POLITECNICO DI MILANO

Facoltà di Ingegneria

Corso di Laurea in Ingegneria delle Telecomunicazioni

Sviluppo di un algoritmo di ricerca di vicini per reti radio ad hoc

Relatore: Ing. Matteo Cesana

Tesi di Laurea di:

Nicolo' Tonato

Matr. 766744

Anno Accademico 2011-2012

Abstract

The rapid deployment of millions of mobile sensors and smart-phones has resulted in a demand for opportunistic encounter-based networking to support mobile social networking applications and proximity-based gaming.

However, the success of these emerging networks is limited by the lack of effective and energy efficient neighbor discovery protocols. While probabilistic approaches perform well for the average case, they exhibit long tails resulting in high upper bounds on neighbor discovery time. Recent deterministic protocols, which allow nodes to wake up at specific time slots according to a particular pattern, improve on the worst case bound, but do so by sacrificing average case performance.

Searchlight, which is a highly effective asynchronous discovery protocol, has been developed in response to these limitations.

At first, Searchlight was designed to address discovery issues in networks in which there was only one device with one frequency available only. This quickly appeared to be a limitation. Thus, multi-frequency discovery protocols were introduced in order to provide incoming devices with the ability to connect to all other devices in the network. The frequency of each device in those networks are either assigned randomly either assigned according to a distribution function.

To conclude, this thesis provides the reader with an extension of the Searchlight protocol by overcoming the limitation identified, described earlier. Numerical simulations, performed with a VB.NET Console Application, help to evaluate the performance of the protocol when network parameters are changing.

Indice

Introduzione

Nel corso degli ultimi anni, l'utilizzo di sensori mobili e di smart phones è in una sensibile fase di crescita . Le reti d'interesse devono essere in grado di supportare le applicazioni di social networking mobile e le applicazioni di giochi. I dispositivi mobili stanno trasfromando queste reti. Benchè la popolarità di queste applicazioni è in grande crescita, i servizi di rete necessari al raggiungimento di prestazioni efficaci ed energeticamente efficenti non sono ancora stati sviluppati. Il successo di queste applicazioni dipende essenzialmente dalla disponibilità di questi servizi a fornire informazioni sugli utenti del vicinato.

Un protocollo di discovery asincrono altamente efficace è stato progettato, esso si chiama Searchlight. Questo protocollo si basa essenzialmente su tre idee , la prima è che esso sfrutta l'offset costante fra slots attivi periodici in modo tale da poter progettare un semplice schema di sondaggio per assicurare la scoperta. La seconda idea è che Searchlight permette agli slots attivi di coprire una piu' larga durata temporale rispetto agli altri slots, il che riduce in maniera molto importante il tempo totale di attività del protocollo. In ultima analisi, Searchlight ha l'opzione di poter utilizzare delle tecniche probabilistiche che permettono di ridurre sensibilmente la latenza della scoperta nel caso medio, e nella situazione in cui tutti i nodi della rete lavorino al medesimo duty-cycle.

In realtà, la situazione in cui si troverà un dispositivo entrante all'interno di una rete ad hoc, sarà quella di dover scoprire un numero a lui non noto a priori di dispositivi ed ognuno di essi che è in ascolto su una frequenza da lui non conosciuta. Questa tesi fornisce un metodo per estendere il protocollo Searchlight nella sua implementazione originale per poter gestire una scoperta totale, oppure anche parziale della rete.

Lo studio effettuato è organizzato come segue :

- Nel primo capitolo si darà una definizone accurata delle reti ad hoc, dei vantaggi di queste reti, degli svantaggi e come essi vengono gestiti e le applicazioni d'interesse per le MANET. Inoltre, Si effettuerà un breve studio di come i servizi centralizzati non siano adeguati per le applicazioni d'interesse. Infine verrà fornita una breve panoramica dei protocolli di discovery energeticamente efficienti attualmente esistenti.

- Nel secondo capitolo sarà discusso in maniera approfondita il funzionamento del protocollo Searchlight nella sua implementazione originale, di come vengono gestite l'asincronia e l'asimmetria e dei vantaggi che comporta l'introduzione di una componente probabilistica all'interno della politica di sondaggio. Infine, si effettuerà un confronto fra il protocollo Searchlight ed i protocolli definiti nel primo capitolo.

-Nel terzo capitolo verrà spiegato come avviene il sondaggio multi frequenza che deve effettuare il dispositivo che vuole accedere alla rete per scoprire un utente già presente in essa. Sarà poi studiato il valore atteso della latenza affinchè la scoperta di questi due dispositivi avvenga con successo, e tale valore verrà poi confrontato con i risultati ottenuti da diverse simulazioni.

- Nel quarto capitolo si introdurrà il problema reale al quale dovrà fare fronte il dispositivo che entra nella rete, ovvero la situazione in cui deve scoprire un numero a lui non noto di dispositivi presenti nel suo interno, e dove ognuno di essi lavora ad una frequenza diversa. Le frequenze utilizzate dai dispostivi nel sistema possono essere distribuite in maniera puramente casuale seguendo un distribuzione uniforme. In questo scenario, si andrà a valutare il tempo medio per la scoperta della totalità della rete, e la latenza media che è necessaria al fine di scoprire una porzione degli utenti che la compongono. Infine, si valuteranno tre diverse politiche di sondaggio per determinare la frequenza utilizzata dai dispositivi presenti nella rete e si confronteranno le latenze medie fornite da ognuna di queste politiche al variare di particolari funzioni di distribuzione.

CAPITOLO 1 - Discovery di dispositivi mobili in reti radio ad hoc

Il mondo dei social Network è in corso di trasformazione a causa dei dispositivi mobili. La tipologia di rete d'interesse per le attività che vogliono svolgere gli utenti di questi servizi, è una rete ad-hoc. In telecomunicazioni una rete ad-hoc mobile (Mobile Ad-hoc NETwork) è definita come un sistema autonomo di terminali mobili connessi mediante collegamenti wireless di tipo ad-hoc. La definizione di rete ad hoc fornita dall'IETF (Internet Engineering Task Force) è la seguente : "Una MANET è un sistema autonomo di router mobili e dei loro host associati, connessi con collegamenti di tipo wireless che sono uniti formando un grafo di forma arbitraria. Tali router sono liberi di muoversi casualmente e di auto organizzarsi arbitrariamente, sebbene la topologia wireless vari rapidamente ed in modo imprevedibile. Tale rete può operare da sola oppure essere connessa alla rete Internet.". Tutti i nodi del sistema collaborano con lo scopo di instradare i pacchetti nel modo corretto secondo la modalità di forwarding di tipo multihop. Le reti ad-hoc vengono utilizzate in ambienti molto dinamici. Con l'ausilio di questa tipologia di rete, gli utenti possono accedere ad applicazioni multimediali.

Dove le prime applicazioni richiedevano agli utenti di accedere tramite server centralizzati per trovare gli altri utenti e coordinarsi a loro, come ad esempio Foursquare, le nuove applicazioni si stanno costruendo intorno alla capacità di effetuare comunicazioni locali fra tutti gli smartphones. Uno dei settori d'interesse che è piu in crescita è la comunità di gioco mobile, dove i diversi utenti cercano e giocano a questi giochi con altri utenti all'interno del vicinato della loro rete wireless. Benchè la popolarità di queste applicazioni proximity-based è aumentata, i servizi di rete necessari che consentono delle prestazioni efficaci ed energeticamente efficienti non sono ancora stati raggiunti. In brevi termini, il successo di tali applicazioni è strettamente dipendente dalla disponibilità dei servizi a fornire informazioni relative agli utenti presenti all'interno del vicinato, o in maniera piu specifica, ai dispositivi d'utente vicini. Si discuterà delle principali caratteritiche e problematiche relative all'utilizzo di servizi centralizzati o di servizi locali per la gestione delle applicazioni proximity-based.

Inoltre, si effettuerà all'interno di questo capitolo una breve panoramica dei protocolli di discovery attualmente esistenti destinati alle applicazioni proximity-based. In particolare si andrà a discutere di due categorie di protocolli distinte, ovvero i protocolli che utilizzano un'approccio probabilistico quali i protocolli del compleanno, ed i protocolli deterministici. Per quanto riguarda questa seconda categoria di protocolli, ci si soffermerà su due protocolli ben noti, il protocollo DISCO ed il protocollo U-Connect.

Lo studio di queste neighbor discovery esistenti ci porterà a sottolineare che per l'utilizzo attuale dei dispostivi mobili viene evidenziata l'occorrenza di un protocollo di discovery energeticamente efficiente, che gestisca in maniera coerente l'asimmetria, e che fornisca una latenza media della scoperta dei vicini relativamente bassa ed un bound massimo per questa latenza ragionevolmente contenuto.

1.1. Le reti ad-hoc

Una rete ad-hoc è una rete composta da nodi mobili ad alta potenza equipaggiati di un'interfaccia wireless per la comunicazione. Le comunicazioni tra gli attori della rete, vale a dire i nodi che la compongono, è una comunicazione multi-hop. Ogni nodo puo' essere connesso direttamente con un sotto insieme di nodi mobili, tale sotto insieme rappresenta il *vicinato* del nodo. Il range di comunicazione di un nodo varia a seconda dello spostamento fisico che esso esegue. Il traffico dei dati è solitamente dipendente dall'applicazione d'interesse, dai requisiti di larghezza della banda, dai vincoli di tempestività, di affidabilità e di sicurezza.

I principali vantaggi di un MANET, Mobile Ad-hoc NETwork, sono una rapida distribuzione, molto utile per le applicazioni di sicurezza, l'indipendenza delle infrastrutture, e la flessibilità che rende l'aggiunta, la soppressione o la riallocazione dei nodi gestita automaticamente e inoltre fornisce la possibilità di usufruire di nuove applicazione dove il numero di nodi è dinamico e impredicibile.

I problemi da affrontare per le reti ad-hoc sono molteplici. Ad esempio, a livello fisico si gestisce il range di comunicazione, la simmetria o ancora il controllo della potenza. A livelllo MAC, Medium Access Control, è presente il problema dell'hidden terminal, ovvero un tipico problema del networking wireless che avviene quando un nodo risulta essere visibile da un'altro nodo X, ma non per gli altri nodi che comunicano con il suddetto nodo X. Sempre a livello MAC, si devono gestire i collegamenti asimmetrici e l'efficienza energetica. A livello di rete invece, si hanno i diversi problemi di gestione dell'instradamento e dell'indirizzamento dei pacchetti. Infine, a livello di trasporto viene gestita la perdita di pacchetti.

Gli obiettivi dello sviluppo di tali reti sono quelli di sviluppare delle soluzioni che possono essere utilizzate all'interno di tutte le reti ad-hoc, soddisfare i vari vincoli dei livelli di applicazione, l'adattamento alle variazioni delle proprietà topologiche, l'integrazione di varie tipologie di nodi della MANET e permettere di interagire con delle infrastrutture fisse.

Se ci interessiamo alla discovery di dispositivi mobili all'interno di una rete ad-hoc, è interessante definire due diverse tipologie di protocolli utilizzati per gestire l'instradamento di pacchetti. I protocolli *reattivi* che determinano i cammini da utilizzare on-demand, ovvero solo qualora avviene un'esplicita richiesta, questa famiglia di

protocolli portano ad una latenza mediamente piu alta in quanto un'instradamento fra due nodi X ed Y viene determinato solamente all'istante di tempo in cui il nodo X cerca il nodo Y. In opposizione, si definisce la famiglia dei protocolli *proattivi* che mantengono i cammini determinati in precedenza nel tempo attivi a prescindere dalle condizioni imposte dal traffico. Quesi protocolli hanno una latenza della scoperta piu bassa rispetto ai protocolli reattivi in quanto gli instradamenti sono mantenuti nel tempo. L'utilizzo di protocolli proattivi porta inevitabilmente a degli overhead significativi.

Alcune applicazioni d'interesse per le reti ad-hoc, sono ad esempio le applicazioni di emergenza e di sicurezza, o le applicazioni civili (reti veicolari , e PAN , Personal Area Network) , o ancora le applicazioni militari.

1.2. Servizi centralizzati e applicazioni proximity-based

Sebbene i servizi centralizzati possono fornire gran parte del supporto di cui necessitano le applicazioni proximity-based, i servizi centralizzati presentano pero' diversi problemi rilevanti.

In primo luogo, se il servizio è inattivo, gli utenti sono isolati, anche se possono effetivamente vedere la presenza di altri dispositivi di utenti che si trovano all'interno del loro vicinato. Per di piu, si possono verificare entrambi i casi in cui un servizio di localizzazione centralizzato puo' inviare informazioni locali a tutti gli utenti, oppure gli utenti stessi possono ricevere informazioni quando lo desiderano, questo fà si che avviene un numero eccessivo di aggiornamenti ed in maniera pressochè continua. Inotre, molte applicazioni hanno un proprio servizio di localizzazione al quale l'utente deve essere registrato, il che forza gli utenti a registrarsi con molte applicazioni. Ancora, molti utenti sono restii ad esporre le proprie informazioni locali ad un servizio centralizzato, ma possono essere disposti a lasciare che gli utenti presenti all'interno del proprio vicinato o che hanno già effettuato la discovery con essi possono vedere il proprio dispositivo e la posizione geografica attuale. Infine, comunicare attraverso un server centrale piuttosto che passare direttamente dagli utenti vicini puo introdurre una latenza alla comunicazione molto alta, limitando cosi l'utilità di applicazioni real-time oppure di applicazioni di giochi. Pertanto, sono necessarie delle soluzioni locali che non confidano nella connessione a servizi e server centralizzati.

1.3. Servizi locali e applicazioni proximity-based

Per raggiungere la totale potenzialità di queste applicazioni proximity-based, tali applicazioni necessitano servizi locali che consentono di trovare i diretti vicini all'interno di un range di comunciazione fra dispositivi. Sebbene la discovery di vicini sia una sfida importante anche in ambienti quali le reti ad hoc e le reti di sensori, in questo scenario si hanno delle aspettative anche sulla stabilità della rete, pertanto le applicazioni d'interesse hanno pretese di connettività e di gestione di eventuali fallimenti quando questi avvengono. Inoltre si hanno delle aspettative di sufficiente densità cosi quando un dispositivo detiene dei dati da trasmettere, un certo numero di vicini sarà attivo al tempo della discovery.

Tuttavia, il mondo dei social networks non ha le stesse pretese di stabilità e di densità dei dispositivi. Gli utenti all'intero di un social network cercano i vicini e decidono in seguito di comunicare o inviare dei dati. Siccome la maggior parte degli smart phones hanno la possibilità di utilizzare sia il Bluetooth che il Wi-Fi, e data l'estrema differenza di energia richiesta per effettuare la discovery fra queste due tecnologie, pressochè tutte le applicazioni proximity-based attualmente presenti limitano la scoperta del vicinato alla radio Bluetooth, che è ovviamente molto meno dispendiosa di energia del Wi-Fi. Pero', limitare i vicini a rientrare all'interno di un range relativo alle comunicazioni tramite Bluethooth, ovviamente limita il raggiungimento e l'efficacia delle applicazioni dei social networks. Esiste quindi la necessità di un protocollo di discovery energeticamente efficiente per delle tecnologie di comunicazione ad alta potenza come appunto il Wi-Fi. L'obiettivo chiave per una discovery dei vicini energeticamente efficiente è quello di ridurre il consumo energetico, consentendo sempre una neighbor discovery efficace, dove per efficienza si intende l'abilità di determinare i vicini insieme ad un'adeguata latenza che occorre per scoprire questi ultimi.

La latenza della discovery ha diversi impatti importanti. All'interno di reti stazionarie, una discovery latency alta puo' portare in lunghi tempi di set up, pero' le operazioni della rete non vengono afflitte. In ambiente mobile invece, il crescere della discovery latency potrebbe causare piccole, ma molto significative, perdite di opportunità di contatti, generando cosi una sensibile riduzione dell'efficacia dell'applicazione che si sta utilizzando. I protocolli di discovery attualmente esistenti, principalmente finalizzati alle reti di sensori, non sono stati in grado di assicurare congiuntamente un tempo medio di discovery basso garantendo sempre un bound massimo di tale latenza contenuto.

1.4. I protocolli di discovery energeticamente efficienti esistenti

La chiave di tutte le applicazioni proximity-based è quella di determinare in maniera costante chi è vicino agli altri, e questo deve avvenire anche se l'apparecchio d'utente non è in corso di utilizzo ed è in tasca dell'utente stesso. Nonostante il fatto che i servizi di

neighbor discovery sono integrati all'interno delle piattaforme smart phone, la principale sfida rimane quella del bilanciamento tra il tempo di vita della batteria e la connettività. In generale, maggiore è il numero di dispositivi che tali applicazioni possono scoprire, migliore sarà il loro funzionamento. Tuttavia, per un dispositivo che funziona tramite una batteria, e quindi con un'energy budget limitato, non risulta pratico ricercare in maniera continua i vicini. Degli approcci esistenti, quali l'approccio di richesta del Bluetooth, sono stati progettati senza confidare una discovery continua. Un'approccio piu realistico è quello di adottare delle tecniche sfruttate dalla communità delle reti di sensori e lasciare l'interfaccia wireless in stato *SLEEP* per gran parte del tempo ed accenderla, portandola ad uno stato *AWAKE*, periodicamente nel tempo per effettuare la discovery.

Il successo di questi schemi basati sul duty-cylce stà nel fatto che si deve assicurare che i tempi di wake up di due dispositivi si sovrappongono temporalmente. Cio' non è difficile da raggiungere quando i clock dei diversi dispositivi possono essere sincronizzati, come per esempio attraverso GPS; che pero' risulta essere una sincronizzazione particolarmente dispendiosa di energia per i sensori mobili e gli smart phones. Questo ha generato il bisogno della progettazione di schemi periodici che assicurino la sovrapposizione entro un bound temporale ragionevole, operando a dei duty-cycles bassi e senza la sincronizzazione dei clock.

Sebbene esistano una molteplicità di protocolli di comunicazione asincroni, quali SMAC o BMAC, la maggior parte di essi assume che tutti i dispositivi abbiano degli schemi di SLEEP simmetrici, ovvero che tutti i nodi lavorino al medesimo duty-cycle. Pero', all'interno di scenari realistici, sia per il social networking mobile che per le reti di sensori, i nodi assumono una variazione dei requisiti energetici e pertanto si hanno dei duty-cycles asimmetrici. Benchè i nodi possono partire con lo stesso duty-cycle, durante l'utilizzo della rete, l'energia disponibile per ogni nodo varierà, portando cosi a duty-cycles asimmetrici. In risposta a questo una nuova classe di protocolli di discovery asincorni ed assimetrici si è evoluta.

Gli algoritmi di discovery dei vicini asincroni lavorano generalmente sulla base di time-slots, dove viene assunto che il tempo viene suddiviso in slot temporali di dimensione uguale e che tutti i nodi utilizzino tale dimensionamento per i diversi slots. Basandosi sul protocollo utilizzato, i diversi nodi decidono di rimanere AWAKE durante certi specifici slots, chiamati *ACTIVE* slots, e di restare spenti durante gli slots rimanenti. Durante uno slot ACTIVE, il nodo in questione puo' inviare o ricevere, oppure effettuare entrambe le azioni simultaneamente, a seconda dei requisiti dell'applicazione in uso. La discovery fra due dispositivi avviene con successo quando uno slot attivo di un dispositivo si sovrappone con uno slot attivo dell'altro device. Per risultare energeticamente efficiente, uno schema di discovery necessita l'utilizzo del numero minimo possibile di slot attivi per effettuare la scoperta dei vicini entro un tempo limite ragionevole. Gli approcci correnti per la neighbor discovery asincrona ed energeticamente efficiente rientrano all'interno di due categorie distinte. Esse sono le classi probabilistiche e deterministiche. I rispettivi punti di forza e le rispettive carenze di tali schemi esistenti possono essere valutati confrontando la loro latenza media e il valore peggiore della latenza, ed inoltre osservando la qualità della loro gestione dell'asimmetria del duty-cycle.

Per le applicazioni opportunistiche entrambe le latenze, media e peggiore, risultano molto importanti. Mentre è chiaro che tali applicazioni traggono beneficio dalla riduzione della

latenza media, molte applicazioni sono progettate per gestire la perdita di contatti e quindi potrebbero sembrare trarre poco vantaggio da una diminuzione della latenza del peggior caso. Tuttavia, le scoperte avvenute con successo sono sempre la chiave che determina l'efficacia di tali applicazioni. Poichè molti incontri possono essere di breve durata temporale, un worst-case bound basso puo' permettere un quantità di discovery avvenute con successo maggiore, e pertanto una maggiore quantità di contatti disponibili.

1.4.1. Approccio probabilistico : Protocolli del compleanno

I protocolli piu' conosciuti all'interno della classe degli approcci probabilistici è la ben nota famiglia dei " protocolli del compleanno " dove si teorizza che i nodi trasmettono e/o ricevono, oppure siano in stato SLEEP, con diverse probabilità. Data la natura probabilistica di questo approccio,tali schemi sono altamente performanti per lo studio del caso medio della latenza e permettono inoltre le operazioni asimmetriche, siccome non è presente alcuna restrizione per quanto riguarda la lunghezza del duty-cycle. Tuttavia, il principale svantaggio del protocollo del compleanno è la sua incapacità di fornire un bound nel peggior caso di latenza della discovery, portando cosi a lunghe code nella probabilità di discovery.

1.4.2. Approccio deterministico

I protocolli deterministici, che possono utilizzare un valore di " quorum " oppure dei numeri primi per determinare la lunghezza del duty-cycle, forniscono delle garanzie sulla latenza della discovery nella peggiore situazione.

1.4.2.1. Protocollo a " quorum "

Per quanto riguarda i protocolli basati sul concetto di " quorum", il tempo viene suddiviso in insiemi di m^2 intervalli contigui. Questi m^2 intervalli vengono organizzati all'interno di un'array bidimensionale di dimensione [m x m] ed ogni host puo' selezionare una riga ed una colonna come intervalli di AWAKE. Questa scelta assicura che poco importa quale riga e quale colonna sono state selezionate, per ogni coppia di nodi esistono almeno due intervalli AWAKE che si sovrappongo.

Nonostante il fatto che il protocollo a " quorum " fornisce un bound ragionevole per la latenza nel caso essa sia la peggiore, esso fornisce prestazioni peggiori nel caso di latenza media rispetto ai protocolli basati su un'approccio probabilistico. Inoltre, dato che m è un parametro globale, il protocollo a " quorum " supporta inizialmente delle operazioni simmetriche. S.Lai, B.Ravindran e H.Cho forniscono un miglioramento a questo protocollo, in modo tale che esso possa gestire i casi asimmetrici quando sono presenti solamente due schedule all'interno della rete. Pero', permettendo l'utilizzo di solamente due diversi duty-cycles, questo protocollo è troppo restrittivo.

1.4.2.2. Protocolli DISCO e U-Connect

Per ovviare a questa limitazione, ci si riferisce ai protocolli deterministici basati sui numeri primi che possono gestire sia le operazioni simmetriche che quelle asimmetriche, fornendo sempre un bound adeguato per il peggior caso della latenza.

Per il protocollo DISCO, ogni singolo nodo sceglie una coppia di numeri primi tali che la somma dei rispetti reciproci sia il piu' vicino possibile al valore desiderato del duty-cycle. I nodi si troveranno quindi in stato AWAKE ad ogni multiplo dei numeri primi scelti. Se un nodo sceglie la coppia di numeri primi [p1 , p2] ed un' altro nodo sceglie la coppia [p3 , p4] , risulta che il bound della latenza per la discovery tra questi due nodi nel peggior caso è pari a : min { (p1 . p3) , (p1 . p4) , (p2 . p3) , (p2 . p4) }, avendo che i due numeri primi della coppia non siano uguali.

Un protocollo deterministico piu' recente che utilizza un'unico numero primo per ogni nodo è il protocollo U-Connect. Invece di svegliarsi solamente uno slot ogni p slots, i nodi si trovano in stato AWAKE ogni $\frac{P+1}{2}$ slots ogni p^2 slots. La worst-case latency per U-Connect è pari a p^2, ovvero un valore similare a quello fornito dal protocollo visto in precendenza, cioè DISCO. Si nota inoltre che U-Connect ha prestazioni migliori nel caso simmetrico rispetto al protocollo DISCO.

Sottolineamo infine che benchè questi protocolli deterministici abbiamo delle buone prestazioni nel caso peggiore di latenza, per quanto riguarda la situazione della latenza media le loro prestazioni sono sensibilmente peggiori rispetto a quelle fornite dal protocollo del compleanno.

1.5. La necessità di un nuovo protocollo

Per raggiungere con successo gli obiettivi di efficienza energetica, asimmetria, ed avere delle latenze medie e massime adeguate, c'è la necessità di progettare un nuovo protocollo di discovery. Una soluzione è presentata all'interno di questo studio. Il protocollo è chiamato Searchlight. Searchlight segue un'approccio deterministico e fornisce un worst-case bound della latenza che è piu' piccolo rispetto a quello fornito dai protocolli esistenti, e questo sia per la situazione simmetrica che per la situazione asimmetrica. Mentre i protocolli deterministici garantiscono la discovery entro un certo bound utilizzando un particolare schema per determinare gli slot AWAKE, questi protocolli trascurano dei sotto-schemi all'interno degli schemi utilizzati. Dove gli slots AWAKE avvengono periodicamente, Searchlight sfrutta il costante offset tra questi slots accesi di qualunque coppia di nodi che utilizzano il medesimo periodo. Per di piu', i protocolli con prestazioni migliori spesso comportano delle scoperte multiple, quando una sola è sufficiente. Searchlight riduce queste discoveries estranee, portando ad una riduzione pari al 50% degli slots attivi all'interno del caso simmetrico e poco meno nello scenario asimmetrico. Infine, Searchlight incorpora delle tecniche casuali che migliorano

sostanzialmente la latenza media della scoperta in tutti i casi simmetrici oppure anche quando solamente alcuni nodi risultano tali.

CAPITOLO 2 - Il protocollo Searchlight con una frequenza disponibile per ogni dispositivo

Lo studio effetuato nel precedente capitolo ha messo a nudo le debolezze dei protocolli di discovery energeticamente efficienti esistenti. Per rispondere a queste limitazioni, è stato progettato Searchlight, un protocollo di discovery dei dispositivi vicini altamente efficace ed energeticamente efficiente basato sul duty-cycle che fornisce migliorie sia nel caso di latenza media che nel caso peggiore della latenza, garantendo un decremento significativo del consumo energetico. Searchlight è progettato attorno l'osservazione che due dispositivi che conservano l'energia seguono degli schemi ben definiti per effettuare la discovery, portando cosi una relazione stabile tra gli schemi di ogni coppia di devices, il chè permette a Searchlight di ridurre in modo drastico il worst-case bound della latenza. Utilizzando ulteriormente una componente radom, ovvero probabilistica, per il protocollo, Searchlight diminuisce fortemente anche la latenza media della discovery. Inoltre, ingrandendo lo spazio temporale di ogni AWAKE slot, Searchlight elimina un significante numero di discoveries ridondanti riducendo cosi il tempo totale di AWAKE, e percio' l'energia totale consumata per effettuare la discovery di oltre il 50%. Infine, Searchlight puo' essere applicato anche a dispositivi con diversi duty-cycles se tutti i duty-cycles scelti sono multipli del piu' piccolo duty-cycle. Siccome i piu' recenti algoritmi di discovery dei vicini basati sul duty-cycle sono destinati ed implementati sui sensor motes, come nel caso del protocollo visto precedentemente U-Connect, questa implementazione fornisce una nuova visione per l'esecuzione di questi protocolli sugli smartphones con interfaccia Wi-Fi in modo tale da supportare le applicazioni di social networking. Detto questo, è importante sottolineare che Searchlight è un protocollo di discovery generale che fornisce delle prestazione sensibilmente migliorate sia nell'ambiente mobile che nell'ambiente dei sensori.

2.1.Definizione del protocollo

Searchlight è un protocollo asincrono di discovery periodico basato sugli slot, dove un periodo consiste in t slots contigui determinati dal duty-cycle desiderato da un dato nodo.

I nodi sono in stato SLEEP durante la maggior durata del tempo, cioè per la gran parte degli slots, e possono scoprire gli altri nodi solamente se essi sono attivi durante il medesimo slot, ed ovviamente se si trovano in range l'uno con l'altro. Searchlight puo' operare in due modalità : la prima è la modalità simmetrica, dove tutti i nodi hanno lo stesso t, e la seconda è la modalità asimmetrica, in cui i nodi possono disporre di t differenti. L'idea sulla quale si basa il protocollo qui' presentato viene dalla semplice affermazione che quando due nodi si svegliano in modo periodico con lo stesso intervallo, la distanza temporale tra questi slots AWAKE rimane sempre la stessa (assumendo trascurabile il drift del clock). Searchlight sfrutta questa costante relazione temporale per fornire un migliore valore peggiore della latenza della discovery rispetto a quello fornito da qualunque protocollo di discovery attualmente esistente. Per di piu', Searchlight migliora il costo energetico, riducendo il numero di slots attivi richiesto per effettuare la scoperta con successo di quasi la metà ; questo avviene dando l'opportunità agli slots attivi di poter effettuare un'overflow all'interno dello slot successivo. Per concludere, Searchlight ha anche l'opzionalità di utlizzare delle tecniche probabilistiche in modo tale da migliorare anche la discovery latency media quando tutti i nodi operano col lo stesso duty-cycle, vale a dire nella situazione sincrona.

Nel seguito del capitolo si andrà a descrivere in dettaglio il protocollo Searchlight quando ogni dispositivo ha a sua disposizione una frequenza di utilizzo. Si spiegherà la progettazione del sondaggio sistematico, l'overflowing degli slots attivi il chè è molto utile per la gestione dell'asincronia, e l'introduzione al sondaggio casuale. Lungo l'analisi che verrà effettuata, si deriverà il bound della latenza nel caso peggiore sia per il caso simmetrico che per quello asimmetrico, e si mostrerà che Searchlight riduce questo valore del 50% nel caso simmetrico.

2.2. Sondaggio sistematico : Searchlight-S

Come detto in precendenza, per il protocollo Searchlight ogni nodo effettua una suddivisione temporale basata sul concetto di slot. In particolare, Searchlight ha due nodi attivi durante un periodo t. Il primo slot attivo è chiamato *Anchor slot*, esso è il primo slot del periodo. Nella casistica simmetrica, siccome la posizione di questo slot ancora è fisso in t ma il tempo di entrata nella rete per i diversi t per diversi nodi varia, gli slots ancora di due nodi si sovrappongono se e solo se la differenza temporale dell'entrata all'interno della rete dei due periodi è inferiore ad un time slot. Per ogni altro caso di offset, sempre sotto la condizione di trascurare il drift dei clock, gli anchor slots non si inconteranno mai, e questo è dovuto al fatto che l'offset rimane costante. In altri termini, la relativa posizione dello slot ancora di un nodo rimane sempre la stessa vista dagli altri nodi ed il suo range è [1 , t-1]. Siccome il maggior numero di anchors slots non si sovrappone con gli altri, Searchlight introduce un secondo slot attivo all'interno del periodo di lunghezza t slot, esso viene chiamato *Probe slot*, che è sistematicamente alla ricerca degli slot ancora

degli altri nodi. E molto importante sottolinare che Anchor slot e Probe slot effettuano esattamente la stessa operazione, la differenza dell'appellativo è usata per esprimere semplicemente il fatto che uno rimane fissato durante ogni periodo, lo slot ancora, mentre l'altro è in costante movimento alla ricerca di slot ancora di altri nodi, questo è il probe slot. Quindi, la discovery risulta effettuata ad ogni tipologia di sovrapposizione; cioè probe-anchor, probe-probe e anchor-anchor. Data la stabilità dell'offset, due nodi non possono avere tale offset maggiore di $\frac{t}{2}$ slots. Pertanto, Searchlight deve sondare solamente la prima metà del periodo costituito da t slots per garantire un'overlap tra un probe slot di un nodo e l'anchor slot di un'altro nodo. Per raggiungere tale obiettivo, la posizione del probe slot è determinata da un contatore che parte dal valore 1, e viene incrementato di un'unità dopo ogni periodo, dopo $\lfloor \frac{t}{2} \rfloor$ il processo termina e riparte nuovamente.

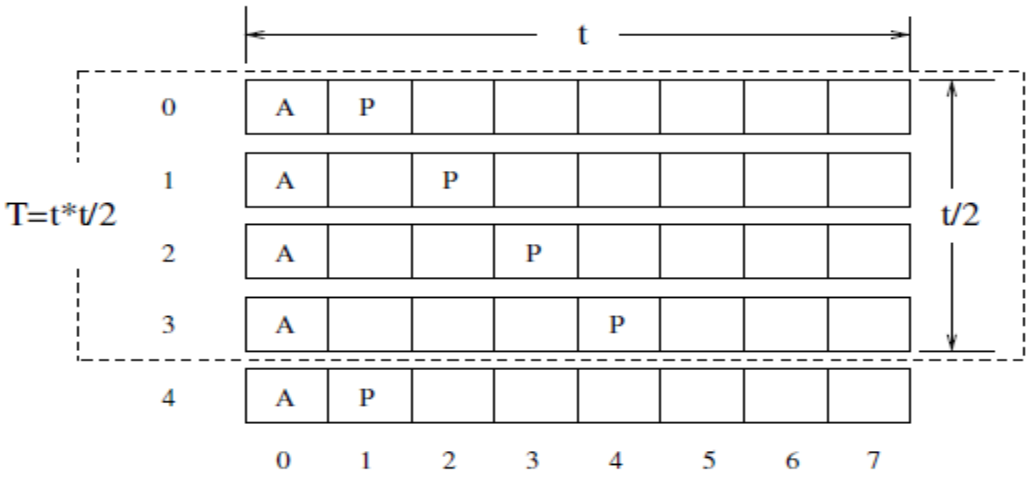

FIGURA 1.1. Sondaggio sistematico (esempio con t = 8)

In altri termini, se indichiamo con P_i la posizione relativa allo slot d'interesse, ovvero il probe slot, durante l'i-esimo periodo temporale di lunghezza t, allora la sua posizione al periodo successivo i + 1 è data da :

$$P_{i+1} = ((P_i) \bmod \lfloor \frac{t}{2} \rfloor) + 1 \qquad (2.1)$$

In pratica, la posizione del probe slot segue un cammino periodico ripetuto ogni $\lfloor \frac{t}{2} \rfloor$ periodi, durata temporale che definiamo come T detta *Hyper-periodo*, che è { 1, 2, 3, ... , $\lfloor \frac{t}{2} \rfloor$ } . Come vediamo chiaramente nell'esempio di Figura 1.1., dove la lunghezza temporale del periodo è stata scelta pari a 8, Searchlight è attivo durante il primo slot di ogni periodo durante l'hyper-periodo, il chè rappresenta gli anchor slots; ed inoltre esso è attivo lungo il pattern periodico { 1, 2, 3, 4 } che determina la posizione temporale del probe slot durante i T = $\lfloor \frac{t}{2} \rfloor$ = 4 periodi d'interesse.

2.3. La latenza della scoperta

Andiamo ad analizzare la latenza peggiore che si potrebbe avere per la discovery utilizzando il protocollo Searchlight ed essendo in presenza di duty-cycles dei due nodi simmetrici.

Supponiamo di essere nella situazione illustrata nella Figura 2.2, dove il nodo x ed il nodo y vogliono effetuare la discovery, essi hanno duty-cycles simmetrici pari a 8. L'offset fra loro è riportato in figura. Si vuole determinare entro quanti slot temporali la discovery avviene con certezza.

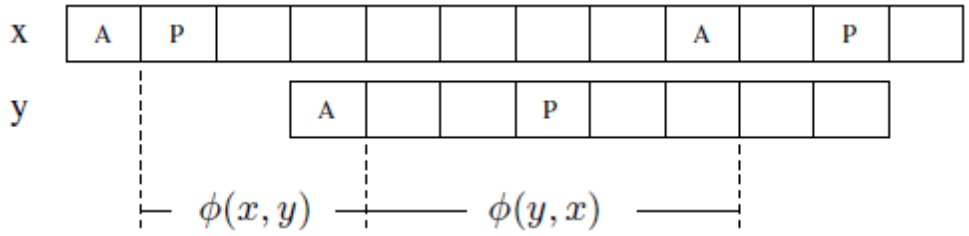

FIGURA 2.2**.** Offset dei nodi con periodo pari a t = 8

Dati due nodi x ed y, sia Φ (x , y) l'offset di fase in termini di slot dallo slot ancora del nodo x, che viene chiamato nel seguito A_x, dello slot ancora del nodo y, chiamato A_y. Si definisce in maniera totalmente analoga l'offset di fase da A_y ad A_x, che viene denotato Φ (y, x). Ovviamente si nota che :

$$\Phi (x , y) + \Phi (y , x) = t \tag{2.2}$$

Per i casi simmetrici dove entrambi i nodi utilizzano il medesimo valore di t, gli offset di fase Φ (x , y) e Φ (y , x) rimangono costanti durante la discovery. Pertanto si ha che :

$$\min (\Phi (x , y) , \Phi (y , x)) \leq \lfloor \frac{t}{2} \rfloor \tag{2.3}$$

E noto che il probe slot và ad esplorare tutte le posizioni all'interno del pattern da 1 a $\lfloor \frac{t}{2} \rfloor$ ogni $\lfloor \frac{t}{2} \rfloor$ periodi. Per semplicità, notiamo con P_x e P_y rispettivamente il probe slot del nodo x ed y. P_x incontrerà A_y entro $\lfloor \frac{t}{2} \rfloor$ periodi purchè :

$$1 \leq \Phi (x , y) \leq \lfloor \frac{t}{2} \rfloor \tag{2.4}$$

Analogamente, P_y ed A_x si sovrapporranno entro $\lfloor \frac{t}{2} \rfloor$ periodi purchè :

$$1 \leq \Phi (y , x) \leq \lfloor \frac{t}{2} \rfloor \tag{2.5}$$

e siccome, come detto in precendenza, si ha sempre valida la seguente disequazione :

$$\min (\Phi (x , y) , \Phi (y , x)) \leq \lfloor \frac{t}{2} \rfloor \tag{2.6}$$

allora possiamo concludere che la discovery avviene entro $\lfloor \frac{t}{2} \rfloor$ periodi, vale a dire entro t . $\lfloor \frac{t}{2} \rfloor$ slots. In altri termini, possiamo concludere che la discovery tramite il protocollo Searchlight nel caso sincrono e simmetrico avviene con certezza entro una durata temporale pari all'hyper-periodo T, definito come $\lfloor \frac{t}{2} \rfloor$ periodi.

2.3.1. Caso sincrono e simmetrico con sondaggio sistematico

All'interno di questo sotto-paragrafo, si andrà a valutare alcuni esempi del protocollo Searchlight operante in situazione asincrona, asimmetrica e con una sola frequenza a disposizione dei due nodi presenti all'interno della rete.

```
The protocol leverages the following slots :

11
11

The device B has accessed in the network :  0 slot later than device A !

The overlap occurs at slot number : 0
```

FIGURA 2.3. Caso Base con t = 2

La situazione qui esposta è il caso base del protocollo Searchlight. Sostanzialente, lo scenario che si andrà a simulare, è quello in cui all'intero della rete ad-hoc che si considera vi sia presente un ed un solo dispositivo, che viene chiamato Device A, e che esso lavori ad una frequenza predefinita. Dopo una certa durata temporale, che è considerata casuale per la realizzazione, un dispositivo, chiamato Device B, entra all'interno della rete stessa. Tale dispositivo entrante puo' utilizzare un'unica frequenza che risulta essere la medesima di quella utilizzata dal Device A.

Le variabili che, in questo semplice e al quanto teorico caso, andranno a fare variare il tempo della scoperta del Device A da parte del Device B, sono essenzialmente due, ovvero la prima è ovviamente la durata temporale del periodo che viene scelta, e la seconda sarà il tempo di offset tra i due devices. Si noti che, come dimostrato nel paragrafo 2.3, questo offset non puo' essere maggiore di $\lfloor \frac{t}{2} \rfloor$.

Il primo caso che andiamo a valutare è una situazione che ovviamente sarebbe molto negativa per la rete. Ovvero, il duty-cycle dei due dispositivi viene considerato pari ad 1, ovvero i nodi sono sempre attivi. Ovviamente, un tale scenario è da scartare a priori nella realtà. Tuttavia viene valutata questa possibilità. Questo avviene quando si pone che t = 2.

Come si vede nella figura 2.3. il dispositivo entrante in rete è entrato nel sistema esattamente allo stesso momento del Device A, pertanto il tempo per effetuare la

discovery è nullo. Anche considerando il caso in cui ci sia un'offset massimo, cioè pari ad 1 slot, la scoperta fra i due dispositivi avviene dopo 1 slot temporale

E' intuitivo che al crescere del numero di slot che compongono la durata temporale del periodo, il tempo medio della discovery aumenta. Inoltre, all'aumentare dell'offset temporale fra i due nodi il tempo per scoprire il vicino, anche in questo caso, cresce. Per mostrare questo, si espone il caso con periodo pari a t = 15 slots.

Valutando l'hyper-periodo temporale sfruttato da Searchlight, nella figura 2.4. si vede come qualora il device B entrasse nel sistema 6 slots temporali dopo il dispositivo già presente in rete, il tempo di discovery è pari a 81 slots.

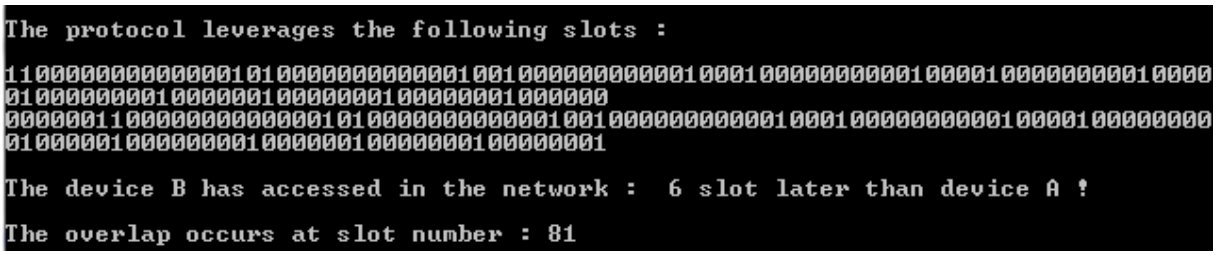

FIGURA 2.4. Caso Base con t = 15 e offset fra i nodi pari a 6

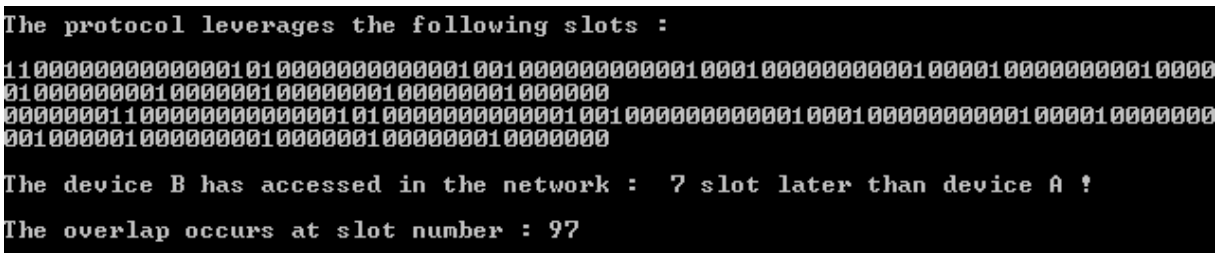

FIGURA 2.5. Caso Base con t = 15 e offset fra i nodi pari a 7

Ponendo che il device B entra dopo un tempo maggiore rispettivamente al caso precedente, il tempo di latenza per scoprire il vicino device A, aumenta. In particolare, qui siamo nella peggiore situazione possibile, vale a dire che il device B entra dopo esattamente 7 slots rispetto al dispositivo presente nel sistema.

Questa relazione fra il tempo di entrata nella rete del device entrante, e durata temporale, in slots, che è necessitata per effetuare la scoperta del vicino presente in rete, è sempre verificata per qualunque valore di t. Si noti infine che qualora si considerasse un valore del periodo temporale t maggiore di $\lfloor \frac{t}{2} \rfloor$, il tempo della latenza della discovery avrebbe un andamento decrescente; questo si spiega in quanto se l'offset fra i due nodi supera il valore di $\lfloor \frac{t}{2} \rfloor$ appunto, è come se si considerasse un caso " virtuale " opposto a quello che si stà valutanto, cioè quando il Device B è già presente nella rete, mentre è il Device A che vuole effettuare la scoperta.

Abbiamo visto come, grazie allo studio della figura 2.5, il tempo massimo per la discovery utilizzando un periodo temporae pari a t = 15 è 97 slots.

Si considerino adesso di valutare le latenze massime per la scoperta del device vicino. Per determinare tale valore, bisogna ovviamente avere che il device B è entrato nella rete

esattamente $\lfloor \frac{t}{2} \rfloor$ slots dopo il device A. Questo procedimento viene realizzato per diversi valori di t, questo porterà a trarre due conclusioni molto interessanti per quanto riguarda il tempo di discovery massimo con l'ausilio di Serchlight in ambiente sincrono e simmetrico.

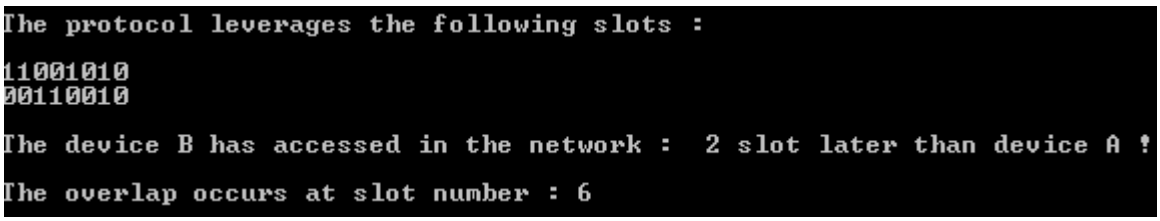

FIGURA 2.6. Caso Baso con t = 4 e offset fra i nodi massimo

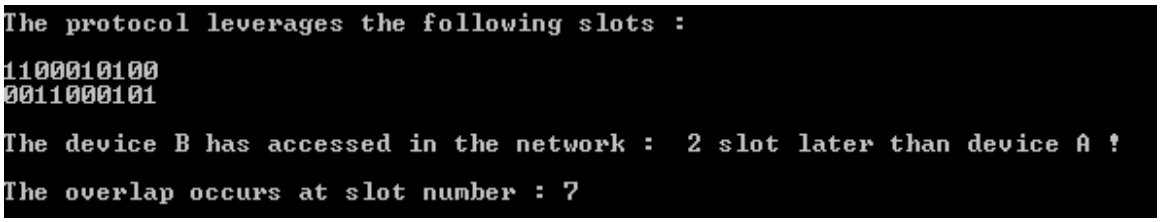

FIGURA 2.7. Caso Base con t = 5 e offset fra i nodi massimo

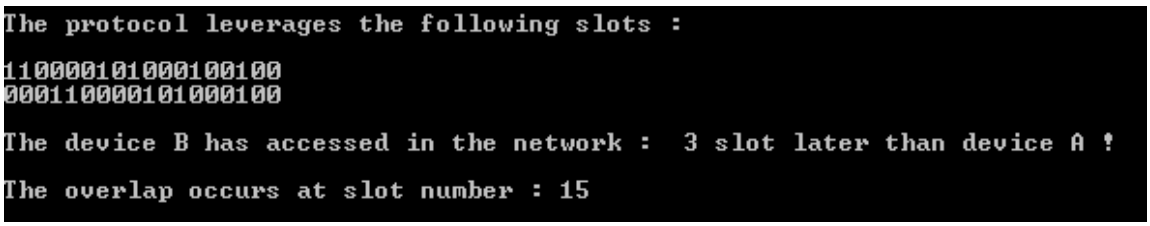

FIGURA 2.8. Caso Base con t = 6 e offset fra i nodi massimo

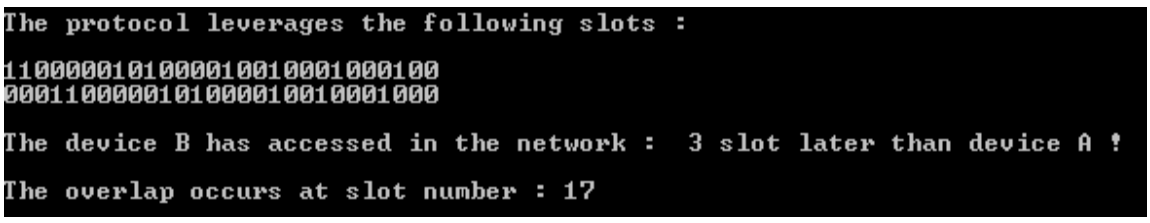

FIGURA 2.9. Caso Base con t = 7 e offset fra i nodi massimo

Con un duty-cycle pari a ½, il ritardo massimo di scoperta dei due dispositivi è pari a 6 slots temporali. Utilizzando un periodo temporale di t = 5, il duty-cycle dimuisce leggermente, pero' la latenza per la discovery del Device B aumenta di 1 slot rispetto al caso con t = 4. Al crescere del periodo, con t = 6 si nota un'aumento abbastanza importante nella latenza rispetto al caso con t = 5. Se ci si pone nella situazione in cui il periodo è pari a 7, la latenza massima in questa situazione è 2 slots piu lunga rispetto al caso con t = 6 riportato dalla figura 2.8. Valutando adesso il periodo temporale t = 8, il tempo massimo della latenza è raggiunto per un'offset tra i nodi pari a 4 slots, e la tale latenza aumenta ancora significativamente rispetto al caso con t = 7, mentre nella figura

2.11 si vede come tale massima latenza è aumentata solamente di 3 slots temporali rispetto al caso riportato della figura 2.11 cioè con t = 8. Si nota infine, sfruttando le figure 2.12 e 2.13 come l'aumento di latenza fra questi due casi è di soli 4 slots temporali, mentre con t = 10, l'aumento della latenza rispetto al caso di t = 9 è di ben 14 slots.

Discutiamo adesso i risutati trovati da queste realizzazioni.

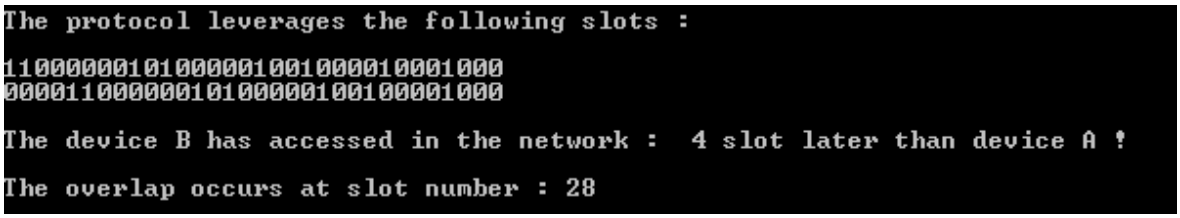

FIGURA 2.10. Caso Base con t = 8 e offset fra i nodi massimo

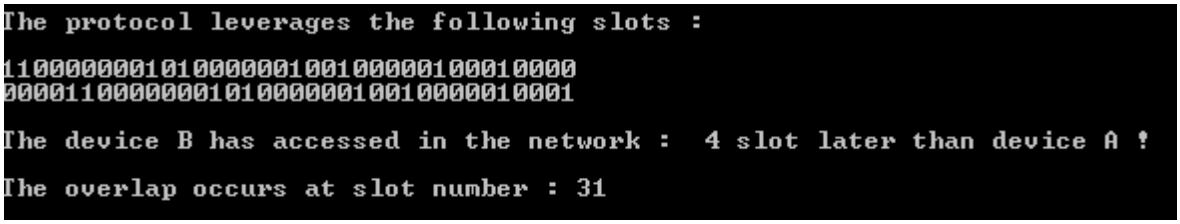

FIGURA 2.11. Caso Base con t = 9 e offset fra i nodi massimo

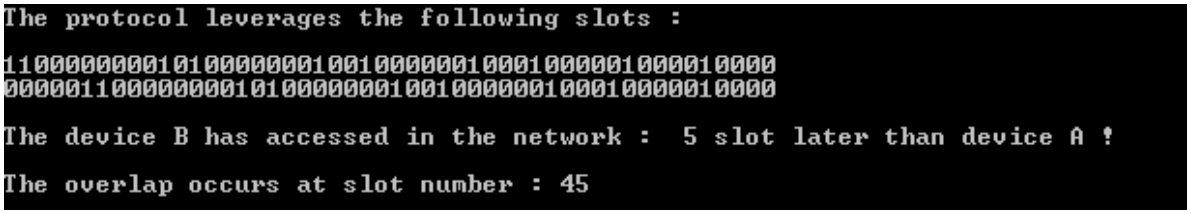

FIGURA 2.12. Caso Base con t = 10 e offset fra i nodi massimo

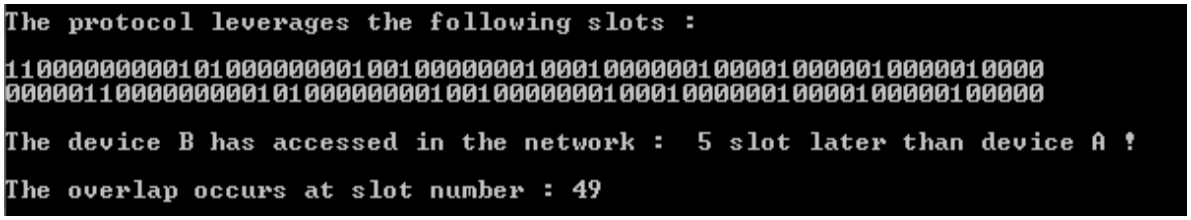

FIGURA 2.13. Caso Base con t = 11 e offset fra i nodi massimo

Osservando con attenzione i valori ottenuti per la discovery di ogni situazione, è importante soffermarsi su due punti.

Il primo, molto intuitivo, è che la crescita del tempo massimo per scoprire il vicino non è lineare. Di fatti, al crescere della durata del periodo, il discovery time massimo cresce in maniera sostanziosa. Pertanto, i due nodi devono scegliere un valore di t conforme ai requisiti della rete; questo valore deve essere di fatto un *trade-off* fra il consumo energetico e il tempo massimo di discovery che si vuole determinare. Si ricorda che un valore alto del periodo t comporta un piccolo duty-cylce, equivalente a basso consumo energetico, mentre un valore basso di t, come nel primo esempio riportato nella figura

2.3., comporta un duty-cycle unitario come valore massimo, che equivale ad un forte consumo energetico.

Infine, osservando i valore della latenza massima, ci si occorge come sia possibile raggruppare in coppie i valori del periodo per la discovery. Se consideriamo ad esempio il caso con t = 8 riportato nell'Immagine 2.8., e il caso con t = 9 riportato nell'esempio 2.9, ci rendiamo conto come l'aumento della latenza sia ragionevolmente piccolo. Questo si traduce in un legame molto stretto fra le coppie di periodi. Se definiamo P_i come il valore riportato di seguito

$$P_i = 2 * i \tag{2.7}$$

, con i intero naturale maggiore di 0. Possiamo definire la coppia di valori del periodo temporale definita da

$$c_i = \{ P_i , P_i + 1 \} \tag{2.8}$$

Se la latenza massima con la scelta del valore del periodo temporale pari a P_i è pari a L_i, possiamo affermare che la latenza massima che è ottenuta con un valore del periodo temporale pari a $P_i + 1$ è pari a

$$L_{i°+1} = L_i + (i - 1) \tag{2.9}$$

Come esempio ulteriore, e maggiormente interessante di questa affermazione, consideriamo di avere la situazione con i = 40.

$P_i = 2 * i = 80$, da cui si trova il secondo elemento della coppia c_{40}, ovvero $P_i + 1 = 81$

```
The device B has accessed in the network :  40 slot later than device A !
The overlap occurs at slot number : 3160
```

FIGURA 2.14. Caso Baso con t = 80 e offset fra i nodi massimo

```
The device B has accessed in the network :  40 slot later than device A !
The overlap occurs at slot number : 3199
```

FIGURA 2.15. Caso Base con t = 81 e offset fra i nodi massimo

Per comodità, si noti che è stato riportato solamente il risultato della discovery, e non gli slots sfruttati dal protocollo Searchlight, come si puo' notare dalla figura 2.14. Si ha dunque un tempo massimo di discovery con t = 80 slots e tempo di offset fra il Device A ed il Device B pari a $\lfloor \frac{t}{2} \rfloor$ = 40 slots, pari a L_{80} = 3160 slots.

Seguendo la tesi esposta in precendenza, andiamo a valutare il tempo massimo di discovery che si ottiene utilizzando il secondo elemento della coppia di periodi temporali c_{40}. Si ha che L_i = 3160 slots, con i = 40, pertanto, il tempo di discovery con t = 81 e tempo di offset massimo quindi pari a $\lfloor \frac{t}{2} \rfloor$ = 40 slots dovrebbe essere pari a

$L_{81} = L_{80} + (i - 1) = 3160 + 40 - 1 = 3199$ slots

Anche all'interno della figura 2.15 vengono ommessi i due hyper-periodi relativi ai due nodi, pero' il risultato teorico qui calcolato è, come ci si attendeva, quello esatto.

Lo studio effettuato nei paragrafi precedentemente esposti, ha mostrato il funzionamento del protocollo Searchlight e come grazie ad esso sia possibile per un device scoprire un suo vicino. Nel seguito dello studio si andrà a vedere come il protocollo Searchlight possa essere sfruttato anche in stuazioni di asincronia ed asimmetria. Inoltre, si vedrà come il sondaggio effettuato dai due slots attivi all'interno del periodo temporale definito da t possa avvenire in maniera casuale, e non piu sistematica come fin qui visto, quest'ultima estenzione porterà a sottolineare come la sovrapposizione possa anche avvenire tra slot probe-ancora, e come questo possa portare un guadagno temporale per la scoperta.

2.4. La gestione della asincronia tramite sondaggio " a strisce "

Fino a questo momento dello studio, gli slots sono stati mostrati allineati. Ovvero l'ipotesi fatta è stata, in termini chiari, che l'offset dei due nodi che vogliono effettuare la discovery sia un numero intero di slot. Pero', Searchlight è stato progettato per poter gestire l'asincronia e pertanto esso non confida nell'allineamento degli slot dei diversi nodi. Per assicurare che la sovrapposizione di due slot attivi avvenga durante la discovery, Searchlight utilizza la stessa strategia del protocollo DISCO. Searchlight sfrutta la tecnica di *beaconing*.

In particolare, un segnale di beacon viene inviato all'inizio e alla fine di uno slot attivo, ed il nodo rimane in ascolto durante il periodo intermedio a questi due segnali di beacon. Utilizzando questa tecnica, il non allineamento degli slot porterà di fatto ad una latenza per effettuare la discovery relativamente bassa, questo dovuto al fatto che ogni slot attivo di un nodo si sovrappone con due slot relativi all'altro nodo, e la discovery puo' quindi avvenire con qualunque delle due sovrapposizioni. Questa tecnica è mostrata in maniera chiara nella Figura 2.3, dove si osserva che ogni slot attivo puo' esplorare due slot di un'altro nodo che non è allineato con esso.

Tuttavia, due sovrapposizioni risultano essere ridondanti in quanto la discovery puo' essere assicurata da un solo overlap. Basandosi su questa osservazione molto importante, Searchlight utilizza una strategia che và a migliorare in maniera molto significativa le prestazioni eliminando i sondaggi ridondanti. Questa strategia, che si andrà a descrivere nel seguito, viene chiamata *Striped probing*, cioè sondaggio " a strisce".

Sondare ogni slot da 1 a $\lfloor \frac{t}{2} \rfloor$ garantisce due sovrapposizioni ogni $\lfloor \frac{t}{2} \rfloor$ periodi temporali qualora gli slot di bordo di due nodi non sono allineati. Questo risultato è ottenuto basandosi sul concetto visto in precendenza di sondaggio sistematico. Per ridurre questa ridondanza, il probe slot puo' sondare, invece che ogni slot, utilizzando un

contatore settato a 2 ed incrementandolo di 2 ad ogni periodo fino al valore ($2 \cdot \lceil \frac{\lfloor \frac{t}{2} \rfloor}{2} \rceil$) e quindi ripartire a 2 nuovamente.

In altri termini, chiamanto P_i la posizione del probe slot durante l'i-esimo periodo temporale, allora la posizione del probe slot durante il periodo i+1 è data da :

$$P_{i+1} = ((P_i) \bmod (2 \cdot \lceil \frac{\lfloor \frac{t}{2} \rfloor}{2} \rceil) + 2 \qquad (2.10)$$

E molto important sottolineare che questo sondaggio strisciato non puo' in alcun modo essere applicato al protocollo del compleanno e al protocollo DISCO in quanto questi due protocolli non presentato l'utilizzo di slot consecutivi. Siccome U-Connect fornisce un'insieme di $\frac{p+1}{2}$ slot attivi sequenziali ogni p^2 slots, lo striped probing puo' essere utilizzato per ridurre questo numero a $\frac{p+1}{4}$, comportando cosi una riduzione del 10% del numero di slot attivi.

Tuttavia, la strategia di striped probing risulta invece inutilizzabile nella particolare casistica in cui gli slot siano completamente allineati. Per gestire questo perfetto allineamento, ogni slot attivo trabocca di un valore δ sullo slot seguente inattivo; questo valore δ piccolo deve essere piccolo il piu' possibile a condizione che lo slot attivo sia in grado di ricevere il segnale di beacon da un'altro nodo.

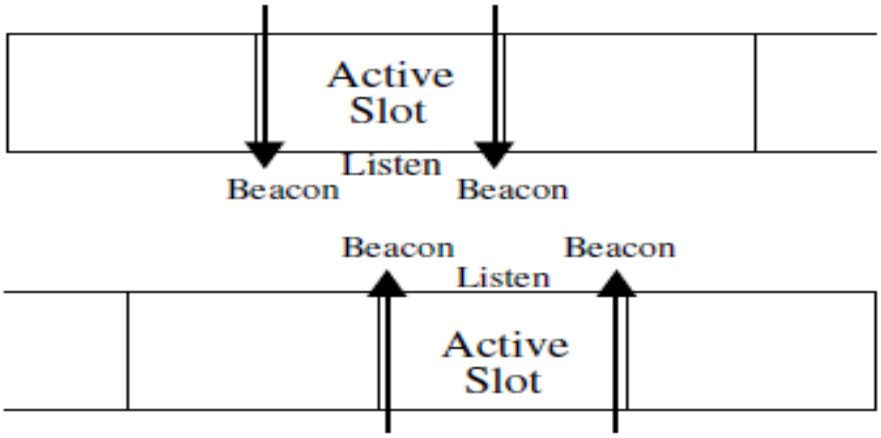

FIGURA 2.16. Beaconing alle estremità di uno slot attivo

Quest'ultima affermazione significa che se la dimensione di uno slot è definita pari ad x, allora ogni slot attivo puo' essere considerato di lunghezza pari a x(1 + δ), mentre la lunghezza di uno slot regolare spento che segue direttamente uno slot attivo viene ridotta pari a x(1 – δ). Il valore di δ è la piu' piccola sovrapposizione possibile che garantisca la discovery fra due nodi. Si noti che a causa dei tempi di trasmissione o anche dell'effetto del clock drift, che abbiamo per il momento trascurato, il valore specifico di δ è strettamente dipendente alla singola casistica in cui ci si trova. Per avere un'idea globale del valore di δ, possiamo dire che esso puo' essere una piccola frazione della dimensione dello slot naturale, ovvero x come definita in precendenza (ad esempio 1%).

Il principale vantaggio che porta il sondaggio " a strisce " stà nel fatto che il suo utilizzo riduce drasticamente il latency worst-case bound. Ad esempio , se ci si riferisce alla situazione asincorna e simmetrica, ovvero quando gli slot non sono tra loro allineati e quanto il duty-cycle dei due nodi è il medesimo, tale bound diventa

$$t \cdot \left\lceil \frac{\lfloor \frac{t}{2} \rfloor}{2} \right\rceil \text{ slots} \qquad (2.11)$$

valore che è ovviamente migliore rispetto a quello determinato dal sondaggio sistematico che era stato calcolato all'interno del paragrafo 2.3 pari a

$$t \cdot \lfloor \frac{t}{2} \rfloor \text{ slots.} \qquad (2.12)$$

2.5. La gestione della asimmetria

Il progetto principale del protocollo Searchlight è basato, come piu' volte detto, sul costante offset relativo tra due slot ancora di qualunque coppia di nodi vicini. Sfortunatamente, questo offset non è mantenuto costante nel caso asimmetrico, ovvero quando due nodi scelgono due valori diversi per la durata temporale del periodo t basandosi sui propri requisiti energetici.

Un possibile approccio per garantire la sovrapposizione entro un bound prefissato è quello di richiedere che il valore di t sia sempre un numero primo. Essenzialmente, restringendo t ad essere un numero primo, simile a quello che avviene con i protocolli DISCO e U-Connect, assicurerebbe che per ogni coppia di nodi che operano con un duty-cycle diverso, le lunghezze dei loro periodi t_1 e t_2 saranno primi fra loro, vale a dire che essi non avranno fattori comuni a parte il valore 1. Siccome le lunghezze dei periodi sono prime fra loro, ne segue che due slot ancora si sovrapporanno almeno una volta ogni $t_1 \cdot t_2$ slots. Pero', questa restrinzione dei periodi ad essere dei valori primi non mantiene necessariament l'offset costante.

Invece che utilizzare dei valori primi, Searchlight utilizza un nuovo approccio nel caso di duty-cycles asimmetrici. E possibile mantere l'offset di due anchor slots di due nodi costante se il valore del periodo piu' lungo è un multiplo intero del piu' piccolo. Si vedrà adesso come viene attuata questa strategia con l'aiuto di un esempio pratico, dove si vede come la scelta del periodo temporale dei due nodi sia fondamentale ai fini della scoperta dei dispositivi vicini.

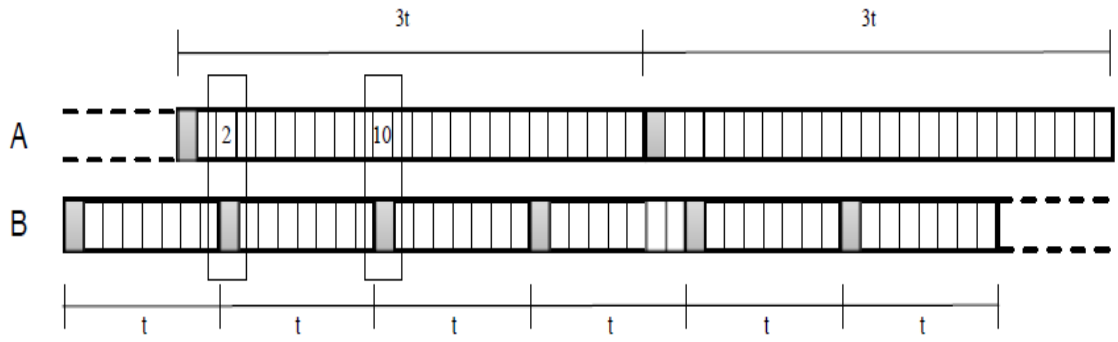

FIGURA 2.17. Nodo A e nodo B con duty-cycles asimmetrici (rispettivamente 3.t e t)

2.5.1. Esempio esplicativo

Andiamo a vedere il funzionamento di questa strategia tramite un'esempio esplicativo, considerando due nodi, ove il nodo A ha un periodo temporale pari a 3.t, mentre il nodo B ha scelto un periodo temporale pari a t. Questa situazione viene riportata nella Figura 2.4.

Gli slot ancora del nodo B hanno sempre la medesima distanza relativa da tutti gli anchor slots del nodo A. Siccome il nodo A sonda solamente la metà dei propri slot, questo assicura che il probe slot del nodo A si incontrerà eventualmente con qualunque anchor slot del nodo B che ricade all'interno della prima metà del ciclo del nodo A.

In questo particolare esempio esposto, il nodo B effettua sempre un sondaggio dei primi $\frac{3*t}{2}$ slots. Siccome gli anchor slots del nodo A sono sempre allineati con il secondo (slot 2) ed il decimo (slot 10) slot all'interno di un periodo del nodo B, la discovery fra questi due nodi è garantita L'esempio valutato puo' essere esteso ad una scelta del periodo temporale da parte del nodo A ad un qualunque valore pari ad n.t, ove n è un numero intero.

Benchè qualunque valore multiplo del piu' piccolo duty-cycle è adeguato per mantere l'offset costante fra una coppia di nodi, Searchlight, per far si che l'offset rimanga costante per tutti i nodi presenti all'interno della rete, restringe il duty-cycle ad una potenza multipla del piu' piccolo duty-cycle, ad esempio 2, 4, 8, 16, ... oppure 3, 9, 27, 81, ... eccetera. Cosi facendo, è garantito che ogni coppia di duty-cycles scelti da due nodi sia un valore multiplo degli altri. Questo significa che se t è il piu' piccolo duty-cycle, gli altri nodi hanno la propria scelta della lunghezza del periodo temporale vincolata ad essere 2.t, 4.t, 8.t, eccetera, per ogni valore di t.

Anchè se questa restrizione sembra ridurre la scelta dei livelli di conservazione dell'energia, è importante sottolineare che l'utilizzo di numeri primi, come avviene nei protocolli di discovery precedentement implementati, è anch'esso molto restrittivo. Dove gli approcci basati sui numeri primi, come DISCO e U-Connect, sono limitati ad una

scelta di numeri primi, Searchlight puo' essere inizializzato ad utilizzare qualunque valore di base di t in modo tale da adattarsi nel miglior modo possibile ai requisti energetici del sistema utilizzato. Per di piu', l'uso di potenze di 2 fornisce la massima flessibilità ai duty-cycles piu' piccoli, dando cosi alle diverse applicazioni piu' possibilità quando questo è necessitato. Infine, Searchlight puo' utilizzare qualunque potenza multipla di x, fornendo un'ulteriore flessibilità se questo risultasse essere necessario.

Insieme alla flessibilità, l'approccio basato sulle potenze multiple fornisce anche il miglior worst-case bound per la latenza, il cui si basa sul tempo necessario al probe slot del nodo A per trovare un'anchor slot del nodo B. Il probe slot del nodo A sonda ogni altro slot da 2 a $\lceil \frac{\lfloor \frac{t_a}{2} \rfloor}{2} \rceil$. Si assuma che all'interno di questo range si trovino presenti m anchor slots del nodo B. Nel caso peggiore, quando A incontra B, il probe slot del nodo A ha appena passato la m-esima ancora del nodo B. Per sovrapporsi, il probe slot deve sempre esplorare fino a $\lceil \frac{\lfloor \frac{t_a}{2} \rfloor}{2} \rceil$, per poi ripartire da 2 di nuovo ed incontrare la prima ancora del nodo B. Cio' significa che il probe slot dovrà "viaggiare"

$$t_B + (\lceil \frac{\lfloor \frac{t_a}{2} \rfloor}{2} \rceil . \bmod t_B) \quad \text{slots} \tag{2.13}$$

Se invece vogliamo inserire la tecnica vista per la gestione della asincronia, precedentemente esposta, ovvero lo striped probing, il probe slot dovrà esplorare

$$(t_B + (\lceil \lceil \frac{\lfloor \frac{t_a}{2} \rfloor}{2} \rceil \rceil \bmod t_B)) / 2 \quad \text{periodi} \tag{2.14}$$

oppure,

$$((t_B + (\lceil \frac{\lfloor \frac{t_a}{2} \rfloor}{2} \rceil \cdot \rceil \bmod t_B)) / 2) . t_A \quad \text{slots} \tag{2.15}$$

Per concludere, si noti che l'approccio basato sui numeri primi e l'approccio basato sulle potenze multiple possono essere combinati se necessario, questo purchè i numeri primi scelti siano rispettivamente primi con il valore del periodo temporale di base, ovvero t. Ad esempio, quando sono richesti quattro duty-cycles, tre valori possibili per t possono essere t_{base}, $2 . t_{base,}$ $4 . t_{base}$ ed il quarto valore puo' essere qualunque numero primo diverso da 2, che siano anche primo con il valore base scelto, ovvero t_{base}.

Adattando il protocollo Searchlight con queste modifiche, i problemi di asincronia ed asimmetria vengono dunque gestiti, senza che essi comportino problemi particolari.

2.6. Introduzione della componente probabilistica: Searchlight-R

Il sondaggio sequenziale studiato fino a questo punto lavora molto bene nell'assicurare la discovery trovando gli slot ancora degli altri nodi, vale a dire la sovrapposizione probe-anchor slot. Nonostante questo, la sovrapposizione di due probe slots puo' anche avvenire per portare ad una discovery avvenuta con successo. Benchè l'overlap probe-probe sia possibile tramite il sondaggio sistematico, i probe slots di due nodi seguono lo stesso schema di sondaggio, e pertanto sono spesso sincronizzati gli uni con gli altri riducendo sensibilmente la probabilità di sovrapposizione probe-probe. Si considerino due nodi A e B con relativo offset di fase pari ad esattamente uno slot ed un valore del periodo temporale pari a t = 8 per entrambi i nodi.

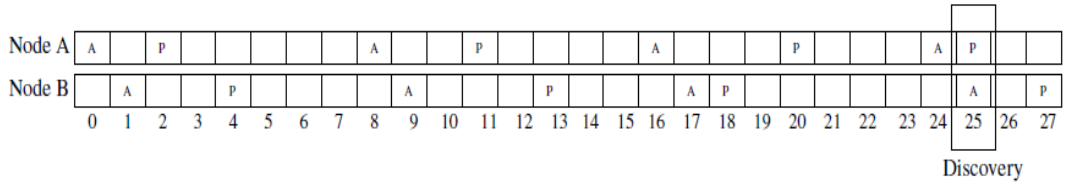

FIGURA 2.18. Sovrapposizione con sondaggio sequenziale con un periodo t = 8

Per semplicità dello studio supponiamo che non venga utilizzato il sondaggio a striscie. Quando i due si incontrano la prima volta, il probe slot del nodo A si trova in posizione 2, mentre il probe slot del nodo B è in posizione 3. Siccome entrambi i probe slots dei nodi A e B seguono il pattern sequenziale { 1, 2, 3, 4 }, le prossime posizioni dei probe slots saranno rispettivamente 3 e 4 per i due nodi a cui ci stiamo riferendo. Percio', i probe slots dei due nodi, A e B, si " inseguono " senza che la sovrapposizione avvenga, e questo in nessuno istante temporaneo all'intero dell'hyper-periodo di riferimento. Questa situazione è riportata nella Figura 2.5.

2.6.1. Randomized probing

Per consentire l'aumento della probabilità di una sovrapposizione fra gli slot probe dei due nodi, Searchlight introduce una componente probabilistica, simile a quello che viene effettuato all'interno del protocollo di discovery visto nel primo capitolo, ovvero il protocollo del compleanno.

Invece che restringere il cammino del probe slot al pattern 1, 2, 3, ..., $\lfloor \frac{t}{2} \rfloor$, i nodi possono scegliere in maniera completamente *casuale* qualunque pattern per il probe slot tale che esso risulti essere una permutazione dei valori compresi nell'intervallo che va da 1 a $\lfloor \frac{t}{2} \rfloor$. Utilizzando invece striped probing, lo schema di sondaggio utilizzato in

presenza di asincronia, le permutazioni avvengono all'intero del range definito dagli slot 2, 4, 6, ... , $2 \cdot \left\lceil \frac{\lfloor \frac{t}{2} \rfloor}{2} \right\rceil$.

Per differenziare l'approccio sistematico visto fino a questo punto del capitolo e quello casuale, si chiamerà nel resto dello studio Searchlight-S la versione con sondaggio sistematico, e Searchlight-R la versione definita all'interno di questo paragrafo. Dove ovviamente la lettera S stà per Systematic, mentre R stà per Randomized.

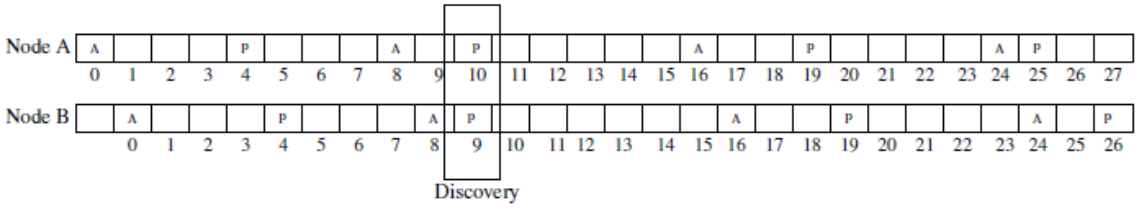

FIGURA 2.19**.** Sondaggio casuale con periodo t = 8

Se consideriamo l'esempio visto in precendenza, riportato nella Figura 2.5. e brevemente discusso nel punto 2.6., invece di restringere il pattern del probe slot dei due nodi a quello sistematico, vale a dire { 1, 2, 3, 4 }, Searchlight-R consente ai nodi di scegliere in maniera random qualunque pattern che risulti essere una permutazione dei numeri interi che vanno da 1 a $\lfloor \frac{t}{2} \rfloor$, cioè 4 nel caso particolare in esame. In questo scenario, con il valore temporale del periodo posto a t = 8, si hanno esattamente 4! permutazioni possibili che possono essere valutate da Searchlight-R.

Per l'esempio fornito, il nodo A sceglie in modo casuale il pattern periodico { 1, 4, 2, 3 }, mentre il suo vicino B sceglie il pattern periodico { 1, 3, 2, 4 }. Questo esempio viene riportato nella Figura 2.6. Al primo incontro, i probe slot del nodo A e B sono rispettivamente in posizione 4 e 4, e l'offset fra loro è pari a uno slot. Si noti che per semplicità siamo in un caso sincrono e simmetrico. Per causa di questo offset di fase, i probe slot non si incontrano durante il primo periodo temporale. Nel secondo periodo, il nodo vede il suo probe slot nella posizione 2 mentre il probe slot del nodo B è in posizione 1, il chè risulta essere una sovrapposizione degli slot attivi durante il secondo periodo, e la discovery avviene con successo. Pertanto, l'utilizzo di diversi pattern per la selezione dei probe slot comporta una discovery piu' veloce grazie allo sfruttamento dell'overlap probe-probe.

Questo approccio probabilistico incrementa sensibilmente la possibilità che la discovery avvenga tramite la sovrapposizione probe-probe, e questa senza diminuire l'abilità di un probe slot di trovare un'anchor slot nel caso simmetrico.

Cio' nonostante, nel caso asimmetrico, il randomized probing incrementa il latency worst-case bound, questo è dovuto al fatto chè nel caso peggiore il probe slot del nodo che presenta il periodo t piu' grande puo' esplorare tutti gli slots regolari prima di trovae un'anchor slot. In questo scenario, il worst-case bound risulta essere pari a

t_A . $\lceil \lceil \frac{\lfloor \frac{t_A}{2} \rfloor}{2} \rceil \rceil$, ove t_A è il periodo del nodo che ha il periodo temporale piu' lungo, ovvero il duty-cycle piu' piccolo.

Detto questo, si puo' raggiungere un bound uguale a quello del caso simmetrico restringendo la scelta delle permutazioni possibili che possono essere effettuate. Un semplice approccio puo' essere quello di limitare la casualità all'interno di blocchi di t_{base} slots pero' questo decrementerebbe le prestazioni nel caso medio in situazione di duty-cycles simmetrici. Si presenta nel seguito delle permutazione piu' accurate che preservano il worst-case bound ma che permettano una maggiore casualità.

2.6.2. Restricted randomized probing

Il worst-case bound fornito con l'ausilio di Searchlight-S si basa sull'osservazione che lo slot di sondaggio del nodo A è assicurato di incontrare uno slot ancora del nodo B ogni t_B periodi. Pero', questa affermazione non è piu' valida utilizzando la tecnica di randomized probing, cioè con Searchlight-R.

Per afferrare meglio questo concetto si dividono gli slot del nodo A in t_B cesti. Assumendo che la posizione dell'anchor slot sia posto in posizione 0, i cesti saranno i seguenti :

Cesto 1 : { 1 , 1 + t_B , 1 + 2. t_B , 1 + 3. t_B , ... }

Cesto 2 : { 2 , 2 + t_B , 2 + 2. t_B , 2 + 3. t_B , ... }

Cesto 3: { 3 , 3 + t_B , 3 + 2. t_B , 3 + 3. t_B , ... }

...

Cesto t_B : { t_B , 2. t_B , 3. t_B , ... }

Basandosi sull'offset tra i nodi A e B, le ancore del nodo B cadranno all'interno di ognuno dei cesti qui definiti. Ad esempio, se l'offset tra i nodi A e B è pari a 3, gli anchors slots del nodo B saranno posizionati negli slot 3, 3 + t_B , 3 + 2. t_B , eccetera , ovvero il cesto 3 predefinito. Utilizzando il probing sequenziale, è garantito che il pattern seguito dal probe slot contenga uno slot di ogni cesto ogni t_B slots. Sfortunatamente, con Searchlight-R, non si puo' fornire tale garanzia. Andiamo quindi a presentare nel seguito una forma leggermente ristretta del randomized probing che fornisca una risposta a questo problema.

L'obiettivo base del randomizing probing è quello di assicurare che solamente un certo numero di pattern di spostamento verrà scelto per sondare dove sia ha esattamente uno slot per ogni cesto ogni t_B slots. Questo viene raggiunto con uno schema di casualità a due passi.

1. Viene scelta una permutazione dei cesti. Questo determina l'ordine in cui gli slots dei diversi cesti appariranno.

2. Viene scelta una permutazione casuale degli slots all'interno di ogni cesto.

2.6.3. Esempi numerici

La strategia de restricted randomized probing definita nel sotto-paragrafo 2.6.2. risulta essere molto piu' chiara valutanto degli esempi pratici. Dentro a questa sezione, si valutano due esempi, il primo molto semplice, ed un secondo piu' interessante, che mostreranno come ottenere un worst-case bound per la latenza pari a quello che si ottiene nel caso simmetrico.

Esempio 1 :

Consideriamo due nodi A e B, che hanno come periodi temporali rispettivamente $t_A = 18$ e $t_B = 3$. Con un sondaggio sequenziale e senza striping probing, il probe slot del nodo A segue il seguente pattern, da leggersi da sinistra a destra, e dall'alto verso il basso

1 2 3

4 5 6

7 8 9

ogni colonna qui rappresentata è di fatto un cesto. Il primo passo del restricted radomized probing è quello di rendere casuale l'ordine dei cesti. Un possibile ordine è dato da

STEP 1 :

2 3 1

5 6 7

8 9 4

Il secondo step, e quello finale, sta nel scegliere un'ordinamento casuale degli slots all'interno di ogni cesto, ad esempio si puo' scegliere l'ordine seguent

STEP 2 :

5 6 4

2 9 7

8 3 1

Il pattern trovato, in particolare { 5, 6, 4, 2, 9, 7, 8, 3 1 } assicura che ci sia esattamente uno slot di ogni cesto ogni $t_B = 3$ slots.

Esempio 2 :

Si prenda in esame adesso due nodi C e D, che hanno come lunghezza temporale del periodi i valori $t_C = 60$ e $t_D = 10$. Si ha che il probe slot del nodo C segue il pattern riportato di seguito

1 2 3 4 5 6 7 8 9 10

11 12 13 14 15 16 17 18 19 20

21 22 23 24 25 26 27 28 29 30

Una possibile prima permutazione è la seguente

STEP 1 :

6 9 1 10 2 5 7 8 4 3

16 19 11 20 12 15 17 18 14 13

26 29 21 30 22 25 27 28 24 23

La permutazione degli slot all'interno di ogni cesto è la seguente

STEP 2 :

16 9 1 10 22 15 7 18 4 23

6 29 11 20 12 25 17 8 24 3

26 19 21 30 2 5 27 28 14 13

Pertanto, adottando questo approccio a due passi, il sondaggio casuale ristretto raggiunge il medesimo peggior bound della latenza che quello che viene fornito effetuando un sondaggio sequenziale nel caso di simmetria dei duty-cycles dei nodi.

2.7. Effetti della deviazione dei clock

Il sondaggio sistematico di Searchlight confida nell'assunzione di un'offset costante tra i nodi ancora di qualunque coppia di nodi. In realtà, i drift dei clock e l'offset tra gli anchors slots cambia lungo la duranta temporale della discovery. Cio' nonostante, se la deviazione del clock è sufficientement piccola, il valore peggiore della latenza per effetuare la discovery non varia.

Per intuire meglio l'effetto della deviazione del clock sul worst-case bound, si consideri la peggiore situazione per lo schedule di due nodi. Si assuma che non avvengano discovery di tipologia probe-probe e ci si concentri unicamente sulla sovrapposizione tra

anchor slot e probe slot. E stato mostrato nel paragrafo 2.3. che l'esistenza di questa tipologia di discovery avviene sempre dati due nodi. Il minimo numero di sovrapposizioni fra anchor slot e probe slot avviene quando lo slot ancora parte immediatamente dopo il probe slot. Siccome Searchlight è un protocollo che sfrutta lo striped probing, la dimensione temporale dell'anchor slot è pari a x (1 + δ) ed esso partirà allo stesso istante dello slot inattivo di dimensione x (1 – δ), producendo pertanto una sovrapposizione pari a x(2. δ) con il prossimo slot attivo.

Siccome l'unico overlap che abbiamo considerato è quello probe-anchor se lo slot ancora è deviato verso il punto in cui non c'è piu' sovrapposizione con questo probe slot all'interno di un'hyper-periodo, la discovery non è piu' garantita. Pertanto, lo scostamento del clock deve rispettare essenzialmente la seguente disequazione

$$\text{Clock Drift} < \frac{x*(2\delta)}{T} \qquad (2.16)$$

per far si che il worst-case bound venga mantenuto. Rispettando questo vincolo, δ puo' essere scelto basandosi sul clock drift per un dispositivo dato tale che il bound adeguatamente stabilito venga mantenuto.

2.8. Confronto con i protocolli deterministici esistenti

Fino a questo punto, ci siamo interessati al worst-case bound per la latenza della discovery fornito dal protocollo Searchlight in diversi scenari. E anche pero' molto interessante considerare il costo energetico per raggiungere un particolare bound. Siccome il costo primario di cui siamo interessati è quello dell'energia, osserviamo con maggiore attenzione il duty-cycle.

Dato un valore del periodo temporale t, utilizzando Searchlight, abbiamo due slots attivi ogni t slots. Se consideriamo il costo che comporta energeticamente l'estensione degli slot attivi del valore δ precedentement discusso, il duty-cycle risultante per ogni t è

$$\eta = \frac{2.(1+\delta)}{t} \qquad (2.17)$$

Nella Figura 2.20 vengono riportati i costi in termini di duty-cycle e le prestazioni in termini di worst-case bound dei tre principali protocolli deterministici. Cio' nonostante, basandosi su queste informazioni, risulta complicato confrontare i diversi protocolli in quanto usano diversi parametri.

Per fornire una linea comune, puo' essere utile fissare il duty-cycle in modo tale che ogni protocollo cosumi energia allo stesso rate e andremo a valutare il worst-case bound fornito da questi tre protocolli. Procedendo come descritto, possiamo quindi confrontare le prestazioni dei diversi protocolli quando essi hanno un costo energetico uguale.

Il protocollo DISCO fornisce le prestazioni migliori nel caso simmetrico, ovvero quando i due numeri primi p_1 e p_2 sono il piu' vicino possibile. Per semplicità nell'analisi, assumiamo che $p_1 = p_2 = p$. Andiamo adesso ad esprimere i parametri di ogni protocollo in funzione di x , dove 1 / x è il duty-cycle comune ai tre protocolli.

Per il protocollo DISCO si ha che

$$\frac{p1 + p2}{p1 * p2} \sim \frac{p + p}{p * p} = \frac{1}{x} \tag{2.18}$$

da cui segue che

$$p_{DISCO} = 2.x. \tag{2.19}$$

In maniera simile, per Searchlight, si ottiene che

$$t_{SEARCHLIGHT} = 2.x \, . \, (1 + \delta) \tag{2.20}$$

Protocol	Para-meters	Duty Cycle	Worst-case bound Symmetric case
Disco	p_1, p_2	$\frac{p_1+p_2}{p_1 \cdot p_2}$	$p_1 \cdot p_2$
U-Connect	p	$\frac{3p+1}{2p^2}$	p^2
Searchlight	t	$\frac{2 \cdot (1+\delta)}{t}$	$t \cdot \lceil \frac{\lfloor \frac{t}{2} \rfloor}{2} \rceil$

FIGURA 2.20. I protocolli deterministici principali

Protocol	Worst-case bound(slots)
Disco	$4x^2$
U-Connect	$\frac{9x^2}{4}$
Searchlight	$2x(1+\delta))\lceil \frac{\lfloor \frac{2x(1+\delta)}{2} \rfloor}{2} \rceil \approx x^2(1+\delta)^2$

FIGURA 2.21**.** I bound per il caso peggiore operando a duty-cycles uguali

Ed infine per U-Connect, si ottiene che

$$\frac{3.p + 1}{2.p^2} = \frac{1}{x} \tag{2.23}$$

e pertanto

$$p_{U\text{-}Connect} \sim \frac{3*x}{2} \tag{2.22}$$

Possiamo adesso esprimere i worst-case bounds dei tre principali protocolli deterministici qui studiati come funzioni del parametro x, vale a dire f(x). Questi valori vengono riportati dalla Figura 2.21. Studiando questoa Tabella, è chiaro come anche se viene effetuata una scelta conservativa per il valore di δ , come ad esempio $\delta = 0{,}1$, che significa ingrandire gli slot attivi di un decimo della dimensione dello slot di base, Searchlight riduce il worst-case bound per la latenza della discovery di un valore attorno al 50% rispetto al protocollo U-Connect, che risulta essere il protocollo deterministico attualmente sul mercato migliore in termini prestazionali.

Abbiamo discusso all'intero di questo Capitolo 2 la definizione, il funzionamento e come avviene la gestione dell'asincronia e dell'asimmetria per il protocollo Searchlight. Si è visto come Searchlight sia un protocollo asincrono di discovery periodico basato sugli slot, dove un periodo consiste in t slots contigui determinati dal duty-cycle desiderato da un dato nodo. Inoltre, grazie allo studio delle realizzazioni effetuate sono state determinate alcune relazioni importanti tra la durata del periodo temporale ed il tempo di offset fra i due nodi, ovvero il lasso di tempo che è intercorso tra l'entrata nel sistema da parte dei due dispositivi. Come è facile intuire, la casistica fin qui discussa è poco realistica. Di fatti, nella realtà, i diversi dispostivi possono lavorare con diverse frequenze e gestire il loro utilizzo come meglio credono. Inoltre, lo studio del Capitolo 2, valuta solamente la situazione "iniziale" della rete, vale a dire quando un dispostivo accede alla rete, ed in tale rete si ha uno ed un solo dispositivo già presente. Nei successivi Capitoli si andrà a discutere di queste situazioni, indubbiamente piu' realistiche, dove i devices possono utilizzare un numero di frequenze predefinito e maggiore di 1, e all'interno delle quali la rete di interesse è costituita da piu' devices.

CAPITOLO 3 - Ricerca di un vicino e della sua frequenza di utilizzo

Nel precedente capitolo è stato esposto l'uso di un nuovo protocollo per la ricerca di vicini nell'ambiente delle reti ad-hoc. Il protocollo Searchlight è implementato per poter lavorare con una ed una sola frequenza per ogni dispositivo, e che quest'ultima sia la medesima per ognuno di essi. L'estensione che verrà effettuata all'interno del Capitolo 3 infrange questa limitazione di Searchlight, dando cosi l'opportunità ai dispositivi di rete di utilizzare una frequenza a piacere.

In questo scenario, la latenza massima della discovery è significativamente maggiore rispetto a quella che è fornita da Searchlight in situazione di singola frequenza.

All'interno di questo capitolo si andrà a vedere come avviene la scoperta del dispositivo presente all'interno della rete, ovvero come il dispositivo entrante effettua un sondaggio multifrequenza.

Nel seguito, si studierà il valore medio teorico della scoperta considerando che il sondaggio delle frequenze avvenga in maniera puramente casuale e tale valore verrà poi confrontato con i risultati ottenuti dalle realizzazioni. Inoltre, si discuterà di come le molteplici varibili dello scenario impongono un grande numero di realizzazioni per poter avere dei risultati interessanti e confrontabili con il valore medio teorico determinato. Infine si confronteranno, fissando un numero di frequenze utilizzabili dalla rete, le latenze della scoperta al variare del consumo energetico, ovvero del duty-cycle, che nel nostro studio viene definito in base alla scelta del valore del periodo temporale t.

3.1. Sondaggio multifrequenza

Un modo per pensare a come avviene la discovery in tale scenario è quello di immaginare un sistema di riferimento cartesiano bidimensionale dove l'asse delle ascisse rappresenta il tempo, in particolare ci interesserà valutare l'hyper-periodo di ogni dispositivo, mentre

l'asse delle ordinate è la frequenza utilizzata da un dispositivo durante l'hyper-periodo relativo. All'interno del sistema cartesiano qui definito possiamo immaginare la presenza dell'hyper-periodo del dispositivo presente in rete, alla quota relativa alla frequenza utilizzata. Per effettuare il sondaggio, il device entrante nella rete andrà a sondare ogni frequenza a lui disponibile e confrontare il proprio hyper-periodo di rifermiento con quello del dispositivo presente in rete. Ovviamente se la frequenza sondata è diversa rispetto a quella del device in rete, tale sondaggio è fittizzio in quanto la quota relativa alla frequenza utilizzata non sarà la stessa per i due dispositivi che cercano di trovarsi tramite la tecnica di sovrapposizione degli slot ampiamente discussa all'interno del Capitolo 2.

In altri termini, la scoperta del dispositivo presente nella rete da parte del dispositivo entrante avviene come segue :

- Viene scelta una frequenza f_i che puo' essere utilizzata all'interno della rete.

- Il dispositivo entrante inserisce il suo hyper-periodo alla quota relativa alla frequenza f_i attualmente sondata, con i relativi slot attivi e in stato di sleep seguendo il pattern definito dal protocollo Searchlight, discusso nel capitolo precendente.

- Se la frequenza f_i è uguale alla frequenza f_A utilizzata dal dispositivo presente in rete, i due hyper-periodi relativi si sovrappongo per $T - \Phi$ slots, dove abbiamo definito con Φ l'offset tra i due nodi che si devono scoprire. A questo punto, avviene la discovery esattamente come definita nel Capitolo 2 con l'ausilio di Searchlight.

- Se la frequenza f_i è diversa dalla frequenza f_A , allora l'hyper-periodo del dispositivo entrante si troverà ad una quota diversa rispetto all'hyper-periodo del device presente in rete, che si trova alla quota f_A. Questo comporta che durante tutta la durata l'hyper-periodo non avverrà alcuna sovrapposizione fra gli slot, attivi e inattivi, dei due dispositivi, e quindi la latenza totale della scoperta viene aumentata di un'hyper-periodo T. A questo punto, il dispositivo entrante tornerà al passo 1 della scoperta, ovviamente non sondando piu le frequenze precedentemente sondate e dove la scoperta non ha avuto successo.

3.2. Studio del valore medio teorico

Come definito in precendeza, si suppone di avere un dispositivo presente all'interno della rete che chiamiamo Device A. Il device A è in ascolto sulla frequenza f_A. Il Device B, definito come il dispositivo entrante nella rete e che deve cercare il device A, ha un'insieme di frequenze utilizzabili che definiamo come F. La cardinalità di questo

insieme F è definita dal numero di frequenze disponibili all'utilizzo per il device A. Questo numero viene chiamato Nf.

Ovviamente, la frequenza f_A appartiene all'insieme predefinito F. All'interno di questo paragrafo 3.2. si andrà a determinare il valore atteso della durata della scoperta. Le variabili che influiscono sulla durata della scoperta sono sostanzialmente tre. La prima variabile è il valore del periodo, t, che viene scelto dai dispositivi. Si noti che per ottenere dei risultati interessanti, si suppone che i device siano in situazione sincrona e simmetrica. La seconda variabile è ovviamente il numero di frequenze Nf in quanto esso determina il numero massimo di frequenze da sondare. Se esso risulta essere troppo alto, la discovery potrebbe essere molto lunga nel caso in cui la frequenza f_A sia una delle ultime, se non l'ultima ad essere sondata. La terza ed ultima variabile, è l'offset tra i due devices.

Come visto nel Capitolo 2, l'offset fra i devices è molto influente sulla durata della discovery, pero' non è un valore deterministico. Pertanto non possiamo determinare a priori la durata della discovery quando la frequenza sondata è f_A. Quindi, si supporrà che quando la frequenza sondata è diversa da f_A la latenza nella discovery sarà pari al valore di un'hyper-periodo, vale a dire T = $\llcorner \frac{t}{2} \lrcorner$ periodi, o meglio T = $\llcorner \frac{t}{2} \lrcorner$ * t slots , valore deterministico. Mentre se la frequenza sondata è f_A, la scoperta avviene entro un valor medio pari a T = $\frac{\llcorner \frac{t}{2} \lrcorner}{2}$ periodi , ovvero T = $\frac{\llcorner \frac{t}{2} \lrcorner}{2}$ * t slots.

Per determinare la durata media della scoperta del device A da parte del device B, si hanno Nf frequenze da sondare.

Si definiscono le seguenti probabilità

P(I) : la probabilità che le frequenza f_A venga sondata all'i-esimo test di frequenza, con i che varia da 1 a Nf.

P($\bar{I}$) = 1 - P(i) : la probabilità che la frequenza sondata all'i-esimo test di frequenza sia diversa dalla frequenza utilizzata dal device A, cioè f_A.

Al primo sondaggio di frequenza, il dispositivo entrante in rete ha una probabilità di sondare a primo tentativo la frequenza f_A pari a

$$P(1) = \frac{1}{Nf}$$

da cui possiamo facilemente calcolare che la probabilità che la frequenza sondata al primo test sia diversa da f_A è

$$1 - P(1) = 1 - \frac{1}{Nf} = \frac{Nf-1}{Nf}$$

Analogamente, al secondo test di frequenza, si ha che

$$P(2) = \frac{1}{Nf-1}$$

mentre per quanto riguarda la probabilità di non sondare la frequenza f_A al secondo sondaggio si ottiene

$$1 - P(2) = 1 - \frac{1}{Nf-1} = \frac{Nf-2}{Nf-1}$$

Se estendiamo questo ragionamento per tutti i Nf sondaggi possibili, otteniamo i seguenti valori di probabilità

$$P(I) = \frac{1}{Nf - i + 1} \tag{3.1}$$

, per quanto riguarda la probabilità di sondare f_A

$$P(\bar{I}) = \frac{Nf-i}{Nf - i + 1} \tag{3.2}$$

, se la frequenza sondata non è f_A.

Poniamo inoltre che l'hyper-periodo T è determinato dal numero medio di slots, e non di periodi, ovvero

$$T = \lfloor \frac{t}{2} \rfloor * t \quad \text{slots}$$

Definite le probabilità P(I) e P($\bar{I}$) possiamo adesso effetuare una media pesata del tempo di scoperta del vicino. In particolare, se si sondasse come prima frequenza proprio la frequenza f_A il contributo di ritardo sarebbe pari a

$$P(1) * (\frac{T}{2})$$

Mentre se la frequenza f_A fosse sondata al secondo tentativo, il contributo del ritardo risulterebbe il seguente

$$(1 - P(1)) * (P (2)) * (1 * T + \frac{T}{2})$$

Estendendo nuovamente questo contributo fino alle Nf frequenze, otteniamo che il tempo di attesa medio per effettuare la scoperta del dispositivo A alla frequenza f_A è il seguente

$$E[D] = \sum_{j=1}^{Nf} \quad [\prod_{i=1}^{j-1} \quad P(\bar{I}) * P(J) * ((i - 1) * T + \frac{T}{2})] \tag{3.3}$$

L'equazione (3.3) è fondamentale al fine di calcolare il valore atteso della latenza per la scoperta del vicino. Se pero' ci soffermiamo sulla produttoria presente nell'equazione, ci rendiamo conto che per i = 1, il contributo è pari a $\frac{1}{Nf} . \frac{T}{2}$, per i = 2 tale contributo è dato da $\frac{Nf-1}{Nf} * \frac{1}{Nf-1} *(\frac{T}{2} + T) = \frac{1}{Nf} .(\frac{T}{2} + T)$. Se osserviamo con attenzione il prodotto delle probabilità di non avere sondato la frequenza f_A per i primi i – 1 tentativi e della probabilità di aver sondato f_A all'i-esimo tentativo, ci rendiamo conto che tale produttoria è semplicemente pari a $\frac{1}{Nf}$ in quanto frazioni che si vanno a moltipicare si annullano con la precedente.

Inoltre, il valore che viene sommato ad ogni cambio di frequenza, vale dire per ogni j, $((i - 1) * T + \frac{T}{2})$ fornisce sempre un risultato pari a $\frac{Nf^2 * T}{2}$. Questa condiserazione fornisce un'equazione molto semplice che determina il valore atteso per la discovery,

senza dover valutare le probabilità relative al test di frequenza. Abbiamo quindi ottenuto che :

$$E[D] = \frac{1}{Nf} * \frac{Nf^2 * T}{2} \quad (3.4)$$

$$E[D] = \frac{Nf * T}{2} \quad (3.5)$$

Questa equazione 3.5 è quella che verrà usata per determinare il valore atteso della durata della scoperta di un singolo dispositivo vicino durante il resto dello studio.

3.2.1. Esempio numerico

Per mostrare come l'equazione 3.5 sia equivalente all'equazione 3.3, consideriamo l'esempio in cui Nf = 5 frequenze, con entrambi i dispositivi, device A in rete e device B entrante in rete, con durata del periodo temporale pari a t = 8 slots. Pertanto si ha che

$$T = \frac{t}{2} * t = \frac{8}{2} * 8 = 32 \text{ slots} \quad ; \quad \frac{T}{2} = 16 \text{ slots}$$

Si elencano le probabilità P(i) per ogni i, la probabilità che la i-esima frequenza sondata sia la frequenza utilizzata dal dispositvo A, f_A.

$$P(1) = \frac{1}{5} \; ; \; 1 - P(1) = \frac{4}{5}$$

$$P(2) = \frac{1}{4} \; ; \; 1 - P(2) = \frac{3}{4}$$

$$P(3) = \frac{1}{3} \; ; \; 1 - P(3) = \frac{2}{3}$$

$$P(4) = \frac{1}{2} \; ; \; 1 - P(4) = \frac{1}{2}$$

$$P(5) = 1 \; ; \; 1 - P(5) = 0$$

Sfruttando quindi la sommatoria 3.3 per determinare il tempo medio della latenza, otteniamo

$$E[D] = P(1) * \frac{T}{2} + (1 - P(1)) * P(2) * (\frac{T}{2} + T) + (1 - P(1)) * (1 - P(2)) * P(3) * (\frac{T}{2} + 2T) + (1 - P(1)) * (1 - P(2)) * (1 - P(3)) * P(4) * (\frac{T}{2} + 3T) + (1 - P(1)) * (1 - P(2)) * (1 - P(3)) * (1 - P(4)) * P(5) * (\frac{T}{2} + 4T)$$

$$E[D] = \frac{1}{5} * \frac{T}{2} + \frac{4}{5} * \frac{1}{4} * \frac{3T}{2} + \frac{4}{5} * \frac{3}{4} * \frac{1}{3} * \frac{5T}{2} + \frac{4}{5} * \frac{3}{4} * \frac{2}{3} * \frac{1}{2} * \frac{7T}{2} + \frac{4}{5} * \frac{3}{4} * \frac{2}{3} * \frac{1}{2} * 1 * \frac{9T}{2}$$

$$= \frac{1}{5} * (\frac{T}{2} + \frac{3T}{2} + \frac{5T}{2} + \frac{7T}{2} + \frac{9T}{2})$$

$$= \frac{1}{5} * (\frac{25T}{2}) = \frac{1}{Nf} * \frac{Nf^2 T}{2} = \frac{Nf * T}{2}$$

Come ci si attendeva, il tempo medio per effetuare la discovery è pari a E[D] = 80 slots. Nel corso del prossimo paragrafo si valuterà anche questo caso, e si potrà vedere quanto questo valore teorico viene rispettato nella realtà.

3.3. Confronto con il valore atteso teorico

Dopo aver determinato il valore atteso teorico, ci si pone adesso il problema di valutare quanto è attendibile tale valore.

Come ricordato all'inizio del capitolo, c'è una terza variabile che è molto influente sul valore della discovery, esso è l'offset tra i due device. Come visto all'interno del Capitolo 2, se l'offset fra i due devices è pari a 0 slot, la discovery avviene dopo 0 slots, mentre se l'offset è pari a 1 slot, la scoperta ha durata temporanea di un solo slot. Invece se si considera un'offset massimo, la durata della scoperta, cresce sensibilmente.

D'altra parte, un'ulteriore variabile casuale entra in gioco. Questa ulteriore, e non ultima di importanza, è rappresentata dall'ordine in cui vengono sondate le Nf frequenze. E' molto facile da intuire che se la frequenza sondata al primo test, è quella utilizzata dal dispositivo presente in rete, la scoperta sarà molto rapida.

Pensiamo a due situazioni che possono avvenire quando il dispositivo entrante puo' utilizzare Nf = 30 frequenze. La prima è lo scenario nel quale il device B che entra nel sistema ha un'offset con il device A pari a 0 slot, in questo caso, la prima frequenza sondata è la frequenza f_A. In questo primo scenario, molto fortunato, la scoperta tra i due nodi ha una durata temporale pari a 0 slot. La seconda casistica è quella in cui il device B entra nella rete con un'offset temporale pari a $\lfloor \frac{t}{2} \rfloor$ slots. In questa situazione, la frequenza f_A risulterà essere la trentesima frequenza sondata. Il tempo di discovery in questo caso sarà pari a 29*T al quale verrà sommato un valore molto vicino a T in quanto siamo nel peggiore scenario possibile. Questo esempio è stato esposto per dimostrare come il valore atteso teorico in realtà si discoterà dal quello reale se si considera la singola realizzazione, in particolar modo al crescere del periodo temporale t e del numero di frequenze utilizzabili dal dispositivo entrante in rete, Nf. Per raggiungere un valore di latenza per la scoperta vicino al valore atteso, serve pertanto effettuare un grande numero di realizzazioni. Per avere un numero di realizzazioni coerente per il confronto con il valore atteso, si effettuerà per ogni valore di t studiato un numero m di realizzazioni dove m è

$$m = Nf * \sum_{i=1}^{\lfloor \frac{t}{2} \rfloor} i \qquad (3.6)$$

3.3.1. Confronto al variare del numero di frequenze utilizzabili

Come prima tipologia di realizzazioni, si andrà a valutare l'andamento della latenza della discovery per alcuni valori di t, coerentemente scelti per far si che il duty-cycle sia ragionevole, al variare di Nf, ovvero il numero di frequenze disponibili per il device entrante all'interno della rete.

Sfruttando l'equazione (3.5), possiamo definire il valore atteso per ogni valore di Nf, ovvero per ogni numero di frequenze da sondare per il device entrante in rete, e per ogni valore di durata del periodo temporale che verrà scelto.

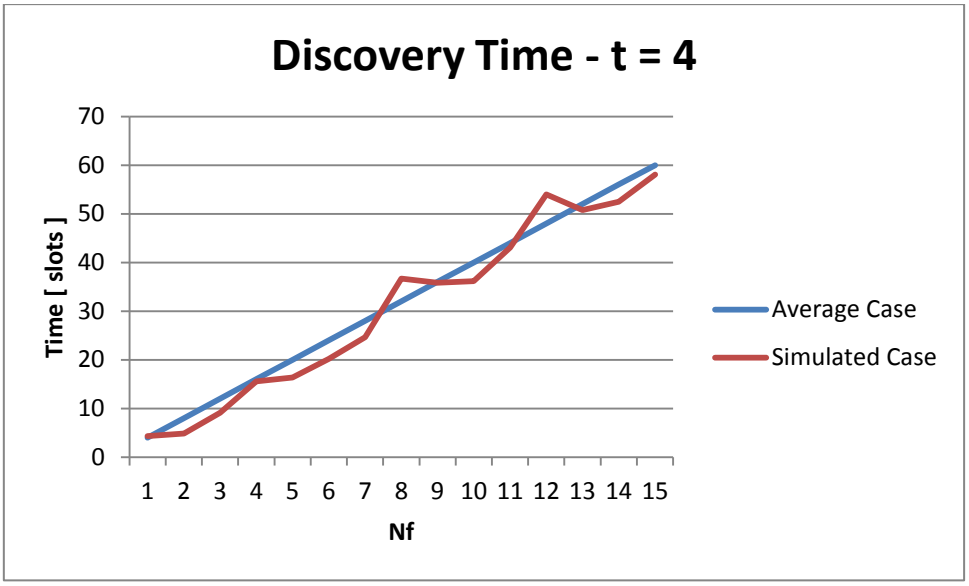

GRAFICO 3.1. Confronto con il valore atteso per t = 4

Come si puo' osservare, per questo primo caso, l'andamento della media delle simulazioni effettuate è molto vicino a quello del valore atteso derivato dall'equazione 3.5. Questo è intuibile dal fatto che avendo un valore del periodo temprale t = 4, l'offset fra i due dispositivi puo' essere pari a 0, 1 o 2 slots solamente, e questo rende i risultati piu' omogenei. Si osserverà per le simulazioni con t maggiore che, il valore atteso teorico sarà comunque molto bene approssimato, in quanto la scelta del numero di realizzazioni da effettuare è stata fatta in modo tale da rendere il meno casuale possibile il contributo alla latenza della scoperta portato dal grande numero di offset possibili.

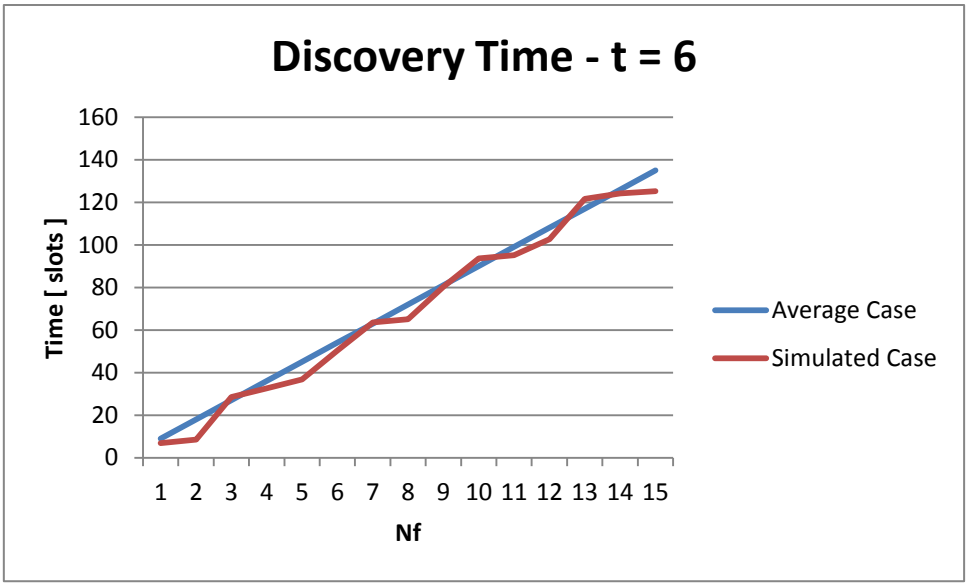

GRAFICO 3.2. Confronto con il valore atteso per t = 6

Simile al caso precendente, anche per t = 6 l'andamento delle simulazioni effettuate è attendibile a quello relativo al valore atteso teorico.

Nella realizzazione seguente, si potrà valutare quanto il valore determinato nell'esempio numerico esposto nel sotto paragrafo 3.1.1. sia rispettato dalle realizzazioni.

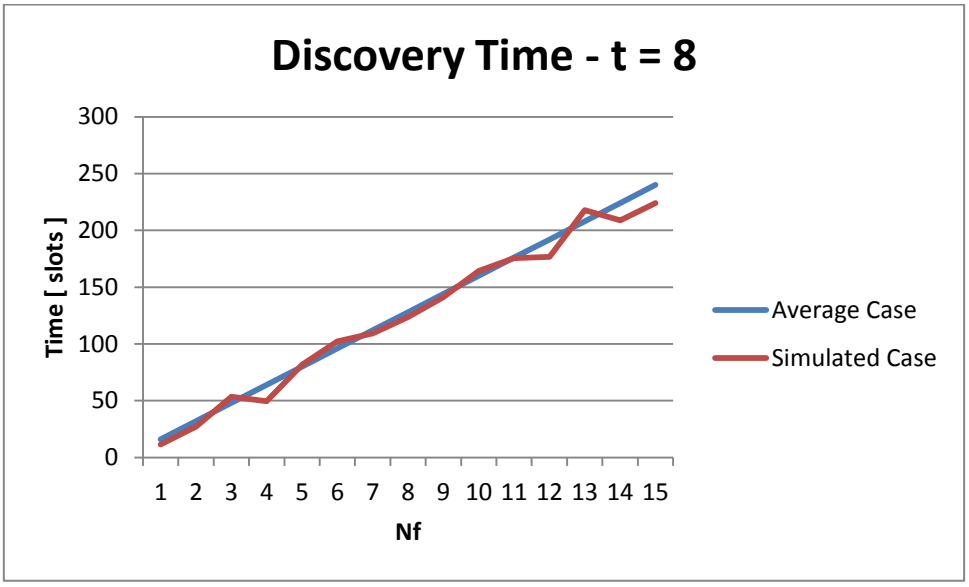

GRAFICO 3.3. Confronto con il valore atteso per t = 8

Come determinato in precendenza, si ha che E[D] = 80 slots per Nf = 5 frequenze, e con t = 8 slots, dalle m = Nf * $\sum_{i=1}^{\lfloor \frac{t}{2} \rfloor} i$ = 5 * (1 + 2 + 3 + 4) = 5 * 10 = 50 realizzazioni, si ottiene un valore mediato pari 81,52. Questo significa che il valore atteso viene rispettato. L'andamento delle simulazioni è molto vicino a quello del valore atteso per valori piccoli di Nf. Mentre al crescere di Nf, l'ordine del sondaggio delle frequenze introduce una

componente casuale sul calcolo del tempo medio di attesa per la discovery molto significativa, questo comporta uno scostamento rispetto al valore atteso.

Nel seguito, si valuteranno delle situazioni con un duty-cycle ragionevolmente basso, quindi un basso consumo energetico che coincide con un valore del periodo temporale t grande, con un numero di realizzazioni diverso. Questo mostrerà come se il numero di campioni di tempi di scoperta a disposizione non sono un numero molto alto, ovvero come visto in precendenza con m = Nf * $\sum_{i=1}^{\lfloor \frac{t}{2} \rfloor} i$ realizzazioni , l'andamento del valore atteso non viene seguito in maniera adeguata.

3.3.2. Confronto al variare del numero di realizzazioni

Come dichiarato nell'equazione 3.6 per le simulazioni precedenti, dove l'andamento del valore atteso è molto vicino a quello delle realizzazioni, il numero di realizzazioni effetutate è stato scelto in modo tale da rendere la casualità delle variabili la piu bassa possibile. All'interno di questa sezione, si andrà a vedere come se il numero di realizzazioni m è scelto in maniera poco accurata, questo puo' portare a dei valori poco attendibili, sopratutto al crescere del valore t e del numero di frequenze Nf a disposizione del dispositivo entrante nella rete. In particolare, si utilizzerà un numero di realizzazioni pari a

$$m_2 = Nf \qquad (3.7)$$

Per primavalutazione, si considera il valore del periodo pari a t = 10. Si andrà in prima analisi a valutare il valore ottenuto con m_1 = Nf * $\sum_{i=1}^{\lfloor \frac{t}{2} \rfloor} i$ realizzazioni e confrontarlo con il valore atteso teorico. In seconda analisi, si valuterà lo stesso scenario, effettuando solamente m_2 = Nf realizzazioni e confrontarlo con il valore ottenuto calcolando il valore atteso, tramite l'equazione 3.5

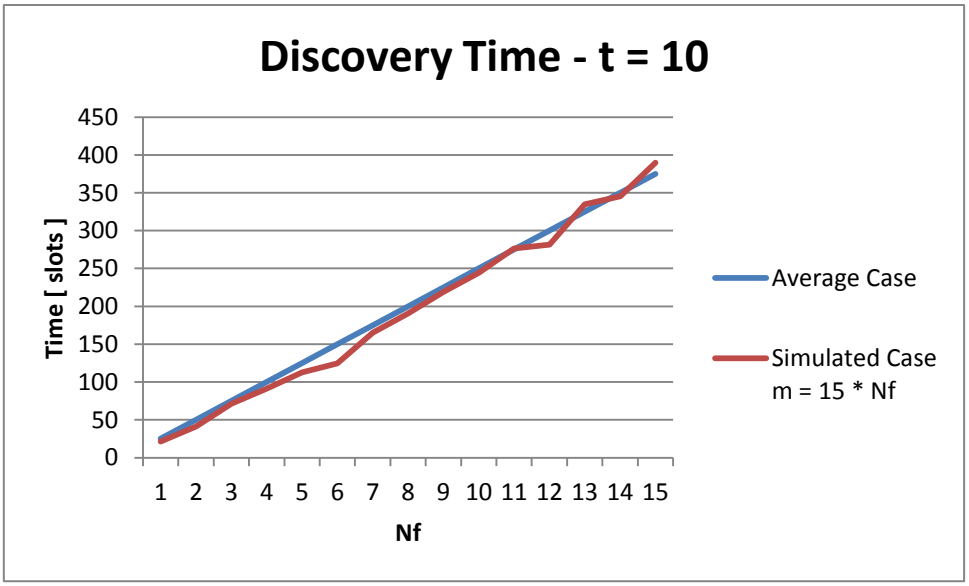

GRAFICO 3.4. Confronto con il valore atteso con t = 10 e m = 15 * Nf

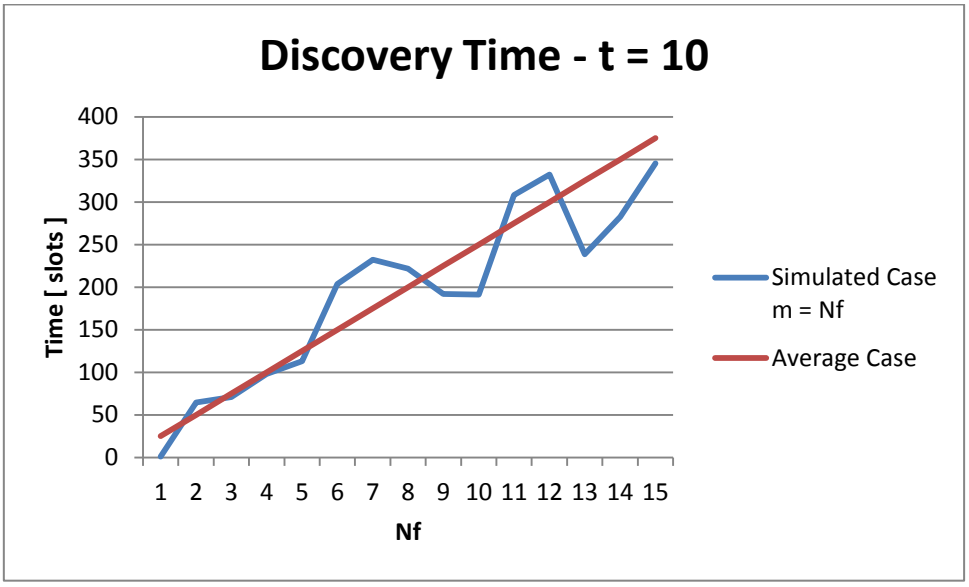

GRAFICO 3.5 Confronto con il valore atteso con t = 100 e m = Nf

Se confrontiamo i grafici 3.4 e 3.5, ci si rende conto molto rapidamente come l'utilizzo di un numero di simulazioni troppo piccolo rende le variabili casuali del processo, vale a dire offset tra i devices e numero di frequenze a disposizione del dispositivo entrante in rete, molto influenti sul risultato delle realizzazioni. Questa considerazione è ancora piu' notevole al crescere del periodo temporale t.

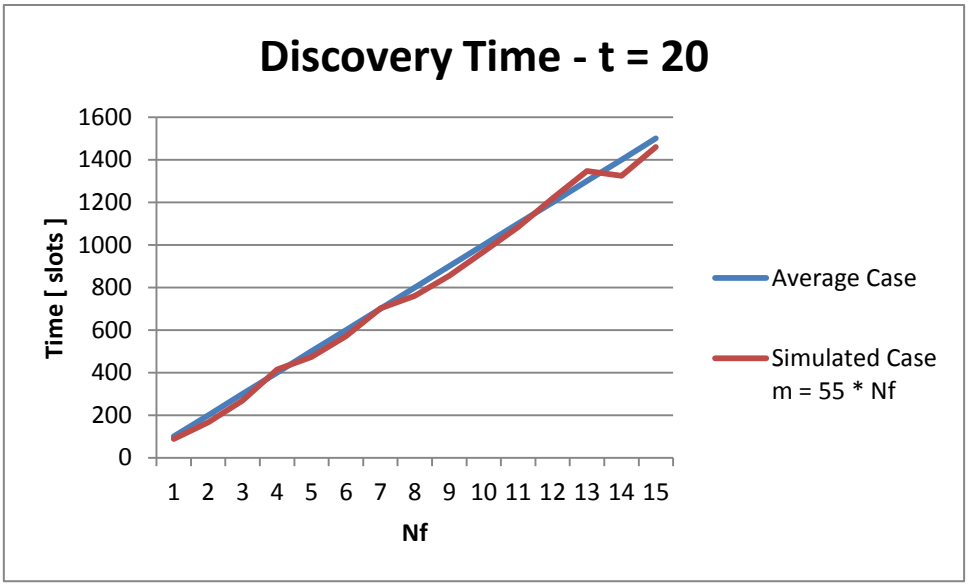

GRAFICO 3.6. Confronto con il valore atteso per t = 20 e m = 55 * Nf

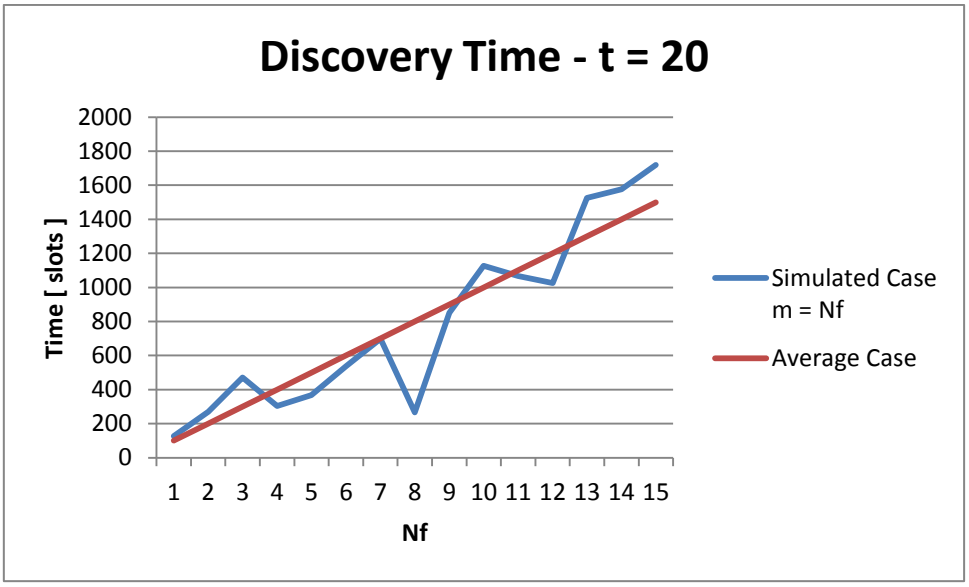

GRAFICO 3.7. Confronto con il valore atteso per t = 20 e m = Nf

Se si osserva il valore atteso per t = 20 e Nf = 8, si calcola un valore atteso pari a

$$E[D] = \frac{Nf*T}{2} = \frac{8*200}{2} = 800 \text{ slots}$$

Mentre se viene considerato il valore ottenuto tramite le 8 realizzazione effettuate per determinare il valor medio dell'attesa, si ha che la discovery avviene dopo 265,25 slots in media. Si ha, per questo punto particolare, che il valore atteso della scoperta è piu di 3 volte piu grande di quello ottenuto. Se si vuole dare una spiegazione a questa situazione,

si puo osservare singolarmente le 8 simulazioni. Si nota che per 6 casistiche, la prima frequenza che viene sondata è f_A. Ovviamente questo renderà il valor medio ottenuto tramite le realizzazioni ampiamente piu basso rispetto al valore atteso.

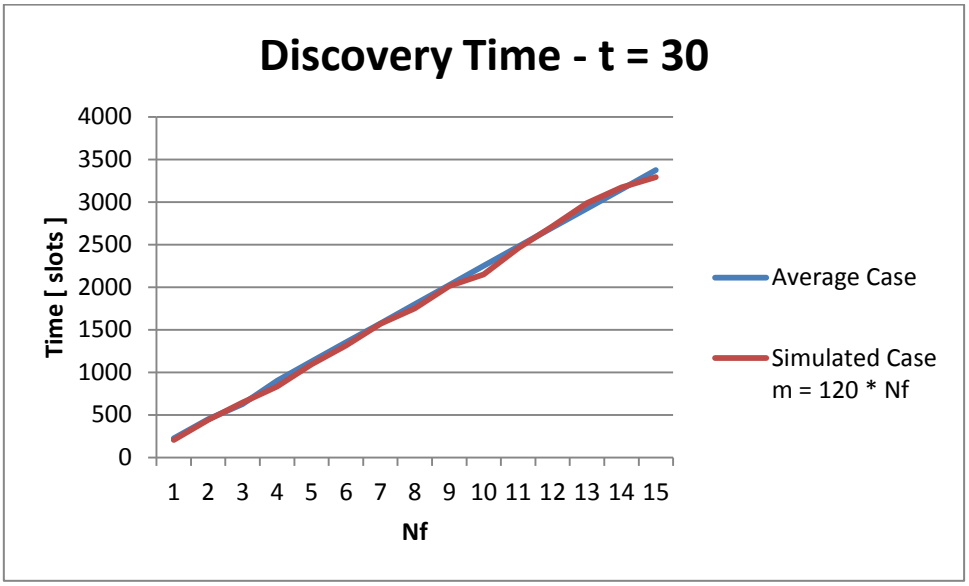

GRAFICO 3.8. Confronto con il valore atteso per t = 30 e m = 120 * Nf

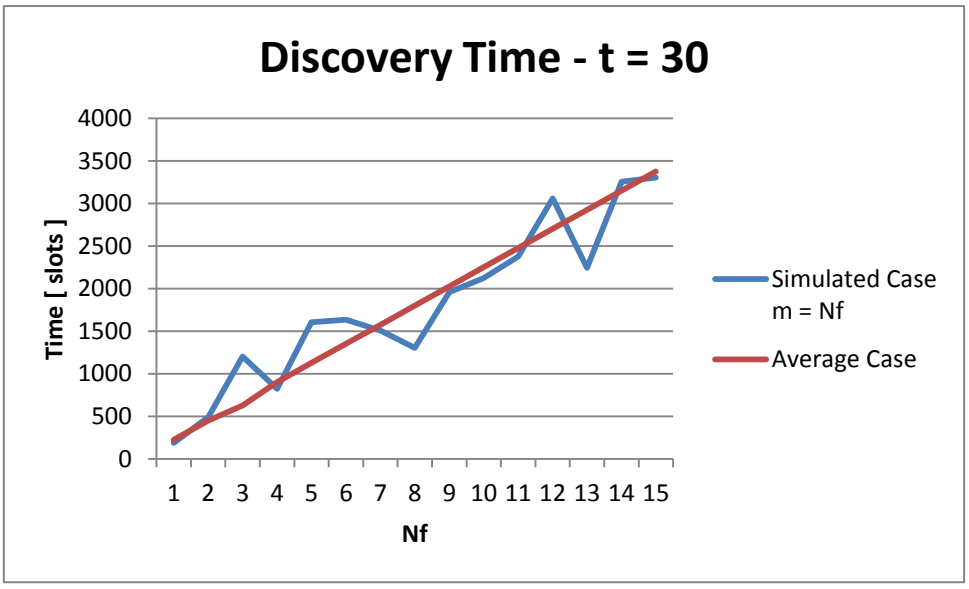

GRAFICO 3.9. Confronto con il valore atteso per t = 30 e m = Nf

Si puo' concludere quindi che la scetlta del numero di simulazioni da effettuare è molto importante, ed essa deve avvenire in maniera molto accurata per poter effettuare una stima del tempo medio della scoperta, in modo tale da poter dimensionare con accuratezza il duty-cycle e per poter gestire numero di frequenze disponibili all'utilizzo per il device entrante in rete.

3.4. Tempo di scoperta al variare del duty-cycle

Per questo paragrafo, si andrà a valutare il tempo medio di scoperta del device B con un numero di frequenze a disposizione Nf prefissato, al variare del valore del periodo temporale t.

Se consideriamo ad esempio una rete Wi-Fi, si ha che il numero di canali di utilizzo è 12. Se ci soffermiamo su questo esempio, il problema posto all'interno di questo paragrafo è quello di valutare al variare del duty-cycle, ricordando che utilizzando Searchlight esso risulta essere

$$\eta = \frac{2}{t}$$

il tempo medio della scoperta, avendo Nf fissato a 12 canali.

Si noti che, come discusso all'intero del sotto paragrafo 3.2.2. il numero di realizzazioni da effettuare per ottenere un valore medio delle realizzazioni adeguato è quello definito dall'equazione 3.6.

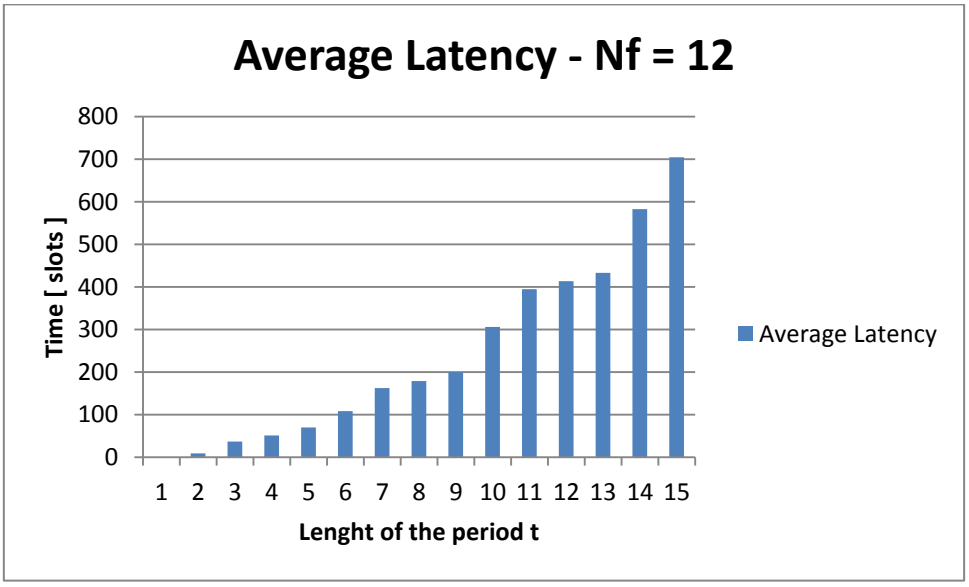

GRAFICO 3.10. Latenza media con Nf = 12

Questo grafico 3.10 dimostra come una scoperta rapida del device B comporta un'attività da parte dei dispositivi molto alta, cioè un valore del periodo temporale basso. Se si osserva la latenza media per t = 14, si vede che consumando poca energia, avendo un duty-cycle pari a $\eta = \frac{2}{14} = \frac{1}{7}$, il tempo di discovery è pari a 582 slots temporali. Si puo'

pertanto concludere che questo istogramma enfatizza la necessità di effetuare una scelta che sia un trade-off intelligente tra consumo energetico e qualità della discovery, che viene valutata in termini di tempo medio di attesa per la scoperta dei due dispositivi della rete.

Si considera adesso un numero di frequenze disponibili meno significativo rispetto a quello di una rete Wi-Fi. In particolare, si pone che Nf = 5. La latenza media in questo scenario viene riportata nel grafico 3.11.

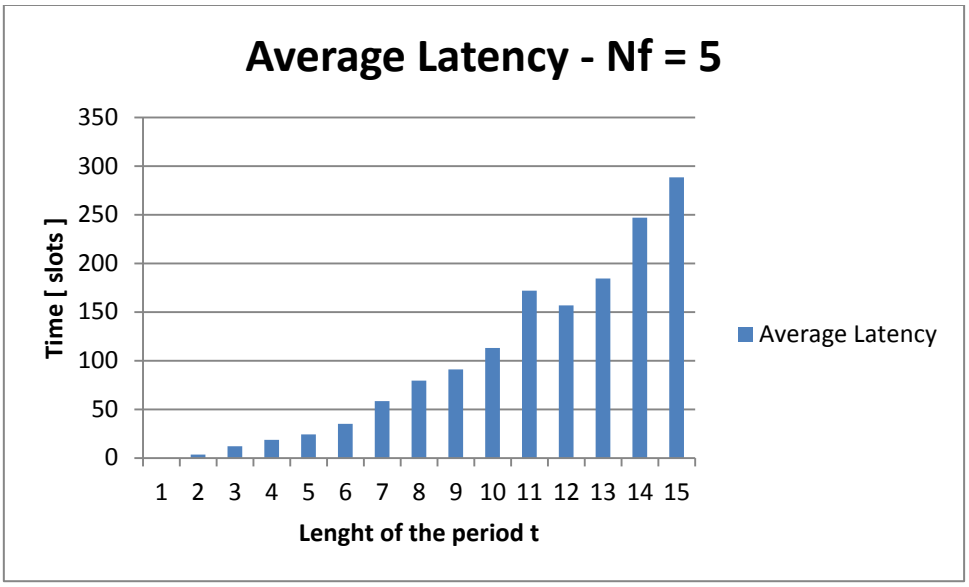

GRAFICO 3.11. Latenza media con Nf = 5

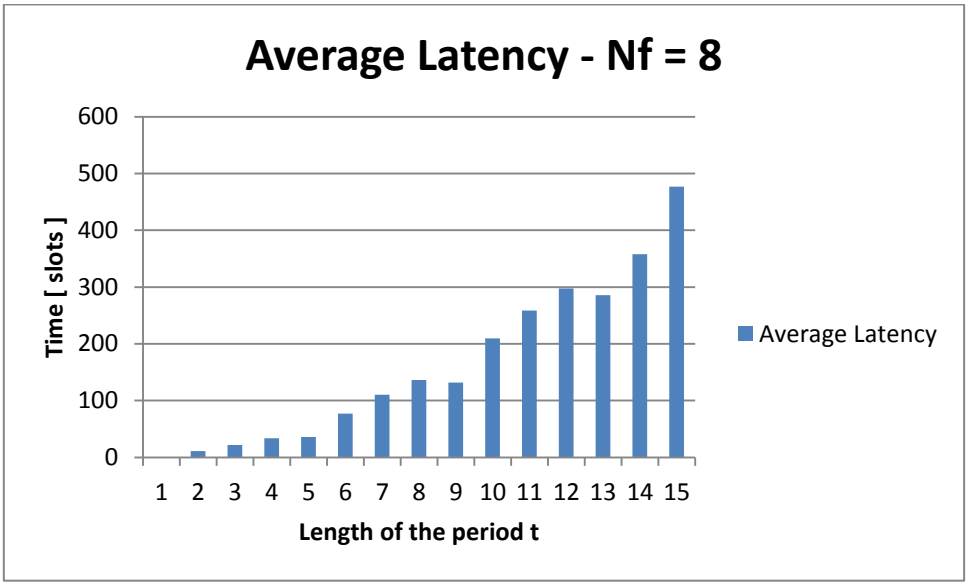

GRAFICO 3.12. Latenza media con Nf = 8

Osservando i risultati di queste simulazioni, ci si rende conto che la crescita della latenza della scoperta del vicino segue un'andamento esponenziale al crescere del valore del periodo temporale. Quindi il prezzo da pagare per ottenere un consumo energetico piu basso è notevole. Per ottenere delle prestazioni adeguate da parte delle applicazioni di

interesse, Searchlight impone un consumo energetico piu basso rispetto ai protocolli di discovery esistenti, come già discusso nei capitoli precendenti, pero' la gestione di questo guadagno energetico deve essere fatta in maniera accurata in quanto la ricerca di un consumo energetico molto basso potrebbe distruggere le prestazioni delle applicazioni, causa un tempo di scoperta dei vari dispositivi molto alto.

All'interno di questo terzo capitolo sono state valutate le prestazioni di Searchlight per la scoperta di un vicino presente in rete e della sua frequenza di utilizzo. Abbiamo visto come la latenza della discovery possa essere difficilmente stimata a priori, causa le molteplici variabili casuali presenti all'interno della rete. Pero' benchè queste variabili cambiano i risultati in termini di latenza in modo significativo da una realizzazione all'altra, si è visto come una scelta accurata del numero di realizzazioni porti ad ottenere un valore medio della latenza molto vicino al valore atteso teorico. Inoltre, abbiamo discusso quanto la gestione del consumo energetico possa influire in maniera sensibile sulle prestazione delle applicazioni che utilizzano il protocollo Searchlight per scoprire la presenza di un vicino.

Nel seguito dello studio, si andrà a parlare della situazione realistica che dovrà affrontare un dispositivo entrante in rete che opera tramite il protocollo Searchlight per effetuare la discovery. Questo scenario è quello in cui un dispositivo entrante in rete deve scoprire un numero di dispositivi a lui non noto a priori, e non come visto fino ad ora che in rete era presente un solo dispositivo. Inoltre, questi dispositivi presenti in rete, funzionano ognuno ad una determinata frequenza. Pertanto, in questo nuovo scenario, il dispositivo che entra nel sistema deve assicurarsi di scoprire tutti i dispositivi già presenti nella rete, oppure una porzione di essi, ed inoltre determinare a quale frequenza essi lavorano.

CAPITOLO 4 - Scoperta di tutti gli utenti della rete

Dopo aver discusso lo scenario in cui un utente entrante nella rete deve scoprire un singolo utente presente in essa, all'interno di questo capitolo si vuole effettuare un'ulteriore estensione. In particolare, si suppone che la rete nella quale il dispositivo entrante vuole far parte sia già composta da un numero predefinito di dispositivi, questo numero viene chiamato Nd nel resto dello studio. Per poter entrare nel sistema, il dispositivo entrante deve scoprire i dispositivi già presenti e la loro rispettiva frequenza di utilizzo.

Le frequenze che vengono utlizzate dai Nd devices presenti nella rete, non sono note a priori al dispositivo entrante. In un primo tempo, si considererà che tali frequenze vengono distribuite uniformemente in maniera casuale ad ognuno dei Nd dispositivi. Mentre in seconda analisi, si supporrà che le frequenze dei Nd utenti rispettino una determinata funzione di distribuzione, dando cosi dei pesi ad ogni frequenza assegnabile. Si vuole valutare il tempo di scoperta della totalità della rete da parte del dispositivo entrante. Inoltre, risulta interessante calcolare il tempo impiegato dal dispositivo entrante per scoprire, non tutti i dispositivi che compongono la rete, bensi una percentuale di essi.

4.1. Valore atteso nel caso uniforme

Come detto nel preambolo del capitolo, nel caso uniforme si ha che le frequenze vengono assegnate ai Nd dispositivi che compongono la rete in maniera totalmente casuale all'interno dell'insieme delle frequenze disponibili Nf.

In sostanza, ci troviamo a valutare la situazione discussa nel capitolo precendente, in particolare nel paragrafo 3.1 dove il dispositivo entrante deve scoprire un'altro dispositivo alla frequenza f_A, estesa a Nd disposivi che lavorano ognuno ad una frequenza f_i. Anche se la frequenza f_i di uno dei Nd devices puo' essere la stessa di un'altro device, mediamente il tempo di latenza della scoperta dell'intero vicinato viene definita come la

latenza E[D], che viene calcolata come la latenza media della scoperta di un singolo dispositivo, precedentemente calcolata come

$$E[D] = \sum_{j=1}^{Nf} \left[\prod_{i=1}^{j-1} P(\bar{I}) * P(J) * ((i - 1) * T + \frac{T}{2}) \right] \quad (4.1)$$

estesa a Nd dispositivi. Pertanto, possiamo scrivere che la latenza media per la totale scoperta dei vicini è definita come

$$E[D] = Nd * \sum_{j=1}^{Nf} \left[\prod_{i=1}^{j-1} P(\bar{I}) * P(J) * ((i - 1) * T + \frac{T}{2}) \right]$$

$$E[D] = Nd * \frac{Nf * T}{2} \quad (4.2)$$

Questa equazione 4.2 è l'equazione che ci permetterà di poter calcolare il valore atteso della scoperta di Nd devices presenti in rete da parte di un device entrante con un'insieme di frequenze disponibili per gli utenti appartenenti alla suddetta rete pari a Nf frequenze.

4.2. Frequenze distribuite uniformemente

Prima di iniziare a valutare i risultati delle simulazioni, soffermiamoci su un punto considerato nel paragrafo 3.2. di questo studio. Quando il dispositivo entrante in rete doveva effetuare la scoperta di un solo device, abbiamo determinato che un numero convenientemente grande di realizzazioni era

$$m = Nf * \sum_{i=1}^{\lfloor \frac{t}{2} \rfloor} i$$

Intuitivamente si potrebbe moltiplicare questo valore per Nd per ottenere dei risultati soddisfacenti. Pero', in questo scenario, si ha che la casualità della scelta della frequenza del dispositivo presente in rete, viene estesa ad ognuno dei Nd devices. Pertanto, si decide che per ottenere un numero di realizzazioni adeguato, si sceglie che

$$m = Nf * Nd * Nf * \lfloor \frac{t}{2} \rfloor \quad (4.3)$$

Questo paragrafo verrà organizzato come segue :

- Si valuterà il tempo medio della scoperta totale dell'intero vicinato del dispositivo entrante nella rete al variare della numero di frequenze disponibili Nf. Il numero massimo di frequenze disponibile viene considerato pari a Nf = 15. E questo verrà valutato per diversi duty-cycles, ovvero per determinati valori del periodo temporale t.

- In seguito, si stimerà il tempo medio della scoperta di una porzione della rete al variare del numero di frequenze disponibili Nf. Questa situazione è molto utile in quanto il dispositivo entrante in rete potrebbe non necessitare di scoprire tutti gli utenti vicini per

poter accedere alle applicazioni d'interesse per lui. Il tempo medio che verrà ricavato, sarà confrontato con quello ottenuto per la scoperta totale della rete.

- Infine, fissando il valore delle frequenze disponibili, si valuterà il tempo di discovery dell'intero vicinato al variare del numero di utenti presenti Nd.

4.2.1. Confronto con il valore atteso al variare del numero di frequenze

Analogamente a quanto fatto nel capitolo 2, andiamo a valutare la latenza della scoperta del vicinato. In questo caso pero', si ha un numero di dispositivi variabile.

Si noti che i valori determinati nel capitolo precendente, as esempio nei Grafici 3.2 e 3.3, rappresentano lo stesso grafico qui riportato per Nd = 1.

Per il primo esempio, con t = 8, andiamo a valutare diversi valori di Nd, e per ognuno di essi si rappresenterà un grafico con il valore atteso teorico ed il valor medio determinato delle simulazioni. In seguito, per altri valori di t, non verrà riportato il valore atteso in quanto esso è facilmente calcolabile tramite la formula 4.2. definita in precendenza.

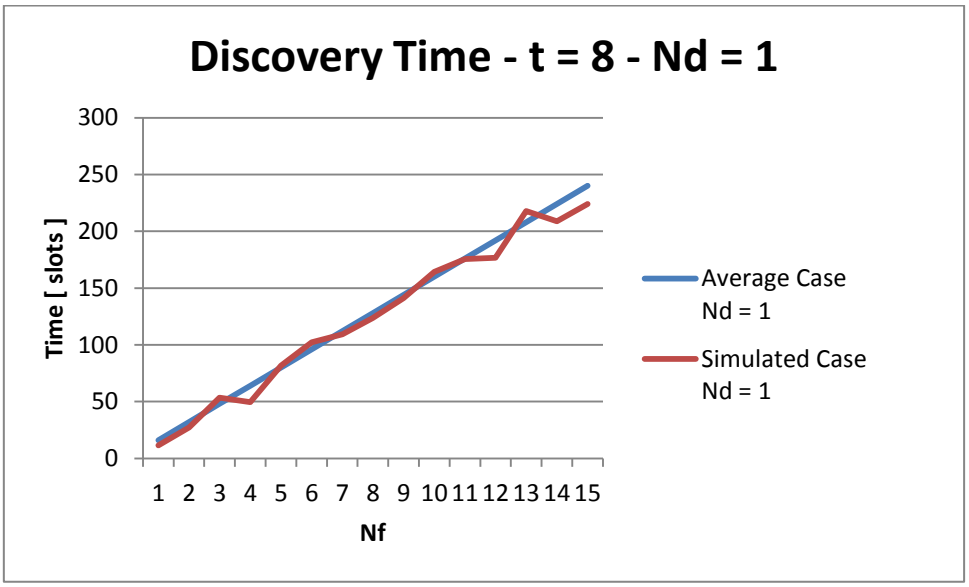

GRAFICO 4.1. Confronto con il valore atteso per t = 8 e Nd = 1

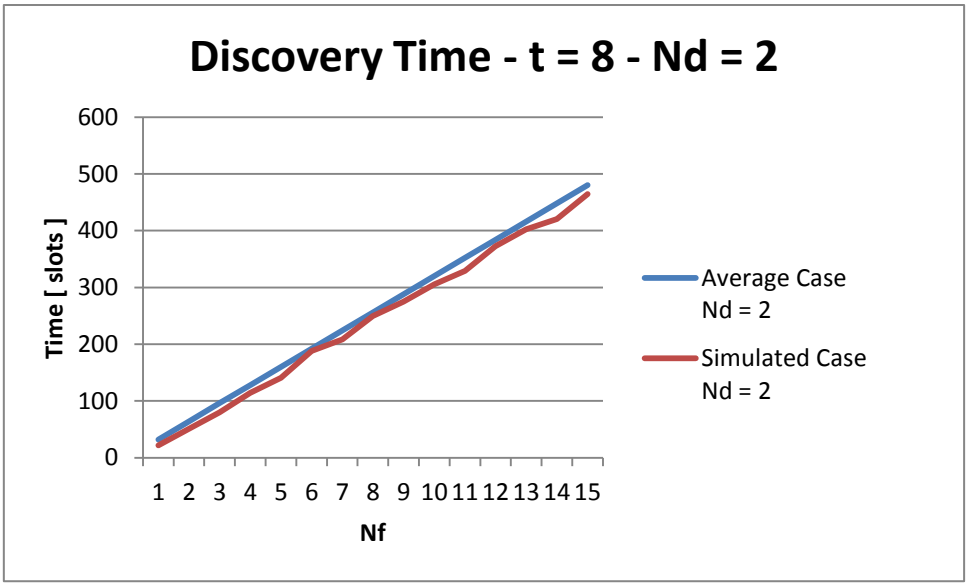

GRAFICO 4.2. Confronto con il valore atteso per t = 8 e Nd = 2

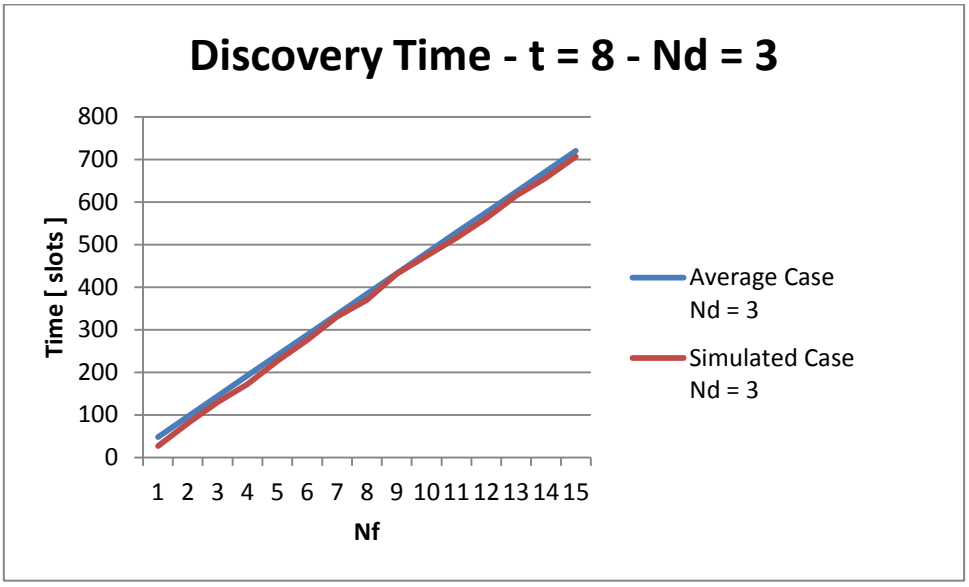

GRAFICO 4.3. Confronto con il valore atteso per t = 8 e Nd = 3

Possiamo notare fin da queste prime realizzazione che il valore atteso viene seguito in maniera molto buona. Quello che si puo' dire è che il valor medio delle simulazioni ha un'andamento sempre sottostante alla curva del valore atteso. Questo è interpretabile sottolineando che nei casi in cui tanti devices utilizzano la stessa frequenza, quando essa è sondata, la latenza totale viene diminuita drasticamente.

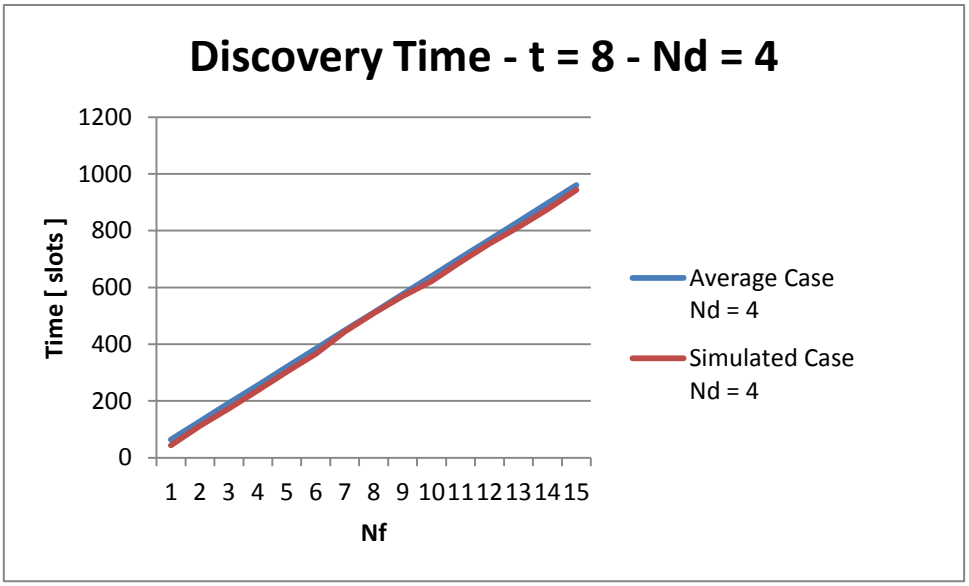

GRAFICO 4.4. Confronto con il valore atteso per t = 8 e Nd = 4

Valutiamo adesso i valori della latenza con un numero maggiore di dispositivi presenti in rete Nd. Nello specifico ci interessiamo a Nd = 10, Nd = 20, Nd = 40 e Nd = 50.

I rispettivi valori della latenza nel caso medio e delle simulazioni vengono riportati in seguito.

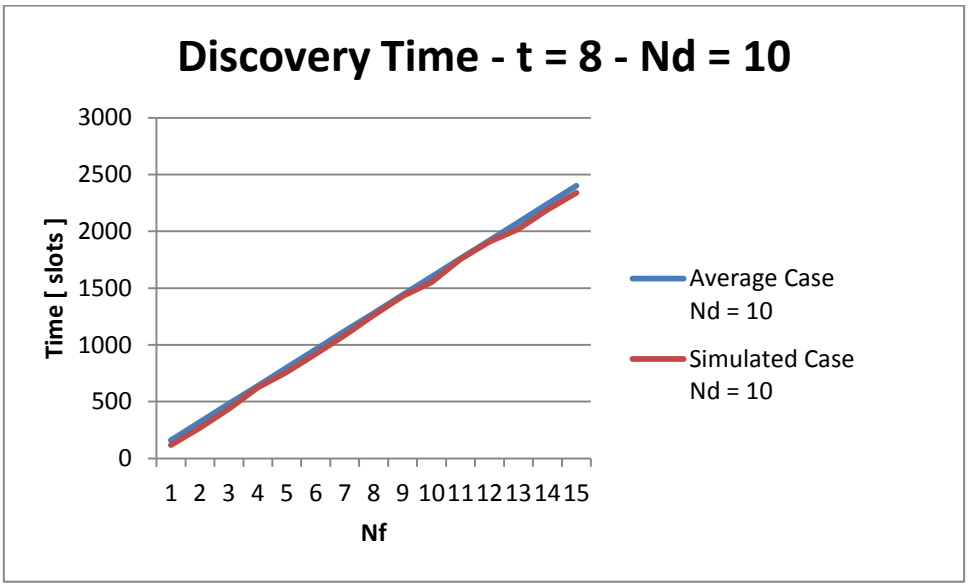

GRAFICO 4.5. Confronto con il valore atteso per t = 8 e Nd = 10

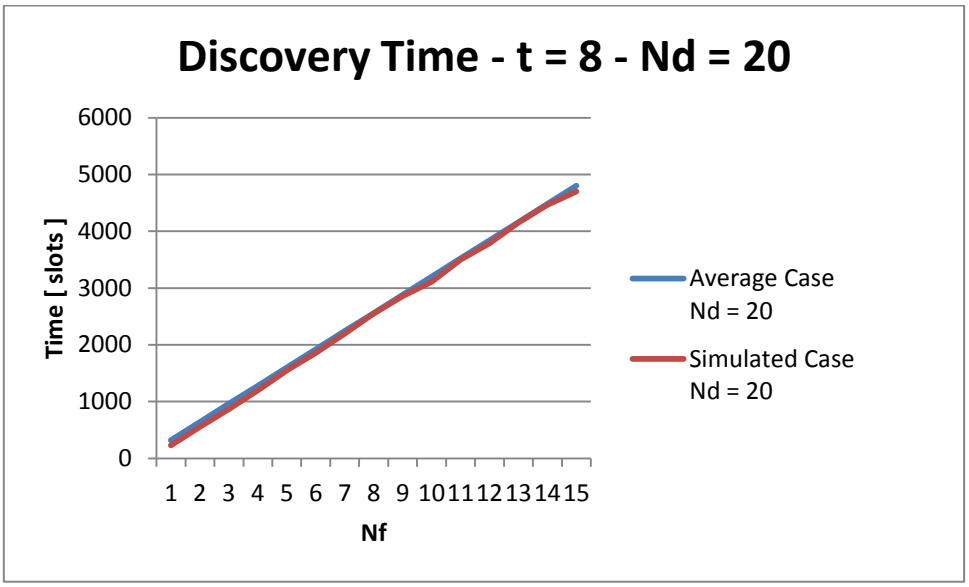

GRAFICO 4.6. Confronto con il valore atteso per t = 8 e Nd = 20

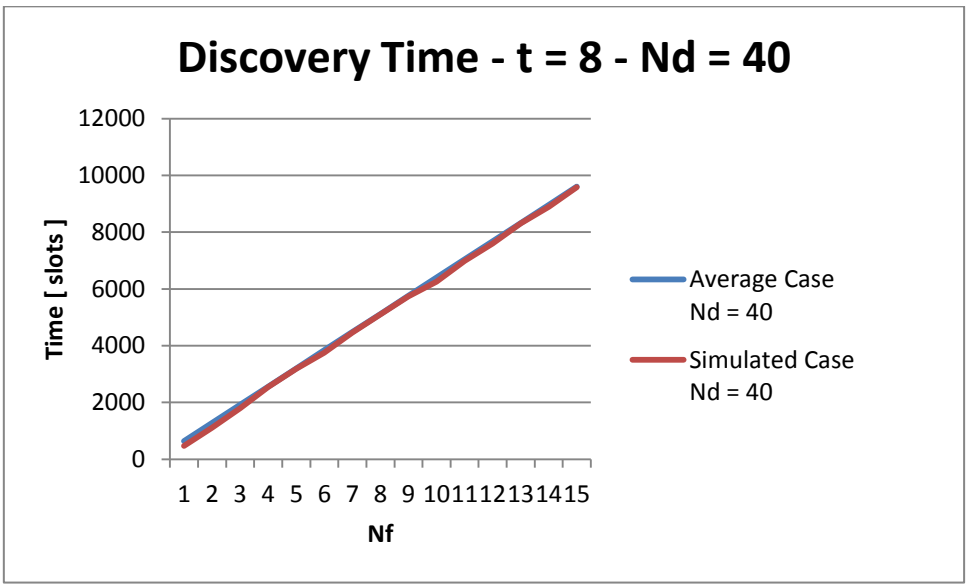

GRAFICO 4.7. Confronto con il valore atteso per t = 8 e Nd = 40

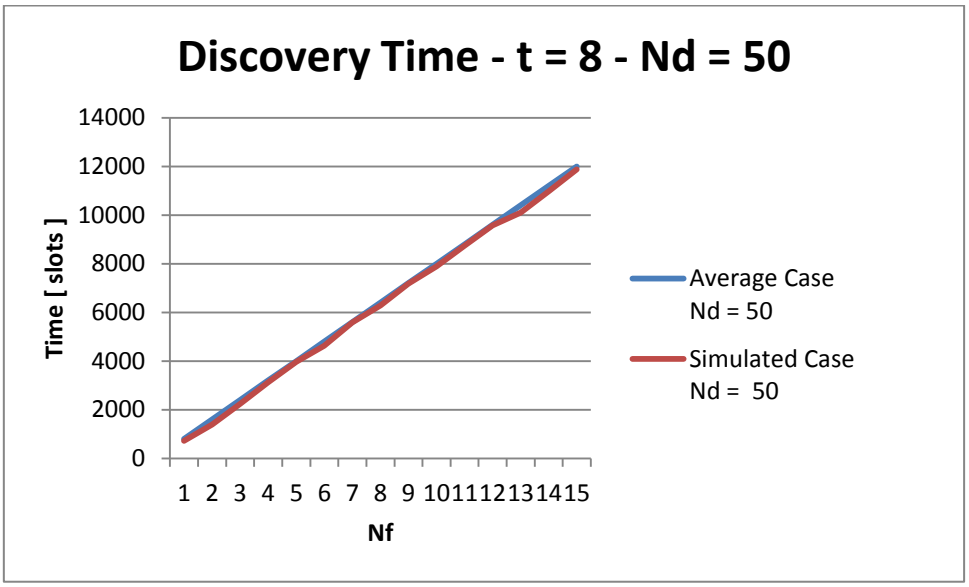

GRAFICO 4.8. Confronto con il valore atteso per t = 8 e Nd = 50

Il valore atteso viene rispettato anche al crescere del numero di dispositivi presenti all'interno della rete Nd. Ricordando il numero di simulazioni definito dall'equazione 4.3, possiamo spiegare come al crescere del numero di dispositivi presenti in rete, le realizzazioni tendono a essere tangenti alla curva determinata dal valore atteso.

I grafici riportati nel seguito, rappresentano la latenza per scoperta di tutti i dispositivi in rete, avendo prefissato il valore di t, sempre al variare di Nf. In questo caso, non si riporterà il valore atteso per ogni valore di Nd

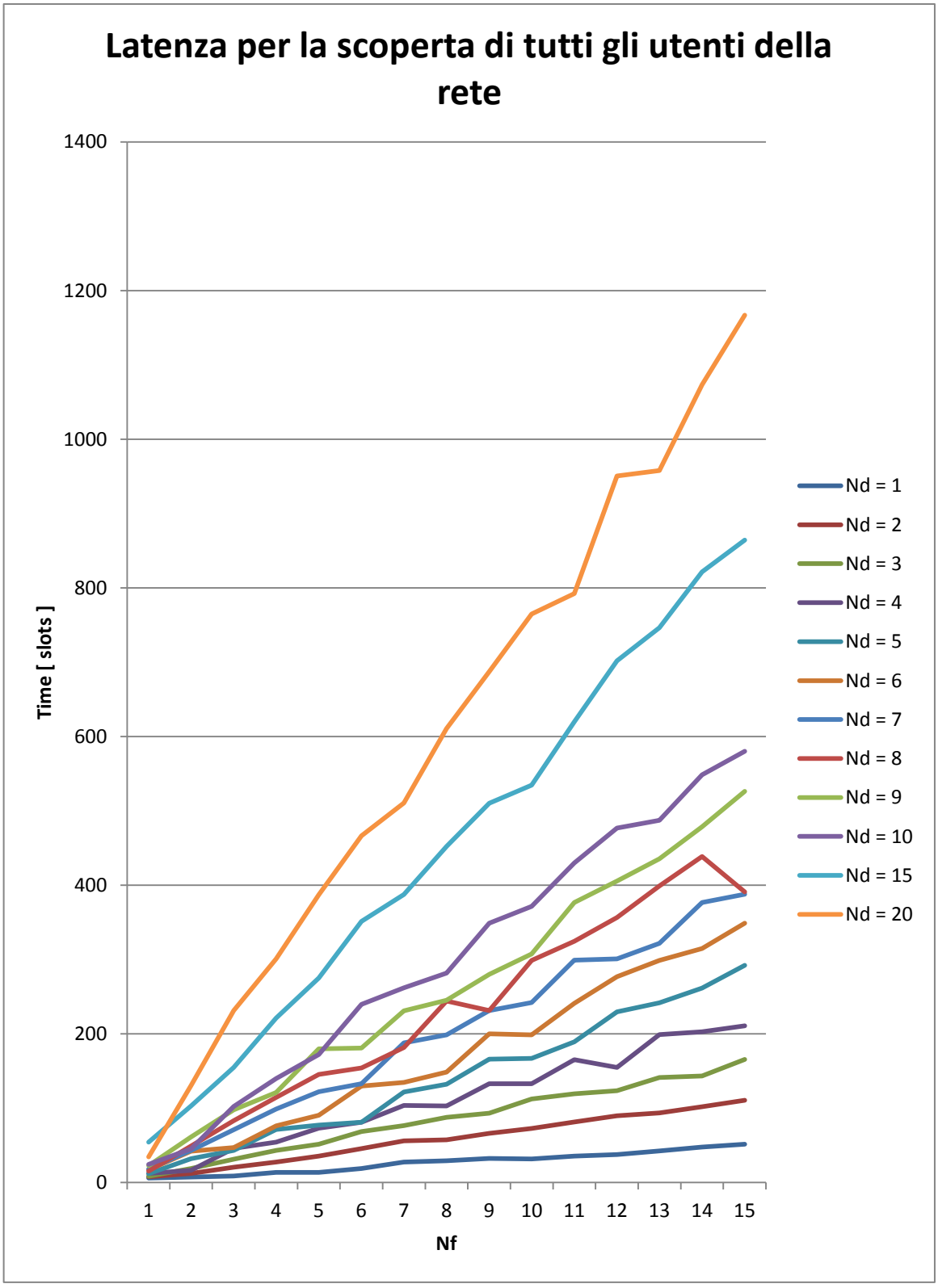

GRAFICO 4.9. Latenza per la scoperta di tutta la rete con t = 4

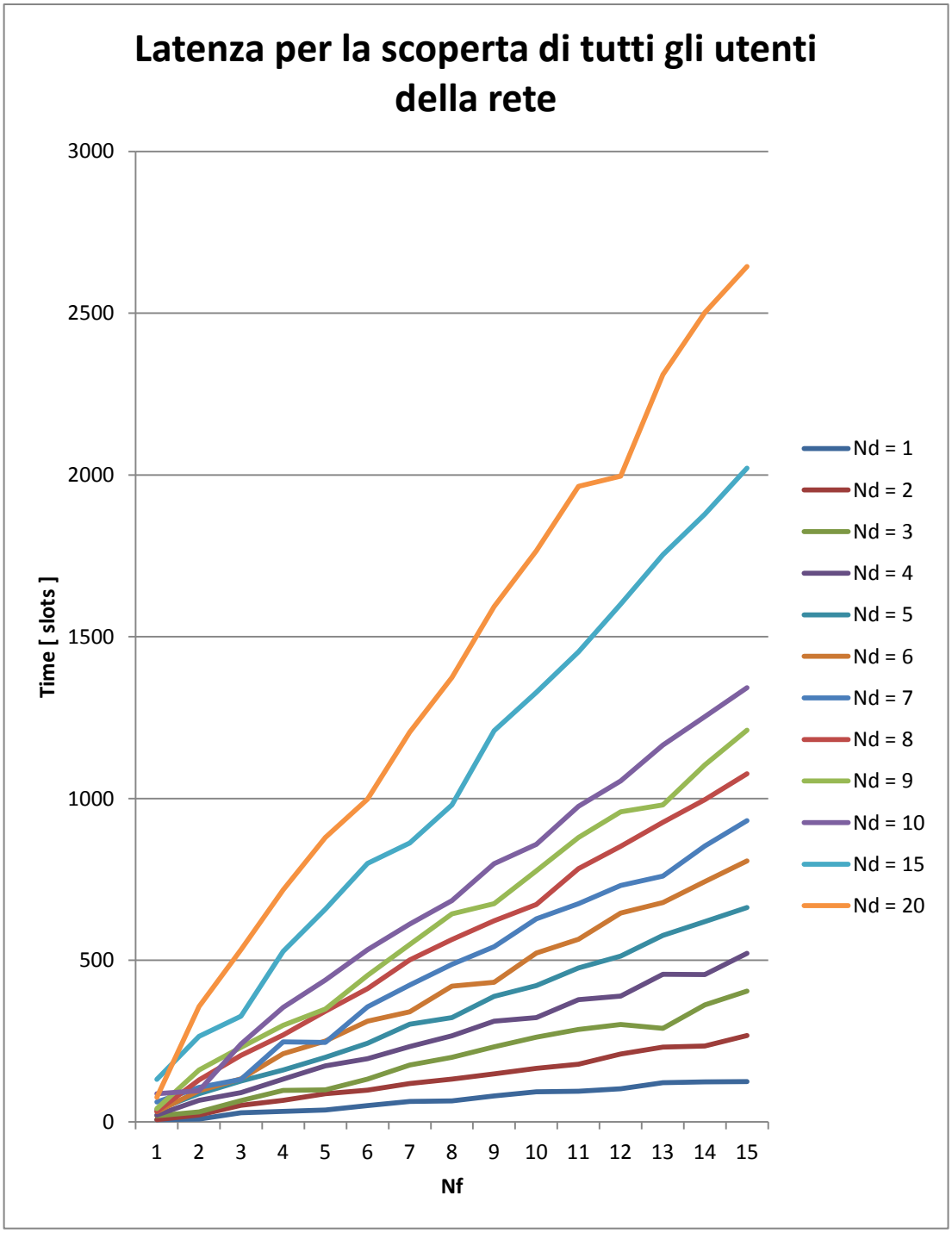

GRAFICO 4.10. Latenza per la scoperta di tutta la rete con t = 6

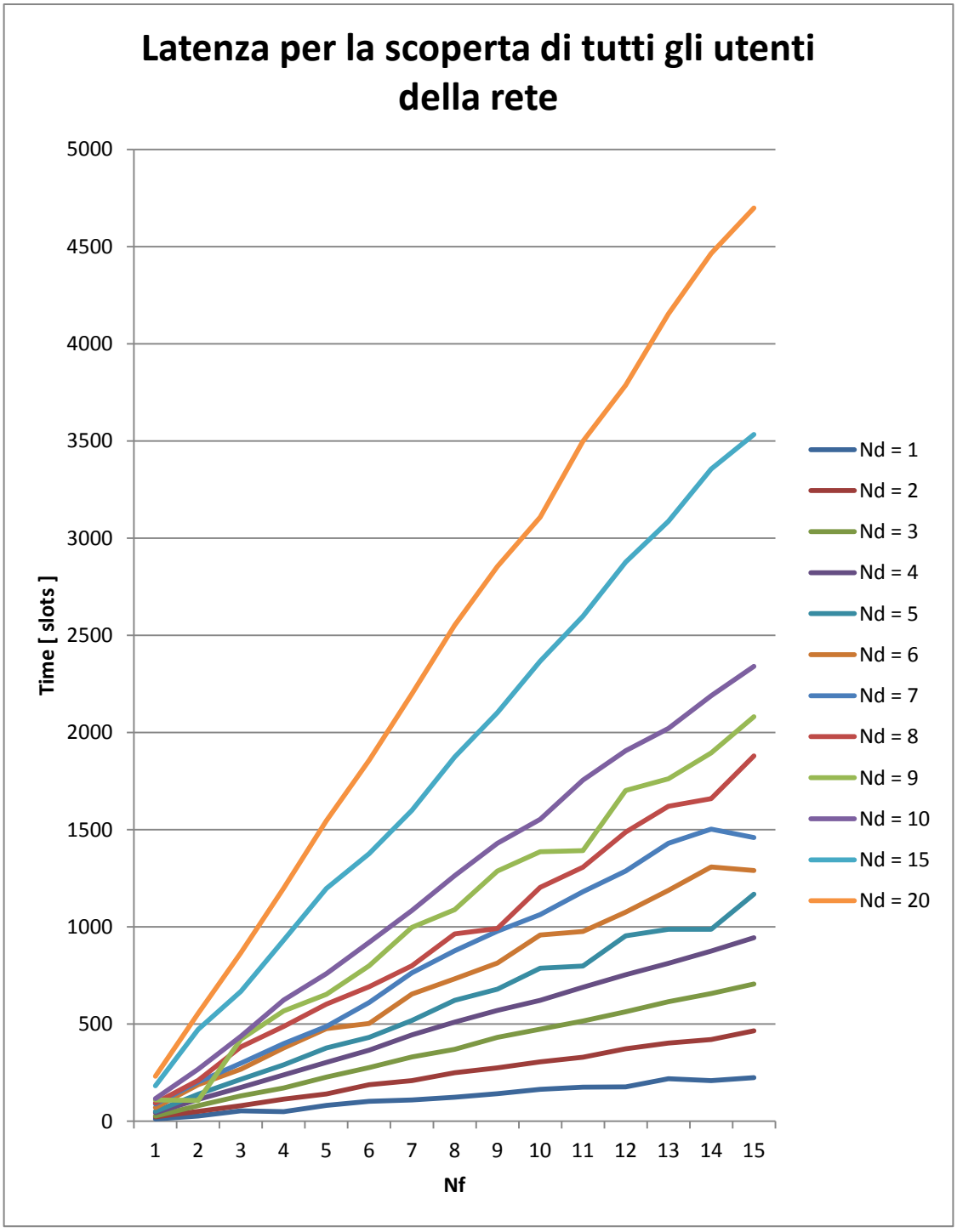

GRAFICO 4.11. Latenza per la scoperta di tutta la rete con t = 8

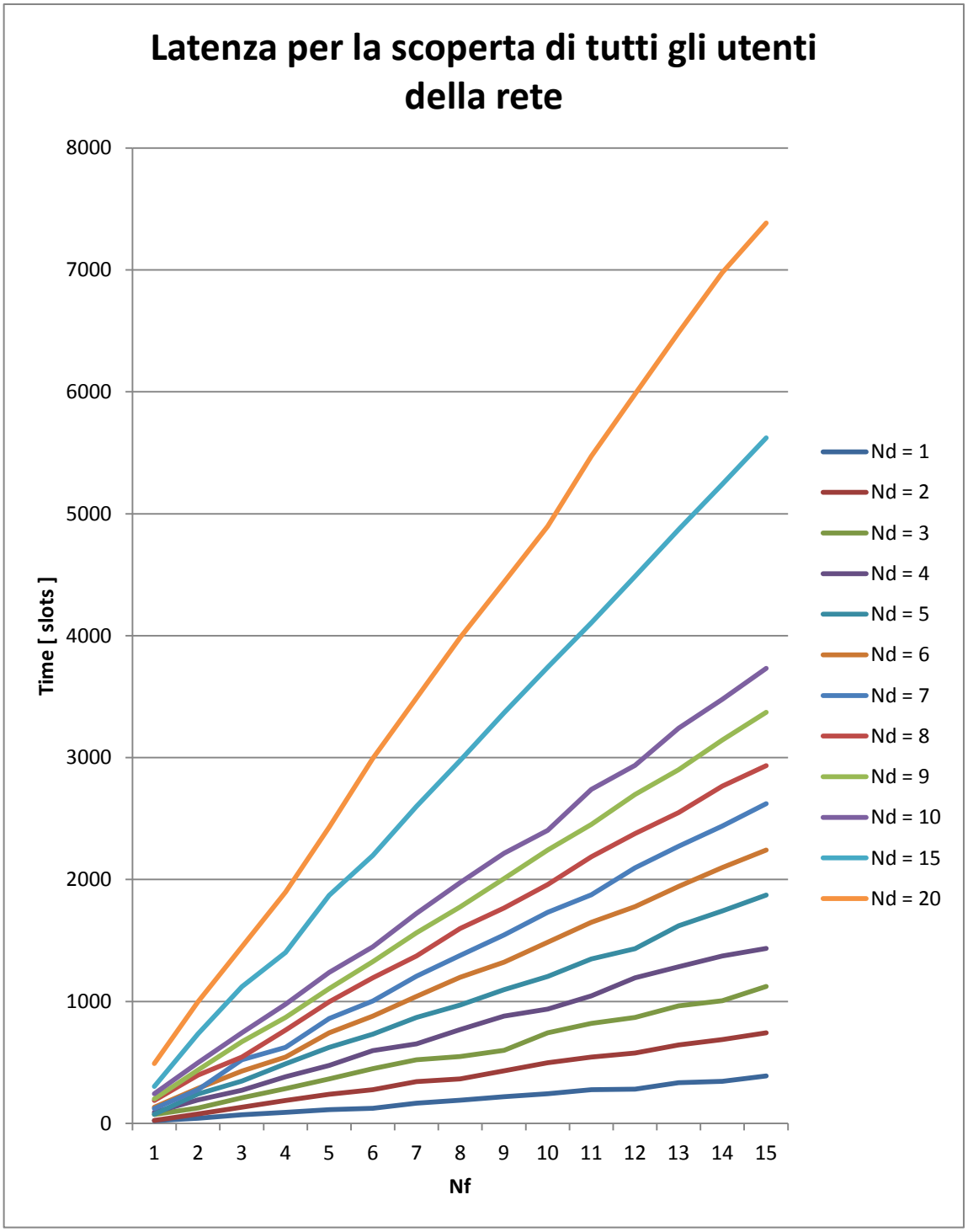

GRAFICO 4.12. Latenza per la scoperta di tutta la rete con t = 10

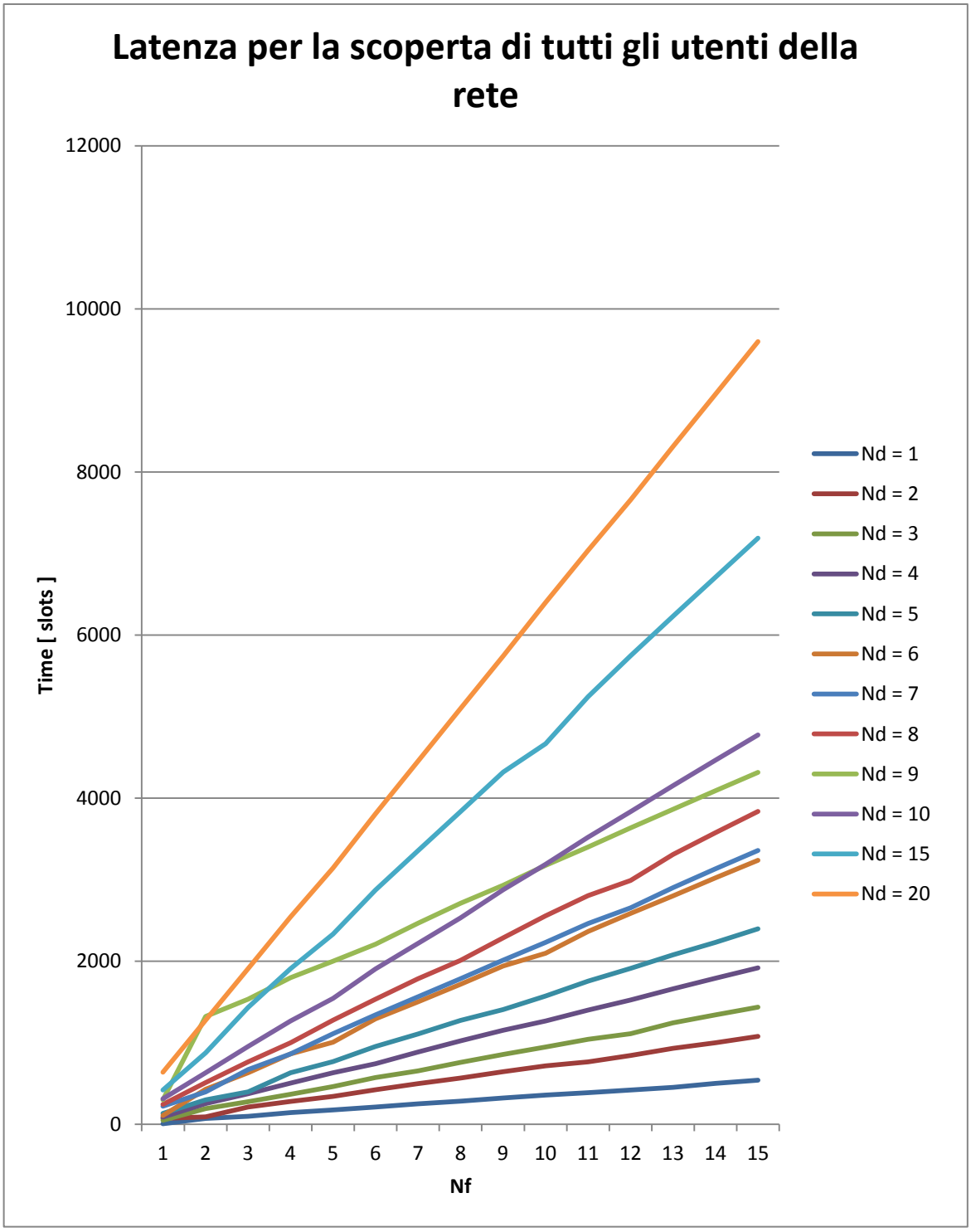

GRAFICO 4.13. Latenza per la scoperta di tutta la rete con t = 12

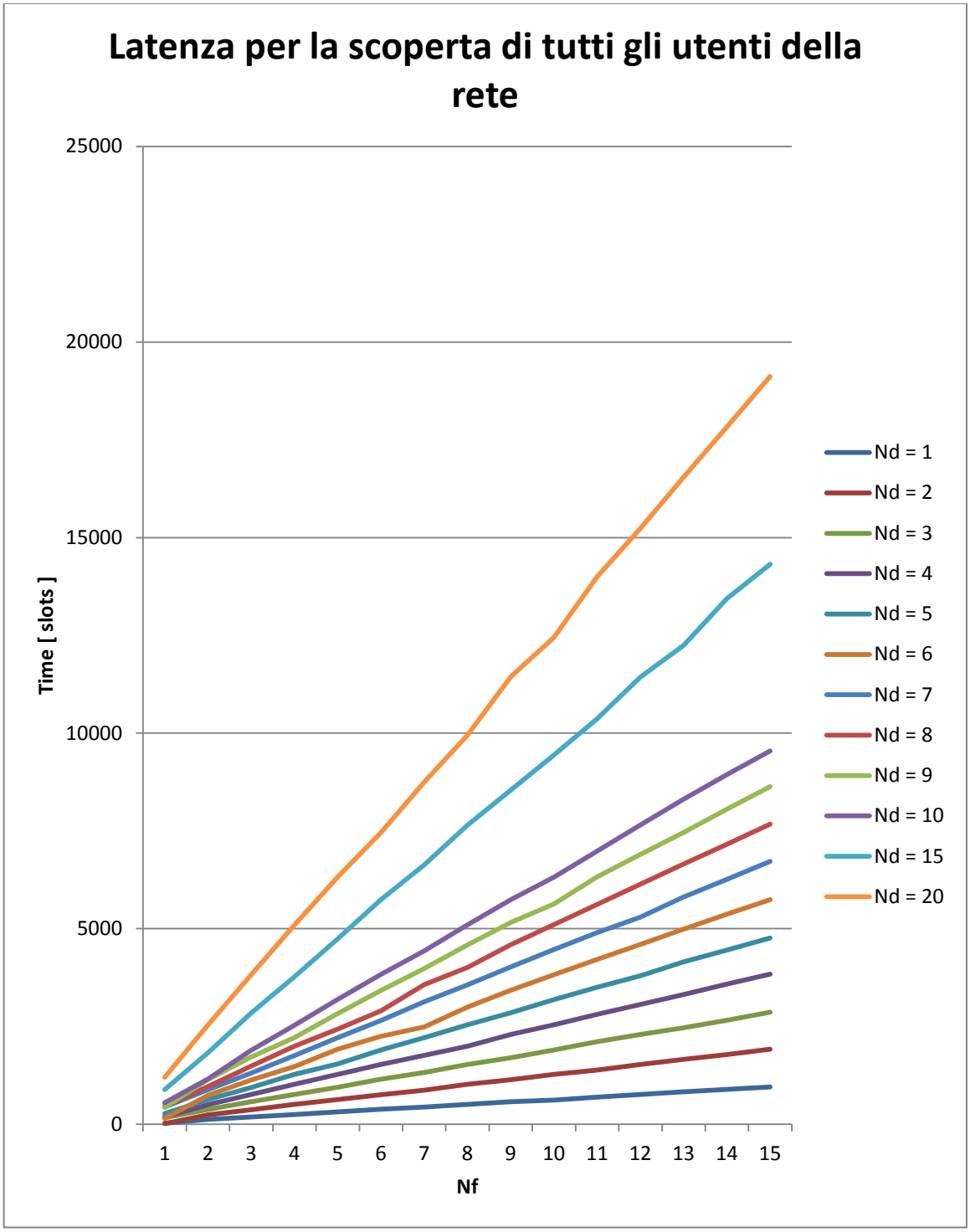

GRAFICO 4.14. Latenza per la scoperta di tutta la rete con t = 16

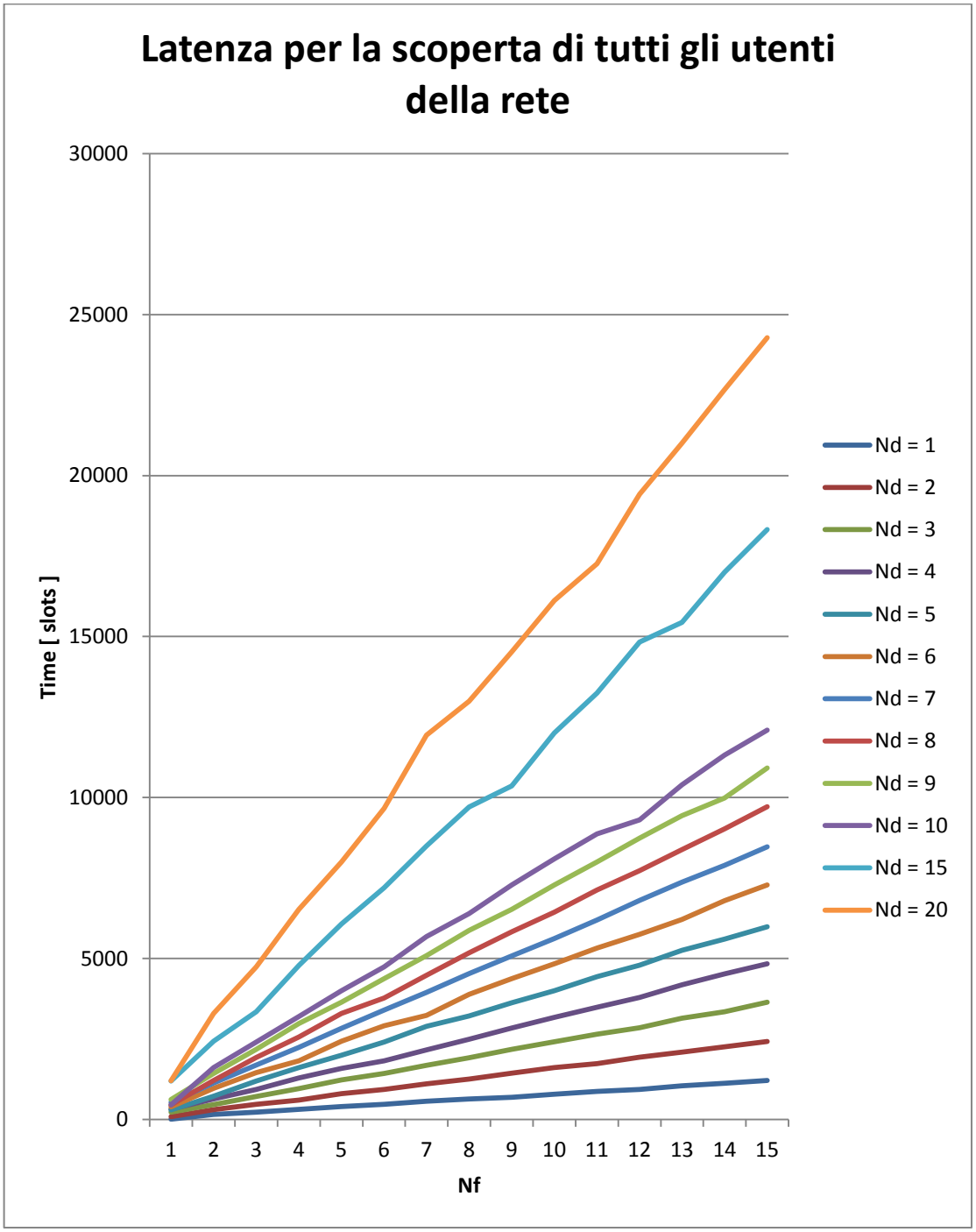

GRAFICO 4.15. Latenza per la scoperta di tutta la rete con t = 18

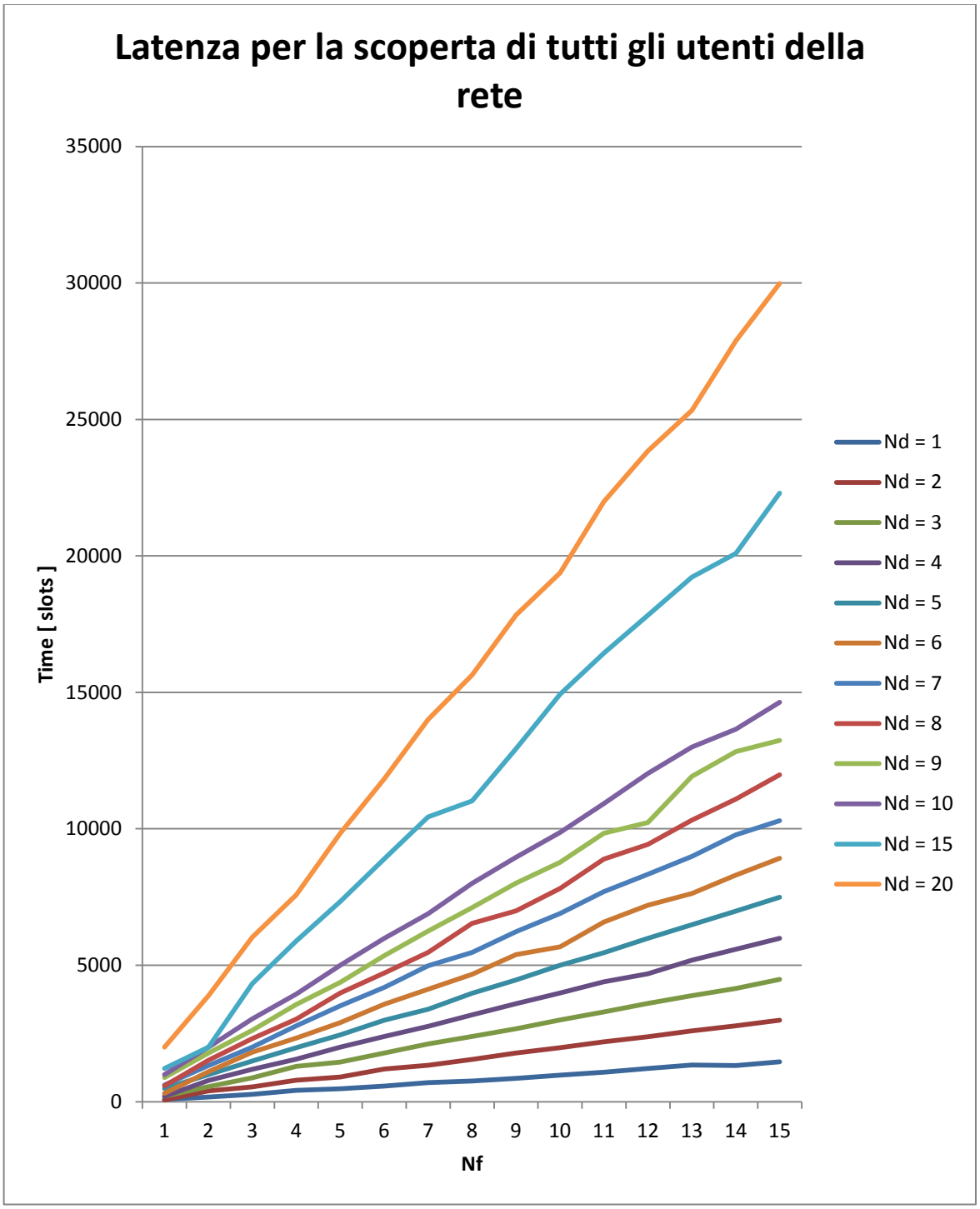

GRAFICO 4.16. Latenza per la scoperta di tutta la rete con t = 20

Osservando i grafici che valutano la latenza per la scoperta di tutti gli utenti della rete, si evince che al crescere del numero di dispositivi presenti nella rete al momento dell'entrata del dispositivo di cui ci si interessa, tale latenza tende ad assumere dei valori estremamente grandi. Per evitare un tempo di discovery troppo elevato, e quindi che rende le prestazioni del protocollo molto basse, risulta essere obbligatorio scegliere di operare con un consumo energetico molto alto.

Di fatti, se consideriamo come esempio la situazione in cui si hanno Nd = 20 dispositivi presenti all'interno della rete, se si sceglie di operare con un duty-cycle basso, ad esempio $\eta = \frac{2}{20} = \frac{1}{10}$, si osserva che il tempo medio che occorre ad effettuare la totale scoperta della rete è pari a 29983 slots temporali. Questo valore è ovviamente problematico in quanto l'utente entrante in rete dovrà attendere troppo tempo prima di poter usufruire del servizio d'interesse. Pertanto, anche se il consumo energetico sarà alto, si opterà per una scelta del periodo temporale piu piccolo, ad esempio con t = 6, si ha che $\eta = \frac{2}{6} = \frac{1}{3}$ e in questo scenario, la latenza è pari a 2644 slots. Quindi si pagherà un costo energetico piu alto, pero' le prestazioni di Searchlight saranno ragionevolmente buone.

Dopo aver visto come viene gestita l'entrata di un'utente all'interno della rete e la sua scoperta dell'intero vicinato con l'ausilio del protocollo Searchlight, ci poniamo adesso il problema di valutare la latenza di questo ipotetico utente per scoprire una porzione della rete.

4.2.2. Scoperta di una porzione della rete al variare di Nf

La scelta delle frequenze utilizzate da ognuno dei Nd dispostivi della rete avviene sempre in maniera uniforme. Cosi operando, accade spesso che una frequenza f_j venga utilizzata da piu dispositivi. Pertanto se il dispositivo entrante in rete non necessita la scoperta dell'intero vicinato, bensi di una predeterminata porzione dei vicini, la latenza per la scoperta di questo numero di dispositivi sarà presumibilmente molto inferiore rispetto a quella necessaria per la totale scoperta della rete.

Come prima situazione, imponiamo t = 6, e ci interessiamo a quanto tempo è necessario per scoprire diverse porzioni della rete. In particolare, si vuole confrontare il tempo per scoprire il 20% , 50% e 100% della rete. Ovviamente per la scoperta del 100% della rete saranno sfruttate le realizzazioni utilizzate per il sotto paragrafo 4.2.1.

Nel seguito, ci interesseremo solamente al confronto fra la latenza media per scoprire la totalità della rete e quella necessaria per effettuare una scoperta di metà del vicinato, ovvero del 50% dei Nd dispositivi presenti al momento dell'entrata del device che sfrutta il protocollo Searchlight.

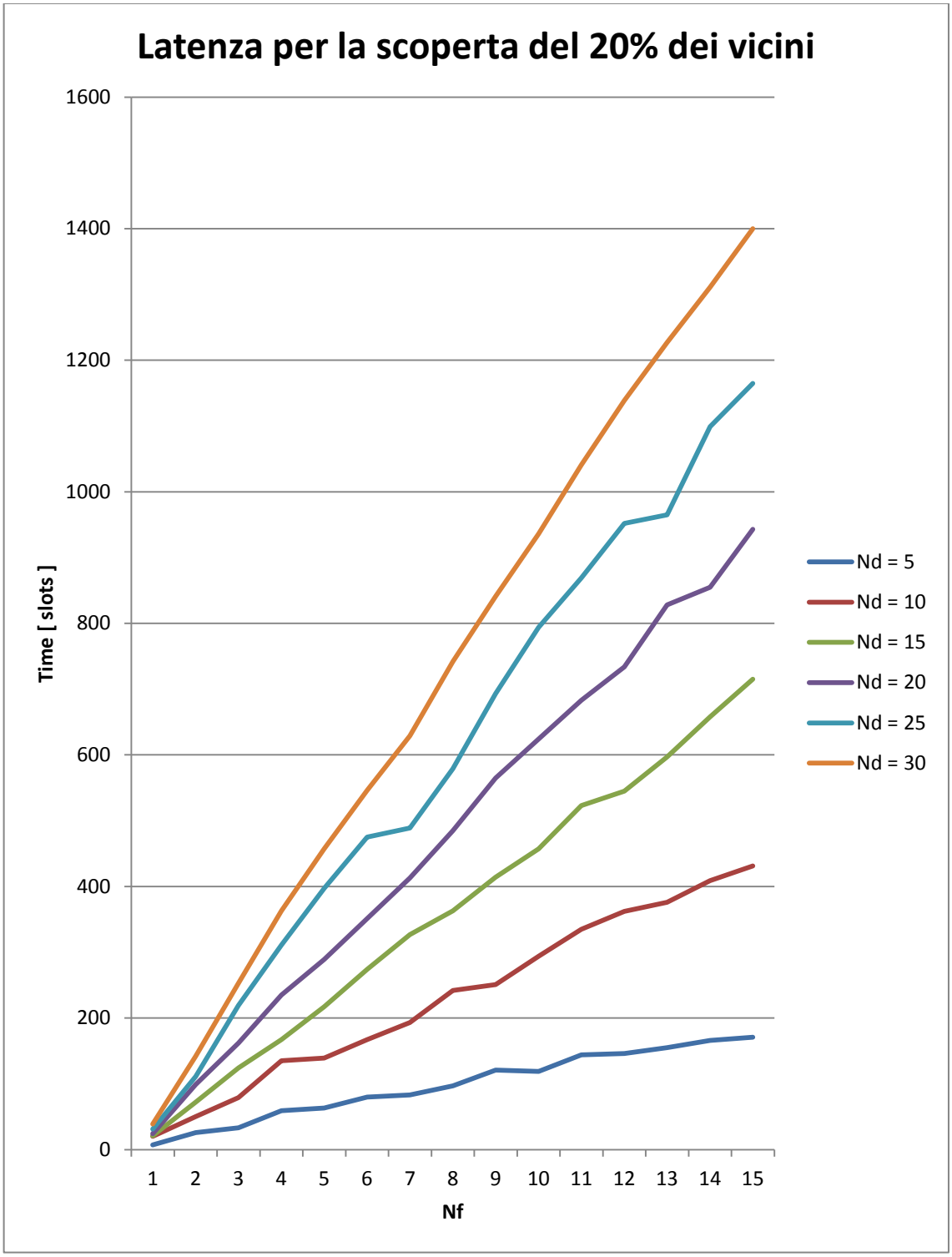

GRAFICO 4.17. Latenza per scoprire 20% della rete con t = 6

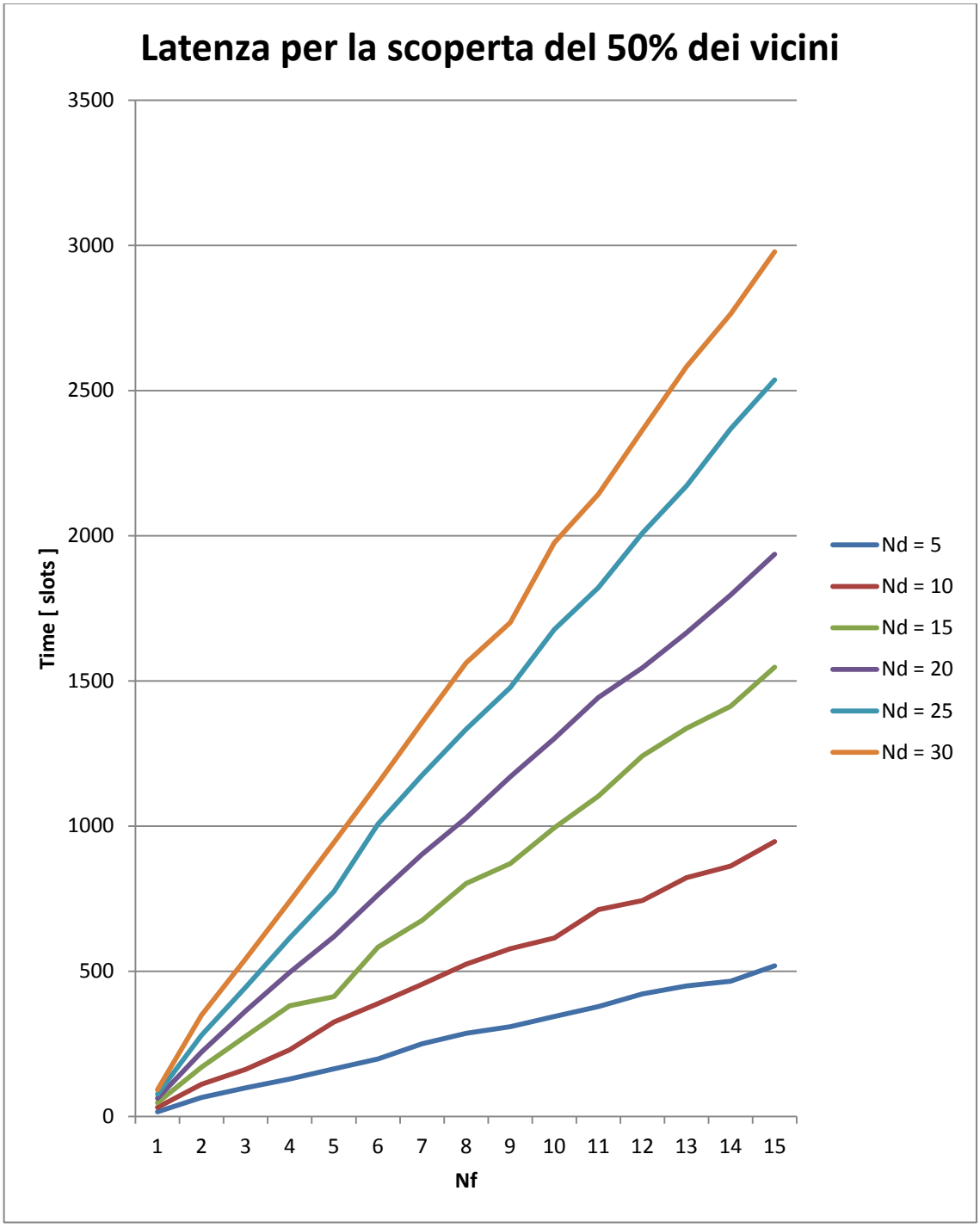

GRAFICO 4.18. Latenza per scoprire 50% della rete con t = 6

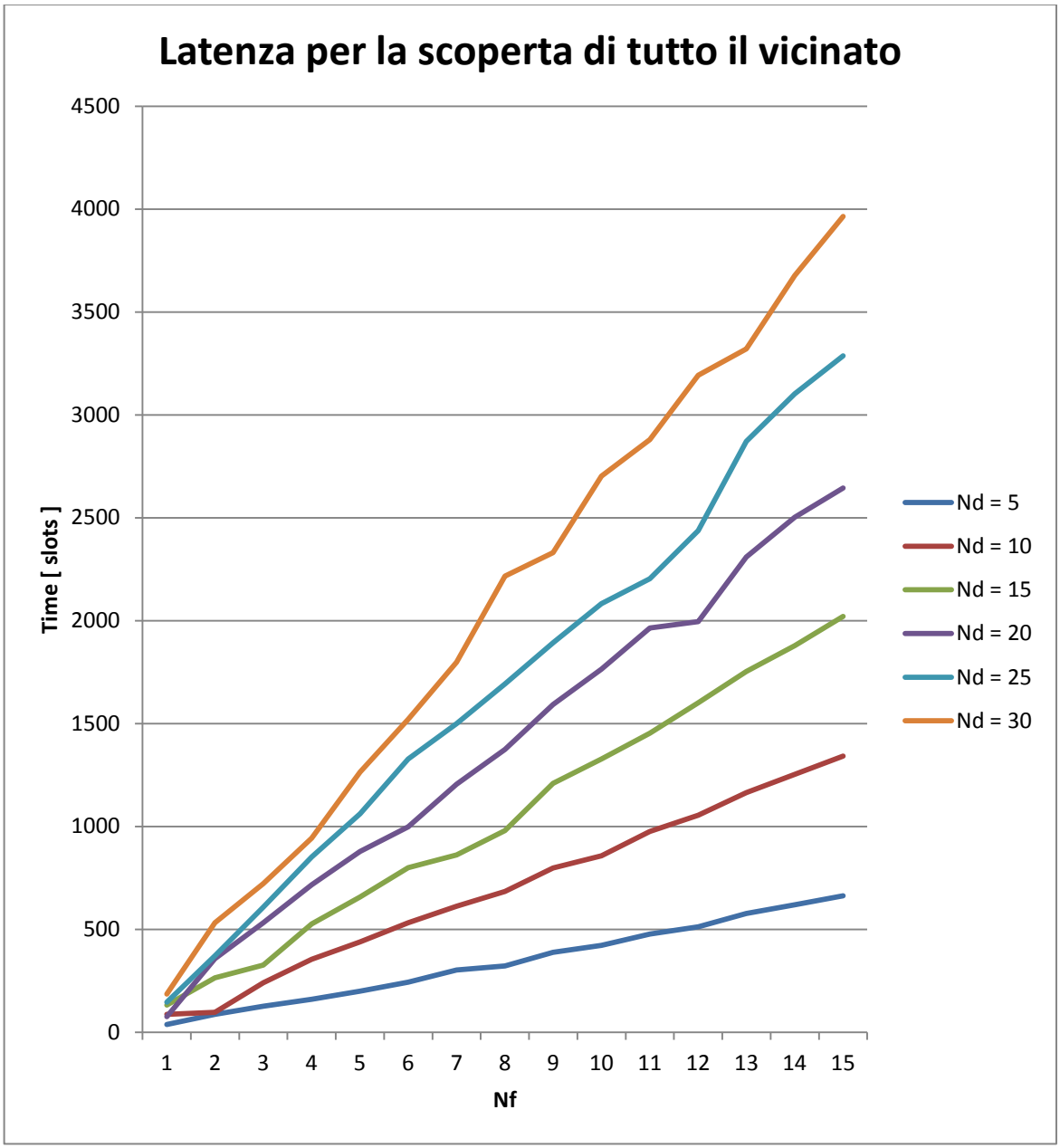

GRAFICO 4.19. Latenza per scoprire 100% della rete con t = 6

Osservando i Grafici 4.17, 4.18 e 4.19, si vede come se il dispositivo entrante nella rete non richiede la scoperta di tutto il suo vicinato, si puo' optare per una migliore gestione del consumo energetico, utilizzando quindi un valore del periodo temporale t maggiore.

Valutiamo quest'ultima osservazione effettuando delle simulazioni che considerano un duty-cycle piccolo rispetto all'esempio visto in precedenza. Per vedere la correttezza di

quanto detto, si andrà a valutare la latenza media della discovery fissando il numero Nd di dispositivi presenti nel sistema per alcuni valori di t, confrontando il valore ottenuto per una scoperta totale della rete e quello ricavato quando il dispositivo entrante necessita di conoscere il 50% degli utenti presenti.

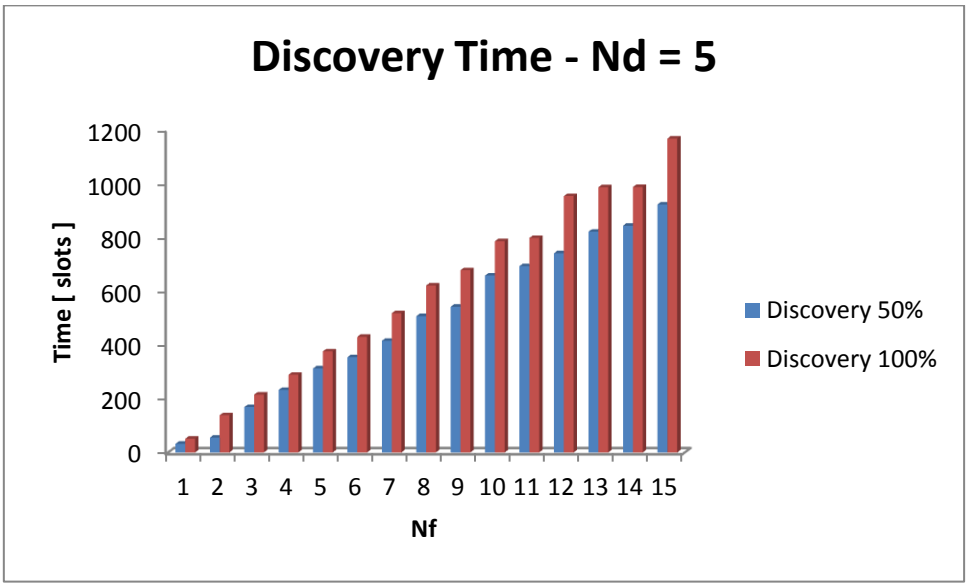

GRAFICO 4.20. Confronto fra scoperta totale e parziale con Nd = 5 e t = 8

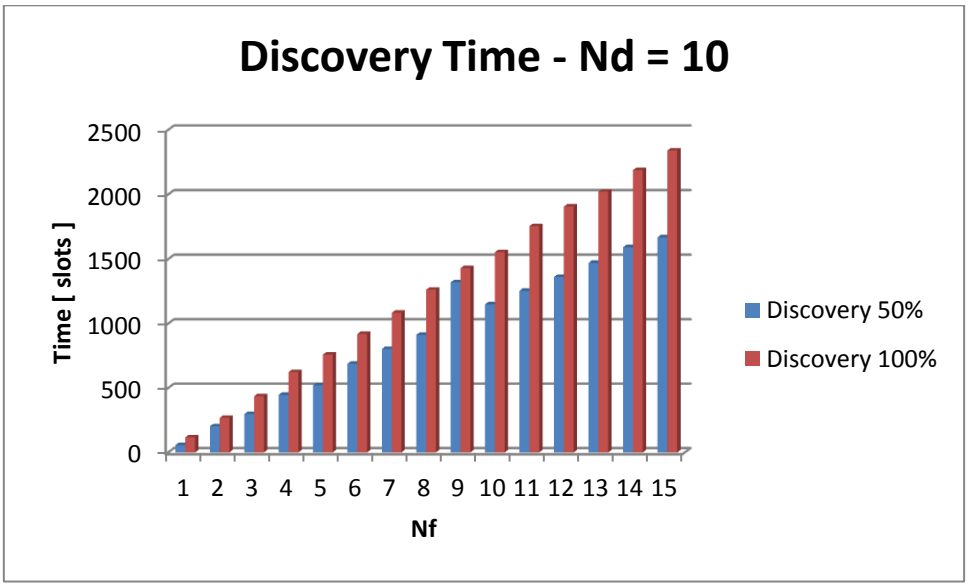

GRAFICO 4.21. Confronto fra scoperta totale e parziale con Nd = 10 e t = 8

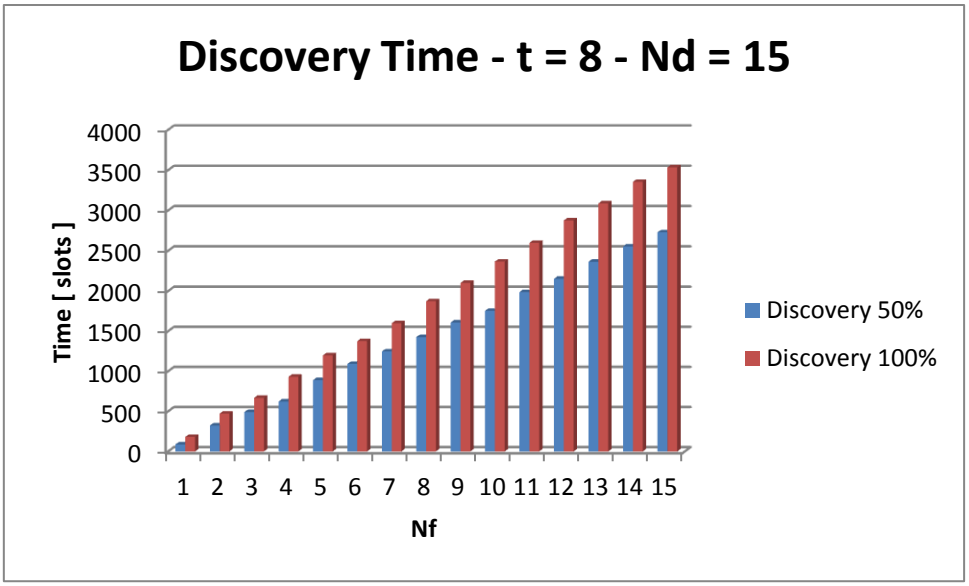

GRAFICO 4.22. Confronto fra scoperta totale e parziale con Nd = 15 e t = 8

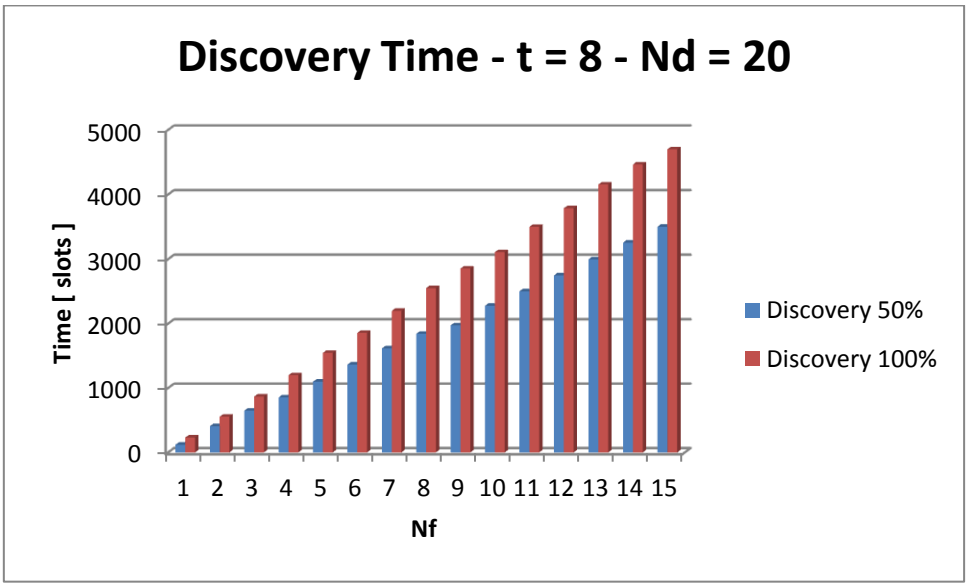

GRAFICO 4.23. Confronto fra scoperta totale e parziale con Nd = 20 e t = 8

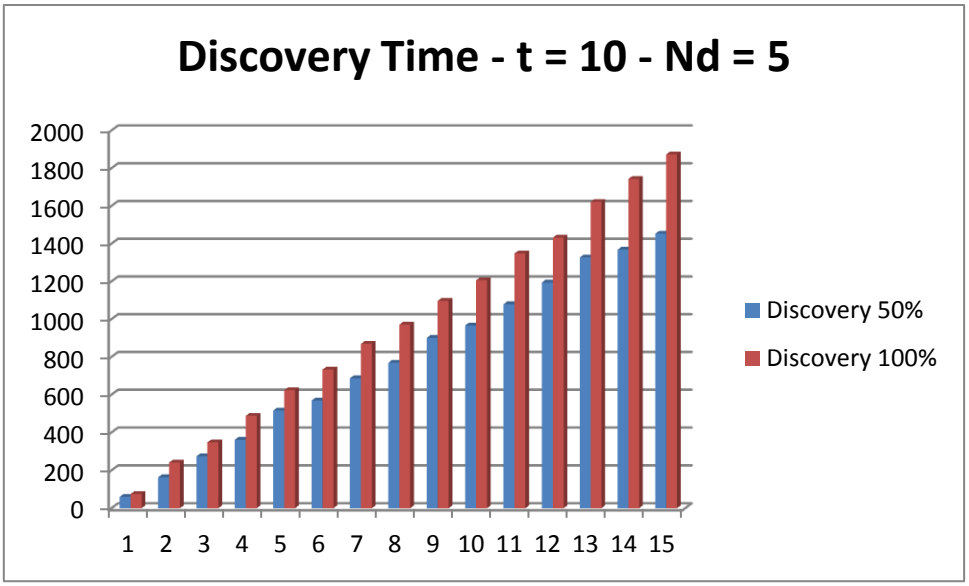

GRAFICO 4.24 Confronto fra scoperta totale e parziale con Nd = 5 e t = 10

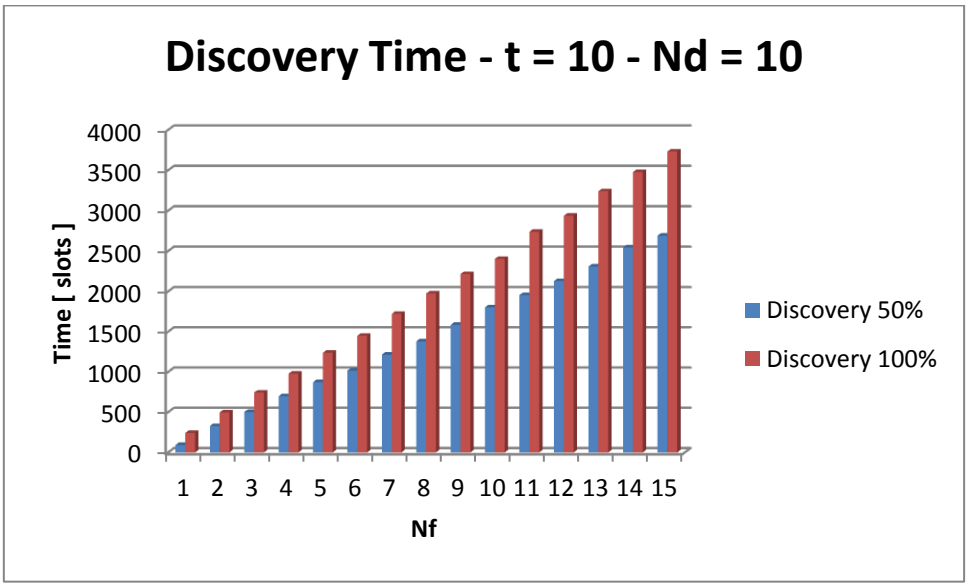

GRAFICO 4.25. Confronto fra scoperta totale e parziale con Nd = 10 e t = 10

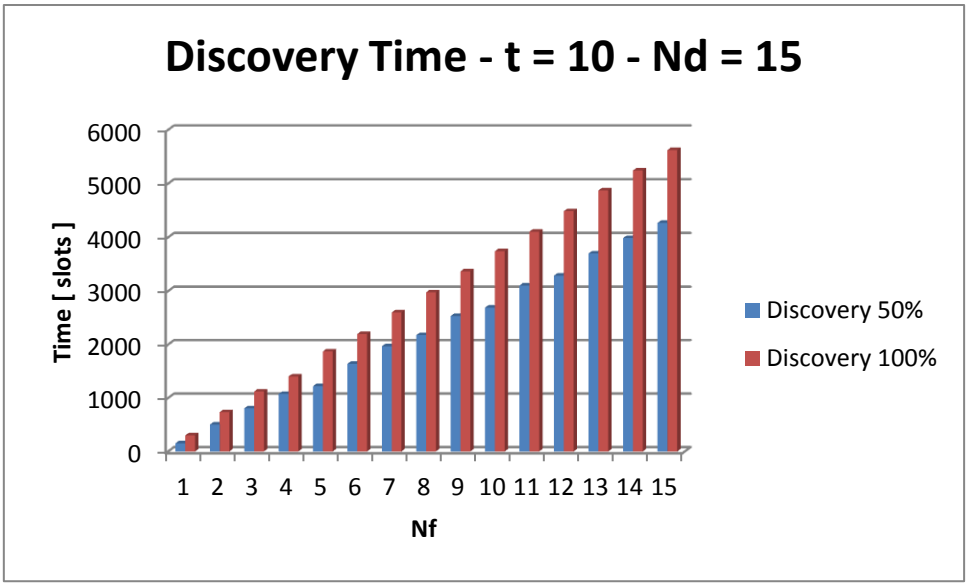

GRAFICO 4.26. Confronto fra scoperta totale e parziale con Nd = 15 e t = 10

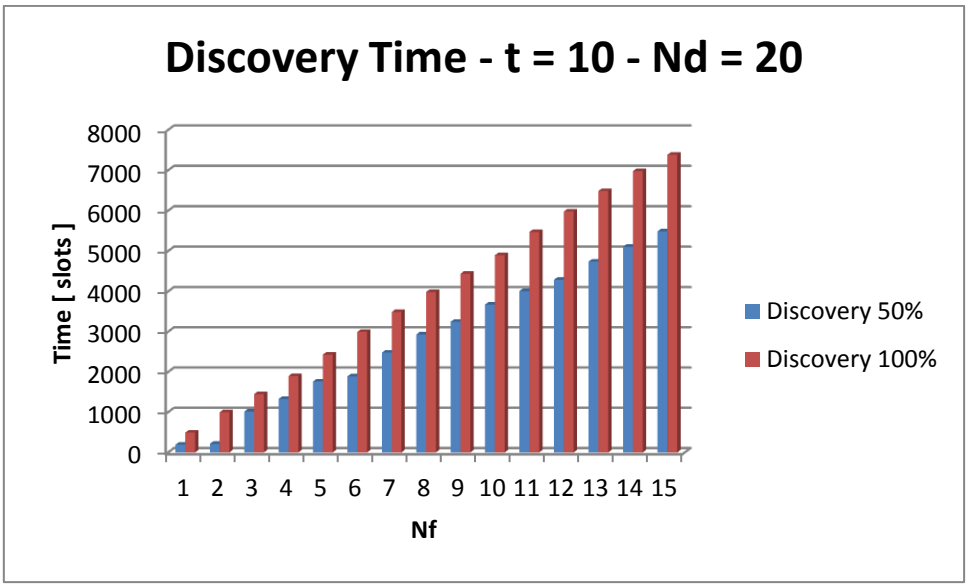

GRAFICO 4.27. Confronto fra scoperta totale e parziale con Nd = 20 e t = 10

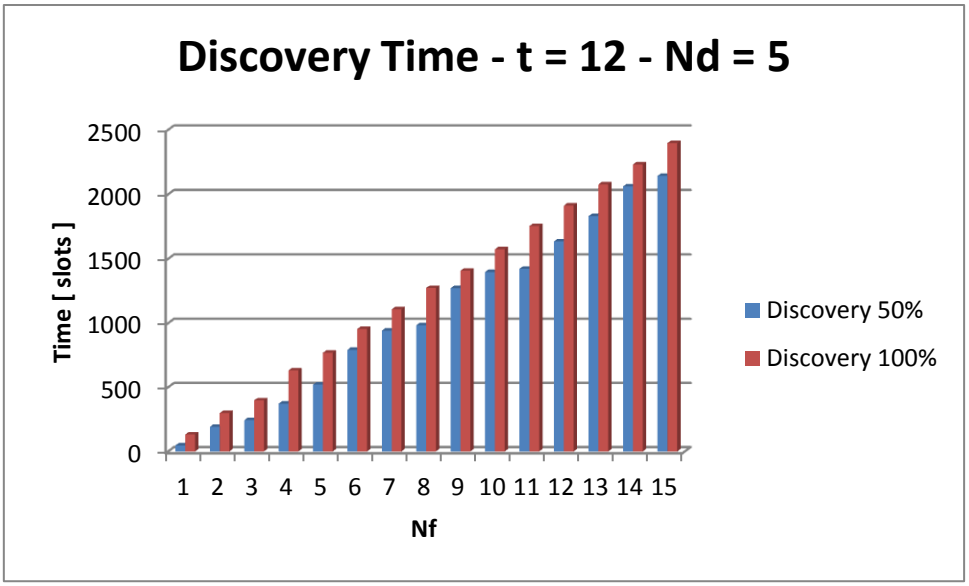

GRAFICO 4.28. Confronto fra scoperta totale e parziale con Nd = 5 e t = 12

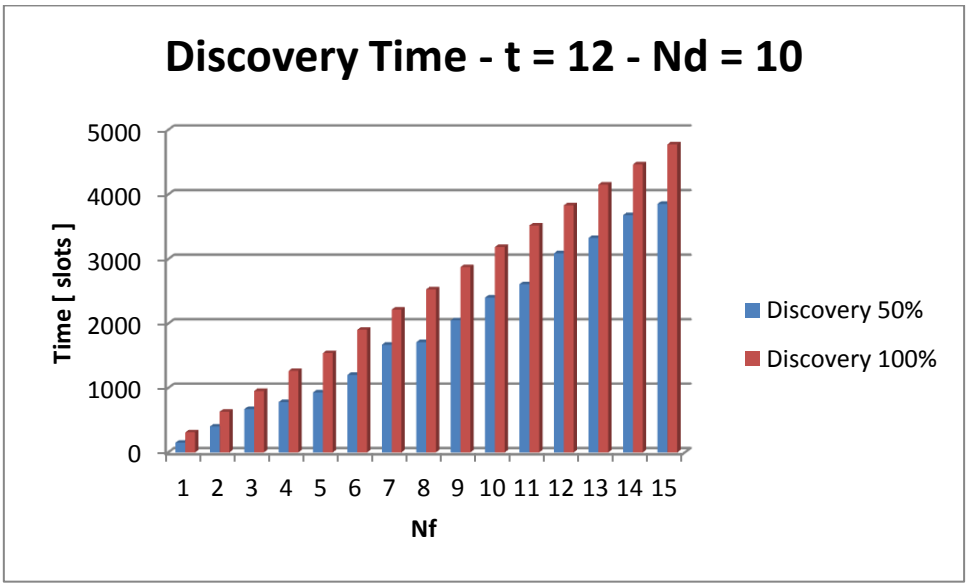

GRAFICO 4.29. Confronto fra scoperta totale e parziale con Nd = 10 e t = 12

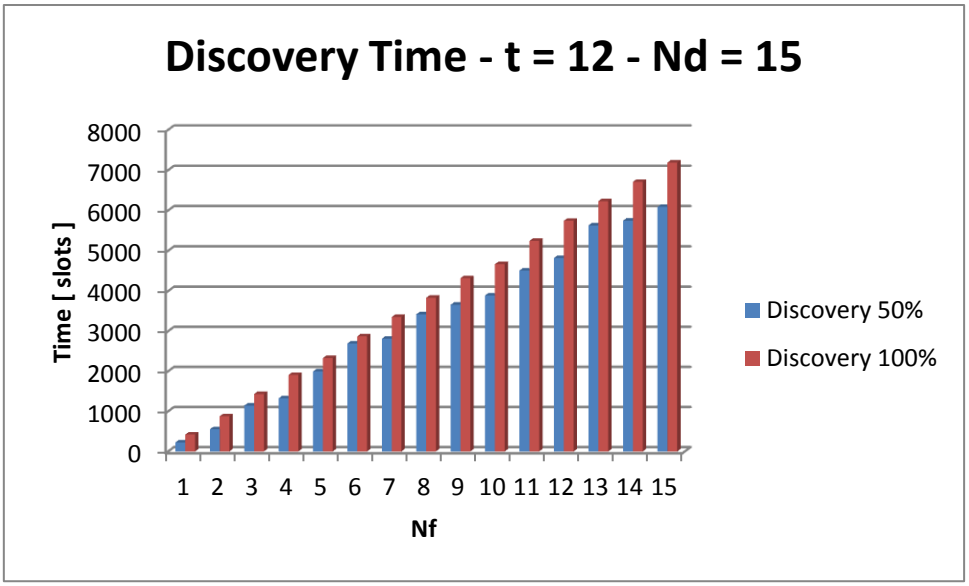

GRAFICO 4.30. Confronto fra scoperta totale e parziale con Nd = 15 e t = 12

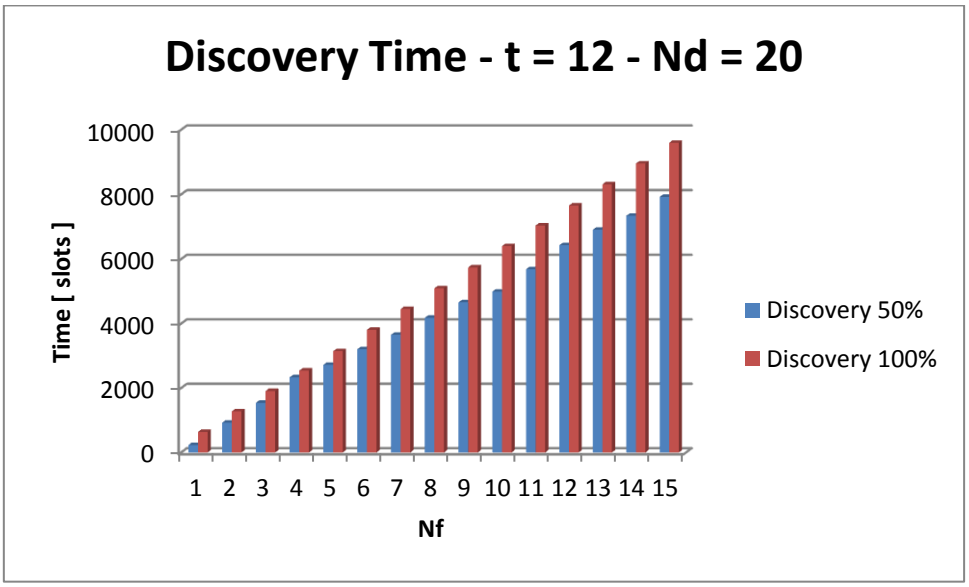

GRAFICO 4.31. Confronto fra scoperta totale e parziale con Nd = 20 e t = 12

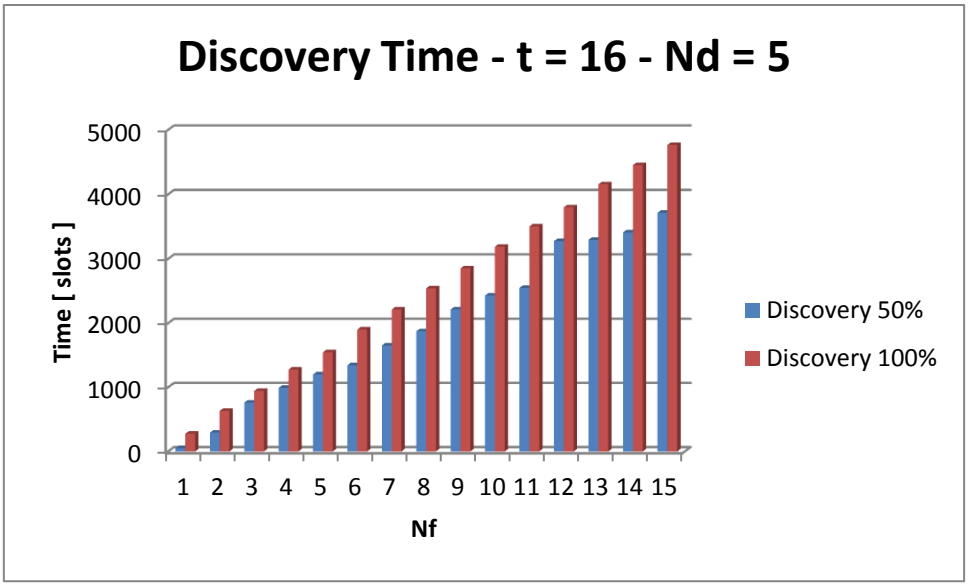

GRAFICO 4.32. Confronto fra scoperta totale e parziale con Nd = 5 e t = 16

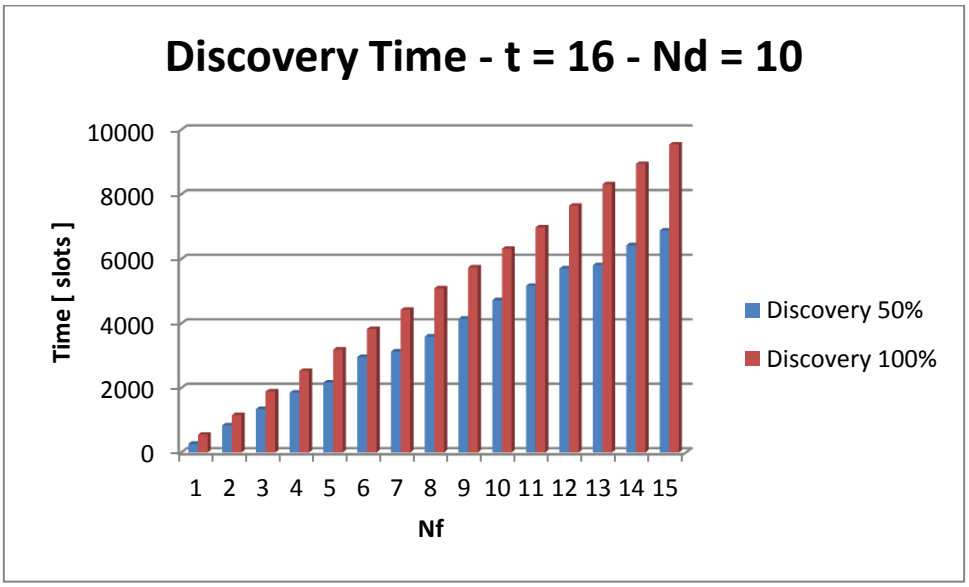

GRAFICO 4.33. Confronto fra scoperta totale e parziale con Nd = 10 e t = 16

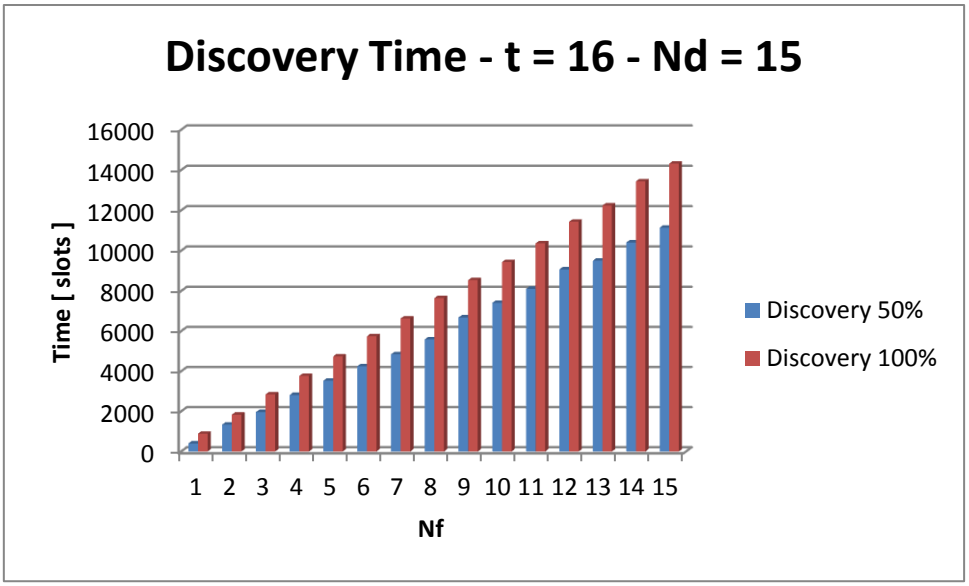

GRAFICO 4.34. Confronto fra scoperta totale e parziale con Nd = 15 e t = 16

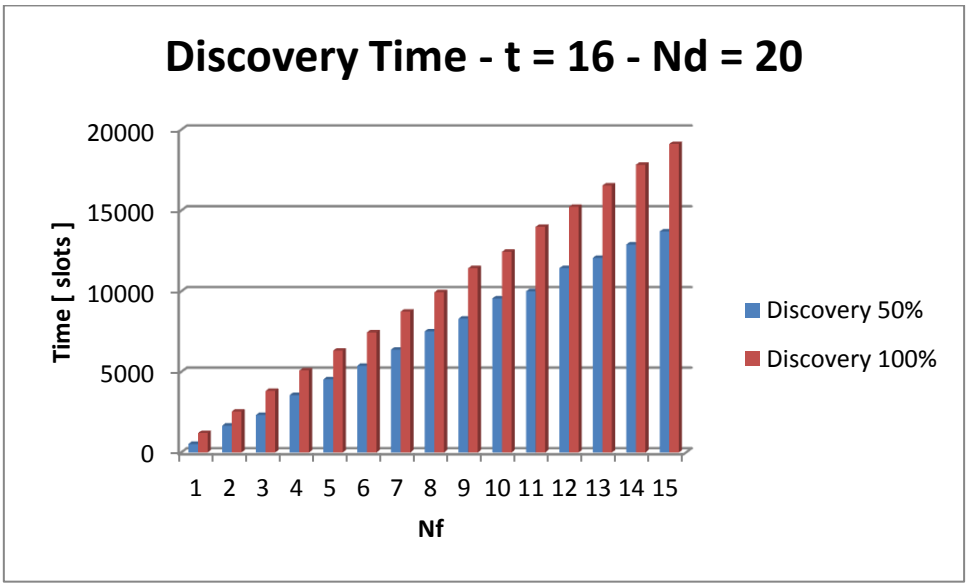

GRAFICO 4.35. Confronto fra scoperta totale e parziale con Nd = 20 e t = 16

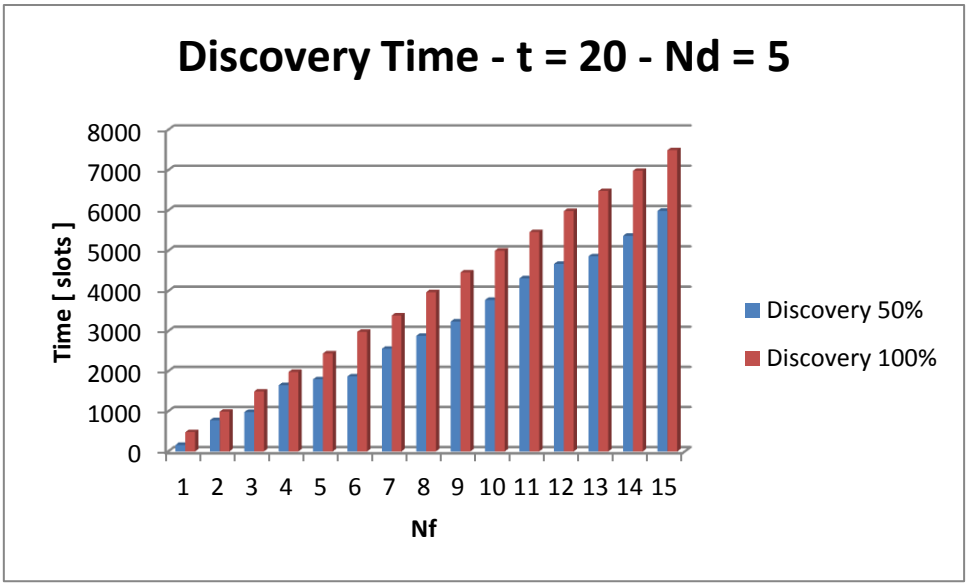

GRAFICO 4.36. Confronto fra scoperta totale e parziale con Nd = 5 e t = 20

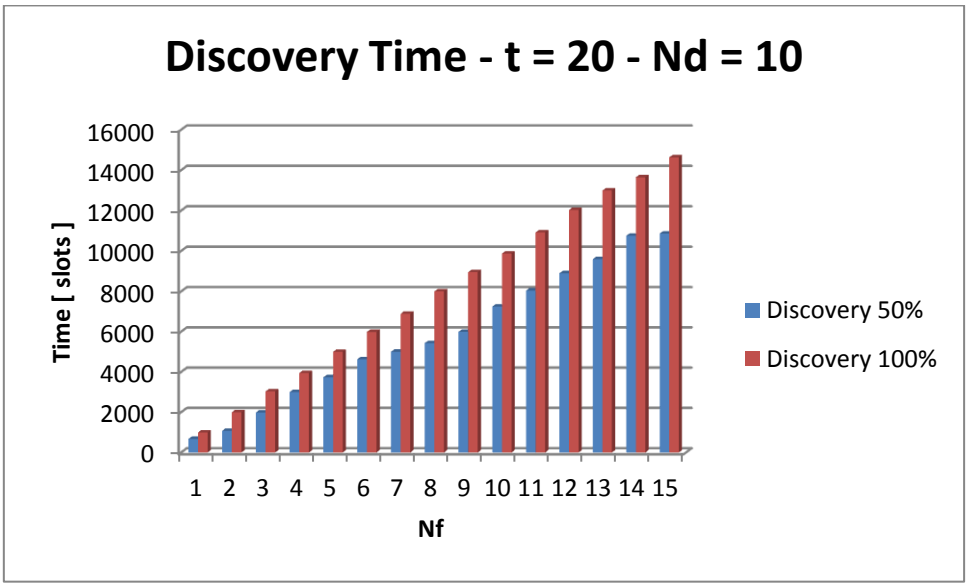

GRAFICO 4.37. Confronto fra scoperta totale e parziale con Nd = 10 e t = 20

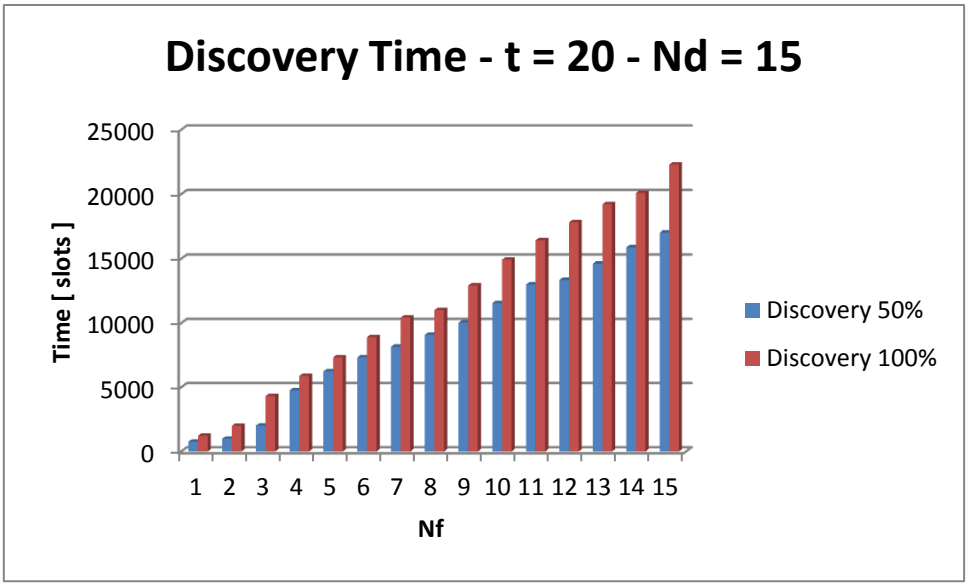

GRAFICO 4.38. Confronto fra scoperta totale e parziale con Nd = 15 e t = 20

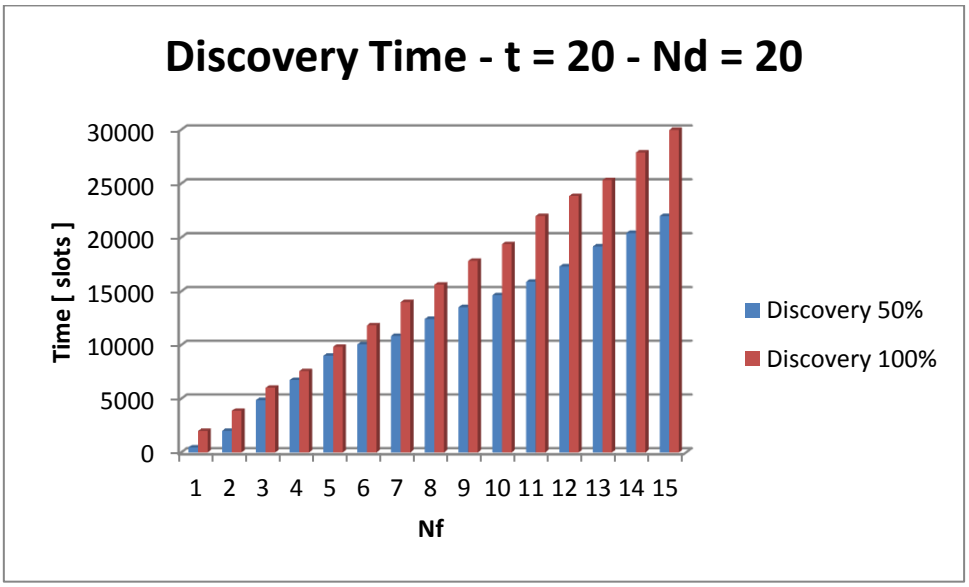

GRAFICO 4.39. Confronto fra scopera totale e parziale con Nd = 20 e t = 20

Valutando tutti i grafici di confronto per la latenza media della scoperta nel caso di scoperta totale della rete e nel caso di scoperta di solamente metà della rete, si evince che questa diminuzione di opportunità richiesta dall'utente entrante porta una diminuzione molto significativa del tempo di discovery.

In particolare, ci si rende conto come questa parziale scoperta è molto utile in termini di prestazioni del protocollo Searchlight per valori grandi del numero di frequenze da sondare. Inoltre, se l'utente entrante non necessita di scoprire tutto il suo vicinato, si puo' osservare dagli utimi grafici riportati che è possibile optare per una scelta migliore in termini di consumo energetico, in altre parole, la scelta del valore del periodo temporale puo' anche essere grande se l'utente entrante non richiede di scoprire la totalità degli utenti presenti in rete prima della sua entrata.

4.2.3. Tempo di scoperta al variare del numero di utenti in rete

In questa sezione, si vuole valutare il tempo di scoperta di tutta la rete da parte del dispositivo entrante, ovvero di tutti gli Nd utenti presenti, fissando il numero di frequenze disponibili Nf. Ci troviamo sempre nella situazione in cui le frequenze utilizzate dagli utenti nella rete sono puramente casuali.

Nello specifico, il tempo d'attesa riportato sarà quello relativo ad ogni scelta del duty-cycle che viene effettuata, al variare di quanti utenti sono presenti all'interno della rete. I duty-cycles scelti sono t = 6, t = 8 , t = 18 e t = 20. Questa scelta è stata fatta in modo tale da poter valutare le prestazioni di Searchlight in presenza di Nd utenti da scoprire nella rete, e vedere come il costo eneregetico influisce su esse.

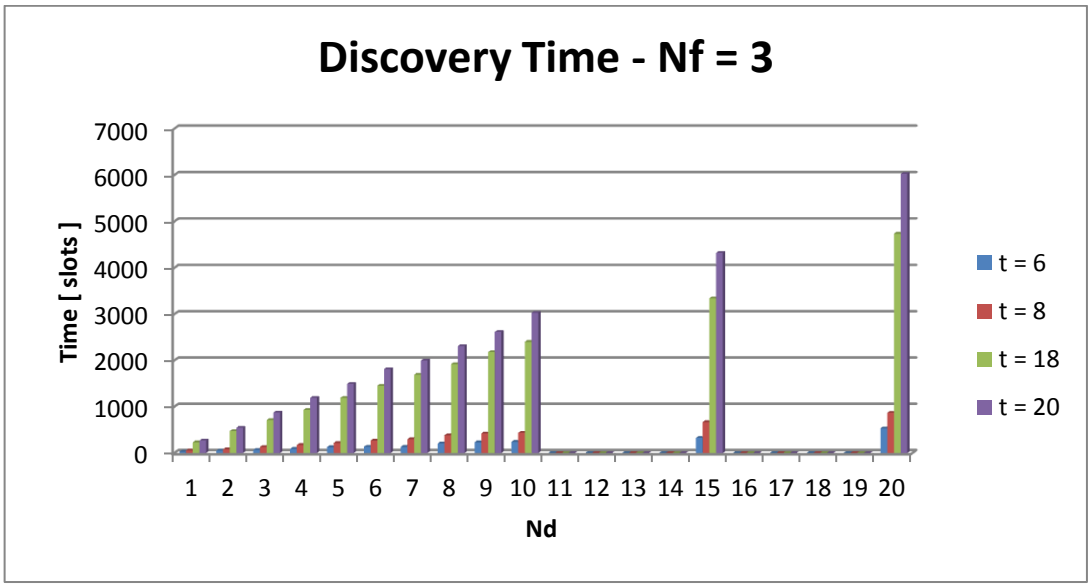

GRAFICO 4.40. Tempo di scoperta totale con Nf = 3

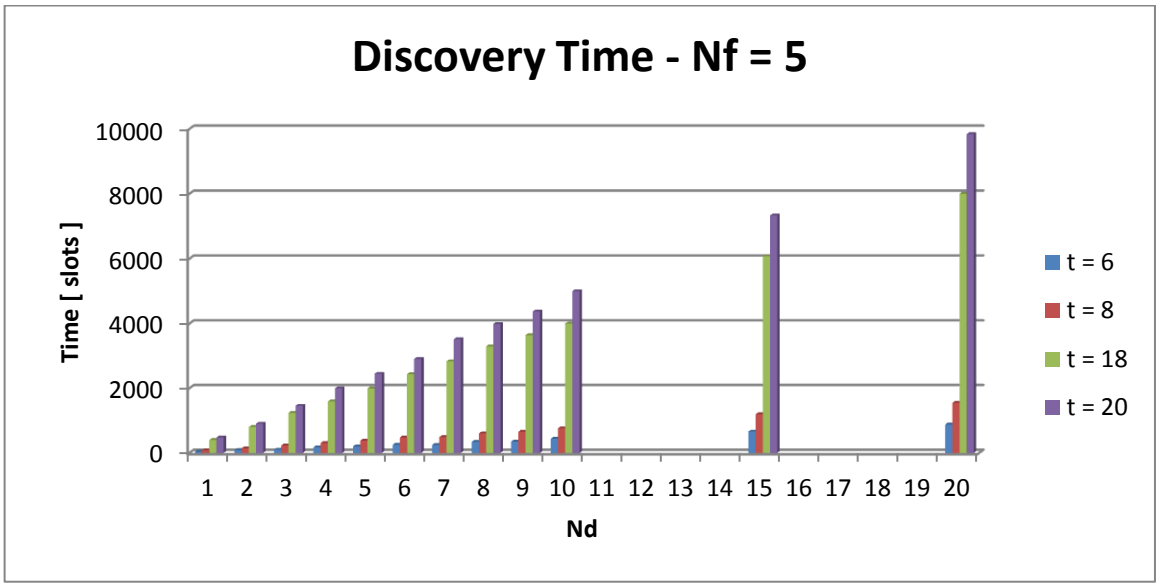

GRAFICO 4.41. Tempo di scoperta totale con Nf = 5

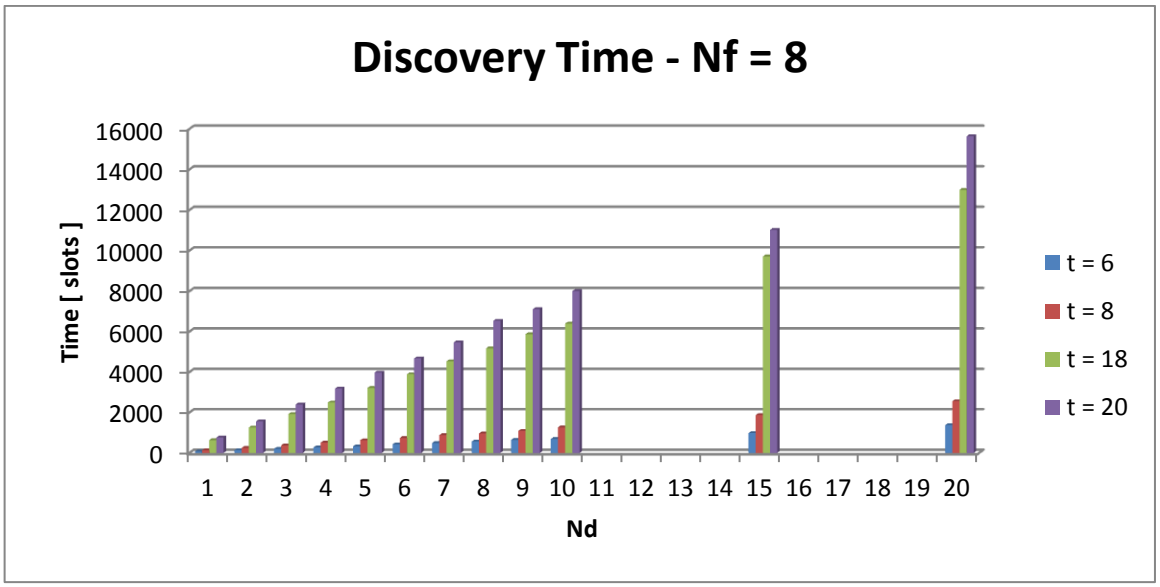

GRAFICO 4.42. Tempo di scoperta totale con Nf = 8

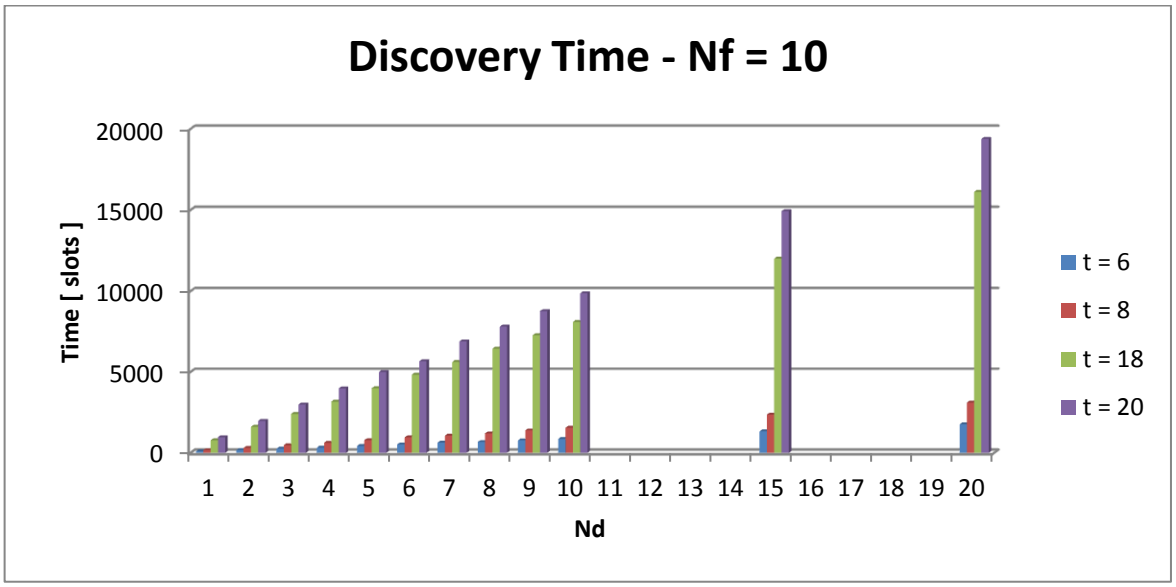

GRAFICO 4.43. Tempo di scoperta totale con Nf = 10

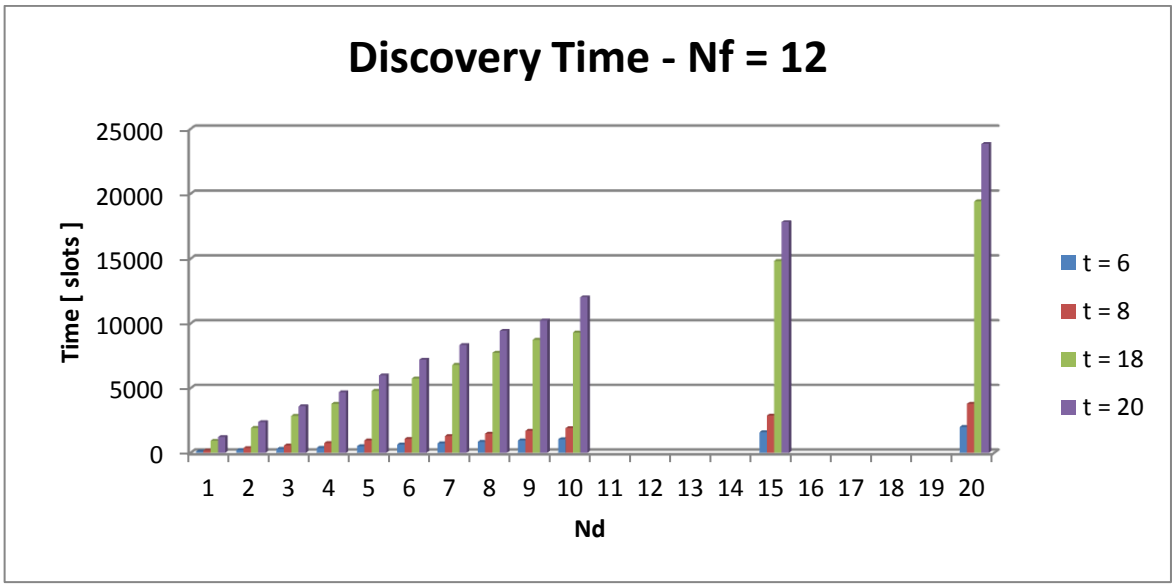

GRAFICO 4.44. Tempo di scoperta totale con Nf = 12

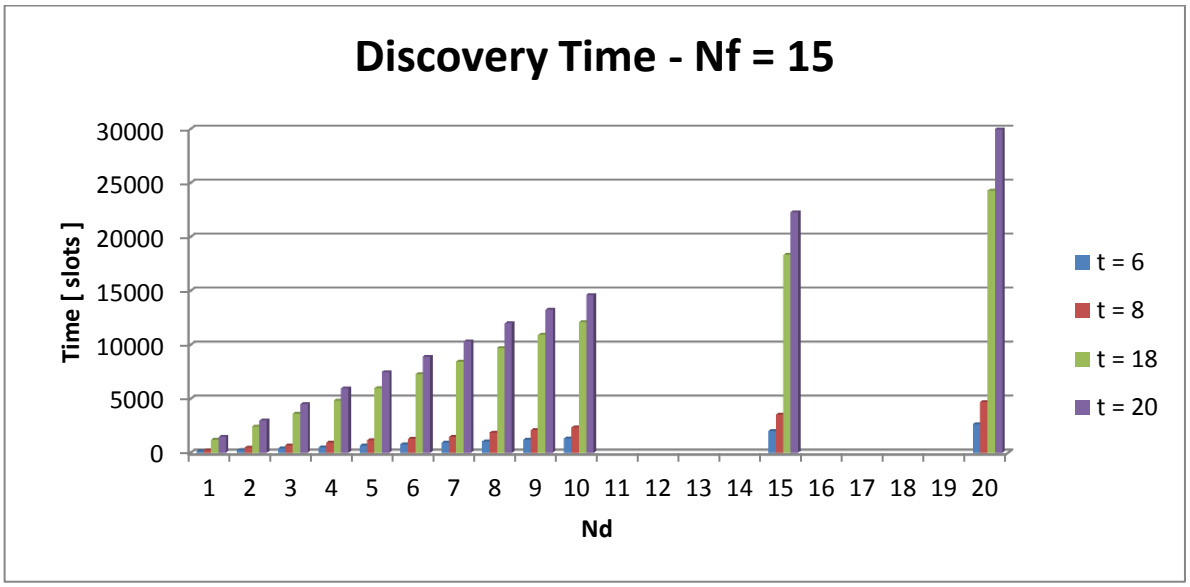

GRAFICO 4.45. Tempo di scoperta totale con Nf = 15

Osservando i grafici riportati, si nota come al crescere del numero di dispositivi da scoprire all'interno della rete, la latenza per la scoperta cresce sensibilmente, anche per valori piccoli del numero di frequenze disponibili.

Si puo' quindi affermare che quando il dispositivo entrante prevede che il numero di dispositivi da scoprire è grande, ad esempio maggiore di Nd = 10, risulta inconveniente al fine della scoperta optare per un risparmio energetico.

Se si osserva il grafico 4.45 , ovvero quando il numero di frequenze utilizzabili all'interno della rete di cui l'utente entrante vuole fare parte è pari a Nf = 15, se il numero di devices presenti in essa è pari a Nd = 20, se si effettua una scelta energeticamente dispendiosa, utilizzando ad esempio t = 8, la latenza per la scoperta totale è pari a 4698 slots temporali in media, mentre qualora si decidesse di fare una scelta conservativa di energia, con un duty-cycle pari a $\eta = \frac{2}{18} = \frac{1}{9}$, Searchlight consente la scoperta, pero' il tempo di scoperta che esso impieghera in media sarà pari a 24283 slots temporali.

Pertanto, ancora una volta si vede come il trade-off fra qualità della scoperta in termini di latenza e scelta di un'opportuno consumo energetico sia problematico. Si noti inoltre che il dispositivo in rete non conosce a priori il numero di devices in rete Nd, pertanto risulterà complesso per lui valutare la gestione di questo trade-off prima della sua entrata nella rete ad-hoc d'interesse.

4.3. Frequenze distribuite secondo una funzione di distribuzione

Fino a questo punto del capitolo, abbiamo supposto che le frequenze di ognuno dei Nd dispositivi presenti all'interno della rete siano state distribuite in maniera uniforme, pertanto in questo scenario la probabilità che la frequenza assegnata ad un dispositivo in rete D_i sia la frequenza F_j appartenente all'insieme di tutte le frequenze utilizzabili nella rete Nf, è pari a $P(F_{ij}) = \frac{1}{Nf}$. Questa probabilità è valida per ogni dispositivo.

All'interno di questo paragrafo 4.3, si infrange questa ipotesi e si considera che ogni dispositivo presente all'interno della rete utilizzi una frequenza non piu casuale, bensi che segue una predeterminata funzione di distribuzione. In altri termini, ogni frequenza F_j puo' essere assegnata al device D_i con una certa probabilità nota a priori, che pero' non è piu la medesima per ogni frequenza F appartenente all'insieme Nf.

In questa situazione, l'utente entrante nella rete effettua la scoperta degli Nd dispositivi appartenenti alla rete. Al contrario della casistica con le frequenza uniformemente distribuite, qui l'utente entrante puo' pensare di sfruttare alcune informazioni che ottiene durante la discovery.

Definiamo tre possibili politiche di sondaggio in frequenza da parte dell'utente entrante.

a) *Politica di sondaggio uniforme – Politica 1*

Questa prima tipologia di sondaggio prevede che l'utente entrante non sfrutti in alcun modo l'informazione che riceve dalle scoperte pregresse. Quindi ad ogni sondaggio in frequenza esso sceglierà, come nella situazione uniformemente distribuita, una frequenza causale all'interno dell'insieme Nf.

b) *Politica di sondaggio "neighbor frequency" – Politica 2*

In questo secondo scenario, l'utente entrante sfrutta l'informazione che ottiene dalla scoperta effettuata dai primi due sondaggi in frequenza.

Al primo sondaggio, il dispositivo che entra nella rete sonda una frequenza casuale f_i. Se non vengono scoperti tutti gli Nd dispositivi in rete occorre continuare il sondaggio. La seconda frequenza sondata è la frequenza $f_i + 1$ ove f_i è la frequenza sondata al primo sondaggio, come definito in precendenza. A questo punto, qualora la discovery totale non fosse raggiunta, vengono confrontati il numero di dispositivi scoperti con il sondaggio alla frequenza f_i e quelle relativi alla frequenza $f_i + 1$.

- Se il numero di devices scoperti al primo sondaggio è maggiore di quello del secondo, allora la terza frequenza sondata sarà la frequenza $f_i – 1$, e cosi via decrescendo, dopo aver sondato la frequenza 1, verrà sondata la frequenza $f_i + 2$, e cosi via in maniera crescente, fino al sondaggio della frequenza Nf qualora fosse necessario per scoprire tutti gli Nd dispositivi.

- Se invece il numero di dispositivi scoperti al secondo sondaggio è maggiore rispetto a quello ottenuto dopo il primo sondaggio, allora si sonderà la frequenza $f_i + 2$, in modo

crescente ad ogni sondaggio, fino ad arrivare alla frequenza Nf, quando essa è raggiunta il sondaggio successivo sarà sulla frequenza $f_i - 1$, si andrà avanti sondando frequenze decrescenti fino al raggiuntimento della frequenza 1, se è necessario.

c) *Politica di sondaggio ai bordi – Politica 3*

Per questa terza politica di sondaggio, la prima frequenza sondata è la frequenza 1. Qualora la scoperta totale della rete non avviene con successo dopo questo primo sondaggio, la seconda frequenza sondata sarà la frequenza Nf. Dopo questi due sondaggi, se non è stata scoperta la totalità del vicinato, si confronta il numero degli utenti scoperti nei due casi.

- Se il primo sondaggio ha portato alla scoperta di un numero maggiore di utenti, allora i sondaggi successivi andranno al crescere della frequenza, ovvero al secondo sondaggio la frequenza 2, al terzo la frequenza 3 e cosi via, fino a quando non verranno scoperti tutti i nodi presenti.

- Se invece il secondo sondaggio, alla frequenza Nf, ha portato ad una scoperta piu consistente rispetto al primo sondaggio, allora si sonderà al terzo sondaggio la frequenza Nf – 1, e cosi via decrescendo fino a quando tutta la rete non è scoperta.

Dopo aver definito queste politiche di sondaggio, ci poniamo il problema di stimare la latenza della scoperta di tutti i vicini del nodo entrante nella rete per ognuna di questa tecniche. Le realizzazioni che si effettueranno andranno a valutare la latenza media, fissato un numero di utenti Nd nella rete ed il numero di frequenze Nf utilizzabili dagli utenti che fanno parte di essa, per ognuna delle politiche di sondaggio definite in precendenza. La latenza media viene calcolata come la media delle latenze medie che vengono ottenute al variare della funzione di distribuzione per l'assegnamento delle frequenze. Si noti infine che si suppone sempre di trovarsi nella situazione sincrona e simmetrica, ed inoltre che il valore del periodo temporale che viene scelto per tutte le simulazioni riportate nel seguito, sia pari a t = 8, offrendo un duty-cycle pari a $\eta = \frac{1}{4}$.

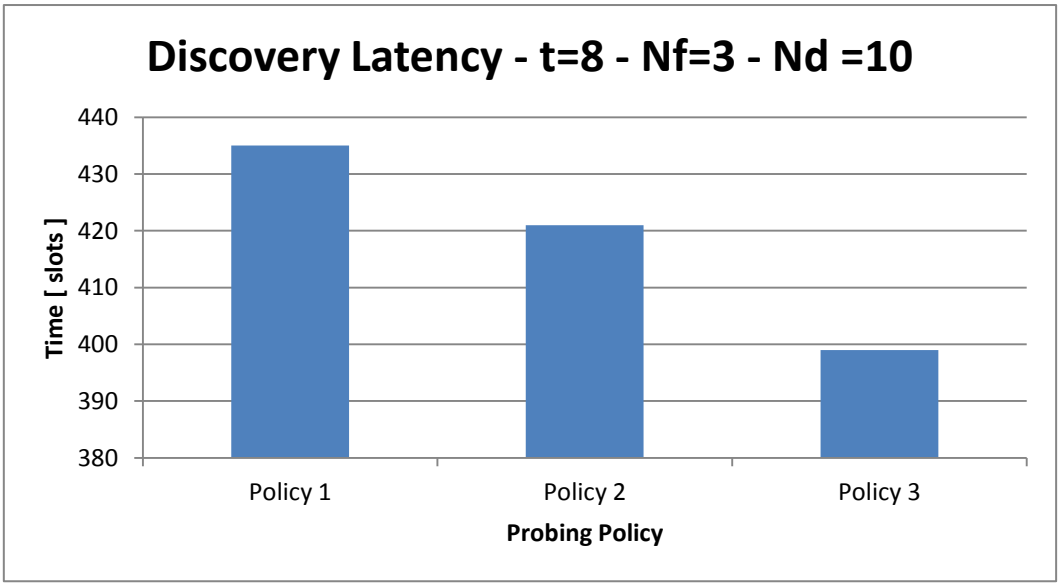

GRAFICO 4.46. Latenza media per 10 distribuzioni con Nf = 3 e Nd = 10

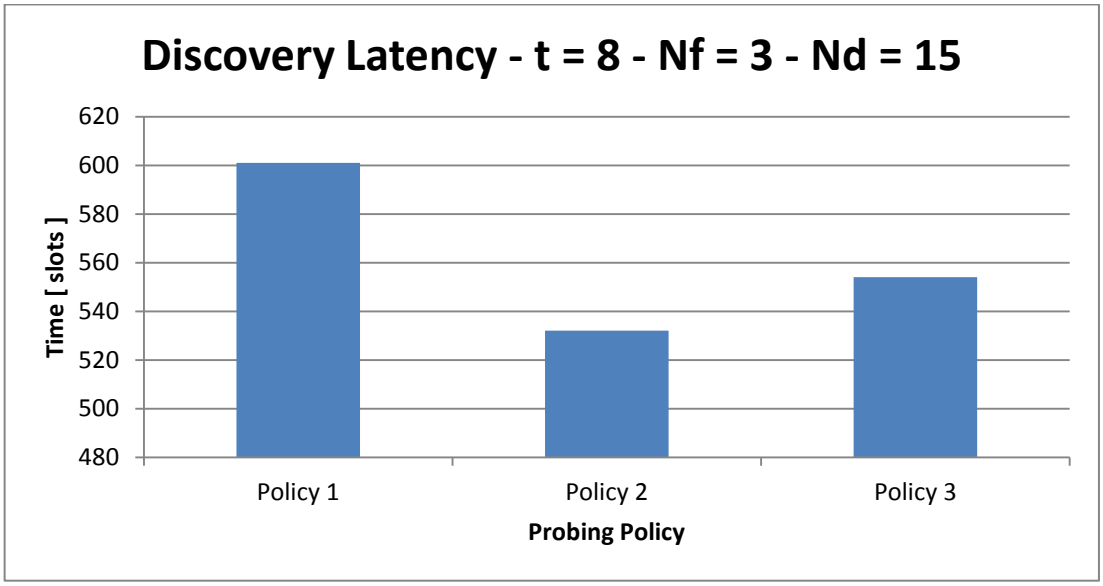

GRAFICO 4.47. Latenza media per 10 distribuzioni con Nf = 3 e Nd = 15

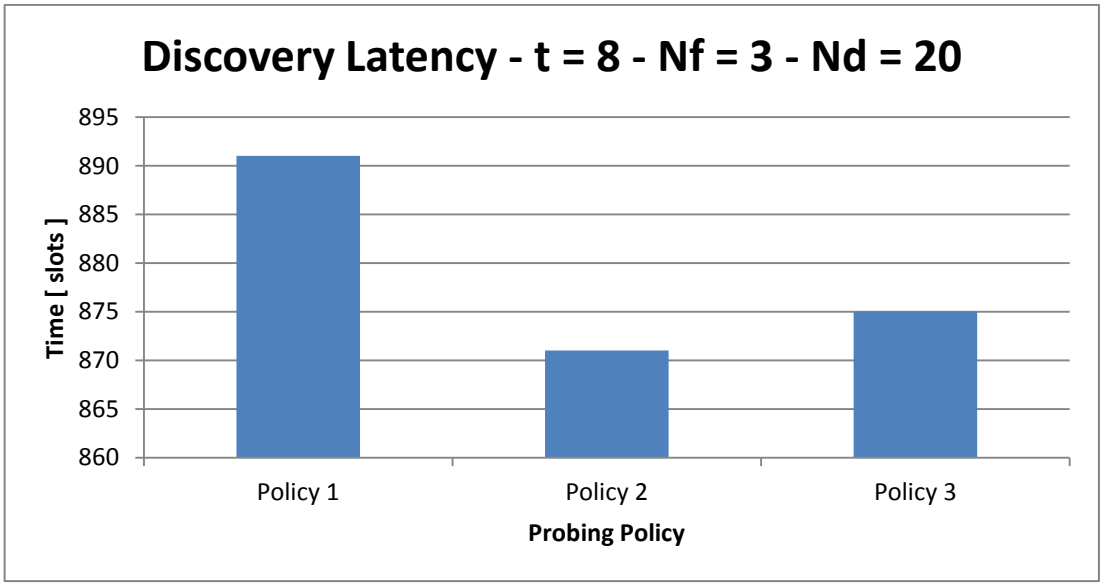

GRAFICO 4.48. Latenza media per 10 distribuzioni con Nf = 3 e Nd = 20

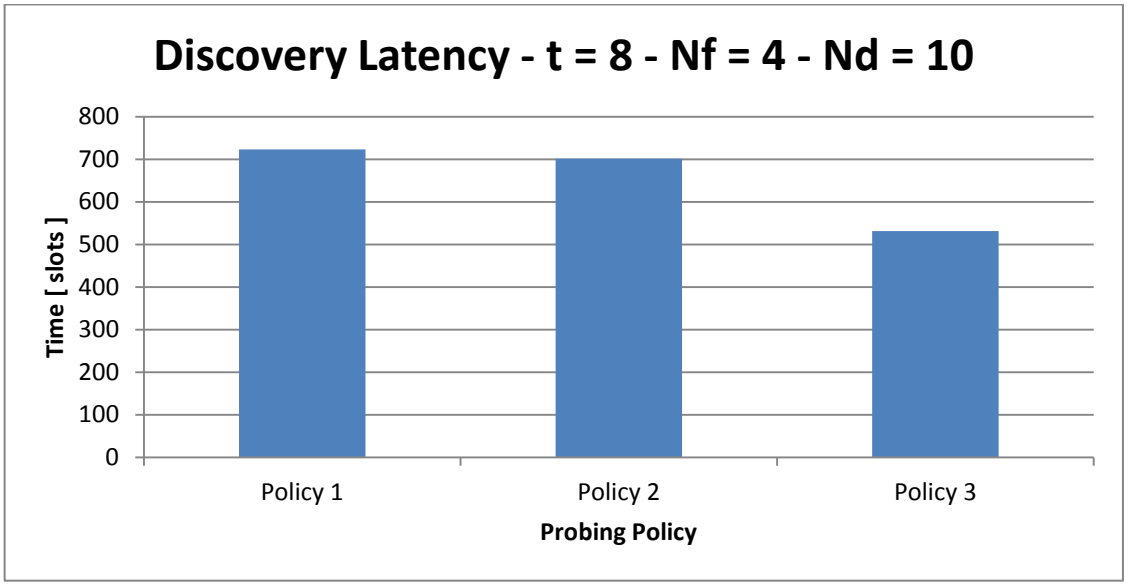

GRAFICO 4.49. Latenza media per 10 distribuzioni con Nf = 4 e Nd = 10

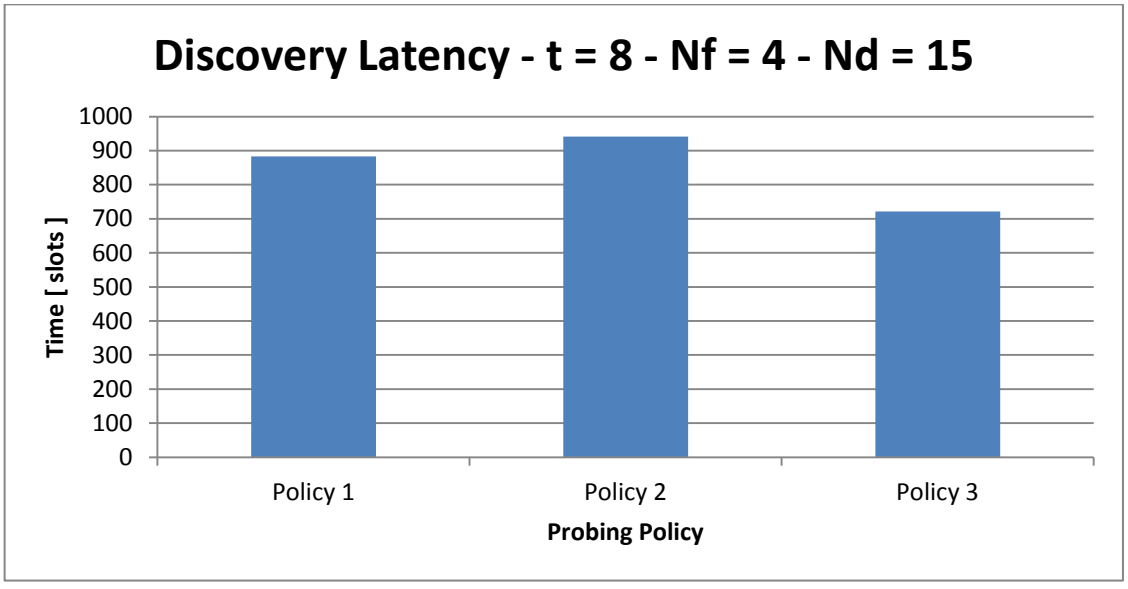

GRAFICO 4.50. Latenza media per 10 distribuzioni con Nf = 4 e Nd = 15

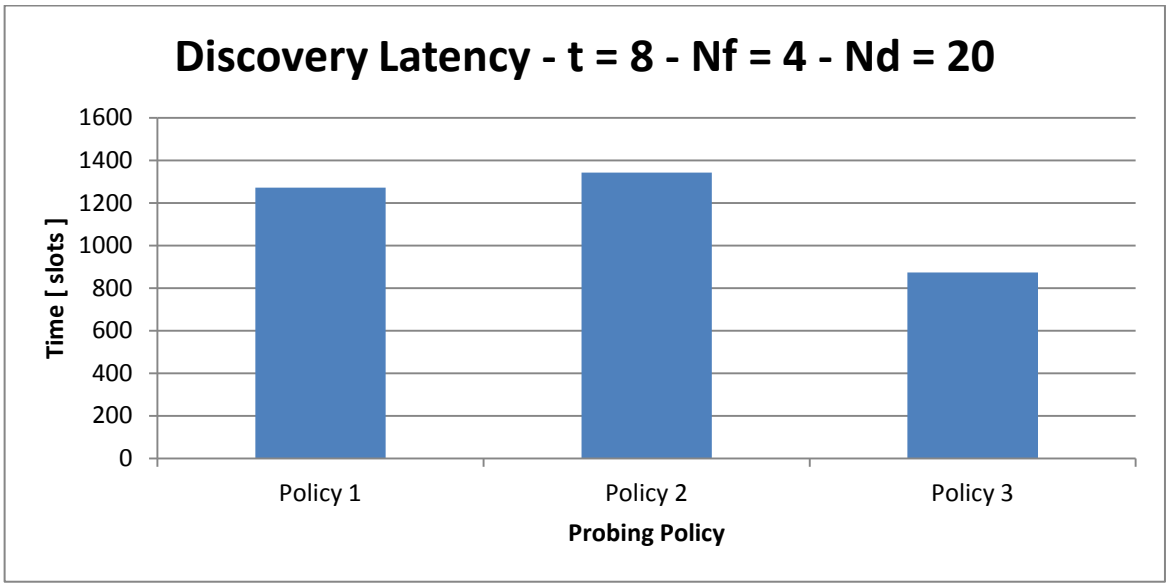

GRAFICO 4.51. Latenza media per 10 distribuzioni con Nf = 4 e Nd = 20

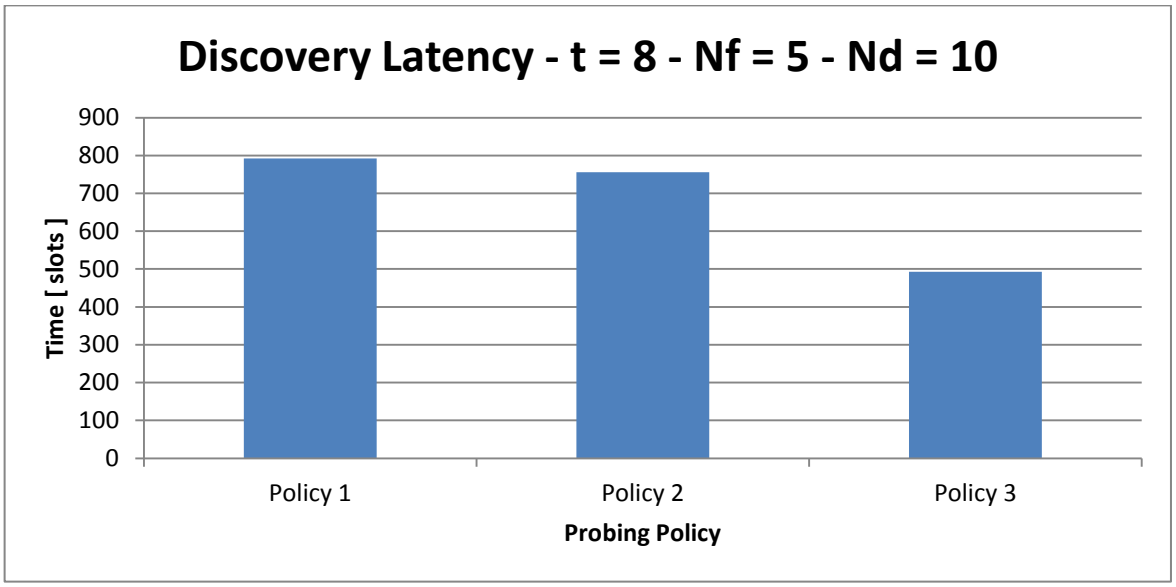

GRAFICO 4.52. Latenza media per 10 distribuzioni con Nf = 5 e Nd = 10

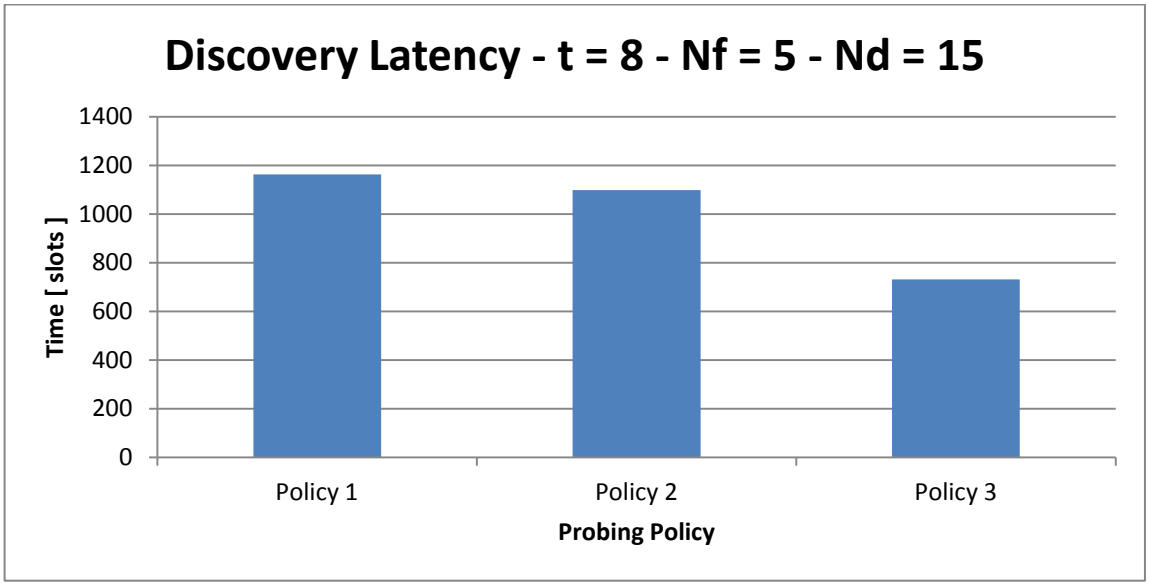

GRAFICO 4.53. Latenza media per 10 distribuzioni con Nf = 5 e Nd = 15

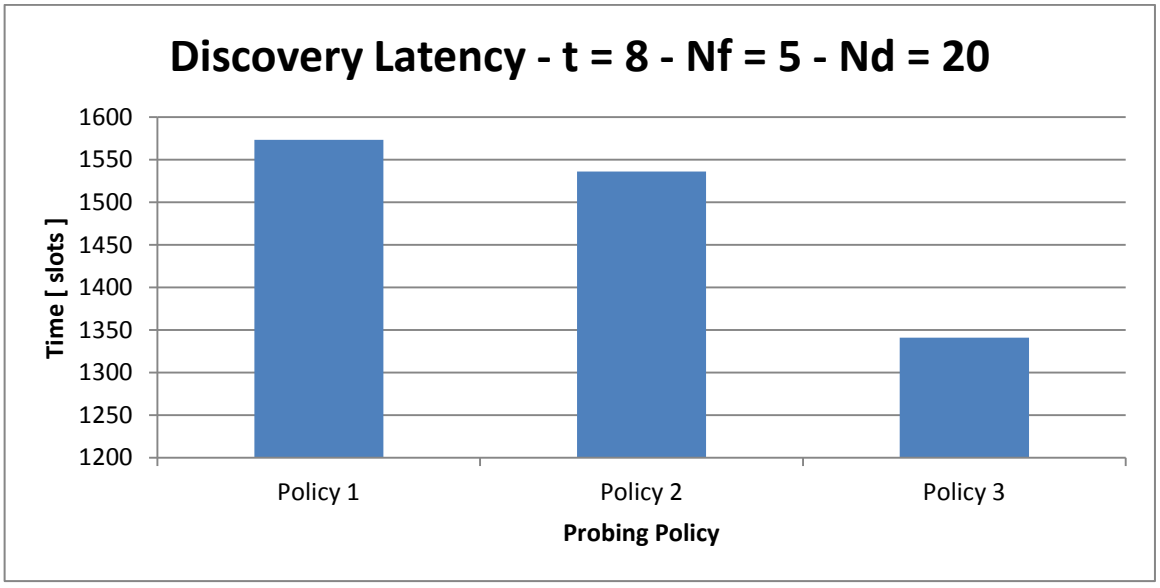

GRAFICO 4.54. Latenza media per 10 distribuzioni con Nf = 5 e Nd = 20

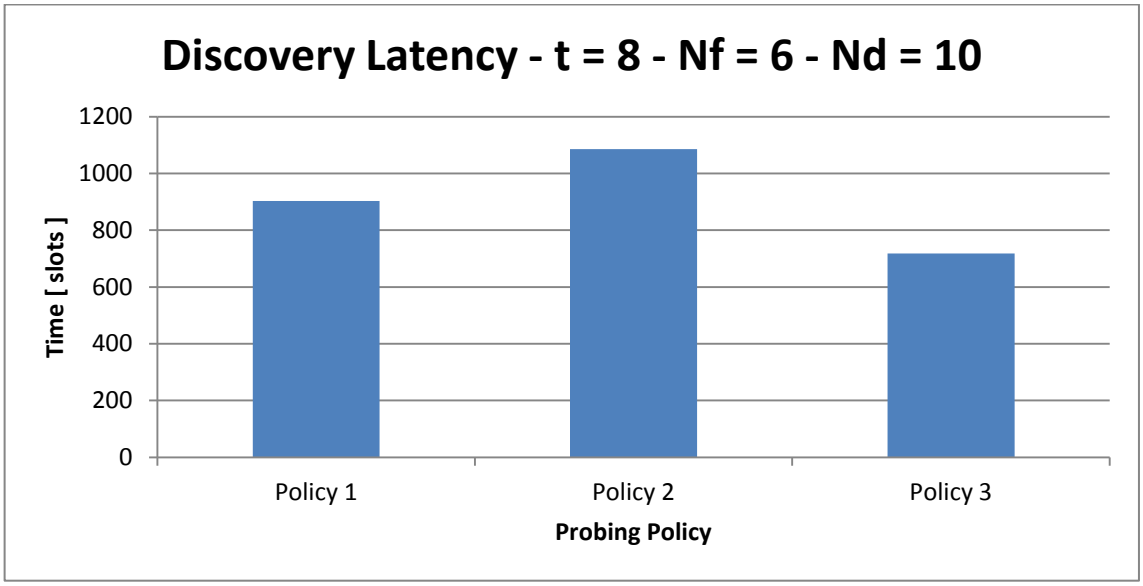

GRAFICO 4.55. Latenza media per 10 distribuzioni con Nf = 6 e Nd = 10

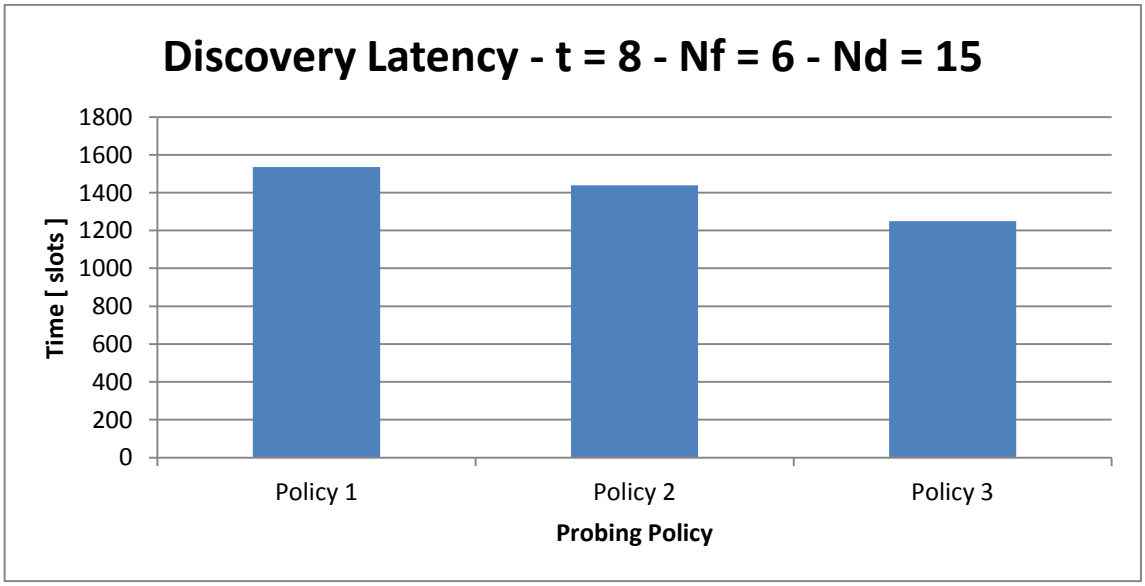

GRAFICO 4.56. Latenza media per 10 distribuzioni con Nf = 6 e Nd = 15

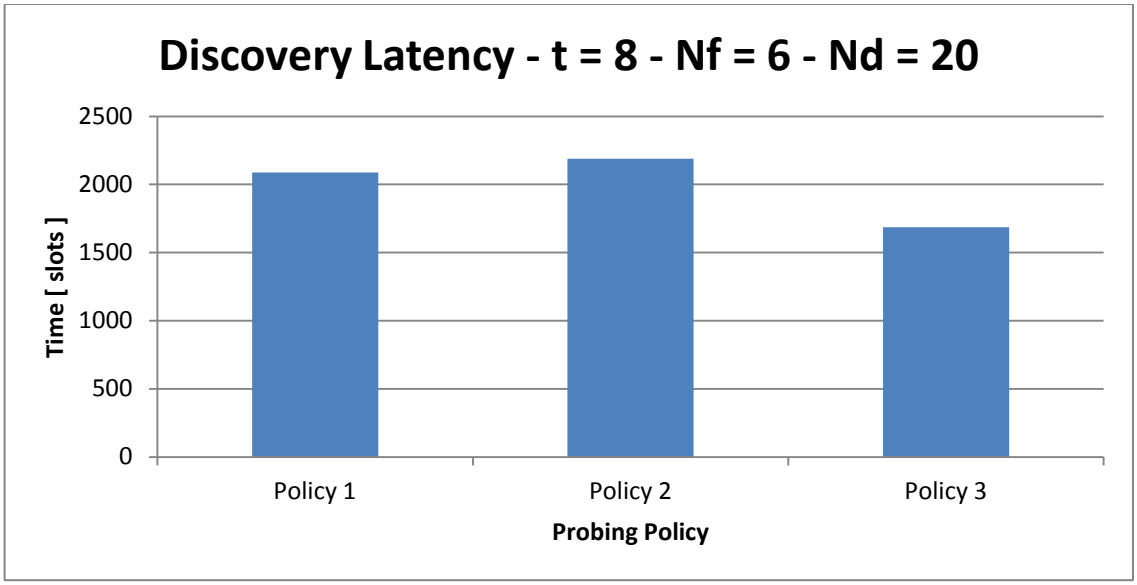

GRAFICO 4.57. Latenza media per 10 distribuzioni con Nf = 6 e Nd = 20

Dai grafici forniti ci rendiamo conto come quando il numero di frequenze è molto basso, per esempio Nf = 3, qualunque sia la politica di sondaggio utilizzata il tempo medio per la latenza è grosso modo il medesimo. Mentre quando il numero di frequenze utilizzabili dagli utenti aumenta, è chiaro come la politica 3, ovvero la politica di sondaggio ai bordi, è quella piu conveniente. Per quanto riguarda la politica 2, ci si rende conto che la latenza media fornita sondando le frequenze come precedentemente esposto è pressochè la medesima di quella che si trova effettuando un sondaggio casuale, ovvero con la politica 1 all'interno dei grafici.

Conclusione

Il protocollo di discovery Searchlight nella sua implementazione originale non considera l'estensione ad una scoperta multi frequenza e di un numero maggiore di un dispositivo. La necessità da parte di un dispositivo entrante di scoprire gli utenti della rete e la loro rispettiva frequenza di utilizzo è soddisfatta dal protocollo Searchlight seguendo le estensioni viste lungo questo studio.

Benchè nella progettazione di base di Searchlight venga enfatizzato un sostanzioso guadagno in termini di consumo energetico, abbiamo visto come questo risulti essere meno evidente quando il numero di utenti presenti nella rete ed il numero di frequenze utilizzabili nel suo interno aumenta in quanto il valore della latenza per avere la scoperta avvenuta con successo rischia di essere molto lunga nel tempo, riducendo in maniera sostanziosa le prestazioni.

Il trade-off fra tempo di attesa per la discovery e consumo energetico, rappresentato dalla lunghezza del periodo nel protocollo Searchlight, puo' essere effettuato tramite delle realizzazioni che sono considerate all'interno di questa tesi grazie al simulatore Visual Basic Net. Lo studio delle simulazioni effettuate ha portato a notare come se il numero di realizzazioni che vengono effettuate per stimare quale possa essere il trade-off ideale fra lunghezza del periodo temporale, ovvero duty-cycle, e tempo di attesa per la scoperta degli utenti , non viene scelto con accuratezza, le molteplici variabili delo scenario della rete ad hoc considerata rendono i valori ottenuti poco interessanti in quanto molto discostati dal valore medio teorico.

Bibliografia

[1] M.Bakht, M.Trower, R.Kravets " Searchlight : Won't you be my neighbor? "

[2] S.Lai, B.Ravindran, H.Cho " Heterogenous quorum-based wakeup scheduling in wireless sensor networks "

[3] Y.C.Tseng, C.S.Hsu, T.Yhsieh " Power-saving protocols for ieee 802.11-based multi-hop ad hoc networks

[4] M.J.McGlynn, S.A.Borbash "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks"

[5] http://it.wikipedia.org/wiki/MANET

[6] http://www.ietf.org/

[7] http://en.wikipedia.org/wiki/Hidden_node_problem

[8] A.Capone "Ad – hoc networks – Reti mobile distribuite"

[9] Teaching material of Prof. Eylem Ekici (Ohio State University at Columbus) and Prof. Nitin H. Vaidya (University of Illinois at Urbana-Champaign)

[10] P.Dutta and D.Culler "Pratical asynchronous neighbor discovery and rendezvous for mobile sensing applications"

[11] R.Zheng, J.C.Hou and L.Sha 'Asynchronous wakeup for ad hoc networks"

Elenco delle figure

Elenco dei grafici