



UNIVERSITÀ DEGLI STUDI DI FIRENZE
SCUOLA DI INGEGNERIA - DIPARTIMENTO DI INGEGNERIA
DELL'INFORMAZIONE

Tesi di Laurea Triennale in Ingegneria Informatica

**PROGETTAZIONE DI APPLICAZIONI DISTRIBUITE
PER AMBIENTI INTELLIGENTI
IN TECNOLOGIA 6G**

Candidato
Giacomo Vitangeli

Relatore
Prof. Francesco Chiti

Anno Accademico 2020-2021

*Alla mia ragazza Claudia,
mi ha cambiato la vita ed è stata al mio fianco
anche nei momenti più difficili,
credendo sempre nelle mie capacità e condividendo insieme
momenti indimenticabili.*

*Ringrazio di cuore la mia famiglia,
in particolare i miei genitori e mio fratello Leonardo,
hanno sempre supportato le mie scelte
e non hanno mai smesso di credere in me.*

*Ringrazio il Prof. Francesco Chiti,
che è stato sempre molto disponibile,
di stimolo e di supporto durante la realizzazione di questa tesi.*

*Un ringraziamento anche a tutti i miei amici,
in particolare Riccardo e Cristian,
con i quali in questi anni universitari ho condiviso bellissimi momenti di
spensieratezza e avventura, oltre che di studio.*

Indice

Elenco delle figure	iv
Introduzione	i
1 Tecnologia 6G	1
1.1 Terahertz	2
1.2 Ultra-Massive MIMO	4
1.3 SPP e Plasmonic Nano-Antenna	7
1.4 Ambienti Intelligenti	10
2 Progetto	14
2.1 Idea	14
2.2 Diagrammi	15
2.2.1 Use Case Diagram	15
2.2.2 Class Diagram	17
2.2.3 Sequence Diagram	18
2.3 Collaborazione tra i Sensori	21
2.4 Topologia di rete	22
2.5 Implementazione	23
3 Test e Risultati	28

3.1	Consumi energetici	28
3.2	Latenza	33
3.3	Packet Loss	34
Conclusioni		38
Bibliografia		40

Elenco delle figure

1.1	Spettro Terahertz e lunghezza d'onda.	3
1.2	Larghezza di banda in uso per 4G, 5G e 6G.	4
1.3	Esempi di sistemi: 2D MIMO e 3D MIMO.	6
1.4	Array di nano-antenne plasmoniche.	8
1.5	Il front end THz è composto da una sorgente plasmonica (feed point), un modulatore plasmonico e una nano-antenna plasmonica.	9
1.6	Esempio di Ambiente Intelligente con array di nano-antenne plasmoniche.	11
1.7	Modello di array plasmonico riflettente, in grado di manipolare le onde EM [8].	12
2.1	Use Case Diagram	16
2.2	Class Diagram	18
2.3	Sequence Diagram	19
2.4	Message Passing Model	22
2.5	le topologie scelte per la rete.	23
3.1	Consumo energetico.	31
3.2	Packet Loss con probabilità del 20%	35
3.3	Packet Loss con probabilità del 50%	36

Introduzione

L'incessante e rapida rivoluzione dell'informazione sta donando all'umanità tecnologie sempre più all'avanguardia che tendono ad avere come unico limite la creatività dell'uomo.

Tra le molte innovazioni tecnologiche che ci circondano quotidianamente ho deciso di concentrarmi sulle reti 6G. Queste sono in fase di sviluppo e con molta probabilità vedranno la luce prima del 2030, da quanto è emerso dai colossi del settore ci si aspetta che permetteranno data rate dell'ordine di 1Tbps, circa 1000 volte più veloce dell'odierno 5G, ed un ritardo di propagazione sul canale di trasmissione inferiore ad 1ms.

Questi risultati saranno le fondamenta per un mondo completamente interconnesso ed aperto a molteplici applicazioni innovative.

L'obiettivo di questo progetto è illustrare le potenzialità del 6G e mostrarne un modello di applicazione, in particolare vedremo l'interazione tra i nodi di una rete che compongono un ambiente intelligente; i nodi rappresentano un certo numero di periferiche interconnesse, le quali cooperano tra loro per ottenere un determinato dato sull'ambiente in cui sono immerse.

Tale modello di simulazione è stato implementato in OMNeT++, un IDE molto usato nell'ambito della progettazione e simulazione di reti.

La descrizione del progetto sarà completata dei test e dai risultati dello stesso.

Capitolo 1

Tecnologia 6G

Questo capitolo prende in analisi la rete 6G, evidenziandone punti di forza e debolezze.

Con il rapido sviluppo di nuove applicazioni emergenti, come real-time e servizi interattivi, il traffico dati wireless è aumentato notevolmente negli ultimi anni e le attuali reti cellulari 5G non possono soddisfare completamente i requisiti tecnici sempre più stringenti.

Per affrontare le prossime sfide, la sesta generazione di rete mobile dovrà vantare un elevato standard tecnico, con trasmissioni efficienti anche dal punto di vista energetico.

Le reti 6G saranno basate principalmente sull'architettura 5G esistente, ereditandone i vantaggi con un'architettura di rete decentralizzata, ed ampliandone le potenzialità di applicazione grazie all'aumento della banda di frequenza fino a raggiungere l'ordine dei Terahertz.

Nell'evoluzione e nello sviluppo di reti mobili sono emersi sempre nuovi schemi di trasmissione:

- 3G: accesso multiplo a divisione di codice (**CDMA**).

- 4G: multi-input multi-output (**MIMO**) e multiplexing a divisione di frequenza ortogonale (**OFDM**).
- 5G: onde millimetriche (**mmWave**) e **Massive MIMO**.
- 6G: polaritoni plasmonici di superficie (**SPP**) e **Ultra-Massive MIMO**.

Perciò è naturale chiederci come poter sfruttare in modo innovativo i gradi di libertà su cui possiamo agire: tempo, frequenza e spazio per soddisfare le diverse esigenze della società [7].

Partendo dal presupposto che il 6G seguirà le orme dell'attuale 5G andiamo a definire i tratti distintivi di quest'ultimo che si andranno a ritrovare nella sesta generazione.

Dopo aver considerato le frequenze su cui trasmetterà, verranno analizzati gli aspetti legati alle tecniche di base che hanno accompagnato il 5G e come queste verranno adattate alla sesta generazione, fino ad arrivare al tipo di antenne utilizzate (array di nano-antenne plasmoniche).

Infine verrà trattato un modello di Intelligent Environment, realizzabile grazie alla tecnologia 6G, aumentando quindi i livelli di sicurezza di edifici e infrastrutture.

1.1 Terahertz

Per garantire data rate elevati il 6G utilizzerà uno spettro di frequenza più alto rispetto alle generazioni precedenti, sarà la prima rete mobile ad utilizzare frequenze di trasmissione dell'ordine dei Terahertz.

In figura 1.1 è rappresentata la larghezza di banda Terahertz con la relativa lunghezza d'onda.

Fino al 4G le frequenze utilizzate non hanno mai superato i 6GHz, mentre il 5G lavora su due intervalli di frequenze diversi, uno inferiore ai 6GHz simile ai canali precedenti, ed un secondo compreso tra i 24GHz e i 100GHz sfruttando le potenzialità delle onde millimetriche.

Sarebbe intuitivo pensare che con l'aumento della frequenza di trasmissione si avrà un conseguente incremento della quantità di dati scambiati nell'unità di tempo, però la fisica ci insegna che aumentando la frequenza del segnale questo avrà maggiore difficoltà ad attraversare oggetti solidi.

Per superare questa criticità di natura fisica il 5G ricorre ad antenne Massive MIMO, che di fatto sono un miglioramento delle antenne MIMO, multi-input multi-output, mentre nel 6G come presto vedremo, un modello Massive MIMO non è sufficiente a garantire la corretta comunicazione per frequenze dell'ordine dei Terahertz.

L'efficienza di un canale è data dalla larghezza di banda e dal rapporto segnale rumore, perciò grazie al sistema MIMO è possibile ottenere un rapporto segnale rumore più alto aumentando l'intensità del segnale, tale incremento è dovuto alla collimazione di più segnali orientati.

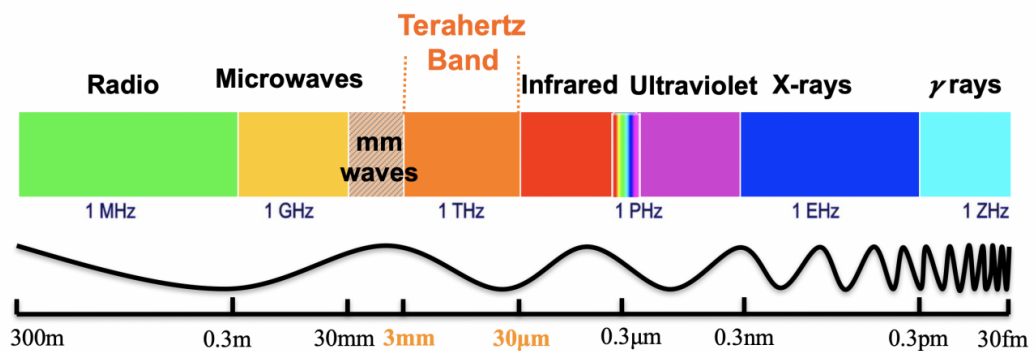


Figura 1.1: Spettro Terahertz e lunghezza d'onda.

Nel 2020 il dipartimento di ricerca della **Samsung** ha pubblicato un white paper in merito alla visione del 6G [9], evidenziando le potenzialità di questa

tecnologia.

Tecnologia che può essere oggetto di sperimentazione da Marzo 2019, data in cui la **Federal Communications Commission** (FCC) ha aperto ad uso sperimentale lo spettro tra 95 GHz e 3,000 GHz, per incoraggiare lo sviluppo di nuove tecnologie wireless.

Nell'utilizzo del THz andranno superati diversi ostacoli, primo tra tutti il path-loss dovuto all'assorbimento atmosferico, infatti nell'aria il path-loss è proporzionale al quadrato della frequenza del segnale.

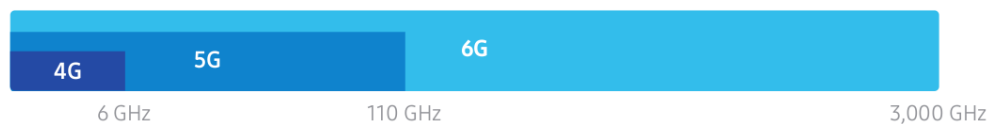


Figura 1.2: Larghezza di banda in uso per 4G, 5G e 6G.

1.2 Ultra-Massive MIMO

Prima di arrivare al Massive MIMO e all'Ultra-Massive MIMO, iniziamo con il definire lo schema di trasmissione MIMO, già adottato da diversi anni in reti wireless come LTE 4G, Wi-Fi e WiMAX.

MIMO: L'acronimo MIMO sta per multi-input multi-output, quindi prevede che vi siano diverse antenne sia lato sorgente che lato destinazione. Questo approccio permette di aumentare l'efficienza di una rete, permettendo al segnale di superare gli ostacoli, sfruttando le molte antenne per determinare il percorso con il minor numero di ostacoli e producendo canali fisici multipli, trasmettendo un maggior numero di flussi di dati a parità di tempo e frequenza di trasmissione.

Fin dalla versione più semplice di questo schema di trasmissione si sono cercati dei metodi per ottimizzare questo tipo di tecnologia, cercando di adattarla al meglio ai molteplici casi d'uso [1].

Con le nuove generazioni di reti wireless le scelte di adattamento verranno prese da un'Artificial Intelligence pervasiva che guiderà le periferiche interconnesse.

Massive MIMO: Insieme alla comparsa della rete 5G è avvenuta anche la diffusione di una versione maggiorata del multi-input multi-output, il Massive MIMO.

Originariamente Massive MIMO non specificava in quante dimensioni si dovesse sviluppare, ma nel tempo si è notato che fornendo alle antenne delle componenti dinamiche e successivamente anche un dato di elevazione, di quanto discostare il segnale da terra, aumentava di gran lunga la qualità di ricezione anche in ambienti urbani, come in palazzi di media altezza.

Perciò da un sistema a uno o due dimensioni, siamo giunti ad un sistema **3D MIMO** che rende la comunicazione più efficace, essendo composto da una matrice di antenne, in cui ciascuna di esse è in grado di mutare la propria inclinazione sui tre assi spaziali, così da coordinarsi con le altre e collimare al meglio il segnale, tale fenomeno è detto **Beamforming** [2].

2D MIMO e 3D MIMO si differenziano in particolare per il numero di antenne presenti nella base station e per le differenti distanze che può raggiungere il secondo come si può notare nei due esempi in figura 1.3, questa maggiore libertà nel beamforming garantisce anche un miglioramento nella gestione di più utenti contemporaneamente [3].

Ultra-Massive MIMO: Prendendo in considerazione la rapida crescita dei data rate per le comunicazioni wireless negli ultimi tre decenni, è norma-

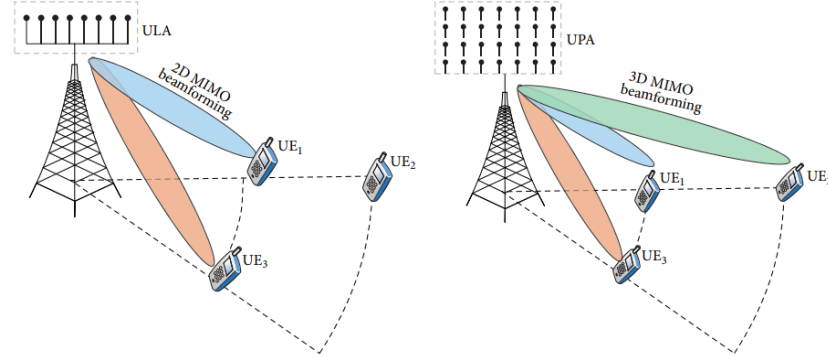


Figura 1.3: Esempi di sistemi: 2D MIMO e 3D MIMO.

le aspettarci che questi raggiungano valori dell'ordine di *Terabit-per-second* (Tbps) entro la fine di questa decade. In tal senso Massive MIMO verrà aggiornato in Ultra-Massive MIMO, facendo uso di antenne di dimensioni estremamente ridotte in modo da poter ospitare molti più ricetrasmittitori. La particolarità di questa nuova versione di MIMO non sta solo nell'adozione di antenne più piccole, ma nei materiali e nelle tecniche di trasmissione utilizzate, adottando materiali plasmonici e trasmettendo i segnali attraverso SPP waves.

La forte criticità nell'aumento della frequenza di trasmissione è data dall'incremento di perdita o attenuazione del segnale (**Path loss**) all'aumentare della distanza.

La larghezza di banda disponibile ad alte frequenze cambia drasticamente con il variare della distanza: da quasi 10THz per distanze inferiori ad un metro, fino a centinaia di GHz per distanze superiori [4].

La realizzazione di Ultra-Massive MIMO sta optando da diverso tempo per vie alternative alle antenne metalliche classiche, con lo scopo di aumentare la densità di ricetrasmittitori per unità di superficie, ad esempio introducendo l'uso di nano-antenne plasmoniche.

1.3 SPP e Plasmonic Nano-Antenna

Per superare il problema del forte path loss all'aumentare della distanza bisogna quindi aumentare il numero di ricetrasmittitori, in modo da aumentare la potenza del segnale ed affinare il beamforming. Vediamo quindi come varia la dimensione delle antenne metalliche in funzione della frequenza trasmessa:

- fino a 5GHz: la dimensione degli array di trasmissione è dell'ordine di qualche metro per decine di antenne, quindi il loro sviluppo è limitato solo dalla base station.
- mmWaves: si riescono a condensare decine di antenne in pochi centimetri quadrati. Sebbene ciò consenta la loro integrazione nei dispositivi mobili, questa dimensione delle antenne non sarebbe sufficiente a superare i problemi di path loss per frequenze superiori alle onde millimetriche, infatti per distanze superiori a poche decine di metri il segnale si andrebbe a perdere.
- THz: per frequenze così elevate c'è la necessità di ricorrere ad antenne molto più piccole e molto più numerose, infatti un antenna di cento elementi è dell'ordine di pochi millimetri quadrati. Tuttavia per far sì che queste antenne possano venir utilizzate c'è bisogno di un forte controllo dinamico delle antenne, in modo da direzionarle al meglio.

La dimensione l di un antenna risonante è quindi legata alla frequenza dell'onda trasmessa, infatti è circa la metà della lunghezza d'onda λ alla frequenza di risonanza.

$$l \approx \lambda/2$$

Sapendo che un segnale con una frequenza di 10THz ha una lunghezza d'onda di $30\mu\text{m}$, possiamo approssimare che per generare un segnale ad 1THz abbiamo bisogno di antenna metallica di $150\mu\text{m}$.

Sebbene il risultato appena ottenuto con antenne metalliche possa sembrare la soluzione definitiva, non è così. Infatti sviluppando delle antenne in materiale plasmonico si avrebbero dei grandi benefici in termini di dimensione delle nano-antenne, con un conseguente miglioramento della quantità di segnali trasmessi per unità di superficie.

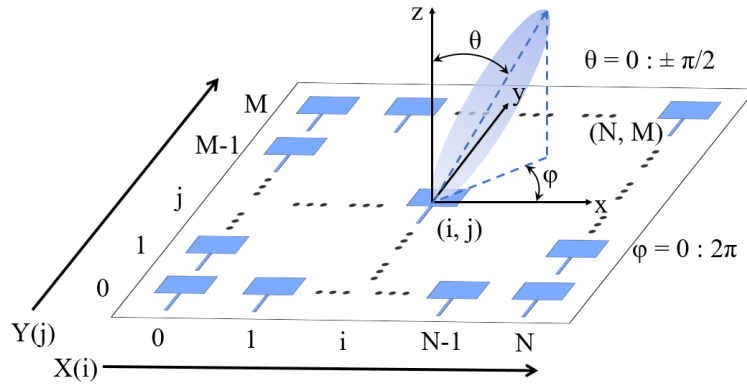


Figura 1.4: Array di nano-antenne plasmoniche.

In figura 1.4 si può notare il modello di un array per nano-antenne plasmoniche, queste si coordinano e trasmettono il segnale collimandolo nella medesima direzione adottando inclinazioni diverse: con θ viene rappresentato l'angolo di elevazione che descrive la deviazione dalla normale del pannello (scostamento dall'asse z), mentre ϕ è lo scostamento dall'asse x .

I materiali plasmonici possono essere metalli o materiali che si comportano come i metalli che supportano la propagazione di **Surface Plasmon Polariton waves** (SPP), delle onde elettromagnetiche particolari che coinvolgono sia il movimento di cariche nel metallo che le onde elettromagnetiche nel dielettrico, ad esempio nell'aria.

La particolarità delle onde SPP e l'uso di questo tipo di materiali, sta nella minore velocità di propagazione rispetto alle onde EM classiche nel vuoto.

La differenza di velocità di propagazione tra i due tipi di onda si ritrova nella lunghezza d'onda, che sarà molto minore per onde SPP ($\lambda_{spp} < \lambda$), ciò porta ad una grande riduzione di dimensione dei ricetrasmittitori in materiali plasmonici.

Le onde SPP si dicono confinate rispetto alle onde EM, tale confinamento viene espresso da un fattore di confinamento γ che varia in funzione del materiale plasmonico utilizzato e della frequenza del sistema.

$$\gamma = \lambda / \lambda_{spp}$$

In merito dei passaggi matematici riportati in [5] possiamo concludere che a differenza delle antenne metalliche, la dimensione di quelle plasmoniche è circa:

$$l_p \approx \lambda_{spp} / 2 = \lambda / (2\gamma) \quad (1.1)$$

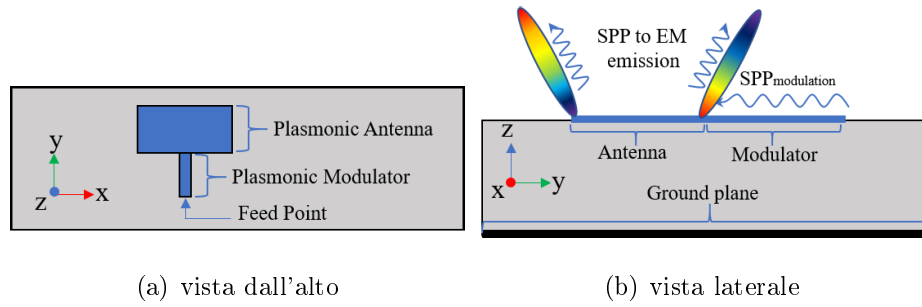


Figura 1.5: Il front end THz è composto da una sorgente plasmonica (feed point), un modulatore plasmonico e una nano-antenna plasmonica.

In figura 1.5 è possibile notare il principio di funzionamento di una nano-antenna plasmonica, gli articoli [10] e [11] forniscono specifiche tecniche più dettagliate, utili per la progettazione delle stesse.

1.4 Ambienti Intelligenti

UM MIMO permetterà la realizzazione di ambienti intelligenti, ovvero l'integrazione con la rete di molti sensori intelligenti che comunicano costantemente tra loro e monitorano lo stato di un ambiente, aperto o chiuso che sia.

Conferendo un ruolo attivo all'ambiente, che non viene visto più come un oggetto passivo attraversato dai segnali, ma come un modulo attivo in grado di direzionarli a seconda delle esigenze.

Non si tratta semplicemente di sensori che inviano dati ad un'unità di calcolo centrale, ma si tratta di tanti nodi distribuiti in tutto l'ambiente, in grado di fare misurazioni e operazioni distribuite.

Gli ambienti intelligenti sono costituiti da due componenti fondamentali:

- **Array di nano-antenne plasmoniche:** in un modello astratto rappresenterebbero i nodi della rete, sono le principali sorgenti dei segnali, ed eseguono calcoli distribuiti.
- **Array plasmonici riflettenti:** rendono l'ambiente di propagazione attivo, principalmente si occupano di inoltrare le onde inviate dalle nano-antenne, sono in grado comunque di modificarne la direzione.

Gli array plasmonici riflettenti possono essere embedded o applicati sulle superfici degli oggetti, come nell'esempio in figura 1.6 in cui sono disposti sulle pareti come fogli adesivi con un basso costo energetico [6].

L'aspetto del consumo energetico per le periferiche in ottica 6G sarà di estrema rilevanza, infatti ad esempio, sarebbe impensabile dover sostituire frequentemente le batterie delle periferiche interne alle pareti.

Si potrebbe pensare che possa essere comodo collegare tutti i sensori a dei

cavi di alimentazione stabili, ma ciò implicherebbe un eccessivo uso di materiali ed in alcuni casi sarebbe impraticabile.

Prendendo in considerazione l'utilizzo di sensori installati all'interno dei pilastri di un ponte per monitorarne l'agibilità, avremmo bisogno di un sistema di alimentazione molto più flessibile rispetto ad un sistema cablato.

Una soluzione può essere rappresentata da un sistema di ricarica che sfrutta l'energia dei segnali ricevuti per alimentare le proprie batterie.

Tornando all'esempio dei sensori posizionati all'interno dei pilastri di un ponte, si potrebbe pensare che periodicamente in prossimità dei pilastri transino uno o più droni in grado di ricaricare le batterie dei nodi della rete intelligente.

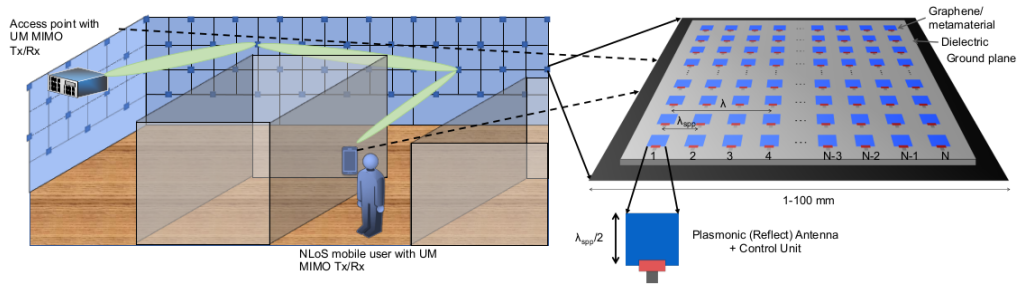


Figura 1.6: Esempio di Ambiente Intelligente con array di nano-antenne plasmoniche.

Gli array plasmonici riflettenti hanno una struttura a strati che permette di agire in diversi modi sul segnale, ad esempio riflettendolo o guidandolo in una certa direzione facendo uso di più strati di materiale plasmonico come in figura 1.7, possono esser immaginati come degli specchi tecnologicamente avanzati che permettono di riflettere le onde SPP con un angolo regolabile. Classifichiamo quindi gli strati che compongono gli array plasmonici riflettenti:

- Livello Metamateriale: rappresenta i metamateriali in due dimensio-

ni, permette di impostare dinamicamente il valore dell'impedenza per orientare il segnale in seguito alla riflessione dello stesso.

- **Livello Rilevamento e Attuazione:** composto da circuiti per cambiare la fase del segnale e da sensori che rilevano l'arrivo di un nuovo segnale mediante l'urto su questo livello.
- **Livello di Schermatura:** isola le estremità della struttura a livelli per limitare le interferenze.
- **Livello Computazionale:** controlla il secondo livello, mediante la gestione dei sensori d'urto ed i phase shifters.
- **Livello di Comunicazione:** connette tutti i livelli superiori e fa da collegamento verso il controller centrale che ha il compito di processare le richieste di connessione e gestire le funzioni d'onda dei segnali.

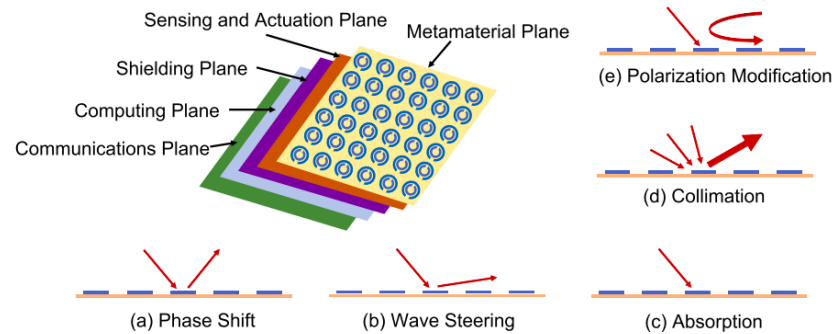


Figura 1.7: Modello di array plasmonico riflettente, in grado di manipolare le onde EM [8].

Per funzionare al meglio questo modello avrà bisogno del supporto dell'Artificial Intelligence per ottimizzare la comunicazione e decidere la direzionalità delle antenne.

In una rete di questo tipo ogni nodo conosce la posizione dei suoi vicini in

modo da poter comunicare con raggi altamente direzionali.

Nonostante i segnali siano estremamente direzionali, per percorrere centinaia di metri, ogni antenna inoltra il pacchetto ai nodi vicini che a loro volta lo ritrasmettono, finché non raggiunge la destinazione.

Gli ambiti di applicazione di questo modello sono molteplici, possono assumere concretezza in sistemi per determinare l'agibilità degli ambienti, ad esempio per rilevare la presenza di sostanze tossiche nell'aria o per l'agibilità sismica.

Altre applicazioni potrebbero essere costituite da sistemi di sorveglianza all'avanguardia, oppure da sistemi per il monitoraggio dello stato di salute, da parte di parenti e medici, di persone anziane.

Capitolo 2

Progetto

2.1 Idea

L'idea è quella di creare una simulazione che rappresenti una rete basata sulla tecnologia 6G, prendendo in considerazione la forte direzionalità del segnale grazie ad UM MIMO ed il basso ritardo sul canale di trasmissione, inferiore ad 1ms.

La topologia scelta per questa rete è data da una struttura a stella, in cui si ha un Master centralizzato e dei Cluster collegati ad esso con dei gate bidirezionali.

Il Master, come una Base Station del mondo reale ha il compito di coordinare la comunicazione, in questo caso avrà anche il compito di alimentare i nodi circostanti con pacchetti particolari.

Il caso d'uso preso in considerazione riguarda gli Ambienti Intelligenti, infatti una rete di cluster facente uso di questo tipo di comunicazione avrà diverse applicazioni con l'avvento del 6G, ad esempio la sicurezza sismica degli ambienti, monitorando con diversi sensori la posizione reciproca.

Tale esempio può essere portato nel mondo reale ponendo dei sensori all'in-

terno delle pareti di un edificio o all'esterno, ad esempio posti nei pilastri di un ponte e verificandone periodicamente i dati registrati, per evidenziare eventuali spostamenti anomali di gruppo che possano pregiudicare l'agibilità dell'ambiente.

I Sensori hanno a disposizione una piccola batteria sufficiente a tenere traccia delle piccole variazioni di posizione.

Per riuscire a comunicare tra loro, eseguendo algoritmi e inviando i risultati al Master, i nodi si avvalgono di un innovativo **sistema di alimentazione**: convertono le onde SPP, emesse dal Master, in tensione elettrica per alimentare le proprie batterie, quando il master "sveglia" i sensori per sapere se l'ambiente è agibile o meno.

L'**invio** e la **ricezione** di istruzioni e dati avviene mediante un sistema basato sul **6G** in modo da garantire una risposta repentina anche a fronte di una grande quantità di dati da processare tra i sensori: in modo **concorrenziale**, schedulando i vari task in modo efficiente e **parallelo**, dato che vi è la possibilità di distribuire i diversi task sui vari sensori.

Paradigma: molte periferiche interconnesse, ciascuna con poca potenza di calcolo che collaborano, mettendo in comunione le risorse, al fine di determinare l'agibilità sismica dell'ambiente di cui sono parte, in altre parole "*L'unione fa la forza*".

2.2 Diagrammi

2.2.1 Use Case Diagram

In figura 2.1 è possibile notare i casi d'uso dei due attori principali del progetto: il *Master* ed i *Sensori*, per chiarezza i sensori nel modello conside-

rato vengono chiamati anche slave perchè è come se fossero delle periferiche del master distribuite nell'ambiente.

Il Master dopo aver alimentato gli slave richiede la struttura della rete, rappresentata da una matrice di adiacenza e una matrice contenente le posizioni dei cluster, ovvero gruppi di slave.

Gli slave, dopo successive request tra i nodi vicini, restituiscono le posizioni dei cluster e la struttura della rete; inoltre nel caso preso come esempio gli slave si occupano anche di determinare l'agibilità dell'ambiente.

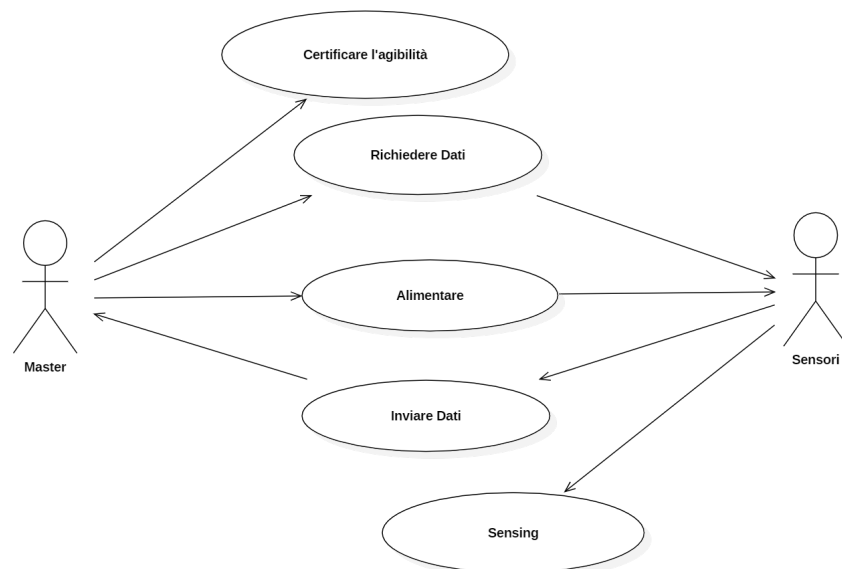


Figura 2.1: Use Case Diagram

Quando i Sensori non devono cooperare vanno in fase di *sleep* o *sensing* in cui semplicemente con la batteria rimanente disattivano tutti i processi inutili e rimangono in attesa di registrare eventuali scosse sismiche. In questa fase il Master non è presente e può venir rimosso per occuparlo in altri contesti.

2.2.2 Class Diagram

Come possiamo vedere in figura 2.2 ogni classe concreta estende le classi astratte *cSimpleModule* e *cMessage* presenti nella libreria dell'ambiente di sviluppo **OMNeT++**, il simulatore utilizzato per questo progetto.

cSimpleModule viene proposta dalla documentazione come la classe base da cui estendere i propri moduli, è molto generica ed ha bisogno dell'override dei principali metodi per funzionare.

cMessage al contrario di *cSimpleModule* è più completa e può funzionare anche senza la necessità di ulteriori implementazioni, anche se alcune modifiche sono state apportate per raggiungere l'obiettivo della simulazione; questa classe, come suggerisce il nome, rappresenta i messaggi o i pacchetti scambiati tra i nodi della rete, proponendo metodi per la personalizzazione degli stessi.

cSimpleModule viene quindi estesa da due classi, che hanno comportamenti diversi, ma entrambe rappresentano i nodi della rete, la classe Master e la classe Slave.

La classe Master come anticipato svolge il ruolo di coordinamento della rete e di alimentazione degli slaves, apprendendo la struttura della rete dopo una fase di Network Detection, descritta in dettaglio nella sezione relativa al Sequence Diagram.

Mentre la classe Slave può rappresentare due tipologie di nodi con caratteristiche molto simili: i Cluster Head e i sottonodi dei cluster.

I Cluster Head sono in minor numero rispetto ai sottonodi, ma hanno più responsabilità, infatti svolgono anche il compito di interfaccia per la comunicazione tra il master e i sottonodi, fatta eccezione per la fase di alimentazione, in cui il Master ha bisogno di comunicare direttamente con i sottonodi, altrimenti il pacchetto da convertire in energia inoltrato dal cluster head

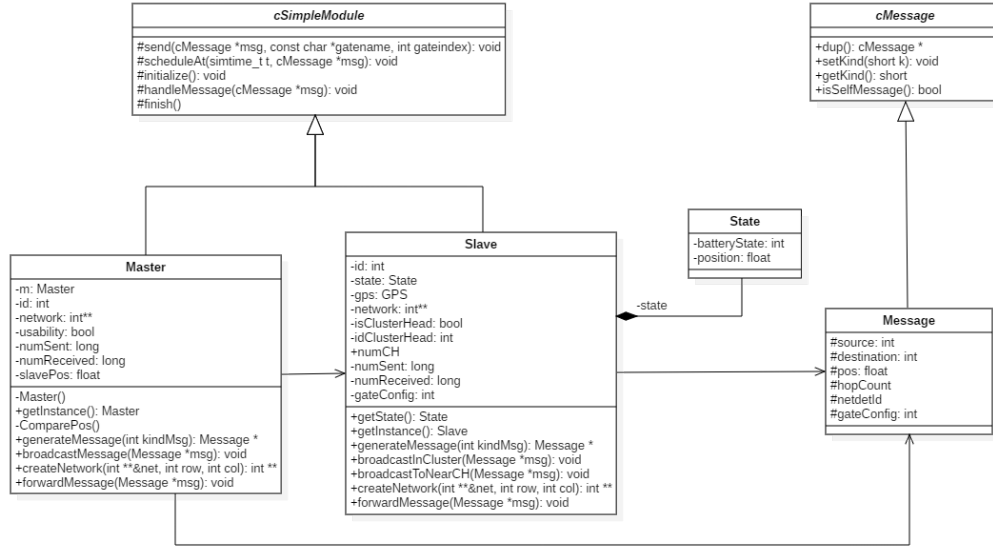


Figura 2.2: Class Diagram

risulterebbe a minor intensità e quindi inefficace, anche a causa del minor numero di antenne di cui dispongono gli slave.

Inoltre è possibile una comunicazione diretta tra i cluster head vicini e i sottotonodi di uno stesso cluster, ciò rende possibile la cooperazione tra gli slave.

La classe **Slave** possiede un puntatore alla classe **State** che, come suggerisce il nome, tiene traccia dello stato corrente della batteria e della posizione.

La classe **Message** infine aggiunge degli attributi utili alla classe base **cMessage**, implementandone i vari metodi getter e setter per il corretto funzionamento.

2.2.3 Sequence Diagram

Il sequence diagram in figura 2.3 rappresenta le interazioni che avvengono tra i vari nodi della rete nel tempo. Possiamo suddividere l'esecuzione in due cicli principali.

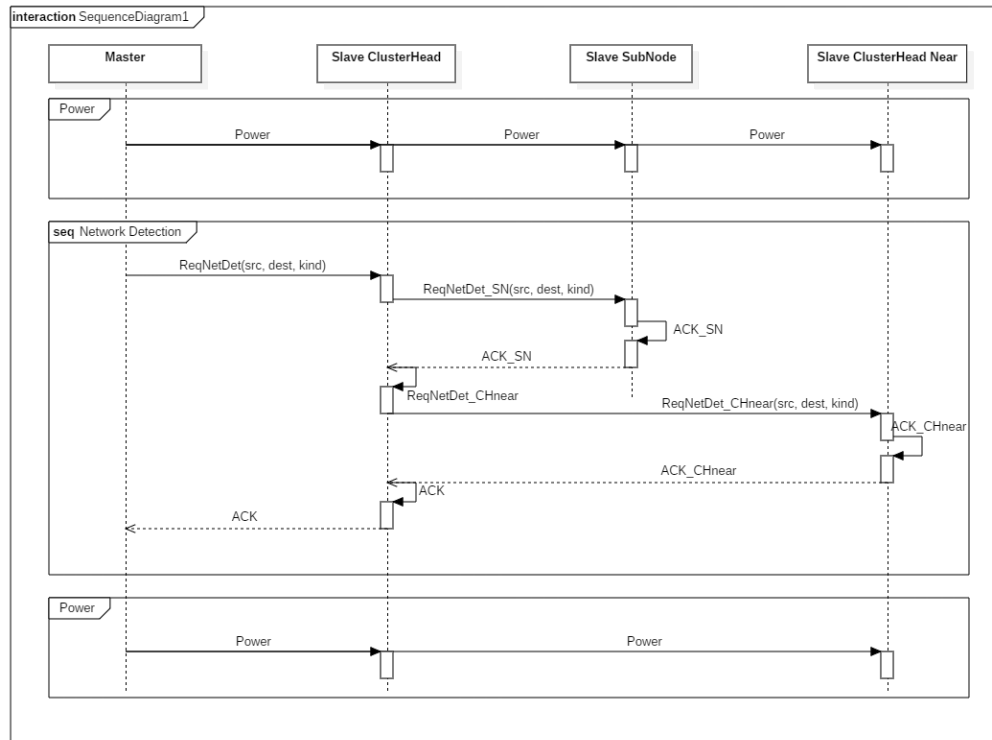


Figura 2.3: Sequence Diagram

- **Alimentazione degli Slave:**

L'alimentazione avviene via wireless, infatti il Master quando si trova nel luogo da certificare illumina con dei pacchetti tutti i volumi che lo circondano per un tempo sufficientemente lungo da alimentare i sensori.

Dal momento che il 6G trasmette un segnale molto direzionale questo viene inviato per settori investendo di energia in modo circolare tutti gli slave, così da dargli la possibilità di rispondere alle successive request. Gli Slave che ricevono tale energia la convertono prontamente in tensione elettrica.

L'energia immagazzinata in questa fase non ne caricherà completamen-

te la batteria, ma solamente quanto basta per eseguire le operazioni di individuazione della rete e comunicazione tra loro.

Ogni volta che uno slave invia un pacchetto ad un altro slave o al master perde parte della propria energia, perciò in un modello reale il master deve fornire l'energia adeguata per permettere agli slave di portare a termine tutte le operazioni richieste.

Si assume che i sensori siano stati installati in assenza di zone d'ombra in modo che l'alimentazione wireless sia realizzabile.

- **Network Detection:**

Per identificare la struttura della rete il Master invia una *Request()* in **broadcast** in tutte le direzioni, in tale messaggio sono inclusi gli identificatori del sender (il master) e del tipo di messaggio (per comunicare che il messaggio è rivolto a tutti i cluster head).

Dopo aver ricevuto le request i cluster head inoltrano la richiesta ai sottonodi, e dopo aver appreso lo stato dei sottonodi con una risposta di acknowledge, la richiesta viene inoltrata anche ai cluster head vicini, in modo da poter compilare le matrici di adiacenza di ogni nodo e i vettori di configurazione dei gate da ritornare al master.

Quando finalmente i sottonodi di ogni cluster hanno risposto al proprio cluster head e i cluster head hanno risposto a tutti i cluster head vicini, allora ogni cluster head ha acquisito dati di posizioni geografiche degli altri nodi e la struttura della rete. Perciò i cluster head sono pronti per rispondere al master con le proprie posizioni e con le configurazioni dei gate dei nodi vicini, in modo da permettere al master di generare automaticamente la matrice di adiacenza di tutta la rete e di compilare la matrice contenente tutte le posizioni dei cluster head.

Non tutte le risposte sono immediate, infatti in molti ack si fa uso di

self messages grazie al metodo *scheduleAt()* fornito da *cMessage*, ciò permette di rispondere dopo un tempo random, evitando così problemi di **deadlock** o stallo, infatti nell'eventualità che due acknowledge ritornino al mittente contemporaneamente, uno di questi inevitabilmente verrebbe scartato. In alternativa al self message la classe *cSimpleModule* fornisce il metodo *sendDelayed()* che permette di inviare direttamente il messaggio, specificando nei parametri formali: il puntatore al messaggio, il delay da applicare prima dell'invio ed il gate su cui instradare il pacchetto.

2.3 Collaborazione tra i Sensori

Infine possiamo notare come grazie a questa interconnessione e comunicazione rapida tra i sensori, si abbia un grosso vantaggio rispetto ad affidare ad ogni sensore il compito di scorrere la lista di tutti i sensori del sistema e confrontare la propria posizione con quella degli altri, ciò andrebbe contro il paradigma che ci siamo preposti. Perciò creando dei cluster ed affidando al Master la coordinazione dei sensori si è permesso ad ogni sensore di contribuire al raggiungimento del risultato, distribuendo il carico computazionale tra tutti i nodi della rete.

Un modello che permette questo tipo di computazione è il **PRAM** (Parallel Random Access Machines), in particolare nella configurazione **Message Passing Model** (figura 2.4) poichè favorisce la portabilità e permette che oltre ai processori anche la memoria sia distribuita, a differenza del *Shared Memory Model* il quale prevede che i diversi processori accedano alla medesima unità di allocazione dati.

Cercando un parallelismo tra i componenti del modello in figura 2.4 e que-

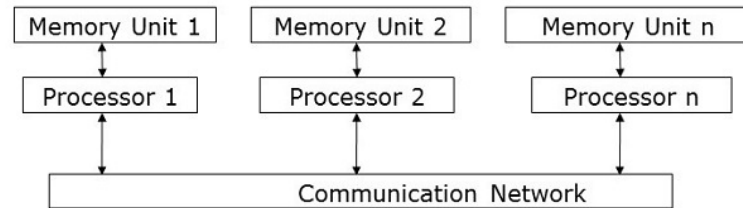


Figura 2.4: Message Passing Model

sto progetto si ha che i processori rispecchiano il comportamento degli slave, le unità di memoria sono i dati salvati nell'oggetto state posseduto da ogni slave, mentre il canale di comunicazione è costituito dal 6G.

2.4 Topologia di rete

La topologia di rete utilizzata in questo progetto consiste in una topologia a stella in cui ogni nodo periferico dà origine ad un cluster.

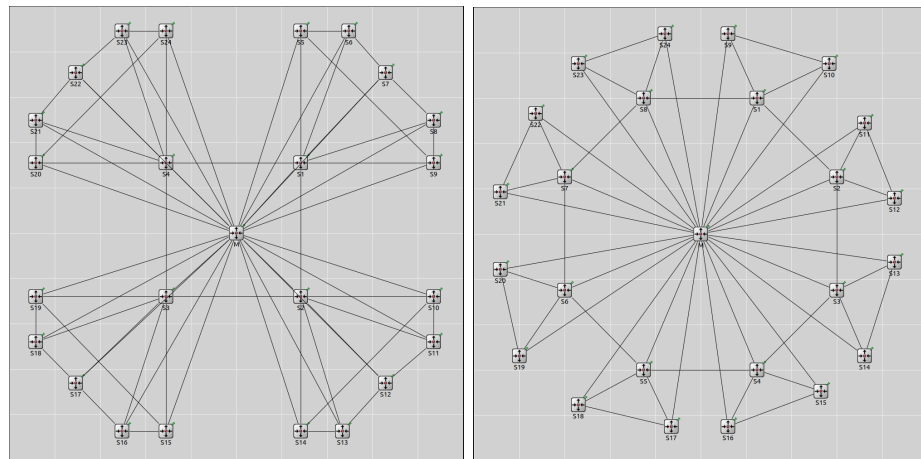
Nella rete si possono trovare tre tipologie di nodi:

- **Master:** un singolo nodo al centro della rete che rappresenta la *base station* del mondo reale, ha il compito di alimentare i nodi con dei pacchetti speciali e di richiedere alla rete lo stato, ovvero il livello delle batterie e la struttura della rete.
- **Cluster Head:** possono essere in quantità variabile ed hanno il compito di inoltrare le richieste dal master verso i sottonodi, inoltre ogni cluster head è interconnesso con i cluster head vicini in modo da permettere una comunicazione completa all'interno della rete.
- **Sub Node:** sono in numero sempre più alto dei cluster head e si occupano di rispondere al proprio cluster head circa il proprio stato.

Inoltre i sottonodi hanno bisogno di minor energia rispetto ai cluster head, poichè inviano un minor numero di pacchetti.

Prendendo in considerazione la topologia appena descritta sono stati fissati 24 nodi oltre al master, ed è stato variato il rapporto tra numero di cluster head e numero di sottonodi.

In figura 2.5 è possibile notare due versioni: una con 4 cluster head e 20 sottonodi e con un'altra con 8 cluster head e 16 sottonodi, in entrambi i casi i sottonodi sono distribuiti in modo omogeneo.



(a) 4 cluster head, 20 sub nodes

(b) 8 cluster head, 16 sub nodes

Figura 2.5: le topologie scelte per la rete.

2.5 Implementazione

Il progetto è stato implementato con il supporto del simulatore **OM-NeT++**, grazie al quale è stato possibile accompagnare il codice *C++* a dei file di tipo *NED* per descrivere la rete con dei nodi, le posizioni degli stessi e i relativi canali di comunicazione, cablati o wireless.

L'architettura del simulatore prevede che vi siano dei modelli di simulazione che vengono realizzati tramite l'impiego di componenti riusabili chiamati moduli, che comunicano tra loro tramite *message passing*.

Possiamo trovare due tipologie di moduli:

- **Simple modules:** integrano la logica della simulazione con codice C++ attraverso delle librerie di supporto messe a disposizione da OMNeT++, si prestano molto ad essere estesi da classi C++ che ne definiscano dei comportamenti specifici in relazione al caso d'uso.
- **Compound modules:** costituiti dalla connessione di moduli semplici facendo uso di **gates**, formando così dei moduli composti.

Le connessioni tra i moduli possono esser arricchite con dei parametri che ne aumentano la verosimiglianza: propagation delay, data rate, bit error rate. Nonostante la tecnologia 6G sia wireless, essendo formata da segnali molto collimati ed altamente direzionali, il canale scelto per la comunicazione tra i nodi della rete nel modello di simulazione è di tipo cablato (gates), con un ritardo di propagazione del segnale inferiore ad 1 ms.

Per eseguire le simulazioni OMNeT++ fornisce oltre ad un **interfaccia grafica** per monitorare lo stato della rete e lo scorrere degli eventi, un **kernel di simulazione** che contiene il codice, gestisce la simulazione e la sua class library.

Prima della fase d'implementazione in C++ è stata scelta una Topologia di rete, le diverse topologie sono state sviluppate nel NED file.

Dopo aver creato la struttura della rete sono stati implementati i metodi relativi all'inizializzazione (*initialize()*) per le classi che estendono *cSimpleModule*, infatti nell'ambiente di simulazione il metodo *initialize()* rappresenta il costruttore dei moduli, mentre il metodo *finish()* rappresenta il distruttore

dei moduli, come per le classi C++, il distruttore viene chiamato automaticamente al termine della simulazione.

Per una comunicazione di base è indispensabile generare un messaggio con il metodo *generateMessage(int kindMsg)* che ritorna un puntatore ad un oggetto di tipo *cMessage*, in questa fase si impostano tutti i campi relativi al messaggio, ad esempio: l'id della sorgente, l'id del destinatario, il tipo di messaggio ed altri campi utili all'esecuzione modificando gli attributi della classe *Message*.

Prima di inviare il messaggio è importante implementare il metodo *handleMessage(cMessage *cmsg)* di cui fare l'*override*, in questo progetto a seconda del tipo di messaggio e del destinatario impostato nel messaggio il nodo ricevente assume comportamenti diversi, e risponde di conseguenza.

A questo punto il pacchetto è pronto per essere inviato, le modalità sono diverse, come anticipato possono esser usati i metodi *send()* o *sendDelayed()*, ma anche *scheduleAt()* con l'utilizzo dei self messages.

Nel caso dei messaggi broadcast ho implementato un metodo che permettesse di inviare delle copie del messaggio originale, grazie al metodo di deep copy fornito da *cMessage*, per evitare che più nodi provassero ad accedere alla medesima locazione di memoria, di seguito c'è un breve esempio di invio dei messaggi in broadcast.

```
void Master::broadcastMessage(Message *msg)
{
    int n = numCH; //number of cluster head
    int k = 1;    //cluster head id
    if(msg->getKind() == 0) //power message kind
        n = gateSize("gate");
    for (int i=0; i<n; i++)
```

```
{  
    Message *copy = msg->dup(); // deep copy  
    if(copy->getKind() == 1) //net detection kind  
        copy->setNetDetId(k);  
    k++;  
    numSent++;  
    float delay = 0;  
    if(copy->getIsLost() && copy->getKind() == 1)  
        delay = retransmitMsg(msg, delay);  
    sendDelayed(copy, delay, "gate$o", i);  
}  
delete msg;  
}
```

Il codice sorgente completo è su **GitHub** al seguente link:

<https://github.com/giacomovitangeli/EarthquakeDetector>

Anche nella classe Slave sono presenti due metodi broadcast per l'invio delle request di network detection ai sottonodi e ai cluster head vicini, viene sempre sfruttato l'uso della *deep copy* per copiare tutti i parametri che definiscono il messaggio, impostati in fase di generazione dello stesso.

I messaggi inviati con metodi di *send()* o *sendDelayed()* non hanno bisogno di essere eliminati con il metodo *delete* nella classe in cui sono stati generati. Infatti in fase di invio il kernel di simulazione si occupa di rimuovere il puntatore all'oggetto generato, dall'oggetto che lo ha generato, per assegnarlo al destinatario mediante il metodo *handleMessage(cMessage *cmsg)*.

In altre parole, prendendo come esempio l'invio dei messaggi nella funzione *broadcastMessage(Message *msg)* riportata, si ha che il messaggio copiato, quando viene inviato viene scollegato del Master, cosicché il kernel di simu-

lazione possa instradarlo sul gate indicato nei parametri formali.

In fase di recezione del messaggio, se questo dovesse arrivare ad un nodo che non è il destinatario, questo mediante il metodo *forwardMessage(Message *msg)*, inoltra il messaggio al nodo più vicino.

Capitolo 3

Test e Risultati

In questo capitolo vengono presi in considerazione i parametri che caratterizzano la rete simulata e ne viene studiato il comportamento in diverse topologie.

I principali parametri d'interesse sono: i consumi energetici e come questi vengono influenzati dalla topologia scelta, la latenza per completare le operazioni di network detection ed infine un'analisi della robustezza della rete a packet loss e guasti temporanei nelle diverse configurazioni.

3.1 Consumi energetici

Uno degli aspetti fondamentali per la realizzazione di reti 6G sarà costituito dall'efficienza energetica, infatti si presuppone che in una rete di questo tipo, i nodi avranno la possibilità di scambiarsi energia attraverso l'invio ripetuto di segnali, convertiti quindi in tensione elettrica.

Il consumo energetico nel simulatore è stato approssimato a delle batterie presenti in ogni nodo, che vengono alimentate dal master e perdono parte della carica ogni volta che inviano un messaggio.

I nodi presi in considerazione per questo tipo di test sono i cluster head e i sottonodi, mentre l'energia del master è stata ipotizzata per semplicità molto più grande di quella consumata dagli altri nodi.

Per capire meglio che benefici portano lo scambio di messaggi ed il conseguente consumo energetico, sono stati affiancati ai grafici dello stato delle batterie altri due indici, per un totale di quattro indicatori che variano in funzione del tempo:

- **Batteria dei cluster head** (*verde*): rappresenta il livello di carica delle batterie contenute nei cluster head.
- **Batteria dei sottonodi** (*arancione*): è il livello di carica delle batterie dei sottonodi.
- **Fattore di interattività** (*rosso*): tiene il conto di quanti pacchetti sono in transito all'interno della rete.
- **Conoscenza della rete** (*blu*): parametro globale che si incrementa dopo ogni acknowledge in fase di network detection, in modo da determinare quanta consapevolezza hanno i nodi della rete in cui si trovano.

In figura 3.1 sono riportati gli indicatori sopra elencati in funzione del tempo di simulazione espresso in secondi.

Per ottenere questi grafici sono stati utilizzati dei metodi forniti dall'ambiente di sviluppo in grado di raccogliere i valori per ogni nodo del modello di simulazione.

Il dataset così formato è stato generato da 100 esecuzioni del modello di simulazione, ottenendo così per ogni parametro considerato 2400 vettori che ne descrivono l'andamento nelle varie simulazioni e nei vari nodi.

OMNeT++ fornisce anche delle funzioni da applicare al dataset per ottenere nel plot risultati più significativi, in particolare in questo test ho applicato in serie le seguenti funzioni:

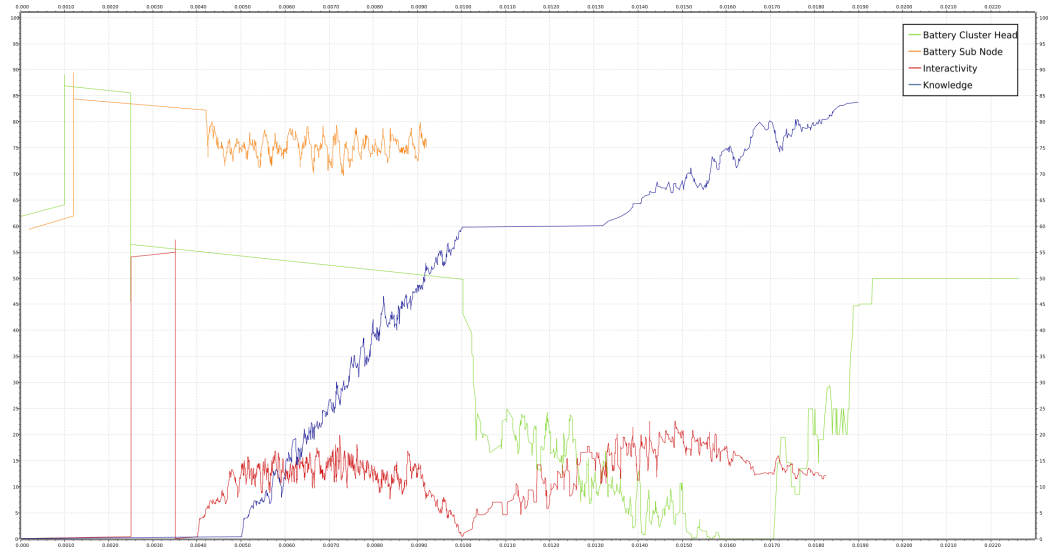
- *Merger*: unisce diversi vettori di dati in uno solo, mantenendo un ordine temporale crescente nell'output.
- *Sliding Window Average*: per calcolare la media dei valori dei vettori uniti nell'intervallo di tempo.

Nei livelli di energia dei nodi, fin da subito si nota come aumentano istantaneamente tutti insieme, questo viene permesso dai pacchetti di energia inviati in broadcast a tutti i nodi dal master.

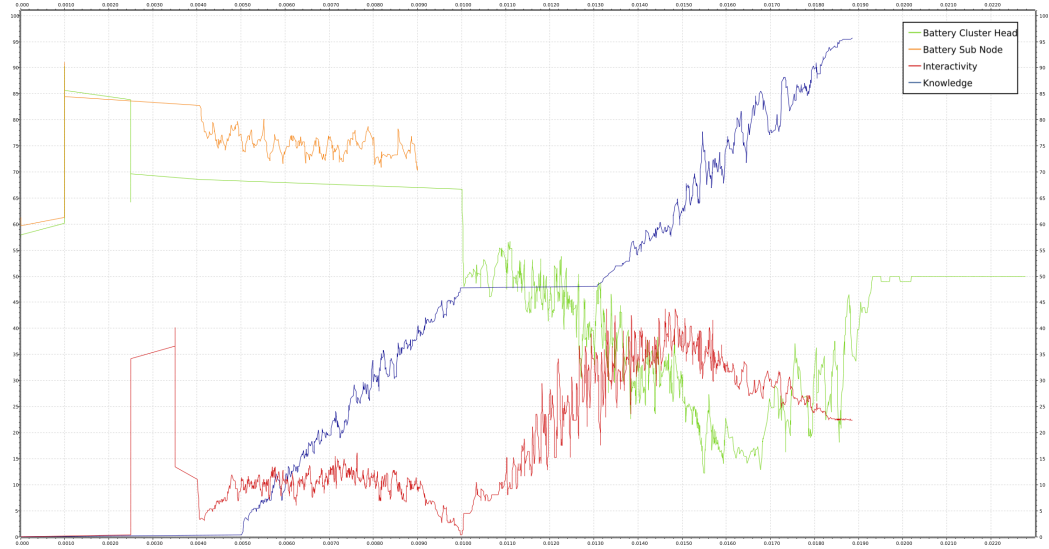
Dopo aver ricevuto l'energia e la request di network detection i cluster head inoltrano sempre in broadcast ai sottonodi del proprio cluster la request del master, perciò nel grafico si può notare come l'energia scende per i cluster head. Confrontando le due topologie si ha inoltre che la diminuzione di energia è minore nella rete con 8 cluster head, poichè ogni cluster head ha un minor numero di sottonodi a cui inoltrare la request, ne consegue che impiega minor energia per questa operazione.

Nel grafico di colore arancione si può notare la diminuzione di energia nei sottonodi quando rispondono ai cluster head con un messaggio di acknowledge. Al fine di evitare situazioni di deadlock in questa fase i sottonodi rispondono dopo un tempo randomizzato, ciò porta ad avere una diminuzione di energia dei sottonodi distribuita nel tempo e non istantanea.

Focalizzando l'attenzione tra 4ms e 5ms si può notare lo sfasamento temporale tra l'inizio della trasmissione dei messaggi di ack a circa 0.004s, caratterizzato dal calo di energia dei sottonodi e dall'aumento del fattore di interattività della rete, ed il momento in cui iniziano ad arrivare tali messag-



(a) Consumo energetico con 4 cluster



(b) Consumo energetico con 8 cluster

Figura 3.1: Consumo energetico.

gi, circa 0.005s. Questo dato grafico ci conferma che il ritardo di propagazione del canale di trasmissione è circa 1ms.

Nel grafico di colore verde si nota un evidente calo di energia poiché il numero di messaggi inviati è triplicato rispetto ai sottonodi, infatti dopo l'ack dai

sottonodi, i cluster head si mettono in comunicazione tra loro con una serie di request e acknowledge fino a completare le reciproche matrici di adiacenza, ed infine inviano un messaggio di ack al master.

Valori troppo elevati del fattore di interattività possono portare a rallentamenti o addirittura a perdite di pacchetti, per questo motivo è importante monitorare questo parametro nelle due topologie.

Durante la fase di comunicazione tra i cluster head si può notare che la rete è molto attiva, in particolare nel modello con 8 cluster, la rete è talmente attiva da portare un disequilibrio.

Il grafico di colore rosso che ci dà informazioni sul numero di pacchetti in circolo nella rete evidenzia come nel modello con 8 cluster il numero di pacchetti attivi aumenta più del doppio rispetto alla comunicazione tra sottonodi.

Questo dato ci porta quindi a considerare questa topologia inefficiente, poiché il carico di lavoro è sbilanciato, al contrario nella topologia con 4 cluster l'interattività della rete è omogenea.

Si può notare come la diminuzione di energia all'interno dei nodi porti benefici in termini di conoscenza della rete (*blu*), infatti dai grafici possiamo notare che tale parametro si incrementa a seguito degli ack.

Confrontando le due topologie sull'aspetto della conoscenza della rete, si nota che sempre nella fase di comunicazione tra i cluster head si ha una crescita più repentina rispetto al primo modello, proprio per la presenza di un maggior numero di nodi e quindi pacchetti scambiati.

Complessivamente la seconda topologia raggiunge la totalità degli ack in media in circa 1/5 di millisecondo in anticipo rispetto alla prima.

3.2 Latenza

La latenza di questo modello di simulazione viene misurata come il tempo che intercorre dal momento in cui il master invia le request di network detection in broadcast ai cluster head, al momento in cui tutti i messaggi di acknowledge ritornano al master.

Come per la comunicazione tra i cluster head, anche per ritornare gli ack finali al master è indispensabile schedulare l'invio dei messaggi in modo da evitare deadlock, anche se questo processo da un lato aumenta i tempi invio dei messaggi, dall'altro migliora l'affidabilità della rete.

Nella tabella 3.1 possiamo vedere un confronto diretto tra i valori di latenza espressa in millisecondi per le diverse topologie proposte.

Le misurazioni sono state fatte su un campione di 100 esecuzioni della simulazione per ogni topologia, quindi ne è stata calcolata la media e la varianza.

n° cluster	AVG latency (ms)	STDEV latency (ms)
4	19.2599	0.8997
8	19.7161	0.6566

Tabella 3.1: Latenza per ogni topologia a 24 nodi

I risultati ottenuti per le diverse topologie mostrano in primo luogo che per eseguire la stessa operazione, ovvero la network detection, la topologia con 4 cluster in media impiega meno tempo rispetto alla topologia ad 8 cluster.

Prendendo in considerazione la deviazione standard si può notare come nella topologia ad 8 cluster ci sia minor dispersione.

Nonostante le misurazioni della latenza nel caso ad 8 cluster portino ad ottenere dei valori più vicini alla media effettiva, e nel caso a 4 cluster si abbia un

valor medio inferiore, bisogna considerare che sommando lo scarto quadratico medio al valore atteso si ottiene che i tempi di esecuzione per la network detection, in alcuni casi, possono coincidere.

3.3 Packet Loss

La robustezza della rete a perdite di pacchetti e guasti temporanei è stata simulata con l'introduzione di una percentuale di packet loss fissata da confrontare con un numero random da 1 a 100 che viene estratto per ogni messaggio al momento della generazione dello stesso.

In caso di perdita del pacchetto viene generato un nuovo messaggio per la ritrasmissione dell'informazione. Per evitare perdite consecutive di pacchetti, una volta che si ha una perdita, il nodo sender attende un tempo random all'interno un certo intervallo, quindi ritrasmette il messaggio.

L'attesa prima della ritrasmissione è necessaria per evitare di appesantire il traffico di pacchetti nella rete, nel caso in cui il pacchetto venga perso per eccesso di pacchetti sul canale o per un guasto temporaneo.

Per questo test sono state prese in considerazione le due topologie a 4 e 8 cluster, confrontandone il comportamento in situazioni di "stress", ovvero in casi di inverosimile eccesso di perdite di pacchetti per capirne il comportamento in situazioni critiche.

Sono stati considerati dei campioni di 100 simulazioni per ogni topologia e per ogni percentuale di packet loss, i vettori di dati estratti sono stati uniti con la funzione *Merger* e ne è stata fatta la media con la funzione *Sliding Window Average*.

Per semplicità i seguenti grafici si riferiscono ai pacchetti persi dai nodi di tipo cluster head e sottonodi dei cluster, mentre il master è stato omesso poi-

ché rimane costante in entrambe le configurazioni.

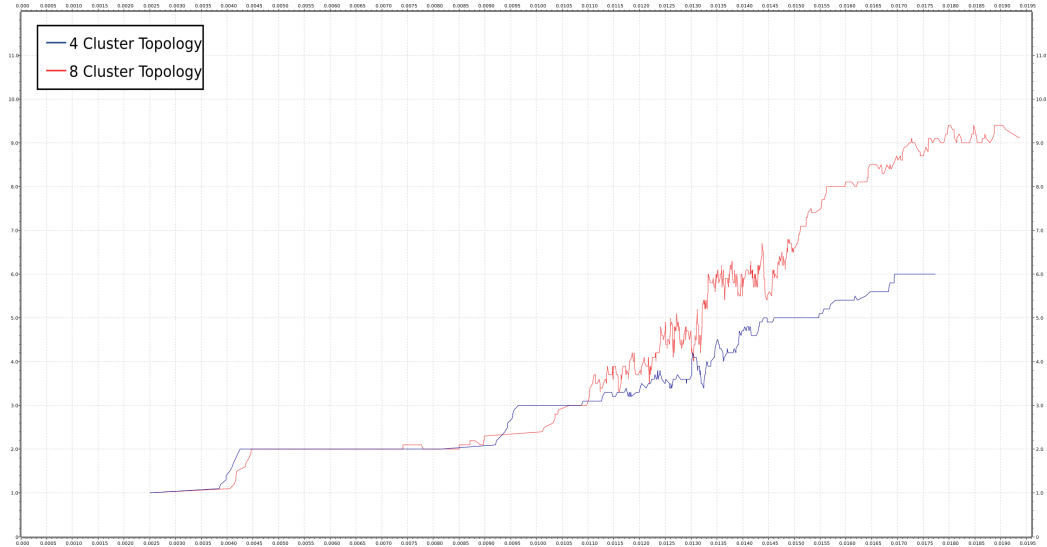


Figura 3.2: Packet Loss con probabilità del 20%

Nel grafico in figura 3.2 sono presenti gli andamenti di packet loss con una percentuale del 20% nelle configurazioni a 4 cluster (*blu*) ed a 8 cluster (*rosso*).

Risulta evidente che nella prima fase di network detection, ovvero quando i sottonodi rispondono ai cluster head, in entrambe le configurazioni non si hanno molte perdite, in media 2 pacchetti persi in tutta la rete.

Nella fase di comunicazione tra i cluster head è invece notevole l'incremento di packet loss, come ci si poteva aspettare la crescita è più rapida nella topologia con 8 cluster head a causa del maggior numero di messaggi scambiati. Un altro dato di interesse è il tempo in cui terminano le perdite nelle due topologie, infatti nella configurazione a 4 cluster non si hanno più segnali di perdita dopo 17.75ms mentre per gli 8 cluster head si arriva quasi a raggiungere 19.5ms. Questa differenza di quasi 2ms è data dai tempi di ritrasmissione che rallentano la rete in cui si verificano un maggior numero di perdite.

I valori massimi di pacchetti persi raggiunti nelle due topologie sono: 6 per la configurazione a 4 cluster, mentre per le reti con 8 cluster raggiungono in media circa 9.5 pacchetti persi.

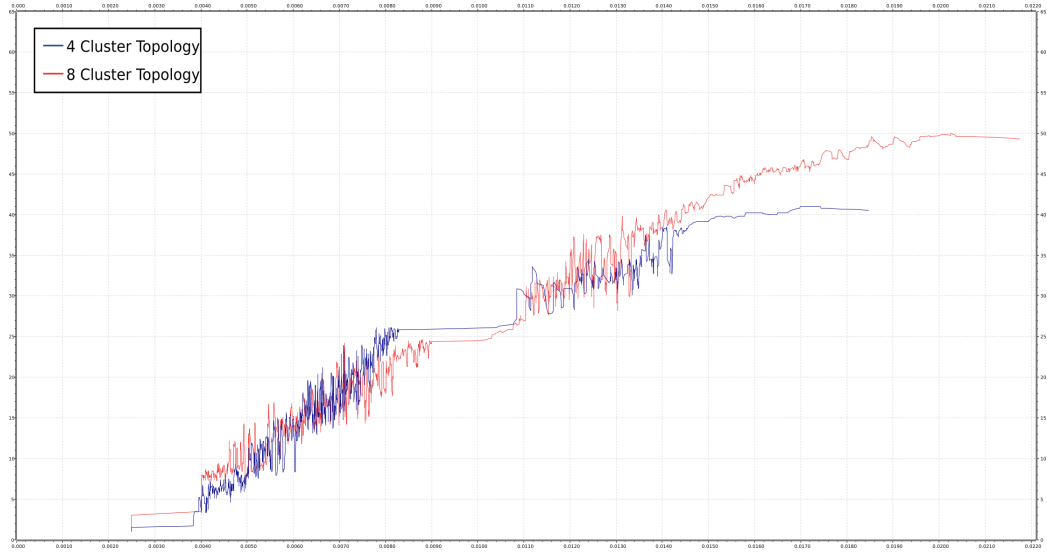


Figura 3.3: Packet Loss con probabilità del 50%

Vediamo quindi in figura 3.3 come variano i valori nelle due diverse topologie aumentando al 50% la probabilità di perdere pacchetti.

Rispetto al caso appena considerato, si nota che adesso anche la comunicazione con i sottogruppi se pur richiedendo un minor numero di pacchetti produce grandi perdite, sia di pacchetti che di tempo.

Le due funzioni tendono a coincidere nella prima fase, totalizzando un numero di perdite che va da 5 a circa 25 pacchetti persi entro i primi 9ms di esecuzione.

Nella seconda fase, al contrario dell'esempio precedente la pendenza di entrambe le curve diminuiscono, raggiungendo un punto di massimo (in media) al termine della simulazione. La configurazione a 4 cluster supera di poco i 40 pacchetti persi, mentre la configurazione con 8 cluster raggiunge le 50 perdite.

Anche in questo caso l'ultimo segnale di variazione di packet loss emesso, in termini di tempo, appartiene alla topologia con 8 cluster che a causa del delay aggiunto dalle ritrasmissioni raggiunge circa 21.5ms.

Eseguendo invece il simulatore con una percentuale di packet loss più verosimile, ad esempio ipotizzando nel caso peggiore dei valori compresi tra l'1% e il 5% si ottiene che vengono persi al più 1-2 pacchetti, per il ridotto numero di pacchetti scambiati in questa simulazione di network detection.

Valori verosimili dell'1% di packet loss portati nel mondo reale e scalati su comunicazioni formate da centinaia di migliaia di pacchetti al minuto sono risultati piuttosto critici.

Nel mondo reale inoltre questa percentuale di packet loss non è temporaria e non costante come abbiamo assunto per semplicità nel modello di simulazione.

Come abbiamo visto nel primo capitolo sulla Tecnologia 6G, il packet loss risente molto della distanza tra i nodi, perciò sarà fondamentale interconnettere quanti più oggetti, diminuendo le distanze che le onde SPP dovranno percorrere per limitare perdite di pacchetti, rendendo così più affidabili questo tipo di connessioni wireless.

Conclusioni

Al termine della sperimentazione sulla tecnologia 6G, prevista intorno al 2030, il modo con cui l'uomo percepisce ed interagisce con il vasto mondo dell'informazione cambierà radicalmente.

Lo sviluppo di questa tecnologia comporterà molteplici applicazioni che potranno essere realizzate con l'utilizzo di nuovi materiali e componenti sempre più interconnessi .

Possiamo immaginare alcuni esempi, come il superamento degli attuali smart-phones con dispositivi di proiezione olografica 3D integrati al nostro abbigliamento oppure la diffusione di veicoli a guida autonoma interconnessi con le strade, sia per ottenere un maggior livello di sicurezza, ma anche per ottimizzare l'efficienza energetica che potrà'essere erogata per induzione direttamente al veicolo.

Gli edifici e le infrastrutture, ponti e gallerie, saranno connessi tra loro e potranno comunicare tra loro per monitorare il proprio stato di usura, comunicando ad esempio la necessità di immediata manutenzione a determinate componenti.

In quest'ottica anche la parete di un palazzo sarà in grado di veicolare informazioni grazie tecnologia 6G, sfruttando array di nano antenne plasmoniche e array plasmonici riflettenti.

Un semplice muro che fino ad oggi è stato solo da ostacolo per le comunica-

zioni wireless, sarà in grado di ricoprire un ruolo attivo all'interno della rete, comunicando e raccogliendo informazioni utili.

Una tecnologia di questo tipo potrà prevenire anche grandi disastri, come il crollo del ponte Morandi di Genova il 14 agosto 2018, dove a causa della mancata manutenzione di un tirante danneggiato dal fenomeno di corrosione, 43 persone hanno perso la vita.

Un aspetto che però non dovrà venir mai trascurato nello sviluppo di questa nuova tecnologia è il rispetto della salute umana.

L'utilizzo di onde estremamente direzionali ed ultra dense in luoghi affollati infatti, almeno in linea teorica, potrebbe portare a delle ripercussioni negative sulla salute.

Oltre alla salute, anche la sicurezza e la privacy delle persone devono essere salvaguardate, in particolare nel caso di applicazioni in ambito medico diagnostico.

Per un'ottima resa di questa tecnologia si potrebbe pensare ad un mondo interconnesso con reti peer to peer decentralizzate, in modo da garantire ad ogni individuo privacy e sicurezza, allo stesso tempo dare libertà ai nodi della rete, senza la costante dipendenza da server centrali.

Bibliografia

- [1] Chan-Byoung Chae et al. “Adaptive MIMO transmission techniques for broadband wireless communication systems [Topics in Wireless Communications]”. In: *IEEE Communications Magazine* 48.5 (2010), pp. 112–118.
- [2] Yuan Yifei e Zhu Longming. “Application scenarios and enabling technologies of 5G”. In: *China Communications* 11.11 (2014), pp. 69–79.
- [3] X. Li et al. “Transmission Scheme with Limited Channel State Information Feedback for 3D MIMO System”. In: *International Journal of Antennas and Propagation* 2015 (2015), pp. 1–12.
- [4] Ian F. Akyildiz e Josep Miquel Jornet. “Realizing Ultra-Massive MIMO (1024×1024) communication in the (0.06–10) Terahertz band”. In: *Nano Communication Networks* 8 (2016). Electromagnetic Communication in Nano-scale, pp. 46–54.
- [5] Chong Han, Josep Miquel Jornet e Ian Akyildiz. “Ultra-Massive MIMO Channel Modeling for Graphene-Enabled Terahertz-Band Communications”. In: *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. 2018, pp. 1–5.
- [6] Shuai Nie, Josep M. Jornet e Ian F. Akyildiz. “Intelligent Environments Based on Ultra-massive MIMO Platforms for Wireless Commu-

- nication in Millimeter Wave and Terahertz Bands”. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 7849–7853.
- [7] Ping Yang et al. “6G Wireless Communications: Vision and Potential Techniques”. In: *IEEE Network* 33.4 (2019), pp. 70–75.
- [8] Ian F. Akyildiz, Ahan Kak e Shuai Nie. “6G and Beyond: The Future of Wireless Communications Systems”. In: *IEEE Access* 8 (2020), pp. 133995–134030.
- [9] Samsung Research. “6G The Next Hyper-Connected Experience for All”. In: *Samsung 6G Vision* (2020).
- [10] Arjun Singh et al. “Design and Operation of a Graphene-Based Plasmonic Nano-Antenna Array for Communication in the Terahertz Band”. In: *IEEE Journal on Selected Areas in Communications* 38.9 (2020), pp. 2104–2117.
- [11] Qing Le Zhang et al. “High-Gain Millimeter-Wave Antennas Based on Spoof Surface Plasmon Polaritons”. In: *IEEE Transactions on Antennas and Propagation* 68.6 (2020), pp. 4320–4331.