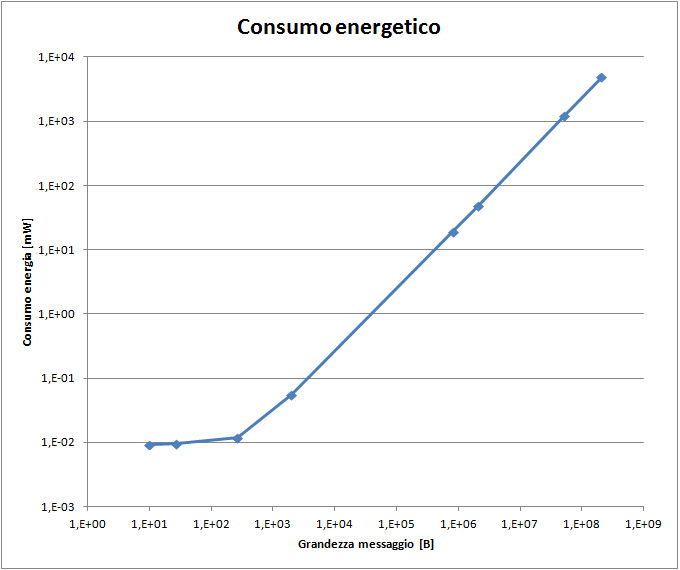
# Capitolo 4 – Progetto logico della soluzione del problema

* Dove è stata ricercata la soluzione
* Perché di quel sistema trasmissivo
* Perché gli smartphone
* SOTTOLINEARE la dinamicità della soluzione
* Ribadire che la soluzione non vincola il dispositivo
* La soluzione permette all’utente di fare cmq tutto ciò che vuole col dispositivo
* Analisi di fattibilità ???

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

La nostra soluzione si concentra quindi nel ricercare un possibile modo di diffondere messaggi e informazioni tra la popolazione, in situazioni dove le principali reti di comunicazione non sono disponibili. Ci siamo quindi concentrati su ciò che rimane disponibile in questi scenari e quali siano i metodi per modellare la situazione. Ci siamo concentrati sullo sfruttare ciò che è di più comune uso, ciò che è molto diffuso e facilmente in possesso di ogni persona. Da questo studio è risultato che gli smartphone rispecchiano questo profilo, infatti, al giorno d’oggi in media ogni persona possiede uno smartphone. Siamo passati quindi a studiare ciò che uno smartphone potesse fornire o ciò che di utile avremmo potuto sfruttare. Uno smartphone, ma in generale ogni dispositivo mobile, senza la presenza delle comuni reti di comunicazione, quali internet e telefonia, ha solo altri due modi di poter comunicare con altri dispositivi mobili: la tecnologia a infrarossi e la tecnologia Bluetooth. Da un rapido studio è emerso che la scelta degli infrarossi non avrebbe avuto portato a una soluzione performante e applicabile, mentre il Bluetooth si è rivelato essere un ottimo punto di partenza per un caso di studio. Il sistema Bluetooth in generale, come presentato nella Sezione 2.2, è un sistema di comunicazione Peer-to-Peer e quindi permetterebbe di mettere in comunicazione due dispositivi mobili anche in assenza delle reti di comunicazione. A questo punto ci siamo concentrati sul capire come poter sfruttare questa tecnologia per progettare una soluzione. Gli aspetti principali studiati sono: il modello di rete da utilizzare, il metodo con cui guidare e gestire la diffusione dell’informazione e infine come tutto ciò vada a impattare sul consumo energetico dei dispositivi. Un altro aspetto fondamentale di questo lavoro è stato tenere in considerazione il consumo energetico aggiuntivo che si va a introdurre su singolo dispositivo, poiché se l’impatto energetico fosse troppo elevato, si avrebbe una considerevole riduzione nell’autonomia del dispositivo stesso, rendendo la soluzione proposta impraticabile.

Abbiamo fatto inizialmente uno studio sul consumo energetico richiesto da una singola trasmissione dati tramite BLE. Abbiamo poi studiato diversi casi, in cui abbiamo fatto variare la grandezza del pacchetto trasmesso e valutato i consumi energetici richiesti. Dal nostro studio è emerso che l’energia richiesta per il trasferimento di una singola informazione, anche di discrete dimensioni, non vada a inficiare in maniera gravosa il consumo energetico medio dei più comuni smartphone in commercio. Lo standard BLE utilizza molti parametri per qualsiasi evento di trasmissione o ricezione. Abbiamo utilizzato Excel per costruire un modello in grado di simulare calcolare il tempo impiegato per la trasmissione di un’informazione, specificandone la grandezza e tutti i parametri che caratterizzano la trasmissione BLE. Molti parametri possono variare su intervalli piuttosto larghi, ma come illustrato nella documentazione ufficiale e anche in [*sensors-12-11734*], più i valori dei parametri crescono più il consumo energetico richiesto mediamente diminuisce. Per questo motivo, il nostro studio ha mantenuto tutti i parametri ai loro valori minimi per studiare il caso di massimo consumo energetico medio.



*Figura 4.1: Consumo energetico per trasmissione di una singola informazione.*

Nel nostro studio abbiamo fatto variare la grandezza dell’informazione da pochi Byte fino a 200 MB, in modo da vedere il comportamento con informazioni grandi e realmente possibili. In figura 4.1 riportiamo su grafico i risultati ottenuti e, come si può notare, dopo un certo valore, l’andamento del consumo energetico rimane lineare e diretto con la grandezza dell’informazione. L’andamento iniziale molto basso è dovuto alla predominazione dei tempi di attesa e connessione dello standard BLE sul tempo di trasmissione dell’informazione. La trasmissione dati su BLE prevede che dopo un’attesa variabile iniziale, la trasmissione avvenga tramite *connection event* consecutivi; un *connection event* è un evento nel quale sono scambiati dati tra due dispositivi connessi tra loro. Infine abbiamo stimato quante trasmissioni un dispositivo possa affrontare, oltre al suo carico medio di lavoro prima di scaricarsi. Il calcolo prevede che si parta sempre a batteria completamente carica e abbiamo considerato i più comuni e attuali smartphone sul mercato. La stima ovviamente dipende dalla grandezza dell’informazione e varia da un minimo attorno alle 200 trasmissioni fino a un massimo nell’intorno delle 400 trasmissioni.

Lo studio sul consumo energetico ci ha confermato che il basso consumo energetico richiesto dalla tecnologia BLE, permette di sfruttare questo sistema di comunicazione per diffondere messaggi tra nodi mobili. Dato il poco consumo energetico, la nostra soluzione prevede che il dispositivo mobile non debba dedicarsi completamente a questo servizio, anzi esso rimarrà sempre a disposizione dell’utente senza che la nostra soluzione limiti all’utente l’uso del dispositivo. Questo perché ormai sui dispositivi mobile, quali smartphone o tablet ad esempio, vi sono tante altre applicazioni o funzioni utili all’utente che debbano rimanere fruibili in ogni situazione, anche in scenari come quelli da noi ipotizzati.

Dopo l’aspetto energetico, ci siamo dedicati allo studio di un modello di rete che potesse modellare correttamente la disposizione dei dispositivi e del mezzo di comunicazione. Il BLE, ma la tecnologia Bluetooth in generale, è un sistema di comunicazione molto legato alla potenza di trasmissione del trasmettitore e dalla relativa distanza che tale potenza consente di raggiungere. Per questo motivo abbiamo scelto di modellare la nostra rete di dispositivi mobili con una *Random Geometric Graph*, presentata nella Sezione 2.3.2.

Infine dopo l’aspetto del modello di rete, ci siamo dedicati allo studio di un sistema di regole che potessero governare la diffusione delle informazioni. Dopo un’attenta analisi è emerso che per com’è composta la rete e per il tipo d’informazioni, un modo efficace di diffondere informazioni è quello del passa parola. Per questo motivo abbiamo preso in considerazione il mondo degli algoritmi di gossip, o epidemici, creati per modellare la diffusione d’informazioni tra le persone e sui social network, o nel caso degli epidemici come si diffonde un’epidemia in una popolazione. Tra gli algoritmi di gossip, abbiamo scelto di estendere l’algoritmo del Fixed Fanout, un algoritmo di gossip di tipo Rumor Mongering presentato nella Sezione 2.4.4. Gli algoritmi di gossip sono stati pensati dal punto di vista di dover modellare la diffusione d’informazioni tra nodi di una rete, non in grado di muoversi. Inoltre la nostra soluzione è stata progettata perché tenga in considerazione il consumo energetico che sta generando sul dispositivo. Per questo motivo abbiamo progettato il nostro algoritmo, estendendo quello già esistente con elementi di forte dinamismo. Infatti, i parametri da noi progettati, regolano l’algoritmo in modo dinamico, facendolo adattare allo stato dell’ambiente esterno, inteso come numero di dispositivi presenti nelle vicinanze, e allo stato interno del dispositivo, in altre parole al livello della batteria in modo da regolare lo “sforzo” in maniera corretta. Il nostro algoritmo cerca sempre di regolare i parametri in modo da garantire un buon compromesso tra prestazioni e consumo energetico/carico di lavoro. Dopo un’attenta analisi abbiamo deciso di progettare l’algoritmo con due differenti valori di reattività, di fronte a cambiamenti nella rete o nello stato interno del dispositivo oppure in entrambi.

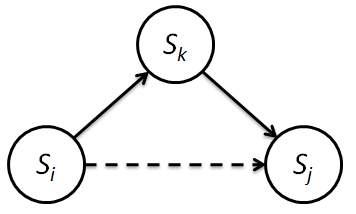
## Modello di rete

* Il modello di rete scelto
* I motivi della scelta
* Le caratteristiche della rete

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

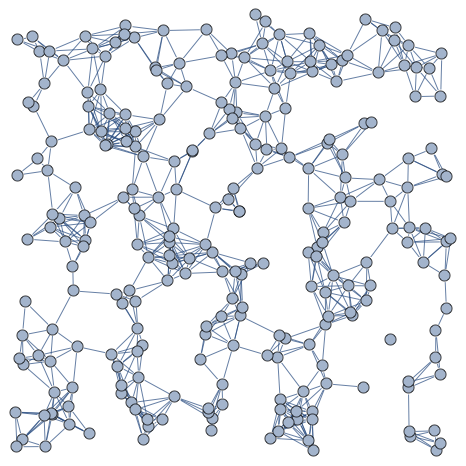
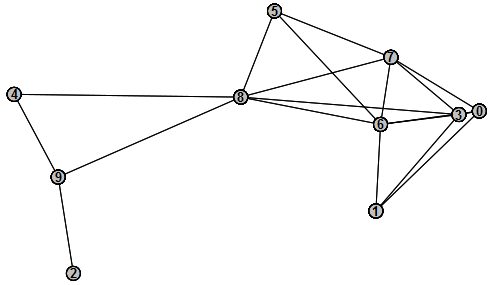
Dopo aver ricercato in letteratura, abbiamo capito che il modello di rete di tipo Peer-to-Peer è adatto a modellare la rete che abbiamo studiato. Le reti Peer-to-Peer sono reti di nodi totalmente paritetici, senza alcuna struttura gerarchica o differenziazione tra i nodi stessi. Per questo motivo ogni nodo della rete è chiamato peer e lo scambio d’informazioni avviene sempre tra due peer alla volta. Vi sono molti modelli di reti in letteratura, ma ne abbiamo studiati solo alcuni, presentati nella Sezione 2.3. Tra quei modelli di rete, abbiamo scelto quello che riesce a modellare al meglio le caratteristiche della rete da noi in esame: il modello che abbiamo scelto è il *Random Geometric Graph*. Questo tipo di modello è costituito da un grafo *G*(*N,ρ)* bidirezionale casuale inserito in un’area limitata ed è composto da N nodi. Il parametro ρ rappresenta il “raggio” entro il quale sono stabiliti i collegamenti tra i nodi. Il raggio ρ rappresenta la distanza geometrica entro la quale un nodo stabilisce collegamenti con altri nodi. Due nodi a una distanza inferiore di ρ avranno un collegamento bidirezionale, mentre due nodi a una distanza maggiore di ρ non avranno nessun collegamento tra loro. Questo tipo di modello presenta una basa degree variance, ma un’alta edge dependency. Sono due parametri che rispettivamente indicano la varianza nella distribuzione del grado dei nodi e un fattore di dipendenza dagli archi del grafo che indica la probabilità che vi siano collegamenti tra nodi vicini.

* ***Edge dependency:*** anche chiamata interdipendenza tra archi. Dato un grafo come in figura 4.2, la edge dependency è definita come la probabilità subordinata che esista un arco tra gli stati *Si* ~ *Sj*, sapendo che esiste un arco tra gli stati Si ~ *Sk* e tra *Sk* ~ *Sj*. Formalmente, questa probabilità si esprime così e si può considerare alta se è maggiore della sua probabilità semplice .
* ***Degree variance:*** con degree variance s’indente la varianza del grado dei nodi del grafo.



*Figura 4.2: Esempio di edge dependency.*

Nel nostro caso di studio il raggio ρ modella la portata del trasmettitore Bluetooth, infatti, solo dispositivi che si trovano entro il raggio d’azione dei rispettivi trasmettitori possono comunicare, mentre dispositivi fuori portata non possono avere alcun collegamento. La costruzione di questa rete è poi discussa più nel dettaglio nel Capitolo 5, ma in generale si sceglie una determinata area e si dispongono i nodi in modo causale in quest’area. Dopo s’inseriscono i collegamenti controllando le distanze tra i nodi. In figura 4.3 sono riportati due esempi di RGG.

*Figura 4.3: Esempi di Random Geometric Graph.*

Com’è possibile intuire, il parametro del raggio ρ è la chiave principale che determina quanto siano connessi i nodi del grafo. Più il raggio è piccolo più la probabilità di avere sottografi isolati e/o archi *bridge* tra i sottografi è elevata. Un bridge è un arco singolo che collega un nodo di un sottografo con un nodo di un altro sottografo, rendendo quell’arco l’unica possibile via di comunicazione tra i due sottografi. Un raggio ridotto, diminuisce la possibilità di avere collegamenti ridondanti per la diffusione dell’informazione e aumenta la possibilità di avere gruppi di nodi isolati e la formazione di bottleneck nella rete. Al contrario, più il raggio cresce più la probabilità di avere nodi o sottogruppi di nodi isolati diminuisce, come anche la probabilità di avere archi bottleneck e la rete tenderà a essere sempre più connessa. Va detto anche che oltre al raggio ρ, è altrettanto importante la dimensione dell’area in cui la rete si trova. Nonostante l’area non sia un parametro del grafo, essa influisce, insieme al raggio ρ, a rendere il grafo più o meno connesso. Il motivo è dovuto al metodo con cui è generato il grafo, in altre parole presa un’area, i nodi vengono uniformemente e casualmente distribuiti in quest’area. Com’è intuibile, con N fissato, più l’area è grande più la densità dei nodi per metro quadro sarà bassa, mentre più l’area è piccola più la densità sarà elevata. Questo fattore di dispersione ci ha permesso di poter simulare scenari diversi, dal grosso centro urbano con alta concentrazione di popolazione, fino al paese di campagna dove la gente è molto più diradata. Per questo motivo, è stato più importante lavorare secondo ipotesi di diverse densità di nodi; scelta la densità, l’area è regolata di conseguenza. Come già detto in precedenza, il nostro algoritmo ha una forte parte dinamica, che tiene in considerazione le variazioni dell’ambiente esterno. Negli scenari da noi considerati, è molto facile che la densità di popolazione in certe zone possa variare velocemente e questo il motivo abbiamo implementato la nostra soluzione con elementi dinamici in grado di adattarsi a questi cambiamenti.

## Bluetooth 4.0 Low Energy

* Richiamo alla struttura del link layer
* Richiamo agli stati della FSA

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

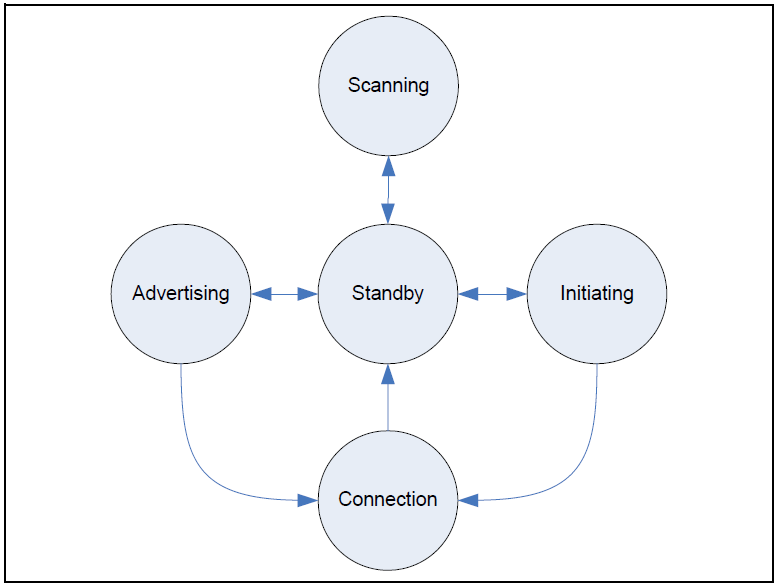
Abbiamo scelto di utilizzare il Bluetooth Low Energy come mezzo di comunicazione perché, in situazioni di assenza di tutte le altre reti di comunicazione, esso rimane comunque operativo. Il BLE, presentato nella Sezione 2.2, è uno standard di comunicazione Peer-To-Peer ufficializzato nel 2010. Nella sua versione 4.0, la casa produttrice ha presentato uno standard denominato Low Energy, infatti, questa nuova tecnologia ha consumi ridotti anche di dieci volte rispetto alle sue versioni precedenti. Ciò ha contribuito alla sempre maggior diffusione e utilizzo di questo standard nei settori del monitoraggio ambientale, della sensoristica, dei dispositivi *wearable* in coppia con i dispositivi mobile. Questa tecnologia è stata sviluppata per lo scambio di piccoli dati tra due enti e quindi ottimizzando i consumi energetici per questo tipo di carico di lavoro; il BLE non è stato progettato per lo streaming. Questo standard presenta ridotti tempi di latenza e parametri di connessione più semplici, rendendo più veloce connettersi a un dispositivo. Questo standard è equipaggiato su ogni dispositivo mobile in commercio e i modelli più recenti sono già dotati della versione 4.1. Da un indagine ISTAT [*Cittadini e nuove tecnologie - 18-dic-2014 - https://www.google.it/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CCcQFjABahUKEwjngduCvILHAhXKBiwKHbEuD38&url=http%3A%2F%2Fwww.istat.it%2Fit%2Ffiles%2F2014%2F12%2FCittadini\_e\_nuove\_tecnologie\_anno-2014.pdf%3Ftitle%3DCittadini%2Be%2Bnuove%2Btecnologie%2B-%2B18%252Fdic%252F2014%2B-%2BTesto%2Bintegrale.pdf&ei=mOS5VefVGcqNsAGx3bz4Bw&usg=AFQjCNHo7XlNYEPm3\_1l5Pab89PpiAEd0A&sig2=PbsvDlmFiF7gxeW8ipSUKQ&bvm=bv.99028883,d.bGg&cad=rja*] del 18 Dicembre 2014, le percentuali sulla presenza della tecnologia all’interno delle famiglie italiane sono tutte in crescita, come la presenza di una connessione a Internet salita 64% o di una connessione a banda larga salita a quasi il 63%, ma dato più significativo per noi è stato quello dei telefoni cellulari, che sono presenti in più del 93% delle famiglie italiane. L’uso di Internet tramite smartphone è cresciuto dal 20.8% al 28% e ciò sta a indicare un rinnovamento anche nei dispositivi stessi; le persone acquistano dispositivi più recenti che permettano loro di navigare più facilmente su Internet. Sempre dall’analisi ISTAT è emerso che il 22.4% delle persone che navigano su Internet dai 14 anni in su lo ha fatto tramite un computer, mentre il 35.4% degli utenti tramite cellulare o smartphone e solo il 6% da altri dispositivi mobili. L’utilizzo dello smartphone è la tipologia di dispositivi più utilizzata per l’accesso a Internet. Per questo motivo abbiamo formulato l’ipotesi che ogni persona possieda uno smartphone e che possa essere rappresentata da esso nella nostra rete. Per questi motivi abbiamo scelto come strumento di comunicazione la tecnologia BLE.

Riassumiamo brevemente la macchina a stati che rappresenta il funzionamento del Link Layer del BLE. Essa è composta di cinque stati:

* Stato di **Standby**: raggiungibile da tutti gli altri stati. In questo stato non è possibile effettuare nessuna trasmissione di pacchetti né riceverne.
* Stato di **Advertising**: un dispositivo in questo stato si chiama advertiser e può inviare pacchetti di advertiser. Resta in ascolto per pacchetti in risposta ai suoi pacchetti di advertising.
* Stato di **Scanning**: un dispositivo in questo stato si chiama scannar e ricompre il ruolo di osservatore. Rimane in ascolto per pacchetti di advertising.
* Stato di **Initiating**: un dispositivo in questo stato si chiama initiator e rimane in ascolto di pacchetti di advertising. Risponderà a quei pacchetti di advertising ai quali è interessato, con l’intenzione di aprire una connessione.
* Stato di **Connection**: lo stato di Connection può essere raggiunto sia dallo stato di Advertising sia dallo stato di Initiating. Indica che il dispositivo sta tentando di connettersi o è connesso con un altro dispositivo. Lo stato Connection si suddivide in due ruoli: Slave e Master.
  + ***Slave***: il ruolo di Slave è affidato a chi arriva dallo stato di Advertising. Utilizza i parametri di connessione decisi dal Master.
  + ***Master***: il ruolo di Master è affidato a chi arriva dallo stato di Initiator. Un dispositivo Master decide i parametri della connessione, che invierà allo Slave.

Nell’implementazione della nostra soluzione abbiamo cercato di mappare uno a uno gli stati del BLE con gli stati dell’algoritmo di gossip scelto, che descriveremo nella sezione successiva.

In figura 4.4 è riportato la macchina a stati del Link Layer.



*Figura 4.4: Bluetooth LE macchina a stati del Link Layer*

## Soluzione: Algoritmo Dynamic Fanout

* Il nostro algoritmo è un’estensione di quello già presentato nel cap2
* Richiamare le caratteristiche del FF
* Spiegare lo stesso i parametri introdotti

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

Dopo una profonda analisi degli algoritmi di gossip presenti in letteratura e della loro classificazione, abbiamo deciso di basare la nostra soluzione su algoritmi di tipo rumor mongering, in altre parole algoritmi che hanno come obiettivo la diffusione di un “rumore” o pettegolezzo, nel nostro caso un’informazione. Nella Sezione 2.4 abbiamo presentato diversi algoritmi di gossip e i diversi aspetti che li caratterizzano. Poiché il nostro studio concerne una rete P2P, abbiamo scelto di adottare e studiare algoritmi di gossip adatti a reti di questo tipo. La nostra scelta è stata di basare la nostra soluzione sull’algoritmo del Fixed Fanout ed estenderlo con l’aggiunta di parametri dinamici, allo scopo di dare dinamicità all’algoritmo e capacità di adattamento ai cambiamenti dell’ambiente esterno e dello stato interno dei dispositivi stessi. Con cambiamenti dell’ambiente estero intendiamo le variazioni della rete nel raggio ρ d’azione del nodo, quindi il variare del numero di nodi che esso percepisce, mentre con variazioni interne al dispositivo intendiamo le variazioni del livello della batteria del dispositivo stesso. Questo perché abbiamo cercato di progettare i parametri in modo che abbiano sempre un valore che possa garantire prestazioni accettabili e che calibri il carico di lavoro del dispositivo in relazione al livello energetico che il dispositivo ha. Se il livello di energia è elevato, l’algoritmo si permette di assegnare al dispositivo un carico di lavoro maggiore se necessario, mentre se il livello di energia è medio basso, l’algoritmo comincia a calibrare i parametri in modo da trovare un compromesso di carico di lavoro ed efficienza in modo da non gravare molto sulla restante poca autonomia del dispositivo. Dato che il sistema non occupa totalmente le risorse di calcolo del dispositivo, non vogliamo privare l’utente degli eventuali altri servizi presenti sul suo smartphone che possano servirgli. Per questo motivo l’algoritmo, dopo che la batteria scende sotto una certa soglia, di non degradare eccessivamente l’autonomia del dispositivo, anche sotto improvvisi carici di lavoro dovuti a rapidi cambiamenti della rete.

Brevemente ricapitoliamo le caratteristiche dell’algoritmo di Fixed Fanout. E’ un algoritmo di gossip di tipo SIR di rumor mongering, con criterio di terminazione di tipo “counter”. Questo significa che nel momento di inviare una nuova informazione, l’algoritmo sceglie un nodo casuale dall’insieme dei suoi vicini e tenta di trasferirgli l’informazione. Se il trasferimento va a buon fine, incrementa un contatore di uno. Ripete quest’operazione fino a quanto il contatore raggiunge un valore limite denominato *fixed fanout*. Il sistema esegue esattamente *ff* trasferimenti. Non vi è nessun elemento di controllo su cambiamenti della rete, né vi sono elementi in grado di variare il valore di *ff*.

La nostra estensione riguarda principalmente due parametri che hanno lo scopo di sostituire le componenti statiche dell’algoritmo originale con componenti dinamiche. La nostra soluzione rimane un algoritmo di tipo SIR (Suscettible-Infected-Removed), ma con controlli in più finalizzati alla riduzione dello spreco energetico. Nell’implementazione del nostro algoritmo, abbiamo associato a ogni stato dell’automa del Link Layer del BLE uno stato del paradigma SIR.

* Suscettibile 🡨🡪 Initiating
* Contagiato 🡨🡪 Advertising
* Rimosso 🡨🡪 Standby/Scanning

Questo è l’accoppiamento generale, relativo alla generica informazione. All’atto pratico, si aggiunge allo stato di Rimosso anche lo stato di Initiating, perché quando un dispositivo perde interesse a diffondere un’informazione, esso può ritornare in stato di Initiating alla ricerca o in attesa di nuove informazioni da condividere.

I parametri considerati sono elementi composti dal due diversi criteri di terminazione: uno di tipo *counter* e uno di tipo *blind*. I parametri in questione sono:

* ***Dynamic Fanout (DF)***
* ***Advertising Limit (AL)***

Il Dynamic Fanout ha la stessa funzione del precedente *fixed fanout*, rappresentare il limite di trasmissioni che il dispositivo può compiere. Nel nostro algoritmo però, il DF è calcolato dinamicamente e tenuto aggiornato in maniera periodica.

L’Advertising Limit è un nuovo parametro di terminazione introdotto e progettato allo scopo di segnalare e far capire al dispositivo, quale sia la situazione della rete attorno a lui in termini di contagio. Serve a far capire al nodo quando i nodi attorno a lui non sono più interessati alla sua informazione. Anche l’AL è aggiornato periodicamente.

### Dynamic Fanout

Il Dynamic Fanout ha la stessa funzione del precedente *fixed fanout*, rappresentare il limite di trasmissioni che il dispositivo può compiere. Nel nostro algoritmo però, il DF è calcolato dinamicamente e tenuto aggiornato in maniera periodica. Nell’algoritmo del FF, il dispositivo avrebbe continuato a provare a trasmettere l’informazione finché non avrebbe raggiunto il limite impostato di *ff*. Nella nostra soluzione invece il dispositivo tenta di trasmettere fino a DF volte, ma non vi è obbligato perché vi sono controlli in grado di stabilire quando fermare la diffusione perché il sistema ha rilevato inutile continuare. Questi controlli coinvolgono l’Advertising Limit. Il DF è composto di due fattori, uno che tiene conto del livello della batteria (parte *Blind*) e uno che tiene conto dello stato della rete attorno al dispositivo quindi il numero di nodi percepiti dal dispositivo (parte che determina il limite per il *Counter*). Il componente che tiene conto della batteria rappresenta un fattore di partizionamento che sarà usato per scegliere il limite di trasmissioni secondo quanti nodi il dispositivo percepisce. Ad esempio, se il dispositivo ha la batteria totalmente carica, il fattore di partizionamento sarà del 50%, mentre se la batteria è sotto la metà sarà ad esempio del 20%. Nel capitolo successivo, esporremo più nel dettaglio come questo parametro è calcolato e com’è caratterizzato l’andamento dinamico.

### Advertising Limit

L’Advertising Limit è un nuovo parametro di terminazione, introdotto allo scopo di segnalare e far capire al dispositivo qual è la situazione della rete attorno a lui in termini di contagio. Questo parametro rappresenta il limite di pacchetti di advertising andati a vuoto consecutivamente. Quando un nodo riceve una nuova informazione da condividere, esso inizia le trasmissioni e comincia a inviare l’informazione a un nodo alla volta. Al termine di ogni trasmissione il nodo inizierà a inviare pacchetti di advertising alla ricerca di nuovi nodi ancora non a conoscenza dell’informazione, se il limite DF non è ancora stato raggiunto. Il sistema, tramite timeout, capisce quando un pacchetto di advertising va a vuoto e nel caso ne invia un altro. Allo stesso tempo conteggia il numero di pacchetti di advertising andati a vuoto consecutivamente. Se questo conteggio raggiunte l’AL, il nodo passa in stato di Rimosso o di Initiating per le motivazioni sopra dette. Lo scopo principale di questo parametro è di intervenire la dove il controllo DF è cieco, cioè nel controllo se qualcuno cui il dispositivo può collegarsi è ancora realmente interessato a quello che il dispositivo ha da trasmettere. Si potrebbe pensare che un eventuale caso di non stallo prima di raggiungere il DF sia un errore di progettazione, ma in realtà nelle nostre ipotesi di lavoro i nodi della rete rappresentano dispositivi mobili di persone e le persone si muovono e si spostano. Potrebbe accadere che la rete cambi proprio dopo aver aggiornato i parametri e quindi un dispositivo si ritroverebbe con un DF non coerente con il reale stato della rete, fino al prossimo aggiornamento. In queste situazioni l’AL entra in gioco e impedisce al dispositivo di sprecare energie se il limite di advertising è raggiunto.

Come il DF anche l’AL è composto di un fattore riguardante il livello della batteria e di un fattore concernente lo stato della rete circostante il nodo. Come sarà poi spiegato nel capitolo successivo, il fattore riguardante la batteria è stato tenuto neutro, poiché l’energia richiesta per fare advertising è molto esigua e quindi abbiamo deciso di non inserire un fattore di partizionamento anche sull’AL perché la sola funzione inserita per il fattore relativo ai nodi percepiti è sufficientemente restrittiva.

## Simulazione

* Richiamo sui simulatori presentati nel capitolo 2
* Su quale simulatore è ricaduta la scelta
* Le caratteristiche principali del simulatore scelto
* Le motivazioni della scelta fatta, o i vantaggi che esso fornisce

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

Durante la fase di studio e progettazione iniziale, abbiamo analizzato diversi sistemi e piattaforme di simulazione in grado di simulare reti come quella da noi studiata. Si faccia riferimento alla Sezione 2.5 per la presentazione generale. La nostra scelta è ricaduta su OMNeT++, un framework e libreria di simulazione basata sul linguaggio C++, estendibile e modulare. OMNeT++ è un software di simulazione molto diffuso sia nel settore commerciale sia nel settore scientifico per la simulazione di reti e protocolli di trasmissione. OMNeT++ è un software che offre un editor di sviluppo basato su Eclipse Sono disponibili una gran varietà di strutture base come reti wired e wireless ed è possibile aggiungere estensioni che permettono di ampliare la gamma di reti supportate. INET è una delle estensioni più corpose del framework e contiene una grossa quantità di reti e protocolli delle più diffuse strutture di rete utilizzate.

Lo sviluppo anche in ambito commerciale ha reso questo strumento molto ricco di funzionalità e strumenti e anche di componenti di simulazione. OMNeT++ è fornito di un motore di simulazione *event based* ma come spiegato anche nella Sezione 2.5, è in grado di gestire anche azioni cicliche. Grazie al suo tool grafico Tkenv, è possibile visualizzare con animazioni la rete e i pacchetti in movimento in essa; vi è anche la possibilità di modificare alcuni aspetti visivi su Tkenv, tramite apposite istruzioni, per avere a runtime maggior espressività e potenzialità della simulazione stessa. OMNeT++ permette anche un’ottima gestione della parte di layout della rete tramite il suo linguaggio di alto livello chiamato Network Definition (NED), che permette di esprimere la struttura della rete in modo semplice e permette anche l’utilizzo di funzioni di casualità, così da poter creare una rete casuale. Il generatore di numeri casuali, per permettere di poter ripetere una simulazione nelle stesse condizioni, utilizza sempre gli stessi *seed*. Ciò significa che la disposizione causale che si ottiene eseguendo la simulazione, sarà uguale alla disposizione ottenuta eseguendo nuovamente la simulazione da capo una volta terminata la prima. Per ottenere diverse strutture causali è necessario dire al simulatore di fare più simulazioni data la configurazione. OMNeT++ poi, necessità l’implementazione di almeno un file di configurazione nel quale specificare tutte le configurazioni di simulazione e i valori dei parametri che la simulazione richiede. Anche in questo caso è possibile utilizzare funzioni casuali messe a disposizione dal sistema per assegnare valori ai parametri. Infine OMNeT++ ha già un sistema interno per la raccolta dati e l’analisi di statistiche fatte sui dati raccolti. Permette anche di visualizzare i dati raccolti e le eventuali statistiche su grafici. Inoltre permette anche di eseguire manipolazioni sui dati raccolti, come la possibilità di raggruppare dati, eseguire operazioni di media o varianza e tante altre; è possibile inoltre per l’utente specificare particolari operazioni tramite l’apposita sintassi del simulatore. Nel Capitolo 5 sono spiegati, più in dettaglio, gli aspetti riguardanti l’utilizzo dei file di Network Definition, dei file di configurazione e delle loro caratteristiche.