# Capitolo 3 – Impostazione del problema di ricerca

## Analisi dati e studio di fattibilità

La nostra ricerca aveva lo scopo di trovare una possibile soluzione al problema della mancanza di reti di comunicazioni, quali reti telefoniche e rete dati, in particolari scenari che direttamente o indirettamente possono portare alla mancanza di tali reti di comunicazione. Situazioni di tale tipo possono essere scenari di emergenza causati da forti e avverse condizioni meteorologiche e/o ambientali, che facilmente possono indirettamente tali reti, intaccando o danneggiando l’infrastruttura adibita alle telecomunicazioni interrompendone il funzionamento. La nostra idea era quella di poter emulare il processo di diffusione informazioni che si fa in scenari di tale genere, ovvero il passaparola da persona a persona. Ci siamo quindi concentrati sul capire cosa avremmo potuto sfruttare tra i dispositivi ancora funzionanti in tali situazioni; nella società odierna uno dei dispositivi che tutti o quasi tutti hanno è il cellulare e per la precisione uno smartphone e il sistema Bluetooth equipaggiato su tali dispositivi, ricalca molto bene quel comportamento peer-to-peer del passaparola. Il nostro obiettivo quindi era quello di capire se fosse possibile implementare un sistema di diffusione informazioni peer-to-peer sfruttando questo sistema di trasmissione.

La prima problematica che abbiamo affrontato è stata quella di capire quale fosse il consumo per una trasmissione tramite Bluetooth, quindi abbiamo fatto ricerche sulle caratteristiche di consumo energico da parte del Bluetooth 4.0. Questa tecnologia viene definita anche LE, in quanto rispetto alle sue versioni precedenti, ha consumi energetici molto bassi e latenze inferiori. Dalle ricerche fatte (Bluetooth [*cit?*], da Wikipedia [*cit?*] e la tesi della tipa ???), abbiamo trovati questi valori di consumo per il Bluetooth 4.0 LE

|  |  |  |
| --- | --- | --- |
| Potenza massima all’output | Potenza mnima all’output | Distanza |
| 10 mW (10dBm) | 0.01 mW (-20dBm) | ~ 50 m |

*Tabella 3.2: dati consumo energetico Bluetooth 4.0*

Dopo di che abbiamo voluto capire quanto la trasmissione di un pacchetto di grandezza non pensata originariamente per questa tecnologia, potesse influire sul consumo energetico richiesto. Quello che abbiamo ottenuto è stato che, per come è implementato lo standard Bluetooth LE, il consumo energetico richiesto è linearmente dipendente rispetto la grandezza del pacchetto.

Un’altra problematica affrontata è stata quella che affligge quasi ogni smartphone in commercio: la durata della batteria. Nell’ipotesi di trovarsi in situazioni come sopra citate, avere il più a lungo possibile un dispositivo che possa garantire la ricezione di messaggi è importante. A priori abbiamo fatto uno studio di fattibilità, nel quale abbiamo cercato di capire quanto l’invio di un singolo messaggio tramite Bluetooth potesse influire sulla durata della batteria di uno smartphone e quanta autonomia, in termini di trasmissione e/o ricezione messaggi, potesse avere il dispositivo mobile. Per questo abbiamo fatto una ricerca per i più comuni smartphone in commercio, guardando sui siti delle case costruttrici [*cit?*]per cercare i valori delle batterie e sui principali forum di testing [*cit?*] per i valori di consumo energetico medio sotto i diversi profili d’uso dello smartphone: ad esempio il consumo medio in standby e il consumo medio sotto carico. Nella tabella 3.1 vengono riportati i principali valori trovati.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Smartphone | Bluetooth | Batteria | | Consumi | |
| capacità [mAh] | capacità [Wh] | avg idle [w] | avg load [w] |
| Samsung Galaxy S5 | 4.0 | 2800 | 10.78 | 0.5 | 3.1 |
| Samsung Galaxy S4 | 4.0 | 2600 | 9.88 | 0.8 | 3.2 |
| Samsung Galaxy S3 | 4.0 | 2100 | 7.98 | 1.4 | 3.3 |
| LG G3 | 4.0 | 3000 | 11.4 | 1.8 | 4.4 |
| LG G4 | 4.0 | 3000 | 11.4 | 1.1 | 3.8 |
| iPhone 6 plus | 4.0 | 2915 | 11.1 | 2.1 | 3.5 |
| iPhone 6 | 4.0 | 1810 | 6.91 | 1.5 | 3.1 |
| iPhone 5 | 4.0 | 1440 | 5345 | 1.2 | 2.2 |
| GoogleNexus 5 | 4.0 | 2300 | 8 | 0.8 | 4.3 |
| Google Nexus 6 | 4.1 | 3220 | 12.2 | 1.5 | 4.8 |
| Nokia Lumnia 930 | 4.0 | 2420 | 9.2 | 1.1 | 3.8 |
| Nokia Lumnia 1020 | 4.0 | 2000 | 7.6 | 2.2 | 3.4 |

*Tabella 3.1: dati batteria smartphone*

Dalle nostre analisi è risultato che grazie ai consumi energetici molto bassi che la tecnologia Bluetooth 4.0 LE offre, essa ha una piccola incidenza su quello che è il consumo energetico totale dello smartphone. Di fatto abbiamo un degrado molto basso della durata dei dispositivi rispetto ai loro standard e un numero di trasmissioni possibili, nell’arco di operatività, molto alto.

## Algoritmo di Propagazione

Il passo successivo è stato quello di trovare un algoritmo di propagazione dell’informazione e abbiamo pensato di utilizzare un algoritmo di gossip, chiamati anche algoritmi epidemici, in quanto volevamo che l’informazione da distribuire potesse raggiungere quanti più nodi possibile esattamente come può fare un epidemia. Abbiamo scelto di applicare un algoritmo di tipo “rumor mongering”, ovvero un algoritmo pensato per simulare la diffusione dei pettegolezzi tra le persone. Gli algoritmi di tipo rumor mongering sono algoritmi di tipo SIR, Suscettible-Infected-Removed, ovvero algoritmi in cui inizialmente tutti i nodi sono suscettibili all’infezione, cioè a ricevere la notizia, poi una volta ricevuta la notizia diventano infetti e a loro volta inizieranno a diffondere anch’essi la notizia; infine in base a certi criteri un nodo infetto smette di diffondere la notizia e diventa rimosso dal processo di diffusione.

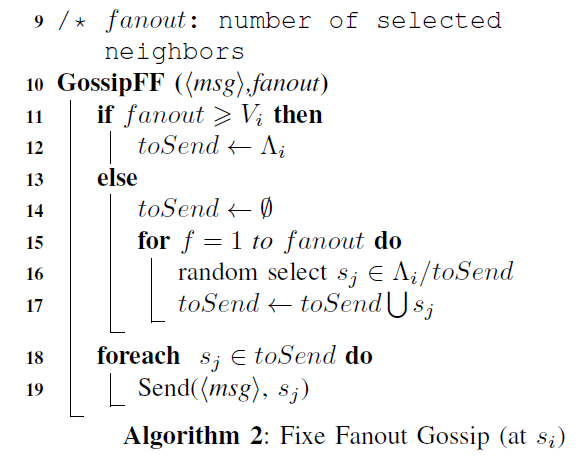
I criteri ti terminazione sono:

* ***Blind***: un nodo infetto decide di interrompere la diffusione in base al suo stato interno.
* ***Feedback***: un nodo contattato risponde dicendo se giù conosce oppure no l’informazione. L’infetto, in base alla risposta decide.
* ***Coin***: il nodo infetto si fermerà con una probabilità 1/k, dove k è il numero di nodi contattati.
* ***Counter***: il nodo infetto incrementa un contatore per ogni informazione inviata con successo. Si ferma quando il contatore supera una soglia prefissata.

Vi sono anche tre tipologie di approcci alla diffusione:

* ***Push***: il nodo infetto invia attivamente a nodi non infetti l’informazione.
* ***Pull***: il nodo infetto conserva l’informazione, mentre devono essere i nodi suscettibili a chiedere al nodo infetto se ci sono nuove informazioni e nel caso farsele mandare.
* ***Push&Pull***: è l’unione delle precedenti, ovvero il nodo infetto manda attivamente ai nodi adiacenti un avviso che lui ha una nuova informazione e i nodi interessati contatteranno direttamente il nodo infetto per farsi mandare la nuova informazione.

Tra i vari algoritmi di gossip abbiamo scelto di usare una versione leggermente modificata dell’algoritmo di Fixed Fanout. Questo algoritmo prevede che ogni nodo, una volta infettato, si prodighi a infettare a sua volta un certo numero di nodi ai quali è connesso e questo numero prefissato si chiama fixed fanout, caratteristica che da il nome all’algoritmo stesso. Questo algoritmo utilizza come strategia di stop, il contatore di invii e il fixed fanout è il suo limite. In figura 3.1 viene riportato lo pseudocodice che descrive tale algoritmo.



*Figura 3‑1: pseudocodice algoritmo Fixed Fanout*

La nostra personalizzazione di questo algoritmo consiste nel