# Capitolo 6 – Simulazioni e valutazione risultati

### Organizzazione simulazioni

* In cosa consiste la realizzazione
* Come sono strutturate le simulazioni
* I Principali parametri che differenziano e caratterizzano i risultati delle simulazioni

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

La parte di realizzazione sperimentale del nostro progetto, è stata fatta tramite simulazioni con lo scopo di verificare il comportamento del nostro algoritmo e raccogliere dati per poter fare un’analisi più oggettiva delle sue prestazioni al variare delle condizioni. Per avere una sufficiente base statistica, abbiamo impostato nel file di inizializzazione il numero di simulazioni a venti. Significa che per ogni configurazione il simulatore eseguirà venti simulazioni consecutive senza reimpostare il generatore di numeri casuali, infatti in questo modo abbiamo ottenuto ogni volta, venti configurazioni di rete differenti. Per impostare il numero di ripetizioni da fare per ogni configurazione, abbiamo inserito nella sezione *General* di ogni file di inizializzazione utilizzato, l’istruzione “*repeat = 20*”.

Le simulazioni sono caratterizzate da due parametri principalmente: la densità di nodi e il raggio di trasmissione del BT. Come presentato nella Sezione 5.2, le densità che abbiamo scelto sono:

1. D = 0.02 nodi/m2,
2. D = 0.01 nodi/m2,
3. D = 0.008 nodi/m2,
4. D = 0.001 nodi/m2,
5. D = 0.0005 nodi/m2,
6. D = 0.0001 nodi/m2.

e i raggi ρ dei trasmettitori sono:

* ρ = 15 m,
* ρ = 50 m.

Le densità da D=0.02 nodi/m2 a D=0.001nodi/m2 sono state pensate per poter simulare ambienti urbani che possono essere dal mediamente popolati al densamente popolati. Situazioni simili possono essere il caso di medie città, grandi città o per le densità più elevate, situazioni di forte concentrazione di persone in un area ristretta; quest’ultimo tipo di situazione può essere generata da forte concentrazione abitativa come una serie di palazzi o condomini vicini tra loro, ma anche da motivazioni esterne alla semplice concentrazione abitativa come possono essere eventi di qualunque genere. Le densità più piccole, D=0.0005 nodi/m2 e D=0.0001nodi/m2 sono state pensate per poter studiare la scalabilità del sistema a fronte di dispersione dei nodi su una vasta area, ma anche per poter simulare situazioni urbanistiche tipiche di piccoli comuni, con un relativo basso numero di abitanti come ad esempio i piccoli paesi in pianure o ampie vallate. Abbiamo quindi organizzato il processo di realizzazione sperimentale in modo da eseguire, per ogni densità, simulazioni per entrambi i raggi ρ e collezionare dati per un’analisi a posteriori delle performance.

Il principale risultato che abbiamo voluto monitorare è stata la percentuale di diffusione di un messaggio, al diminuire della densità, per studiare quali fossero i limiti di applicabilità in termini di efficienza della nostra soluzione e per poter avere un insieme di scenari in cui l’algoritmo da noi proposto presentasse buoni livelli di prestazioni. Parallelamente abbiamo anche raccolto dati sul tempo totale di trasmissione, inteso come l’istante ti tempo in cui l’ultima trasmissione viene conclusa. Abbiamo scelto di raccogliere anche quest’ultimo dato perché abbiamo cercato di capire con che tempistica il messaggio viene diffuso. Il motivo è stato di voler dare un corrispettivo valore temporale alla percentuale di nodi che l’algoritmo raggiunge. Da solo un valore di tempo massimo di trasmissione non è valutabile, poiché è molto dipendente dalla dispersione spaziale della rete, infatti esso deve essere analizzato in coppia con il valore percentuale di rete coperta durante la diffusione per la rispettiva simulazione presa in analisi. In questo modo possiamo avere un idea di come il tempo di propagazione evolva al variare della copertura e della densità della rete.

### Algoritmo Dynamic Fanout: funzionamento

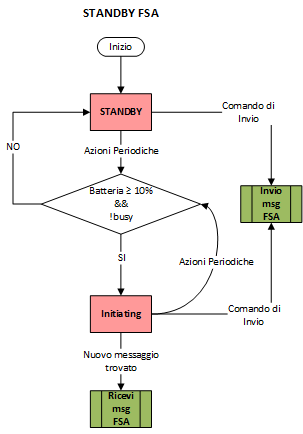
* Caratteristiche dell’algoritmo, come funziona e quali/come vengono eseguite certe operazione in fase di simulazione
* Diagrammi di flusso dell’algoritmo

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

In questa sezione diamo una breve spiegazione sul funzionamento dell’algoritmo all’atto pratico dell’esecuzione. Presenteremo il suo comportamento tramite diagrammi di flusso, uno per ognuna delle possibili situazioni che l’algoritmo si può trovare ad affrontare. Ogni diagramma rappresenta il comportamento della relativa macchina a stati virtuale che descrive il comportamento dell’algoritmo in quella specifica situazione. I tre principali automi sono quelli di: *Standby*, *Invio Messaggio* e *Ricevi Messaggio*. Vi è anche una quarta parte che viene eseguita in parallelo ai primi tre, che si occupa solo di organizzare e temporizzare l’esecuzione delle azioni periodiche.

#### Standby fsa

Partiamo con la *Standby fsa*, il diagramma che descrive il comportamento dell’algoritmo durante i periodi di attesa e ascolto. Il diagramma riportato in Figura 6.1, descrive il generale comportamento del dispositivo quando esso non è impegnato in nessun tipo di scambio dati attivo. Dato che lo stato di Standby è raggiungibile da qualsiasi stato, l’algoritmo forza senza alcun controllo l’entrata in quello stato. Poi, in maniera temporizzata, l’algoritmo esegue delle azioni periodiche che hanno un duplice scopo: aggiornare la visione dello stato circostante il dispositivo e aggiornare i parametri di conseguenza e simulare il consumo energetico del dispositivo nel tempo. Dopo aver eseguito queste operazioni periodiche, l’algoritmo esegue un controllo sulla batteria del dispositivo. Se essa è superiore al 10%, il sistema si porta in stato di ascolto (Initiating) in attesa di eventuali nuovi messaggi, mentre se inferiore alla soglia limite l’algoritmo considera il dispositivo fuori dalla zona di lavoro e lo riporta in stato di Standby. Continuerà comunque ad effettuare le azioni periodiche e qual ora la batteria tornasse sopra il 10%, il sistema tornerebbe operativo.



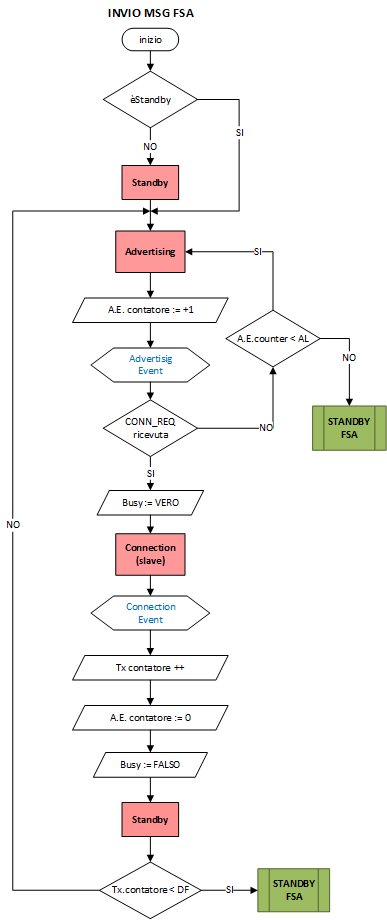
*Figura 6.1: Diagramma di flusso della Standby fsa.*

Il sistema, oltre al livello della batteria, controlla anche che il dispositivo non sia “*busy”* (occupato) in quale operazione o scambio di informazione. Questo ulteriore controllo serve a garantire il completamento delle operazioni o transazioni iniziate quando la batteria era sopra il 10% e durante l’esecuzione è scesa sotto la soglia limite. E’ anche vero che lo standard BLE da solo implementa un insieme di controlli per rilevare connessioni perse, ma in questo facciamo si che l’algoritmo garantisca che anche quest’ultima operazione di invio o ricezione venga completata e non interrotta.

Sia se il dispositivo si trovi in stato di Initiating sia se si trovi in quello di Standby, se l’utente intende inviare un messaggio, il sistema si sposta ad eseguire un’altra serie di operazioni. Il sistema permette un invio di un nuovo messaggio generato dall’utente anche se la batteria è inferiore alla soglia limite in quanto crediamo che negli scenari da noi ipotizzati, se l’utente vuole iniziare a diffondere un nuovo messaggio, sia una valida ragione per eseguire l’operazione anche in condizioni di scarso livello energetico. In questa particolare situaizone, l’algoritmo avrà comunque un DF pari a uno o al massimo pari a due, quindi si cerca solo di far fare uno sforzo al dispositivo per iniziare la diffusione del messaggio e non molto di più.

#### Invio Messaggio fsa

Il diagramma in figura 6.2 illustra il funzionamento del sistema quando esso cerca di inviare un nuovo messaggio arrivatogli dalla rete o che l’utente vuole inviare. Per prima il sistema controlla di essere nello stato di standby e nel caso non lo fosse, ci si porta. Una volta in Standby è possibile passare allo stato di Advertising.



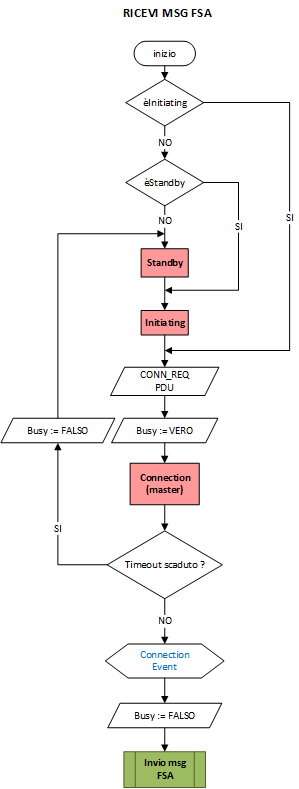
*Figura 6.2: Diagramma di flusso Invio Messaggio fsa.*

Dopo essere entrato in Advertising, viene incrementato il contatore degli advertising e poi viene eseguito l’advertising event vero e proprio. Il sistema si mette quindi in attesa di eventuali pacchetti di tipo CONN\_REQ in riposta ai suoi pacchetti. L’attesa di un pacchetto di CONN\_REQ è regolato da un timeout, quindi l’attesa non sarà mai infinita. Il significato di quel preciso nodo decisionale significa appunto se è stata ricevuta una richiesta di connessione entro il timeout. Se il sistema non riceve una richiesta di connessione entro lo scadere del timeout allora controlla se il contatore di advertising è inferiore all’AL (Advertising Limit); se lo è si riporta in stato di Advertising, incrementa nuovamente il contatore e ricomincia, se invece il contatore è maggiore o uguale all’AL, il sistema si ferma e ritorna a seguire il flusso di istruzioni della Standby fsa, mettendosi poi in ascolto di eventuali nuove informazioni. Questo pezzo di flusso esecutivo rappresenta una delle due parti di controllo dinamico che abbiamo progettato per estendere e adattare l’algoritmo originale al nostro problema.

Se invece il sistema riceve una richiesta di connessione entro il timeout, esso si marca come occupato e passa nello stato di Connection in *Slave Role*. A questo punto arriverà il primo pacchetto di *pool* da parte del *Master* *Role*, inizierà il connection event e all’interno di esso verrà trasmetta tutta l’informazione. Una volta terminato il connection event, il sistema incrementa il contatore di trasmissioni effettuate con successo, azzera il contatore degli advertising a vuotoconsecutivi e resetta lo stato di busy a falso. Successivamente si porta in Standby e controlla se il contatore di trasmissioni effettuate è minore del DF. Se si, il sistema ricomincia tutta la procedura di invio messaggio, se no vuol dire che ha completato il suo lavoro per quell’informazione e può tornare ad eseguire la Standby fsa e mettersi in ascolto per la prossima informazione.

#### Ricevi Messaggio fsa

Nel caso vi siano uno o più dispositivi che stanno tentando di diffondere informazioni che il nodo in questione non ha ancora ricevuto, esso inizia ad eseguire la sequenza di istruzioni del Ricevi Messaggio fsa. Dallo stato di Initiating del diagramma della Standby fsa, si passa al diagramma in figura 6.3. All’inizio il dispositivo si porta correttamente in stato di Initiating se già non lo era e poi invia una richiesta di connessione al mittente dei pacchetti di advertising. Poiché sta tentando di instaurare una nuova connessione con un determinato dispositivo, dopo aver mandato la richiesta di connessione, viene messo a vero il flag busy. Dopo esser entrato in stato di Connection in *Master Role*, il sistema invia il pacchetto di *pool* necessario a autorizzare lo slave a iniziare il trasferimento dati. Dopo aver inviato la pool request, il dispositivo rimane in attesa dell’informazione; se scatta il timeout vuol dire che il mittente non ha accettato la sua richiesta oppure qualche altro dispositivo ha inviato una richiesta di connessione ed è stato scelto (la politica di accettazione è FIFO, anche se non vi è una vera e propria coda). Nel caso il timeout scade, il dispositivo capisce di non essere stato scelto, quindi imposta a falso il flag busy e si riporta in stato Initiating, di ascolto. Nel caso invece la sua richiesta sia la prima ad arrivare e venga accettata, dopo la pool request, il dispositivo slave manda al master l’informazione durante il Connection Event. Al termine del trasferimento la connessione viene chiusa e il sistema automaticamente inizia ad eseguire la sequenza di istruzioni della Invio Messaggio fsa.



*Figura 6.3: Diagramma di flusso Ricevi Messaggio fsa.*

#### Azioni Periodiche

Il sistema ha un processo in background, che schedula delle azioni da eseguire periodicamente e queste azioni servono per aggiornare i parametri utilizzati in modo che dinamicamente si adattino all’ambiente esterno e allo stato attuale del dispositivo stesso, ma anche per effettuare dei controlli. Il controllo principale è quello fatto sulla batteria. Il sistema periodicamente rileva lo stato della batteria e se il livello è al di sopra della soglia limite (10% soglia limite), il sistema, se può, porta il dispositivo in stato di Standby rendendolo inabile a partecipare alla diffusione dei messaggi. Il motivo è di non lasciare al sistema la possibilità di consumare quell’ultima porzione di batteria e di lasciarla all’utente per le sue necessità. L’unica eccezione è di permettere all’utente, anche se la batteria è al di sotto della soglia limite, di inviare un messaggio. Abbiamo ritenuto necessario lasciare questa possibilità, in quanto negli scenari di lavoro da noi analizzati, potrebbe essere utile per poter diramare qualche informazione d’emergenza molto utile, anche a costo della rimanente poca batteria rimasta.

Questi controlli avvengono periodicamente in background, anche durante eventi di trasmissione o ricezione. Per questo motivo abbiamo inserito il flag busy, in modo da segnalare qualora il sistema sia impegnato in una transazione e quindi non sia possibile da parte del controllo periodico forzare uno stato Standby. Questo perché non vogliamo che vada a intralciare il sistema nel mentre di una transazione, ma può comunque agire tra una trasmissione e l’altra.

Anche quando la batteria è sotto la soglia limite, i controlli vengono sempre effettuati vista la loro esigua richiesta energetica e anche per poter riabilitare il sistema al lavoro attivo qualora la batteria venga ricaricata e torni sopra soglia.

Una parte dei controlli periodici sono già stati inseriti e rappresentati nel diagramma di flusso della Standby fsa, per dare una semplice idea di quale sia il compito di questi controlli, ma non sono vincolati a nessun stato né macchina a stati; come detto prima queste azioni periodiche sono eseguite sempre e comunque in background nel sistema.

### Valutazione risultati

* Valutazione risultati
* Sottolineare che il decadimento delle prestazioni avviene per densità più basse di un paesino di campagna
* Riferimento al documento delle prestazioni (da ritrovare) in cui dice che per algoritmi rumor mongering non è possibile stabilire una complessità, a causa delle possibili scelte nei metodi di terminazione.
* Il nostro caso presenta estrema variabilità e quindi non è possibile stabilirne con che legge si diffondano le informazioni né in quanto tempo.
* Dire che In una ipotesi ideale avremmo certi tipi di comportamenti
* Ribadire che vi sono altri fattori non considerati, quali la mobilità dei nodi, che possono aumentare l’efficacia dell’algoritmo.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*