

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea in Informatica

**PROGETTAZIONE E
IMPLEMENTAZIONE DI UNA
RETE DI SENSORI MOBILI
ARDUINO PER SCENARI
DI DISASTER RECOVERY**

Tesi di Laurea in Sistemi e Reti Wireless

Relatore:
Chiar.mo Prof.
Luciano Bononi

Presentata da:
Matteo Martelli

Correlatori:
Marco Di Felice
Angelo Trotta

**Sessione I
Anno Accademico 2012/2013**

Indice

I	Stato dell'Arte	5
1	Disaster Recovery	7
1.1	Classificazione dei disastri	7
1.2	Classificazione dei sistemi	8
1.3	Misure di Controllo	8
1.4	DR nelle Reti Wireless	9
2	Reti ad-hoc e Radio Cognitive	11
2.1	Reti wireless ad-hoc	11
2.1.1	Reti wireless ad-hoc nel Disaster Recovery	11
2.2	Radio Cognitive	12
2.2.1	Caratteristiche delle Radio Cognitive	13
2.2.2	Radio Cognitive nel Disaster Recovery	16
3	STEM-Net	19
3.1	Riconfigurazione Protocollore (intra-stack)	19
3.2	Multimodalità (riconfigurabilità protocollore inter-stack)	20
3.3	Apprendimento	20
3.4	Cooperazione/Coordinazione	21
3.5	Scenari d'uso	21
3.5.1	STEM-Net nel Disaster Recovery	22
4	STEM-Mesh	25
4.1	Architettura degli STEM-Node	27
4.2	Algoritmo di Controllo Distribuito	27
4.3	Prototipo dell'unità STEM-Node	31
II	Progettazione e Implementazione	33
5	Piano di Processo	35

5.1	Scenario di Progetto	35
5.2	Analisi dei Requisiti	36
5.2.1	Rilevamento della potenza di segnale	36
5.2.2	Controllo del movimento	36
5.2.3	Routing del Traffico	36
5.3	Risorse e strumenti impiegati	37
6	Progettazione	39
6.1	Client UDP Python	41
6.2	Programma Wiring per Arduino	41
7	Implementazione	43
7.1	Implementazione del client UDP Python	43
7.2	Implementazione del core per Arduino	44
8	Risultati Sperimentali	49
8.1	Raccolta dati	49
8.2	Fase di test	49
8.3	Codice Sorgente	52
9	Conclusioni e Sviluppi Futuri	53

Sommario

Il progetto descritto in questo documento consiste nella realizzazione di una prima applicazione pratica di uno specifico studio di ricerca rivolto al ripristino di reti wireless in scenari post-calamità naturali. In principio è stata descritta un'ampia analisi delle problematiche di rete che si vengono a creare in seguito ad eventi catastrofici. Successivamente, analizzando le varie tecniche e tecnologie oggetto di studio di diversi gruppi di ricerca, si è scelto di collaborare con il progetto STEM-Mesh, essendo ancora in fase sperimentale, il quale affronta il problema di ristabilire la connettività di rete in questi particolari scenari, attraverso l'utilizzo di tecnologie Cognitive Radio (CR), mobilità controllata e principi di reti auto-organizzanti. Di questo primo approccio pratico sono state poi descritte le fasi di progettazione, implementazione e testing. Nella fase di progettazione sono state studiate le componenti hardware e software che rispettassero il più possibile i requisiti e le caratteristiche dei dispositivi "staminali" STEM-Node cuore del progetto STEM-Mesh, ovvero dei dispositivi wireless altamente auto-riconfiguranti ed auto-organizzanti che possono diventare dispositivi sostituivi ai nodi compromessi in una rete, riconfigurandosi appunto in base alle funzionalità interrotte. Nella fase di implementazione si è passati alla stesura del codice, in Python e Wiring, abilitante il dispositivo STEM-Node. Infine nella fase di testing si è verificato che i risultati fossero quelli desiderati e che il sistema realizzato funzionasse come previsto.

Introduzione

I recenti disastri naturali a livello mondiale e nazionale, come i frequenti terremoti in Giappone e in Italia (Abruzzo 2009, Emilia Romagna 2012), fanno riflettere sull'inadeguatezza delle reti di comunicazione esistenti e sulla mancanza di piani di emergenza rivolti al loro ripristino per far fronte a queste situazioni di emergenza. Dalle ultime analisi è stato riscontrato che nelle situazioni meno critiche le strutture di comunicazione non venivano danneggiate, ma i loro servizi di rete venivano gravemente compromessi dall'eccessivo carico di traffico generato dagli utenti finali. Al contrario nelle situazioni più drammatiche, si è notato che la maggior parte delle strutture di comunicazione venivano distrutte e la rete wireless si riduceva ad un insieme di sparse isole di connessione. In base alle passate esperienze sulle reazioni delle reti di comunicazione in scenari post-calamità, sono partiti degli studi sul problema di ristabilire la connettività in queste situazioni critiche. Tutto ciò fa parte del processo di *Disaster Recovery (DR)*. La sua importanza è direttamente correlata ai benefici che ne comporta quali: la divulgazione dell'informazione tramite la rete globale, la possibilità di coordinare operazioni di soccorso e la possibilità di effettuare operazioni di rilevazione dell'ambiente e del territorio disastroso. Ad esempio sarebbe utile che i sopravvissuti potessero comunicare con i soccorritori tramite una qualsiasi rete già presente o di supporto che sia in grado di riconfigurarsi in base alle tecnologie wireless in possesso dai primi. Sarebbe inoltre interessante far fronte al problema dell'interoperabilità tra organizzazioni di pubblica sicurezza, organizzazioni umanitarie e associazioni di primo intervento, per poter offrire linee di emergenza che permettano ai vari enti di comunicare fra di loro senza incorrere nel problema della sovrapposizione di frequenze radio. A questo proposito si è rilevato utile lo studio delle radio cognitive sotto l'aspetto dell'uso opportunistico dello spettro radio e l'interoperabilità della rete. Le radio cognitive sono state esaminate anche nel progetto STEM-Net [2], il cui obiettivo è stato quello di offrire un modello di dispositivi "staminali" (STEM-Node) che potessero auto organizzarsi, in modo da gestire l'intera rete e modificare la loro configurazione senza l'intervento umano. Quest'ultimo progetto non è nato per il disaster recovery, ma per risolvere alcuni problemi che si presentano nella gestione delle nuove reti wireless; queste solitamente sono complesse e composte da più stazioni e tecnologie differenti. Al contrario negli ultimi mesi è nato il progetto STEM-Mesh [5] che, utilizzando i dispositivi "staminali" di STEM-Net, ha proprio come

obiettivo offrire una soluzione per ristabilire la connettività di una rete a seguito di un disastro naturale, la quale sia in grado di combinare le tecnologie Cognitive Radio con la teoria di mobilità controllata e con i principi delle reti ad-hoc. Essendo quest'ultimo un progetto ancora in fase di sperimentazione, ho deciso di contribuire alla realizzazione di una sua applicazione pratica. Nello specifico ho sviluppato un nodo wireless staminale utilizzando una piattaforma Arduino e vari componenti hardware ben interfacciabili ad essa. Tra questi ho adoperato un unità mobile provvista di due ruote motorizzate e un modulo Wi-Fi. Sono passato alla fase di programmazione di questi componenti solo dopo aver studiato come interfacciarli tra di loro e come utilizzare le librerie necessarie al loro corretto funzionamento. Infine nella fase di testing ho verificato che questo dispositivo fungesse da collegamento tra nodi Wi-Fi in una rete ad-hoc e che cercasse la posizione ottimale rispetto alla qualità del collegamento.

Parte I

Stato dell'Arte

Capitolo 1

Disaster Recovery

Generalmente il Disaster Recovery include tutte le misure tecnologiche e organizzative/logistiche atte a ripristinare sistemi, dati e infrastrutture necessarie all'erogazione di servizi di business per imprese, associazioni o enti, a fronte di gravi emergenze che ne intacchino la regolare attività. Tali misure vengono esplicitate tramite il documento *Disaster Recovery Plan (DRP)*. Nella stesura del piano vengono innanzitutto analizzati i possibili livelli di disastro e la criticità dei sistemi/applicazioni. Successivamente vengono stabilite le misure di controllo per la prevenzione di disastri o per ridurre al minimo i rischi in tali situazioni.

1.1 Classificazione dei disastri

Nella stesura di un piano di disaster recovery, i disastri vengono classificati in due categorie:

- **Disastri naturali**

Sono appunto le calamità naturali come alluvioni, uragani, tornado e terremoti.

- **Disastri causati dall'uomo**

Tra questi ci sono dispersione di materiale pericoloso o tossico, guasto delle infrastrutture o bioterrorismo.

Prevenire i disastri non è semplice e a volte quasi impossibile, è quindi molto importante cercare di stendere dei buoni piani di emergenza per ridurre o contenere i danni e le perdite.

1.2 Classificazione dei sistemi

Nella stesura di un piano di disaster recovery, i sistemi vengono classificati in:

- **Critici**

Un sistema è considerato critico se le sue funzioni possono essere eseguite solo da un altro sistema con le medesime caratteristiche e non possono essere svolte manualmente. La tolleranza in caso di interruzione è molto bassa e il costo di una interruzione è molto alto.

- **Vitali**

Un sistema è considerato vitale se le sue funzioni possono essere svolte manualmente, ma solo per un limitato periodo di tempo. Rispetto ai sistemi critici, la tolleranza in caso di interruzione è più alta mentre il costo di una interruzione è più basso.

- **Delicati**

Un sistema è considerato delicato se le sue funzioni possono essere svolte manualmente, a costi tollerabili, per un lungo periodo di tempo. Nonostante queste funzioni possano essere eseguite manualmente, il loro svolgimento richiede comunque l'impiego di un numero di persone maggiore rispetto a quello previsto in condizioni normali.

- **Non-Critici**

Un sistema è considerato non critico se le sue funzioni possono rimanere interrotte senza compromettere gli altri sistemi e se il costo dell'interruzione è nullo o comunque molto modesto. Inoltre anche lo sforzo e il costo di ripristino del sistema devono essere contenuti.

Ovviamente la priorità di ripristino dei sistema è direttamente proporzionale al loro livello di criticità.

1.3 Misure di Controllo

Le misure di controllo sono quei meccanismi che tentano di ridurre o eliminare i rischi nei disastri. Queste si possono classificare in tre tipi:

- **Misure di prevenzione**

Sono quei controlli il cui scopo è prevenire la nascita di un evento.

- **Misure di ispezione**

Sono i controlli il cui scopo è scoprire eventuali disastri indesiderati, nel caso in cui non fossero immediatamente rilevati (o di facile rilevazione).

- **Misure correttive**

Sono quei controlli il cui scopo è ripristinare il sistema e le infrastrutture dopo un disastro.

Un buon piano di disaster recovery prevede che questi tre tipi di controlli siano ben documentati e testati regolarmente.

1.4 DR nelle Reti Wireless

In molti contesti la tecnologia odierna offre la possibilità di realizzare varie soluzioni di continuità e Disaster Recovery, fino alla garanzia di un'erogazione continua dei servizi informatici. Si pensi ad esempio ad un'azienda/organizzazione la quale maggior parte della sua attività economica o informativa viene gestita via web (sito internet e/o database), in questi casi si adotta la tecnica di *ridondare* i dati di maggiore importanza in un "sito secondario" o "sito di Disaster Recovery" affinché, nel caso in cui il sito primario venisse messo fuori uso a seguito di un disastro, si possano attivare le funzionalità del sito secondario nel minor tempo possibile. Le tecniche di Disaster Recovery sono comunque ancora oggetto di ricerca, in quanto in altri contesti è ancora difficile avere una soluzione informatica automatizzata e in tempi rapidi. Si consideri lo scenario in cui a seguito di un disastro naturale, la connettività di un'area venga compromessa (ad esempio alcuni nodi della rete subiscono dei guasti fisici). In questi casi i problemi da considerare possono essere molti, tra cui:

- **Eterogeneità della rete**

Guardando ai recenti sviluppi delle reti wireless, si nota chiaramente che si sta passando da sistemi composti da una singola tecnologia, poche stazioni di trasmissione a larga copertura e posizionate in luoghi strategici, a sistemi più complessi composti da molte stazioni, tecnologie e attori. Inoltre con il sopraggiungere di nuove tecnologie come Wi-Fi e WiMax, si è visto incrementare il numero di tecnologie wireless differenti presenti nelle reti. Tra l'altro i nuovi sistemi sono quasi sempre gestiti, allo stesso tempo, da più attori come compagnie telefoniche, nuovi ISP (Internet Service Providers), amministrazioni locali, associazioni civiche e così via. Tutti questi aspetti fanno intuire come può essere complessa la struttura dei nuovi sistemi.

- **Personale tecnico inadeguato**

Poiché le tecnologie sono molteplici, è molto difficile trovare personale qualificato che riesca a gestire un problema che coinvolge l'intera rete o una sua buona parte. Sarebbero necessari operatori che, oltre ad essere qualificati su tutte le tecnologie presenti, dovrebbero anche avere accesso alle disposizioni di rete e di gestione di tutti gli attori in gioco.

- **Mobilità ridotta**

C'è da considerare anche che, a seguito di calamità naturali, la viabilità dei trasporti possa essere ridotta o che alcune aree possano essere totalmente inagibili. Questo potrebbe intralciare o impedire del tutto gli interventi manuali sulle stazioni fisiche della rete.

Da queste problematiche sono nati vari studi di ricerca, di cui molti hanno analizzato le potenzialità che potrebbero avere le reti ad-hoc per supportare la comunicazione multi-hop nelle misure di disaster recovery. Tra questi gli aspetti più interessanti sono: protocolli di routing adattivi [7] [12], strategie di sensing e collezione dei dati [9] e algoritmi di mobilità controllata per creare strutture di comunicazione auto-organizzanti per le operazioni di ricerca e salvataggio. Si è iniziato anche a prendere in considerazione la possibilità di utilizzare dei robot per esplorare aree sconosciute senza rompere la maglia di connettività tra di essi [4]. Successivamente, oltre alle reti ad-hoc, sono state studiate possibili applicazioni delle tecnologie *Cognitive Radio (CR)* [1] negli scenari di DR, per far fronte ad alcune questioni quali la massimizzazione della copertura delle reti di robot mobili, la scarsità di spettro e l'interoperabilità di rete in scenari di emergenza [3] [8]. Le radio cognitive sono state esaminate anche nel progetto STEM-Net [2], il cui obiettivo è stato quello di offrire un modello di dispositivi *staminali* (STEM-Node) che potessero auto organizzarsi, in modo da gestire l'intera rete e modificare la loro configurazione senza l'intervento umano. Negli ultimi mesi, partendo dai risultati di STEM-Net è nato anche il lavoro STEM-Mesh [5], su cui si basa il progetto che verrà descritto nel Capitolo 5. L'obiettivo di STEM-Mesh è offrire una soluzione per ristabilire la connettività di una rete a seguito di un disastro naturale, utilizzando una maglia (da cui Mesh) di robot mobili, i quali sono costituiti dalla tecnologia dei dispositivi staminali descritti in STEM-Net. Si avrebbe quindi una soluzione che combini le tecnologie di Cognitive Radio con la teoria di mobilità controllata e con i principi delle reti ad-hoc.

Capitolo 2

Reti ad-hoc e Radio Cognitive

2.1 Reti wireless ad-hoc

Una rete wireless ad-hoc è una rete wireless di tipo decentralizzato [11]. Una rete è *ad hoc* quando non si basa su una struttura preesistente, come può essere un router o un access point. Infatti ogni nodo partecipa al routing della rete inoltrando i pacchetti della comunicazione ad altri nodi interessati, che vengono determinati dinamicamente in base alle regole e ai protocolli specifici della rete. Oltre al routing come tecnica di inoltro pacchetti, i nodi di una rete ad-hoc possono inoltrare i pacchetti verso tutti gli altri nodi visibili (flooding).

2.1.1 Reti wireless ad-hoc nel Disaster Recovery

Si è visto come le reti wireless ad-hoc sono state e continuano ad essere prese fortemente in considerazione per la ricerca sul Disaster Recovery. Questo perché in situazioni di emergenza si può creare una rete wireless ad-hoc di nodi mobili senza la necessità di appoggiarsi ad infrastrutture già esistenti, le quali potrebbero essere state danneggiate da un disastro o potrebbero essere totalmente assenti (ad esempio operazioni di recupero in aree sprovviste di connettività). Un altro aspetto importante per i nodi di queste reti è mantenere un consumo energetico basso, in quanto si vuole ottenere un'autonomia di movimento più alta possibile. Per queste ragioni i protocolli di routing per le reti ad-hoc hanno un'importanza fondamentale. Inoltre, mentre la mobilità richiede uno scambio di informazioni costante tra i nodi, la scarsità di risorse energetiche richiede che tale scambio venga limitato il più possibile. I nodi, tra l'altro, non conoscono a priori come è composta la rete in cui si trovano, devono per cui scoprirlo dinamicamente comunicando tra di loro. In genere ogni nodo annuncia periodicamente la sua presenza nella rete e ne ascolta il traffico, arrivando a conoscere gli altri nodi che si sono presentati e il modo

attraverso cui raggiungerli; ciò potrebbe anche essere annunciato in broadcast. Esistono vari tipi di protocolli di routing per le reti ad-hoc mobili, che possono essere classificati in:

- **Proactive**

Permettono di avere un immediato forwarding dei pacchetti in quanto ogni nodo possiede una tabella di routing costantemente aggiornata con le informazioni sui collegamenti verso tutti gli altri nodi. C'è però un overhead considerevole anche quando non vi è traffico da smaltire in rete, perché comunque i nodi devono continuamente scambiarsi informazioni per tenere aggiornate le tabelle.

- **Reactive**

Nei protocolli reactive, invece, il routing viene stabilito *on demand*, ovvero i nodi vengono a conoscenza del destinatario solo quando devono comunicarci. Quindi se non c'è traffico di dati nella rete non c'è neppure traffico di controllo per il routing, ma si deve considerare una latenza iniziale per la fase di route discovery quando si inizia una sessione di traffico dati.

- **Ibridi**

Visto che i protocolli proactive sono più vantaggiosi in termini di rapidità di forwarding e quelli reactive sono più vantaggiosi in termini di consumo di risorse, non possono che esistere protocolli ibridi che cercano di ottimizzarli e sintetizzarli, usando il primo tipo per i nodi “vicini” e il secondo per quelli “lontani”. Alcuni protocolli utilizzano il *GPS* per determinare la posizione dei nodi e stabilire quindi la distanza tra di essi.

- **Hierarchical**

In questi protocolli si suddivide la rete in cluster in cui viene scelto un capo cluster. All'interno dei cluster si utilizzano tecniche proactive, mentre per le comunicazioni tra i vari cluster si usano tecniche reactive dove i pacchetti vengono forwardati da capo cluster in capo cluster fino al destinatario.

2.2 Radio Cognitive

Le reti wireless odierne sono regolate da una politica di assegnazione fissata dello spettro, ad esempio lo spettro è regolato da agenzie governative ed è assegnato a a possessori di licenze o servizi con contratti a lungo termine per vaste aree geografiche. Inoltre l'uso dello spettro è concentrato su alcune sue porzioni mentre un ammontare significativo di spettro rimane inutilizzato. Secondo la Federal Communications Commission (FCC) [6], l'utilizzo temporale e geografico dello spettro assegnato ha una variazione che oscilla

tra il 15% e il 85%. Anche se la politica di assegnazione fissata dello spettro non ha fornito problemi in passato, negli ultimi anni c'è stato un drastico incremento nell'accesso allo spettro da parte di servizi mobili. L'innovazione tecnologica consiste quindi nel creare sistemi radio capaci di usare lo spettro più intensivamente e più efficacemente che in passato. Da qui l'idea delle tecnologie radio cognitive, in quanto questi device supportano la capacità di determinare la propria posizione, percepire l'uso dello spettro da parte dei dispositivi vicini, cambiare frequenza, adattare la propria potenza in uscita e modificare i parametri e le caratteristiche di trasmissione. Queste capacità permettono di modellare delle reti con un approccio dinamico per l'accesso allo spettro e in grado di fornire un'alta bandwidth agli utenti; tutto questo grazie all'utilizzo di architetture wireless eterogenee fornite dalle tecnologie radio cognitive. Le reti di questo tipo vengono anche chiamate Dynamic Spectrum Access Networks (DSA) o NeXt Generation (xG) Networks [1]. Oggi queste tecnologie sono implementate in applicazioni come wireless LAN e servizi di rete wireless mobile, nonché sono un forte argomento di ricerca nel settore delle reti informatiche, reti di sensori e disaster recovery.

2.2.1 Caratteristiche delle Radio Cognitive

Facendo riferimento a [1], [6] e [13], le principali caratteristiche e funzioni che identificano una CR sono:

- **Frequency Agility**

Un dispositivo CR può cambiare la propria frequenza di lavoro scegliendone una più appropriata, evitando di utilizzare lo stesso canale in uso da un altro sistema radio. Il meccanismo di selezione viene chiamato *selezione dinamica della frequenza* (DFS).

- **Adaptive Modulation**

Con la *modulazione adattiva* i dispositivi CR possono modificare le proprietà della trasmissione e le sue forme d'onda per avere una migliore possibilità di accesso allo spettro e intensificarne l'uso quando l'ambiente è già pieno di altri segnali e trasmissioni.

- **Transmit Power Control**

Per *controllo di potenza in trasmissione* si intende la capacità di un dispositivo di cambiare dinamicamente i livelli di potenza trasmessa durante il processo di trasmissione dati. Questo può essere utile in situazioni come: presenza di altri dispositivi nelle vicinanze, regolamentazioni locali e/o nazionali o per adattare la potenza al minimo necessario al fine di stabilire un link di comunicazione affidabile.

- **Geographic Location Monitoring**

Una radio cognitiva può supportare la capacità di determinare la sua posizione e quella di altri ricetrasmittitori, e in seguito modificare alcuni parametri quali la potenza e la frequenza in base ai valori ammessi in quella posizione. Questo può essere fatto con tecniche di geolocalizzazione come il GPS ad esempio. Questa capacità può essere interessante anche per evitare le interferenze, in quanto non è possibile avere una misurazione affidabile della posizione con le tecnologie di sensing (rilevazioni tramite sensori).

- **Spectrum Sharing**

Il meccanismo di condivisione dello spettro delle radio cognitive permetterebbe la coesistenza di più utenti, come ad esempio tra utenti primari, che hanno il diritto di utilizzare le frequenze in licenza, e utenti secondari, che vogliono sfruttare i buchi nello spettro nel dominio sia del tempo che delle frequenze.

- **Spectrum Sensing**

La caratteristica probabilmente più importante delle radio cognitive è quella dello spectrum sensing che permette di conoscere varie informazioni su l'utilizzo dello spettro e l'esistenza di utenti primari in una certa area geografica. Infatti oltre alle informazioni riguardanti il dominio temporale, spettrale e spaziale, l'analisi dello spettro include anche la modulazione, la forma d'onda, la banda, la frequenza portante di tutte le trasmissioni che la radio riesce a captare, in modo da avere un chiaro panorama dell'ambiente circostante.

- **Spectrum Management**

Per gestione dello spettro si intende la capacità da parte dei dispositivi CR di catturare la miglior porzione di spettro disponibile, evitando interferenze pericolose con altri utenti. Le funzioni di gestione dello spettro sono classificate in *funzioni di analisi dello spettro* e *funzioni di decisione dello spettro*. L'implementazione pratica delle funzioni di Spectrum Management è una questione delicata e complessa in quanto coinvolge una varietà di requisiti sia tecnici che legali. Un requisito tecnico è ad esempio la scelta di un sensing threshold appropriato per rilevare altri utenti, mentre un esempio di requisito legale è la necessità di rispondere a leggi e norme disposte per l'accesso allo spettro radio nella legislazione nazionale ed internazionale.

- **Spectrum Mobility**

La *spectrum mobility* è la capacità di mantenere attiva una comunicazione senza fili durante il passaggio ad un migliore spettro. Ad esempio utenti secondari possono compiere un passaggio di spettro trasferendo una comunicazione in corso su un canale libero.

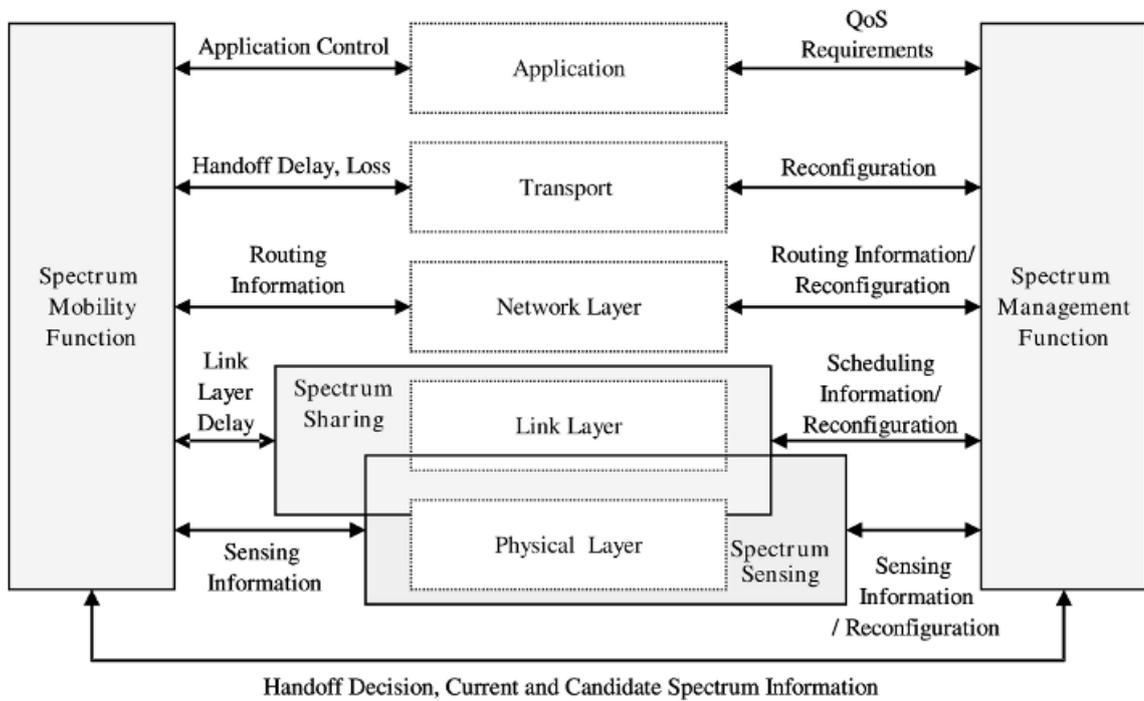


Figura 2.1: Rappresentazione di un nodo Cognitive Radio

- **Intelligence**

L'intelligenza è la caratteristica che distingue le radio cognitive dalle altre tipologie di radio. Infatti si vuole cercare di fornire un cervello alle radio, in modo che possano reagire autonomamente a certe situazioni invece di svolgere semplicemente delle routine software preimpostate. Come i cervelli biologici infatti, l'intelligenza delle radio cognitive richiede che il loro cervello sia dotato di memoria e che sia in grado di apprendere.

Queste caratteristiche abilitano protocolli di comunicazione *spectrum-aware*. La Figura 2.1 mostra una rappresentazione dei nodi cognitivi e delle loro interazioni. È evidente dal significativo numero di interazioni che le funzionalità delle reti cognitive necessitano un approccio di progettazione cross-layer. Nello specifico, le operazioni di spectrum sensing e spectrum sharing cooperano insieme per migliorare l'efficienza dello spettro. Nelle funzioni di spectrum management e spectrum mobility le funzionalità dei livelli fisico, MAC, routing, trasporto e applicazione sono gestite in un modo cooperativo, considerata la natura dinamica dello spettro sottostante. Nella Figura 2.2 viene rappresentata la sequenza di operazioni effettuate dalle radio CR durante il loro *ciclo cognitivo*, in cui i dispositivi determinano i parametri di comunicazione appropriati all'ambiente radio sottostante adattandoli ad esso. Gli step del ciclo cognitivo sono i seguenti:

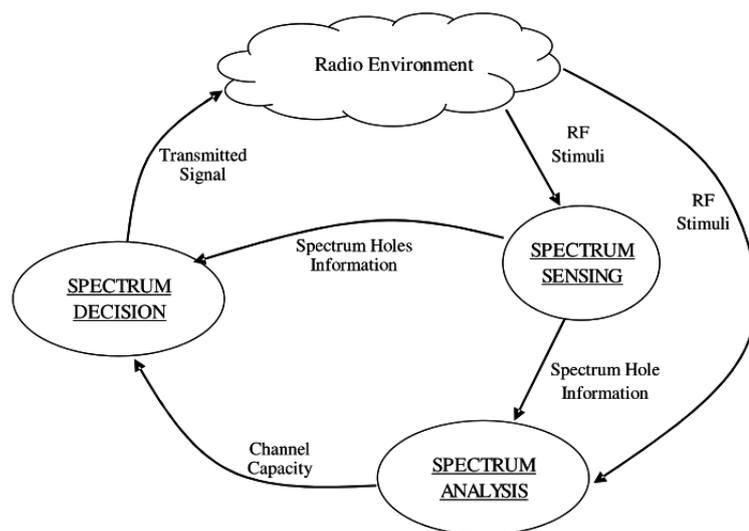


Figura 2.2: Rappresentazione di un ciclo cognitivo

1. Spectrum sensing: una radio cognitiva monitora le bande di spettro disponibili, cattura le loro informazioni e rileva i buchi nello spettro.
2. Spectrum analysis: vengono valutate le caratteristiche dei buchi nello spettro rilevate attraverso l'operazione di spectrum sensing.
3. Spectrum decision: una radio cognitiva determina il data rate, la modalità di trasmissione e la bandwidth della trasmissione. Poi sceglie la banda appropriata secondo le caratteristiche dello spettro e i requisiti dell'utente.

Una volta che la banda viene determinata, la comunicazione può essere attuata su di essa.

2.2.2 Radio Cognitive nel Disaster Recovery

Come già osservato in precedenza, in scenari post-calamità si possono verificare vari problemi nell'utilizzo delle reti wireless per operazioni di disaster recovery. Ad esempio le comunicazioni wireless cessano di funzionare in queste situazioni perché le infrastrutture possono venire danneggiate, o possono crearsi delle congestioni causate dall'overhead di richieste, o per condizioni di comunicazione estreme (ad esempio interferenza atmosferica causata dalle condizioni meteo durante disastri naturali). Per instaurare comunicazioni sotto queste condizioni, sono necessari dispositivi radio riconfigurabili e intelligenti, i quali dovrebbero comprendere l'ambiente radio e il suo uso e di conseguenza adattare i propri parametri autonomamente. Un altro problema importante è l'interoperabilità

tra i vari attori in gioco in scenari post-calamità, come le organizzazioni di pubblica sicurezza, le organizzazioni umanitarie e le associazioni di primo intervento, il cui numero è incrementato negli anni con il crescere della popolazione, al contrario delle risorse tecnologiche che non sono cambiate molto nel tempo. Quindi si rivela molto difficile offrire linee di emergenza per far comunicare questi attori fra di loro, a causa della scarsità di risorse come le frequenze nello spettro radio o infrastrutture comuni in grado di coprire tutte le organizzazioni e associazioni di una certa regione. Le radio cognitive vengono quindi ampiamente studiate come alternativa emergente per le comunicazioni di sicurezza pubblica sotto l'aspetto dell'uso opportunistico dello spettro radio e l'interoperabilità della rete [8].

Capitolo 3

STEM-Net

Dall'osservazione delle varie problematiche riscontrate nelle operazioni di disaster recovery discusse nella Sezione 1.4, nasce il progetto STEM-Net [2] che si basa principalmente sull'idea di creare una rete di dispositivi *auto-configuranti*, *auto-regolarizzanti* e *auto-organizzanti*. Da qui viene il nome *STEM-Net*, dall'inglese *Rete-Staminale*, in quanto possiamo immaginare che un device della STEM-Net, si comporti similmente ad una cellula staminale biologica: come una cellula staminale ha la capacità di trasformarsi in diversi altri tipi di cellula del corpo, un dispositivo staminale della STEM-Net (chiamato *STEM-Node*) è capace di riconfigurarsi a più livelli di protocollo, in base alla situazione, al contesto, ai servizi richiesti, alle informazioni precedentemente apprese e alle interazioni con gli altri dispositivi analoghi. L'intera organizzazione della rete viene quindi gestita a livello di una singola unità staminale che viene definita dalle seguenti capacità e caratteristiche:

3.1 Riconfigurazione Protocollare (intra-stack)

- L'architettura protocollare dei dispositivi permette di includere nuove funzionalità, di fungere da repository delle informazioni apprese durante il ciclo di vita del dispositivo staminale, di consentire lo scambio di informazioni tra diversi livelli in approccio cross-layer e, ancora, di consentire l'esistenza di funzioni di controllo trasversali allo stack.
- I parametri del livello fisico e livello MAC dei dispositivi possono essere modificati in base alle caratteristiche dell'ambiente wireless in cui opera il dispositivo e alle caratteristiche delle infrastrutture in esso presenti.

- I dispositivi supportano delle tecnologie abilitanti per la riconfigurabilità dello stack protocollare (fisico e MAC) che permettano di attivare on-the-fly estensioni/varianti del protocollo MAC; queste sono basate sui paradigmi di *Dynanamic Spectrum Access* (DSA) e *Cognitive Radio* (CR) (Sezione 2.2).
- I dispositivi supportano un insieme di metriche per valutare l'impatto che ha la mobilità dei nodi su ognuno dei livelli dello stack protocollare.

3.2 Multimodalità (riconfigurabilità protocollare inter-stack)

- Per i livelli rete e trasporto i dispositivi permettono l'attivazione di funzionalità di routing con switching tra protocolli diversi, di funzionalità di bridging tra tecnologie differenti (ad esempio Wi-Fi e UMTS), l'attivazione di servizi come DHCP, DNS, Proxy HTTP, Proxy SIP in modalità on-the-fly ed infine permette la disattivazione di interfacce.
- I dispositivi supportano la riconfigurabilità protocollare anche per i livelli alti dello stack (sessione/applicazione).
- Ogni dispositivo può assumere un ruolo diverso in modo da soddisfare un preciso criterio di ottimizzazione.
- I dispositivi supportano l'aggiornamento software/firmware per implementare funzioni/protocolli inizialmente non previsti.

3.3 Apprendimento

Per *apprendimento* si intende la capacità dei dispositivi di prendere nuove decisioni in base ai dati precedentemente raccolti sull'ambiente e in base alle configurazioni passate (storia). Ad esempio questi dispositivi possono modificare i parametri trasmissivi del livello fisico e MAC, apprendendo dinamicamente la loro configurazione ottimale; alternativamente si possono modificare certi specifici parametri in base ai precedenti dati sul consumo energetico, banda trasmissiva e ritardi di propagazione, al fine di ottimizzare l'utilizzo delle risorse.

3.4 Cooperazione/Coordinazione

La cooperazione e coordinazione della rete STEM-Net avviene in modo decentralizzato e automatico in quanto ogni nodo è *intelligente* e capace di auto-organizzarsi in base alle informazioni e alle configurazioni degli altri nodi. È stato effettuato un ampio studio delle tecniche che possono essere utilizzate in questo ambito. Ad esempio si sono studiate soluzioni di supporto al multihoming concorrente per incrementare il potenziale della comunicazione, meccanismi di coordinamento e scambio di informazioni tra i nodi per ottenere un certo comportamento globale, tecniche di coordinazione che permettano di implementare uno scheduling distribuito tra i nodi, ed altre soluzioni che riguardano molte delle caratteristiche dei dispositivi elencate precedentemente.

3.5 Scenari d'uso

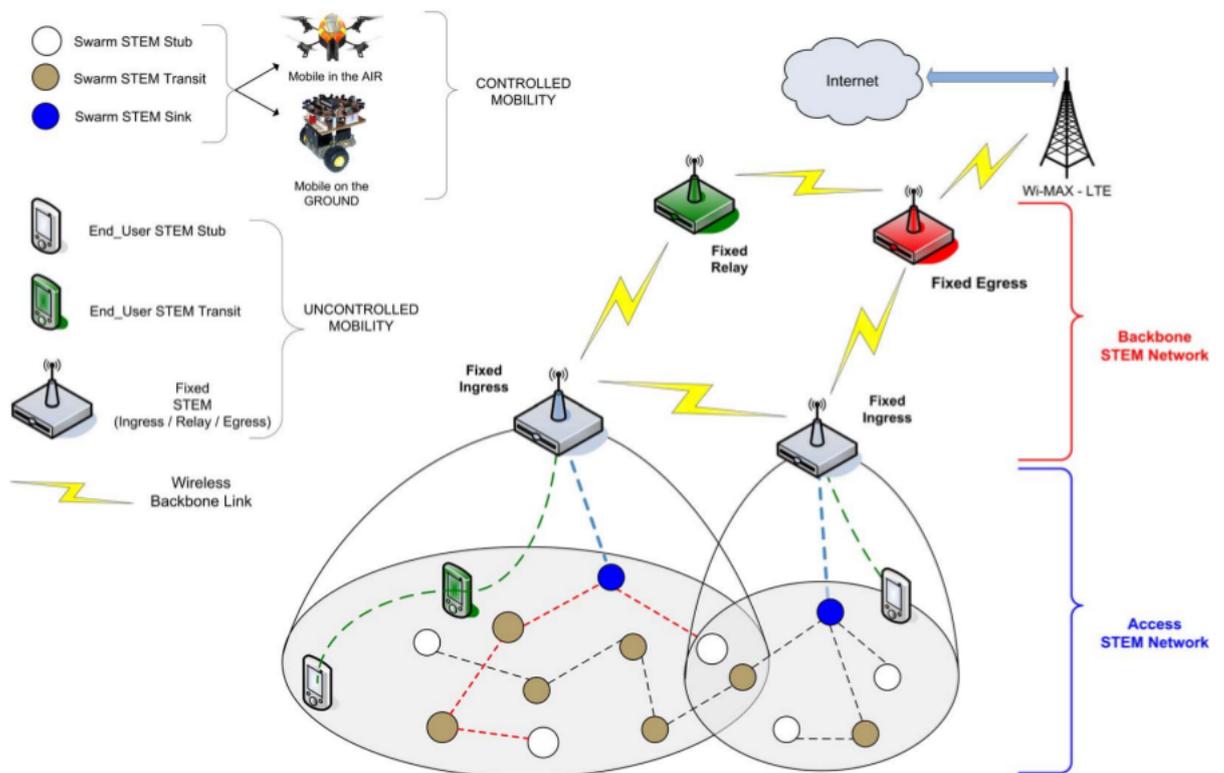


Figura 3.1: Architettura generica di una STEM-Net

Un'infrastruttura generica della STEM-Net è illustrata nella Figura 3.1. L'architettura si divide in due livelli: *Access STEM Network* (ASN) e *Backbone STEM Network* (BSN). In sostanza, i dispositivi appartenenti alla BSN sono basati su tecnologie wireless solide come UMTS, WiMAX, etc. Questi dispositivi possono assumere ruoli diversi che si distinguono in tre tipi: Ingress, Relay e Egress. I nodi Ingress offrono connettività ai nodi della ASN, i nodi Relay partecipano al forwarding dei dati ai/dai nodi Ingress e Egress; i nodi Egress tipicamente agiscono da gateway verso Internet. Nel livello ASN sono inclusi sia i nodi con mobilità controllata (Swarm STEM Nodes, SSN) che i nodi con mobilità non controllata (End-User STEM Nodes, EUSN). Ogni SSN possiede la capacità di muoversi e coordinarsi con altri SSN in modo da creare una comunità impegnata ad una determinata operazione. Un SSN può assumere il ruolo di Swarm STEM Transit (SST) e fare da tramite dei dati tra gli altri nodi oppure può assumere il ruolo di Swarm STEM Sink (SSS) e interfacciarsi alla BSN. I nodi EUSN sono dispositivi wireless appartenenti agli utenti (come smartphone ad esempio) la cui mobilità non è controllabile essendo dipendente dall'utente. Come gli SSN, anche gli EUSN possono fungere da nodi di transito ed estendere l'accesso alla rete oltre il single-hop. Una possibile applicazione di quest'architettura può attuarsi nel disaster recovery.

3.5.1 STEM-Net nel Disaster Recovery

Disastri ambientali potrebbero danneggiare i dispositivi o le infrastrutture di rete e impedire la connettività urbana. I dispositivi dei sopravvissuti (telefoni cellulari ad esempio) potrebbero venir isolati a causa dell'eterogeneità dei protocolli di copertura reciproca di questi dispositivi o a causa dell'assenza delle infrastrutture di rete. Come prima reazione i dispositivi dei sopravvissuti potrebbero creare una comunicazione in modalità infrastructure trasmettendo informazioni di base, aiutare la diagnosi o rilevare situazioni pericolose. Ogni SSN o EUSN può auto-riconfigurarsi secondo le proprie funzionalità e ruoli in modo da instaurare delle connessioni con i nodi vicini e definire un percorso di routing rivolto verso un access point. Potrebbero comunque verificarsi dei casi in cui qualche dispositivo, supportante le capacità di connettività e mobilità controllata, sia mandato nell'area disastrosa con il compito di ripristinare la connettività tra i segmenti della rete, creati dagli SSN o EUSN in maniera autonoma e auto-organizzata. Questo viene rappresentato in Figura 3.2 dove i nodi etichettati come recovery devices rappresentano nodi autonomi con differenti ruoli e posizioni specifiche.

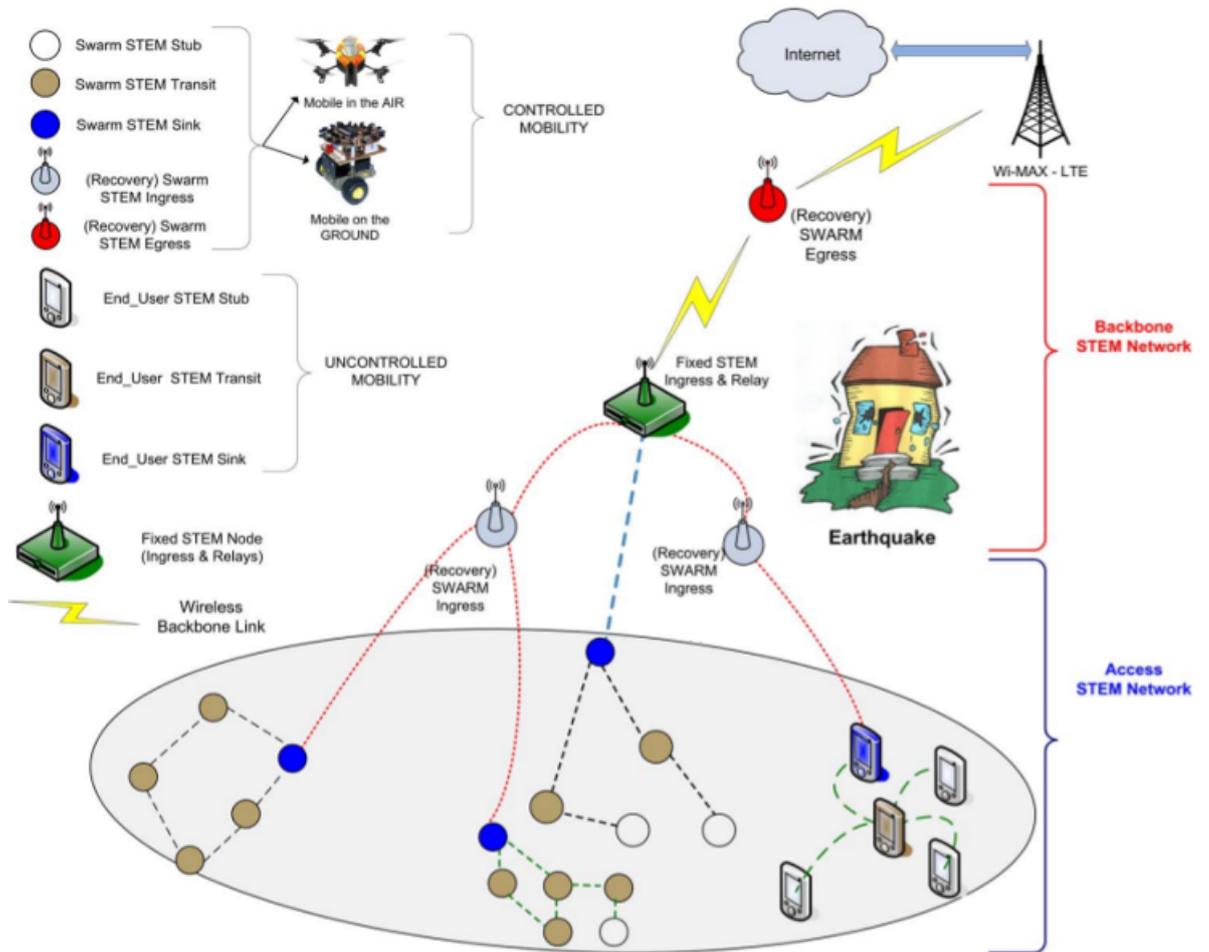


Figura 3.2: STEM-Net per il disaster recovery

Capitolo 4

STEM-Mesh

Nel progetto STEM-Mesh [5] si affronta il problema di ristabilire la connettività di rete in scenari post-calamità naturale attraverso l'utilizzo di tecnologie Cognitive Radio (CR), mobilità controllata e principi di reti auto-organizzanti. Si considera nello specifico uno scenario post-calamità, in cui la rete wireless è partizionata in diversi frammenti disconnessi tra loro chiamate isole. Ogni isola può utilizzare una tecnologia specifica (Wi-Fi, Wi-MAX, etc.) e uno specifico range di frequenze per le loro operazioni. Per garantire la comunicazione tra le diverse isole è stato proposto l'utilizzo di dispositivi wireless chiamati *STEM-Node* (SN) dal progetto STEM-Net presentato nel Capitolo 3. Come le cellule staminali sono oggetto di studio da parte dei ricercatori per curare determinate malattie, in STEM-Mesh si sono presi in considerazione i *dispositivi staminali* del progetto STEM-Net per studiare delle "cure" applicabili negli scenari post-calamità. Essendo quindi gli STEM-Node dei dispositivi wireless altamente auto-riconfiguranti ed auto-organizzanti, possono diventare dispositivi sostituitivi ai nodi compromessi nella rete, riconfigurandosi appunto in base alle funzionalità interrotte. Ogni SN supporta tre livelli di riconfigurabilità:

1. Riconfigurabilità dello spettro

Ognuno dei dispositivi SN è equipaggiato di più piattaforme Software Defined Radio (SDR), attraverso le quali può comprendere la struttura radio dell'ambiente, operando a diverse frequenze e cercando così di scoprire le varie isole di connessione. Per Software Defined Radio si intendono ricevitori radio non più hardware ma quasi completamente software che, a differenza degli altri, funzionano con più tipi di standard (ad esempio UMTS, GSM, Wi-Fi, Wi-Max) in quanto sono in grado di modificare via software dei parametri di livello fisico come per esempio frequenza di lavoro e modulazione.

2. Riconfigurabilità di rete

Un SN può fare routing del traffico tra le isole e allo stesso tempo comunicare con altri SN.

3. Riconfigurabilità di movimento

Ogni SN può controllare il proprio movimento e spostarsi in posizioni specifiche per supportare il ritrovamento delle isole.

Inoltre, insiemi di SN possono coordinarsi in modo da formare una maglia virtuale di nodi staminali (in inglese appunto STEM-Mesh), che rimpiazza le funzionalità delle porzioni di rete compromesse e fornisce una comunicazione wireless multi-hop tra le isole. Queste isole hanno una cardinalità n_i , ovvero il numero di dispositivi connessi all'isola, e una frequenza h_i sulla quale opera l'isola. Inoltre tutti i dispositivi di un'isola i sono collegati ad un gateway G_i che compie il routing all'interno dell'isola. Per favorire il ritrovamento dell'isola, ogni gateway manda in broadcast messaggi *HELLO* periodici sulla frequenza h_i , con all'interno anche n_i . Si può pensare che i dispositivi STEM-Node vengano immessi nello scenario da organizzazioni di pubblica sicurezza per ristabilire la connettività tra i gateway delle isole.

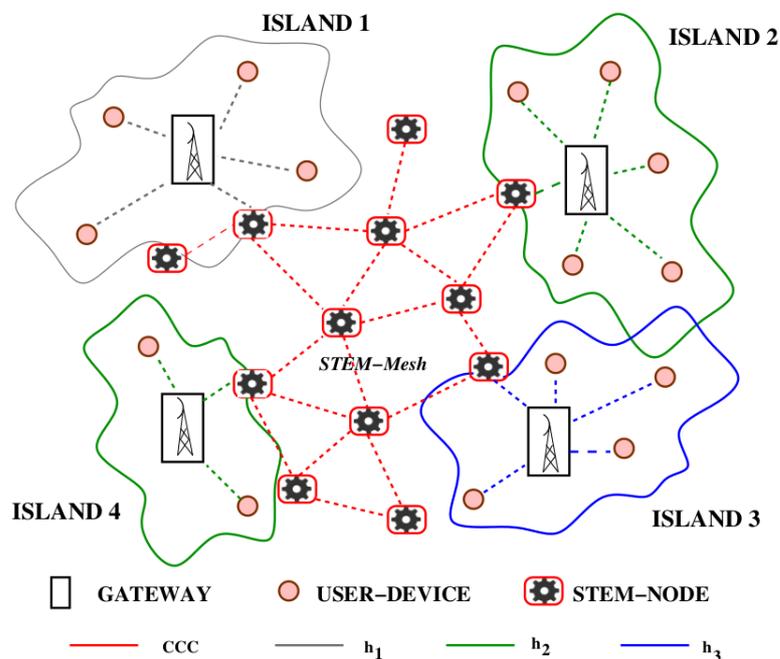


Figura 4.1: Struttura della maglia STEM-Mesh

4.1 Architettura degli STEM-Node

L'architettura di un SN è raffigurata nella Figura 4.2. Ogni SN è equipaggiato con $T > 1$ piattaforme Software Defined Radio (SDR). Si assume che ogni radio può supportare fino ad H frequenze operative, dove ogni frequenza è mappata su specifiche tecnologie wireless e stack di rete (PHY/MAC/Net) da utilizzare. Una radio di controllo è usata per comunicare con altri SN e instradare i dati su un canale predefinito (Common Control Channel), attraverso la backbone della STEM-Mesh mentre le rimanenti $T - 1$ interfacce sono usate per connettere i gateway. Ogni radio può essere nello stato *free* o *bridged* (ad esempio usata per comunicare con un gateway o meno). Inoltre ogni SN è equipaggiato con un dispositivo GPS per l'auto-localizzazione e può muoversi autonomamente all'interno dello scenario (attraverso delle ruote). A periodici intervalli, ogni SN fa rilevazioni sull'ambiente per scoprire la presenza di gateway tramite un modulo di discovery e coordina gli altri SN attraverso un modulo di cooperazione. Le informazioni riguardanti l'ambiente sono mantenute in un Database. Basandosi su queste informazioni, il modulo del *Brain Control* decide la prossima direzione verso cui muoversi (secondo l'algoritmo descritto nella Sezione 4.2) e la frequenza da utilizzare. Nel caso di ritrovamento delle isole, l'SN riconfigura una delle sue SDR disponibili, in modo operare alla frequenza usata dal gateway.

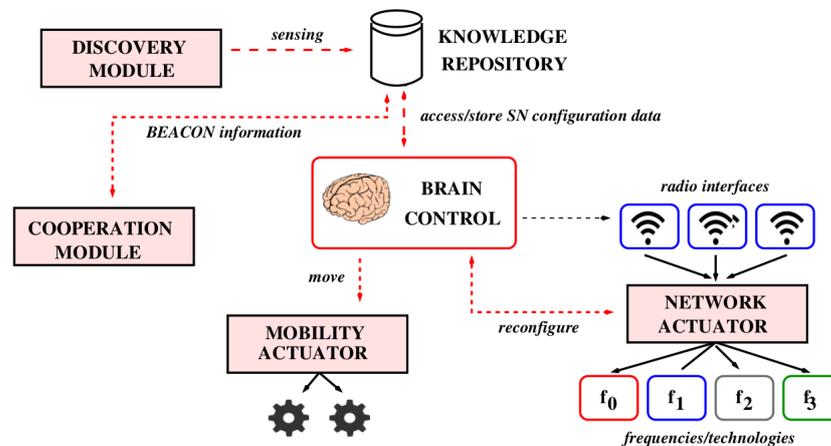


Figura 4.2: Architettura di uno STEM-Node

4.2 Algoritmo di Controllo Distribuito

Il progetto fornisce un algoritmo di controllo distribuito (basato sull'approccio di forze elastiche virtuali [10], [4]) attraverso il quale gli SN possono attuare l'esplorazione

sia in termini di frequenza che in termini di spazio e auto-organizzarsi in una struttura di tipo STEM-Mesh. L'algoritmo proposto utilizza la stima del link budget (bilancio di collegamento) per garantire una comunicazione robusta tra i membri della maglia anche in presenza di condizioni di propagazione difficili. I meccanismi di esplorazione e bridging delle isole sono anch'essi modellati come insieme di forze elastiche virtuali che agiscono su ogni SN. Ne risulta che la maglia si adatta dinamicamente all'ambiente e cerca di ricostruire la massima connettività tra le isole.

Algorithm 1 Controllo di movimento di uno STEM-Node SN_i

Mandare in broadcast messaggi BEACON strutturato come in (4.1)

Calcolare ogni forza elastica \vec{F}_{ij} secondo l'Equazione (4.2).

Determinare la forza risultante \vec{R}_i agente sull' SN_i .

Decidere la probabilità di movimento p_i secondo l'Equazione (4.6).

if $Random < p_i$ **then**

Muoversi secondo il vettore \vec{R}_i .

end if

Decidere la frequenza h da utilizzare secondo l'Equazione (4.8).

if l'isola j trovata sulla frequenza h e un'interfaccia radio è *free* **then**

Cambiare lo stato dell'interfaccia in *bridged*.

Riconfigurare l'interfaccia radio per operare sulla frequenza h .

end if

Ad intervalli regolari, ogni SN (SN_i) manda in broadcast messaggi BEACON con le seguenti informazioni:

$$\langle gps, criticità, g_{list}, frequenza_h \rangle \quad (4.1)$$

dove gps sono le coordinate GPS del nodo, $criticità$ è un parametro per la stabilità della STEM-Mesh, g_{list} rappresenta la lista di gateway al momento connessi alla STEM-Mesh e $frequenza_h$ è appunto la frequenza utilizzata dall'SN durante l'intervallo di tempo trascorso dall'ultimo messaggio BEACON mandato. L'algoritmo estende lo schema della Virtual Spring Mesh [10], [4], e lo adatta allo scenario di disaster recovery descritto in precedenza. Come le molle meccaniche, le molle virtuali hanno una lunghezza l_0 e una costante di durezza k . Ogni forza elastica virtuale \vec{F}_i^j applicata all' SN_i è modellata secondo la ben nota legge di Hooke:

$$\vec{F}_i^j = -k_j^{ST} \cdot \delta \cdot \vec{u} \quad (4.2)$$

Qui \vec{u} è il vettore unitario tra SN_i e gli altri endpoint della molla, δ è l'allungamento (ad esempio la differenza tra l'attuale lunghezza della molla e la sua lunghezza naturale l_0) e k_j^{ST} è la durezza della molla che assume valori differenti ma prefissati, in base al tipo di forze che si considerano. Senza entrare troppo nel dettaglio STEM-Mesh propone tre tipi di forze elastiche (definite come tipo di molla):

- Molla *Mesh-to-Mesh*. Gli estremi della molla sono uno STEM-Node SN_i e un qualsiasi altro vicino SN_j , così che la connettività è preservata tra i membri della STEM-Mesh, garantendo un livello di qualità di link minimo (attraverso il Link-Budget).
- Molla *Mesh-to-Island*. Gli estremi della molla sono SN_i ed un gateway di un isola j , così che la STEM-Mesh può connettersi alle isole.
- Molla *Mesh-to-Frontier*. Gli estremi della molla sono SN_i e un punto della frontiera della STEM-Mesh. La forza viene introdotta per abilitare l'esplorazione in spazio tramite nodi speciali della maglia chiamati *scout*.

Nel caso di molle *Mesh-to-Island*, il parametro k^{ST} è proporzionale alla cardinalità dell'isola j . Da questo ne risulta che gli SN saranno più attratti verso gateway di isole con un numero maggiore di nodi connessi, così da massimizzare il numero di nodi connessi in tutta la STEM-Mesh. Il parametro δ viene formulato in base alla qualità del collegamento della comunicazione tra due SN o tra un SN ed un gateway di un isola. A questo proposito un SN, dopo aver ricevuto un messaggio BEACON o un messaggio HELLO, calcola il Link Budget del collegamento $i \leftrightarrow j$ ($LB(i, j)$):

$$LB(i, j) = Pr^i - RS_{thr}^i \quad (4.3)$$

dove Pr^i è la potenza ricevuta da SN_i e RS_{thr}^i è la sua sensitività di ricezione.

Da qui δ viene definito in termini di distanza tra dal valore corrente $LB(i, j)$ e LB_{req} (Link Budget minimo richiesto nella STEM-Mesh):

$$\delta = \sqrt{\frac{\max(LB(i, j), LB_{req})}{\min(LB(i, j), LB_{req})}} - 1 \quad (4.4)$$

Ad intervalli regolari, ogni SN_i calcola le forze elastiche virtuali $\vec{F}_i^0, \vec{F}_i^1, \dots, \vec{F}_i^n$ che agiscono su di esso e determina la forza risultante $\vec{R}_i = \sum_{k=0}^n \vec{F}_i^k$.

L'algoritmo fornisce anche un meccanismo per evitare la perdita di connettività tra gli STEM-Node o tra uno STEM-Node ed un gateway: dopo aver determinato $|\vec{R}_i|$ (intensità della forza), ogni nodo SN_i applica un filtro probabilistico per decidere se modificare la sua posizione o meno. La probabilità di movimento p_i di uno STEM-Node viene calcolata sia in funzione dello stimolo che in funzione del rischio di rottura di

collegamenti già esistenti con altri STEM-Node o gateway. Quest'ultimo fattore viene definito in base alla criticità dei collegamenti (ad esempio $i \leftrightarrow j$):

$$C(i, j) = 1 - \frac{\min(LB(i, j), LB_{req})}{LB_{req}} \quad (4.5)$$

Il valore dell'Equazione (4.5) è nell'intervallo $[0, 1]$, quindi un valore prossimo ad 1 indica un alta probabilità di rottura del collegamento e dovrebbe essere interpretata dall' SN_i come un avvertimento di non modificare la sua posizione rispetto a G_j o SN_j . La *criticità del nodo* (per SN_i si denota $C(i)$) è poi definita come il massimo delle criticità di tutti i collegamenti che coinvolgono il nodo in considerazione. Quindi se $C(i) = 1$, si imposta automaticamente la probabilità di movimento p_i a 0 (l'SN non si muoverà) in quanto è indice del fatto che qualche collegamento quasi sicuramente si romperebbe in caso l'SN facesse una mossa. Altrimenti p_i viene calcolata come segue:

$$p_i = |\vec{R}_j^{norm}|^\gamma \quad (4.6)$$

Qui $|\vec{R}_j^{norm}|$ è l'intensità della forza risultante, normalizzata tra $[0,1]$, e l'esponente γ è definito in funzione della criticità del nodo $C(i)$:

$$\gamma = \frac{A}{1 - C(i)} \quad (4.7)$$

dove A è un parametro che governa l'intensità dell'attenuazione della forza. Con varie simulazioni si è visto che p_i incrementa con l'intensità dello stimolo ($|\vec{R}_j^{norm}|$) e per gli stessi valori di $|\vec{R}_j^{norm}|$ la probabilità è attenuata con l'incrementare di $C(i)$.

Questo fornisce l'idea che un SN dovrebbe evitare di modificare la sua posizione nel caso in cui un collegamento wireless stia per rompersi. Allo stesso tempo, la stabilizzazione della maglia è favorita dal fatto che gli SN reagiranno improbabilmente a piccoli stimoli. Muovendosi, ogni SN_i utilizza le sue interfacce radio libere per fare sensing sulle frequenze H , al fine di rilevare messaggi HELLO dai gateway delle isole. Per bilanciare lo sforzo di sensing tra gli SN e per favorire il ritrovamento delle isole, ogni SN_i mantiene delle statistiche sul numero di azioni di sensing compiute su ogni frequenza h , considerando anche le azioni di sensing dei nodi vicini (questa informazione viene fornita dal parametro *frequenza_h* del messaggio BEACON). Sia $SE^i(h)$ la statistica attuale per la frequenza h dell' SN_i . Ogni volta che viene inviato un BEACON, l' SN_i per le azioni di sensing sceglie di utilizzare la frequenza h' che risulta essere il canale meno utilizzato (e più inesplorato) secondo la sua storia (statistiche sulle frequenze):

$$h' = \operatorname{argmin}_{\forall h \in H} SE^i(h) \quad (4.8)$$

4.3 Prototipo dell'unità STEM-Node

Il progetto STEM-Mesh è ancora in fase sperimentale. A seguito dello studio teorico sono state testate le performance e le reazioni del sistema, tramite simulazioni software su scenari 2-D generati casualmente, utilizzando il tool Omnet++. Non è stata ancora effettuata però una fase di test pratici che portasse in evidenza successi e problematiche nell'applicazione reale. Nasce quindi questo mio lavoro che mira a realizzare, in collaborazione con il gruppo di ricerca del progetto STEM-Mesh, un primo prototipo parziale dell'unità STEM-Node.

Parte II

Progettazione e Implementazione

Capitolo 5

Piano di Processo

A seguito di un ampio studio sugli scenari di Disaster Recovery nelle reti wireless e sulle varie tecniche che possono essere utilizzate in questo contesto, si vuole progettare ed implementare un dispositivo che sia utile nelle applicazioni reali di questi scenari. In particolare si vuole far riferimento al progetto STEM-Mesh trattato nel Capitolo 4 e nello specifico ai dispositivi staminali STEM-Node (nodi staminali). Di seguito verranno descritte le fasi progettuali e nel capitolo successivo quelle implementative di un primo modello dello STEM-Node.

5.1 Scenario di Progetto

Nel progetto si considera uno scenario semplificato rispetto a quello esposto in STEM-Mesh. In riferimento ai tre livelli di riconfigurabilità descritti nel Capitolo 4, in questo progetto ci si focalizza esclusivamente sullo studio del terzo livello di riconfigurabilità: quella di movimento. Inoltre ci si vuole ridurre al caso di un singolo dispositivo Stem-Node (SN) in modo da poterne studiare nello specifico le possibilità implementative. L'idea è quella di posizionare l'SN tra altri due nodi Wi-Fi simulatori che si comporterebbero come delle isole o, tramite movimento manuale, come altri SN. Il sistema di tutti i nodi Wi-Fi (SN compreso) viene distribuito su una linea retta per facilitarne lo studio iniziale, eliminando le problematiche della determinazione della posizione geografica dei nodi (ci si focalizza sul movimento dipendente esclusivamente dal Link Budget). Una volta che lo sviluppo di un singolo SN sarà terminato e testato, si passerà ad ampliare lo scenario aggiungendo molteplici SN, apportando le opportune modifiche hardware e software, studiandone poi il comportamento.

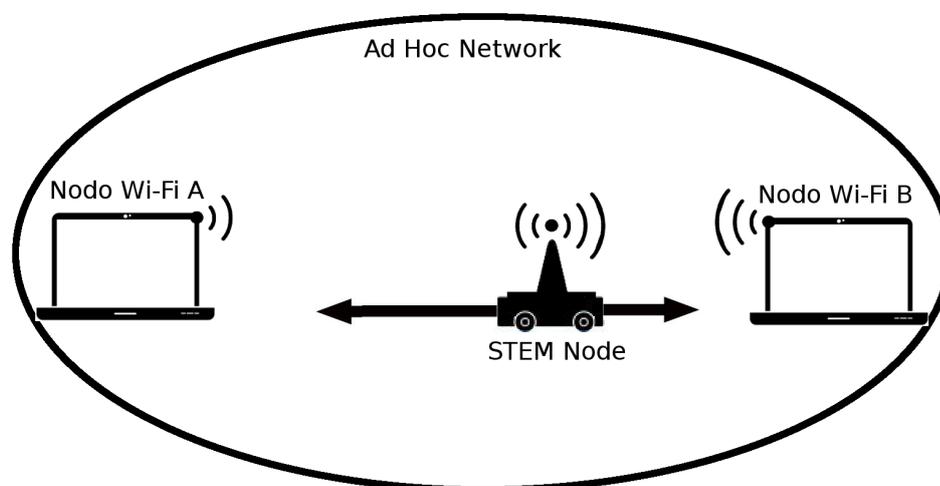


Figura 5.1: Schema dello scenario di progetto

5.2 Analisi dei Requisiti

Lo scenario di progetto richiede che:

5.2.1 Rilevamento della potenza di segnale

l'SN sia dotato di almeno un modulo radio (Wi-Fi in questo caso), tramite il quale l'SN possa percepire la potenza di segnale di altri punti radio nel suo raggio di ricezione (determinato dalla sensibilità dello stesso modulo) che sono connessi alla sua stessa rete.

5.2.2 Controllo del movimento

l'SN si possa muovere autonomamente e che quindi possa controllare uno o più dispositivi di movimento, ad esempio uno o più motori con delle ruote o delle eliche, se si pensa ad un contesto aereo.

5.2.3 Routing del Traffico

l'SN sia in grado di ricevere e trasmettere del traffico di rete, permettendo quindi la comunicazione tra i due nodi Wi-Fi estremi al sistema, anche qualora essi non siano nello stesso raggio di copertura.

5.3 Risorse e strumenti impiegati

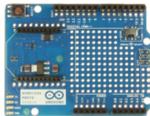
- Hardware
 - Arduino Uno: piattaforma hardware programmabile tramite seriale USB. Monta un microcontrollore ATmega328P, 14 pin I/O digitali e 6 pin I/O analogici.
<http://arduino.cc/en/Main/ArduinoBoardUno>
 - Magician Chassis ROB-10825: piattaforma robot. Monta due motori con ruote da 65mm e una rotella posteriore.
<https://www.sparkfun.com/products/10825>
 - Motor Driver 1A Dual TB6612FNG: scheda per il controllo dei motori. Può controllare fino a due motori.
<https://www.sparkfun.com/products/9457>
 - Wireless Proto Shield: scheda per connettere facilmente dei moduli wireless all'Arduino board.
<http://arduino.cc/en/Main/ArduinoWirelessProtoShield>
 - WiFly Shield RN-XV: scheda che incorpora il modulo Wi-Fi RN171. Permette di connettersi ad una rete Wi-Fi in modalità infrastructure o ad-hoc.
<http://rovingnetworks.com/products/RN171XV>
 - Due interfacce Wi-Fi su sistema GNU Linux.
- Software e Linguaggi di Programmazione
 - Wiring per Arduino: è una piattaforma di sviluppo open source composta da un linguaggio di programmazione derivato da C e C++ ed un ambiente di sviluppo integrato (Integrated Development Environment o IDE) con elementi per gestire l'hardware dell'Arduino Board.
 - Python per la parte di applicazione GNU Linux.

Capitolo 6

Progettazione



WiFly Shield RN-XV



Wireless Proto Shield



Arduino UNO



Motor Driver TB6612FNG



Magician Chassis ROB-10825

Si è scelto di utilizzare i componenti hardware elencati nella Sezione 5.3 affinché potessero venir soddisfatti i requisiti specificati nella Sezione 5.2. L'SN consiste quindi di un Arduino UNO collegato ad un'unità mobile (Magician Chassis ROB-10825) composta da due motori utilizzati per il movimento e collegato ad un modulo radio con tecnologia Wi-Fi (WiFly Shield RN-XV). I motori vengono controllati tramite un motor driver (Motor Driver 1A Dual TB6612FNG) al quale il microcontrollore trasmette in output tensione modulata in tipo PWM dai pin digitali. In questo modo si riesce a comunicare ai motori anche una certa velocità. Il modulo Wi-Fi è collegato all'Arduino tramite una Wireless Shield la quale ha il solo compito di interfacciare il modulo Wi-Fi all'Arduino, tramite una connessione seriale che permette alle due schede di comunicare.

Trascurando l'aspetto di riconfigurabilità spettrale, si è scelto di lavorare in una rete ad-hoc Wi-Fi affinché in una futura estensione dell'implementazione, tutti gli SN possano conoscere la potenza di segnale ricevuta (RSSI) dai nodi della rete all'interno del loro raggio di copertura. Questo è possibile in una rete Wi-Fi ad-hoc in quanto ogni nodo manda dei pacchetti beacon periodici in broadcast a tutta la rete, quindi ogni altro nodo della rete può calcolare la potenza di segnale ricevuta da quel nodo. La Figura 6.1 mostra un esempio di monitoraggio del traffico dei frame beacon inviati in broadcast dai nodi di una rete ad-hoc. Nell'esempio vengono utilizzate tre interfacce Wi-Fi connesse ad una rete ad-hoc.

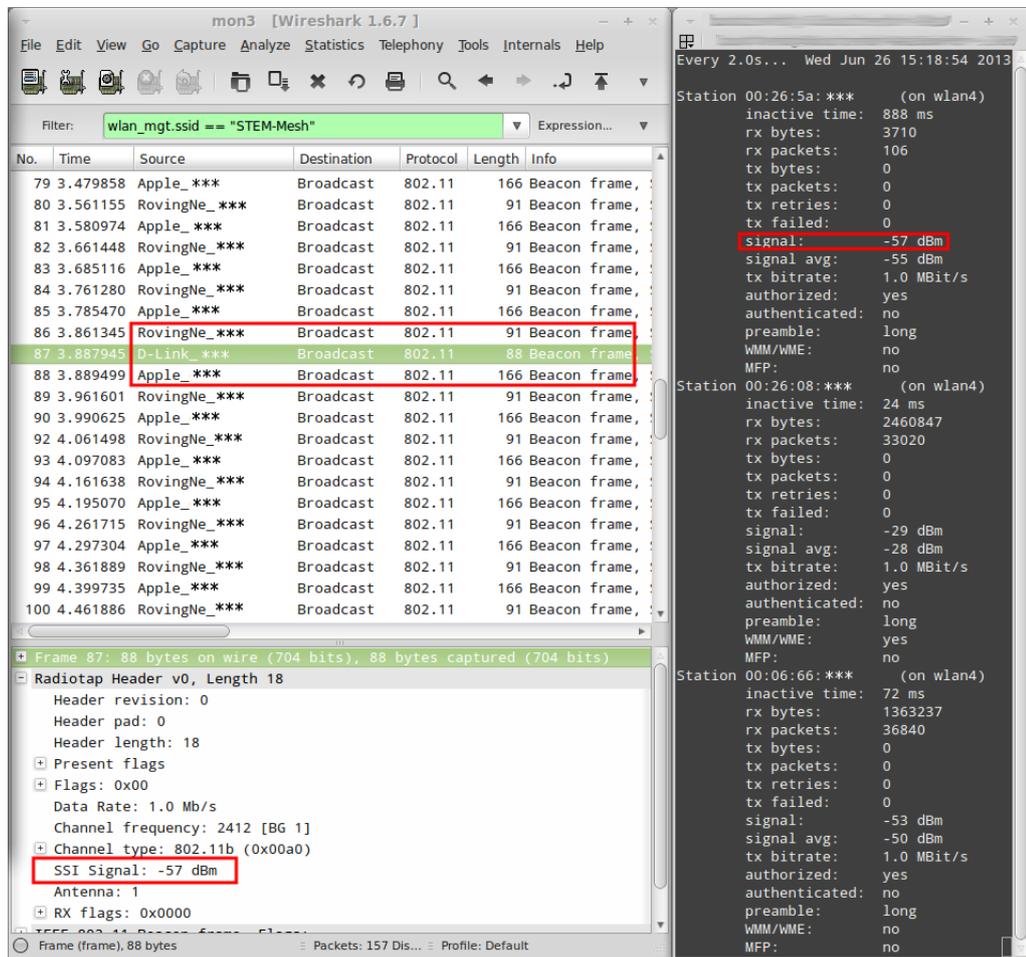


Figura 6.1: Monitoraggio del traffico di pacchetti beacon in una rete ad-hoc

Sulla sinistra compare la lista dei pacchetti beacon monitorati, in basso a sinistra l'SSI all'interno dell'header Radiotap, un meccanismo utilizzato dai driver delle interfacce per fornire informazioni aggiuntive ai frame (quindi viene calcolato dall'interfaccia stessa), e a destra il monitoraggio delle informazioni relative ai tre nodi, tra cui la potenza di segnale. Nel nostro caso però il firmware del modulo Wi-Fi scelto non supporta alcuna funzione per ottenere gli RSSI di più nodi all'interno di una rete ad-hoc, quindi per il momento gli RSSI vengono inviati da un programma che gira direttamente sui due nodi Wi-Fi simulatori (come già specificato nello scenario di progetto). Il programma invia pacchetti UDP periodici all'SN con all'interno del payload la potenza di segnale ricevuta da esso.

Per questa parte della progettazione, la soluzione appena descritta è sufficiente per permettere lo studio del movimento dell'SN e delle sue reazioni in base alle dinamiche del

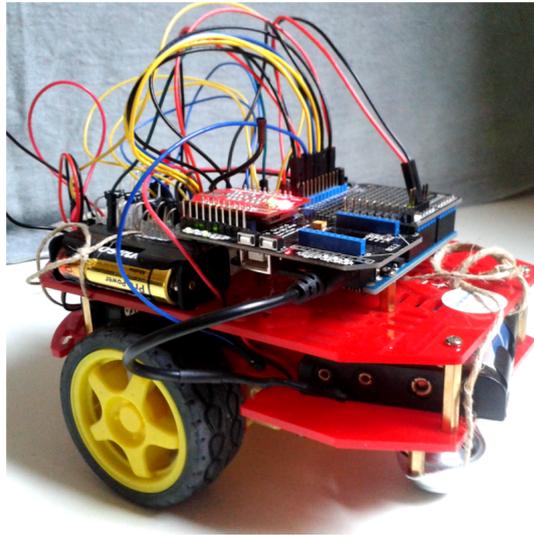


Figura 6.2: Primo prototipo del dispositivo STEM-Node

sistema. Nell'ultimo capitolo verranno descritte delle soluzioni alternative possibili per lo sviluppo di una futura implementazione.

La parte software è divisa essenzialmente nelle due parti spiegate di seguito.

6.1 Client UDP Python

Il programma scritto in python per sistemi GNU Linux, come descritto precedentemente, semplicemente invia periodicamente dei pacchetti UDP all'SN con all'interno l'RSSI ricevuto dall'SN.

Per ottenere l'RSSI, il programma utilizza l'utility iw (<http://wireless.kernel.org/en/users/Documentation/iw>). Questa utility permette di elencare, divisi per MAC address, tutti i nodi presenti nella rete ad-hoc, con le relative potenze di segnale. Quindi essendo a conoscenza del MAC address dell'SN, ogni nodo Wi-Fi simulatore può determinare il proprio RSSI correlato all'SN ed inviarglielo.

6.2 Programma Wiring per Arduino

Il programma scritto in Wiring per Arduino è la parte di software cuore del progetto. Come già accennato in precedenza, con il termine "Wiring" si intende generalmente una piattaforma di sviluppo open source composta da un linguaggio di programmazione, un ambiente di sviluppo integrato (Integrated Development Environment o IDE) ed

un circuito stampato basato su un microcontrollore. Nel nostro caso il linguaggio di programmazione è composto da un ibrido tra C e C++ creato appositamente dal team Arduino. Lo stesso team ha messo a disposizione anche un IDE apposito che facilita l'interfacciamento alla board Arduino (scrittura del codice, compilazione e flashing).

Questa parte software si basa sull'Algoritmo 1 (Controllo di movimento di uno STEM-Node SN_i), senza però considerare la parte dell'algoritmo che effettua la scelta della frequenza su cui fare sensing. Implementando questo algoritmo, l'Arduino riesce quindi a controllare il suo movimento analizzando la potenza di segnale dei nodi connessi alla rete, stabilendo se muoversi o meno, in quale direzione e con quale velocità. Prima di effettuare la decisione l'Arduino analizza se la rete è critica ovvero se è a rischio di rottura. Inoltre a priori viene stabilito un fattore di attenuazione della forza di movimento che, insieme alla criticità della rete, modellano una Probabilità di Movimento. L'SN quindi tiene in considerazione la probabilità di movimento prima di prendere la decisione. Questo comporta che l'SN non si muoverà se il rischio di rottura della rete è alto e diminuirà la sua reazione a piccoli stimoli con l'aumentare del fattore di attenuazione (e viceversa). I dettagli implementativi verranno spiegati di seguito nel Capitolo 7.

Capitolo 7

Implementazione

7.1 Implementazione del client UDP Python

Come già accennato in precedenza, si è scelto di realizzare un programma python per ovviare al problema dell'assenza di un modo che consenta al modulo Wi-Fi utilizzato di ottenere l'RSSI dei nodi nella rete ad-hoc. L'idea è quella di utilizzare l'utility iw per i sistemi operativi GNU Linux. L'utility iw è basata su nl80211, l'interfaccia utente in fase di sviluppo per netlink. Netlink è una famiglia di socket utilizzata per IPC (Inter-process communication) tra il kernel e i processi in user-space.

Per ottenere le informazioni dei nodi connessi alla stessa rete ad hoc della macchina in questione, basta il comando:

```
$ iw dev <devname> station dump
```

dove <devname> è il nome dell'interfaccia Wi-Fi che si vuole utilizzare. In output fornisce le informazioni nel formato seguente:

```
Station 12:34:56:78:9a:bc (on wlan0)
    inactive time: 304 ms
    rx bytes:      18816
    rx packets:    75
    tx bytes:      5386
    tx packets:    21
    signal:        -29 dBm
    tx bitrate:    54.0 MBit/s
Station 00:38:12:e8:9a:bc (on wlan3)
```

```
inactive time: 201 ms
rx bytes: 10213
rx packets: 53
tx bytes: 33124
tx packets: 16
signal: -58 dBm
tx bitrate: 54.0 MBit/s
```

Se si vogliono avere le informazioni specifiche di un solo nodo è invece sufficiente il seguente comando:

```
$ iw dev <devname> station get <peer-MAC-address>
```

dove <peer-MAC-address> identifica l'indirizzo MAC dello specifico nodo. Nel caso in cui l'interfaccia sia connessa ad una rete in modalità infrastructure, l'output del comando restituirà solo le informazioni degli Access Point che costituiscono la rete, ma non dei client.

C'è da specificare che il tutto è possibile solo se i driver dell'interfaccia Wi-Fi utilizzata supportano nl80211.

Il programma non fa altro che inviare ogni n secondi fissati, un pacchetto UDP di cui il payload è composto in questo modo:

```
####P_ADDRESS,MAC_ADDRESS,RSSI;
```

e ad esempio:

```
## #10.42.1.11,00066671e1b7,-31;
```

Il formato scelto non ha una particolare importanza, stabilisce solo una convenzione nota tra i client e l'SN che dovrà scansionare e fare il parsing del pacchetto ricevuto.

7.2 Implementazione del core per Arduino

Il core si divide essenzialmente in due parti:

- Scansione e parsing dei pacchetti UDP.
- Controllo del movimento.

La prima parte è in relazione con il client UDP prima descritto. Questo verrà rimpiazzato con una lettura diretta degli RSSI sul modulo Wi-Fi collegato all'Arduino.

Il programma è composto da un main loop in cui viene costantemente controllato se ci sono pacchetti in lettura sull'interfaccia Wi-Fi. Nel caso ci fosse qualche pacchetto da

leggere, il programma inizia a scansionarlo, analizzandolo e controllando che il formato sia lo stesso di quello definito nella Sezione 7.1. Se il formato è corretto, le informazioni riguardanti il client che ha inviato il pacchetto vengono memorizzate in una struct del tipo:

```
typedef struct WiFiNode{
    char ip [IP_BUFFER_SIZE];
    char mac [MAC_BUFFER_SIZE];
    short int rssi;
    short int lb;
    short int position;
    float force;
    float criticality;
    bool empty;
}WiFiNode;
```

L'indirizzo IP, il MAC address e l'RSSI, sono già noti dopo il parsing del pacchetto UDP in quanto erano già scritte all'interno del pacchetto stesso. La variabile *lb* indica il Link Budget e viene calcolato secondo l'equazione (4.3), ovvero come la differenza tra la potenza di segnale ricevuta dal nodo che ha inviato il pacchetto e la sensibilità del modulo Wi-Fi collegato all'Arduino:

```
endPoints [idx].lb = endPoints [idx].rssi - SENSITIVITY;
```

La posizione di ogni nodo viene stabilita a priori. Dal momento che i nodi Wi-Fi simulatori e l'SN formano una linea retta, ciascun nodo simulatore può essere posizionato davanti o dietro l'SN e la loro posizione viene indicata rispettivamente con +1 e -1. Nell'algoritmo 1 si utilizza l'equazione (4.2) basata sulla legge di Hooke per definire la forze elastiche agenti su un SN. Trovandoci però in uno scenario ridotto dove non vengono considerati il versore \vec{u} e la costante di durezza k_j^{ST} , la variabile *force*, che indica la forza di attrazione o repulsione che l'SN ha verso il nodo simulatore di indice *idx*, viene calcolata esclusivamente in termini del parametro di spostamento δ (4.4), ovvero:

```
float force = (sqrt((max(endPoints [idx].lb , LB_REQ)) /
    (min(endPoints [idx].lb , LB_REQ))) -1) * forceDirection;
```

dove *LB_REQ* è il Link Budget minimo garantito nella rete e *forceDirection* indica il verso della forza che è uguale alla posizione (si ricorda che la posizione è +1 o -1) se la forza è attrattiva oppure opposta alla posizione se la forza è repulsiva. Se il link budget del nodo *idx* è minore di *LB_REQ* allora la forza è attrattiva, altrimenti è ripulsiva.

La variabile *criticality* indica la criticità del collegamento con il nodo *idx*, ovvero il rischio di rottura di tale collegamento. Viene calcolato in base all'equazione (4.5). La variabile booleana *empty* indica semplicemente se la struct di indice *idx* è ancora vuota o meno. Questo è necessario in quanto in questa fase le struct sono collezionate in un array di dimensione pari al numero di nodi Wi-Fi simulatori presenti nel sistema (2 per adesso). Inizialmente l'array è vuoto e viene riempito mano a mano che vengono ricevuti pacchetti dai nodi simulatori. Il controllo del movimento dell'SN viene effettuato dall'Arduino applicando l'algoritmo 1, quindi a intervalli periodici viene chiamata la funzione *checkRobot()* che effettua le seguenti operazioni:

1. Viene calcolata la forza risultante R (variabile che ha valore e direzione) come somma vettoriale delle forze dei nodi Wi-Fi simulatori. La forza agente su ogni nodo viene calcolata al momento della ricezione dei pacchetti UDP, come specificato sopra. Viene infine calcolata $RNorm$ come l'intensità della risultante R normalizzata nell'intervallo $[0, 1]$.
2. Viene calcolata la probabilità di movimento secondo l'equazione (4.6). In questo modo abbiamo una probabilità di movimento che cresce con l'aumentare di $RNorm$ ma decresce all'aumentare dell'attenuazione e della criticità C . La reazione dell'SN si può a questo punto immaginare: se l'SN è posto in una posizione pessima rispetto alla sua posizione ottimale per il sistema, esso si muoverà inizialmente a velocità massima, decelerando mentre si avvicinerà alla posizione ottimale. Inoltre, con valori di attenuazione alti (prossimi ad 1), l'SN reagirà sempre meno avvicinandosi alla posizione ottimale, in quanto diminuisce l'intensità degli stimoli. Ne risulta un avvicinamento molto graduale. Al contrario, con valori di attenuazione bassa (prossimi a 0) l'SN avrà un comportamento oscillatorio intorno alla sua posizione ottimale in quanto la reazione a piccoli stimoli sarà alta. Se poi si considera anche la criticità, l'SN tenderà a muoversi sempre meno frequentemente se il rischio di rottura di qualche collegamento è alto.
3. Infine viene lanciato un numero random. Se questo numero è minore della probabilità di movimento, viene mandato ai motori il segnale PWM corrispondente ad una velocità proporzionale ad $RNorm$ e nel senso corrispondente al verso della forza R . Altrimenti viene mandato un $PWM = 0$ ai motori fermandoli. Si noti che il cambio di velocità e direzione o spegnimento dei motori viene effettuato solo alla successiva chiamata della funzione *checkRobot()* ed è quindi molto importante la scelta del valore dell'intervallo temporale tra le loro chiamate. Un valore troppo basso potrebbe inserire un overhead di calcolo troppo grande per il microcontrollore e un intervallo troppo grande potrebbe portare l'SN ad oscillare ciclicamente tra i due estremi e non riuscirebbe a fermarsi nella posizione ottimale. Un comportamento simile potrebbe verificarsi anche nel caso sia assente il controllo probabilistico del movimento (ad esempio con attenuazione a 0 e p sempre ad 1),

in quanto l'SN reagirebbe molto ai piccoli stimoli (piccole variazioni di potenza di segnale ad esempio) e potrebbe quindi oscillare intorno alla posizione ottimale.

Capitolo 8

Risultati Sperimentali

8.1 Raccolta dati

Prima di descrivere la fase di test è opportuno spiegare come si è elaborata la parte software necessaria alla raccolta dei dati: l'SN manda ad un server Python in esecuzione su un nodo Wi-Fi simulatore, dei pacchetti UDP al termine di ogni chiamata *checkRobot()* con all'interno tutti i dati utili per l'analisi dello scenario e dei test. In futuro questa funzionalità potrà essere utilizzata per implementare il coordinamento della STEM-Mesh mandando questi pacchetti come dei BEACON in broadcast nella rete di SN.

8.2 Fase di test

Nella fase di test si è visto che l'SN ha un comportamento che rispecchia quanto precedentemente immaginato. Per effettuare i vari test si è posizionato l'SN tra i due nodi Wi-Fi simulatori a distanze differenti. Successivamente si è acceso l'SN il quale ha creato la rete ad-hoc, a cui si sono poi connessi i nodi simulatori. A questo punto l'SN era pronto a ricevere pacchetti UDP da scansionare per capire la potenza di segnale dei due nodi Wi-Fi simulatori. Per convenzione si iniziano a mandare pacchetti prima dal nodo simulatore, che è posizionato dietro l'SN, poi da quello posizionato davanti. Inoltre per semplicità in questa fase non viene effettuato il controllo sulla criticità dei collegamenti, anche per concentrarsi maggiormente sulle reazioni del dispositivo agli stimoli, che verrebbero eccessivamente attenuate dai controlli di criticità. Ovviamente il controllo verrà reinserito nelle successive fasi di test per verificare la stabilità globale della STEM-Mesh. Di seguito verranno esposti i test effettuati i cui i grafici in Figura 8.1 ne aiutano l'analisi.

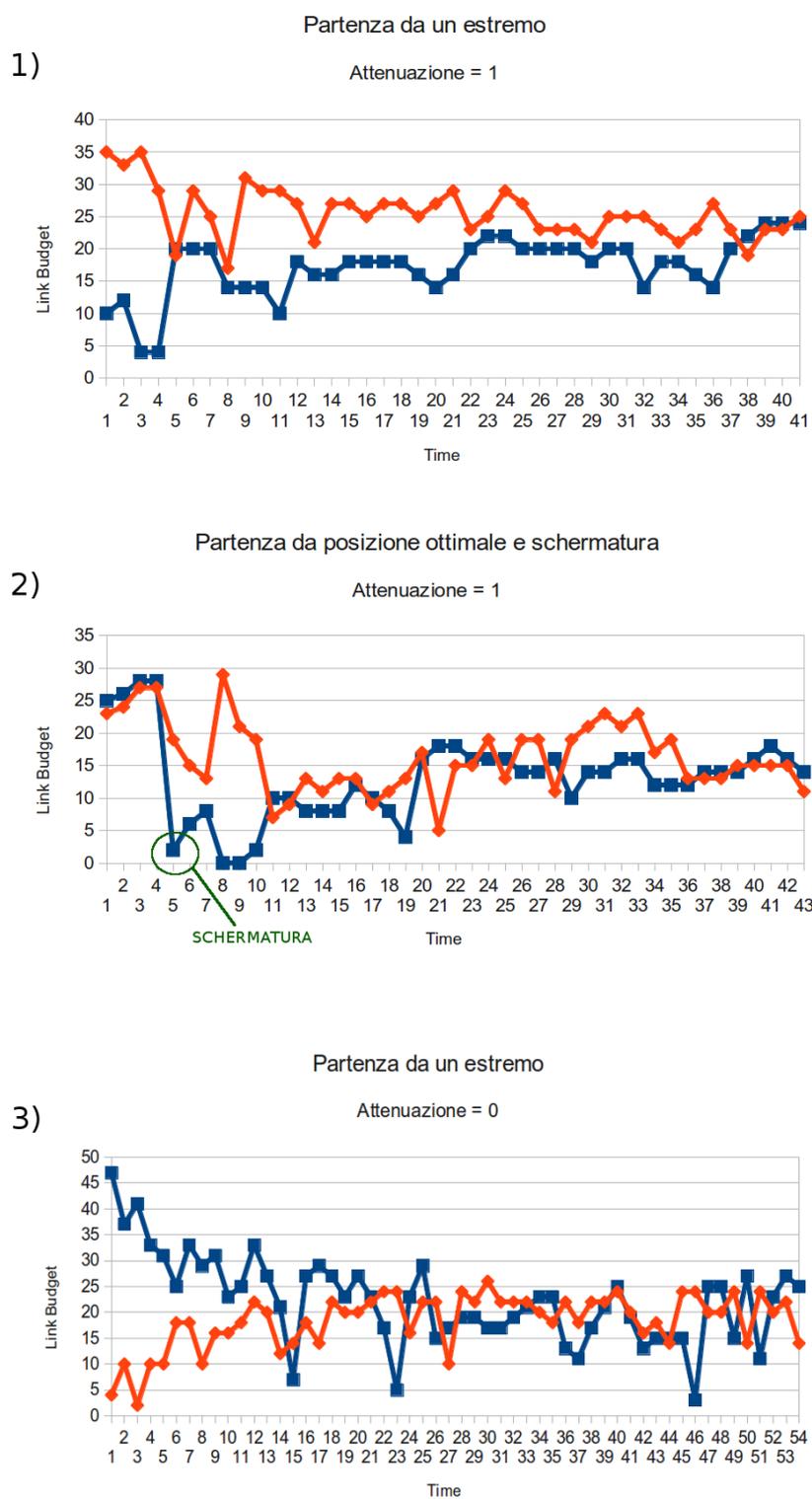


Figura 8.1: Test del dispositivo STEM-Node

1. Nel primo test si è posizionato inizialmente l'SN vicino ad un estremo del sistema, ne risulta quindi un buon Link Budget per il collegamento tra l'SN e il nodo simulatore dell'estremo vicino. Al contrario il Link Budget è molto basso per il collegamento con l'estremo opposto. Si può vedere dal grafico 1 che l'SN inizia a muoversi spostandosi esattamente verso il punto in cui il Link Budget è ottimale per poi oscillare intorno a quel punto fino a stabilizzarsi.
2. Nel secondo test invece si è voluto provare la reazione dell'SN a seguito di un cambio del sistema. Si può pensare a questo test come ad una continuazione del precedente, infatti nella situazione iniziale l'SN si trova già nella posizione ottimale rispetto al Link Budget. Successivamente (tempo 5 del grafico 2) viene schermato uno dei nodi simulatori simulando ad esempio l'allontanamento di un altro STEM-Node, oppure la caduta di un nodo o di un isola. Anche in questo caso, il grafico 2 mostra come, a seguito della schermatura, l'SN si muove stabilizzandosi intorno la posizione ottimale rispetto il Link Budget.
3. Nel terzo test si è modificato il parametro di attenuazione ponendolo a 0. In questo modo si è annullato il controllo probabilistico del movimento. L'SN è stato posto ad un estremo del sistema (come per il test 1) e si può notare dal grafico 3 che le due linee si avvicinano come per gli altri due test, ma continuando ad oscillare molto anche nei tempi finali. Questo è causato dal fatto che l'SN si muove anche per piccolissimi stimoli (ad esempio piccole modifiche dell'RSSI), che possono essere generati da piccoli disturbi, interferenze o schermature temporali del segnale.

È da notare anche che il Link Budget al termine di ogni esperimento tende ad assumere un valore prossimo al valore medio tra i due Link Budget iniziali (successivi alla schermatura per il test 2). Questo ovviamente deriva dal fatto che non si è considerata la criticità dei collegamenti, infatti con questo comportamento non si potrebbe assicurare che una rete sia stabile a garanzia di un Link Budget minimo prefissato. I test si sono però rilevati molto interessanti per accorgersi che il dispositivo effettivamente comprende la struttura dell'ambiente e reagisce in base alle sue modifiche. Con il controllo della criticità nei test futuri si aggiungerà un fattore stabilizzante ma il comportamento del dispositivo non dovrebbe subire modifiche radicali.

Segue una breve analisi dei consumi energetici di cui non è stata effettuata una reale verifica ma se ne è calcolata una stima in base alle specifiche dei componenti hardware utilizzati:

COMPONENTI	CONSUMO
Arduino UNO	~ 20-30mA
Motori	Corrente di stallo: ~1A
Modulo Wifly (Ricezione)	~ 40mA
Modulo Wifly (Trasmissione)	~ 180mA
TOTALE	~ 1250 mA

Figura 8.2: Analisi dei consumi energetici

8.3 Codice Sorgente

Tutto il codice sorgente scritto per l'implementazione di questo progetto è pubblicato sotto licenza GPLv2 Copyright (C) 2013 Matteo Martelli nei repository github: <https://github.com/matteomartelli/WiFlyBot>. È quindi stato rilasciato come software libero in modo da permettere a chiunque di utilizzarlo e per incoraggiarne lo studio, le modifiche e la redistribuzione secondo i termini della licenza.

Capitolo 9

Conclusioni e Sviluppi Futuri

L'obiettivo di questo progetto era quello di contribuire con il gruppo di ricerca del progetto STEM-Mesh per realizzare un primo prototipo parziale dell'unità STEM-Node. Il primo approccio allo studio e alla realizzazione di questa unità si è rivelato senz'altro promettente e molto interessante. Durante le varie fasi del processo ci si è accorti di come la materia del Disaster Recovery per le reti wireless fosse ampiamente trattata in molti studi di ricerca, i quali ne hanno proposto differenti modelli e applicazioni utilizzando varie tecnologie. Tra queste si sono rilevate particolarmente interessanti le reti wireless ad-hoc e le Cognitive Radio. Inoltre sono state esaminate possibili soluzioni e algoritmi per creare e gestire reti di dispositivi mobili automatizzati utili in queste situazioni. Si è scelto di basarsi sul progetto STEM-Mesh in quanto confrontando questo lavoro con la letteratura già esistente, si può appurare che si tratta di uno dei primi approcci che combina tecnologie Cognitive Radio con la teoria di mobilità controllata e con i principi delle reti ad-hoc. Durante lo svolgimento del progetto si è constatato che la documentazione dei dispositivi hardware e dei linguaggi di programmazione utilizzati fosse adeguata; tutto ciò ha agevolato il lavoro implementativo ma si sono riscontrati problemi nell'implementazione del Requisito 5.2.1 a causa di un problema hardware. Infatti i componenti scelti offrono una buona soluzione ai fini dello studio del movimento dell'SN, ma non forniscono un modo per ottenere l'RSSI dei nodi Wi-Fi presenti nella rete ad-hoc da parte dell'SN. Quest'ultima funzionalità è stata però simulata rendendo comunque possibile la realizzazione di un primo modello per l'SN. Questa questione potrà essere risolta tramite una modifica delle risorse utilizzate nelle successive fasi di implementazione. A questo proposito verranno fornite qui di seguito, alcune di quelle che potrebbero essere delle soluzioni hardware alternative che permettano il proseguimento dell'implementazione degli Stem-Node:

- La soluzione meno invasiva per il progetto sarebbe quella di rimpiazzare la Wireless Proto Shield e il modulo WiFly Shield RN-XV con l'Arduino Wi-Fi Shield. Il

firmware di questa scheda è software libero (al contrario della WiFly Shield che ha firmware proprietario), ma non supporta ancora le reti ad-hoc. Si potrebbe però lavorare alla scrittura del firmware cercando di introdurre il supporto per le reti ad-hoc ed il supporto per la ricezione degli RSSI dei nodi connessi alla rete.

<http://arduino.cc/en/Main/ArduinoWiFiShield>

- Un'altra soluzione interessante potrebbe essere quella di rimpiazzare l'Arduino board con il Raspberry Pi. Il Raspberry Pi è una scheda che, invece di un microcontrollore come l'ATmega328P per Arduino, monta un microprocessore ARM da 700Mhz. L'aspetto interessante per il progetto non sarebbe tanto la potenza di calcolo aggiuntiva offerta dal Raspberry Pi ma la sua ottima integrazione con i sistemi operativi GNU Linux. In questo modo si avrebbe la possibilità di utilizzare la parte di software che si occupa di ottenere gli RSSI nella rete ad-hoc (descritta nella Sezione 7.1) direttamente dall'SN, invece che utilizzarla come client dai nodi Wi-Fi simulatori. C'è però da considerare anche la differenza di consumo energetico: il Raspberry Pi consuma circa 300mAh (modello A) contro i circa 30 mAh dell'Arduino. Ciò può essere fondamentale nelle scelte di progettazione in quanto un punto chiave del progetto è garantire un'alta autonomia degli STEM Node e un minor consumo energetico possibile.

<http://www.raspberrypi.org/faqs>

- La comunità Arduino sta progettando un'altra board chiamata Arduino Yún. Questa scheda monterà sempre un microcontrollore con CPU a 8 bit (ATmega32U4) ma a differenza dell'Arduino UNO avrà in più una scheda Wi-Fi integrata (Atheros AR9331). Inoltre, un aspetto fondamentale è che su di essa si potrà far girare una distribuzione GNU Linux OpenWRT. Questo permetterebbe di avere una soluzione ottimale dal punto di vista implementativo, come descritta sopra per il Raspberry Pi, ma comunque mantenendo i consumi ridotti del microcontrollore ATmega.

http://store.arduino.cc/it/index.php?main_page=product_info&cPath=11_12&products_id=313

<https://openwrt.org>

Come ultima breve analisi, la prima fase di progetto è stata conclusa con i risultati desiderati.

Bibliografia

- [1] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks Journal*, 50(1), pages 2127–2159, 2006.
- [2] G. Aloï, L. Bononi, M. Di Felice, V. Loscrì, A. Molinaro, E. Natalizio, P. Pace, and G. Ruggeri. Stem-net: An evolutionary network architecture for smart and sustainable cities. 2012. submitted for publication.
- [3] C. Costanzo, V. Loscrì, E. Natalizio, and T. Razafindralambo. Nodes self-deployment for coverage maximization in mobile robot networks using an evolving neural network. *Computer Communications: 35(9)*, pages 1047–1055, 2012.
- [4] K. Derr and M. Manic. Extended virtual spring mesh (evsm): The distributed self-organizing mobile ad hoc network for area exploration. *IEEE Transactions on Industrial Electronics: 58(12)*, pages 5424–5437, 2011.
- [5] M. Di Felice, A. Trotta, L. Bedogni, L. Bononi, F. Panzieri, G. Ruggeri, V. Loscrì', and P. Pace. Stem-mesh: Self-organizing mobile cognitive radio network for disaster recovery operations. 2013. To appear on Proc. of the 9-th IEEE International Wireless Communications and Mobile Computing Conference(IWCMC 2013), July 1-5, 2013, Cagliari, Italy.
- [6] FCC. Et docket no 03-322. 2003.
- [7] S. M. George, W. Zhou, H. Chenji, M. Won, Y. O. Lee, A. Pazarloglou, R. Stoleru, and P. Barooah. Distressnet: a wireless ad hoc and sensor network architecture for situation management in disaster response. *IEEE Communication Magazine: 48(3)*, pages 128–136, 2010. Cape Town, South Africa.
- [8] A. Gorcin and H. Arslan. Public safety and emergency case communications: opportunities from the aspect of cognitive radio. *Proc. of IEEE DYSPAN*, 2008. Chicago, USA.

-
- [9] N. Pogkas, G. E. Karastergios, C. P. Antonopoulos, S. Koubias, and G. Papadopoulos. Architecture design and implementation of an ad-hoc network for disaster relief operations. *IEEE Transactions on Industrial Informatics: 3(1)*, pages 63–72, 2007.
 - [10] B. Shucker and J. Bennet. Target tracking with distributed robotic macrosensors. 2005.
 - [11] Chai Keong Toh. *Ad Hoc Mobile Wireless Networks*. Prentice Hall Publishers, 2002.
 - [12] N. Uchida, K. Takahata, and Y. Shibata. Evaluation of cognitive wireless networks in rural area for disaster information network. *Proc. of IEEE ICCSA*, 2011. Santander, Spain.
 - [13] Leonardo Vianello. *Radio Cognitive: Stato dell'Arte*. Padova, 18 Febbraio 2010.

Ringraziamenti

Desidero ringraziare Luciano Bononi, Marco di Felice e Angelo Trotta per avermi aiutato nella realizzazione del progetto e della mia tesi, fornendomi sia tutto il materiale che tutto il supporto tecnico necessario.

Ringrazio Stefania sia per l'aiuto pratico che per il supporto morale che è riuscita a dedicarmi non solo per questo lavoro ma durante tutto il corso dei miei studi.

Ringrazio poi la mia famiglia per avermi sempre appoggiato e sostenuto nelle mie scelte.

Un ultimo ringraziamento va anche ai miei colleghi universitari nonché amici per avermi prestato il loro aiuto, tra cui Stefano per essersi interessato al mio lavoro ed essere stato spesso soggetto di confronto tecnico e Raffaele per avermi aiutato nello svolgimento dei test esposti in questo documento.