks in different topologies has been proposed and practically implemented. The results show that packets and delays have minimal losses in the cluster topology.
2. An algorithm for sensor network's energy efficiency with cluster topology and protocols LEACH, LEACH-N and LEACH-H is proposed and studied. The results show that LEACH-H is the most energy-efficient.
3. An algorithm for sensory data - ThingSpeak cloud integrating through REST HTTPS web services protocol using the request-response method has been proposed and implemented.
4. An algorithm for checking the transmitted sensor data security with MQTT in ThingSpeak cloud, using TLS protocol and a data stream implemented with Node-RED, has been proposed, experimented and verified.
5. An algorithm for construction and experimentation was proposed and implemented for the simulation system of biosensors data integration to the cloud, monitoring insulin levels through hyperglycemia, hypoglycemia and normal blood sugar levels simulation.
6. An algorithm for studying the sensor network - cloud integration efficiency through HTTP, HTTPS, MQTT and MQTT-SN protocols have been proposed, implemented and verified.
7. The performance of sensor data - cloud integration of HTTP, HTTPS, MQTT and MQTT-SN is determined and evaluated, depending on various parameters such as number of packets, number of topics per packet and bits value per topic on the transmissions data to the cloud speed (delay).
8. It has been found that the MQTT provides the smallest data transmission delay, CPU and RAM load, and lowest power consumption, respectively. It was found that HTTPS and MQTT-SN work with higher delay and power consumption due to additional processing.

Applicable contributions

9. A model is proposed, and a test information-measuring system is created to study the sensor network - cloud integration, through which an experimental study was conducted, including 17 scenarios.
10 Python codes have been developed for experimental testing of the HTTP, HTTPS, MQTT and MQTT-SN protocols.

.

IV. PUBLICATIONS ON THE DISSERTATION TOPIC

[A1]. **Pandurski M.,** Tsvetanov F., Application of Sensor Networks for Measuring Insulin Levels, International Journal of Online and Biomedical Engineering (iJOE) Vol 16, No 14 (2020) pp.122-136, https://doi.org/10.3991/ijoe.v16i14.17185, https://www.learntechlib.org/p/218476/ *Scopus, Web of Science*

[A2]. F A Tsvetanov and **M N Pandurski**, 2021, Security of the Sensory Data in the Cloud, Journal IOP Conference Series: Materials Science and Engineering, International Scientific Conference of Communications, Information, Electronic and Energy Systems – CIEES 2020, vol. 1032, pp. 012005, *Web of Science*

[A3]. Tsvetanov F., **Pandurski M.**, 2020, Methodology for Integration of Sensors data in the cloud, Fifth International Scientific Conference "Telecommunications, Informatics, Energy and Management", pp 109-113, *Nacid*

[A4]. Tsvetanov F., **Pandurski M**., Some aspects for the integration of sensor networks in cloud structures, Proceedings of International Conference on High Technology for Sustainable Development HiTech 2018 p.p. 249-253.q *Scopus*

[A5]. Tsvetanov F., Georgieva I., **Pandurski M**., 2019, The influence of the topology on the radio communication range of sensor networks, Technium: Romanian Journal of Applied Sciences and Technology (ISSN: 2668-7798) Proceedings of the Technium International Conference Vol. 2 (2020): p.p. 17-24 https://techniumscience.com/index.php/technium/issue/view/3, *Ebsco, Google Scholar, Crossref,* Publons, DOAJ с др.

[A6]. **Pandurski M.,** Storage of sensor data in cloud databases, XXIX International Conference for Young Scientists, 2020

[A7]. **Pandurski M.,** Tsvetanov F., (2020), research of energy resources of the cluster sensor network, "Journal of Engineering Science and Technology Review", Special Issue on Conference in Telecommunications, Informatics, Energy and Management, (ISSN: 1791-2377) vol.13, pp.32-36, *Scopus, DOAJ, Google Scholar, EBSCO*, ACS Publications, CERN

[A8]. Tsvetanov F., **Pandurski M,** 2019, Multidisciplinary Journal of Science, Education and Art, Технически приложения на различни типове сензори в зависимост от измерваната величина, ISSN 1313 – 5236, p.p. 415-427, *Nacid*

[A9]. Filip Tsvetanov, **Martin Pandurski**, Ivanka Georgieva, Grigor Mihajlov, Simulation of the performance of wireless radio controlled fire protection system, 2020 7th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE), pp. DOI: 10.1109/EEAE49144.2020.9279048, *Scopus, Web of Science*

Five of the publications are referenced in Scopus, Web of Science, two in Nacid and two in international secondary databases Google Scholar, EBSCO, Crossref, Publons, DOAJ and others.

South-West University "Neofit Rilski"
Faculty of Engineering

DEPARTMENT of "Communication and computer technique and technologies "

MARTIN NIKOLOV PANDURSKI, M.Sc,

INVESTIGATION OF THE POSSIBILITIES FOR INTEGRATION OF SENSOR NETWORKS IN CLOUD STRUCTURES

ABSTRACT of PhD. THESIS

The current PhD thesis subject is wireless sensor networks (WSN), protocols like Zigbee, HTTP, HTTPS, MQTT, MQTT-SN and cloud platforms. In the dissertation, research tasks are solved, testing the performance of the protocols (HTTP, HTTPS, MQTT, MQTT-SN) regarding their ability to integrate sensor data on cloud platforms and offering a modification of the standard algorithms, offering new models for studying and evaluating protocols' performance, and conducting theoretical and practical research on the most popular representatives of the wireless sensor networks, protocols, and cloud platforms.

Analysis of the results of the studies found significant differences between the performance of HTTP, HTTPS, MQTT and MQTT-SN regarding the data speed transmitted from the sensor network to cloud platforms and energy consumption. MQTT shows the highest-level performance, while HTTPS and MQTT-SN work at a lower-level performance due to their specifications.

The results of the theoretical and practical studies can serve as a useful tool for professionals when designing WSN and choosing a protocol for the integration of sensor data in cloud platforms.