UNIVERSITATEA POLITEHNICA DIN BUCUREȘTI

FACULTATEA DE ELECTRONICĂ, TELECOMUNICAȚII ȘI TEHNOLOGIA INFORMAȚIEI

SISTEM AVANSAT DE SECURITATE BAZAT PE REȚELE WIRELESS DE SENZORI

LUCRARE DE DISERTAȚIE

prezentat ca cerință parțială pentru obținerea titlului de
Master în *domeniul Inginerie electronică, telecomunicații și tehnologii informaționale*
programul de studii de masterat
Ingineria Informației și a Sistemelor de Calcul

Conducător științific
Prof. univ. Dr. Ing. Burileanu Corneliu

Absolvent
Ing. Călin Horia-Alexandru

București
2019

Universitatea "Politehnica" din Bucureşti
Facultatea de Electronică, Telecomunicaţii şi Tehnologia Informaţiei
Programul de masterat **Ingineria informaţiei şi a sistemelor de calcul**

Anexa 2

TEMA LUCRĂRII DE DISERTAŢIE
a masterandului **CĂLIN Al. Horia-Alexandru , 421-IISC**

1. Titlul temei: Sistem avansat de securitate bazat pe rețele wireless de senzori

2. Descrierea contribuţiei originale a masterandului (în afara părţii de documentare) şi specificaţii de proiectare:
- Se va stabili protocolul wireless cel mai potrivit pentru contextul studiat si prezentat, din perspectiva securitatii (criptarea traficului, securizarea cu chei asimetrice), ariei acoperite, numarului de noduri ce pot constitui o retea functionala si eficienta, consumul de energie al nodurilor wireless.
- Se va construi o retea de 10 noduri wireless imperecheati cu acelasi numar de senzori care sa poata fi plasati in cadrul unui apartament sau in cadrul unei institutii, in locuri cheie pentru monitorizarea diversilor factori (miscare, lumina, scurgeri de gaz etc.)
- Datele acumulate de catre nodurile retelei vor fi trimise si stocate in mod real intr-o baza de date conectata la aceeasi retea
- Datele vor fi putea fi accesate, vizualizate si sterse din cadrul unei aplicatii web si din cadrul unei aplicatii Android
- Pe baza valorilor datelor stocate utilizatorul isi va putea stabili diverse alerte de securitate in cadrul celor doua aplicatii puse la dispozitia acestuia.

3. Resurse folosite la dezvoltarea proiectului:
Android, Python, MySQL, PHP, Arduino, WiFi, ESP8266

4. Proiectul se bazează pe cunoştinţe dobîndite în principal la următoarele 3-4 discipline:
MMSI, TIRW, SPD

5. Proprietatea intelectuală asupra proiectului aparţine: U.P.B.

6. Data înregistrării temei: 2019-01-22 11:23:47

Conducător(i) lucrare,
Prof. dr. ing. Corneliu BURILEANU

semnătura: .....

semnătura: .....

Student,
semnătura:.....

Responsabil program master,
Prof. dr. ing. Radu DOGARU

semnătura:.....

Decan,
Prof. dr. ing. Cristian NEGRESCU

semnătura:.....

Cod Validare: **e91accf503**

Declarație de onestitate academică

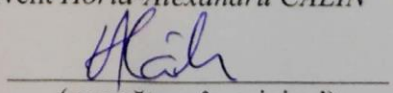
Prin prezenta declar că lucrarea cu titlul *"SISTEM AVANSAT DE SECURITATE BAZAT PE REȚELE WIRELESS DE SENZORI"*, prezentată în cadrul Facultății de Electronică, Telecomunicații și Tehnologia Informației a Universității "Politehnica" din București ca cerință parțială pentru obținerea titlului de *Master* în domeniul *domeniul Inginerie electronică, telecomunicații și tehnologii informaționale* programul de studii *program Ingineria Informației și a Sistemelor de Calcul* este scrisă de mine și nu a mai fost prezentată niciodată la o facultate sau instituție de învățământ superior din țară sau străinătate.

Declar că toate sursele utilizate, inclusiv cele de pe Internet, sunt indicate în lucrare, ca referințe bibliografice. Fragmentele de text din alte surse, reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și fac referință la sursă. Reformularea în cuvinte proprii a textelor scrise de către alți autori face referință la sursă. Înțeleg că plagiatul constituie infracțiune și se sancționează conform legilor în vigoare.

Declar că toate rezultatele simulărilor, experimentelor și măsurătorilor pe care le prezint ca fiind făcute de mine, precum și metodele prin care au fost obținute, sunt reale și provin din respectivele simulări, experimente și măsurători. Înțeleg că falsificarea datelor și rezultatelor constituie fraudă și se sancționează conform regulamentelor în vigoare.

București, *10.06.2019*

Absolvent *Horia-Alexandru CĂLIN*



(semnătura în original)

CUPRINS

LISTĂ DE ACRONIME

A

AP = Access Point

API = Application Programming Interface

B

BLE = Bluetooth Low Energy

BPSK = Binary Phase Shift Keying

BSS = Basic Service Set

C

CPU = Central Processing Unit

CSMA/CD=Carrier-senseMultiple Access/Collision Detection

CSMA/CA=Carrier-senseMultiple Access/Collision Avoidance

D

DSSS = Direct Sequence Spread Spectrum

E

ESS = Extended Servcice Set

F

FFD = Full Functionality Device

I

IBSS = Independent Basic Service Set

IDE = Integrated Development Environment

IoT = Internet of Things

IP = Internet Protocol

P

PAN = Personal Area Network

PWM = Pulse Width Modulation

R

RAM = Random Access Memory

ROM = Read Only Memory

RFD = Reduced Functionality Device

U

UWB = Ultra Wide Band

W

WAP = Wireless Access Point

Wi-Fi = Wireless Fidelity

WPAN = Wireless Personal Area Network

WSN = Wireless Sensor Networks

LISTĂ DE FIGURI

CAPITOLUL 1

INTRODUCERE

Ne aflăm într-o eră modernă în care transmisia datelor se face cu o viteză foarte ridicată, având în același timp nevoie de o mobilitate pe măsură, întrucât majoritatea dispozitivelor capabile de generare a traficului sunt mobile, alimentate cu baterii și capabile să-și schimbe poziția foarte des. În vederea realizării unei rețele de senzori wireless care să fie capabilă de mobilitate și transmisie continuă de date, cu cât mai puține pierderi, în continuare mi-am propus analizarea din punct de vedere teoretic și practic a câtorva dintre cele mai utilizate protocoale wireless la ora actuală. Considerentele care vor fi luate în calcul sunt următoarele - securitatea (criptarea traficului, securizarea cu chei asimetrice), ariei acoperite, numărul de noduri ce pot constitui o rețea funcțională și eficientă, consumul de energie al nodurilor wireless

Se va studia de asemenea arhitectura tipică de client-server în rețelele de date, în vederea folosirii nodurilor pe post de client și o bază de date pe post de server. În cazul de față, nodurile vor generate date pe baza informațiilor pe care le măsoară și le vor stoca, în vederea deciziilor ce pot fi luate plecând de la acestea.

În cadrul lucrării curente mi-am propus prezentarea generală și explorarea proprietăților protocoalelor de comunicații wireless folosite pentru transmisii de date pe distanțe mici și medii, accentuând avantajele lui Wireless Fidelity (Wi-Fi). Acest protocol va fi analizat și comparat cu altele care îndeplinesc funcții asemănătoare și sunt folosite în prezent în producție.

Necesitatea protocoalelor wireless a crescut exponențial în ultimul deceniu, întrucât tehnologiile curente permit folosirea unui număr foarte mare de dispozitive de monitorizare și control pe o suprafață relativ mică. Prin folosirea unui mod de interconectare standard, cu ajutorul firelor, indiferent dacă vorbim despre fire de cupru sau fibră optică, are loc o aglomerare foarte mare a spațiului de lucru, amănunt care poate conduce către diverse probleme: de natură tehnică, prin apariția diafoniilor între cablurile foarte apropiate sau de natură umană, putând să apară confuzii între cablurile ce trebuie conectate la un anumit pin/interfață/port la un anumit moment de timp pentru a îndeplini o anumită funcționalitate. Deși soluțiile wireless au fost explorate încă de la începutul apariției calculatoarelor personale, doar în trecutul recent s-a pus accentul pe folosirea lor într-un număr mare de situații, datorită necesității rezolvării problemelor de mai sus.

Numărul de aplicații în care protocoalele de comunicații wireless și -au găsit sau își pot găsi aplicabilitatea este destul de mare. Acesta a crescut destul de mult în ultima vreme întrucât s-a căutat optimizarea parametrilor de folosire a mediului wireless: viteza de transmisiune, limitarea interferențelor între canale, dimensiunea pachetelor transmise etc. Chiar dacă cel mai probabil nu se va ajunge niciodată la performanțe egale cu cele ale cablurilor de cupru torsadate, coaxiale sau ale cablurilor ce folosesc fibră optică, soluțiile wireless urmăresc să reprezinte o alternativă viabilă pentru comunicațiile pe distanțe medii, fără a crea inconveniente utilizatorului.

Din punct de vedere practic, implementările realizate în cadrul lucrării prezentate au ca scop evidențierea superiorității protocolului Wi-Fi în diverse circumstanțe și de asemenea demonstrarea faptului că este un candidat ideal pentru implementarea în tehnologiile Internet of Things. Scopul final al lucrării este acela de a obține mai multe sisteme ce se pot integra cu ușurință într-o locuință ce adoptă tehnologia Internet of Things pentru a conduce către conceptul de ”home automation”

Peste toate acestea se adaugă un nivel de securitate care va fi implementat pentru a ajunge la un sistem securizat din punct de vedere informatic.

CAPITOLUL 2

REȚELELE WIRELESS DE NODURI

2.1 Evoluția rețelelor wireless de noduri

Conceptul de rețele wireless de noduri se află încă la început, chiar dacă au trecut câțiva ani de când s-a început cercetarea în acest domeniu, respectiv implementarea la nivel mic, crescându-se treptat spre rețele de dimensiuni mai ridicate. Totuși, tendința este aceea de a renunța la nodurile clasice, conectate cu ajutorul firelor și de a migra spre protocoale wireless, mai eficiente din punct de vedere al designului și al implementării. Domeniul în care acest gen de noduri vor avea aplicabilitate este unul foarte vast, existând aplicații dintr-un număr foarte mare de specializări: medicină, supraveghere de securitate, monitorizare mediu înconjurător, militar etc. Prin conectarea într-o rețea a unui număr foarte mare de noduri, obținem posibilitatea să colectăm date legate de fenomene fizice, care ar fi mai greu de monitorizat în moduri convenționale. Probabil, cel mai important domeniu în care această tehnologie va avea aplicabilitate este cel al automatizării – indiferent dacă ne referim la automatizările casnice (pornirea automată a anumitor electrocasnice, deschiderea geamurilor în funcție de diverși parametrii etc.) sau la cele din fabrici (acolo unde poate automatiza întreg procesul de fabricare a diverselor echipamente

Figura 2.1 – Interconectarea echipamentelor în era modernă [4]

Dacă ritmul prezent al evoluției tehnologiei se va păstra și în următorii ani, iar prețul de fabricație al nodurilor va continua să scadă, atunci ne putem aștepta la rețele care să conțină sute, poate chiar mii de senzori interconectați, având același scop final. După cum vom prezenta în capitolele următoare, Wireless Fidelity (Wi-Fi) este singurul protocol capabil să mențină interconectate rețele de dimensiuni atât de mari fără a avea probleme legate de interferențe sau stabilitate a conectivității.

Referindu-ne la diversitatea de aplicații în care își găsesc locul aceste noduri wireless, trebuie luate în considerare și necesitățile ce stau în spatele rețelelor care îi interconectează pe aceștia. Pentru a se îndeplini toate cerințele, de-a lungul timpului au fost propuse mai multe modele de rețele, în jurul cărora au fost construite mai multe modele de protocoale și stive de protocoale. Deși, după cum este exemplificat în cadrul [5] există numeroase moduri în care se poate realiza clasificarea rețelelor de senzori, în continuare scoatem în evidență doar câteva caracteristici ce au un impact major în designul unui protocol.

- Punctele de colectare a datelor (data sink(s)) – în cadrul rețelelor wireless de senzori, avem două tipuri de terminale majore: sursele ce generează date și punctele de colectare, unde datele generate sunt necesare în vederea prelucrării sau trimiterii către sistemele care realizează prelucrarea. În funcție de locația acestor puncte, și intervalul în care acestea au nevoie de date pentru a putea realiza prelucrarea, protocolul trebuie să fie capabil să se adapteze cerințelor.

- Mobilitatea nodurilor - În cazul cel mai general, nodurile sunt presupuse ca fiind imobile. Totuși, după cum vom prezenta și în cadrul părții practice a lucrării, există situații în care trebuie luată în considerare mișcarea senzorilor pentru a satisface

nevoile utilizatorilor: exemplul cel mai relevant stă în domeniul roboticii, unde un robot trebuie să fie capabil să ia decizii pe baza senzorilor atașați, sau în domeniul militar, unde necesitățile de supraveghere se schimbă în mod constant. Mobilitatea influențează într-un mod major protocoalele și serviciile de localizare.

- Resursele nodurilor - În funcție de necesitățile de procesare a fiecărui senzor în parte (aici trebuie luate în considerare mărimile măsurate de către aceștia), memoria internă și frecvența de procesare trebuie adaptate corespunzător. Protocoalele pentru rețelele wireless de senzori trebuie să ia în considerare și aceste detalii în momentul în care sunt proiectate.

- Modele de generare a traficului – În majoritatea aplicațiilor, senzorii sunt în cea mai mare parte a timpului într-un mod de ”așteptare”, generând trafic doar atunci când au loc evenimente de interes. În felul acesta se urmărește o eficientizare a consumului de energie. Există totuși și domenii în care este nevoie de monitorizare și generare continuă de date, cum ar fi supravegherea mediului înconjurător. Evident, între aceste două cazuri nu vor putea fi folosite aceleași protocoale.

După cum se poate observa în cele de mai sus, există un număr mare de criterii ce trebuie luate în considerare în momentul în care este conturat și proiectat un protocol de comunicație.
În continuare ne propunem să prezentăm câteva dintre caracteristicile acestui tip de rețele care oferă avantaje respectiv dezavantaje față de sistemele deja existente în infrastructură.

2.2 Caracteristicile principale ale rețelelor de noduri wireless

Precizăm încă de la început că rețelele de senzori au niște caracteristici ce sunt împrumutate de la rețelele ad-hoc, așa cum este prezentat în cadrul lucrării [1]. Totuși, există numeroase diferențe între acestea întrucât cele despre care discutăm în lucrarea curentă sunt mult mai avansate, având o mulțime de funcționalități de care rețele ad-hoc nu dispun.

Totuși, în cadrul proiectării unui protocol pentru rețele de senzori trebuie luate în considerare câteva dintre proprietățile ”moștenite” de la rețelele ad-hoc. Printre acestea se numără:

1. Constrângerile legate de timpul de viață a unui nod, date de către modul de alimentare. În cadrul lucrării prezente mi-am propus alimentarea unor noduri cu ajutorul bateriilor sau a panourilor fotovoltaice, oferind în felul acesta un timp de viață mai ridicat și o mobilitate crescută. Există câteva constrângeri legate de

folosirea energiei solare pentru alimentare în general, dar acestea nu se aplică în cazul nostru întrucât senzorii, în general, nu consumă foarte multă energie.

2. Probleme în cadrul procesului de comunicație generate de către mediul prin care se face transmisia wireless. Aici facem referire la interferențele ce pot apărea în mediul în care funcționează nodurile – acestea pot fi cauzate de dispozitive ce transmit de asemenea datele într-o manieră wireless și folosesc o frecvență apropiată de cea a protocolului (cum ar fi telefoanele fixe, care folosesc frecvența de 2.4 GHz pentru conectarea la bază).

3. Abilitatea de auto-configurare, fără a fi necesar un grad de intervenție ridicat din partea administratorilor, odată ce sistemul este configurat și funcțional. În momentul în care apare o problemă în cadrul unei rețele, iar datele nu mai pot fi transmise de la nodurile ce realizează achiziția de date către nodul central, reconfigurarea se face în mod automat, fiind căutate alternative.

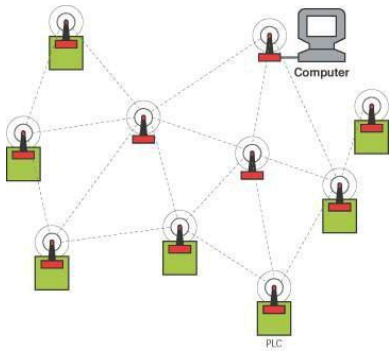


Figura 2.2 – Noduri interconectate wireless [2]

După cum observăm , aceste caracteristici conțin atât avantaje cât și dezavantaje, lăsând deschise numeroase opțiuni de dezvoltare în combaterea dezavantajelor. În cadrul rețelelor de senzori prezentate în lucrarea prezentată, au fost adăugate o serie de caracteristici, prezentate și în [2], la care s-a ajuns plecând de la diverse probleme întâmpinate în implementarea rețelelor ad-hoc.

1. Din punct de vedere al performanțelor, rețele ad-hoc, în forma lor brută nu pot conecta în mod eficient mai mult de 30-40 de senzori, întrucât procesul de comunicație nu este optimizat pentru un număr atât de mare. În WSN-uri, acest lucru este rezolvat de așa natură încât să permită conectarea unui număr de noduri care să ajungă la ordinul miilor.

2. Senzorii în general sunt imobili într-un mediu în care se face monitorizarea diverșilor parametri. Totuși, în cadrul protocoalelor dezvoltate pentru WSN-uri, s-a luat în considerare și posibilitatea deplasării nodurilor, întrucât există domenii în care acest lucru este necesar.

3. Introducerea informațiilor de localizare spațială a nodurilor, relativ la spațiul supravegheat – un detaliu semnificativ pentru mai multe domenii. Ca și exemplu putem considera monitorizarea scurgerilor de gaz – în momentul în care un senzor detectează o astfel de problemă trebuie știut cu exactitate localizarea acestuia.

4. Rețelele de senzori au de cele mai multe ori un model de trafic de tip many-to-one, lucru care poate conduce către probleme de aglomerare a datelor și dificultate în prelucrarea acestora de către nodurile centrale. În WSN-uri aceste probleme se rezolvă prin implementarea mai multor puncte capabile de prelucrare a datelor și cu ajutorul protocoalelor, ce au rolul de a transmite datele către nodul cel mai liber, pentru a nu cauza blocaje. De asemenea, protocolul principal al lucrării, Wireless-Fidelity, folosește CSMA/CA pentru a evita coliziunile în cadrul rețelelor.

5. Dimensiunile pachetelor de date transmise sunt destul de mici, pentru a simplifica procesul de switching între noduri

În procesul de incorporare a tuturor acestor caracteristici și a multor altele în designul unui protocol, trebuie considerat mereu faptul că resursele fizice ale fiecărui nod sunt limitate și trebuie folosite în mod cât mai eficient. În același timp, protocoalele trebuie să fie foarte simple, pentru a executa operațiile într-un timp cât mai scăzut. Toate aceste reglementări au condus către un număr foarte mare de protocoale dezvoltate pentru a lucra între nivelele acces la rețea și transport , fiecare dintre ele având ca și obiectiv principal proiectarea unei rețele de sine stătătoare, care să opereze un timp cât mai îndelungat fără să fie necesară intervenția unui administrator, păstrând un număr cât mai mare de canale de comunicație și o calitate a serviciilor livrată cât mai ridicată.

2.3 Utilizarea senzorilor

Deși în cadrul lucrării îmi propun să implementez mai multe scenarii funcționale în care se va folosi un anume protocol wireless de comunicație, ne dorim să demonstrăm de asemenea și utilitatea acestui protocol în rețelele de senzori wireless pentru lucrări viitoare. În continuare vom prezenta câteva caracteristici și proprietăți ale senzorilor și traductorilor, accentuând diferențele dintre acestea în cadrul unui mediu tehnic.

De-a lungul istoriei ne-a fost dovedit în repetate rânduri că progresele făcute atât în știință cât și în inginerie au reprezentat factori majori în dezvoltarea tehnologiei senzorilor. Un prim exemplu din punct de vedere istoric poate să fie reprezentat de sensibilitatea termică a unei rezistențe, observată la începutul anilor 1800 de către Wilhelm von Siemens și implementată ulterior în dezvoltarea unui senzor de temperatură, bazat pe un rezistor de cupru. Stabilitatea cristalelor de cuarț, precum și proprietățile lor piezoelectrice au jucat un rol foarte important în dezvoltarea senzorilor ce sunt folosiți în zilele noastre în aproape orice aplicație, inclusiv în domeniul militar.

Raportându-ne la trecutul mai apropiat, o noua eră în tehnologiile senzorilor a fost declanșată de către posibilitatea explorării proprietăților siliconului în vederea creării unor noi metode de a transforma fenomene fizice în unele electrice, pentru a putea fi citite și procesate de către un calculator. Cercetările ce se realizează în prezent ne vor oferi un control mai bun asupra proprietăților materialelor și a modului în care acestea se comportă, oferindu-ne astfel perspectiva unor noi generații de senzori, mai eficienți și cu un preț de producție redus.

În ciuda faptului ca există, la momentul actual, o cantitate semnificativă de materiale scrise pentru a descrie senzorii și traductorii, se păstrează în continuare o ambiguitate destul de mare în folosirea unei singure definiții pentru aceștia. Motivul este unul destul de simplu: acest domeniu revoluționar, care este într-o continuă dezvoltare este unul interdisciplinar, presupunând cunoștințe din mai multe domenii. De aici deducem că nu ar trebui să ne surprindă inexistența unei definiții acceptate în unanimitate legate de conceptul unui senzor.

Pentru a putea prezenta importanța senzorilor este suficient să ne îndreptăm atenția către dispozitivele electronice pe care le folosim în viața de zi cu zi, atât în mediul de acasă cât și în cel în care ne desfășurăm activitatea profesională, indiferent de domeniu. Atât ei cât și traductorii sunt prezenți peste tot, chiar dacă uneori suntem conștienți de acest lucru întotdeauna, simplificându-ne viața prin eficientizarea activităților întreprinse, de la cele mai semnificative până la cele minore, cărora nu le dăm suficient de multă atenție. Lucrarea de față își propune prezentarea conceptelor generale a acestor senzori și traductori, împreună cu noile tehnologii care înglobează folosirea senzorilor, precum Internet of Things.

Un senzor, în termeni cât mai generali, conform [3], reprezintă o componentă hardware menită să măsoare o cantitate fizică, pe baza căreia emite o un semnal ce poate fi citit și interpretat de către un observator sau un instrument specializat. Unul dintre cele mai simple de înțeles exemple este reprezentat de către termocuplu, care măsoară temperatura la care este expus, iar pe baza unei referințe, emite o tensiune, ce poate fi citită cu ajutorul unui voltmetru. Din punct de vedere al preciziei, toți senzorii sunt calibrați cu ajutorul unor standarde bine cunoscute și definite.

Traductorul reprezintă un dispozitiv ce este alimentat cu energie de către un sistem, în timp ce la rândul său alimentează un al doilea sistem, dar cu energie prezentată sub altă formă. Exemplul cel mai banal este reprezentat de către un difuzor(Figura 1) care transformă semnalele electrice în energie sonoră. Ne putem referi însă și la termocuplul specificat mai sus, capabil să transforme căldura în energie electrică.

În anumite contexte, termenii de senzor și traductori sunt folosiți ca și sinonime, dar acest lucru depinde mult de aplicația în care întâlnim noțiunile.

Senzorii analogici produc un semnal continuu(sau o tensiune), care în general este proporțional cu cantitatea de informație măsurată. Măsurile fizice precum: temperatura, viteza, presiunea etc. reprezintă toate cantități analogice, întrucât sunt de cele mai multe ori continue. Legându-ne la exemplul de mai devreme, temperatura unui lichid poate să fie măsurată cu ajutorul unui termometru sau a unui termocuplu, care răspunde în mod continuu la schimbările de temperatură, pe măsură ce acesta este încălzit sau răcit.

Traductorii ce se află la ieșirea unui sistem sunt denumiți în mod uzual acuatori (dispozitive de acționare) și convertesc energia electrică într-un alt tip de energie, cum ar fi căldură sau energie mecanică

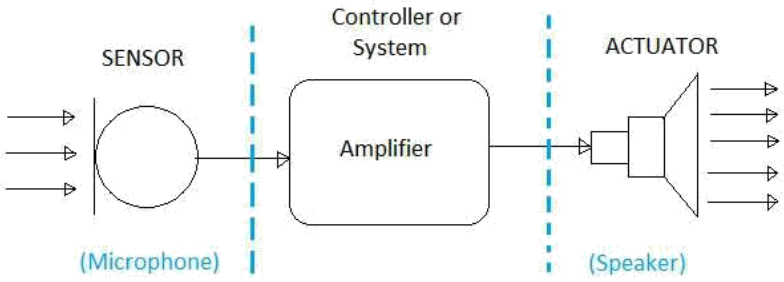


Figura 2.3 – Principiul de funcționare a unui sistem cu senzori [3]

Senzorii digitali au un semnal discret la ieșire, sau o tensiune ce reprezintă cantitatea ce este măsurată. Acest semnal de ieșire poate să fie reprezentat sub forma unui singur bit sau a unei secvențe de biți. Un astfel de senzor este reprezentat, ca și caz general, de către un micro senzor ce are integrat un convertor analog-digital pe o singura pastilă de silicon, pentru a forma o componenta micro-electro-mecanică, ce poate procesa informație sau poate comunica cu un procesor integrat. Aceste caracteristici au dat naștere senzorilor inteligenți din zilele noastre, întâlniți sub denumirea de smart-sensors.

2.4. PROTOCOALE WIRELESS FOLOSITE PENTRU CONECTAREA NODURILOR

La momentul actual, există un număr relativ mare de protocoale wireless ce pot fi folosite pentru diverse aplicații. Acestea sunt prezentate pe larg și detaliat în [4]. În continuare ne propunem să prezentăm o parte dintre aceste protocoale, cele mai folosite și utile din cele existente, făcând o comparație cu protocolul principal al lucrării prezentate – Wireless Fidelity.

Protocoalele pe care le vom analiza în continuare în vederea comparației sunt: ZigBee (IEEE 802.15.4), Bluetooth (IEEE 802.15.1), Ultra-Wide-Band (IEEE 802.15.3) și Wi-Fi (IEEE 802.11). Toate cele patru protocoale au un design general pentru rețele wireless de dimensiuni mici, în care toate dispozitivele sunt conectate pe o distanță maximă de câțiva metri – până la câteva zeci. De asemenea, în cadrul proiectării celor patru protocoale, s-a urmărit un consum cât mai mic de energie, pentru a putea oferi dispozitivelor ce folosesc aceste protocoale, o durată de viață a bateriei cat mai mare (sau a unui ciclu de baterie.) Din punct de vedere obiectiv, aplicațiile țintă ale protocoalelor sunt următoarele:

- Bluetooth – tastaturi, mouse fără fir, căști hands-free etc.
- UWB – legături multimedia ce necesită lățime mare de bandă
- ZigBee – orientat pe conexiune, cu un design special pentru rețelele de monitorizare și control a unui număr mare de dispozitive.
- Wi-Fi – conexiunile de tip PC – PC, cu ajutorul unui intermediar (de cele mai multe ori un wireless AP) – vine ca un substituent al rețelelor clasice, ce folosesc cabluri de cupru (coaxiale, torsadate) sau fibre optice.

În continuare, aceste protocoale vor fi comparate din punct de vedere al standardelor, evaluând atât metricile de funcționare, timpul de transmisiune, eficiența codării, complexitatea și consumul mediu de putere. Vor fi scoase în evidență avantajele fiecărui protocol în parte, dar vom pune accent pe utilitatea lor în rețelele de senzori, precum cele utilizate în cadrul acestei lucrări.

2.4.1 Bluetooth și Bluetooth Low Energy

Bluetooth-ul tradițional nu poate fi comparat cu ZigBee sau alte servicii din suita celor prezentate mai sus, întrucât are un defect major, care l-ar descalifica din start din folosirea sa în rețelele wireless de senzori și anume – consumul ridicat de energie. În cazul în care am avea o aplicație practică care ar avea nevoie de alimentare pe bază de baterie, pe o perioadă mai îndelungată de timp, protocolul Bluetooth tradițional (802.15.1) nu ar putea sa fie folosit. Acesta consumă în jur de 1W, valoare care pentru aplicațiile Internet of Things este foarte mare. Bluetooth Low Energy are un consum de aproximativ 10 – 100 ori mai mic, încadrându-se între 10 – 100 mW [8].

Majoritatea caracteristicilor protocolului BLE au fost împrumutate de la Bluetooth, deci se poate discuta în paralel despre acestea, scoțând în evidență doar diferențele majore de interes.
BLE este folosit pentru rețelele personale (PAN), se bazează pe conexiunile wireless pentru scurte distanțe, având un cost scăzut de implementare. În principiu, acest protocol, împreună cu tradiționalul Bluetooth au fost concepute pentru a scădea numărul de cabluri necesare conectării perifericelor la un calculator personal (tastatură, mouse, boxe, cameră web, imprimantă, scanner etc.)

Figura 2.4 – Echipamente interconectate prin Bluetooth [4]

Există două topologii ce sunt folosite în mod special pentru Bluetooth – piconet și scatternet. Piconet are o structură generală ce permite conectarea unui master și a mai multor periferice secundare (slave). Toate perifericele din cadrul acestei topologii se sincronizează între ele cu ajutorul unui clock – rate setat de către master. Pentru a nu exista interferențe, în momentul inițializării conexiunii, masterul este responsabil cu scanarea mediului înconjurător și stabilirea frecvenței de funcționare, astfel încât să nu apară interferențe cu alte conexiuni. Perifericele sunt responsabile de comunicare punct – la – punct doar cu masterul, nu și între ele. Masterul, pe de altă parte, poate avea conexiuni punct – la – punct sau punct – la – multipunct. Pentru economisirea de energie, un periferic (slave), poate intra în mod de stand-by în momentul în care nu are date de transmis.

Scatternet reprezintă o colecție de piconet ce se suprapun în domeniul frecvenței și din punct de vedere spațial. Un dispozitiv ce folosește Bluetooth se poate conecta în mod simultan la mai multe piconet-uri, permițând în felul acesta un flux de date între acestea, devenind astfel, un scatternet.
BLE are avantajul major de a fi suportat de foarte multe sisteme de operare ce se folosesc în prezent pe dispozitivele mobile: Android, iOS, Windows 8/10 sau macOS. Nu multe alte protocoale se bucură de această incorporare, având nevoie de dispozitive suplimentare pentru comunicare activă.

2.4.2 Ultra-Wide-Band (UWB)

Unul dinte protocoale care s-a bucurat de o majoră creștere a popularității în ultima vreme este UWB. Una dintre caracteristicile căreia i se datorează această popularitate este lățimea de bandă posibilă în cadrul transmisiunii de date – aceasta poate să ajungă până la 480 Mbps, fiind mai mult decât suficientă pentru o multitudine de aplicații ce au nevoie de transmisie de date pe rază scurtă, mai ales într-un mediu închis, casnic. În lipsa interferențelor majore, acesta poate să fie folosit și pe post de înlocuitor al clasicului serial USB 2.0.

Pe lângă acest avantaj se mai adaugă consumul relativ de putere, existând posibilitatea alimentării dispozitivelor cu ajutorul unei baterii, pe o perioadă mai lungă de timp; rezolvarea problemelor de comunicație în cazul apariției mai multor căi de transmisie a datelor și posibilitatea localizării precise a unui nod emițător de date [4].
Totuși, există și un dezavantaj în prezent, la această tehnologie, întrucât durează destul de mult de timp pentru a alege un canal de transmisie. Pe scurt, timpul în care un emițător și un destinatar ajung la sincronizare, este de ordinul milisecundelor. De aceea, este important ca tehnologia de acces la mediul de transmisie (MAC – Medium Access Control) să aibă în vedere acest lucru. Există două protocoale care se ocupă de acest acces – CSMA/CA și TDM.

2.4.3 Wireless Fidelity (Wi - Fi)

În momentul în care se discută despre protocolul Wi-Fi, sunt incluse automat standardele 802.11a/b/g/n/ac/ad, folosite pentru conexiunea într-o rețea locală. Acestea permit conectarea utilizatorului la Internet, când este în apropierea unui AP. Arhitectura generală a protocolului are la bază o celulă ce se numește basic service set (BSS), ce constă într-un set de stații mobile și fixe. Dacă una dintre stații iese din aria de acoperire a BSS-ului la care era conectată, aceasta nu va mai avea conexiune și nu va mai putea comunica direct cu vecinii săi. Mai departe, protocolul IEEE 802.11 a evoluat cu independent basic service set (IBSS) și extended service set (ESS).
Există și posibilitatea interconectării mai multor BSS -uri, cu ajutorul unui sistem distribuit, conducând în felul acesta la apariția unui ESS, cu dimensiuni și complexitate ce pot fi stabilite de către administratorul de rețea.

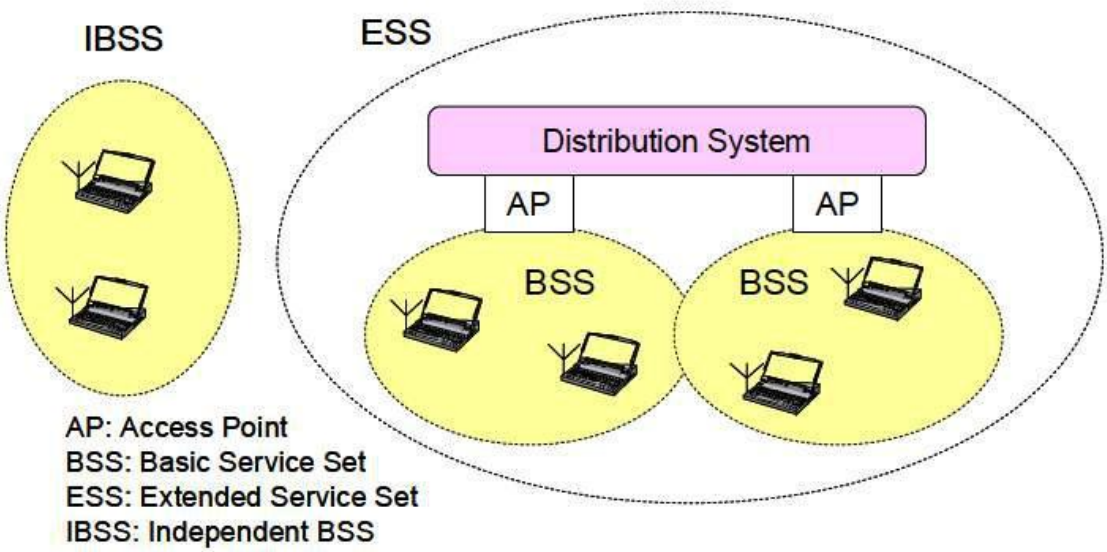


Figura 2.5 – Principiul sistemelor Wi-Fi [4]

Avantajele acestor protocoale sunt viteza foarte mare de transmisie (ultimele versiuni permit traficul de date la o viteza de 1.3 GB/s) și o rezistență ridicată la interferențe cauzate de mediul de transmisiune. Totuși, în cadrul rețelelor wireless de senzori, precum cele ce urmează să fie prezentate, nu viteza este elementul principal ce trebuie urmărit.

Un dezavantaj major este necesitatea echipamentelor intermediare, a căror preț poate crește exponențial în momentul în care se discută conectarea mai multor noduri (peste 10-20). Deși se pot conecta mai multe noduri și la echipamentele ieftine, componentele interne ale acestora (CPU, memorie RAM, ROM), nu fac față volumului de trafic.

2.4.4 ZigBee

ZigBee, așa cum este descris în cadrul standardului IEEE 802.15.4 [7], este un protocol de transmisie de date cu o rată mică de transmisie pentru WPAN. Acesta a fost conceput pentru o conexiune simplă între dispozitive, păstrând consumul de energie la un minim.

Rețeaua cu ZigBee se auto-organizează, fiind necesar un minim de intervenție al utilizatorului sau al administratorului, în momentul configurării inițiale. Intervențiile ulterioare sunt necesare doar în situații cu probleme majore, în care se defectează un număr foarte mare de noduri sau în cazul în care sunt șterse și resetate configurațiile ce rulează. Rețelele organizate de către ZigBee pot fi atât de tip multi-hop, stea (star), cât și de tip mesh (plasă).

Figura 2.6 – Sisteme interconectabile cu ZigBee [13]

În cadrul unei rețele ZigBee, dispozitivele au două moduri în care pot funcționa: dispozitiv complet funcțional (FFD) sau dispozitiv cu funcționare redusă (RFD). Dispozitivele FFD pot îndeplini trei roluri în cadrul unei rețele: coordonator al PAN, coordonator sau periferic. Un dispozitiv FFD poate comunica în activ cu alte FFD-uri sau cu alte RFD-uri, în timp ce un dispozitiv RFD poate comunica doar cu alte dispozitive FFD [11].

Nodurile ce îndeplinesc rolul de dispozitiv cu funcționare redusă au scopuri mai puțin importante în cadrul unei rețele – de cele mai multe ori, au un rol pasiv (întrerupător, senzor pasiv cu infraroșu). Nu au necesitatea de a transmite cantități mari de date și pot comunica cu un singur DCF într -un moment de timp, dar pot face trecerea de la unul la altul.

După ce un nod complet funcțional a fost activat pentru prima oară, este capabil să își formeze propria rețea și să devină coordonator de PAN, formând o rețea stea. Mai multe rețele de tip stea pot opera independent unele de celelalte, separându-se cu ajutorul unui identificator al rețelei, unic în spațiul radio de operare. Odată ce a fost ales un identificator pentru PAN, coordonatorul poate permite conectarea altor noduri în cadrul rețelei.

Datorită avantajelor protocolului (interferențe reduse), posibilitatea criptării datelor, posibilitatea conectării unui număr foarte mare de noduri în cadrul unei singure rețele (> 65000), ZigBee se pretează foarte bine unor aplicații din domeniul Internet of Things, dar și altora mai complexe. Câteva exemple sunt: automatizarea proceselor casnice și de la locul de muncă (pornirea cafetierei, mașinii de spălat, a frigiderului etc.), monitorizare medicală (EEG, EKG), monitorizare a securității și monitorizare seismologică. În cadrul tuturor acestor aplicații, nodurile compatibile ZigBee pot fi alimentate cu baterie, consumul de energie asociat protocolului fiind foarte redus.

În cadrul tabelului 2.1 prezentăm caracteristicile generale ale protocoalelor prezentate, dar și a lui ZigBee, care urmează să fie detaliat în continuare

Tabel 2.1 – Comparație a protocoalelor wireless[2]

Standard	Bluetooth	UWB	Wi-Fi	ZigBee
IEEE spec.	802.15.1	802.15.3	802.11/a/b/g/n/ac/ad	802.15.4
Banda de frecvențe	2.4 GHz	3.1 – 10.6 GHz	2.4 GHz; 5 GHz	868/915 MHz; 2.4 GHz
Rata transmisie pentru semnal maxim	1 Mb/s	110 Mb/s	Până la 1.3 Gb/s	250 Kb/s
Raza acoperită	10m	10m	100m	10 - 100m
Putere de transmisiune	0 – 10 dBm	-41.3 dBm/Mhz	15 – 20 dBm	(-25) 0 dBm
Banda unui canal	1 MHz	500 MHz – 7.5 GHz	22 MHz	0.3/0.6 MHz; 2 MHz
Celula de bază	Piconet	Piconet	BSS	Stea
Extensia celulei de bază	Scatternet	Punct-la-punct	ESS	Plasă
Numărul maxim de noduri	8	8	>20, 25	>65000
Criptarea	E0	AES1	RC4 (WEP[5])	AES

datelor				
Autentificare	Cheie secretă	CBC-MAC[2] (CCM)	WPA2[4]	CBC-MAC (CCM)
Protecția datelor	CRC[3] 16-biți	CRC 32-biți	CRC 32-biți	CRC 16-biți

2.4.5 Concluzii

Fiecare protocol în parte se prezintă cu caracteristici și avantaje ce îl recomandă pentru folosirea în diverse aplicații, dar în același timp, și cu dezavantaje pentru aplicațiile propuse în cadrul lucrării prezentate sau în cadrul proiectelor viitoare de același tip.

După cum am ilustrat în cadrul prezentării de mai sus, ne vom axa asupra protocolului ZigBee și vom demonstra utilitatea acestuia în diverse aplicații, cu o complexitate din ce în ce mai mare. Am ales acest protocol deoarece îl consider ca fiind unul dintre cele mai utile și ”de viitor”, întrucât, spre diferență de orice alt protocol, permite conectarea simultană a până la 65.000 noduri, sau chiar și mai mult decât atât.
Pe scurt, el poate să fie folosit la scară mare, având nevoie de o configurare corectă și la început, astfel încât să reziste diverselor mici probleme ce pot interveni, putând ulterior să funcționeze o durată foarte lungă de timp fără intervenția unui administrator.

În principiu, nu am ales folosirea Bluetooth sau ZigBee deoarece consumul de energie ridicat ar fi făcut protocolul inadecvat pentru lucrul în cadrul rețelelor de senzori. De asemenea, după cum am prezentat deja în tabel, aceste protocoale nu permit conexiunea între dispozitive pentru prea multe dispozitive. Mai mult decât atât, pentru a permite o conectare de tip point – to – multipoint, dispozitivele trebuie să ruleze un software care consumă mai multe resurse hardware interne, iar consumul acestora crește pe măsură ce crește și numărul de dispozitive conectare. ZigBee permite acest tip de conexiuni în mod implicit, iar consumul de energie și resurse nu crește semnificativ odată cu creșterea numărului de dispozitive conectate.

Un alt motiv care m-a împins să aleg Wireless-Fidelity ca și principal protocol descris în cadrul acestui lucrări este reprezentat de raza de acțiune a acestuia – cu echipamentul potrivit, Wi-Fi poate oferi conectivitate eficientă pe o rază de până la 100 metri chiar și în medii indoor. Alte protocoale sunt de asemenea capabile să atingă această distanță, dar întrucât mi-am propus să lucrez în mediul ”Internet of Things”, iar scenariile pe care le voi prezenta se desfășoară în principiu într-un mediu

casnic sau de birou, nu este neapărată nevoie să ajungem la această distanță – în concluzie, nu am avea de ce să consumăm energie și putere de alimentare din moment ce distanța maximă pe care ne propunem să emitem/recepționăm nu depășește 20 – 25 metri.

CAPITOLUL 3

PROTOCOLUL WI-FI ȘI TRANSMITEREA INFORMAȚIEI ÎN REȚEA

3.1. Modelul TCP/IP

Modelul de protocoale TCP/IP pentru comunicații în rețea a fost creat în prima jumătate a anilor 1970 pornind de la necesitatea definirii unei suite de protocoale care să asigure livrarea end-to-end a informației între două calculatoare prin Internet. Acest model definește patru funții ce trebuie îndeplinite astfel încât o comunicație să fie realizată cu succes, după cum este prezentat în Figura 3.1.

Suita de protocoale TCP/IP este un standard disponibil în mod gratuit publicului, ceea ce se traduce prin faptul că orice producător de echipamente este autorizat să implementeze aceste protocoale în hardware-ul sau software-ul pe care îl scoate pe piață.

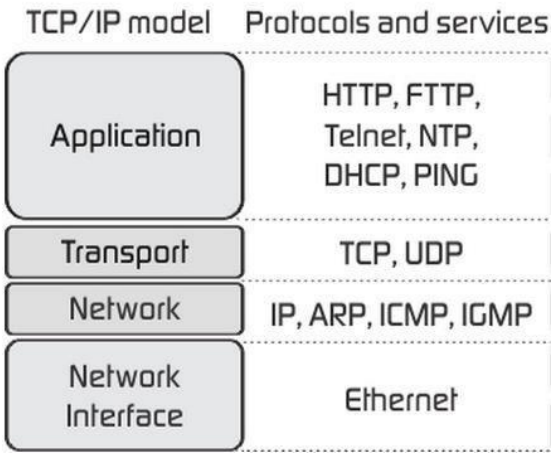


Figura 3.1 [8] - Modelul TCP/IP

3.1.1. Nivelul Aplicație

Nivelul Aplicație este nivelul de top al modelului TCP/IP și este cel mai apropiat de utilizator. Acest nivel include un set de protocoale ce îndeplinesc funcționalități specifice pentru o varietate de aplicații. Dintre acestea putem aminti:

- DNS – rezolvă numele folosite în Internet în adrese IP;
- Telnet – oferă acces de la distanță către servere și dispozitive folosite în rețelistică, precum rutere sau switchuri;
- SMTP – transferă mesajele și atașamentele mail-urilor;
- DHCP – protocol folosit pentru a asigna o adresă IP, masca de rețea, default gateway și adresa serverului DNS unui host în mod dinamic;
- HTTP – transferă fișiere ce aparțin unei pagini web;
- POP – protocol folosit de către clienții de email pentru a descărca emailurile de la un server;
- IMAP – un alt protocol pentru descărcarea emailurilor.

3.1.2. Nivelul Transport

Nivelul Transport este responsabil pentru stabilirea unei sesiuni de comunicare temporară între două aplicații și transmiterea datelor între ele. O aplicație ce rulează pe un host sursă generează date care sunt transmise către o altă aplicație ce aparține unui host destinație, fără a ține cont de tipul clientului destinație, tipul mediului pe care vor circula datele, calea pe care va circula informația sau dimensiunea rețelei. Nivelul Transport este cel care se asigură că datele sunt segmentate și pot fi reconstituite în mod corect la destinație. Astfel, principalele sale funcții sunt:

- Monitorizarea conexiunii între aplicațiile care rulează pe dispozitivele sursă, respectiv destinație;
- Segmentează datele primite de la nivelul Aplicație și reasamblează la destinație segmentele primite în stream-uri de date care sunt transmise mai departe nivelului Aplicație;

- Identifică aplicația căruia îi este destinat fiecare stream de date.

În modelul TCP/IP, procesele de segmentare și reasamblare de la nivelul Transport pot fi îndeplinite de două protocoale: TCP și UDP. Dintre acestea, pentru a implementa comunicația wireless nodurile wireless, eu am ales protocolul TCP din motivele prezentate în subcapitolul 2.2.

3.1.3. Nivelul Internet

Nivelul Internet furnizează patru servicii prin intermediul cărora să ofere dispozitivelor o comunicare eficientă end-to-end:

1. Adresarea dispozitivelor: Un dispozitiv trebuie să fie configurat cu o adresă IP unică pentru a putea fi identificat și adresat într-o rețea. Putem numi host un dispozitiv din cadrul unei rețele ce este configurat cu o adresă IP;
2. Încapsularea: Nivelul Internet primește o unitate de date numită segment de la nivelul Transport. Printr-un proces numit încapsulare, la acest nivel este adăugat un antet cu informații despre IP-ul sursă și destinație, formând astfel un pachet;
3. Rutarea: Nivelul Internet furnizează serviciile necesare pentru a directa pachetele către un host destinație aflat într-o altă rețea. Pentru a ajunge către alte rețele un pachet trebuie să fie procesat și redirecționat de către un ruter. Un pachet poate traversa mai multe astfel de dispozitive intermediare în drumul său către destinație. Fiecare rută intermediară pe care un pachet este nevoit să o traverseze pentru a ajunge la host-ul destinație este numită hop;
4. Decapsularea: Atunci când un pachet ajunge la nivelul Internet al host-ului destinație este verificat antetul pachetului respectiv. Dacă IP-ul conținut în acest antet este același cu propriul său IP, antetul este îndepărtat de pachet printr-un proces numit decapsulare. Unitatea de date astfel obținută este trimisă mai departe către nivelul Transport.

Unul dintre serviciile implementate de către suita de protocoale TCP/IP de la nivelul Internet este numit IP. Acesta furnizează doar funcțiile necesare pentru a transmite un pachet de la sursă la destinație într-un sistem interconectat de rețele. Protocolul nu a fost proiectat pentru a urmări și controla flow-ul de pachete, aceste funcții fiind furnizate de alte protocoale de la alte nivele. Principalele caracteristici ale protocolului IP sunt:

- Connectionless: Nu este stabilită o conexiune cu destinația înainte ca pachetele să fie trimise.
- Best effort (nesigur): Sosirea pachetelor la destinație nu este garantată.
- Independent de mediul pe care sunt transmise datele.

3.1.4. Nivelul Network Access

Nivelul Network Access este responsabil pentru transmiterea cadrelor între nodurile unei rețele printr-un mediu fizic (cu sau fără fir). Acest nivel acceptă pachetele primite de la nivelul superior, le impachetează în unități de date numite cadre pe care apoi le codează ca fiind o serie de biți ce este transmisă pe mediul fizic sub forma unor semnale.

Codarea reprezintă o metodă de conversie a unui flux de date într-un grup de biți – 0 sau 1 - folosiți pentru a furniza un tipar predictibil ce poate fi recunoscut atât de emițător cât și de receptor. În plus, metodele de codare de la nivelul fizic pot furniza și coduri de control, ca de exemplu pentru a marca începutul și sfârșitul unui pachet/cadru. Există numeroase tipuri de codare. Dintre acestea printre cele mai des folosite în rețelistică se numără:

- Codarea Manchester: Bitul 0 este reprezentat printr-o tranziție de la un nivel înalt de tensiune la un nivel de joasă tensiune, iar bitul 1 este reprezentat printr-o tranziție de la un nivel de joasă tensiune la un nivel de înaltă tensiune.
- Non-Return to Zero: Această metodă de codare este caracterizată de două stări numite „zero” și „unu”. Un 0 poate fi reprezentat ca un nivel de tensiune, iar un 1 poate fi reprezentat ca un alt nivel de tensiune diferit. Uzual, pentru 0 se foloseste nivelul de tensiune de 0V, iar pentru 1 nivelul de tensiune de 5V.

Nivelul Network Access trebuie să genereze semnalele electrice, optice sau radio prin care sunt transmise datele. Standardele de la acest nivel trebuie să definească ce tip de semnal reprezintă un 1 și ce tip reprezintă un 0. Semnalele pot fi transmise în două moduri:

- Asincron: Semnalele sunt transmise fără un semnal de ceas asociat. Spațierea temporară dintre două blocuri de date poate fi de durată arbitrară, nefiind asociată unui semnal de ceas. De aceea, cadrele necesită indicatori de start și stop.
- Sincron: Semnalelor de datele este atașat un semnal de ceas, biții fiind transmiși la intervale egale de timp.

De asemenea, nivelul Network Access controlează modul în care datele sunt transmise și recepționate de pe mediul de transmisiune și execută detecții de erori pentru a asigura integritatea cadrelor transmise.

3.2. Protocolul TCP

Nivelul Transport este responsabil din punct de vedere al siguranței comunicației. La nivelul rețea, protocolul IP este preocupat doar de structura, adresarea și rutarea pachetelor. IP nu specifică în ce mod este efectuat transportul pachetelor, acesta fiind rolul nivelul Transport. Modelul TCP/IP ne pune la dispoziție două protocoale la nivelul Transport, TCP și UDP.

Am ales protocolul TCP pentru realizarea comunicației dintre nodurile wireless și AP deoarece, spre deosebire de UDP, TCP este considerat un protocol fiabil și are toate caracteristicile nivelului Transport. TCP include procese care să ne asigure că informația transmisă între nodurile wirless ajunge la destinație, prin metode de livrare confirmată.

Folosind protocolul TCP ne asigurăm că sunt îndeplinite următoarele trei funcții definitorii unui protocol fiabil de nivel Transport:

- Urmărirea (trackingul) segmentelor de date transmise;
- Confirmarea (acknowledgementul) datelor primite;
- Retransmiterea oricăror date ce nu au fost confirmate ca fiind primite.

Protocolul TCP împarte un mesaj în unități de date denumite segmente. Segmentele sunt apoi numerotate în ordine și pasate mai departe către protocolul IP care la rândul său adaugă acestora antente cu adresele sursă și destinație, formând astfel pachete. TCP urmărește numărul de segmente transmise către un host specific de la o aplicație specifică. Dacă expeditorul nu primește înapoi un mesaj de confirmare într-o anumită perioadă de timp, acesta presupune că segmentul respectiv a fost pierdut și retrimite doar acel segment. Pe dispozitivul receptor, protocolul TCP este responsabil de reasamblarea segmentelor în ordine și transmiterea lor către nivelul Aplicație. TCP a fost inițial descris în RFC 793 [6] . În adiția suportării funcțiilor de bază de segmentare și reasamblare, TCP furnizează, de asemenea, și următoarele funcții:

- ✓ Conversații orientate pe conexiune (connection-oriented) prin stabilirea de sesiuni;
- ✓ Livrare reliable;
- ✓ Reconstrucție ordonată a segmentelor
- ✓ Control al fluxului de date (flow control).

TCP este un protocol orientate pe conexiune. Un astfel de protocol negociază și stabilește o conexiune permanentă (sau sesiune) între sursă și destinație înainte de a trimite orice trafic. Prin stabilirea unei sesiuni dispozitivele negociază cantitatea de trafic ce poate fi transmisă la un moment de timp, iar datele ce sunt transmise între cele două dispositive pot fi controlate. Sesiunea este terminată doar atunci când toate datele au fost transmise cu succes. Pentru a stabili o conexiune, fiecare dintre dispozitivele între care se realizează conexiunea trebuie să transmită un

semnal de SYN (Synchronize) și să primească un mesaj de ACK (Acknowledgment) de la celalalt dispozitiv. De aceea, conceptual vorbind, trebuie să avem patru mesaje de control ce sunt schimbate între dispozitive.

Totuși, este ineficientă transmiterea semnalelor SYN și ACK în mesaje separate când ar putea fi ambele transmise simultan. Acesta este motivul pentru care în succesiunea normală de evenimente din stabilirea conexiunii unul dintre semnalele SYN și unul dintre semnalele ACK sunt transmise împreună în același mesaj numit adesea SYN+ACK. Însumând, avem astfel un număr total de trei mesaje și de aceea acest proces poartă denumirea Three-Way-Handshake. Acest proces este prezentat conceptual în Figura 3.2 [7] de mai jos.

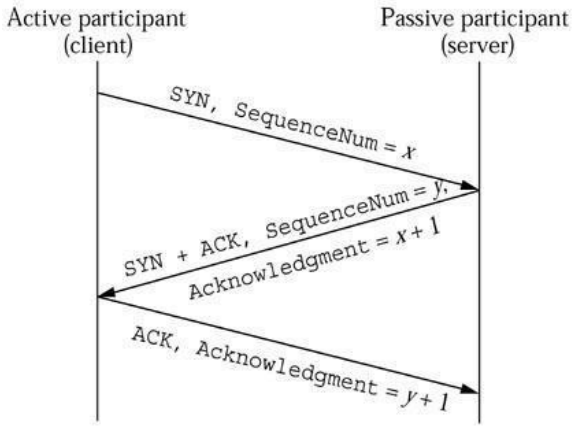


Figura 3.2 [1] – TCP Three-Way-Handshake

În termini de rețelistică termenul de reliability înseamnă asigurarea că fiecare unitate de date transmisă de sursă ajunge la destinație. Din motive diverse este posibil ca o unitate de date să fie coruptă sau pierdută complet atunci când este transmisă prin rețea. TCP poate asigura că toate informațiile ajung la destinație prin retransmiterea de la sursă a unităților de date corupte sau pierdute.

Deoarece rețelele pot furniza mai multe rute pe care să circule pachetele, fiecare cu rate de transmisie diferite, datele pot ajunge la dispozitivul destinație în ordinea greșită. Prin numerotarea și secvențializarea segmentelor, protocolul TCP se asigură că segmentele sunt reasamblate la destinație în ordinea corectă.

Dispozitivele conectate la rețea au resurse limitate, precum memorie sau lățime de bandă. Atunci când TCP observă că resursele sunt suprasolicitate poate cere ca aplicația care transmite date să

reducă rata de transmisie, asigurând în acest mod controlul fluxului de date prin care pot fi prevenite pierderile de segmente pe rețea și evitarea retransmiterii segmentelor pierdute.

3.3. Concepte Wireless

Pentru a asigura flexibilitatea comunicării dintre nodurile wireless și controlul de la distanță al acestuia am ales ca mediul de transmisie de la nivelul Network Access al stivei TCP/IP să fie unul wireless.

Rețelele Wireless pot fi clasificate în [8]:

- Wireless Personal-Area Networks (WPAN) – Operează pe o distanță de ordinul metrilor. În acest tip de rețea sunt folosite dispozitive ce folosesc tehnologii precum Bluetooth sau Wi-Fi Direct pentru a comunica.

- Wireless LANs (WLANs) – Operează pe o distanță de câteva sute de metri, motiv pentru care am folosit acest tip de rețea în realizarea comunicație wireless.

- Wireless Wide-Area Networks (WWANs) – Operează pe distanțe mult mai mari, precum arii metropolitane.

- Pentru conectarea dispozitivelor la aceste rețele există mai multe tipuri de tehnologii, printre care se numără:

- Bluetooth – Definit prin standardul WPAN IEEE 802.15, tehnologie ce folosește un proces de creare a perechilor de dispozitive pentru a comunica pe distanțe de până la 100 de metri.

- Wi-Fi – Definit prin standardul WLAN IEEE 802.11, folosit pentru transferul date, voce sau video pe distanțe de până la 300 de metri.

- WiMAX – Definit prin standardul WWAN IEEE 802.16, tehnologie folosită pentru comunicațiile pe distanțe mult mai mari, până la aproximativ 50 de km.

- Date mobile – 2G/3G/4G

- Comunicații prin satelit – Tehnologie ce furnizează accesul la rețea a zonelor izolate prin intermediul unui satelit geostaționar. Este de obicei o tehnologie mult mai scumpă și necesită o cale liberă, fără obstacole, între satelit și zona izolată care dorește acces la internet.

- Comunicațiile wireless operează în domeniul undelor radio (cu frecvențe între 3HZ și 300GHz) ale spectrului electromagnetic. Domeniul undelor radio este la subdivizat la rândul său în secțiunea frecvențelor radio și secțiunea microundelor. Tehnologiile Bluetooth, WLANs, datele mobile și comunicațiile prin satelit operează toate în secțiunea microundelor caracterizată de benzi frecvențe UHF, SHF și EHF.

Pentru a se putea conecta la rețeaua wireless, dispozitivele au incorporate sau atașate transmițătoare și receptoare reglate la o frecvență specifică a domeniului undelor radio. Concret, următoarele benzi de frecvențe sunt alocate rețelelor wireless descrise prin standardul 802.11:

- 2.4 GHz (UHF) – 802.11 b/g/n/ad
- 5 GHz (SHF) – 802.11 a/n/ac/ad
- 60 GHz (EHF) – 802.11 ad

Standardul WLAN IEEE 802.11 [9] definește cum sunt folosite benzile de frecvențe radio la nivelul Network Access al modelului TCP/IP. Variații ale acestui standard au fost dezvoltate de-a lungul timpului, printre care putem număra:

- 802.11 – Emis în anul 1997 și acum scos din uz, acest standard este versiunea originala a specificațiilor WLAN care opera la frecvența de 2.4 GHz și oferea viteze de până la 2 Mb/s. La momentul emiterii acestui standard rețelele LAN cu fir ofereau viteze de aproximativ 10 Mb/s, motiv pentru care această nouă tehnologie wireless nu a fost adoptată cu entuziasm.

- IEEE 802.11a – Emis in anul 1999, acest standard opera în banda de frecvențe de 5GHz, bandă mai puțin aglomerată decât cea de 2.4 GHz, și oferea viteze de până la 54 Mb/s. Deoarece acest standard operează la frecvențe mai înalte, are o arie de acoperire mai mică și este mai puțin eficace în străpungerea structurilor unei clădiri. Dispozitivele ce operează folosind acest standard nu sunt compatibile cu cele care folosesc standardele 802.11b și 802.11g.

- IEEE 802.11b – Emis în anul 1999, acest standard operează în banda de frecvențe de 2.4 GHz și oferă viteze de aproximativ 11 Mb/s. Dispozitivele ce folosesc acest standard au o arie de acoperire mai mare și sunt mai eficiente în folosirea lor în clădiri datorită faptului că structurile clădirilor nu atenuează așa tare semnalul.

- IEEE 802.11g – Emis în anul 2003, acest standard operează în banda de frecvențe de 2.4 GHz și oferă viteze de până la 54 Mb/s. Astfel, dispozitivele ce folosesc acest standard operează în aceeași bandă de frecvențe ca cele care folosesc standardul 802.11b, dar cu viteza celor care folosesc standardul 802.11a. Totuși, în moemntul în care este suportat un client 802.11b, viteza de transmisie este redusă.

- IEEE 802.11n – Emis în anul 2009, acest standard operează atât în banda de transfer de 2.4 GHz, cât și în cea de 5 GHz. Vitezele de transfer variază de între 150 Mb/s și 600 Mb/s pe o distanță de până la 70 de metri. Acest standard este compatibil cu dispozitivele 802.11 a/b/g. Totuși, suportând un mediu mixt viteza de transfer este redusă.

- IEEE 802.11ac – Emis în anul 2013, operează în banda de frecvențe de 5GHz și furnizează viteze de transfer între 450 Mb/s și 1.3 Gb/s. Acest standard este compatibil cu versiunile 802.11 a/n.

- IEEE 802.11ad - Emis în anul 2014 și cunoscut sub numele WiGig, acest standard oferă o soluție ce operează în trei benzi de frecvențe: 2.4GHz, 5 GHz și 60 GHz și oferă viteze teoretice de până la 7 Gb/s. Este compatibil cu dispozitivele care folosesc toate celelalte versiuni ale standardului 802.11, descrise anterior.

În realizarea comunicației wireless între nodurile wireless ce alcătuiesc rețeaua prezentată am folosit standardul 802.11g deoarece reprezintă o soluție potrivită atât din punctul de vedere al vitezei de transfer, al distanței pe care poate opera, precum și al benzii de frecvențe în care funcționează.

Standardul IEEE 802.11 identifică două tipuri de topologii wireless:

- Ad-hoc – Modelul în care două dispozitive sunt conectate wireless între ele, fără nevoia suplimentară a unui echipament de infrastructură suplimentar, precum un router.
- Modelul de infrastructură – În acest model doi clienți wireless se conectează între ei prin intermediul unui echipament suplimentar.

Într-o rețea wireless de tip ad-hoc două dispozitive comunică între ele într-o manieră peer-to-peer (P2P) fără a folosi echipamente suplimentare, motiv pentru care am ales acest tip de topologie în realizarea comunicației dintre echipamente. În vederea obținerii acestui tip de topologie am stabilit o legătură de tip client-server între dispozitive.

Pentru ca dispozitivele wireless să comunice între ele într-o rețea, acestea trebuie să se asocieze cu un AP sau cu un router wireless. O parte importantă a procesului 802.11 este reprezentată de descoperirea unui WLAN și apoi conectarea la acesta, proces ce se realizează în trei etape:

1. Descoperirea unui nou AP wireless;
2. Autentificarea la AP;
3. Asocierea cu AP-ul.

Pentru a se asocia, clientul wireless și AP-ul trebuie să stabilească de comun accord anumiți parametri care trebuie configurați pe AP și apoi pe client pentru a activa procesele enunțate mai sus:

- SSID – Un SSID reprezintă un identificator unic pe care clienții wireless îl folosesc pentru a distinge între mai multe rețele wireless în aceeași vecinătate. Numele SSID-ului apare în lista de rețele wireless disponibile pe client. Depinzând de configurarea rețelei, mai multe AP-uri pot avea același SSID care uzual este un nume ce cuprinde între 2 și 32 de caractere. În cazul rețelei noastre, având în vedere elementele hardware folosite, am putut configura orice SSID am vrut noi.

- Parola – Deseori numită și cheie de securitate, parola este necesară clientului pentru a se autentifica la AP. Aceasta previne atacurile de securitate prin prevenirea accesului la rețeaua wireless a utilizatorilor nedoriți. În cazul prezentei lucrări, am setat această parolă implicit "password".

- Modul de rețea – Se referă la standardele WLAN 802.11 a/b/g/n/ac/ad. AP-urile și ruterele pot opera într-un mod mixt în care pot fi folosite mai multe standarde simultan. În realizarea comunicației wireless dintre nodurile wireless ale rețelei am folosit modul de rețea 802.11g.

- Modul de securitate – Se referă la setările parametrilor de securitate, precum WEP, WPA sau WPA2. Toate ruterele wireless și AP-urile trebuie securizate la cel mai înalt nivel.

Tipurile None și WEP ar trebui evitate sau utilizate doar în cazul în care securitatea rețelei nu este de interes. Pentru a oferi comunicației wireless dintre nodurile noastre securitatea necesară, am ales modul de securitate WPA-Personal.

Setările de canal – Fac referire la banda de frecvență folosită pentru a transmite date wireless. Ruterele wireless si AP-urile pot alege setările de canal în mod dinamic sau acestea pot fi setate manual pentru a evita interferențele cu un alt AP sau router wireless. Cum implicit standardul 802.11g operează în banda de frecvențe 2.4 GHz, alegând acest standard și mod de rețea, implicit banda de frecvențe în care este realizată comunicația wireless dintre nodurile wireless este de 2.4 GHz.

Figura 3.3 - Parametri de asociere wireless

În Figura 3.3 sunt prezentați parametrii de asociere wireless așa cum apar pe client înainte de asocierea cu AP-ul de pe nodurile wireless.

Clienții wireless se conectează la AP folosind un proces de scanare care poate fi:

- Pasiv – AP-ul anunță public serviciul său prin trimiterea periodică a unor cadre broadcast ce conțin SSID-ul, standardele suportate și setările de securitate. Principalul scop al acestor cadre este acela de a permite clienților wireless să învețe ce rețele și AP-uri sunt disponibile într-o arie dată și astfel să aleagă ce rețea sau AP să folosească.

- Activ – Clienții wireless trebuie să știe numele SSID-ului. Clientul wireless inițiază procesul prin trimiterea broadcast a unui cadru cerere pe mai multe canale, cadru ce conține numele SSID-ului și standardele suportate. Modul activ poate fi necesar dacă un AP sau ruter wireless este configurat să nu trimită cadre broadcast prin care să se facă public.

În realizarea comunicației dintre nodurile wireless am folosit modul pasiv de scanare a rețelelor wireless disponibile, mod în care periodic AP-ul – nodul central trimite cadre de tip broadcast în care anunță existența sa. Astfel, pe client va apărea în lista de SSID-uri disponibile și SSID-ul nodurilor fără a mai fi necesară o cerere suplimentară din partea clientului - nodul wireless.

După ce SSID-ul nodului central (sau al ruterului) îi este disponibil, pentru a se conecta la AP, clientul trebuie să se autentifice. Standardul IEEE 802.11 a fost inițial dezvoltat cu două tipuri de autentificare:

- Autentificare deschisă – Fundamental o autentificare de tip NULL în care clientul wireless spune AP-ului "autentifică-mă" iar AP-ul răspunde cu "da" fără a fi nevoie de o parolă. Acest tip de autentificare furnizează conectivitate wireless oricărui dispozitiv dorește să se conecteze și ar trebui folosit doar în situații în care securitatea datelor nu este de interes.

- Autentificare cu cheie – Tehnică bazată pe o cheie care este pre-distribuită între client și AP.

Din motive de securitate IT, pentru a nu lăsa utilizatori nedoriți să se conecteze la AP-ul folosit în cadrul lucrării, evitând astfel incidente precum furtul de informație – datele măsurate, alertele trimise sau cheile utilizate, am ales modul de autentificare cu cheie pre-distribuită, în care cheia de acces la AP este parola "TelAcad2019".

Procesul de autentificare este următorul:

1. Clientul wireless – nodul wireless– trimite un cadru de autentificare către AP-ul folosit în cadrul implementării practice
2. AP-ul răspunde cu un text provocare (challenge) clientului.
3. Clientul criptează textul primit de la AP folosind cheia pre-distribuită și trimite AP-ului textul criptat.
4. AP-ul decriptează textul criptat folosind aceeași cheie.
5. Dacă textul decriptat primit de la client corespunde textului provocare trimis de AP, atunci AP-ul autentifică clientul. Dacă cele două mesaje nu corespund, atunci clientul nu este autentificat și accesul wireless este blocat.

După ce clientul este autentificat, AP-ul continuă procesul de asociere, la sfârșitul căruia setările sunt finalizate iar legătura de date este stabilită între nodurile wireless și AP-ul utilizat. Etapa de asociere constă în următorii pași:

1. Clientul wireless trimite un cadru cerere de asociere ce include adresa sa MAC.

2. AP – ul trimite un răspuns de asociere ce include BSSID-ul AP-ului – adresa MAC a AP-ului dronei.

3. AP-ul mapează clientului un port logic cunoscut ca identificator de asociere. Identificatorul de asociere este echivalent unui port pe un switch și permite urmărirea cadrelor destinate clientului, precum și transmiterea acestora.

La finalul acestor procese, după ce clientul a fost asociat cu AP-ul, informația poate fi transmisă în siguranță între AP și nodurile wireless care alcătuisesc rețeaua.

3.4. Asigurarea securității informației

Securitatea informației a fost mereu o preocupare mai mare în cazul Wi-Fi deoarece limita rețelei a fost mutată. Este nevoie ca problemele de securitate ce pot apărea în cazul comunicației cu o rețea de senzori wireless– furt de informație, schimbarea traseului acesteia, denial of service etc. – să fie adresate.

Pentru a înțelege mai bine riscurile existente din punct de vedere al securității datelor, pot fi enumerate următoarele tipuri de atacuri informatice aplicabile comunicației wireless dintre punctul central și nodurile wireless:

- Recunoașterea - Acesta este procesul de descoperire folosit pentru a găsi infirmații despre rețea și a determina potențiale vulnerabilități și reprezintă de obicei primul pas într-un atac de securitate. Acest proces poate include scanări ale rețelei pentru a afla ce adrese IP răspund, precum și scanări ulterioare pentru a vedea ce porturi sunt deschise.

- Escalarea Privilegiului - Acesta este un proces de obținere a unui nivel de acces autorizat sau nu în urma căruia poate fi obținut un nivel de acces și mai mare. Un exemplu ar fi acela în care atacatorul obține acces la rețeaua dintre nodurile wireless, în urma căruia poate escala accesul la toate informațiile deținute de stația de control. În cazul în care această stație controlează mai multe noduri wireless, atacatorul poate obține acces și controla toate celelalte noduri. O rețea este pe atât de sigură pe cât este cel mai slab dispozitiv conectat la aceasta.
- Man-in-the-Middle – Este acel tip de atac în care atacatorul se poziționează pe linia dintre două dispozitive ce comunică între ele cu intenția de a performa recunoaștere sau a manipula datele ce sunt transmise. Principalul obiectiv este acela ca atacatorul să vadă tot traficul.
- DoS și DDoS – Denial of Service și Distributed Denial of Service, în funcție de câte dispozitive sunt implicate în atac, unul sau mai multe. Ambele tipuri de atacuri urmăresc același scop, iar în cazul comunicației nodurile wireless și AP-ul care menține rețeaua, acesta ar fi de blocare al oricărui mesaj sau proces de comunicare.

Din punct de vedere al securității IT există următoarele trei principale obiective [10]:

- Confidențialitate – Există două tipuri de date: date în mișcare care traversează rețeaua și date în stare de repaus ce sunt stocate local, pe servere, în cloud etc. Confidențialitatea se referă la faptul că doar indivizii sau sistemele autorizate pot avea acces la informații sensibile sau confidențiale. Acest lucru presupune ca niciun alt individ neautorizat nu ar trebui să aibă niciun tip de acces la aceste date. În ceea ce privește datele aflate în mișcare care traversează rețeaua, principala metodă de a le proteja este criptarea tuturor datelor înainte de trimiterea acestora în rețea. O altă opțiune adițională criptării ar fi aceea de a folosi o rețea separată pentru transmiterea informațiilor confidențiale. Acest obiectiv al securității datelor poate fi afectat de atacuri precum Recunoașterea sau Escalarea Prvilegiului.

- Integritate – Se referă la faptul că doar indivizii sau sistemele autorizate pot face modificări asupra datelor. Coruperea datelor reprezintă o eșuare a menținerii integrității datelor. Acest principiu poate fi încălcat de atacuri precum Man-in-the-Middle.

- Disponibilitate – Acest concept se aplică atât sistemelor cât și datelor. Dacă rețeaua sau datele nu sunt disponibile utilizatorilor sau sistemelor autorizate atunci când este nevoie de ele – spre exemplu din cauza unui atac de tip DoS sau poate pur și simplu din cauza căderii rețelei – impactul poate fi semnificativ incluzând pierderea comunicației cu AP-ul. Acest obiectiv poate fi complet pierdut în urma atacurilor de tip DoS sau DDoS.

Pentru a îndeplini obiectivul confidențialității în realizarea comunicației wireless, clientul – stația de control – trebuie să se autentifice la AP-ul reprezentat de routerul utilizat, folosind parola setată pe acesta, iar pentru îndeplinirea obiectivului integrității datelor, datele transmise între cele două dispozitive sunt criptate

Standardul IEEE 802.11 introduce trei tehnici de autentificare folosind o cheie pre-distribuită: []

  - WEP – Specificație 802.11 originală, această tehnică a fost proiectată să furnizeze un nivel de acces asemănător conectării la o rețea folosind o conexiune cu fir. Datele sunt securizate folosind o metodă de criptare RC4 cu o cheie statică. Cum cheia nu se schimbă niciodată, această metodă nu este foarte sigură.
  - WPA – Un standard Wi-Fi Alliance care folosește WEP, dar care securizează datele folosind un algoritm de criptare mult mai puternic – Temporal Key Integrity Protocol. TKIP schimbă cheia la fiecare pachet, ceea ce face algortimul mai greu de spart.

- IEEE 802.11i/WPA2 – IEEE 802.11i este un standard industrial pentru securizarea rețelelor wireless. Versiunea Wi-Fi Alliance este numită WPA2. Ambele tehnici folosesc metoda de criptare AES.

WPA și WPA2 suportă două tipuri de autentificare:

1. Personal – Pentru rețele destinate folosirii personale acasă, sau în birouri de dimensiuni mici, utilizatorii se autentifică folosind o cheie pre-distribuită – PSK. Nu este necesar un server de autentificare special.

2. Enterprise – Pentru rețele de dimensiuni mari, în general în întreprinderi și birouri mari, însă este nevoie de un server suplimentar de autentificare numit RADIUS, ceea ce face acest tip de autentificare mai greu de implementat.

Standardele 802.11i și Wi-Fi Alliance WPA și WPA2 folosesc următoarele tipuri de protocoale:

- TKIP – Metoda de criptare folosită de WPA. Această tehnică de criptare folosește WEP, însă criptează datele folosind TKIP și are un câmp suplimentar pentru Message Integrity Check în pachetul criptat pentru a asigura faptul că mesajul nu a fost alterat.

- AES – Metoda de criptare folosita de WPA2. AES efectuează aceleași funcții ca și TKIP, însă folosește un protocol suplimentar prin care dispozitivul destinație poate recunoaște dacă biții criptați sau necriptați au fost alterați.

- Metoda de autentificare folosită în comunicația wireless dintre cele două dispozitive – AP și nodurile wireless – este de tip WPA Personal. Astfel, datele transmise între nodurile wireless sunt criptate folosind protocolul TKIP.

CAPITOLUL 4

SISTEM SECURIZAT DE TRANSMITERE A INFORMAȚIEI WIRELESS

4.1 Motivația din spatele implementării

De cele mai multe ori, în momentul în care se menționează termenul de securitate, cei mai mulți oameni se duc cu gândul către securitatea fizică, accesul securizat în cadrul unei încăperi sau a propriei locuințe. Deși este un lucru indicat și recomandat, referirea la această securitate, trăim într-o perioadă în care problemele cele mai mari de securitate sunt de ordin informatic. Motivul pentru care acestea reprezintă probleme în adevăratul sens al cuvântului este reprezentat de faptul că oamenii nu conștientizează importanța securizării informatic asupra datelor personale, datelor măsurate și monitorizate și cad pradă de foarte multe ori unor atacuri care pot să fie mitigate cu ușurință, dacă se intervine la momentul potrivit.

Am considerat astfel că mai importantă decât securitatea fizică este cea informatică și mi-am concentrat eforturile pentru o implementare care să ofere un acces securizat asupra datelor monitorizate de mai mulți senzori wireless. Datele monitorizate pot fi generale, cât timp sistemul de securitate este pus la punct. Pentru a avea totuși un set de date concret care să aducă și o utilitate în cadrul unui sistem de automatizări, am optat pentru parametrii legați de temperatură, umiditate, luminozitate etc. în cadrul unui apartament. Implementarea poate să fie însă utilizată în orice fel de mediu în care parametrii de mai sus au relevanță, întrucât nodurile wireless se bucură de o portabilitate crescută, având nevoie doar de o alimentare de la o sursă de 5V, după ce codul a fost încărcat pe acestea. De asemenea, în cadrul implementării aplicației am folosit sistemul Docker, care permite migrarea cu ușurință a aplicațiilor instalate pe o altă plăcuță Raspberry PI.

Pentru a implementa securitatea menționată am folosit autentificare de tip SSL/TLS atât între aplicațiile de Grafana și InfluxDB, dar de asemenea am realizat și criptare a datelor transmite wireless cu ajutorul unui certificat de securitate. În felul acesta, dacă s-ar încerca captarea datelor în oricare dintre punctele de acces, atunci datele ar fi criptate și nu ar avea vreo utilitate.

4.2 NodeMCU & ESP8266

NodeMCU este o platformă open source IoT. Acesta include firmware-ul care rulează pe ESP8266 Wi-Fi SoC de la Espressif Systems și hardware-ul pe care se bazează pe modulul ESP-12. Termenul general de "NodeMCU" se referă mai degrabă la firmware decât la kiturile de dezvoltare. Firmware-ul utilizează limbajul de programare Lua. Acesta se bazează pe proiectul eLua și este construit pe SDK Espressif Non-OS pentru ESP8266. Utilizează multe proiecte open source, cum ar fi lua-cjson și SPIFFS. [19]

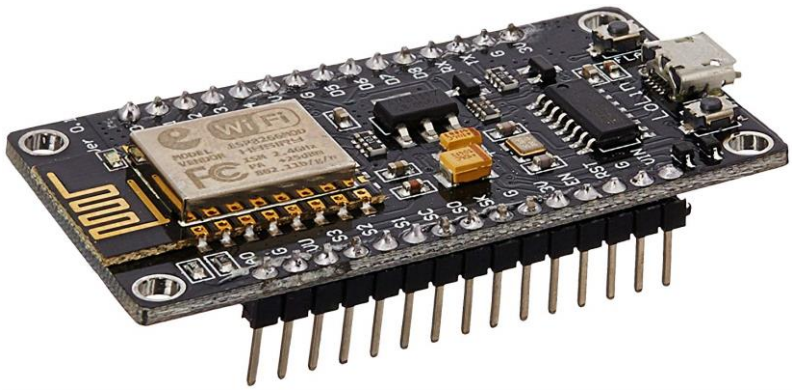


Figura 4.1. Nodul wireless NodeMCU cu controller ESP8266 [18]

Modulul WiFi ESP8266 este un SOC cu protocol TCP / IP integrat care îi poate da acces la rețeaua WiFi oricărui microcontroler. ESP8266 este capabil de hosting pentru o aplicație sau sa preia toate funcțiile de rețea Wi-Fi de la un alt procesor de aplicație. Fiecare modul vine pre-programate, cu firmware cu comenzi AT. [10]

Acest modul are o capacitate de procesare si stocare suficient de puternica, care îi permite să fie integrat cu senzori si dispozitive specifice prin intermediul pinilor GPIO. ESP8266 suporta APSD pentru aplicatii VoIP si interfete cu Bluetooth si contine un RF auto-calibrat care ii permite sa lucreze in toate condițiile de functionare si nu necesita piese RF externe.

Cipul a ajuns pentru prima dată în atenția producătorilor occidentali în august 2014 cu modulul ESP-01, realizat de un producător al treilea producător Ai-Thinker. Acest modul mic permite ca microcontrolerelor să se conecteze la o rețea Wi-Fi și să facă simple conexiuni TCP / IP utilizând comenzi în stil Hayes. Cu toate acestea, la început nu exista aproape nicio documentație în limba engleză despre chip și comenzile pe care le-a acceptat. Prețul foarte scăzut și faptul că pe modul au existat foarte puține componente externe, ceea ce a sugerat că ar putea fi în cele din urmă foarte ieftin în volum, a atras mulți hackeri pentru a explora modulul, cipul și software-ul pe acesta, precum și traduce documentația chinezeasca.[18]

Caracteristicile lui ESP8266 sunt următoarele:

- Procesor: L106 Miez RISC de 32 biți ce rulează la 80 MHz
- Memorie:
    - 32 RAM de instrucțiuni pentru KiB
    - 32 MB RAM cache de instrucțiuni KiB
    - 80 KiB RAM de date de utilizator
    - 16 KiB ETS sistem de date RAM
    - Flash extern QSPI: sunt acceptate până la 16 MiB (de obicei 512 KiB până la 4 MiB)
- IEEE 802.11 b / g / n Wi-Fi
- Comutator TR integrat, balun, LNA, amplificator de putere și rețea de potrivire
- WEP sau WPA / WPA2 sau rețele deschise
- 16 pini GPIO
- SPI
- I²C (implementarea software-ului)
- I²S interfețe cu DMA (partajarea pini cu GPIO)
- UART pe pinii dedicați, plus un UART de transmisie poate fi activat pe GPIO2
- ADC pe 10 biți (ADC de aproximare succesivă)

4.3 Raspberry PI

Plăcuța pe care se încarcă proiectul este de fapt un SBC(Single-Board Computer) și se numește Raspberry Pi 2 Model B și este prima plăcută din a doua generație de Raspberry PI, înlocuind modelul anterior Raspberry Pi în Februarie 2015. Acest model dispune de un procesor quad core pe 64 de biți cu frecvența de 1.2 GHz și are memorie RAM de 1GB. [15]

Raspberry PI 2 dispune de posibilitatea conectării plăcii la o rețea de internet atât prin cablu Ethernet de mare viteză cât și prin rețea wireless. Conectarea se poate face și prin Bluetooth cu ajutorul tehnologiei de ultimă generație BLE(Bluetooth Low Energy). Pe lângă porturile USB discutate anterior placă are port de display DSI pentru conectarea cu un display cu touchscreen. Cardul de memorie MicroSD a fost cumpărat separat și are rolul de a stoca sistemul de operare (OS=operating system) Raspbian și datele pe care dorim să le stocăm în Raspberry. Alimentarea plăcii se face prin intermediul unui MicroUSB și este construit să poată fi folosit atât în SUA pentru tensiuni de 110V/60Hz cât și pentru Europa 220-230V/50Hz. Output-ul este o tensiune aproximativă de 5V la un curent de 2.5 A.

Figura 4.2. Plăcuța Raspberry PI 2 – Model B [17]

Descărcarea sistemului de operare numit Raspbian se face de pe site-ul oficial Raspberry Pi și poate fi ales în varianta cu Desktop sau în varianta Lite în care toate comenzile se fac din consolă. În cadrul proiectului se va utiliza varianta Lite cu scopul de a învăţa și în special a folosi comenzi Linux care ajută în cele mai multe cazuri în dezvoltarea abilităților unui programator. Se downloadează arhiva ce conține software-ul cu ajutorul unui adaptor MicroSD și se dezarhivează sistemul de operare direct în cardul de memorie. [14]

Se introduce cardul de memorie in slot-ul specific și se conectează alimentarea, nu înainte de a verifica că toate componentele de I/O precum tastatura și ecranul sunt conectate. Procesul de instalare a sistemului de operare este automat. Raspberry Pi nu deține un buton fizic de pornire a plăcii , aceasta pornește când este alimentat.

Odată ce instalarea sistemului de operare Raspbian este finalizată se cere utilizatorului să se logheze prin intermediul unui username și al unei parole. Valorile lor default sunt :

Username:pi
Password:raspberry

Acestea pot fi ulterior modificate dacă utilizatorul dorește acest lucru. În cadrul proiectului dezvoltat de mine, am modificat accesul pe această plăcuță întrucât am expus accesul de conectare și configurare pe un IP public.

O provocare frecvent întâlnită în realizarea unui proiect într-un sistem ce operează pe Linux este modul de lucru în linia de comandă. Un inginer software trebuie să fie capabil să efectueze anumite operații în linia de comandă rapid și fără redundantă.

Sudo este o comandă ce oferă libertatea utilizatorului să aibă acces total. Din aceste motive, parola nu mai trebuie introdusă pentru anumite task-uri administrative.

Această comandă este folosită foarte des. Un exemplu în care este folosit sudo este oprirea/restartarea corectă a Raspberry-ului prin :

- sudo shutdown now/sudo reboot

Un beneficiu major întâlnit în majoritatea sistemelor Linux este salvarea comenzilor pe care utilizatorul le face într-o listă de tipul FIFO (first în first out). Astfel, pentru a vedea ultimele comenzi efectuate proiectantul trebuie să apese butonul de pe tastatură ce este marcat cu o săgeată în sus. Cu toate acestea, dacă proiectantul a executat o comandă foarte lungă și nu își mai amintește sintaxa exactă o metodă eficientă este tastarea Ctrl+R care are ca scop deschiderea unui text box cu rol de căutare în istoricul comenzilor.

Linux este un sistem de operare sigur, cu toate acestea există posibile atacuri și bug-uri. Securitatea este un factor foarte important şi el se poate asigura prin actualizarea constantă a sistemului.

- sudo apt-get upgrade- se ocupă cu reîmprospătarea depozitelor
- sudo apt-get update- se ocupă cu actualizarea propriu-zisă

Sistemul de operare Raspbian conține numeroase pachete ce sunt disponibile în depozitele oficiale și oferă numeroase beneficii proiectantului. De exemplu, pentru compilarea unui fișier python pe Raspberry PI este necesară instalarea pachetului pip3.

- sudo apt-get install pip3, instalarea pachetului
- sudo apt-get remove pip3, dezinstalarea pachetului

Alte comenzi de bază [7]:

- cp calea_fisierului calea_directorului_unde_se_copiază, copierea unui fişier şi mutarea copiei în alt fișier mv cale_fişier cale_director_unde_se_mută, mutarea fişierului în altă locaţie
- mv nume_fişier_vechi nume_fişier_nou, redenumirea unui fişier
- rm nume_fişier, ştergerea unui fişier
- cd /home/ANVSIB, se ocupă cu schimbarea directorului în care se lucrează
- mkdir new_folder, aceasta creează un director no

4.4 Docker

Docker ne va ajuta în continuare în realizarea unui sistem izolat și standardizat pentru aplicația noastră și toate dependințele acesteia, în vederea obținerii unui mod de lucru independent.

Docker aduce un plus față de mașini virtuale, prin faptul că este bazat pe un sistem de containere. Aceste containere ne ajută să rulăm fragmente de cod, tool-uri de sistem, librării de sistem sau orice altceva ce se poate instala pe un server, într-un mod independent de mediul de lucru, garantând că acestea se vor rula în același mod.

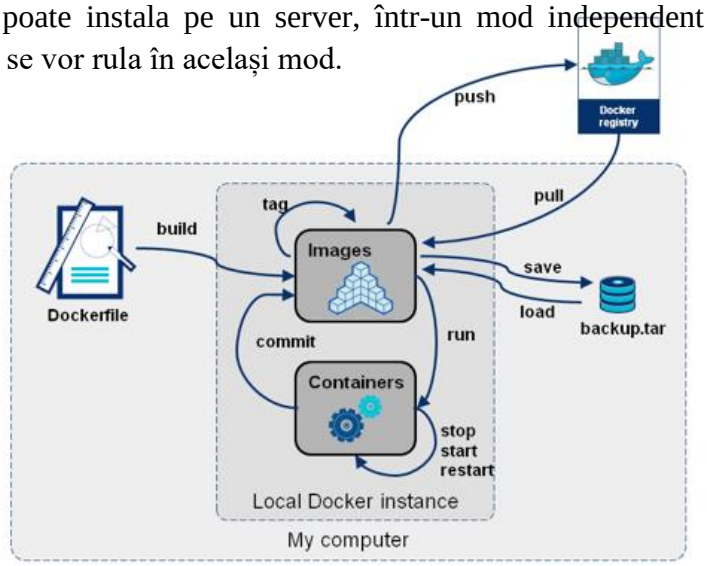


Figura 4.3. Arhitectura aplicațiilor bazate pe Docker [11]

Diferența dintre o mașină virtuală și un container Docker este faptul că aceste containere vor avea într-un final același kernel și nu sunt legate de nicio infrastructură sau sistem de operare gazdă. O altă diferență este că nu ne obligă la o limitare din punct de vedere al memoriei.

Pentru instalarea și pornirea serviciului de Docker pe plăcuța Rasperry PI, vom rula următoarele comenzi:

- sudo apt-get install docker-engine
- sudo service docker start

Ulterior, pentru instalarea și pornirea tuturor serviciilor ce urmează să fie utilizate, vom utiliza următoarele comenzi:

- sudo -i
- apt-get install software-properties-common
- add-apt-repository universe
- apt-get update
- apt-get install -y apparmor-utils apt-transport-https avahi-daemon ca-certificates curl dbus jq network-manager socat
- curl -fsSL get.docker.com | sh
- docker pull influxdb //instalarea serviciului de InfluxDB
- docker pull grafana/grafana //instalarea serviciului de Grafana
- docker run -p 8086:8086 -v influxdb:/var/lib/influxdb influxdb //Ponirea serviciului de InfluxDB
- docker run -d --name=grafana -p 3000:3000 grafana/grafana //Pornirea serviciului de Grafana

În urma rulării acestor comenzi, se instalează și pornesc serviciile de Grafana și InfluxDB, care vor fi utilizate pentru stocarea datelor preluate de la senzorii wireless, respectiv pentru afișarea acestora și gestionara grafică.

4.5 InfluxDB & Grafana

Un server de backend pentru un proiect IoT este un server care colectează date de la senzori. De asemenea, poate procesa datele și le poate afișa pe o pagină web sau poate oferi modalități de accesare a datelor.

Am studiat câteva dintre soluțiile disponibile pentru back-end pentru proiectele IoT. Există multe opțiuni de alegere. Cu toate acestea, fiecare vine cu avantajele și defectele sale. Unele dintre soluțiile disponibile pentru backend sunt: AdafruitIO, DeviceHub, ThingSpeak, Sense IoT etc.

InfluxDB este o bază de date de timpurie cu o sintaxă asemănătoare SQL. Aceasta înseamnă că intrările sunt indexate de o marcă de timp, astfel încât acestea sunt mai ușor de procesat. Spre deosebire de alte soluții de tip backend, acesta oferă doar baza de date cu puncte finale HTTP și UDP care permit dispozitivelor la distanță să introducă sau să interogheze date.

InfluxDB este gratuit și open-source. Aceasta înseamnă că poate fi implementat pe orice mașină deținută de utilizator care rulează Linux sau Mac OSX. Acesta este un mare avantaj deoarece oferă o mai mare flexibilitate. De asemenea, poate rula pe mai multe mașini într-o infrastructură de tip cluster.

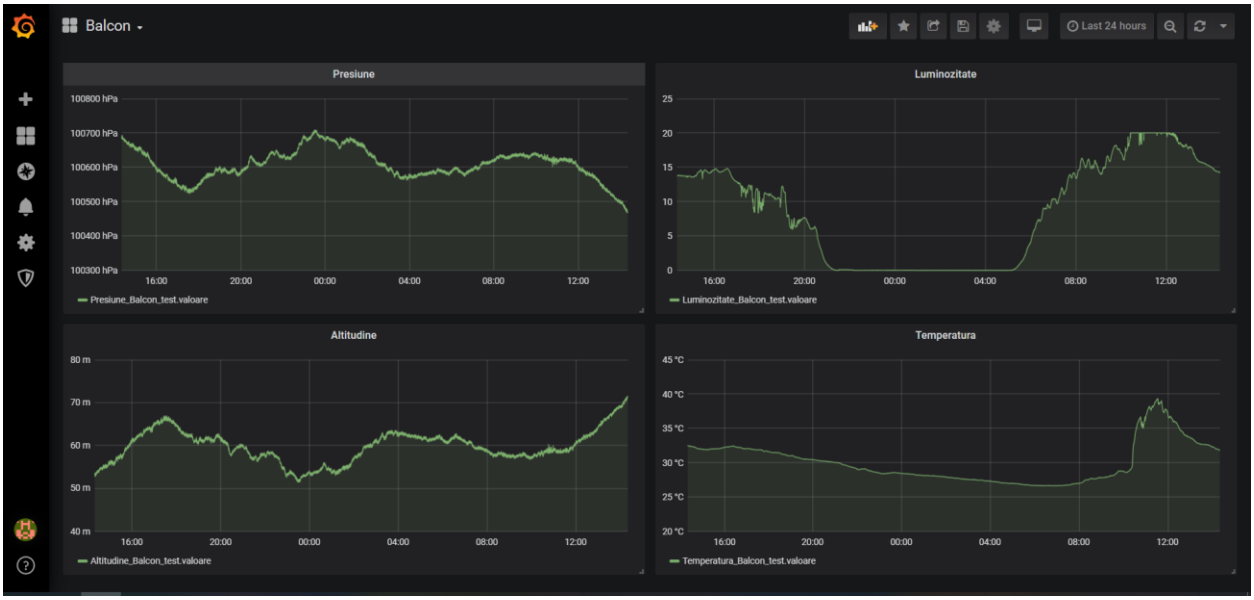


Figura 4.4. Interfața grafică a Grafana cu date preluate din InfluxDB

Am ales InfluxDB în proiectul nostru deoarece am reușit să îl implementăm pe mașinile noastre și să-l configuram cum vrem. De asemenea, am putut încerca diferite configurații fără a afecta alți utilizatori. Un alt motiv important este faptul că comunitatea InfluxDB este foarte activă în acest moment.

InfluxDB este foarte configurabil. Aceasta este o caracteristică excelentă, deoarece o face flexibilă pentru o gamă larg de uzanțe. Configurația se face modificând fișierul /etc/influxdb/influxdb.conf.

Grafana reprezintă o aplicație web folosită pentru prezentarea datelor stocate într-o bază de date. Din punct de vedere al configurării suportă mai multe tipuri de stocare, cum ar fi: ElasticSearch, InfluxDB, MySQL, SQL Server, openstdb etc.

Grafana reprezintă o soluție open-source și a fost preferată în cadrul proiectului deoarece oferă posibilitatea configurării acesului prin HTTPS, oferind astfel proiectului un nivel de securitate din toate punctele de vedere: achiziție de date, trimitere către baza de date și afișarea acestora.

4.6 Implementarea sistemului și securizarea datelor

Din punct de vedere hardware al sistemului am folosit senzori de temperatură, presiune atmosferică, umiditate și luminozitate pentru a realiza o măsurătoare continua a parametrilor în cadrul unor încăperi. Datele de pe acești senzori au fost eșantionate cu ajutorul microcontrolerului ESP8266 instalat și configurat cu ajutorul NodeMCU.

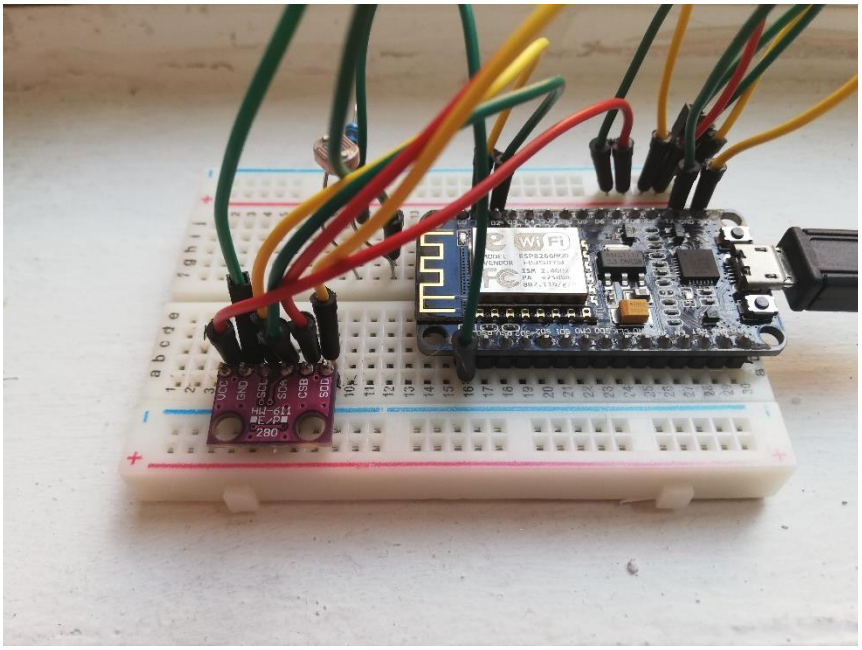


Figura 4.5. Configurația realizată pentru monitorizare pe balcon

Senzorul utilizat pentru presiune, altitudine și temperatură în cadrul balconului este reprezentat de BMP280 GY, care a fost selectat datorită preciziei pe care o pune la dispoziție. Pentru luminozitatea din cadrul balconului a fost utilizat un simplu fotorezistor de pe care se măsura tensiunea cu ajutorul unui pin analogic. În cadrul figurii 4.5 pot fi observate conexiunile realizate în vederea monitorizării parametrilor menționați.

Se poate observa de asemenea și faptul că pentru alimentarea lui NodeMCU este suficientă o sursă de 5V cu microUSB.

Pe de altă parte, pentru măsurarea temperaturii și umidității în cadrul mai multor încăperi, a fost preferat senzorul DHT11, în detrimentul lui DHT22, criteriul care a făcut diferența fiind reprezentat de cost.

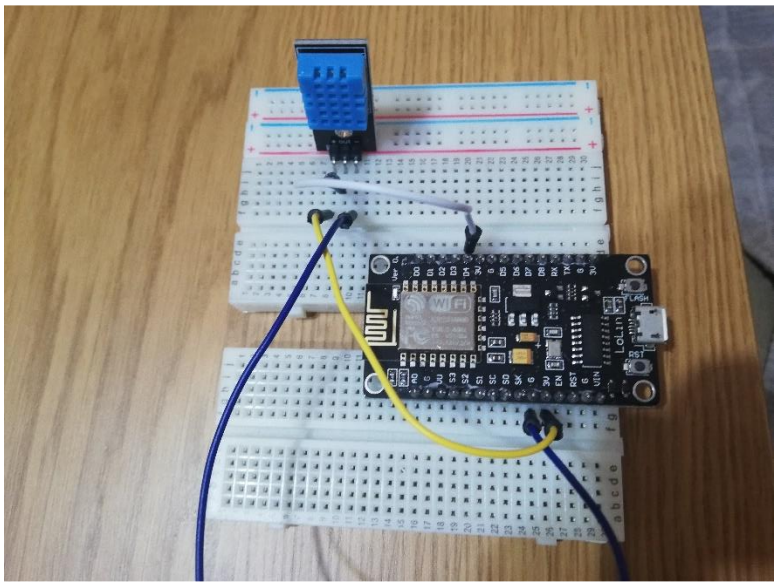


Figura 4.6. Configurația realizată pentru monitorizare în dormitor și sufragerie

Conexiunile necesare pentru acești senzori sunt relativ simplu de realizat întrucât au atașați rezistorii și condensatorii necesari, astfel încât să nu aibă probleme în cazul în care se lovesc de o alimentare instabilă. De asemenea, toți senzorii au fost selectați și pentru faptul că au biblioteci compatibile cu ESP8266.

Pentru a programa aceste chip-uri am folosit interfața pusă la dispoziție de către Arduino IDE și bibliotecile specifice senzorilor.

În vederea securizării datelor între toate componentele rețelei au fost preferate conexiunile de tip HTTPS (pentru vizualizarea datelor dintr-o interfață grafică) și de tip SSL/TLS în vedere încărcării datelor, atât dintre InfluxDB și Grafana dar și dintre nodurile ESP8266 și baza de date în care se realiza stocarea. Un exemplu de autentificare de tip SSL/TLS este prezent în cadrul figurii 4.7

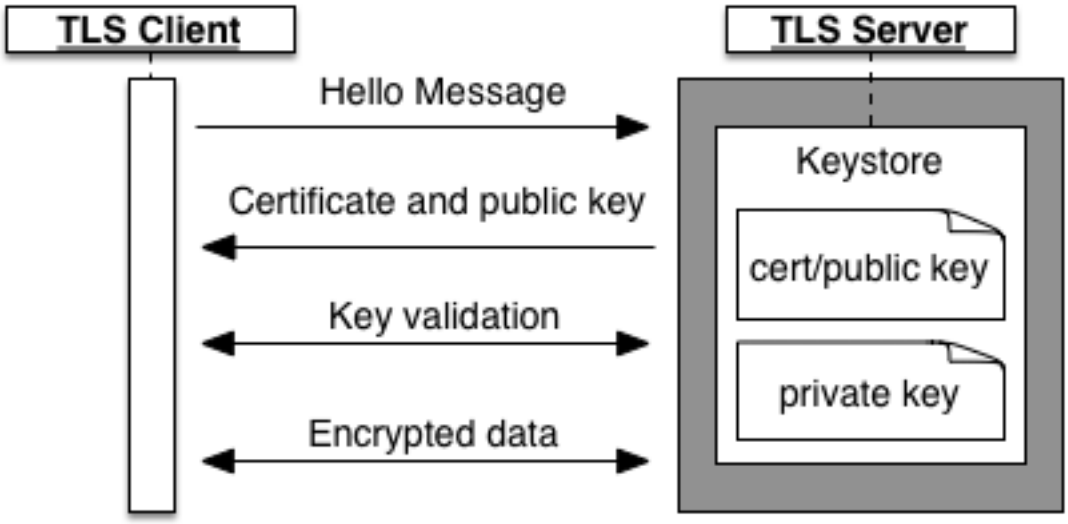


Figura 4.7. Exemplu de autentificare TLS client/server [15]

Faptul că TLS 1.2 este utilizat în cadrul transmisiei de date de la nodurile wireless către baza de date InfluxDB poate fi observat printr-o captură de trafic realizată la nivelul Raspberry PI-ului. În cadrul figurii 4.8 se poate observa handshake-ul ce are loc între client și server, respectiv începutul transmisiei de date.

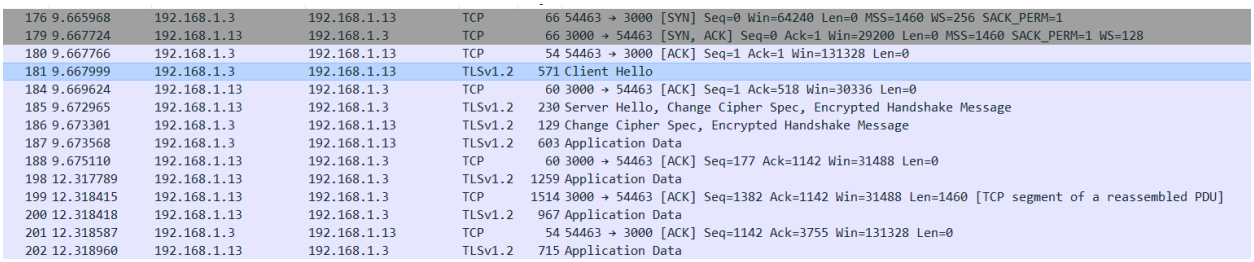
Menționăm că în cadrul capturii de mai jos serverul este reprezentat de InfluxDB și este găzduit pe plăcuța Raspberry PI 2, care este identificată în cadrul rețelei cu adresa IP 192.168.1.13. Nodul wireless MCU este reprezentat de client, cel care transmite informațiile legate de parametrii monitorizați și este identificat cu adresa IP 192.168.1.11.

42	6.210647	192.168.1.11	192.168.1.13	TLSv1.2	271 Client Hello
43	6.210873	192.168.1.13	192.168.1.11	TCP	54 8086 → 51737 [ACK] Seq=1 Ack=218 Win=30016 Len=0
44	6.336059	192.168.1.13	192.168.1.11	TLSv1.2	590 Server Hello
45	6.336100	192.168.1.13	192.168.1.11	TCP	590 8086 → 51737 [ACK] Seq=537 Ack=218 Win=30016 Len=536 [TCP segment
46	6.336293	192.168.1.13	192.168.1.11	TLSv1.2	395 Certificate, Server Key Exchange, Server Hello Done
47	6.366685	192.168.1.13	192.168.1.11	TCP	395 [TCP Retransmission] 8086 → 51737 [PSH, ACK] Seq=1073 Ack=218 Win
48	6.380699	192.168.1.11	192.168.1.13	TCP	60 51737 → 8086 [ACK] Seq=218 Ack=1073 Win=1072 Len=0
49	6.404943	192.168.1.11	192.168.1.13	TCP	60 51737 → 8086 [ACK] Seq=218 Ack=1414 Win=731 Len=0
50	6.405287	192.168.1.11	192.168.1.13	TCP	60 [TCP Window Update] 51737 → 8086 [ACK] Seq=218 Ack=1414 Win=1267
51	6.405520	192.168.1.11	192.168.1.13	TCP	60 [TCP Window Update] 51737 → 8086 [ACK] Seq=218 Ack=1414 Win=1803
52	6.823454	192.168.1.11	192.168.1.13	TLSv1.2	96 Client Key Exchange
53	6.823690	192.168.1.13	192.168.1.11	TCP	54 8086 → 51737 [ACK] Seq=1414 Ack=260 Win=30016 Len=0
54	6.829654	192.168.1.11	192.168.1.13	TLSv1.2	97 Change Cipher Spec, Encrypted Handshake Message
55	6.829786	192.168.1.13	192.168.1.11	TCP	54 8086 → 51737 [ACK] Seq=1414 Ack=303 Win=30016 Len=0
56	6.832084	192.168.1.13	192.168.1.11	TLSv1.2	97 Change Cipher Spec, Encrypted Handshake Message

Figura 4.8. Autentificarea TLS între NodeMCU și InfluxDB

Puteam realiza de asemenea o captură a traficului generat de către orice calculator/laptop sau telefon mobil folosit pentru a accesa interfața web oferită de Grafana. Înainte de pornirea capturii de date, trebuie accesată pagina, prin intermediul adresei IP unde Grafana este găzduită (în cazul nostru 192.168.1.13), împreună cu portul utilizat de aceasta (în cazul nostru). Pentru a ne asigura că beneficiem de o conexiune securizată, în browser introducem https:// la începutul conectării.

În cadrul figurii 4.9 clientul web este reprezentat de browserul Chrome, ce se regăsește pe o stație identificată cu adresa IP 192.168.1.13



176	9.665968	192.168.1.3	192.168.1.13	TCP	66	54463 → 3000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
179	9.667724	192.168.1.13	192.168.1.3	TCP	66	3000 → 54463 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
180	9.667766	192.168.1.3	192.168.1.13	TCP	54	54463 → 3000 [ACK] Seq=1 Ack=1 Win=131328 Len=0
181	9.667999	192.168.1.3	192.168.1.13	TLSv1.2	571	Client Hello
184	9.669624	192.168.1.13	192.168.1.3	TCP	60	3000 → 54463 [ACK] Seq=1 Ack=518 Win=30336 Len=0
185	9.672965	192.168.1.13	192.168.1.3	TLSv1.2	230	Server Hello, Change Cipher Spec, Encrypted Handshake Message
186	9.673301	192.168.1.3	192.168.1.13	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
187	9.673568	192.168.1.3	192.168.1.13	TLSv1.2	603	Application Data
188	9.675110	192.168.1.13	192.168.1.3	TCP	60	3000 → 54463 [ACK] Seq=177 Ack=1142 Win=31488 Len=0
198	12.317789	192.168.1.13	192.168.1.3	TLSv1.2	1259	Application Data
199	12.318415	192.168.1.13	192.168.1.3	TCP	1514	3000 → 54463 [ACK] Seq=1382 Ack=1142 Win=31488 Len=1460 [TCP segment of a reassembled PDU]
200	12.318418	192.168.1.13	192.168.1.3	TLSv1.2	967	Application Data
201	12.318587	192.168.1.3	192.168.1.13	TCP	54	54463 → 3000 [ACK] Seq=1142 Ack=3755 Win=131328 Len=0
202	12.318960	192.168.1.13	192.168.1.3	TLSv1.2	715	Application Data

Figura 4.8. Autentificarea TLS între browser și Grafana [13]

De menționat faptul că certificatul este de tip ”self-signed”, motiv pentru care browser-ul Web nu va recunoaște autoritatea acestuia. Pentru a avea încrederea acestora e necesară generarea de certificat din partea unei entități recunoscute. Totuși, putem în cadrul handshake-ului realizat în figura 4.9 să identificăm cifrul utilizat pentru criptarea datelor.

```
Session ID Length: 0
Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
```

Figura 4.9. Cifrul folosit pentru criptarea datelor

CAPITOLUL 5
CONCLUZII

În cadrul lucrării prezentate am reușit implementarea practică a unui sistem de noduri wireless care să transmită informații către un punct central (bază de date în cazul nostru), într-un mod securizat din mai multe puncte de vedere, baza fiind în protocolul TLS, aplicat fie direct pe transmisia datelor, fie ca și intermediar pentru protocolul HTTP. Implementarea este una realistică și poate fi utilizată în cadrul unui apartament sau în cadrul unor magazii unde este necesară monitorizarea unei serii de parametrii. Deși în cadrul lucrării curente au fost monitorizați niște parametrii generali, aceștia pot fi cu ușurință modificați prin utilizarea senzorilor corespunzători.

Protocolul wireless folosit pentru transmisia de date a fost ales după o serie de criterii care include distanța acoperită într-un mediu indoor, numărul de noduri suportate în cadrul rețelei, flexibilitatea acestora, consumul de energie în timpul transmisiei și nivelul de securitate oferit în cadrul procesului de transmitere a datelor. În urma analizei realizate și prezentate în cadrul capitolului 2, s-a concluzionat că cel mai potrivit protocol pentru acest proces este Wireless-Fidelity (Wi-Fi).

După cum am menționat și în cadrul capitolului 4, securitatea la care facem referire în cadrul acestei lucrări se leagă de latura informatică, nicidecum de cea fizică. Am considerat că acest nivel de securitate este de preferat în cadrul lucrării întrucât foarte multă lume trece superficial peste el, gândindu-se doar la siguranța locuinței proprii, fără a-și da seama că se poate intra în aceasta și altfel decât pe ușă.

Securitatea informatică reprezintă una dintre cele mai importante probleme ale secolului în care trăim, având în vedere viteza constantă de dezvoltare a tehnologiei și a includerii acesteia în orice fel de obicei zilnic al oamenilor. Internetul și mai exact, stiva de protocoale TCP, discutată și explorată în cadrul capitolului 3. Aceasta asigură un proces de comunicare eficient între protocoalele aflate pe mai multe nivele, de la nivelul fizic până la cel de aplicație, care reprezintă interacțiunea cu utilizatorul final. Avantajul major al acestei stive de protocoale și a implementării prin intermediul acesteia este reprezentat de garanția de interacțiune între diverse aplicații, indiferent de producător.

Sistemul de monitorizare, raportare și control dezvoltat și prezentat în cadrul acestei aplicații respectă principiul din spatele stivei TCP/IP, folosind cele mai importante caracteristici puse la dispoziție de protocoale importante și sigure precum Wi-Fi, TCP, TLS și HTTP. Toate acestea, combinate, fac sistemul unul avansat și propice de folosit în orice fel de mediu și încăpere, cât timp aria este acoperită de protocolul wireless de transmisie de date.

BIBLIOGRAFIE

[1] Perrilo, M., Heinzelman, W., „Wireless Sensor Network Protocols”, Department of Electrica land Computer Engineering, 2015

[2] Faludi, R., „Building Wireless Sensor Networks”, Editura O’Reilly, 2011

[3] Orner, R., Grünbacher, E., Guger, C., „State of the Art in Sensors, Signals and Signal Processing”, în GmbH/Guger Technologies OG, 2009, pp 9 – 14.

[4] Lee, J., Suu Y., Shen, C., “A Comparative Study of Wireless Protocols”, Bluetooth, UWB, ZigBee, and Wi-Fi”, The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON), Taiwan, 2007

[5] Muqri, M., Alfaro, R., “BuildingWireless Sensor Networks with Zigbee”, 12th ASEE Annual Conference and Exposition, 2013

[6] Bavarva, A., Pathel, N., “Wireless Sensors Networks using ZigBee”, ResearchGate, 2016

[7] ZigBee Alliance, ZigBee Specification, Ianuarie 2008, ZigBee Document 05347r17

[8] Augusto, R., Severino, R., “On the use of IEEE 802.15.4/ZigBee for Time-Sensitive Wireless Sensor Network Applications”, Polytechnic Institute of Porto

[9] XBee Datasheet

[10] DS18B20 Sensor Datasheet

[11] Olsson, J., Lönn, J., “ZigBee for wireless Networking”, Universitatea Linköping, lucrarea de disertație, 2015

[12] Bragisnky, D., Estrin, R., ”Rumor routing algorithm for sensor networks” – Editura O’Reilly, 2009

[13] http://www.zigbee.org - accesat la data: 11.06.2019

[14 https://www.digi.com/lp/xbee - accesat la data: 10.06.2019

[15] https://en.wikipedia.org/wiki/Zigbee - accesat la data: 12.06.2019

[16] https://www.arduino.cc/en/Reference/HomePage - accesat la data: 08.06.2019

[17] https://www.arduino.cc/en/Main/ArduinoBoardUno - accesat la data: 08.06.2019

[18] Călin, H. ”Comunicarea între noduri wireless cu protocolul ZigBee”, Laboratorul SpeeD, Lucrare de licență, 2017

[19] Tănasi, S., ”Întrerupător electric wireless”, Lucrare de licență, 2017

[20] Tomescu, D., ” Comunicație wireless între o dronă terestră și un UAV”, Lucrare de licență, 2017

ANEXA 1

```
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>

#define INFLUXDB_HOST "192.168.1.13"
const int httpsPort = 8086;
const char fingerprint[] PROGMEM = "68 77 21 EF 51 F3 B7 27 BC 68 6E 79 06 F0 26 CD 69
00 F7 BD";
#define INFLUXDB_USER "esp"
#define INFLUXDB_PASS "telacad2019"
#define WIFI_SSID "DIICOT_Echipaj673019"
#define WIFI_PASS "HORIAFACUTNEt1"

#include "DHT.h"
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
#include <ESP8266WiFi.h>

#define BMP_SCK 13
#define BMP_MISO 12
//#define BMP_MOSI 11
#define BMP_CS 10

Adafruit_BMP280 bme; // I2C

ESP8266WiFiMulti WiFiMulti;

float Temperatura_Sufragerie;
float Umiditate_Sufragerie;

uint8_t DHTPin = D4;

#define DHTTYPE DHT11
DHT dht(DHTPin, DHTTYPE);
WiFiClientSecure httpsClient;
```

```
void setup() {
  Serial.begin(9600);

   Serial.println(("Test pentru senzorul DHT..."));
   dht.begin();

  WiFiMulti.addAP(WIFI_SSID, WIFI_PASS);
  Serial.println("In proces de conectare la reteaua Wi-Fi...");
  while (WiFiMulti.run() != WL_CONNECTED) {
    delay(1000);
  }
  Serial.println("Conectare la WiFi: COMPLETA");
  Serial.println("Adresa IP obtiuta: ");
  Serial.println(WiFi.localIP());

//  influx.setDbAuth("dizertatie", INFLUXDB_USER, INFLUXDB_PASS); //autentificare la baza
de date cu credentialele configurate
  Serial.println("Initializare completa...");
}

void send_value(String masuratoare, float valoare){
  String content = masuratoare + ",camera=camera valoare="+String(valoare);

  httpsClient.print("POST /write?db=dizertatie HTTP/1.1\r\n");
  httpsClient.print("User-Agent: esp8266/0.1\r\n");
  httpsClient.print("Host: 192.168.1.13:8086\r\n");
  httpsClient.print("Content-Length: " + String(content.length()) + "\r\n");
  httpsClient.print("Content-Type: application/x-www-form-urlencoded\r\n");
  httpsClient.print("\r\n");
  httpsClient.print(content + "\r\n");

  while (httpsClient.connected()) {
    String line = httpsClient.readStringUntil('\n');
    if (line == "\r") {
      Serial.println("Datele au fost trimise cu succes!");
      break;
    }
```

```
  }
    Serial.println("Raspuns>:");
  Serial.println("==========");
  String line;
  while(httpsClient.available()){
   line = httpsClient.readStringUntil('\n');
   Serial.println(line);
  }
  Serial.println("==========");
  Serial.println("Inchidem conexiunea");
  }

void open_HTTPS_session(){
   httpsClient.setFingerprint(fingerprint);
   httpsClient.setTimeout(15000); // 15 Seconds
   delay(1000);

    Serial.print("Initializare conexiune HTTPS");
    int r=0; //retry counter
    while((!httpsClient.connect(INFLUXDB_HOST, httpsPort)) && (r < 30)){
        delay(100);
        Serial.print(".");
     r++;
     }
   if(r==30) {
    Serial.println("Conexiune nereusita!");
   }
   else {
    Serial.println("Conexiune reusita!");
   }
  }

void loop() {
//Deschiderea sesiunii HTTPS pentru transferul datelor in mod securizat
  open_HTTPS_session();


  Temperatura_Sufragerie = dht.readTemperature();
  send_value("Temperatura_Sufragerie_test", Temperatura_Sufragerie);
```

```
  Umiditate_Sufragerie = dht.readHumidity();
  send_value("Umiditate_Sufragerie_test", Umiditate_Sufragerie);

  delay(5000);
}
```

ANEXA 2

```
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <InfluxDb.h>

#define INFLUXDB_HOST "192.168.1.13"
const int httpsPort = 8086;
const char fingerprint[] PROGMEM = "68 77 21 EF 51 F3 B7 27 BC 68 6E 79 06 F0 26 CD 69
00 F7 BD";
#define INFLUXDB_USER "esp"
#define INFLUXDB_PASS "telacad2019"
#define WIFI_SSID "DIICOT_Echipaj673019"
#define WIFI_PASS "HORIAFACUTNEt1"

#include "DHT.h"
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
#include <ESP8266WiFi.h>

#define BMP_SCK 13
#define BMP_MISO 12
#define BMP_CS 10

Adafruit_BMP280 bme; // I2C

ESP8266WiFiMulti WiFiMulti;
Influxdb influx(INFLUXDB_HOST);

float Luminozitate_Balcon;
float Temperatura_Balcon;
float Presiune_Balcon;
float Altitudine_Balcon;

WiFiClientSecure httpsClient;
```

```
void setup() {
  Serial.begin(9600);
  Serial.println(("Test pentru senzorul BMP280..."));

  if (!bme.begin()) {
    Serial.println("Conectarea la BMP280 a esuat. Verificati conexiunile!");
    while (1);
  }

  WiFiMulti.addAP(WIFI_SSID, WIFI_PASS);
  Serial.println("In proces de conectare la reteaua Wi-Fi...");
  while (WiFiMulti.run() != WL_CONNECTED) {
    delay(1000);
  }
  Serial.println("Conectare la reataua WiFi: COMPLETA");
  Serial.print("Adresa IP obtiuta: ");
  Serial.println(WiFi.localIP());
  Serial.println("Initializare completa...");
}

void send_value(String masuratoare, float valoare){
  String content = masuratoare + ",camera=camera valoare="+String(valoare);

  httpsClient.print("POST /write?db=dizertatie HTTP/1.1\r\n");
  httpsClient.print("User-Agent: esp8266/0.1\r\n");
  httpsClient.print("Host: 192.168.1.13:8086\r\n");
  httpsClient.print("Content-Length: " + String(content.length()) + "\r\n");
  httpsClient.print("Content-Type: application/x-www-form-urlencoded\r\n");
  httpsClient.print("\r\n");
  httpsClient.print(content + "\r\n");

  while (httpsClient.connected()) {
    String line = httpsClient.readStringUntil('\n');
    if (line == "\r") {
      Serial.println("Datele au fost trimise cu succes!");
      break;
    }
  }
```

```
    Serial.println("Raspuns>:");
  Serial.println("==========");
  String line;
  while(httpsClient.available()){
   line = httpsClient.readStringUntil('\n');
   Serial.println(line); /
  }
  Serial.println("==========");
  Serial.println("Inchidem conexiunea");
  }

void open_HTTPS_session(){
   httpsClient.setFingerprint(fingerprint);
   httpsClient.setTimeout(15000);
   delay(1000);

    Serial.print("Initializare conexiune HTTPSS");
    int r=0; //retry counter
    while((!httpsClient.connect(INFLUXDB_HOST, httpsPort)) && (r < 30)){
        delay(100);
        Serial.print(".");
     r++;
     }
   if(r==30) {
    Serial.println("Conexiune nereusita!");
   }
   else {
    Serial.println("Conexiune reusita!");
   }
  }

void loop() {
//Deschiderea sesiunii HTTPS pentru transferul datelor in mod securizat
  open_HTTPS_session();

  int valoareSenzorLumina_Balcon = analogRead(A0);
  float Lumina_Balcon = valoareSenzorLumina_Balcon * (5.0/256.0);
  send_value("Luminozitate_Balcon_test", Lumina_Balcon);
```

```
  Temperatura_Balcon = bme.readTemperature();
  send_value("Temperatura_Balcon_test", Temperatura_Balcon);

  Presiune_Balcon = bme.readPressure();
  send_value("Presiune_Balcon_test", Presiune_Balcon);
  Altitudine_Balcon = bme.readAltitude(1013.25);
  send_value("Altitudine_Balcon_test", Altitudine_Balcon);


}
```