



UNIVERSITA' DEGLI STUDI DI ROMA  
" LA SAPIENZA "

TESI DI LAUREA SPECIALISTICA IN  
INGEGNERIA DELLE TELECOMUNICAZIONI  
NUOVO ORDINAMENTO

# **Spotty Networks: da un accesso al mezzo distribuito alla creazione di gruppi coordinati**

*Candidato:*  
**Michele de Chirico**  
(matricola: 792329)

*Relatore:*  
**Prof. Maria Gabriella Di Benedetto**

*Correlatore:*  
**Ing. Luca De Nardis**

Anno accademico 2005-2006

*Alla mia Famiglia,  
la mia fonte di  
forza inesauribile.*

• **INDICE**

INTRODUZIONE.....	5
CAPITOLO 1.....	9
ANALISI DEGLI STANDARD STUDIATI.....	9
1.1 BREVE ANALISI DELLO STANDARD ZigBee .....	11

1.2 BREVE ANALISI DELLO STANDARD 802.15.4a .....	13
1.3 BREVE CONFRONTO 802.15.4a vs ZigBee.....	20
CAPITOLO 2.....	24
ANALISI DELLO SCENARIO DI LAVORO.....	24
2.1 LE RETI AD HOC.....	26
2.2 IL SISTEMA UWB NEL CONTESTO 802.15.4a.....	28
2.3 DEFINIZIONE E CARATTERISTICHE DELL'UWB RADIO.....	31
2.4 LA CLUSTERIZZAZIONE.....	37
CAPITOLO 3.....	43
ANALISI DI ALCUNI ALGORITMI DI CLUSTERING.....	43
3.1 ALGORITMO PER IL CONTROLLO DI ANOMALIE DOVUTE AD INTRUSIONI..	45
3.2 ALGORITMI PER FORNIRE MAGGIORE STABILITA' AI LAYER SUPERIORI.....	48
3.3 ALGORITMI PER FORNIRE MAGGIORE QoS, SICUREZZA ED UN ROUTING MIGLIORE.....	57
3.4 ALGORITMO BASATO SULLA DENSITA' DEL CLUSTERING.....	62
3.5 ALGORITMO BASATO SULLA DIMINUZIONE DELL'OVERHEAD DEI PACCHETTI.....	64
3.6 ALGORITMI BASATI SU UN USO EFFICIENTE DELL'ENERGIA.....	67
CAPITOLO 4.....	73
L'AMBIENTE DI SIMULAZIONE.....	73
4.1 STRUTTURA DEL SIMULATORE.....	75
4.2 MODELLO FISICO.....	77
4.3 MODELLO ED ANALISI DELLA MOBILITÀ.....	83
4.4 STRATO MAC.....	87
4.5 STRATO DI ROUTING.....	90
4.6 STRATO DI APPLICAZIONI.....	94
4.7 GESTIONE DELL'INTERFERENZA.....	95
4.8 IL CLUSTERING.....	98
4.9 MODELLO DI TRAFFICO.....	108
CAPITOLO 5.....	109
ANALISI DELLE SIMULAZIONI.....	109
5.1 ANALISI TOPOLOGIA GRID.....	112
5.2 ANALISI TOPOLOGIA RANDOM.....	116
CONCLUSIONI FINALI.....	129
APPENDICE A .....	131
A.1 La gerarchia dei moduli e Module Types.....	133
A.2 Messaggi, gates, links, parametri e trasmissione dei pacchetti.....	135
A.3 Descrizione della topologia ed uso di OMNet++.....	137
BIBLIOGRAFIA.....	139
Ringraziamenti.....	142



# INTRODUZIONE

Nel mondo delle comunicazioni tra dispositivi di ogni genere, è sempre più importante cercare di ottenere le migliori prestazioni proporzionalmente alla topologia ed alle richieste dello scenario in cui si opera.

L'avvento della tecnologia wireless unito alla richiesta di una maggiore mobilità dei dispositivi ha portato ad uno sviluppo maggiore verso apparati appartenenti ad un'emergente classe di architettura di reti self-organizing, rapidamente schierabili (ad hoc network).

Le prospettive future derivanti dalle caratteristiche delle reti ad hoc hanno portato allo studio ed alla definizione di svariate innovazioni inerenti a diversi aspetti dei sistemi di comunicazione, dal dimensionamento fino alle strutture hardware, passando per lo sviluppo dello strato delle applicazioni.

La maggiore innovazione introdotta da questo tipo di sistemi è l'introduzione dell'energy-aware, capace di estendere la vita totale del singolo apparato ricetrasmittente e di conseguenza di tutta la rete (condizione basilare nel momento in cui si ha una limitata potenza disponibile in ogni terminale).

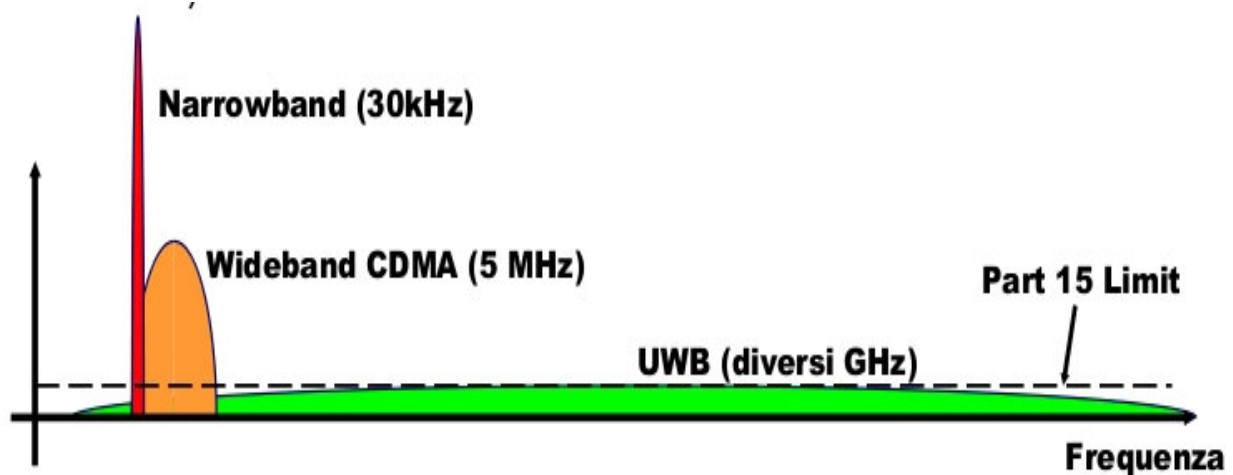
Una particolare topologia di reti ad hoc è data dalle reti di sensori, in cui ogni singolo apparato è capace di captare degli eventi esterni specifici, presenti in tutta l'area di sviluppo della rete, tipicamente per monitorare le variazioni del valore di uno o più parametri fisici (come umidità o temperatura): dato il posizionamento spesso non facilmente raggiungibile e la locazione di suddetti sensori, l'energia disponibile è una caratteristica chiave per tutta la gestione dell'intera rete.

Una soluzione ideale per lo spiegamento e la connessione delle reti ad hoc e delle schiere di sensori è l'Ultra Wide Band radio, che fa dell'alto bit rate in trasmissione, dell'alta robustezza al multipath in collegamenti NLOS e della possibilità di gestire il posizionamento degli apparati che compongono la rete, delle caratteristiche di

## 2 - INTRODUZIONE

grandissima attrattiva per sistemi di comunicazione wireless indoor e outdoor.

Queste peculiarità derivano dalla caratteristica chiave dei sistemi UWB, ovvero l'uso di una banda del segnale che si espande su diversi GHz, nella gamma di frequenze che va da 0 a 10 GHz: di conseguenza l'emissione copre una larga porzione dello spettro di frequenza e deve coesistere con la propagazione delle altre forme d'onda Hertziane presenti in aria (come mostrato in Figura 1): per questo motivo sono state create delle maschere di emissione per cercare di limitare l'interferenza che si ottiene nei confronti dei sistemi narrowband preesistenti.



**Fig.1: Analisi delle frequenze occupate dal segnale UWB.**

Basandosi sulle caratteristiche fin ora trattate, l'IEEE ha creato un nuovo Working Group volto alla definizione di un nuovo standard, l'802.15.4a, per sistemi di comunicazione indoor/outdoor (sensor network) con un'accurata capacità di posizionamento: questo nuovo standard è derivato dal già esistente 802.15.4 ed ha costituito la base per lo sviluppo dello standard industriale ZigBee.

Il nuovo standard adotta inizialmente il MAC di quello precedente (in attesa di una nuova definizione dello strato per usufruire meglio delle informazioni di posizionamento elaborate a strato fisico dell'802.15.4a) ed è stato creato soprattutto per comunicazioni a basso rate e con un numero abbastanza ridotto (nell'ordine di una centinaia) di apparati.

Tra le diverse tecniche applicabili all'intero network per ottenere migliori prestazioni,

### 3 - INTRODUZIONE

vi è sicuramente il clustering, (analisi dei cluster o analisi di raggruppamento) una tecnica di analisi multivariata di informazioni volta alla selezione ed al raggruppamento di elementi omogenei in un insieme di dati.

In un sistema di telecomunicazioni nel momento in cui si ha un numero elevato di componenti della network, è possibile adottare la tecnica precedentemente citata, in modo da ottenere migliori prestazioni in base alle richieste del cliente che usufruirà in futuro del servizio.

Tutte le tecniche di clustering si basano sul concetto di distanza tra due elementi, infatti la bontà delle analisi ottenute da suddetti algoritmi dipende essenzialmente da quanto è significativa la metrica e quindi da come è stata definita la distanza (concetto fondamentale in quanto l'appartenenza o meno di un nodo ad un insieme dipende da quanto l'elemento preso in esame dista dall'insieme stesso).

In questo lavoro si cerca di trovare l'algoritmo di clusterizzazione che dia le migliori prestazioni in un'ad hoc network formata da sedici Host; nel Capitolo 1 saranno brevemente trattati gli standards 802.15.4a (ed il protocollo d'accesso (UWB)<sup>2</sup>) e ZigBee, con un confronto sulle caratteristiche principali dei due.

Nel Capitolo 2 saranno descritte le caratteristiche dell'UWB radio, dalla formazione del segnale alle modulazioni PPM e PAM, quali sono le caratteristiche degli algoritmi di clusterizzazione (in generale) ed i vantaggi della loro applicazione ad una ad hoc network.

Nel Capitolo 3 sono analizzati diversi algoritmi di clustering implementati ed illustati da diversi ingegneri all'interno delle MANET, in modo da vedere quali sono le caratteristiche principali e gli accorgimenti più usati per nuove topologie di rete.

Nel Capitolo 4 è illustrato lo scenario delle simulazioni, i cinque diversi algoritmi di clusterizzazione implementati (tre per la topologia GRID e due per quella RANDOM) e le funzioni basilari che compongono i diversi moduli creati per OMNet ++.

Nel Capitolo 5, infine, saranno spiegati tutti i parametri settati nel sistema e saranno illustrati i risultati delle simulazioni per ogni singola topologia di rete.

# **CAPITOLO 1**

## **ANALISI DEGLI STANDARD STUDIATI**

Lo studio e l'evoluzione delle comunicazioni a basso data rate e costo contenuto degli apparati usati, sia in ambiente indoor che outdoor, è un argomento sempre attuale e di grandissimo interesse all'interno delle reti ad hoc formate da schiere di sensori.

L'interesse verso network a basso rate ha portato, nel 2003, alla definizione dello standard IEEE 802.15.4 per sistemi wireless a low rate, bassa complessità e bassa potenza in trasmissione; l'802.15.4, inoltre, forma la base della tecnologia ZigBee che fornisce una soluzione molto esaustiva per il passaggio del traffico low rate dallo strato fisico alle applicazioni.

Il problema dello standard IEEE 802.15.4 e dello ZigBee, però, è quello di non essere forniti di una caratteristica (al giorno d'oggi) fondamentale per sistemi che trasmettono a basse distanze e rate contenuto: la capacità di localizzare, con precisione sufficientemente alta, oggetti ed individui attraverso l'uso di algoritmi distribuiti per il posizionamento di infrastrutture indipendenti.

L'introduzione della localizzazione della posizione dei sistemi è stata, di fatto, nelle reti a basso rate, la maggiore priorità dell'IEEE 802.15.4a Task Group recentemente formato; questo gruppo, infatti, ha proposto l'IR-UWB (Impulse Radio Ultra Wide



## 5 - ANALISI DEGLI STANDARD STUDIATI

Band) come una tecnica molto attrattiva di trasmissione per networks wireless a low rete in ambienti sia indoor che outdoor.

Successivamente la definizione del protocollo (UWB)<sup>2</sup> (Uncoordinated Baseborn Wireless medium access control for UWB networks) si è basata su specifiche e fondamentali caratteristiche dell'IR-UWB; (UWB)<sup>2</sup>, inoltre, stima ed immagazzina l'informazione inerente alle distanze degli apparati, per poter successivamente applicare gli algoritmi di positioning e routing.

In questo capitolo saranno brevemente trattati, nei loro punti essenziali, gli standard precedentemente citati; infatti, dopo un'analisi di ZigBee, dell'IEEE 802.15.4a e del conseguente protocollo di accesso al mezzo (UWB)<sup>2</sup>, sarà presente una terza ed ultima parte in cui si evidenzieranno le principali differenze tra questi due standard.

# **1.1 BREVE ANALISI DELLO STANDARD ZigBee**

## 6 - ANALISI DEGLI STANDARD STUDIATI

ZigBee Alliance è un consorzio mondiale di aziende, aperto, no-profit ed in rapida crescita, la cui missione è definire prodotti per il controllo, la rilevazione e il monitoraggio basati su uno standard aperto e che siano affidabili, economici, a basso consumo energetico e collegati in reti wireless ([1] e [2]).

Requisiti fondamentali delle applicazioni in cui è presente questo standard sono semplicità, lunga durata delle batterie, capacità di collegamento in rete, affidabilità e costi ridotti.

ZigBee sfrutta appieno un potente physical layer radio e un MAC definito dalla specifica 802.15.4 messa a punto dall' IEEE.

Questa specifica definisce la modalità di conversione dei bit di dati nelle trasmissioni radio e un efficiente protocollo per consentire a più dispositivi di condividere lo stesso canale radio. Lo standard ZigBee aggiunge, inoltre, un software che illustra come è costruita una rete, come funziona, e definisce una modalità standard per aggiungere software applicativi; ZigBee Alliance fornisce anche test di interoperabilità, certificazione e branding, ed inoltre è in grado di operare correttamente anche quando uno o più nodi della rete non sono disponibili o sono danneggiati.

Una rete ZigBee è formata da due o più dispositivi associati tra loro; sono supportate diverse configurazioni di rete, tra cui reti STAR, CLUSTER TREE e MESH (meglio analizzate nel capitolo successivo), proprio come avviene per lo standard 802.15.4a, ed anche in questo caso sono presenti principalmente due tipi di dispositivi (che svolgono le stesse funzioni dell'802.15.4a): RFD (Reduced-Function Device) ed FFD (Full-Function Device), la cui trattazione è presente nel paragrafo successivo.

Un concetto fondamentale della tecnologia ZigBee è la capacità di molti dispositivi radio a breve raggio di coprire un'area estesa trasferendo messaggi attraverso più "hop": di conseguenza, un dispositivo a un capo della rete è in grado di inviare informazioni a un dispositivo all'altro capo, anche se questi sono abbastanza distanti da non rientrare direttamente all'interno dei reciproci raggi d'azione.

Per quanto riguarda le caratteristiche principali di ZigBee, si può affermare che sono

## 7 - ANALISI DEGLI STANDARD STUDIATI

molto simili (se non coincidenti fino allo strato MAC) a quelle del protocollo 802.15.4; inoltre tutte le tecniche di clusterizzazione ed elezione del CH (Cluster Head) sono uguali per entrambi i protocolli.

Alcune differenze si possono notare analizzando gli strati più alti della pila protocollare dei due standard: compito dell'ultimo paragrafo è quello di evidenziare quali sono le innovazioni apportate da ZigBee a livello applicativo.

## **1.2 BREVE ANALISI DELLO STANDARD 802.15.4a**

Questo standard definisce il protocollo ed il collegamento di dispositivi di comunicazione via radio in una PAN (Personal Area Network); le WPANs (Wireless PAN), dunque, sono usate per trasportare informazioni su distanze relativamente brevi. La connessione che si ottiene attraverso WPANs, a differenza delle WLANs (Wireless Local Area Network), implica una presenza sporadica o addirittura assente di infrastrutture ([3] e [4]).

Questo standard usufruisce di un meccanismo di carrier sense multiple access con collision avoidance (CSMA-CA) per accedere alla risorsa e supporta tipologie di rete sia a Stella che Point-to-point.

L'accesso al mezzo è basato su un meccanismo di contesa e la connettività tra i vari apparati, per ottenere prestazioni di rete maggiori, è resa possibile attraverso un PAN coordinator.

## 8 - ANALISI DEGLI STANDARD STUDIATI

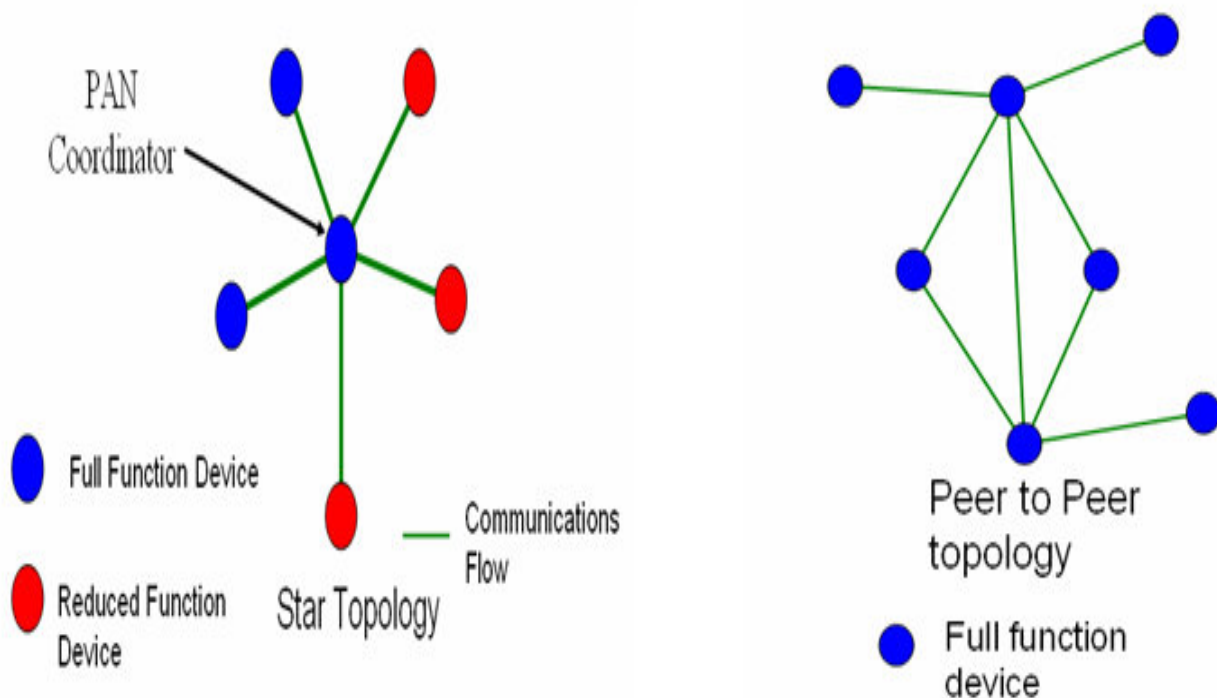
Questo standard è nato per rispondere a requisiti di sistema come una complessità molto bassa, un costo molto contenuto, un consumo di potenza molto basso e un data rate abbastanza contenuto (circa  $250 \div 20$  kb/s in base all'area d'applicazione) per connettività radio tra dispositivi poco costosi, fissi, portatili o mobili, che operano in un POS (Personal Operating Space ) di circa 10 m.

In questo sottogruppo di reti denominato LR-WPAN (Low Rate-WPAN) si possono distinguere due tipi di dispositivi:

- a) FFD (Full-Function Device): possono operare in tre modi diversi, ovvero come coordinatore della PAN, semplice coordinatore o dispositivo; possono dialogare con gli RFDs o con altri FFDs, a differenza degli RFDs che possono farlo solo con gli FFDs.
- b) RFD (Reduced-Function Device): usati per applicazioni molto semplici, in cui non si ha un elevato traffico di dati e si può avere solo una comunicazione univoca (in un determinato lasso di tempo) tra RFD e FFD; in questo modo un RFD può essere implementato usando minime risorse e capacità.

Come precedentemente anticipato, a seconda delle specifiche applicazioni, una LR-WPAN può operare in due diverse topologie di rete (Figura 1.1):

- 1) A stella: la comunicazione è istaurata tra i vari dispositivi e un coordinatore centrale, che gode di svariate proprietà che lo rendono il fulcro della rete (PAN coordinator); tipicamente un dispositivo ha diverse applicazioni associate ed inoltre può essere il punto iniziale o finale di una comunicazione di rete.
- 2) Point-to-point: anche questa topologia ha un PAN coordinator, ma differisce dalla precedente in quanto ogni dispositivo può comunicare autonomamente con un altro, purché questo appartenga al suo raggio di copertura radio; è una struttura di rete molto più complessa da implementare e può essere ad hoc, self-organizing e self-healing.



**Fig. 1.1: Topologia di rete A Stella e Point-to-Point [4].**

E' molto interessante analizzare più a fondo la topologia Point-to-point in quanto ha come suo esempio classico l'organizzazione della rete in cluster-tree; in questo caso la maggior parte dei dispositivi sono FFDs, mentre un RFD si può connettere all'albero solo come nodo alla fine di un ramo, in quanto può essere associato solo con un FFD alla volta.

Alcuni FFD possono agire come coordinatori e possono provvedere alla sincronizzazione con altri dispositivi o altri coordinatori; soltanto uno di questi coordinatori può essere il PAN coordinator globale, che può avere maggiori risorse computazionali rispetto ad ogni altro apparato appartenente alla PAN stessa.

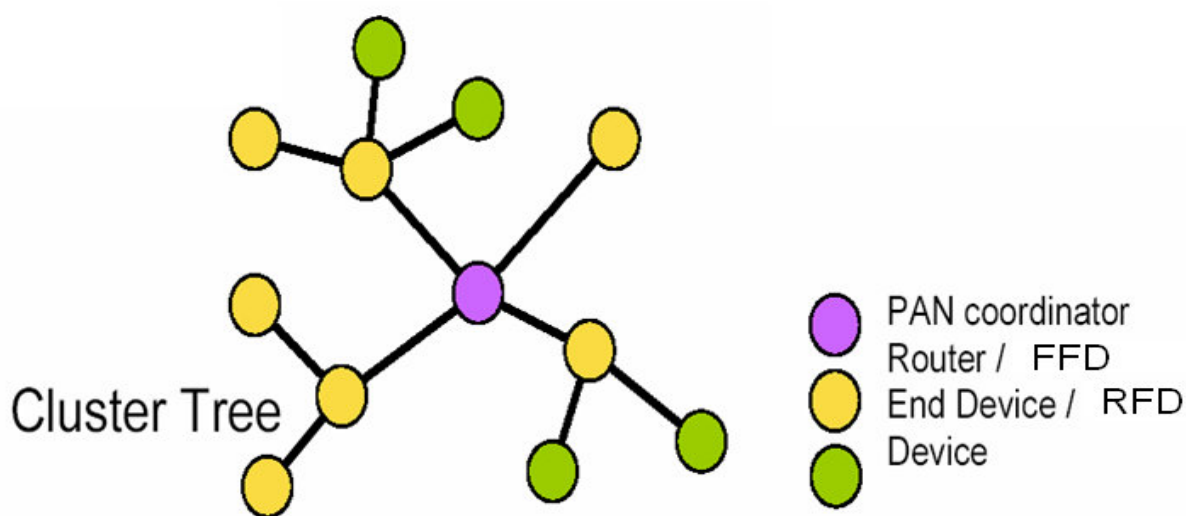
Il PAN coordinator forma il primo cluster eleggendosi come cluster capo (CH) con un cluster identifier (CID) pari a zero, un PAN identifier (non usato) e diffondendo in broadcast dei beacon frames ai dispositivi vicini.

Successivamente, un apparato candidato a far parte del cluster, che ha ricevuto i beacon frames, può richiedere di essere associato al cluster stesso; in questo caso il

## 10 - ANALISI DEGLI STANDARD STUDIATI

CH deciderà se portare a termine o meno questa operazione, aggiornando di conseguenza la sua lista di appartenenza; se il candidato non è accorpato al cluster dal CH, è libero di trovare un'altra collocazione.

L'esempio più semplice di cluster-tree è una singola rete di cluster, ma è possibile ottenere reti più larghe formate da diverse maglie di cluster multipli vicini (Figura 1.2).



**Fig. 1.2: Esempio di Cluster-tree [4].**

Una volta esaurite tutte le applicazioni predeterminate e le richieste di rete, il PAN coordinator può istruire un dispositivo a diventare il CH di un nuovo cluster a se adiacente: in questo modo gli altri dispositivi si andranno a connettere gradatamente formando una struttura di rete multicluster, col vantaggio di aumentare l'area totale di copertura a discapito, però, di un incremento del tempo di latenza dei messaggi.

Nei sistemi UWB considerati nel proseguo del lavoro, è usato un modello analitico per la MUI (Multi User Interference) in cui si definisce un protocollo MAC per networks a basso rate basato sulla probabilità di collisione degli impulsi del segnale; considerando il BER (Bit Error Rate) ed il PER (Packet Error Rate) come funzioni dei parametri che contraddistinguono i sistemi UWB, si seleziona una strategia di

sincronizzazione capace di garantire un'alta probabilità di successo allo strato fisico. Sfruttando i parametri dei sistemi UWB (come ad esempio un alto guadagno di processo), si può ottenere un'alta probabilità di sincronizzazione anche nel caso di reti asincrone, in cui più utenti accedono alla risorsa senza nessun tipo di coordinazione, né centralizzata né distribuita.

Queste caratteristiche formano la base per la definizione del protocollo MAC (Medium Access Control) chiamato (UWB)<sup>2</sup> (Uncoordinated Baseborn Wireless medium access control for UWB networks), in cui gli utenti inviano pacchetti informativi, senza nessun tipo di coordinazione ([5]).

(UWB)<sup>2</sup> trae vantaggio, in trasmissione dati, dalla capacità di accesso multiplo data dai TH-Codes (Time Hopping Codes) e si basa, per quanto concerne l'accesso al canale condiviso, sulla grande robustezza alla MUI, data dal valore del guadagno di processo nei sistemi UWB.

Nel momento in cui si seleziona il TH come tecnica di codifica, il TH-CDMA (Time Hopping Code Division Multiple Access) diventa una scelta naturale per quanto riguarda l'accesso multiplo; l'adozione del TH-CDMA può introdurre un addizionale livello di libertà, dato dal fatto che l'effetto delle collisioni degli impulsi è abbastanza ridotto attraverso l'adozione di codici differenti su diversi links.

Nella determinazione della robustezza alla MUI cooperano due principali fattori:

1. un basso duty cycle dei segnali emessi;
2. l'associazione di diversi TH-Codes sui diversi links.

La robustezza del sistema alla MUI, inoltre, è determinata dalle proprietà di cross correlazione dei codici: più bassa è la cross correlazione tra due diversi codici, più alto è il numero di possibili trasmissioni simultanee.

(UWB)<sup>2</sup> è un protocollo MAC multicanale basato sulla combinazione dell'accesso Aloha con il TH-CDMA; adotta la contemporanea coesistenza tra un codice *Comune* per la segnalazione, dove i terminali dividono proprio lo stesso codice (e si evitano collisioni grazie ad un opportuno *phase shift* tra link diversi e la scelta di appropriati protocolli di selezione del codice), ed un codice *Trasmittente*, dove ogni singolo

## 12 - ANALISI DEGLI STANDARD STUDIATI

terminale ha un codice “personalizzato” per la trasmissione ed il ricevitore si sincronizza con esso per la decodifica dei dati e la ricezione dei pacchetti informativi. Il protocollo (UWB)<sup>2</sup> applica il concetto di multi-code allo specifico caso di un sistema TH-IR UWB, adottando uno schema ibrido di codici, come precedentemente descritto. Lo scambio di dati attraverso il trasmettitore (TX) ed il ricevitore (RX), che prende spazio durante la fase di set-up della connessione, serve anche per rendere possibile una semplice procedura di allineamento, basata su tre fasi: durante una fase di set-up, il TX e l’RX preparano un packet DATA di trasmissione, scambiano un LE (link establishment) packet trasmesso con il Common Code, seguito da un LC (link confirm) packet trasmesso sul Transmitter Code dell’RX ed alla fine attraverso il DATA packet sul codice *Trasmittente* del TX .

Durante questa procedura, inoltre, si raccolgono informazioni sul posizionamento dei terminali, in cui ogni dispositivo  $i$  mantiene un database sul posizionamento di tutti i dispositivi vicini; ogni informazione entrante nel database contiene l’ID  $j$  del vicino, la distanza stimata da esso ed un *timestamp* che indica il tempo al quale è stata fatta la stima precedentemente illustrata.



## 1.3 BREVE CONFRONTO 802.15.4a vs ZigBee

Com'è da abitudine di tutti gli standards IEEE 802, l'802.15.4a è stato studiato soltanto con lo scopo di creare delle specifiche al physical layer ed in porzioni del DLL (Data Link Layer).

Lo studio dei protocolli presenti agli strati più alti è lasciato all'industria delle telecomunicazioni oppure ad applicazioni individuali (come illustrato in [6]).

ZigBee Alliance è un'associazione di compagnie coinvolte nella creazione di standards di livello superiore basati sull'IEEE 802.15.4; in Figura 1.3 sono illustrate tre diverse pile protocollari che mettono in risalto quale sia stata l'evoluzione avuta nella creazione di strati di livello sempre più alto, da semplici comunicazioni in ambienti Wireless a comunicazioni tra gruppi di nodi (clusters) sempre più complesse [6]:

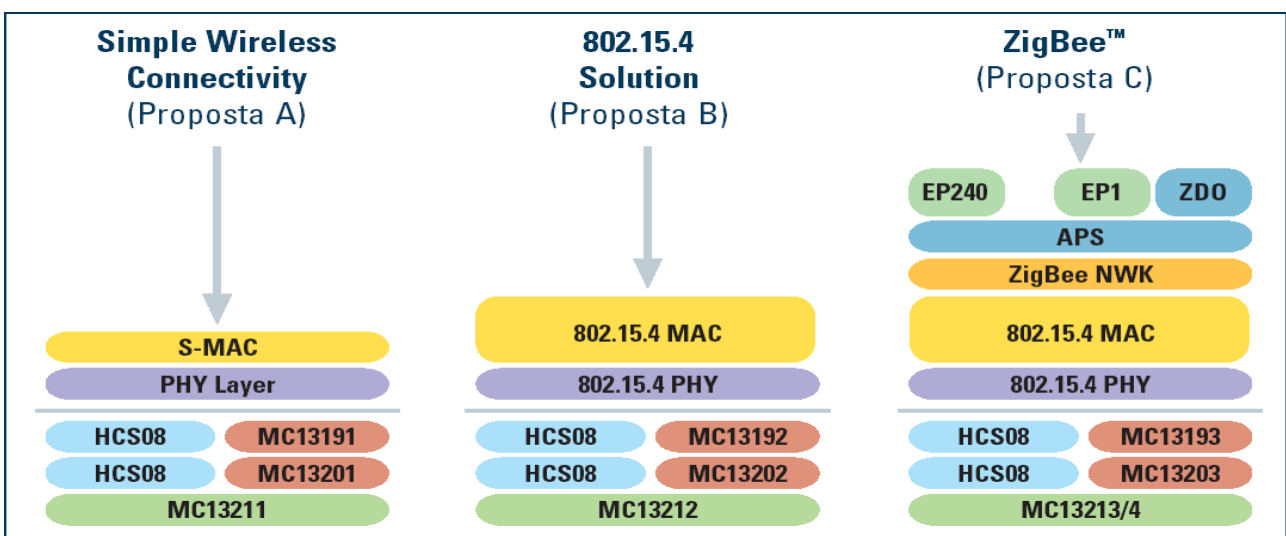


Fig. 1.3: Panoramica dell'evoluzione della pila protocollare in comunicazioni Wireless sempre

**più complesse [6].**

Mentre la Proposta A, per connessioni Wireless semplici, consente solo di realizzare topologie di rete A Stella o Point-to-point in soluzioni proprietarie (senza poter migrare verso la Proposta successiva), la Proposta B consente la realizzazione di reti proprietarie fino ad ottenere un'organizzazione più complessa di tipo cluster-tree; inoltre prevede maggiore sicurezza, ulteriore risparmio energetico grazie al beacon mode e livello MAC secondo le specifiche dello standard, oltre alla possibilità di poter migrare facilmente verso la Proposta successiva.

Per quanto riguarda la Proposta C, infine, si ha un'offerta molto completa sia in termini di prestazioni che di flessibilità: si ha un supporto totale di tutti i tipi di rete, con un livello di protezione più elevato rispetto alle specifiche precedenti.

I punti di forza di ZigBee sono l'APS (Application Support Sub-Layer ) e lo ZDO (ZigBee Devices Object): l'APS ha il compito di agire da interfaccia tra strato NWK (strato di rete) ed APL (strato d'applicazione) attraverso un insieme generale di servizi che sono usati dagli oggetti presenti nello ZDO e dallo ZDO stesso (con la presenza di due entità diverse per la trasmissione di dati tra più dispositivi e per il discovery o il binding di altri dispositivi in rete).

Per quanto concerne lo ZDO, invece, questo ha il compito di definire il ruolo dei dispositivi all'interno della rete (End Point, EPx), scoprirne nuovi, determinare a che applicazioni servono, inizializzare richieste o rispondere a richieste di binding e stabilire una relazione sicura tra dispositivi della rete.

In Figura 1.3, inoltre, sono illustrati i tre apparati ricetrasmittitori in pacchetto (SiP) che supportano la tecnologia ZigBee (MC1321Xx), posti naturalmente al di sotto dello strato fisico e del successivo strato di accesso al mezzo: sono la soluzione ideale di basso costo per le applicazioni della rete a Stella e Point-to-Point, con una bassa tensione del segnale IN/OUT ed una bassa potenza erogata.

Gli MC1319x, inoltre, sono un'altra famiglia di ricetrasmittitori (sempre della Freescale) che servono ad attuare una trasmissione a bassa potenza e basso raggio d'azione in sistemi Point-to-Point ed a Stella molto semplici, grazie all'uso di

## 15 - ANALISI DEGLI STANDARD STUDIATI

particolari software che contribuiscono ad attuare diverse funzioni di controllo, soprattutto in reti in cui si ha un'organizzazione in cluster; un altro apparato che costituisce la piattaforma illustrata di ricetrasmisione è l'HCS08 ed è un microcontrollore che agisce sempre a bassa potenza e bassa banda (munito anch'esso di appropriati software) per aumentare le prestazioni complessive del sistema.

Per concludere si possono illustrare schematicamente le diverse peculiarità degli standard precedentemente illustrati:

- L'802.15.4a è adeguato per differenti architetture di ricevitori e diversi bilanciamenti, per supportare diversi bit rate e diversi range, offrire una migliore organizzazione nella fruizione dei servizi offerti e conformarsi alle regolamentazioni mondiali; inoltre sono in fase di studio nuove implementazioni del protocollo, miglioramento in fase di consumo di energia e migliore scalabilità. Problemi ancora da risolvere sono l'incertezza sulle prestazioni in diversi range, un miglior accertamento della robustezza del sistema alle interferenze e la limitazione in alcune applicazioni dovuta ad un basso duty cycle in bande basse.
- Per quanto riguarda ZigBee, si ha un'ottima flessibilità dovuta alla possibilità da parte dei nodi di autoconfigurarsi (e risolvere guasti), una grande possibile interoperabilità con altri costruttori, una grande semplicità d'uso grazie ad HW e SW molto diffusi, un basso consumo di energia e basso costo.

Attraverso le innovazioni apportate da ZigBee dal punto di vista applicativo, dunque, si potrebbe cercare di intervenire per risolvere soprattutto i problemi di sicurezza e prestazioni presenti in 802.15.4a.

## **CAPITOLO 2**

# **ANALISI DELLO SCENARIO DI LAVORO**

Le tecniche di clusterizzazione illustrate nella seconda parte di questo lavoro (nel Capitolo 4) saranno ottimizzate per uno scenario composto da sistemi di trasmissione UWB (Ultra Wide Band), inseriti all'interno di un contesto più ampio, in cui si considera la coesistenza di più dispositivi organizzati in modo completamente decentralizzato: le reti ad hoc.

I sistemi UWB precedentemente citati sono nati come soluzione di scambio dati più sicuro in sistemi radar; successivamente si sono applicati ad una più ampia gamma di sistemi di telecomunicazioni grazie alla loro capacità di diminuire il rumore di

interferenza multiutente (MUI) attraverso appropriate tecniche di codifica.

Come conseguenza di queste caratteristiche, un sistema UWB ha la potenzialità di allocare comunicazioni simultanee tra un largo numero di utenti ad un elevato bit rate; inoltre, l'alta risoluzione temporale inerente a questo tipo di sistemi, conferisce una buona robustezza contro il fading multicammino, da cui nasce la particolare attrattiva per applicazioni in LAN (Local Area Network ) indoor.

Un sistema UWB è anche capace di recuperare e fornire informazioni sulla posizione degli apparati con grandissima precisione, fornendo un'organizzazione della rete più precisa e contribuendo ad ottenere un maggiore risparmio di batteria, sfruttando la direttività in trasmissione e ricezione degli apparati.

Grazie alle altissime frequenze d'azione dei segnali UWB, inoltre, si evitano sovrapposizioni con segnali a frequenze minori, a discapito, però, della necessità di regole severe sulla limitazione della loro densità di potenza, proprio per evitare problemi con sistemi che trasmettono in bande strette; questo è il problema principale da considerare nella creazione di un algoritmo che possa effettuare una clusterizzazione efficiente della rete, insieme ad altri accorgimenti necessari per prestazioni complessive migliori.

In questo capitolo saranno analizzati principalmente tre aspetti che costituiranno la base di tutto il lavoro svolto successivamente:

- introduzione al concetto di rete ad hoc;
- analisi delle principali caratteristiche dei sistemi UWB;
- introduzione al concetto di clusterizzazione .

Per le reti ad hoc, si farà riferimento brevemente a quali sono le loro principali caratteristiche e quali sono i vantaggi e gli svantaggi che si possono trarre dal loro uso.

Per quanto concerne i sistemi UWB, invece, saranno illustrate le caratteristiche di low-power, coesistenza, localizzazione e di trasmissione del segnale, con un cenno anche a due tipi di modulazioni possibili: PPM (Pulse Position Modulation) e PAM (Pulse Amplitude Modulation) .

Per la clusterizzazione, invece, saranno introdotti i concetti base per poter meglio comprendere le tecniche spiegate ed applicate allo scenario nei capitoli successivi.

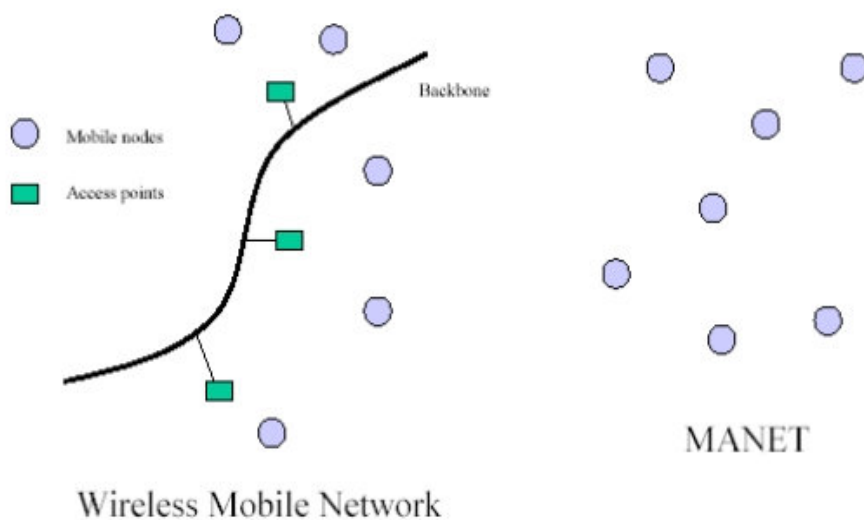
## 2.1 LE RETI AD HOC

Le reti ad hoc sono strutture costituite da un insieme di terminali autonomi che mantengono la connettività in modo decentralizzato, senza posarsi su infrastrutture fisse preesistenti (non c'è necessità di access point).

Ciascun nodo opera come router e inoltra pacchetti per gli altri nodi della rete (Figura 2.1).

Nel caso di terminali mobili la rete è più propriamente chiamata MANET (Mobile Ad Hoc Network).

La rete, quindi, è indipendente e autoconfigurata e permette una grande dinamicità dei nodi e un basso costo di setup ([7]).



**Fig. 2.1: Rete ad hoc [7].**

Le motivazioni che hanno portato allo sviluppo di questa tecnologia sono le seguenti:

- necessità di un maggiore supporto alla mobilità;
- bisogno di superare la ridotta tollerabilità ai guasti propria dei sistemi centralizzati;
- necessità di consentire la comunicazione in “ambienti ostili”, dove l’infrastruttura fissa non garantisce funzionalità.

Applicazioni di questa tecnologia fanno riferimento al campo militare, alla gestione dei disastri, alle reti NAN (Neighborhood area networks, accesso a internet condiviso in zone ad alta densità urbana) ed alle comunicazioni improvvisate tra gruppi di utenti.

L’attenzione rivolta alle reti ad hoc tanto dagli organi di standardizzazione quanto dagli sviluppatori di hardware e software per questa tipologia di reti, è giustificata da alcune caratteristiche proprie delle MANET che possono decretarne un ampio successo. Le caratteristiche di base di questo tipo di network sono:

- topologia dinamica;
- eterogeneità di terminali collegabili;
- collegamenti a capacità variabile a larghezza di banda limitata;

Al tempo stesso, però, le reti ad hoc presentano diversi limiti legati alle caratteristiche intrinseche dell’architettura di rete; in particolare si evidenziano:

- problema nel controllo dei terminali collegati;
- gestione difficoltosa della globalità della rete;
- nodi con limitate capacità di archiviazione dati e ridotta autonomia (legata alle batterie del terminale collegato).

## **2.2 IL SISTEMA UWB NEL CONTESTO 802.15.4a**

Negli ultimi anni il concetto di rete ad hoc ha acquisito sempre maggior interesse grazie alle sue caratteristiche di flessibilità, auto-organizzazione e soprattutto all'introduzione di sistemi sempre più sofisticati per l'autocontrollo ed autogestione dell'energia, che hanno reso possibile l'estensione della vita dell'intero network, nel momento in cui si aveva una potenza disponibile limitata da parte di tutti i dispositivi. Quest'ultima caratteristica è stata fondamentale soprattutto per la creazione di reti in cui i terminali dovevano monitorare degli specifici eventi esterni presenti nell'area circostante (come ad esempio alcuni agenti atmosferici); per questo motivo, infatti, si è fatto riferimento ad un particolare tipo di reti ad hoc chiamato “schiere di sensori”. Una tipica applicazione delle schiere di sensori, infatti, è la capacità di poter gestire in modo efficiente l'energia disponibile per ogni apparato, in un ambiente dove, di solito, non si trovano più di cento dispositivi.

Considerate queste caratteristiche, l'UWB radio diventa una buonissima soluzione per lo schieramento di reti di sensori e ad hoc networks [8].

Diverse caratteristiche dei sistemi UWB li rendono particolarmente adatti per reti di tipo wireless in ambienti indoor e outdoor:

- la capacità di trasmettere dati a velocità molto elevata, nell'ordine di centinaia di Mb/s o anche di alcuni Gb/s.
- l'alta risoluzione temporale, che conferisce all'intero sistema una buonissima robustezza in presenza di multipath, in modo da rendere possibili comunicazioni anche in presenza di diversi ostacoli ed in condizioni di propagazione del segnale NLOS (Non-Line-Of-Sight).
- l'accurata capacità di posizionamento, resa possibile anche dall'alta risoluzione temporale dei segnali UWB, offrendo informazioni sulle distanze che possono essere usate per derivare le reali coordinate fisiche del terminale, all'interno della rete.

Le caratteristiche precedentemente illustrate derivano dal fatto che il segnale UWB



## 21 - ANALISI DELLO SCENARIO DI LAVORO

usa una banda che si aggira tra 0 e 10 GHz; a causa di questa larga occupazione di banda, dunque, lo spettro di emissione dei segnali UWB copre una larga porzione dello spettro di frequenza, e può dunque coesistere con la propagazione delle altre forme d'onda Hertziane che interferiscono in etere.

Allo stesso momento, però, sono state necessarie anche delle limitazioni in potenza di trasmissione per non creare interferenza sui sistemi narrowband già esistenti prima, tradotte nell'emanazione nel 2002 da parte della FCC (Federal Communication Commission) di alcune maschere di emissione in USA (successivamente nel 2003 da parte dell'ERO (European Radio Organization), per quanto riguarda l'Europa).

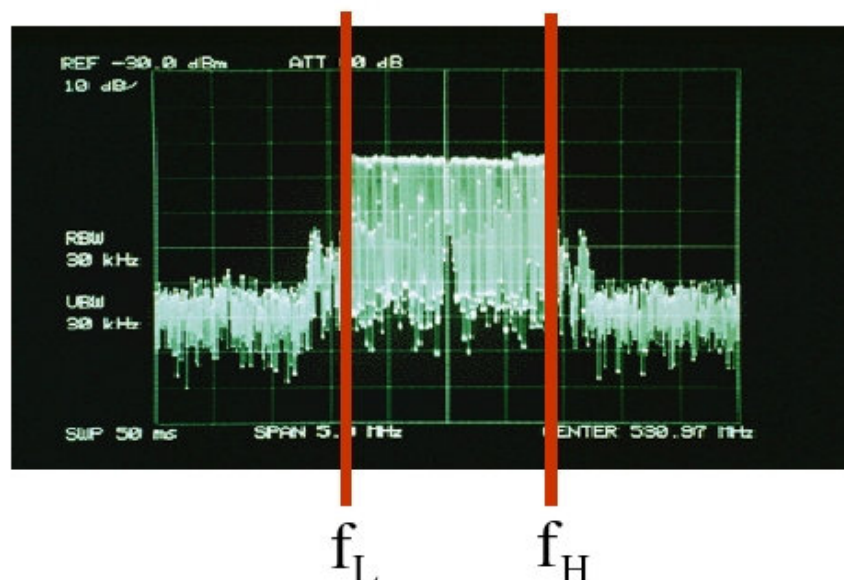
Grazie alle caratteristiche del segnale UWB l'IEEE ha creato un nuovo Working Group per la definizione di un nuovo standard, l'802.15.4a, per reti di sensori in scenari indoor e outdoor con la possibilità di avere accurate caratteristiche di posizionamento; questo standard è derivato dal già esistente 802.15.4, che ha formato la base per la creazione dello standard ZigBee.

Il nuovo standard ha inizialmente adottato la definizione del MAC fatta in 802.15.4, con la consapevolezza di dover creare successivamente degli accorgimenti per una gestione efficiente delle informazioni di posizionamento che derivano dalle innovazioni del nuovo strato fisico di 802.15.4a.

## 2.3 DEFINIZIONE E CARATTERISTICHE DELL'UWB RADIO

Il termine UWB deriva dalla sua trattazione nel mondo radar e si riferisce a qualsiasi forma d'onda che ha una banda frazionaria di energia istantanea maggiore del 20÷25%.

Se si considera  $E$  come l'energia istantanea della forma d'onda, l'energia frazionaria (come mostrato in Figura 2.2) è identificata dalle frequenze  $f_H$  ed  $f_L$  che delimitano l'intervallo dove la maggior parte dell'energia (per esempio intorno al 90%) decade: la banda di energia è definita proprio dall'intervallo  $f_H - f_L$  [9].



**Fig. 2.2: Banda Frazionaria di un segnale UWB [9].**

Posti come limite superiore  $f_H$  e limite inferiore  $f_L$  della ESD (Energy Spectral Density), il centro della frequenza dello spettro del segnale è dato da  $(f_H + f_L)/2$ ; la banda frazionaria (FB) è definita come il rapporto tra la banda di energia ed il centro della frequenza ed è espressa come:

$$FB = \frac{(f_H - f_L)}{\left(\frac{f_H + f_L}{2}\right)}$$

Se la banda frazionaria, dunque, è maggiore o uguale al 20÷25%, si può identificare il segnale come appartenente alla famiglia UWB.

Un segnale con una banda di energia di 2 MHz, per esempio, può essere considerato UWB soltanto se la frequenza centrale del suo spettro è minore di 10 MHz; la definizione di un segnale UWB, infatti, può essere data tenendo soltanto in considerazione la frequenza centrale del segnale stesso.

L'FCC, inoltre, usa un valore minimo di larghezza di banda pari a 500 MHz per definire segnali UWB in combinazione con una soglia di 2,5 GHz; i segnali al di sotto della soglia sono considerati UWB, infatti, soltanto se la loro banda frazionaria supera il 20%, mentre per quelli sopra la soglia la condizione per essere considerati UWB è quella di avere un'ampiezza di banda superiore ai 500 MHz.

Il modo più comune per emettere e trasmettere un segnale UWB, in sistemi di comunicazioni, è quello di emanare le forme d'onda che lo compongono attraverso una sequenza di impulsi molto piccoli, considerando la durata di ogni singolo impulso intorno ad alcune centinaia di picosecondi; per questo motivo questa tecnica di trasmissione prende il nome di IR (Impulse Radio).

Le maggiori tecniche di modulazione usate per questo tipo di segnali sono la PPM (Pulse Position Modulation) e la PAM (Pulse Amplitude Modulation) e saranno

trattate meglio nella parte successiva di questo paragrafo.

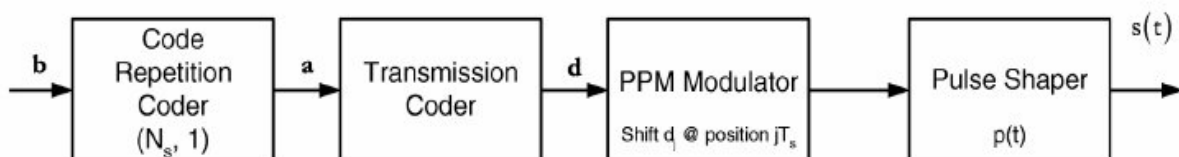
In aggiunta alla modulazione ed allo scopo di sagomare lo spettro del segnale generato, sono usati dei codici pseudorandom o pseudonoise (PN) per codificare i dati; nella maggior parte dei casi la codifica dei dati introduce una distorsione temporale generata sul segnale che introduce il così detto TH-UWB (Time-Hopping UWB).

Un altro possibile schema di codifica è il DS-SS (Direct-Sequence Spread Spectrum), dove i dati codificati modulano l'ampiezza degli impulsi base; nel caso dell'IR-UWB questa tecnica viene chiamata DS-UWB.

Per descrivere in modo generale come sono generati i segnali UWB, è possibile considerare la definizione più comune di un segnale appartenente a questa famiglia, ovvero la combinazione di un IR-UWB con un TH-PPM; il segnale UWB trasmesso  $s(t)$ , dunque, può essere espresso come segue ([9]):

$$s(t) = \sum_{j=-\infty}^{+\infty} p(t - jT_s - c_j T_c - a_j \epsilon)$$

Il segnale  $s(t)$  si può generare partendo dalla sequenza binaria che deve essere trasferita, usando un codificatore di ripetizione che associa ad ogni digit binario,  $N_s$  bits con lo stesso valore, generati ad un rate di  $1/T_s$  bit/s; il  $j$ -esimo bit codificato è indicato con  $a_j$ , mentre la durata del bit,  $T_b$ , è calcolata considerando il tempo richiesto per trasmettere gli  $N_s$  bit codificati:  $T_b = N_s T_s$  (come mostrato in Figura 2.3).



**Fig. 2.3: Schema a blocchi di un trasmettitore UWB con modulazione PPM [10].**

Successivamente si applica un codice stimato  $\mathbf{c}=(\dots, c_0, c_1, \dots, c_j, c_{j+1}, \dots)$  ai bit

codificati: usualmente si assume  $\mathbf{c}$  pseudorandom ed ogni elemento  $c_j$  del codice soddisfa la condizione  $0 \leq c_j \leq N_h - 1$ .

Il generico elemento di  $\mathbf{c}$  è usato, inoltre, per introdurre un ritardo random nella trasmissione di ogni impulso e per questo motivo assume il nome di codice TH (Time Hopping).

Il ritardo introdotto dal  $j$ -esimo valore del codice TH è dato da  $c_j T_c$ , con  $T_c$  pari al tempo di un chip.

La sequenza entra, poi, in un modulatore PPM (o PAM, a seconda dei casi) che genera una sequenza di impulsi di Dirac  $\delta(t)$  al rate di  $1/T_s$  impulsi/s; questi impulsi sono traslati nel tempo, successivamente, (Time Hopping) da una posizione nominale  $jT_s$  di un valore  $c_j T_c + a_j \varepsilon$ , dove  $\varepsilon$  verifica la condizione  $c_j T_c + \varepsilon < T_s$  per tutti i  $c_j$ . La traslazione introdotta dalla modulazione PPM,  $a_j \varepsilon$ , dipende dal valore del bit codificato  $a_j$  ([10]).

Infine la sequenza di impulsi ritardati entra in un filtro sagomatore di impulsi a risposta impulsiva  $p(t)$  che restituisce, in uscita, una sequenza di impulsi molto ravvicinati, ma non sovrapposti; la scelta dell'impulso è cruciale in un sistema IR, in quanto l'ESD (Energy Spectral Density) dell'impulso determina la copertura della PSD (Power Spectral Density) del corrispondente segnale UWB.

Nel caso in cui, invece, si vuole attuare una modulazione PAM al segnale in trasmissione (come mostrato in figura 2.4), prima di codificare il segnale con il codice  $\mathbf{c}$ , presa la sequenza binaria  $\mathbf{b}$  generata ad un rate  $R_b = 1/T_b$ , un primo sistema ripete ogni bit  $N_s$  volte e genera una sequenza binaria  $\mathbf{a}^*$  ad un rate  $R_{cb} = N_s/T_b = 1/T_s$  bits/s; un secondo sistema, successivamente, trasforma questa sequenza in  $\mathbf{a}$ , composta da simboli antipodali ( $\pm 1$ ) ([11]).

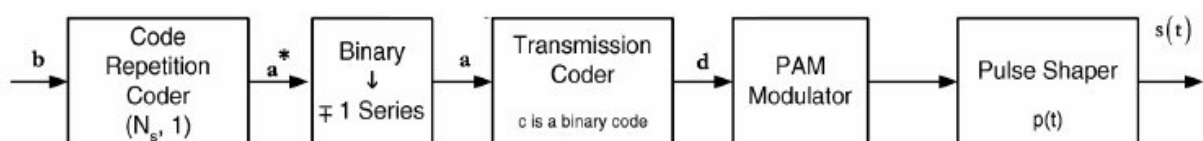


Fig. 2.4: Schema a blocchi di un trasmettitore UWB con modulazione PAM [11].

Il codificatore di trasmissione applica un codice binario  $\mathbf{c}$  composto da  $\pm 1$  e periodo  $N_p$  (comunemente assunto come un multiplo di  $N_s$ ), generando una nuova sequenza  $\mathbf{d}=\mathbf{a}\bullet\mathbf{c}$  composta da elementi  $d_j=a_jc_j$ .

La sequenza  $\mathbf{d}$  entra in un terzo sistema (modulatore PAM) che genera una sequenza di impulsi di Dirac  $\delta(t)$  ad un rate  $R_p=N_s/T_b=1/T_s$  e posizionati negli istanti  $jT_s$ . Infine il segnale entra nell'ultimo sistema, ovvero un filtro sagomatore di impulsi a risposta impulsiva  $p(t)$ , ottenendo un un segnale d'uscita composto da impulsi molto ravvicinati tra loro e non sovrapposti; il segnale  $s(t)$  ottenuto si può dunque scrivere come:

$$s(t) = \sum_{j=-\infty}^{+\infty} d_j p(t-jT_s)$$

L'efficienza di entrambe le modulazioni (sia PPM che PAM) si può considerare in termini di distanza inter-simbolo come una funzione dell'energia per bit. Dalle costellazioni dei simboli usate per queste due tecniche di modulazione, si può notare come la PAM binaria porti ad una maggiore distanza inter-simbolo rispetto alla PPM, avendo una maggiore efficienza in energia (valore pari a 3dB): la PPM, infatti, richiede il doppio dell'energia rispetto ad una PAM per ottenere lo stesso BER (Bit Error Rate).

Di contro, però, la PPM richiede soltanto una singola polarità per gli impulsi generati, richiedendo una struttura del trasmettitore più semplice (come facilmente osservabile dalle figure 2.3 e 2.4) rispetto alla PAM.

Nel caso in cui si considerano  $N_u$  trasmettitori attivi nello stesso istante il segnale ricevuto (nel ricevitore prescelto) sarà dato dalla combinazione del corrispondente segnale  $s_{rec}(t)$  trasmesso:

$$s_{rec}(t) = \sum_{k=1}^{N_u} \sum_{j=-\infty}^{\infty} p^{(k)}(t-jT_s - c_j^{(k)}T_c - a_j^{(k)}\varepsilon - \delta^{(k)}) + n(t)$$

Nell'equazione sovrastante i termini  $p^{(k)}$  e  $\delta^{(k)}$  sono inseriti per considerare l'effetto della propagazione nel canale di ogni segnale e l'asincronismo esistente tra i diversi utenti; tuttora il problema della modellazione della Multi User Interference nei sistemi UWB è ancora aperto e soggetto a nuove evoluzioni, in quanto la soluzione adottata tradizionalmente (modello Gaussiano) è lontana dai casi che si verificano realmente in questo tipo di dispositivi.

## 2.4 LA CLUSTERIZZAZIONE

Il clustering (analisi dei cluster o analisi di raggruppamento) è una tecnica di analisi multivariata di informazioni volta alla selezione ed al raggruppamento di elementi omogenei in un insieme di dati; è un processo con il quale si tenta di determinare le relazioni tra suddetti oggetti ed organizzarli in raggruppamenti (cluster) in modo che gli oggetti in un cluster siano tra loro il più simile possibile rispetto a quelli assegnati ad un altro cluster. Per clustering si intende, quindi, una tecnica con la quale si creano raggruppamenti di oggetti simili, che presentano fra loro caratteristiche omogenee; quindi, se un oggetto in un cluster possiede determinate caratteristiche, è molto probabile che anche gli altri oggetti nel cluster presentino le stesse proprietà.

Definendo un cluster in modo più generico, si può affermare che sia dato da una collezione di oggetti “simili” tra loro che sono “dissimili” rispetto agli oggetti degli altri cluster: dati un insieme di esempi, dunque, (descritti da un insieme di attributi) e una misura di similarità, operare un processo di clusterizzazione significa trovare un insieme di cluster tali che:

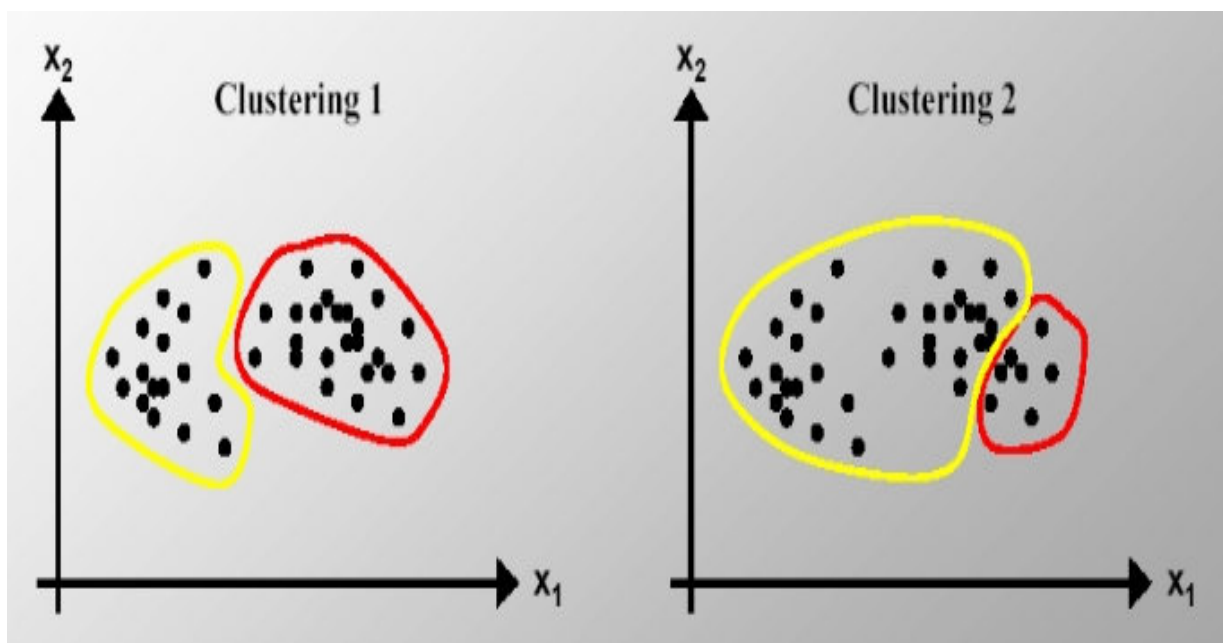
- gli esempi appartenenti allo stesso cluster risultino simili;
- gli esempi appartenenti a cluster differenti risultino dissimili.

La misura di similarità è il cuore del problema e, come sarà spiegato più avanti, si baserà soprattutto sul calcolo della distanza Euclidea tra i due oggetti ([12]).

In un sistema di telecomunicazioni nel momento in cui si ha un numero elevato di componenti della network, è possibile adottare la tecnica precedentemente citata, in modo da ottenere migliori prestazioni in base alle richieste del cliente che usufruirà in futuro del servizio.

Tutte le tecniche di clustering si basano sul concetto di distanza tra due elementi; infatti, la bontà delle analisi ottenute da suddetti algoritmi dipende essenzialmente da quanto è significativa la metrica e quindi da come è stata definita la distanza (concetto fondamentale in quanto l'appartenenza o meno di un nodo ad un insieme dipende da quanto l'elemento preso in esame dista dall'insieme stesso).

In Figura 2.5 è possibile osservare come, partendo da un numero finito di nodi posizionati su un piano bidimensionale  $(x_1, x_2)$ , ci siano due (o comunque molteplici) possibili decisioni nella clusterizzazione della rete, di cui la prima risulta visivamente più vantaggiosa a livello di equidistribuzione e distanza tra i componenti del Network.



**Fig. 2.5: Due possibili tipi di clustering per un insieme di nodi [13].**

Le tecniche di clustering si possono basare principalmente su due filosofie:



## 29 - ANALISI DELLO SCENARIO DI LAVORO

- a) Dal basso verso l'alto: inizialmente tutti gli elementi sono considerati cluster a sé e successivamente l'algoritmo provvede ad unire quelli più vicini, fino ad ottenere un numero predefinito di cluster o fino a che non si raggiunga una distanza minima prefissata tra essi.
- b) Dall'alto verso il basso: si parte da un unico cluster e successivamente l'algoritmo crea elementi di dimensioni sempre minori, omogenei, fino a che non si raggiunge un numero prefissato (approccio gerarchico).

Una volta definito un adeguato criterio di clustering, si pone il problema della strategia da adottare per raggiungere una partizione che ottimizzi il criterio scelto: in altri termini, la clusterizzazione non è altro che un metodo di ottimizzazione dello spazio discreto di tutte le possibili partizioni; in una ricerca esaustiva della soluzione ottima, partizionare  $P$  nodi in  $C$  clusters equivale a considerare un numero  $S$  di soluzioni pari approssimativamente a:

$$S = C^P / C!$$

E' evidente che il clustering basato su una tecnica esaustiva non rappresenta una soluzione pratica; per questo la soluzione più usata è quella basata su una ricerca iterativa di  $S$  partendo da un clustering iniziale e, ad ogni iterazione, trasferendo un sottoinsieme di nodi da un cluster ad un altro, in modo da migliorare il valore del criterio scelto (come mostrato in Figura 2.6, con la scelta di dividere lo stesso scenario, prima in due e poi in tre cluster diversi).

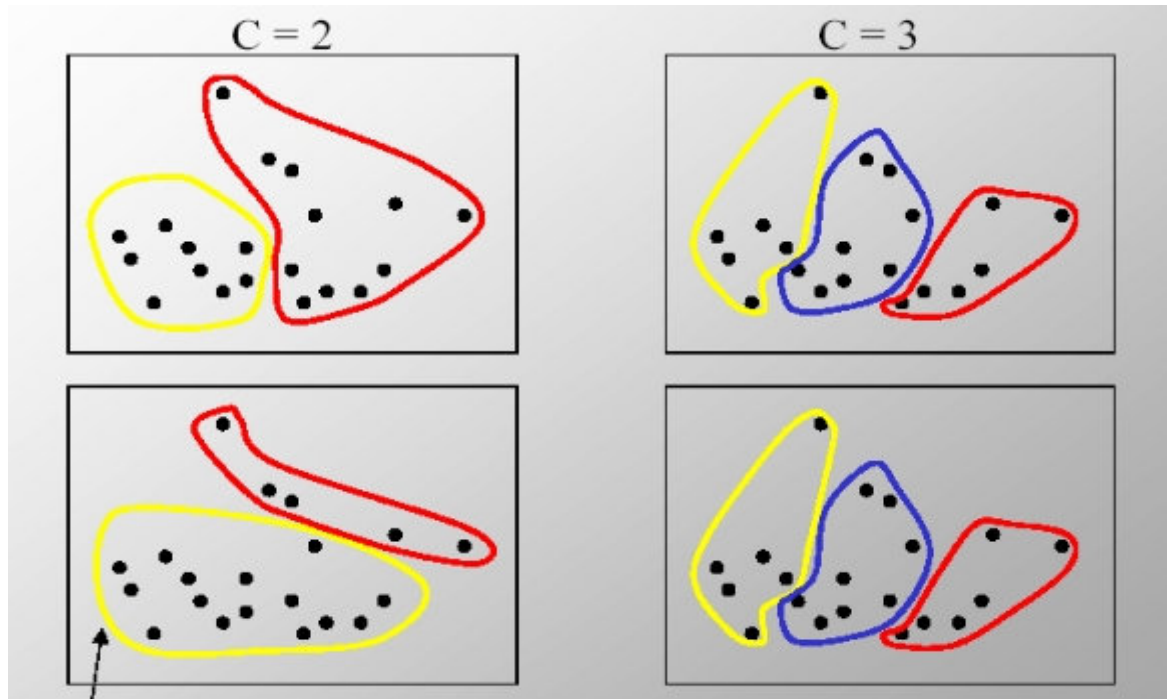


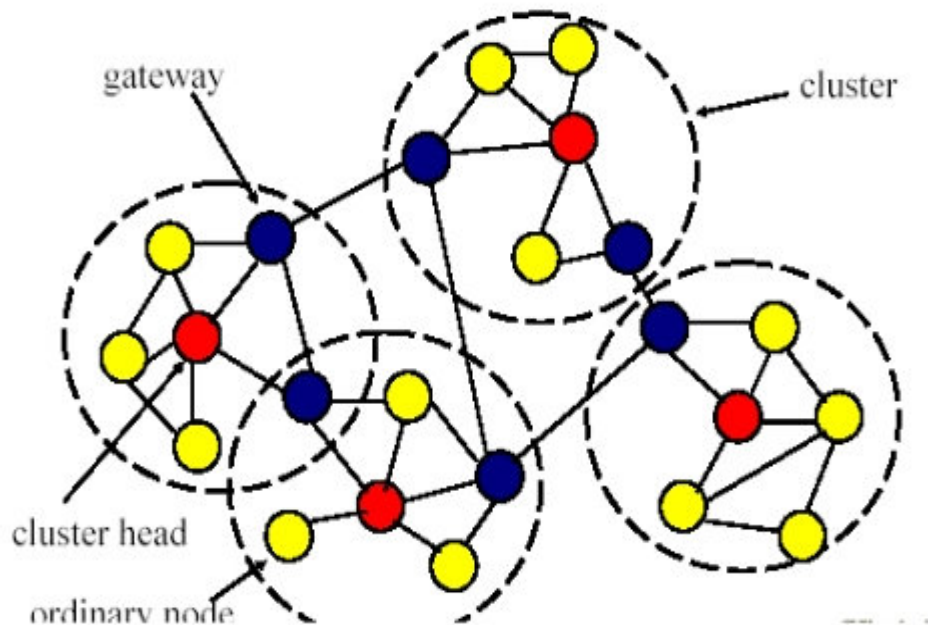
Fig. 2.6: Diversi tipi di clusterizzazione su uno stesso scenario [13].

Tuttavia quest'approccio non garantisce il raggiungimento della soluzione ottima in assoluto e può rivelarsi sensibile alla scelta di clustering iniziale: nonostante tale approccio sia sub-ottimo, però, il suo onere computazionale relativamente basso lo rende interessante.

L'operazione di "clustering", in altre parole, consiste nel trasformare la rete fisica in una rete virtuale di gruppi (cluster) di nodi interconnessi (come si può vedere in Figura 2.7).

All'interno di un singolo cluster si delineano tre distinte entità gerarchiche:

- *nodo di controllo, cluster controller o cluster head (CH)*: agisce per conto degli altri membri del cluster per prendere decisioni di controllo;
- *nodo gateway*: stabilisce la comunicazione tra cluster differenti.
- *nodo ordinario*: terminale ordinario, con caratteristiche standard dei nodi di rete ad hoc.



**Fig. 2.7: Funzionalità dei nodi che compongono un cluster [7].**

L'obiettivo (a livello fisico) dell'operazione di Clustering è quello di migliorare l'efficienza dell'utilizzo delle risorse al fine di:

- ridurre i conflitti sul canale;
- formare delle dorsali di instradamento per ridurre le dimensioni delle reti;
- astrarre l'informazione sullo stato della rete per ridurre la variabilità.

Un'architettura di rete che si appoggi al clustering deve necessariamente implementare protocolli per stabilire quali nodi debbano operare da controllori e quali da gateway; al tempo stesso, un'infrastruttura di questo tipo fornisce una dorsale virtuale costituita da nodi di controllo e gateway, che può introdurre una riduzione del throughput ed una maggiore sensibilità ai guasti ([7]). La robustezza della rete è infatti ridotta dalla possibilità di guasto sul singolo nodo (controllore o gateway).

Generalmente le tecniche di Clustering sono utilizzate quando si hanno tanti dati eterogenei e si è alla ricerca di elementi anomali; i vantaggi nell'uso di questa tecnica nell'organizzazione di una rete ad hoc multihop sono principalmente tre:

- 1) In ambienti multihop una struttura a cluster facilita il riuso spaziale delle risorse per incrementare la capacità del sistema: in una struttura a multicluster non sovrapponibili, infatti, due cluster possono usare la stessa frequenza o lo stesso codice se non appartengono a gruppi vicini tra loro.
- 2) Si facilita l'aggiornamento della topologia gerarchica della rete: quando un nodo mobile cambia la sua posizione, infatti, è sufficiente solo ai nodi che appartengono al suo stesso cluster aggiornare le loro informazioni topologiche, e non anche a tutti gli altri componenti del sistema.
- 3) Si ha una riduzione della generazione e propagazione delle informazioni inerenti alla procedura di routing: i clusterheads (CH, coordinatori di tutti i cluster che compongono il sistema), infatti, possono costituire lo strato più alto per il supporto del multicasting, dei flussi di dati ed anche della tolleranza agli errori di istradamento.

## **CAPITOLO 3**

# **ANALISI DI ALCUNI ALGORITMI DI CLUSTERING**

Scopo principale di questo capitolo è quello di illustrare diverse tipologie di algoritmi per la divisione della rete in clusters, considerando sei principali famiglie in base alle finalità per cui sono stati creati.

### 33 - ANALISI DI ALCUNI ALGORITMI DI CLUSTERING

Tutti gli algoritmi trattati sono di tipo distribuito, perché meglio applicabili alle WPAN prese in considerazione.

Nella prima famiglia sarà presentato un algoritmo per il controllo di anomalie dovute ad intrusioni di altri dispositivi nella MANET considerata; la seconda famiglia è composta da tre algoritmi che hanno come scopo principale quello di fornire una maggiore stabilità, attraverso la clusterizzazione, ai layers superiori; la terza famiglia raggruppa tre algoritmi che sono stati creati per ottimizzare i processi di routing, mantenimento di una determinata QoS e gestione della sicurezza; alla quarta famiglia appartiene un algoritmo che cerca di ottimizzare la densità dei nodi che vanno a formare i clusters della rete, per ottimizzare le prestazioni; nella quinta famiglia è illustrato un algoritmo che mira a diminuire il più possibile l'overhead scambiato tra i vari nodi per l'elezione del cluster head, quando questo cambia troppo repentinamente; nella sesta famiglia, invece, gli algoritmi illustrati cercano di raggiungere una migliore efficienza dal punto di vista energetico.

## **3.1 ALGORITMO PER IL CONTROLLO DI ANOMALIE DOVUTE AD INTRUSIONI**

Attraverso questo algoritmo si affronta il problema di dover difendere una MANET contro attacchi esterni (come spiegato in [14]).

Gli autori assumono come obiettivo principale la difesa della rete da attacchi esterni, attraverso un sistema gerarchico a tre livelli, in cui i dati vengono, in ordine, raccolti, processati e trasmessi.

Il sistema su cui si basa l'algoritmo è una rete di sensori co-locata, in cui si monitora assiduamente lo stato di salute di tutto il network; in pratica si fornisce ogni nodo della MANET di un IDS (Intrusion Detection System), ovvero un indice che si ottiene dopo aver raccolto puramente i dati che si hanno dalle varie elaborazioni di rete e vi si è associato un indicativo di anomalia, calcolato dalla differenza tra i valori del set di parametri caratterizzanti il nodo allo stato corrente e quelli che ci si aspetta di avere in uscita dalla rete per il suo normale funzionamento.

Nel momento in cui questo meccanismo viene applicato ad una rete formata da più clusters, si ottiene che gli indici di anomalia devono essere periodicamente trasmessi

ad un cluster head, il quale fonde tutti i dati ottenuti in una sorta di tabella di indici di anomalie.

Di conseguenza i vari cluster heads di tutta la rete trasmettono periodicamente queste tabelle ad un nodo manager, che a sua volta fonde tutti questi dati in una tabella di livello superiore che racchiude tutti gli indici di anomalia della rete, considerando anche che, a causa della mobilità del network, l'insieme di tutti i clusters e di tutti i cluster heads sono tempo varianti.

Dallo studio della tabella finale, si può vedere come i nodi del network operano e dove bisogna intervenire per risolvere eventuali anomalie; si può osservare, in base allo studio dei vari parametri della tabella, ad esempio, a che bit rate, a che distanza, a che livello di potenza è più facile attaccare o meno il sistema, prendendo di conseguenza le opportune precauzioni.

Nell'applicazione dell'algoritmo precedentemente illustrato, inoltre, è importante considerare che tutte le implementazioni ed i test effettuati sono stati fatti sotto particolari condizioni di routing (l'Ad-Hoc On Demand Distance Vector Routing, AODV, ed Optimized Link State Mobility, OLSR), mobilità dei cammini (Random Walk Mobility Model e Reference Point Group Mobility a diverse velocità) e tipologia degli attacchi (Traffic flooding Denial-of-Service e Black Hole); inoltre per una migliore valutazione delle prestazioni si determina il ROC (Receiver Operating Characteristics), per tener conto delle varie condizioni operazionali ai nodi, ai cluster heads ed ai nodi manager.

In conclusione del lavoro, infine, tutti i risultati ottenuti (e graficati in funzione delle curve ROC), confermano l'efficacia dell'algoritmo nel cercare di trovare la soluzione migliore, una volta classificate ed analizzate tutte le anomalie.

L'accuratezza nel trovare e combattere le anomalie generalmente migliora muovendosi verso l'alto nella gerarchia della rete ed al livello dei clusters è più alta rispetto al livello locale, ottenendo dei risultati migliori per il complessivo funzionamento del network; inoltre le prestazioni aumentano allungando l'intervallo di campionamento ed adottando il routing AODV.

## **3.2 ALGORITMI PER FORNIRE MAGGIORE STABILITA' AI LAYER SUPERIORI**

Gli algoritmi facenti parte di questa famiglia cercano di operare un clustering che può



dare come risultato finale una struttura generica e stabile per i layers superiori: questo fine può essere raggiunto, nella prima tipologia presa in considerazione ([15]), andando a selezionare (in connettività spaziale e temporale) in modo ottimo il nodo da eleggere come cluster head (CH).

In questo algoritmo gli autori considerano principalmente la durata della connessione tra il CH ed i nodi vicini e la locazione all'interno del cluster di specifici nodi mobili. A differenza di altri algoritmi usati in diverse MANET, che studiano il comportamento della rete attraverso il routing e la gestione delle risorse, in questo caso si cerca di raggiungere una determinata stabilità attraverso la clusterizzazione, e l'operazione di clustering e di selezione del CH son fatte dal punto di vista della connettività tra nodi, andando a trovare la soluzione spazio-temporale migliore grazie ad un algoritmo genetico (Genetic Algorithm o GA): l'obiettivo principale è quello di preservare la struttura della rete il più possibile nel momento in cui i nodi si muovono e la sua topologia cambia lentamente.

Una rete clusterizzata può essere identificata da un grafo  $G=(V, E)$ , dove  $V$  indica il set dei nodi presenti ed  $E$  il set dei collegamenti wireless; ogni nodo è contraddistinto da un unico ID perché appartiene ad un unico cluster ed i vari clusters non sono sovrapponibili tra loro (questa notazione può essere assunta per tutti e tre gli algoritmi illustrati in questa sezione e non verrà ripetuta successivamente).

Nello studio di questo primo algoritmo è fondamentale considerare la netta suddivisione tra fattori vitali e fattori banali nel clusterizzare l'intera rete; da una parte, l'analisi precisa dei nodi mobili deve includere la direzione, la velocità e l'accelerazione del movimento (ma questi parametri sono complicati e quindi non vitali per l'analisi finale); dall'altra, dal punto di vista della riduzione della potenza di trasmissione (inteso come parametro fondamentale nella creazione dell'algoritmo), è importante, dopo aver visto se i nodi vicini possono o meno comunicare tra loro, che il CH sia quanto più vicino possibile al centro del cluster, in modo da ridurre il più possibile il raggio di trasmissione dati.

Inizialmente si studia la durata di connessione tra i nodi vicini, considerando che:

### 38 - ANALISI DI ALCUNI ALGORITMI DI CLUSTERING

- 1) Si calcola la durata di connessione tra due nodi dopo l' $n$ -esima riconnessione (dato che la loro associazione e deassociazione è resa necessaria dalla loro mobilità) e si considera che essa è nulla se i nodi non comunicano direttamente.
- 2) Una volta misurato questo tempo di connessione, si ricalcola attraverso l'algoritmo di clustering che ne da una nuova stima.
- 3) Il tempo totale di connessione di un nodo  $i$  (che indica la durata di connessione con tutti i nodi della rete) fornisce un'indicazione sulla stabilità del nodo stesso e viene considerato anche per la selezione del CH; questo tempo è ottenuto dividendo per  $m$  (numero totale di nodi del network) la sommatoria di tutte le durate dei tempi di connessione tra i vari nodi della rete stessa.

Successivamente si cerca di trovare il centro di ogni singolo cluster e le sue coordinate, assegnando ad ogni nodo un peso  $w_i$  che ne rappresenta la stima e la capacità di potenza.

Una volta ottenuti i  $k$  centri dei cluster, si modella il problema della selezione del CH come un problema multi object (perché si considerano simultaneamente durata di connessione e locazione del CH, per ottenere la soluzione spazio-temporale migliore), che da maggiori risultati per avere migliori prestazioni: questa, in breve, è la descrizione del GA trattato in [15].

Dalle simulazioni finali, misurando la stabilità come la determinazione del numero di nodi mobili capaci di diventare o meno CH, è possibile osservare le migliori prestazioni rispetto ad altri tipi di algoritmi (Figura 3.1).

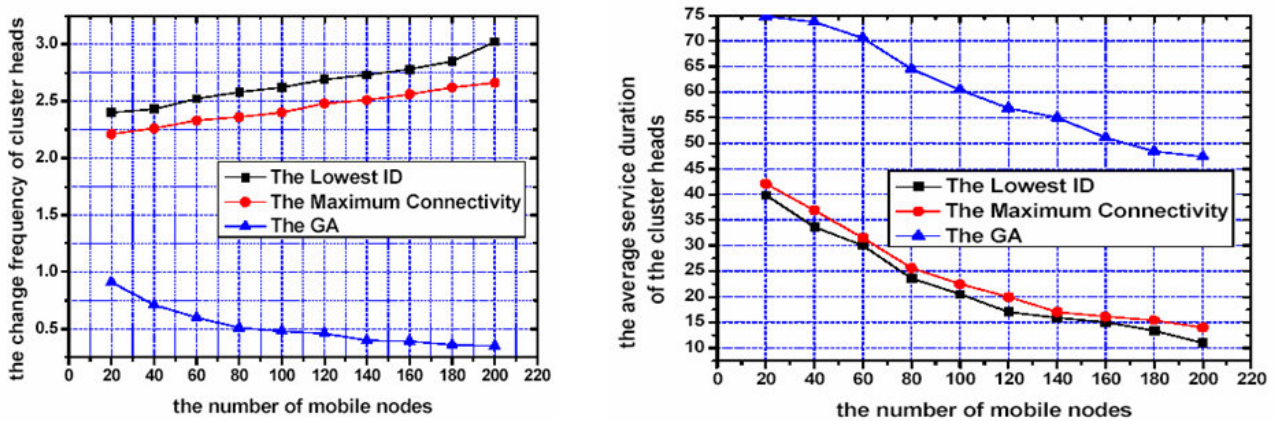


Fig. 3.1: Migliori prestazioni del GA rispetto ad altre due tipologie di algoritmi [15].

Si può notare come al crescere del numero di nodi del sistema, da un lato il GA garantisce una minore frequenza di cambio del CH, dall'altro fornisce un tempo di servizio del CH maggiore rispetto ad altri algoritmi, anche se decrescente: per queste caratteristiche il GA rende più stabile tutta la rete.

Il secondo algoritmo facente parte di questa famiglia definisce un nuovo Access-Based Clustering Protocol (ABCP) la cui formazione dei clusters è pesantemente influenzata dall'esito dell'accesso multiplo ([16]).

Mettendo in conto svariati aspetti delicati nell'operazione di clusterizzazione, l'ABCP fornisce un'architettura generica, flessibile, rapidamente schierabile e stabile per i protocolli degli strati superiori.

Dato che le reti ad hoc non hanno una struttura fissa e stabilita, è più opportuno operare una segnalazione separata, fuori banda, per il controllo delle informazioni trasmesse. Gli autori, dunque, in questo algoritmo, preferiscono trasmettere tutte le informazioni su due canali distinti, uno per il controllo e l'altro per lo scambio dei dati.

Come nell'algoritmo precedentemente illustrato ([15]) anche in questo caso la rete è rappresentata da un grafo  $G=(V, E)$ , in cui si considerano anche i diversi cluster heads  $C_i$  e la loro non sovrapposizione unita al fatto che ogni nodo appartiene ad un solo cluster.

In ABCP, dunque, con la considerazione di uno scheduling broadcast per i messaggi

#### 40 - ANALISI DI ALCUNI ALGORITMI DI CLUSTERING

di controllo, lo schema di accesso presentato minimizza l'overhead nella formazione dei clusters e quindi aumenta la stabilità di tutto il sistema complessivo.

Si assume che, considerando un ambiente multihop, se due nodi vogliono mandare simultaneamente due messaggi di controllo e distano più di due hop l'uno dall'altro, non si verificano collisioni e quindi non si deve aggiungere un OH al messaggio; inoltre, il riuso spaziale dei time slot del canale fornisce una maggiore efficienza a tutto il protocollo.

In pratica l'ABCP fornisce una semplice selezione di broadcast request-response con FCFS (First-Come-First-Service) che è unita ad uno schema probabilistico di risoluzione delle contese; è disegnato da un punto di vista protocollare in cui si definiscono i formats dei messaggi, si descrive come risponde un nodo quando arriva un messaggio e come si comporta in caso di errori o malfunzionamenti.

Un nodo che si appresta a diventare attivo assume la condizione di "nodo ordinario" e si va a settare in base al suo CH di appartenenza, attraverso lo scambio di appropriati messaggi; in base alle sue caratteristiche, però, anch'esso può concorrere all'elezione del CH e prenderne il posto. Attraverso il messaggio di controllo *SUCCESSOR*, è possibile sostituire anche un CH che diventa inattivo: il vecchio CH sceglierà il successore in base al maggior numero di link attivi che ogni nodo ha con i suoi vicini. Per evitare che il numero di clusters cresca con il tempo, inoltre, si applica una particolare "Funzione di Fusione", che unisce due CH che operano nello stesso raggio d'azione.

Dalle simulazioni effettuate in specifici ambienti, infine, si osserva come questa soluzione dia una maggiore stabilità in confronto ad altri protocolli, in base a diversi parametri come la velocità dei nodi, il raggio di trasmissione o il numero di salti da fare per comunicazioni interne ai clusters.

Si considera un ambiente di circa 1,5 Km<sup>2</sup>, composto da  $N$  nodi, in cui ogni componente alterna un periodo di inattività ad uno di attività (per avere un maggiore risparmio di batteria) e si muove a velocità costante uguale agli altri.

In Figura 3.2 si nota come l'algoritmo ABCP sia più performante e stabile rispetto ad

un algoritmo, chiamato CM (Cluster Maintenance), di tipo adattivo.

Da un lato al crescere del numero di one-hop tra nodi vicini, la frequenza di cambio del cluster di appartenenza è molto più bassa nell'ABCP rispetto al CM, tranne nel caso in cui  $55 < N < 80$ , dove l'ABCP si comporta addirittura peggio; dall'altro, al crescere della velocità dei nodi, il cambio di cluster, da parte dei nodi stessi, cresce in entrambi gli algoritmi, ma in modo proporzionalmente molto inferiore che nell'ABCP.

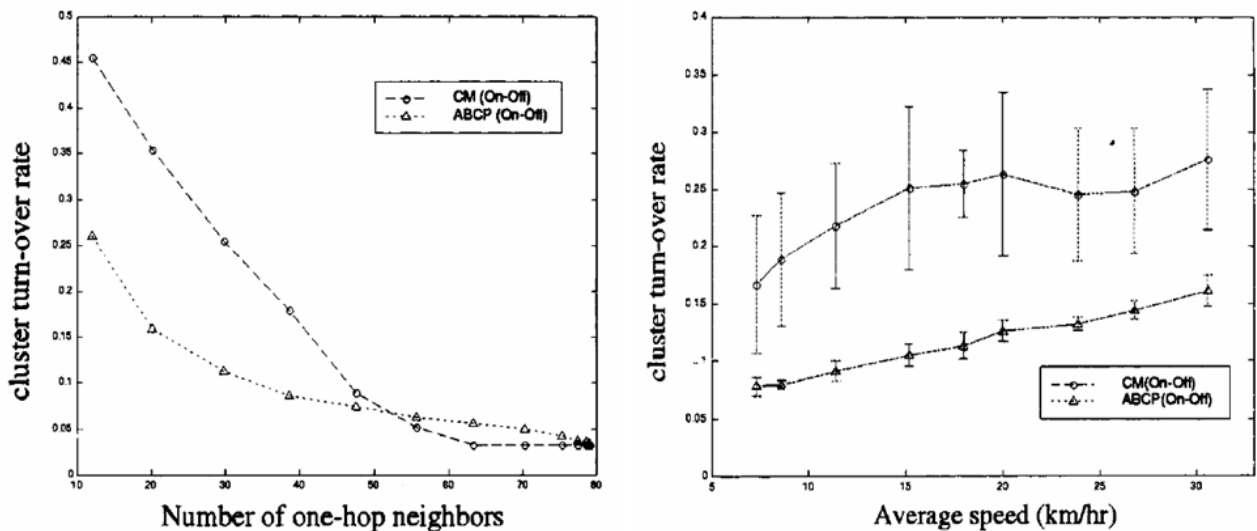


Fig. 3.2: Stabilità della struttura dei cluster in ABCP vs CM [16].

Per quanto concerne il terzo ed ultimo algoritmo appartenente a questa famiglia, si illustra una topologia di clustering chiamata MOBIC (come evidenziato in [17]), che sfrutta una nuova metrica di mobilità per le MANETs, basata sul rapporto del livello di potenza dovuto alle ricezioni successive su ogni nodo conseguente alle trasmissioni di quelli vicini; questa nuova metrica è usata per la selezione del cluster head e dimostra come porti una maggiore stabilità rispetto a soluzioni precedenti, come ad esempio la diminuzione fino al 33% del rate di scambio dati per l'elezione del CH.

In una MANET che usa servizi basati su clusterizzazione scalabile, inoltre, le prestazioni metriche della rete come ad esempio il throughput ed il ritardo di trasmissione sono strettamente accoppiate alla frequenza di riorganizzazione dei clusters, dovuta alla mobilità dei nodi che li compongono. In definitiva grazie all'uso di MOBIC si ottengono una configurazione complessiva del sistema più stabile e migliori prestazioni.

Per quanto riguarda la metrica usata, si implementa e si raffinano le capacità dell'algoritmo del Lowest-ID (per cui si elegge il CH con l'ID più basso all'interno del cluster) in quanto questo, da solo, non considera contemporaneamente anche la mobilità dei nodi; se un nodo con l'ID più basso ha grande mobilità, renderà necessario un re-clustering repentino e continuato quando si muove nel range d'azione di altri clusters.

Con MOBIC si avrà, dunque, la contemporanea considerazione della mobilità dei nodi rispetto anche a quelli vicini: infatti un nodo col più basso ID potrebbe anche non essere eletto CH se ha una mobilità troppo elevata rispetto a quella degli altri componenti del cluster stesso, poichè questa situazione fa aumentare la probabilità di rottura del cluster e rielezione del CH.

Gli autori, nello studio di questo algoritmo, assumono che la variazione del segnale di potenza in ricezione è un indicatore migliore della mobilità del nodo trasmettitore rispetto al puro studio della velocità o della distanza degli apparati che compongono l'ambiente considerato: maggiori sono le variazioni dell'ambiente circostante ai nodi, maggiori saranno le informazioni sulla mobilità dei terminali.

In MOBIC si assume che la mobilità è calcolata, attraverso lo studio della trasmissione (da parte del nodo  $X$  verso quello  $Y$ ) di due pacchetti successivi, nel nodo ricevente; si ottiene, dopo un intervallo di tempo, la stima di una determinata metrica di ricezione  $M_Y(X)$ . A causa della mobilità di tutti gli altri trasmettitori che sono nel raggio di azione di  $Y$ , si assume che i nodi che non trasmettono verso  $Y$  nell'intervallo di tempo che serve per il calcolo di  $M_Y(X)$  vengono automaticamente esclusi dal calcolo, considerando, dunque, solo quelli che si trovano nel raggio di

azione del ricevitore.

Questo calcolo è effettuato per ogni nodo ed ogni terminale manterrà una propria metrica, relativa a tutti i trasmettitori in azione.

Questa metrica, inoltre, non considera la disponibilità nei singoli nodi di dare informazioni sulla loro posizione (es. GPS) e sulla velocità che hanno nel muoversi.

Per calcolare, dunque, i parametri relativi alla possibile elezione del CH si considera anche che il segnale di potenza misurato ai nodi in ricezione in aggiunta ad un'indicazione sulla distanza tra trasmettitore e ricevitore e quindi una migliore "mappatura" di tutti i nodi che compongono i clusters.

Per le analisi sulle simulazioni, in conclusione, è opportuno osservare come in base alla diversa mobilità dei nodi si abbia, grazie a MOBIC, un decremento di cambio di CH ed una conseguente migliore stabilità dell'intero sistema (Figura 3.3).

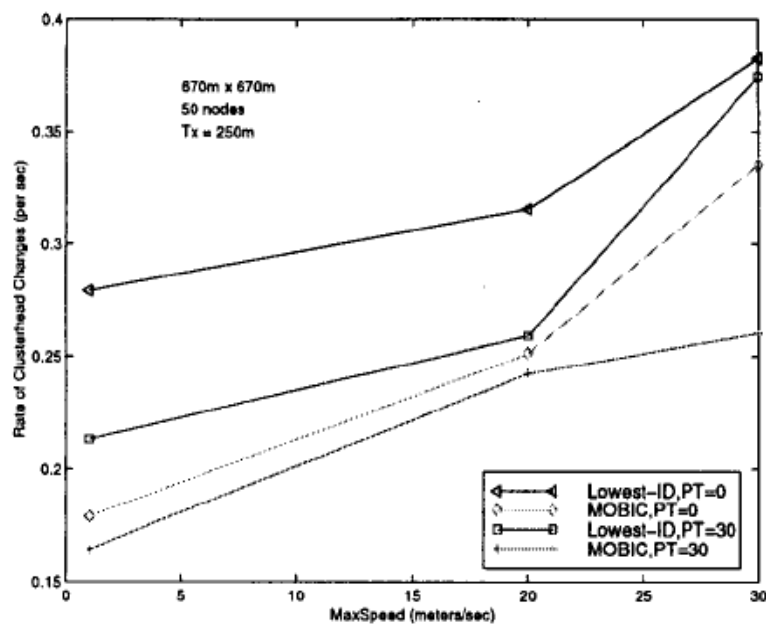


Fig. 3.3: Effetti della variazione della mobilità sugli algoritmi [17].

### **3.3 ALGORITMI PER FORNIRE MAGGIORE QoS, SICUREZZA ED UN ROUTING MIGLIORE**

Il primo dei tre algoritmi facenti parte di questa famiglia illustra come sia possibile creare una particolare struttura per ottenere un'organizzazione dinamica degli MN (Mobile Nodes), o nodi mobili, in MANETs a larga scala, mirando ad ottenere il supporto di una particolare QoS ([18]).

L'approccio di clustering illustrato è basato su una predizione intelligente della mobilità dei nodi che permette ad ogni MN di conoscere anticipatamente la disponibilità degli apparati vicini.

Questo tipo di soluzione è motivata dal fatto che la banda dei link interni alle MANETs e la potenza in trasmissione sono scarsi: quindi, si cerca di rispettare al meglio un set di parametri di QoS; per questo è necessario che ogni MN abbia informazioni sulla topologia di rete attraverso un overhead minimo nei dati di controllo di traffico.

Innovativa rispetto alle tecniche di clusterizzazione precedentemente viste è l'introduzione del Virtual Cluster: l'idea è che un'area geografica è divisa in regioni uguali di forma circolare e che ogni MN identifica in esse il suo raggio d'azione in modo da creare dei clusters virtuali anche sovrapponibili; se si trovano degli MNs realmente esistenti in questi virtual cluster, diventano reali.

Ognuno di questi ha un unico identificativo ed in questo modo si suppone che ogni singolo MN abbia una conoscenza totale della locazione dei VCC (questa informazione può essere, infatti, facilmente reperibile da ogni MN attraverso un semplice scambio di messaggi con i nodi adiacenti). In pratica si crea una



suddivisione della rete preventiva e ufficiosa, che può essere successivamente confermata in base alle caratteristiche degli apparati che la compongono.

In questo algoritmo si seleziona un MN come CH, in modo che abbia la maggior probabilità ( $p$ ), rispetto agli altri, di rimanere all'interno del virtual cluster, e che abbia la minima distanza ( $d$ ) dal centro del virtual cluster stesso (VCC); in questo modo non verrà eletto CH l'MN più mobile e l'area di copertura non cambierà a causa di continui cambi di CH.

Considerando un tempo specifico ( $t$ ) in cui si studiano i due parametri precedentemente illustrati, si ottiene proprio l'algoritmo di clustering illustrato ( $p, t, d$ ).

In conclusione si può affermare che attraverso questo algoritmo si hanno quattro benefici principali: una maggiore facilità nella gestione della mobilità, un possibile previsione della posizione di ogni MN in ordine con una specifica richiesta di deterioramento di QoS, il mancato cambiamento dell'identità dei cluster in base al cambio dei CH e infine la non frequente alternazione dei CH stessi.

Il secondo algoritmo appartenente a questa famiglia organizza le MANET in clusters nei quali la probabilità di disponibilità dei collegamenti può essere controllata (come illustrato in [19]).

Lo scopo di questa seconda soluzione (chiamata ( $\alpha, t$ ) cluster) è quello di minimizzare gli effetti dei cambi topologici della rete bilanciandoli con una richiesta di supporto di routing migliore.

Si considera un modello di mobilità per reti ad hoc e, attraverso questo, si deriva l'espressione per la probabilità di cammino disponibile in funzione del tempo.

Poiché il criterio per l'organizzazione in cluster dipende direttamente dalla stabilità dei collegamenti e la struttura della topologia dei cluster è adattiva nel rispetto della mobilità dei nodi, deve essere supportata un'architettura di routing adattivo ibrida che può essere comunque applicabile nel momento in cui il rate in mobilità dei nodi è maggiore o minore.

L'idea base di questo algoritmo è quella di creare una ripartizione dell'intera rete in

clusters di nodi che sono mutuamente raggiungibili attraverso collegamenti interni al cluster stesso, praticabili solo in un periodo di tempo  $t$  con una probabilità  $\alpha$ ; l'unione di tutti i clusters che compongono la rete coprirà, dunque, tutti i nodi presenti.

I clusters  $(\alpha, t)$  sono entità dinamiche create, espanse, contratte ed eventualmente terminate, basate su informazioni di routing che sono mantenute su un set di nodi cooperanti tra di loro; oltre ai messaggi divulgati riguardanti la topologia della rete attraverso i protocolli di routing, il protocollo di clustering non richiede altri messaggi addizionali; è necessario un aggiornamento solo nel caso in cui un cluster si inserisce o abbandona la rete.

I nodi inoltreranno le informazioni di routing solo ai dispositivi il cui identificativo li contraddistingue come facenti parte al cluster di interesse; non si richiede un controllo centralizzato sui processi di clusterizzazione e l'azione dell'algoritmo dipende solo dalla capacità dei nodi di soddisfare i criteri  $(\alpha, t)$  nel rispetto dell'equilibrio totale di tutto il network.

Dalle simulazioni effettuate si mette in evidenza come questo algoritmo supporti un routing adattivo ibrido che è efficiente sotto particolari condizioni di sistema (in base all'ambiente che si va a considerare) e può fornire soluzioni di routing migliori quando la mobilità dei nodi lo permette.

Rispetto ad altri algoritmi, infine, si mette in evidenza come al crescere del traffico, dato dallo scambio di pacchetti dati o di segnalazione tra i vari apparati della rete, si ottengano risposte migliori per una maggiore stabilità e funzionalità di tutto l'ambiente.

Il terzo ed ultimo algoritmo considerato è il CBRP (Cluster Based Routing Protocol) e tratta il problema della sicurezza per reti ad hoc; avere un protocollo di routing sicuro nelle MANET non è un problema semplice da risolvere.

Ci sono degli aspetti vulnerabili da considerare nel CBRP, il quale usa la formazione di singoli clusters con un unico cluster head; quest'ultimo ha la responsabilità di mantenere una lista dei membri che compongono i cluster e instradare correttamente i pacchetti dati che gli arrivano, verso i destinatari corretti. Il CH, infatti, è il vero e

proprio “motore” di tutti i clusters che compongono la rete e, grazie alla conoscenza dell’insieme gli apparati presenti nel suo raggio d’azione, permette di ottenere una migliore gestione del sistema.

Il problema di avere un singolo CH, però, è quello di avere un singolo punto di rottura che può paralizzare tutto il sistema, per cui in questo lavoro ([20]) si crea un algoritmo che distribuisce i servizi del CH originario ad un gruppo di CH chiamato Concilio.

Una volta che il Concilio basato sui clusters sarà formato, ogni suo membro può applicare una soglia  $(k,n)$  che indica un numero  $k$  minimo di CH su  $n$  possibili richieste di partecipare insieme per soddisfare le aspettative della rete.

Ottenendo dei CH multi-clusters, dunque, ognuno avrà una conoscenza “segreta” del resto del network da poter usare nel momento in cui si verificano dei problemi; per avere questa possibilità, però, i nodi devono essere sempre connessi con collegamenti bidirezionali verso tutti gli altri.

Alla fine di questa particolare organizzazione, quindi, si può vedere come si possa incrementare la tolleranza ai fallimenti di ogni tipo di sistema, dovuti ad attacchi contro la sicurezza.

I risultati sulle simulazioni effettuate dimostrano che nel momento in cui il Concilio è formato da clusters di grandi dimensioni, è possibile applicare il concetto di “segreto”  $(k,n)$  con risultati soddisfacenti dal punto di vista della stabilità della rete.

Questo schema, inoltre, è più adatto con una bassa mobilità delle reti unita ad una mancanza di overhead nel momento in cui si rende necessaria una riorganizzazione dei cluster.

## **3.4 ALGORITMO BASATO SULLA DENSITA' DEL CLUSTERING**

Questo algoritmo si fonda sull'organizzazione dei clusters in base alla densità dei nodi che li compongono; i due vantaggi principali che si ottengono sono:

- 1) la possibilità di poter allocare una forma arbitraria al cluster;
- 2) la possibilità di non dover richiedere, come dato in input, il numero dei clusters.

Nel momento in cui, analizzando la totale configurazione della rete, dei clusters toccano quelli confinanti, sia i clusters centrali che quelli di bordo (corrispondenti al picco ed alla parte più bassa del grafico della densità) diventano confusi e difficili da determinare (come illustrato in [21]).

Quando i clusters sono ben separati tra di loro, i metodi basati sulla densità lavorano bene in quanto le regioni di picco e di minimo della funzione di distribuzione sono ben definite e facili da scoprire: in questo modo le CIFs (Cluster Intensity Function) hanno un andamento molto simile alle funzioni di densità.

Quando i clusters sono troppo vicini tra loro, invece, (cosa che si verifica nelle situazioni reali) diventa molto più difficile determinare sia i clusters centrali che quelli di bordo, ma le CIFs riescono ancora a determinare chiaramente i clusters centrali, di bordo ed il livello di aggregazione di ogni data point verso il cluster a cui appartiene.

In definitiva questo algoritmo si basa su una struttura a “ricerca di collisione” tra i

vari clusters che compongono la rete con degli accorgimenti per risolvere i problemi presenti in altri algoritmi basati sulla densità dei nodi; i passi principali per lo sviluppo di questo metodo sono:

- Ottenere una funzione di densità di probabilità (PDF) attraverso la KDE (Kernel Density Estimation), che stima la densità interna del centro dei clusters e la estende a tutta la struttura.
- Identificare le regioni di picco della funzione di densità usando un'equazione di evoluzione della superficie, implementata dall'LMS (Level Set Methods).
- Costruire una funzione basata sulla distanza chiamata CIF (Cluster Intensità Function).
- Applicare un algoritmo per la ricerca delle zone meno dense sulla CIF.

Dalle simulazioni e l'applicazione dell'algoritmo su casi reali è evidenziato come si risolva il problema di densità irregolare dei clusters e come si riescano ad identificare facilmente anche le zone più isolate, rendendo più stabile e robusto tutto quanto il sistema.

### **3.5 ALGORITMO BASATO SULLA DIMINUZIONE DELL'OVERHEAD DEI PACCHETTI**

Nella maggior parte dei protocolli di clustering nelle MANET, i nodi CH hanno un ruolo speciale nella gestione ed elaborazione delle informazioni di routing; di

conseguenza, il frequente cambio di CH intacca le prestazioni del protocollo di instradamento ed eventualmente la stabilità nella MANET.

In questo algoritmo si cerca di trovare una soluzione per evitare un cambio troppo continuato dei CH che compongono la rete e rendere più stabile, quindi, tutto quanto il sistema ([22]).

Il collegamento tra due nodi cambia costantemente in proporzione al cambio dei set di CH del network; dal momento che il CH racchiude tutte le informazioni aggiornate di instradamento dei nodi del cluster, se si verifica un collasso del cluster head stesso, tutte queste informazioni vanno perse e si richiede l'elezione di un nuovo CH con un ulteriore scambio di messaggi (soprattutto di overhead), che appesantisce l'intasamento della rete.

L'obiettivo finale, dunque, è quello di cercare di allungare il più possibile il tempo di residenza medio nel cluster per ogni nodo; questo tempo è definito come il periodo in cui un nodo è associato ad un determinato cluster, non un CH.

Per ottenere una maggiore stabilità ed affidabilità si possono utilizzare due CH per ogni cluster; l'idea base è quella di eleggere un cluster head primario (PCH) ed uno secondario (SCH) per ogni cluster, in modo tale che entrambi possano accumulare tutti i messaggi di segnalazione ed i pacchetti che sono spediti dai nodi a valle della rete.

PCH ed SCH lavoreranno e invieranno pacchetti dati nello stesso momento, solo che l'SCH lo farà considerando un periodo limitato di immagazzinamento (che può essere considerato come un limite di sicurezza oltre il quale si avrà un funzionamento sicuro del PCH).

Basandosi sul suo livello di potenza della batteria, se il PCH non può operare a lungo, informerà il suo SCH correlato, che a sua volta cambierà stato e diventerà PCH.

Lo stesso processo si può ottenere se il PCH, a causa della sua mobilità, non può più svolgere le sue mansioni e la nuova elezione del CH contribuisce a diminuire la quantità di OH da dover scambiare tra i nodi della rete per riottenere un assetto stabile (come si può vedere in Figura 3.4).

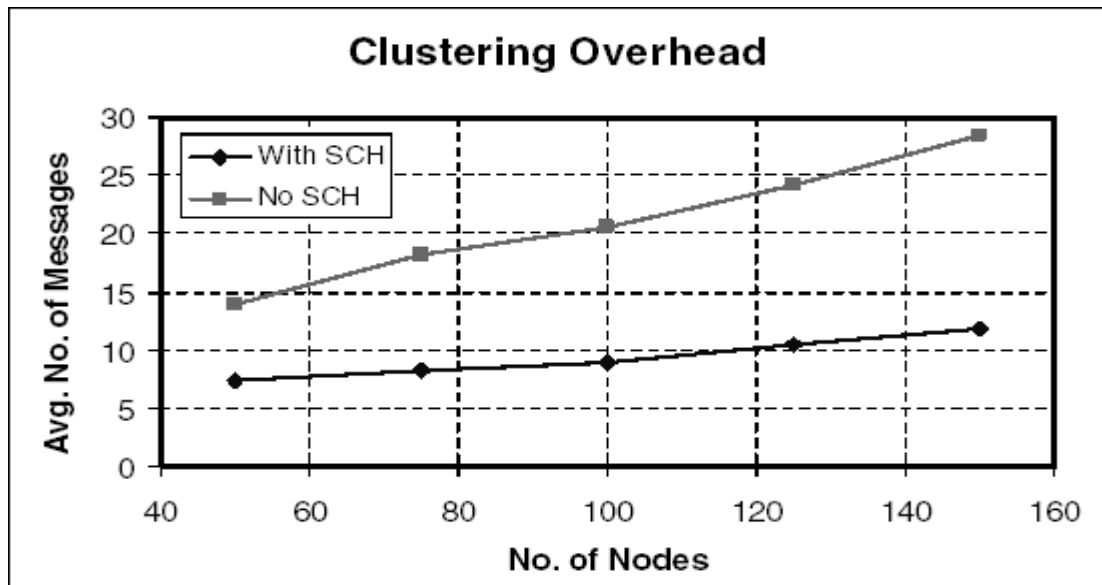


Fig. 3.4: Minor OH scambiato con l'uso dell'SCH [22].

In pratica l'SCH lavora come copia di back-up del PCH, dando all'intero cluster maggior tempo di vita e stabilità anche in termini di routing.

Per concludere, quindi, il trasferimento delle informazioni da un PCH ad un SCH mira ad incrementare il tempo di residenza nel cluster che sosterrà la stabilità della rete, decrementare l'OH di comunicazione del cluster, e minimizzare il tempo impiegato da ogni nodo a entrare o uscire da un cluster.

## **3.6 ALGORITMI BASATI SU UN USO EFFICIENTE DELL'ENERGIA**

Il Clustering può essere considerato una buona soluzione per prolungare la vita di un sistema (in particolare, in questo caso, di una rete di sensori).

Gli algoritmi di uso corrente utilizzano principalmente due tecniche, selezionando il CH, con maggiore energia residua e ruotando periodica la sua elezione, per poter così distribuire il consumo di energia totale su più nodi in ogni cluster (per estendere la vita totale del sistema), ma non si considera il problema dei “punti caldi” che si verifica principalmente nelle reti di sensori multihop.

Per esempio, se dei CHs cooperano con altri nodi per inoltrare pacchetti informativi verso una BS (Base Station), i CHs più vicini alla BS saranno dotati di un traffico da smaltire relativamente pesante e tenderanno a morire troppo presto, lasciando diverse aree della rete scoperte e causando una sua partizione forzata (come spiegato in [23]).

Per cercare di risolvere o quantomeno limitare questo problema, si è creato un algoritmo EEUC (Energy-Efficient Unequal Clustering) che riunisce dei nodi periodicamente e ne sfrutta il lavoro in cooperazione.

In pratica tutto il network è partizionato in un numero di cluster di diversa

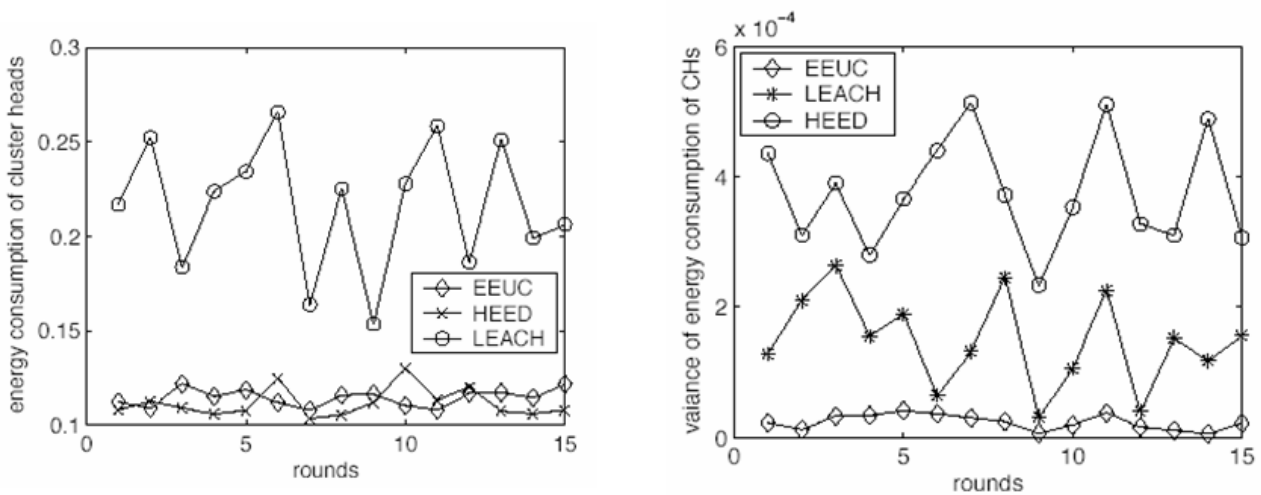


dimensione, con i clusters più vicini alla BS che hanno un'estensione maggiore rispetto a quelli più lontani, in modo da poter preservare più energia per le comunicazioni e l'inoltro dei dati intra-cluster.

Andando a considerare un sistema con particolari specifiche (sempre illustrate in [23]), l'EEUC è un algoritmo applicato che seleziona il CH principalmente in base al controllo dell'energia residua di ogni nodo; all'inizio si selezionano una serie di nodi che hanno la stessa probabilità  $T$  (soglia prestabilita) di diventare CH, con gli altri che si pongono fuori uso fino ad elezione completata; se si suppone che  $s_i$  sia un possibile CH con un raggio di competenza  $R_{comp}$  (funzione della sua distanza dalla BS), si può notare come se  $s_i$  diventa effettivamente CH, non ci può essere un altro  $s_j$  nel suo raggio di competenza che possa diventare CH: in questo modo si controlla la distribuzione dei CH in tutta la rete, allocando dimensioni maggiori (e quindi maggior energia) ai cluster più vicini alla BS per il carico superiore di lavoro che sopportano.

Controllando tutto il sistema, si possono creare più cluster vicino alla BS (dato che quelli lontani in genere hanno mansioni meno gravose) andando a trovare un ottimo compromesso tra energia spesa per l'inoltro dei pacchetti e lunghezza dei messaggi stessi: maggiori saranno i clusters vicini alla BS, minore sarà il peso dei pacchetti da inoltrare e la corrispondente energia spesa.

In Figura 3.5 è possibile notare come l'algoritmo EEUC abbia un dispendio di energia minore rispetto ad altre due tipologie, prima in valore assoluto e poi in termini di variazione.



**Fig. 3.5: Migliori prestazioni dell'EEUC rispetto ad altre due tipologie di Algoritmi [23].**

Un altro algoritmo appartenente a questa famiglia è il LEACH (Low-Energy Adaptive Clustering Hierarchy) visto, in questo caso, in una sua particolare implementazione in cui si fa in modo che ogni nodo possa aumentare le proprie risorse sfruttando l'energia solare (come mostrato in [24]).

Nel momento in cui una rete è creata per poter operare in ambienti ostili ed in aree non facilmente accessibili, la conservazione dell'energia gioca davvero un ruolo cruciale: mentre i sensori con una batteria autonoma sono prima o poi destinati ad esaurirsi, l'uso di apparecchi "rinnovabili energeticamente" attraverso fonti come la luce del sole, possono estendere la vita dell'intera rete in cui operano.

Nell'intera rete organizzata in clusters, i CH sono responsabili di svolgere mansioni che provocano un maggior dispendio di energia rispetto a tutti gli altri nodi; per questo motivo è molto attraente la possibilità di poter alimentare i CH attraverso una fonte esterna, anche se si hanno complicazioni dovute al fatto che il sole (fonte primaria considerata), ad esempio, non è permanente e quindi bisognerebbe creare un sistema per accumulare e conservare l'energia stessa.

In questa versione di LEACH la BS (Base Station), nel caso di algoritmo centralizzato, oppure ogni singolo nodo, nel caso distribuito, sceglie qual'è il CH (in base a determinati requisiti che saranno analizzati successivamente), che resterà tale per un certo lasso di tempo, chiamato "round".

In [24] gli autori, dunque, si soffermano su due diverse tipologie di LEACH: una distribuita ed una centralizzata: per le caratteristiche dello scenario che sarà analizzato successivamente, è più opportuno soffermarsi sulla prima tipologia, andando ad evidenziare quali sono i benefici che si possono ottenere.

La creazione dei clusters e la scelta dei CH è fatta in base a dei bilanci energetici, effettuati su ogni singolo apparato appartenente alla rete; nel momento in cui sono stati creati i clusters, ogni nodo  $n$  decide autonomamente se sarà un CH per il round successivo.

La selezione è fatta in modo stocastico: ogni nodo determina un numero random tra 0 e 1: se questo numero è più basso di una determinata soglia  $T(n)$ , allora il nodo stesso diventa CH.

$T_1(n)$  è determinato risolvendo l'equazione:

$$T_1(n) = \frac{P}{1 - P * (r \bmod \frac{1}{P})}$$

per i nodi che non sono stati eletti CH negli ultimi  $1/P$  rounds, altrimenti  $T_1(n)$  è pari a zero;  $P$  è un numero determinato a priori che indica il numero medio di CH durante un round mentre  $r$  è il numero del round corrente. Usando questo algoritmo, quindi, ogni nodo diventerà CH una volta all'interno di  $1/P$  rounds, mentre dopo  $1/P-1$  rounds,  $T_1(n)=1$  per tutti i nodi che non sono mai stati CH.

Nel momento in cui un nodo si elegge CH da solo, manda un messaggio in broadcast per avvertire tutti i nodi del suo nuovo ruolo; un nodo non-CH usa questi messaggi provenienti dal CH per scegliere il cluster a cui vuole appartenere, in base alla potenza del segnale ricevuto.

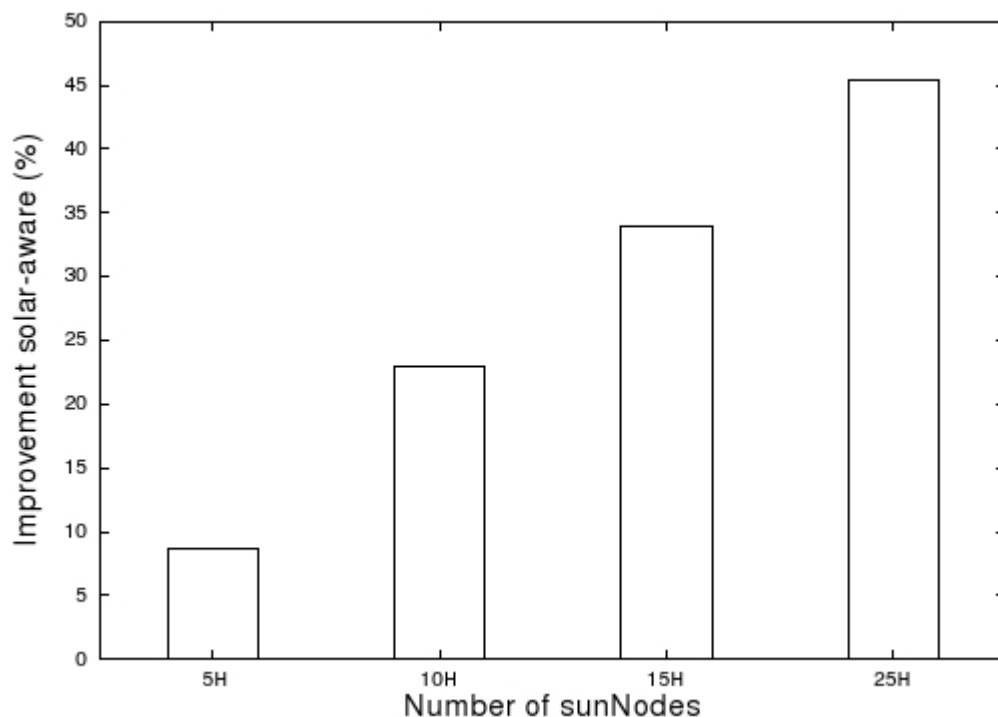
La differenza che si ottiene introducendo la possibilità di fornire una fonte di energia alternativa a tutti i nodi (la luce del sole) da due diverse conseguenze:

1. i nodi alimentati con il sole hanno una probabilità maggiore di diventare CH;
2. un nodo che riceve l'alimentazione solare dopo esser stato CH, può ridiventarlo nei successivi  $1/P$  rounds.

## 56 - ANALISI DI ALCUNI ALGORITMI DI CLUSTERING

Attraverso l'analisi di simulazioni e risultati, gli autori in [24] mostrano come la soluzione distribuita abbia delle prestazioni leggermente inferiori rispetto a quella centralizzata, ma è comunque evidenziato come si ottengano i maggiori miglioramenti attraverso una conoscenza globale di tutta quanta la rete.

In conclusione, dunque, in Figura 3.6 si può notare come, fissato il valore di  $P=0.05$  (con massimo 5 CH per round), la durata del sole a 1200, con 10 frames per round, all'aumentare del numero di nodi si ha un aumento percentuale di prestazioni globali del sistema, con una durata crescente (fino a 25h) di tutta la vita del network.



**Fig. 3.6: Migliori prestazioni del solar-aware LEACH [24].**

## **CAPITOLO 4**

# **L'AMBIENTE DI SIMULAZIONE**

In tale Capitolo saranno descritte le caratteristiche essenziali dell'ambiente di simulazione.

Tutti i risultati che saranno mostrati nel Capitolo successivo si sono ottenuti grazie ad OMNet++ , un simulatore orientato agli algoritmi scritto in C++ secondo il paradigma della programmazione basata sui componenti, che usa il linguaggio Ned per la configurazione della simulazione (una descrizione più accurata del funzionamento del simulatore è rimandata in Appendice A).

La differenza tra OMNet++ ed altri tipi di simulatori sta nel maggior livello di astrazione, nel minor numero di funzionalità offerte e nella maggiore semplicità della simulazione; in OMNet++ non esistono librerie di componenti che implementano ciascuno strato dello stack protocollare del software di rete; il simulatore è concepito per una rappresentazione molto più astratta dei nodi della rete, che nella maggior parte dei casi avviene tramite un unico componente o un insieme di pochi componenti interni, che si occupano unicamente di descrivere come il nodo in questione gestisca

la ritrasmissione dei messaggi (intesi come semplici strutture dati con poche informazioni basilari).

Il file di configurazione non consente di inserire eventi nella simulazione, di osservare lo stato interno degli elementi della rete o di svolgere altre operazioni al di fuori della dichiarazione o connessione di componenti.

Una grande peculiarità di questo simulatore, infine, è la capacità di generare files eseguibili, per ogni simulazione, che possono essere eseguiti su altre macchine anche senza aver installato OMNet++.

Nella parte successiva di questo Capitolo sarà analizzato più a fondo l'ambiente di simulazione e le componenti che ne renderanno possibile l'intero sviluppo; il punto di partenza è un simulatore costruito da Nicola Concer (Ad Hoc Sim, [27]) in cui saranno inserite delle funzioni aggiuntive per la creazione di diversi algoritmi di Clustering in una rete di sensori.

Nei diversi paragrafi che comporranno questo Capitolo, saranno analizzati i diversi strati che compongono l'intera struttura e, all'interno di essi, le funzioni principali che ne rendono possibile l'esecuzione.

## 4.1 STRUTTURA DEL SIMULATORE

Il simulatore, come anticipato precedentemente, è basato su OMNet++ v3.4 ed è stato scritto sull'ambiente Linux Ubuntu.

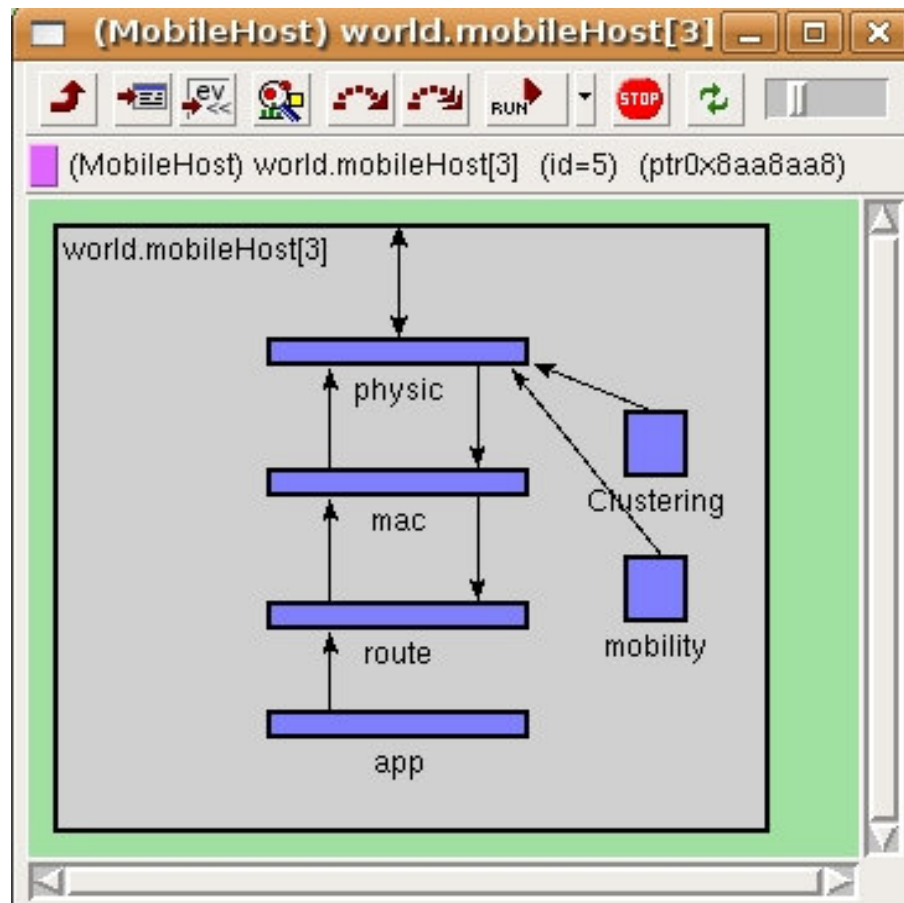
Il simulatore raffigura un'ad hoc network composta da 16 Host che si muovono in uno spazio libero da ostacoli; tutti i nodi hanno una stessa potenza in trasmissione iniziale e quindi uno stesso range (dipendente anche dall'energia di ogni nodo), stabilito all'inizio della simulazione in base alle caratteristiche della topologia della rete prescelta.

La degradazione del segnale di potenza è modellata attraverso il *Free Space Propagation Model*, il quale stabilisce che la forza con cui è ricevuto il segnale su ogni Host è inversamente proporzionale alla radice quadrata della distanza dal nodo.

Ogni mobile Host è un modulo *Compound* (il cui significato è spiegato in Appendice A) che incapsula i seguenti moduli *Simple* (come mostrato in figura 4.1):

- strato Fisico;
- strato MAC;
- strato di Routing;
- strato di Applicazione;
- strato di Mobilità;
- strato di Clustering.

La comunicazione tra i diversi moduli è fatta attraverso uno scambio di messaggi; ogni modulo (*Simple* o *Compound*) può essere modificato attraverso la variazione dei suoi parametri, assegnati nel file *omnetpp.ini*; tutti i moduli descritti sono racchiusi all'interno di una rete chiamata *world*, di cui si dispone soltanto del file *ned*.



**Fig. 4.1: Schema a blocchi degli strati della rete.**

Le analisi statiche, successivamente, sono inserite in dei files di testo (*statData.dat*) ed elaborate alla fine della simulazione; dato che ogni nuova simulazione sovrascrive i dati precedentemente creati, è creato uno script file che racchiude una copia di tutti i risultati per ogni simulazione.

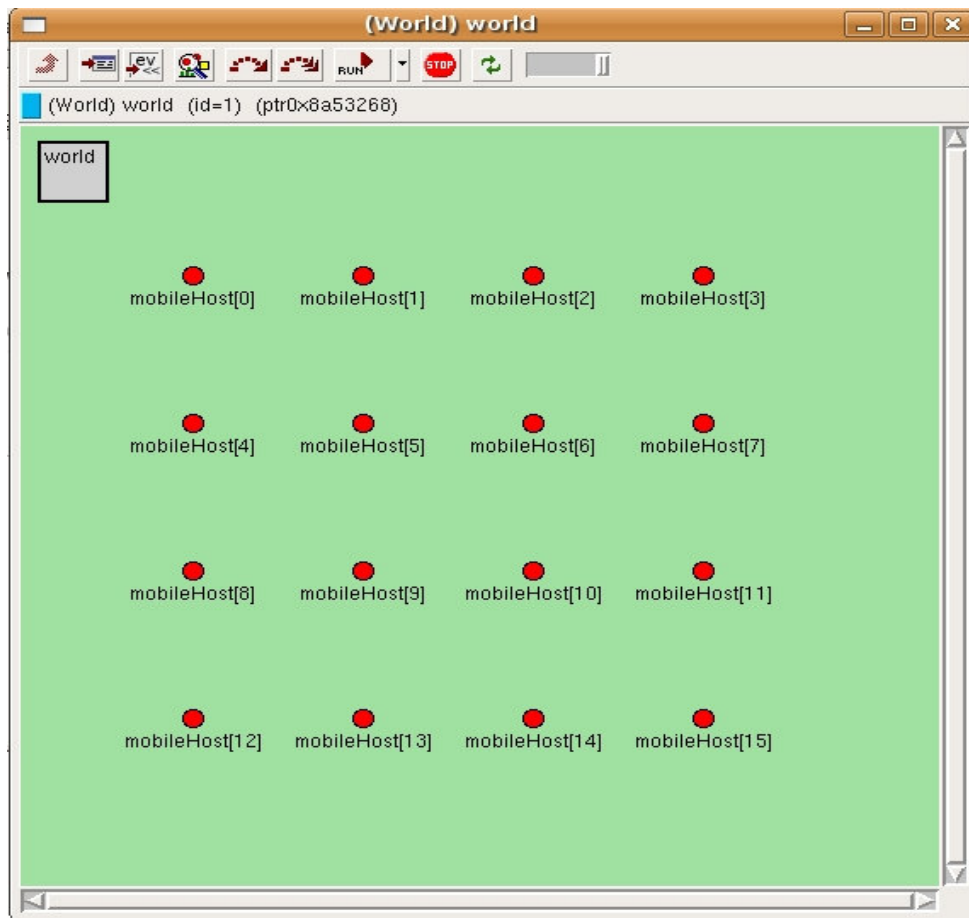
## 4.2 MODELLO FISICO

Il modello fisico implementa lo strato fisico di ogni Host; in particolare cura la creazione on-fly dei gates che permettono lo scambio di messaggi tra i vari apparati della rete.

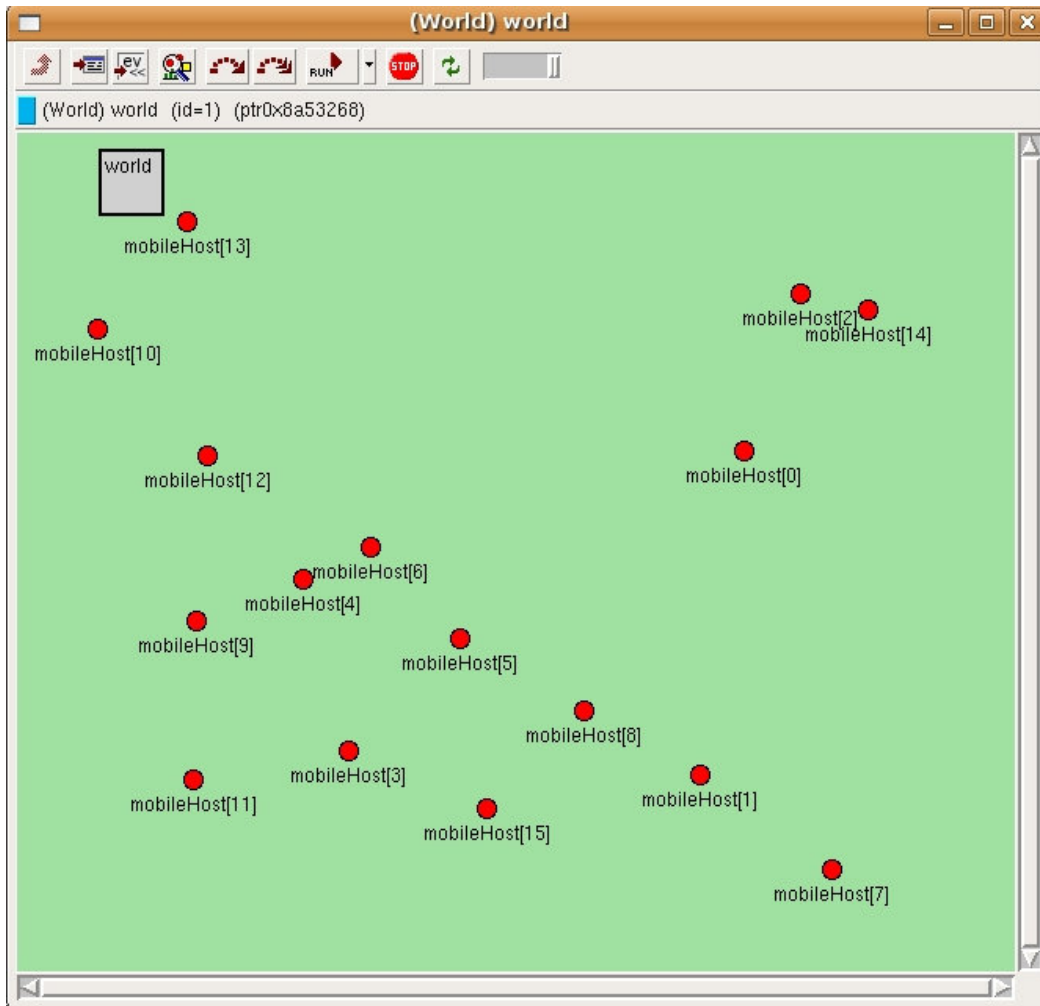
Inizialmente, attraverso la funzione *topology\_selection()*, inserita in aggiunta al



lavoro di Concer, si crea la topologia di rete sulla quale si vuol far operare tutta quanta la simulazione; le due topologie prescelte (come mostrato in Figura 4.2a e 4.2b) per operare una cluserizzazione su tutto il network sono: GRID e RANDOM (rispettivamente selezionate attraverso il settaggio nel file *omnetpp.ini* della variabile *world.topology = 'g'* oppure *'r'*).



**Fig. 4.2a: Topologia di rete GRID.**



**Fig. 4.2a: Topologia di rete RANDOM.**

Nella trattazione dell'intero scenario è importante operare una distinzione tra:

- ◆ **Topologia Logica:** definisce l'interconnessione tra nodi mediante canale;
- ◆ **Topologia Fisica:** tiene conto dell'effettivo percorso dei mezzi trasmissivi (considerando i link attivi e non).

Nelle considerazioni successive sarà sottintesa la trattazione della topologia fisica della rete, in quanto si va a considerare l'effettiva raggiungibilità dei nodi e la loro distribuzione in condizioni di totale visibilità, nel momento in cui si opera la clusterizzazione.

Per quanto riguarda la disposizione GRID (detta anche di Manhattan), si ha principalmente in reti a basso diametro, in cui i contatti tra i nodi sono stabiliti

tenendo conto della geometria dello scenario; si considera un insieme di nodi corrispondenti ai punti di una griglia  $k$ -dimensionale (nel nostro caso  $k=2$ ): nel caso bidimensionale si può calcolare la distanza tra due nodi di coordinate  $(i,j)$  e  $(k,l)$  come:

$$D((i,j);(k,l)) = |k-i| + |l-j|.$$

Uno dei maggiori vantaggi di questa topologia è dato dalla buonissima predisposizione ad essere clusterizzata, con una distribuzione uniforme dei contatti tra gli Host, a discapito, però, di un conseguente protocollo di routing e correzione dei guasti poco facilmente applicabili.

Per quanto concerne la topologia RANDOM, invece, i nodi sono solitamente sparsi uniformemente su tutta la rete ed assumono posizioni (considerando una matrice bidimensionale) in base al tipo di distribuzione adottata (es. Poissoniana); un caso particolare di distribuzione RANDOM è la maglia, in cui  $N$  nodi sono correlati da  $N(N-1)/2$  ed  $N-1$  collegamenti, nel caso in cui si tratti di maglia completamente connessa oppure no.

I vantaggi di questa topologia sono una buona tolleranza ai guasti, grazie alla presenza di più percorsi tra due nodi, una grande diffusione per la sua adattabilità a svariate situazioni, un numero di canali in trasmissione selezionabili a piacere ed una scelta ovvia, in caso di maglia completa (ovvero in cui tutti i nodi sono collegati tra loro), di un percorso a minima distanza.

Ogni volta che un Host si sposta dalla sua posizione, viene applicata una funzione di “controllo inter-distanza” (*detectNeighbours()*) alla quale si associa un'altra funzione (*isReachable()*) per controllare che effettivamente sia possibile stabilire una connessione tra i due nodi.

Se un Host si trova abbastanza vicino ad un altro (sempre considerando la potenza in trasmissione del nodo in movimento), sono effettuate queste operazioni:

- si crea un nuovo gate tra i due moduli *Compound* che identificano i due diversi

nodi;

- si crea un nuovo gate su ognuno dei moduli *Simple* *Physic* contenuti nel modulo del mobile Host (come anche per il passo precedente, funzioni *addNewGate()* e *setUpConn()*);
- si crea un link tra il nuovo gate creato sul modulo *Simple* e quello creato sul modulo *Compound* (funzione *connectModules()*);
- si crea un link tra i moduli dei due Host: quest'ultimo link usa le proprietà che contraddistinguono le trasmissioni attraverso un canale via etere (ovvero un determinato ritardo, throughput e probabilità d'errore), grazie all'uso della funzione *updateConnections()*.

Nel momento in cui i due nodi, a causa della loro mobilità, raggiungono posizioni troppo distanti, i gates precedentemente creati sono eliminati (attraverso la funzione *disconnectFrom()*) in base al proprio range di trasmissione, considerato a differenza di Concer uguale per tutti i nodi, ed il link per la comunicazione sarà interrotto (*putDownConn()*).

Ogni nodo ha la stessa potenza in trasmissione e quindi si ottiene che le connessioni tra essi avvengono in entrambe le direzioni.

Un'ulteriore restrizione da considerare nel momento in cui si stabiliscono le connessioni tra i nodi, è l'analisi dei campi *cHead*, *myHead* e *myCluster* che verranno analizzati nel paragrafo inerente al clustering e che serviranno a far instaurare connessioni soltanto tra tutti i CH ed i nodi appartenenti allo stesso cluster.

Il modulo *Physic* di un determinato nodo può ricevere messaggi da altri Host oppure dal modulo della mobilità, attraverso l' *handleMessage()*, in base allo stato di provenienza del messaggio considerato; compie operazioni di acquisizione o broadcasting (attraverso la funzione *broadcast()*) dello stesso; inoltre, all'interno dell' *handleMessage()*, è richiamata la funzione *calcolo\_Energia()* (sempre definita all'interno del modulo *Physic* e innovativa rispetto a Concer) attraverso la quale è possibile aggiornare il valore dell'energia disponibile per ogni nodo (partendo da un'energia iniziale settata nell' *omnetpp.ini*), considerando due diversi valori da

sottrarre, nel caso di ricezione ( $E_{RX}$ ) oppure trasmissione ( $E_{TX}$ ) del messaggio:

$$E_{TX} = E_{start} + L * (E_{Tx\_bit\_fixed} + N_s * T_s * P_{tx})$$

$$E_{RX} = E_{start} + L * E_{Rx\_bit}$$

dove  $L$  indica la lunghezza del pacchetto considerato (settato all'inizio de file physic.cc), mentre gli altri valori sono riportati nella tabella seguente:

Parameter	Value
$R_b$	1 Mb/s
$R_c$	50 Kb/s
$E_{start}$	$2.76 \cdot 10^{-5} J$
$E_{Tx\_bit\_rate}$	$3.25 \cdot 10^{-7} J$
$E_{Tx\_bit\_prop}$	$1.25 \cdot 10^{-11} J$
$E_{Rx\_bit\_fixed}$	$1.13 \cdot 10^{-7} J$
$E_{Rx\_bit\_rate}$	$2.79 \cdot 10^{-7} J$

Nel momento in cui un Host riceve un messaggio, se proviene dall'esterno e non contiene errori e non è affetto dall'interferenza provocata da altri apparati che trasmettono o ricevono contemporaneamente, è inviato direttamente ai livelli superiori; in caso contrario, se il messaggio proviene dal modulo Mobility, (e quindi serve a settare i nuovi parametri di posizione del nodo) il ricevitore provvederà a memorizzare tutte le modifiche contenute.

Nel momento in cui un modulo appartenente ad un livello superiore decide di inviare un messaggio, lo farà verso lo strato fisico che si occuperà del corretto inoltrò dello stesso; questo modulo, infatti, contiene una lista dei nodi vicini raggiungibili, scandendo la quale invia una copia del messaggio originale verso tutti i gates che connettono l'Host verso gli altri nodi.

## 4.3 MODELLO ED ANALISI DELLA MOBILITÀ

Il modulo presentato in questo paragrafo è l'unico che non è presente nella struttura del sottomodulo dell'Host, ovvero quella ISO/OSI: proprio come si può capire dal nome, questo modulo si occupa della mobilità del nodo nel quale è incapsulato.

Ogni Host ha un proprio modulo di mobilità che può essere uno dei diversi algoritmi implementati: cambiando, soltanto all'interno del file *omnetp.ini*, il valore del parametro *world.mobileHost[\*].mobilityModel*, è possibile assegnare uno dei seguenti algoritmi, seguiti da una breve spiegazione che ne caratterizza il funzionamento:

- 1. *Random Walk mobility model*:** questo algoritmo è inteso nell'interpretazione discreta del modello di mobilità Browniano; è caratterizzato da una imprevedibilità di movimento, in cui velocità e direzione non sono correlate. A causa di suddetta imprevedibilità, questo algoritmo produce un movimento non realistico ed è quindi usato solo come il peggior caso possibile di mobilità. Ci possono essere vari modi di implementare questo algoritmo, uno dei quali è generare delle angolazioni e delle velocità uniformemente distribuiti tra dei

valori definiti da alcuni parametri, che resteranno tali finchè non si copre una distanza predefinita; per coprire distanze diverse, si sceglieranno nuove angolazioni e velocità.

2. ***Restricted Random Walk mobility model (RRW<sup>3</sup>)***: deriva dal modello standard dell'algoritmo descritto precedentemente e si diversifica da esso soltanto nel momento in cui bisogna scegliere un nuovo angolo ed una nuova direzione. (*RRW<sup>3</sup>*) prova a risolvere il problema della casualità del movimento dei nodi scegliendo questi nuovi valori all'interno di un range ben limitato da altri parametri; in questo modo la posizione dei nodi è molto più uniformemente distribuita e consente una maggiore copertura di tutta quanta la rete.
3. ***Random Waypoint mobility model***: è uno degli algoritmi di mobilità più usati (infatti è quello selezionato nel file *omnetpp.ini* di questo simulatore per effettuare le simulazioni inerenti soltanto agli ultimi due algoritmi di clustering analizzati, caratterizzati da una topologia di rete RANDOM, più libera da vincoli fissi rispetto a quella a *GRID*) nelle simulazioni su reti ad hoc: riesce ad essere molto semplice e nello stesso momento abbastanza realistico. Un Host sceglie un punto di arrivo all'interno dei suoi possibili movimenti, una velocità uniformemente distribuita in un range ben preciso, e si muove verso la sua destinazione nel tempo richiesto: una volta raggiunto il punto prestabilito, il nodo vi permane per un periodo chiamato *pause time* e successivamente sceglie una nuova destinazione. Una pecca di questo algoritmo è il concentrare un grosso numero di nodi al centro del sistema, creando una densità molto alta; questo problema si può aggirare aumentando la velocità media di ogni Host, in modo da ottenere più destinazioni possibili ed un maggiore distribuzione su tutta quanta la rete.
4. ***Random Direction mobility model***: questo algoritmo è un mix tra il precedente ed il *Random Walk mobility model*; per evitare una concentrazione massiccia al centro della rete, ogni nodo sceglie una velocità ed una direzione distribuite

uniformemente e va avanti finchè non raggiunge un bordo del network: arrivato a questo punto, l'Host, dopo aver atteso per un *pause time*, sceglierà una nuova direzione ed una nuova velocità; maggiore sarà la velocità del nodo, in minor tempo raggiungerà il bordo della rete e più lungo sarà il *pause time*.

**5. *Normal Markovian mobility model*:** è un tipo di algoritmo differente dai precedenti, il quale deriva da modelli Markoviani, usati in reti cellulari piuttosto che in ad hoc networks; in ogni passo, questo modello sceglie una nuova direzione ed una nuova velocità selezionando un valore determinato da un range di parametri molto vicini a quelli precedentemente scelti di velocità e direzione. In questo modo si conferisce al nodo un movimento più uniforme che si mantiene costante intorno all'angolazione scelta in origine.

Una volta scelto uno degli algoritmi precedentemente elencati si deve studiare il comportamento dei nodi nel momento in cui si hanno delle collisioni con il bordo della rete durante il movimento; si possono attuare due accorgimenti distinti:

- i. l'Host rimbalza elasticamente dopo l'urto con la parete;
- ii. l'Host, successivamente all'urto, compie un percorso virtuale per cui rientra dalla parte opposta delle rete.

Ogni modulo inerente ad ognuno di questi algoritmi è implementato nello stesso modo: all'inizio il modulo Mobility schedula un *Self Message*; una volta consegnato, il modulo elabora una nuova posizione (attraverso l'applicazione dell'algoritmo prescelto)  $\langle x,y \rangle$  in accordo anche con la corrente posizione dell'Host. Questi valori saranno salvati in due parametri del nuovo *MOVE message* creato; successivamente il modulo di mobilità invia questo messaggio al modulo Physic e rischedula il *Self message* da consegnare nell'arco di  $moveInterval^2$  secondi.



## 4.4 STRATO MAC

Questo modello raffigura tutta la pila protocollare ISO/OSI MAC: in questa struttura è possibile inserire diversi protocolli di contesa al canale come, ad esempio, il CSMA/CA (Carrier Sensin Multiple Access-Collision Avoidance) o molti altri algoritmi esistenti.

L'implementazione di questo strato è la più semplice di tutti i moduli *Simple* che compongono la rete: i messaggi in uscita sono fatti passare senza particolari manipolazioni, mentre quelli d'ingresso sono consegnati ai livelli più alti con una politica *MMI queue*.

Quando un messaggio entra nello strato MAC, il modulo controlla un flag che indica

se lo strato superiore è occupato; in questo caso lo pone nel buffer o, se questo è pieno, lo scarta. Quando lo strato superiore si libera, il modulo MAC prende il primo messaggio nel buffer, lo invia al layer superiore e schedula un messaggio di *end of service* che renderà possibile l'invio di un altro pacchetto dati oppure farà settare il *busy-flag* come libero.

Questo livello, inoltre, controlla il *MAC address* di tutti i messaggi entranti e fa passare soltanto quelli che sono indirizzati a lui oppure sono stati inviati in broadcast. Il nodo, inoltre, può anche lavorare in *promisue mode*, ovvero “scaricando” il compito di elaborare i messaggi (anche quelli rivolti a se) al livello superiore: questa soluzione può essere anche molto dannosa per l'intero sistema di comunicazione, ma è una risorsa molto importante per tutti i possibili protocolli di routing on-demand che si possono applicare.

## **4.5 STRATO DI ROUTING**

Il modulo per la gestione del routing è fondamentale nell'economia dell'intero sistema: l'algoritmo adoperato è l' AODV (Ad hoc On-demand Distance Vector) ed è settato tra lo strato delle applicazioni e quello di accesso al mezzo.

Il Routing riceve i messaggi DATA dai livelli più alti e cerca di instradarli verso la destinazione prescelta nella sua routing table oppure inviando un messaggio di controllo (*RREQ*) per cercare una nuova via.

Tutti questi messaggi di controllo hanno una lunghezza pari a 512 byte (mentre OMNet++ la misura in bit) e contengono diversi campi di controllo; la lunghezza del pacchetto DATA, invece, può essere scelta settando il suo valore nel file *omnetpp.ini*.

L'AODV è stato studiato per essere il più adattabile e flessibile possibile: questa particolarità conferisce al protocollo la possibilità di avere molte opzioni che l'utente può scegliere per le implementazioni, a seconda delle caratteristiche del sistema.

Questo simulatore implementa le seguenti opzioni, adatte ad un'ad hoc network:

- messaggio *HELLO* scambiato tra nodi vicini: per carpire informazioni sugli Host presenti nel range del nodo corrente, nel momento in cui gli strati superiori non forniscono valori soddisfacenti;
- l'ottimizzazione *expanding ring search* usata per regolare l'invio broadcast degli *RREQ*: prima è inviata una richiesta di istradamento verso una destinazione con un numero di hop *tll* limitato e conosciuto da tutti i nodi; successivamente, se la richiesta fallisce, se ne invia un'altra con un numero di *tll* più alto; questo procedimento può essere effettuato solo un numero prestabilito di volte, prima che il tentativo di trasmissione sia terminato.
- uso del messaggio di *ACK* per confermare la corretta ricezione dei messaggi *RREP*: questo messaggio è usato soprattutto nello strato di trasporto;

- l'uso di una *black list* per evitare i nodi “da diffidare”: un nodo *X* inserisce un Host *Y* nella *black list* nel momento in cui, dopo aver cercato di inviargli un *RREP* un determinato numero di volte, non riceve nessun *ACK*. Questa situazione accade quando il link che collega *X* con *Y* è unidirezionale, ovvero il primo ascolta i messaggi inviati dal secondo, ma non avviene il caso contrario; *X* non considererà nessun messaggio *RREQ* proveniente da *Y*, mentre la stessa cosa non avverrà per il messaggio *HELLO*; nello scenario considerato in questo lavoro, come già detto precedentemente, a differenza di Concer i link saranno considerati bidirezionali e quindi la situazione sarà diversa da quella spiegata in precedenza.

Come visto nella parte precedente, il modulo AODV usa una grande varietà di messaggi, ognuno dei quali è definito da un valore costante e da determinati parametri (nel file *constants.h*).

Di seguito è riportata una breve lista dei messaggi più utilizzati:

- ✓ *HELLO*: messaggio di Hello;
- ✓ *RREQ*: messaggio di Route Request;
- ✓ *RREP*: Messaggio di Route Replay;
- ✓ *RRER*: Messaggio di Route Error, contenente una lista di tutte le destinazioni non più valide;
- ✓ *DATA*: Messaggio di dati;
- ✓ *ACK*: Messaggio di conferma in risposta ad un *RREQ* o *RREP*;
- ✓ *DELETE*: Un Self Message per attestare un istradamento fallito;
- ✓ *SEND\_HELLO*: Un Self Message che chiede ad un Host di inviare un nuovo messaggio di Hello.

Ognuno di questi messaggi contiene dei parametri, come anticipato precedentemente, usati per lo scambio di informazioni di controllo tra i vari nodi, inseriti in una lista che li classifica in base al loro ordine di importanza; i campi più importanti sono:

- × *originator*: contenuto nei messaggi *RREP* e *RREQ*, contenendt l'*ID* dell'Host che ha generato il messaggio;

- x *dest*: usato nel *RREQ* per specificare l'Host con cui vuole comunicare il nodo trasmittente;
- x *seqNumS*: immagazzina il numero di sequenza del messaggio originato; se è processato in molti hop, rimane invariato per tutto il percorso;
- x *seqNumD*: immagazzina l'ultimo numero di sequenza conosciuto del nodo destinatario (pari a 0 se l'Host non è conosciuto);
- x *source*: immagazzina l' *ID* dell'ultimo Host da cui è passato il messaggio (inteso naturalmente come nodo vicino);
- x *mac*: registra l' *ID* del nodo successivo verso cui è rivolto il messaggio, durante l'istadamento;
- x *ttl*: esprime il valore del *time to live* del messaggio, espresso in numero di hop;
- x *hopNum*: conta il numero di hop attraversati dal messaggio.

Nel momento in cui un nodo riceve un messaggio *DATA*, sono immagazzinate diverse informazioni per generare le analisi statistiche; in particolare sono effettuate tre tipi di operazioni:

- si registra il numero di hop compiuto dal pacchetto in un istogramma che può essere esaminato dopo un determinato tempo di attesa;
- si misura il tempo di latenza del messaggio inviato e si immagazzina in un vettore che servirà a calcolare il tempo di latenza di tutto il cammino del messaggio in accordo con il numero complessivo di hop attraversati;
- si compiono le stesse operazioni descritte precedentemente anche per il valore del throughput, calcolato come latenza/dimensione del messaggio;
- si incrementa un counter per ogni messaggio consegnato per decretare il rapporto di consegna di tutto il protocollo.

## 4.6 STRATO DI APPLICAZIONI

Questo modulo *Simple* è molto semplice ma fondamentale per il funzionamento di tutto quanto il simulatore; grazie ad esso, infatti, cinque possibili Host possono generare contemporaneamente una richiesta di connessione ed inviarla ad una destinazione qualsiasi per poter successivamente trasmettere dati.

Attraverso le funzioni osservate precedentemente nello strato fisico, sarà deciso l'eventuale scambio di informazioni tra trasmettitore e ricevitore, oppure l'interruzione della connessione.

L'indirizzo del nodo destinatario del messaggio è scelto attraverso l'elaborazione, da parte del protocollo di routing, di una tabella contenente tutti gli indirizzi dei vari mobile Host, per poter così iniziare lo scambio di dati.

Rispetto a [27], saranno aggiunte due semplici funzioni: *getsent\_pkt()* e *resetsent\_pkt()*: la prima servirà a far ritornare i valori all'interno del vettore *sent\_pkt[]*, costituito per il calcolo dei messaggi ricevuti da parte di ogni nodo (utile per l'applicazione dell'algoritmo di clusterizzazione 5), mentre la seconda servirà a resettare, una volta usato, il vettore precedentemente citato.

## 4.7 GESTIONE DELL'INTERFERENZA

Nel lavoro illustrato in [27], punto di partenza per la creazione di questo scenario, non è stato affrontato un problema molto importante, inerente all'interferenza che si ottiene nel momento in cui più Host trasmettono messaggi contemporaneamente.

Il modulo *Simple inspector.cc*, insieme all'inserimento di altre funzioni basilari all'interno di *physic.cc*, permette di considerare tutta quanta la rete da un punto di vista molto più realistico e corretto.

All'interno di *physic.cc* sono inserite delle funzioni che permettono di ottenere un canale trasmissivo con requisiti uguali a quelli che si hanno in una UWB Network e quindi nel protocollo 802.15.4a; da questo punto di vista è fondamentale l'introduzione della funzione *addError()*, in quanto si considera la propagazione del segnale attraverso il canale radio ed i conseguenti fenomeni di shadowing (o fading lento), che affievolisce il segnale a causa del multipath presente, delay e pathloss; inoltre, sempre nella funzione precedentemente citata, sono presenti il calcolo della *packetErrorProbability* (ovvero la probabilità che un pacchetto dati arrivi in modo errato alla destinazione) ed il computo, per ogni nodo trasmittente, del numero di interferenti che ne possono limitare l'azione.

Nella tabella seguente sono riportati tutti i valori delle costanti utilizzate per la creazione del canale UWB e dell'interferenza multi-utente:

<i>Costante</i>	<i>Valore</i>
PL0_802_15_4a[5]	{0, 43.9, 48.7, 43.29, 43.29}
pathloss_exp_802_15_4a[5]	{0, 1.79, 4.58, 1.76, 2.5}
shadow_stdv_802_15_4a[5]	{0, 2.22, 3.51, 0.83, 2}
pathLossdB	$47 + 17 * \log_{10}(\text{distance})$
snapshotInterval	100s
statInterval	5
Ts	6.41e-8
Ns	16
Tm	2e-9



Ptx	3.66e-05
R0	0
EU	$P_{tx} * T_s$
EbN0	$E_u * N_s / N_0$
BER	$0.5 * \text{erfc}(\sqrt{0.5 * E_b N_0 * (1.0 - R_0)})$
pktSize	1024
packetErrorProbability	$1 - \text{pow}(1 - \text{BER}, \text{pktSize})$
noisePower	$\text{noiseDensity} * \text{PacketRate} = 1.99\text{E-}12$
PacketRate	1000000
NoiseFactor	1.5; Linear value (NOT dB!)
NoiseDensity	4E-21
targetSNR	316.22; Linear value (NOT dB!)

Nel considerare l'effetto dell'interferenza multi-utente nella trasmissione dei dati, è molto importante sottolineare come si prenda il caso più pessimistico possibile nello scarto dei pacchetti affetti da errori: basta un solo bit errato, infatti, per marcare il pacchetto come corrotto e quindi scartarlo.

L'azione di tutte le funzioni correlate per la considerazione dell'interferenza è abbastanza semplice: una volta consolidato che la trasmissione di un messaggio debba avvenire sul canale definito precedentemente (*REAL\_CHANNEL*), ogni utente che si appresta ad inviare dati deve calcolare la durata del pacchetto, di tutti i ritardi correlati nella propagazione, creare una copia del messaggio ed inviarlo attraverso un *sendDirect()* a tutte le possibili destinazioni, scandite in precedenza e considerate in base alla loro distanza. Dal punto di vista della ricezione, invece, ogni Host controlla chi sta trasmettendo contemporaneamente, con quale potenza, e in base a quello decide se scartare il pacchetto (per l'interferenza che è stata creata) oppure no.

Nel momento in cui subentrano gli algoritmi di clusterizzazione, comunque, indipendentemente da quale delle due topologie (*GRID* e *RANDOM*) è stata precedentemente scelta, sono considerati degli accorgimenti aggiuntivi sia in trasmissione che in ricezione dei pacchetti, assumendo che i quattro CH comunichino su una sorta di “canale privilegiato”, diverso da quello usato per le comunicazioni intercluster :

- Lato Trasmissione: se trasmette un CH sentirà l'interferenza soltanto dei nodi che trasmettono all'interno del suo cluster (nel caso trasmetta verso uno di essi)

oppure degli altri CH, se trasmette sul “canale privilegiato”.

- Lato Ricezione: l'interferenza è sentita dal nodo che riceve in modo speculare rispetto a quanto avviene in trasmissione.

Nel capitolo successivo saranno analizzati, studiando gli esiti delle simulazioni terminate, i vantaggi che la clusterizzazione della rete apporta anche dal punto di vista dell'interferenza e del numero di pacchetti dati arrivati correttamente a destinazione.

## 4.8 IL CLUSTERING

Come precedentemente anticipato nel paragrafo 4.2, lo scenario analizzato comprende due diverse topologie di rete, ognuna composta da sedici mobile Host: GRID e RANDOM.

Durante le simulazioni saranno applicati cinque diversi algoritmi di clustering distribuiti (i primi tre per la topologia GRID, molto rigidi e con delle imposizioni iniziali ben precise, e gli altri due per quella RANDOM), scelti attraverso il settaggio del parametro *world.topologyAlgo* all'interno del file *omnetpp.ini*.

Prima di descrivere i vari algoritmi, è fondamentale analizzare la definizione ed il settaggio di tre diversi parametri in base ai quali sono effettuate tutte le connessioni nello strato fisico:

1. ***cHead***: questo parametro può assumere solo due valori: 1 se il nodo in questione è un CH; 0 se l'Host è un semplice componente del cluster;

2. *myCluster*: indica a quale cluster appartiene il nodo corrente;

3. *myHead*: indica quale nodo è il CH del cluster a cui appartiene l'Host corrente.

Per quanto concerne le connessioni tra i vari nodi, invece, esse saranno effettuate in base ai valori *myCluster* e *cHead*: per tutte e cinque gli algoritmi, infatti, saranno connessi tra di loro tutti e quattro i CH scelti, mentre gli altri nodi semplici potranno comunicare soltanto con gli Host che avranno il parametro *myCluster* settato con il loro stesso valore.

Di seguito saranno brevemente descritti i diversi tipi di algoritmi applicati e saranno illustrati graficamente gli scenari che compongono (considerando un fatto di scala pari a 10 per ottenere una visualizzazione dei nodi più ristretta):

- *world.topologyAlgo = '1'*: La griglia di nodi è divisa in quattro cluster di diverso colore composti ognuno da quattro nodi (verde, giallo, blu, bianco, come mostrato in Figura 4.3), mentre i CH son tutti dello stesso colore (nero); solo quattro dei sedici nodi complessivi saranno CH e colorati di nero, in quanto avranno la possibilità di comunicare tra loro; la disposizione dei CH e dei cluster di appartenenza, in questo algoritmo così come nel *world.topologyAlgo = '2' e '3'*, è decisa a priori attraverso il settaggio dei tre parametri precedentemente citati (*cHead*, *myCluster*, *myHead*).

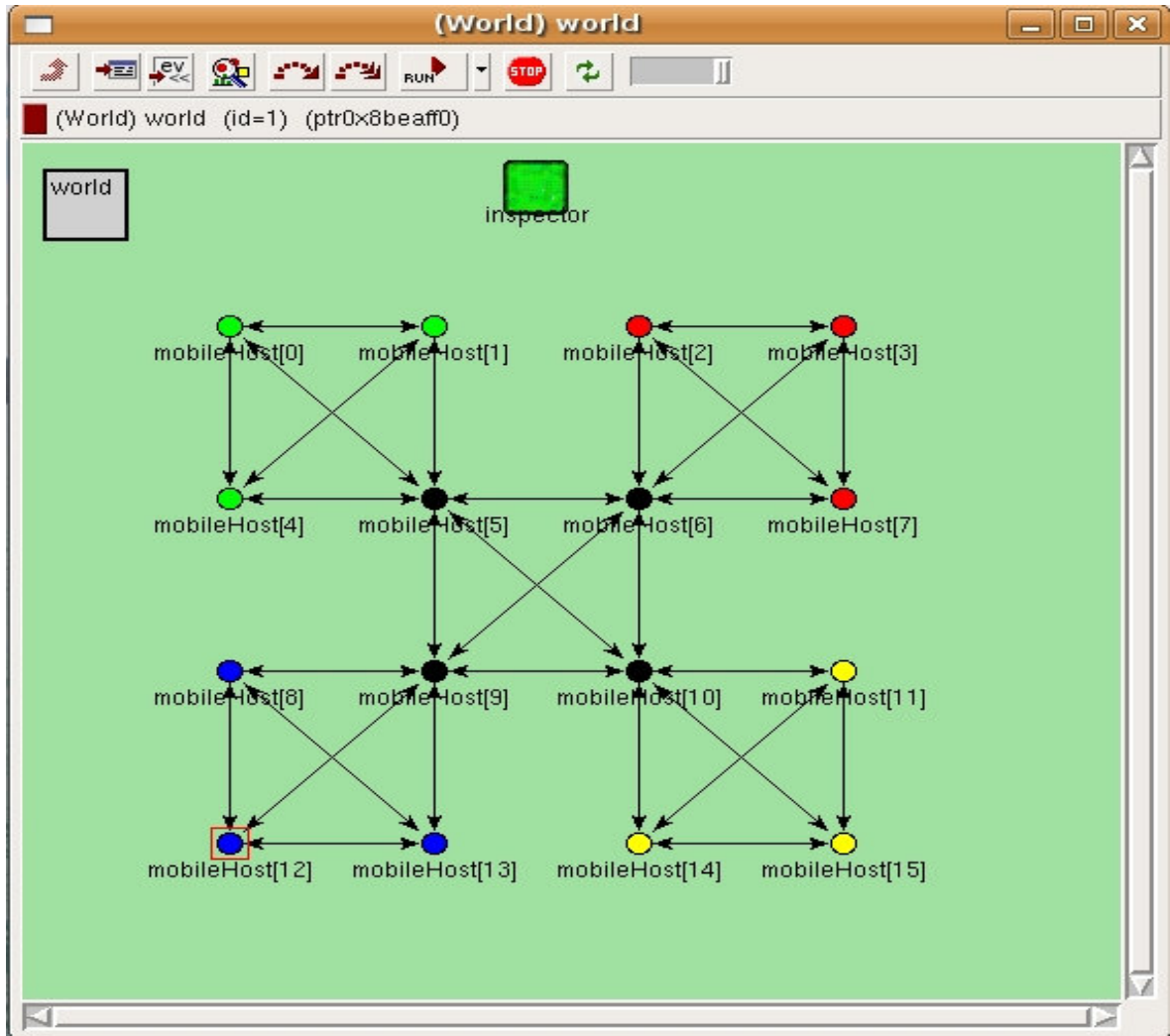
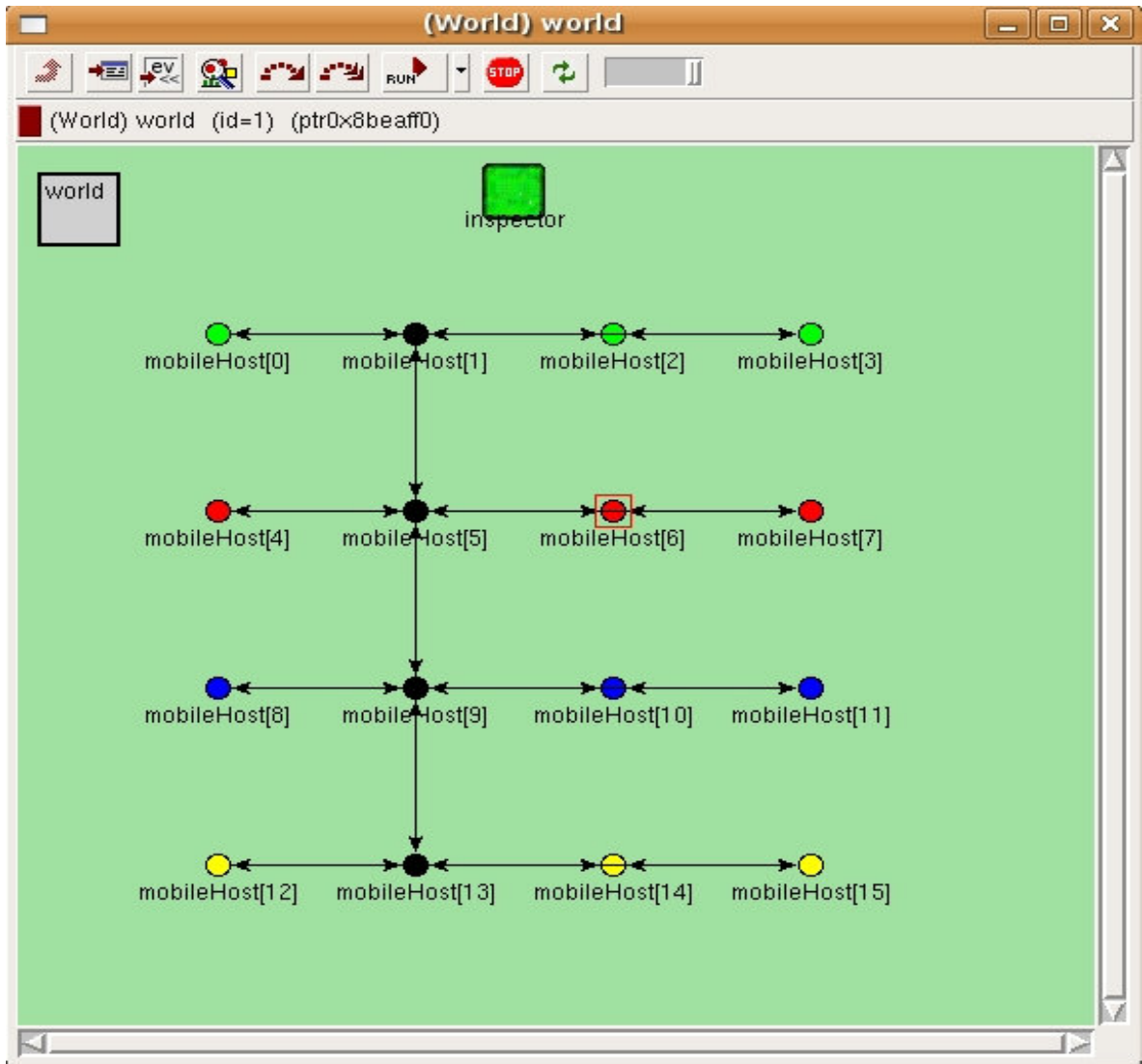


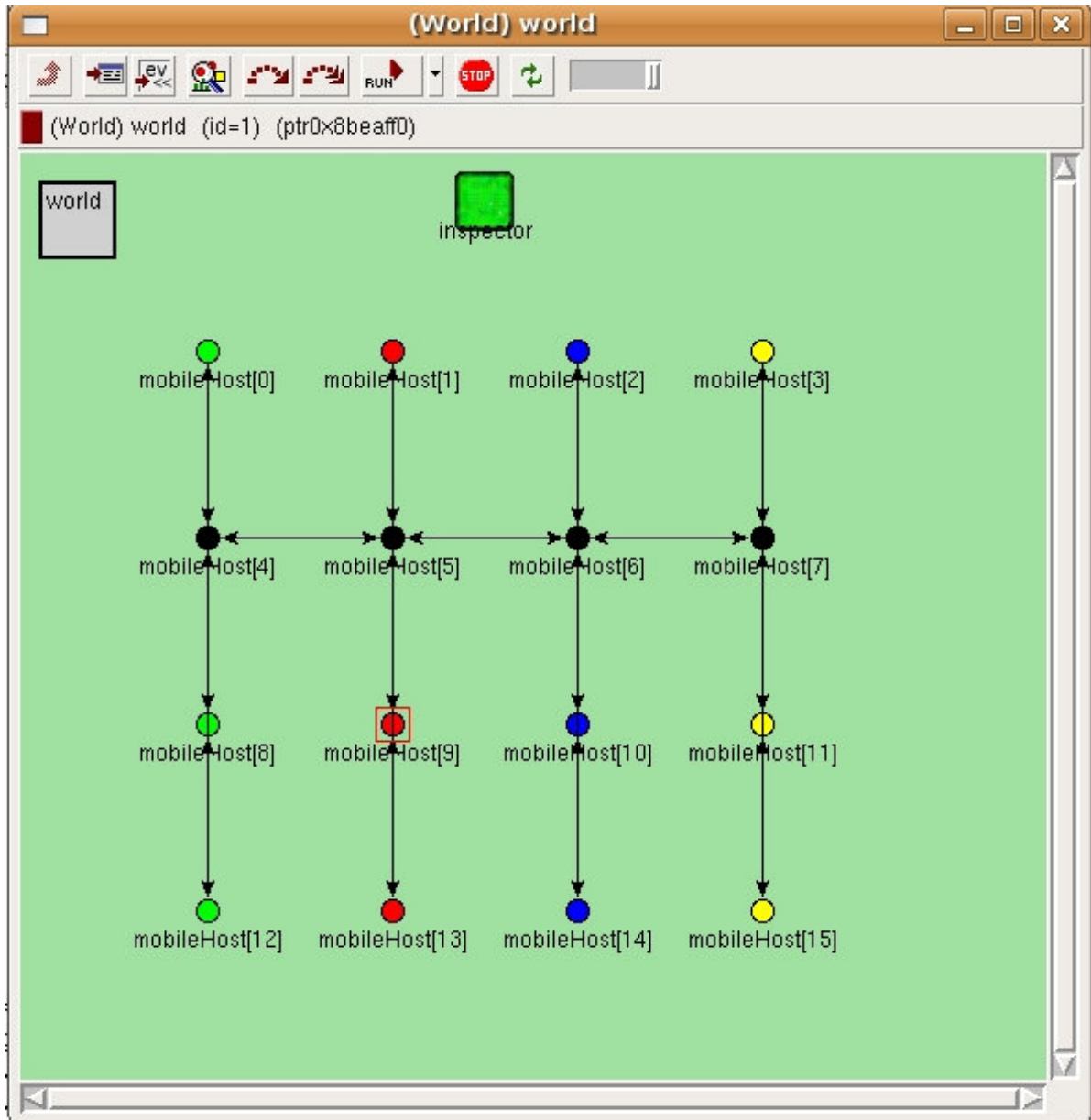
Fig. 4.3: *world.topologyAlgo = '1'* in Topologia GRID.

- *world.topologyAlgo = '2'*: La griglia di nodi è divisa in altri quattro cluster (di diverso colore, come mostrato in Figura 4.4) composti ognuno da quattro nodi; rispetto all'algorithm precedente cambia soltanto la composizione dei clusters, mentre a livello strutturale avvengono le stesse operazioni;



**Fig. 4.4:** *world.topologyAlgo = '2'* in Topologia **GRID**.

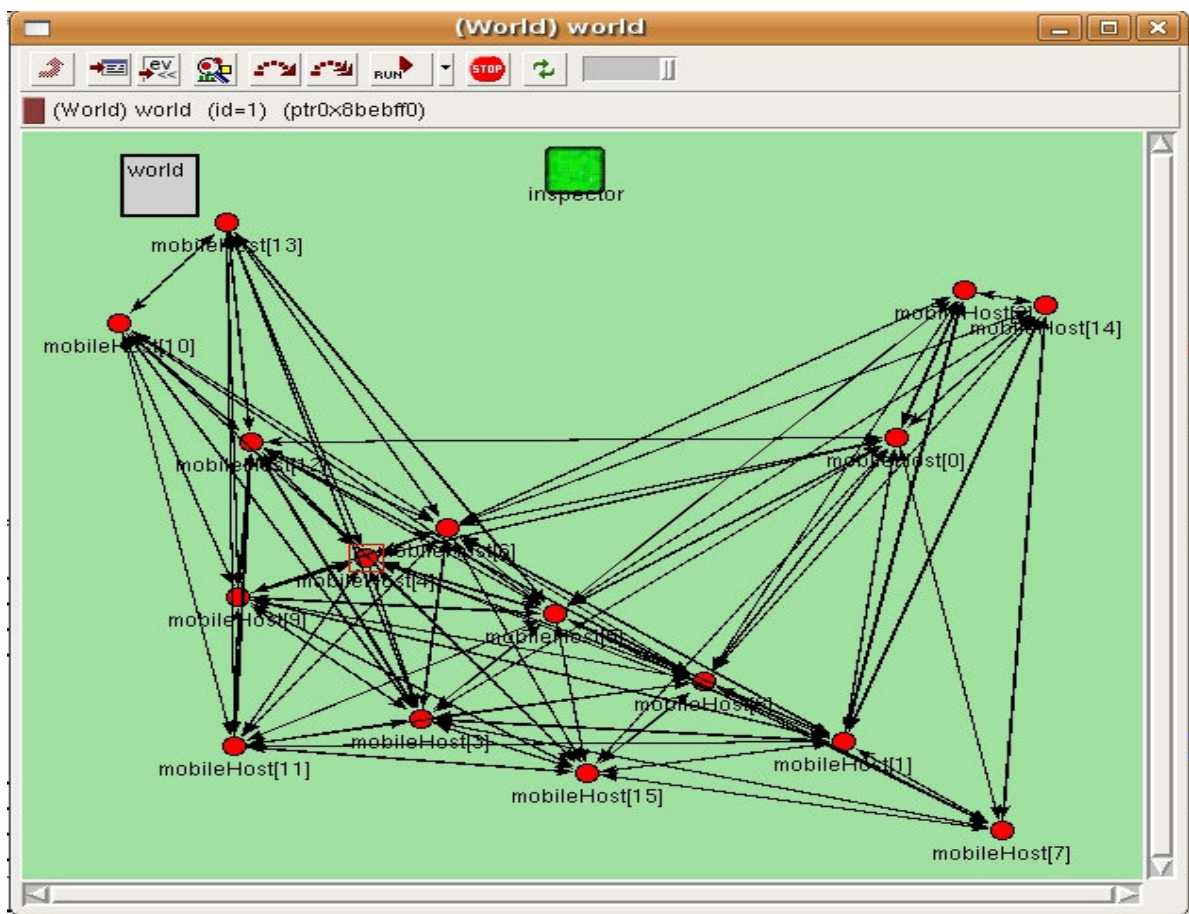
- *world.topologyAlgo = '3'*: La griglia di nodi è divisa sempre in quattro cluster (di diverso colore, come mostrato in Figura 4.5) composti ognuno da quattro nodi, disposti in un modo ancora diverso rispetto ai precedenti due casi e con le stesse caratteristiche di funzionamento;



**Fig. 4.5:** *world.topologyAlgo* = '3' in Topologia GRID.

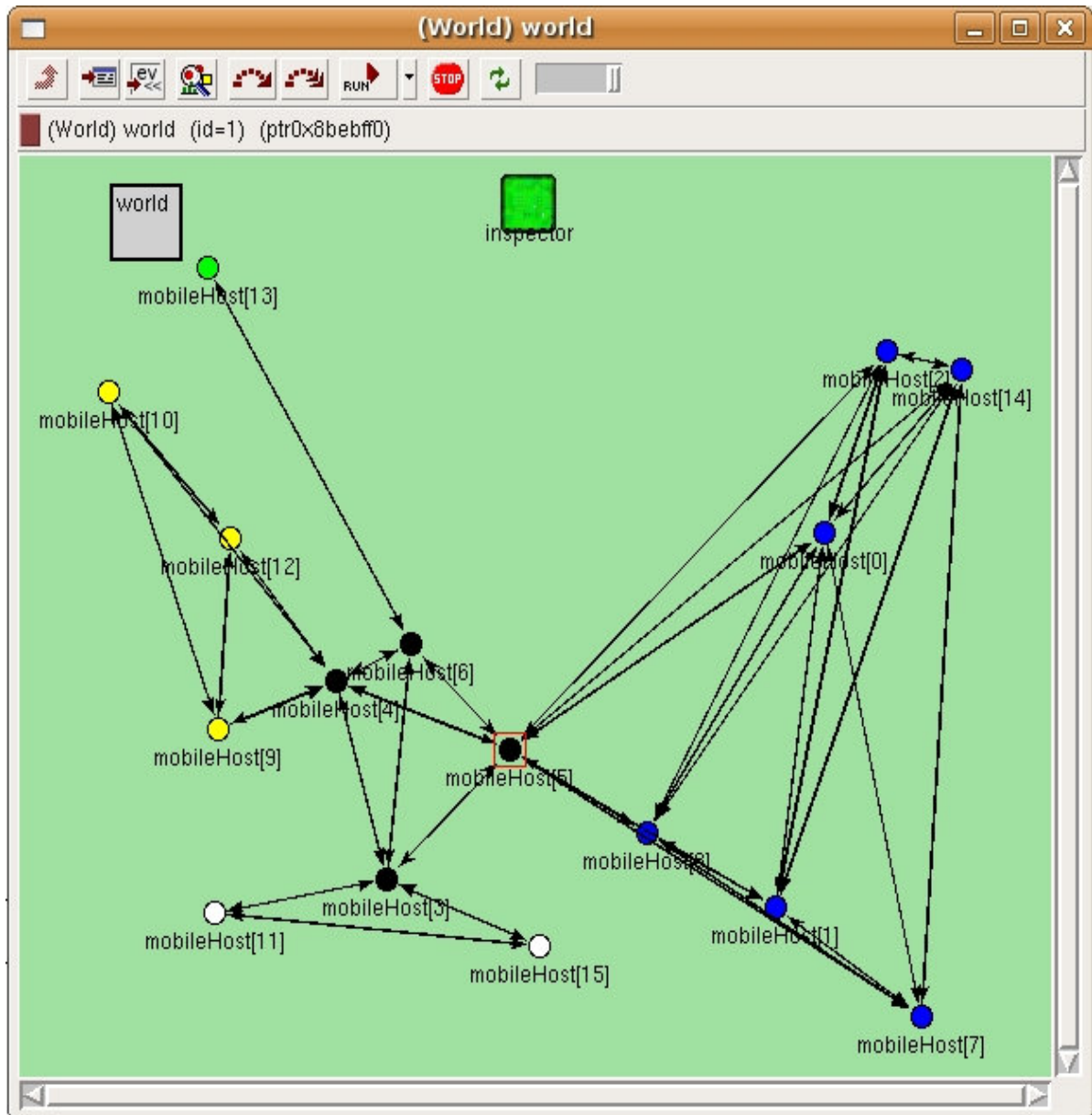
- ***world.topologyAlgo* = '4':** L'insieme dei nodi è diviso in quattro cluster (di diverso colore, come mostrato in Figura 4.6b) composti ognuno da un numero variabile di mobile Host; la scelta dei CH e dei componenti di ogni cluster, in questo caso, verrà fatta attraverso l'applicazione di un algoritmo (EEUC, analizzato nel Capitolo 3) che predilige i nodi a maggior energia disponibile, nel momento in cui si deve effettuare l'elezione dei quattro CH; inizialmente si considera uno scenario in cui si ha un unico grande cluster che comprende tutti i nodi presenti (come mostrato in Figura 4.6a): dopo un certo *timeout* (settato

in *Clustering.cc* a 200 secondi), sarà effettuato un calcolo dell'energia disponibile per ogni nodo della rete ed i quattro Host col valore più alto saranno scelti come CH; successivamente si calcolerà la distanza tra ogni CH ed ogni nodo semplice e, nel caso in cui essa sia minore di una determinata soglia prescelta ( $R_{comp}$ , considerata in base al *range* di trasmissione, uguale per tutti i nodi all'interno della rete) il nodo sarà associato al suo cluster di appartenenza.



**Fig. 4.6a:** *world.topologyAlgo = '4'* in Topologia *RANDOM*, prima della clusterizzazione.





**Fig. 4.6b:** *world.topologyAlgo* = '4' in Topologia *RANDOM*, dopo la clusterizzazione.

Si noti come nel caso in cui un nodo possa appartenere a più clusters diversi, sarà associato a quello che avrà il CH con distanza minore da esso. E' importante notare anche come, allo scadere del *timeout*, cambino anche tutte le connessioni in base alla formazione dei vari cluster.

- ***world.topologyAlgo* = '5':** L'insieme dei nodi è diviso in quattro cluster (di diverso colore, come mostrato in Figura 4.7b) composti ognuno da un numero variabile di mobile Host; anche in questo caso, così come nel caso precedente,

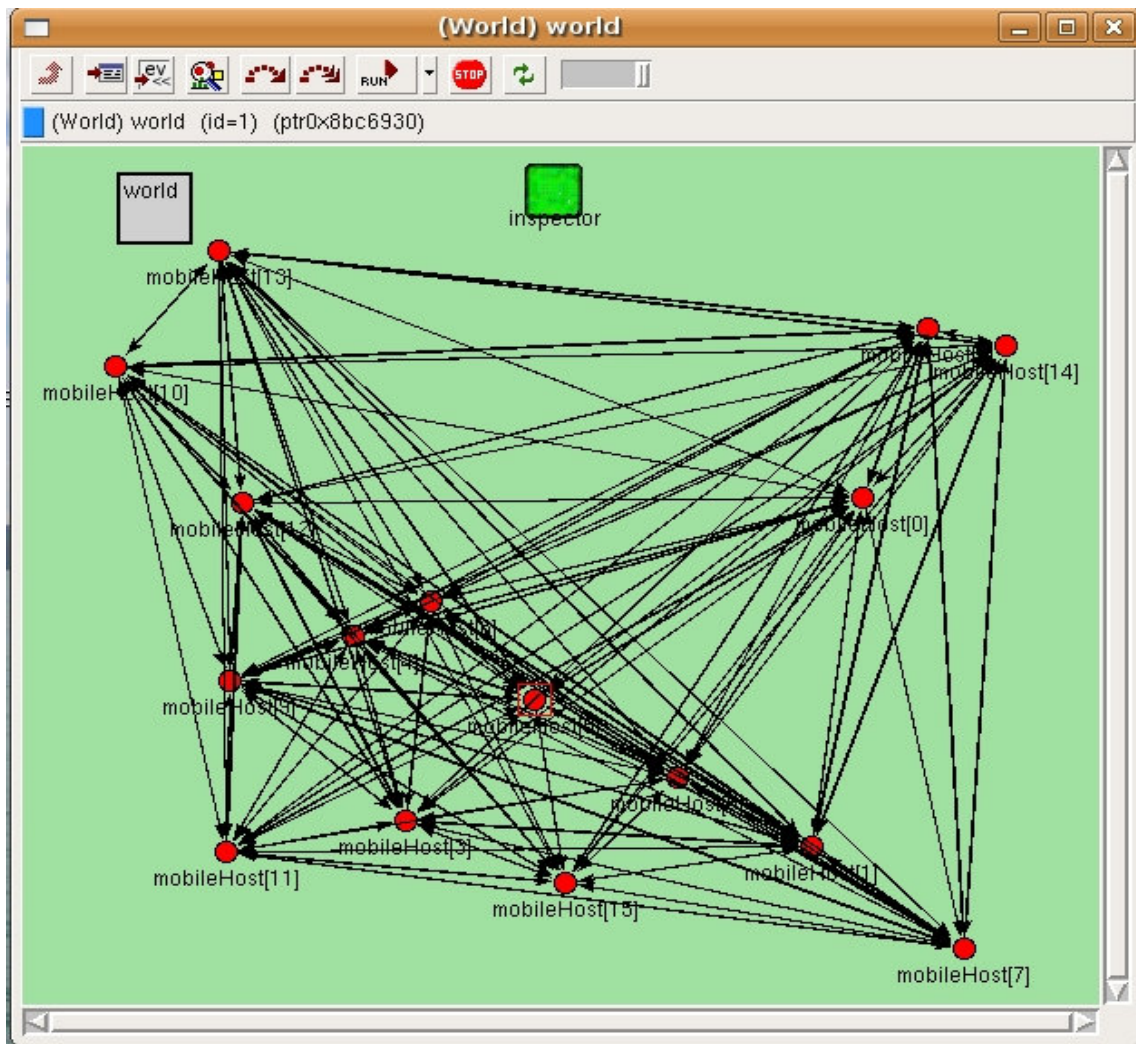


tutti i nodi che compongono lo scenario saranno clusterizzati, ma il procedimento sarà fatto in base a requisiti differenti. In questo algoritmo la scelta dei diversi CH sarà operata in base ad un criterio ben preciso: si considera il numero di messaggi che ogni nodo invia verso ogni singola destinazione (*dest*); questo conteggio è reso possibile attraverso l'incrementazione nel file *application.cc* del vettore *sent\_pkt[dest]*.

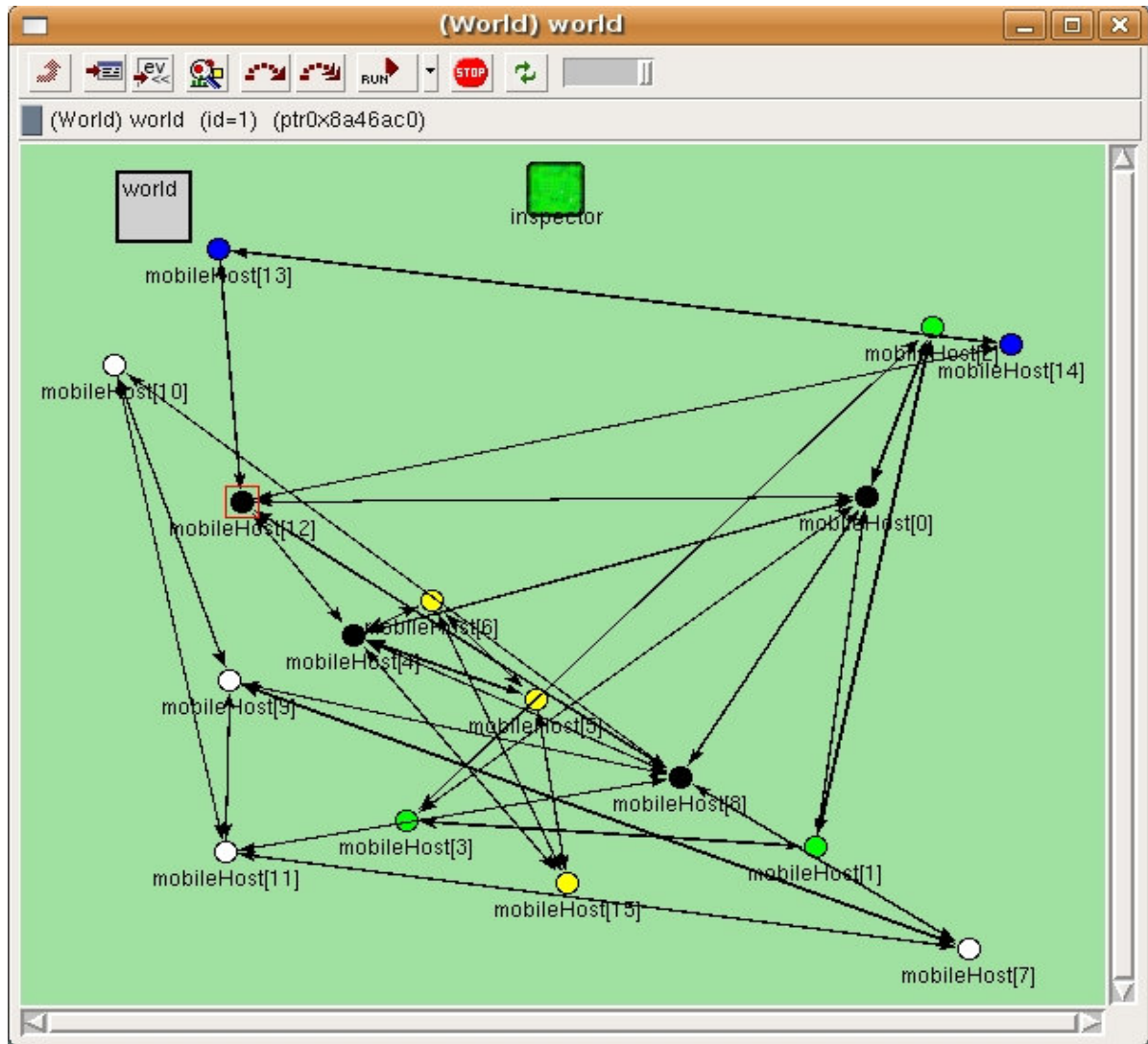
Successivamente in *Clustering.cc* si crea la matrice *matrixsent\_pkt[[[]]*, in cui ogni riga è data dal vettore precedentemente definito e relativo ad ogni singolo nodo; sommando tutti gli elementi di ogni colonna della matrice, quindi, si ottiene il numero totale di ricezioni che compie ogni singolo Host.

Questa topologia, però, è molto simile a quella a GRID (dal punto di vista della clusterizzazione molto rigida) in quanto si “forza” sia la scelta dei quattro CH che quella dei nodi che compongono i clusters relativi; si scelgono gli Host [0], [4], [8], [12] come uniche destinazioni possibili e, di conseguenza, quattro gruppi composti da tre nodi che trasmettono soltanto verso uno dei futuri CH stabiliti.

Anche in questo caso, come nell'algoritmo precedente, si considera prima il sistema come un unico grande cluster (come mostrato in Figura 4.7a), mentre allo scadere di un *timeout2* (anch'esso settato a 200 secondi) si effettuerà la formazione di cluster di dimensioni minori; in questo caso la scelta dei componenti di ogni cluster sarà fatta in base al numero di messaggi scambiati da ogni nodo con ogni CH e quindi associerà in modo naturale ad ogni CH i soli tre nodi che trasmettono verso di lui: ogni nodo semplice, infatti, avrà un vettore (*Vett\_connect\_to\_CH[]*) che racchiude il numero di messaggi scambiati tra se ed ogni singolo CH e che servirà per l'attuazione della clusterizzazione.



**Fig. 4.7a:** *world.topologyAlgo = '5'* in Topologia *RANDOM*, prima della clusterizzazione.



**Fig. 4.7b:** *world.topologyAlgo = '5'* in Topologia *RANDOM*, dopo la clusterizzazione.

Nel capitolo successivo saranno analizzati i risultati delle simulazioni effettuate per ogni singola topologia ed ogni singolo algoritmo di cluster, mettendo in evidenza quali sono i benefici e gli svantaggi susseguenti all'applicazione di queste tecniche.

## 4.9 MODELLO DI TRAFFICO

Questo modulo serve per generare il traffico dati che è manipolato in tutte le successive operazioni di routing.

Ogni Host ha un proprio generatore di traffico che può essere considerato attivo o meno, in base al settaggio del parametro *world.mobileHost[\*].app.active* presente nel file *omnetpp.ini*.

Il traffico è modellato generando un pacchetto a burst di sessantaquattro messaggi spediti verso una destinazione casuale, che rimane la stessa per tutta la durata del burst: il rate al quale sono inviati i dati è definito dal settaggio presente nel file *omnetpp.ini*; all'interno di questo file è definito, inoltre, il *burstInterval* che indica il tempo trascorso tra due application burst.

Come già detto in precedenza, ogni Host contiene un ID assegnatogli dal kernel di OMNet++ all'inizio della simulazione; questi ID non sono in sequenza e possono variare a seconda del numero totale di moduli che lavorano durante la simulazione; per generare un corretto *destination number*, evitando di dover scansionare tutto il vettore di puntatori del modulo considerato dal kernel del simulatore, il modulo operante usa un semplice puntatore allo strato fisico che restituisce una lista di tutte le destinazioni possibili; questo accorgimento non si distingue per flessibilità ma consente un risparmio sostanzioso di memoria e cicli della CPU del sistema.

## CAPITOLO 5

# ANALISI DELLE SIMULAZIONI

In quest'ultimo capitolo saranno analizzati i risultati ottenuti dalle simulazioni effettuate attraverso OMNet++ e saranno spiegati i modi ed i parametri assunti per

ottenere le conclusioni finali.

Come anticipato nel capitolo precedente, le topologie di rete analizzate sono essenzialmente due:

- GRID (più rigida e analizzata senza mobilità);
- RANDOM (più flessibile ed analizzata con e senza mobilità).

Per la prima topologia sarà considerato solo uno scenario di dimensioni 375x375m, mentre per la seconda si prenderanno in considerazione uno scenario 300x300m ed uno 450x450, andando ad effettuare un accorgimento in frequenza di clusterizzazione nel momento in cui si prende in considerazione uno spazio più ampio in cui sono predisposti i nodi.

Per quanto riguarda il numero di Host che compongono la rete, esso rimarrà costante (16) e non varierà a seconda della scelta della topologia, mentre il numero di trasmettitori attivi sarà lo stesso per tutte le simulazioni considerate (12).

I parametri principali su cui si basa l'analisi dei risultati di tutte le simulazioni sono essenzialmente quattro:

1. *Troughput*: inteso come  $(\text{numero di pacchetti dati ricevuti})/(\text{numero pacchetti dati effettivamente trasmessi})$  alla fine di ogni simulazione.
2. *Energia Media del Sistema*: intesa come energia disponibile, mediando il valore presente in ogni singolo nodo, di tutto il sistema, partendo da un valore iniziale di 100 J. E' importante notare come i valori ottenuti in presenza e assenza di clusterizzazione siano comunque vicini, in quanto, a causa del modello di interferenza introdotto, i messaggi sono sempre ricevuti da tutti i nodi (il che comporta comunque dispendio energetico) e successivamente immagazzinati o rilanciati a seconda che sia destinazione o meno.
3. *Numero di Hop Medio*: inteso come il numero medio di hop che deve attraversare un pacchetto dati nella rete per ogni connessione instaurata.
4. *Tempo di Vita del Sistema*: inteso come il tempo che impiega un nodo, per primo, ad esaurire la sua energia.

La durata delle simulazioni varia a seconda della topologia considerata ed è pari a

## 90 - ANALISI DELLE SIMULAZIONI

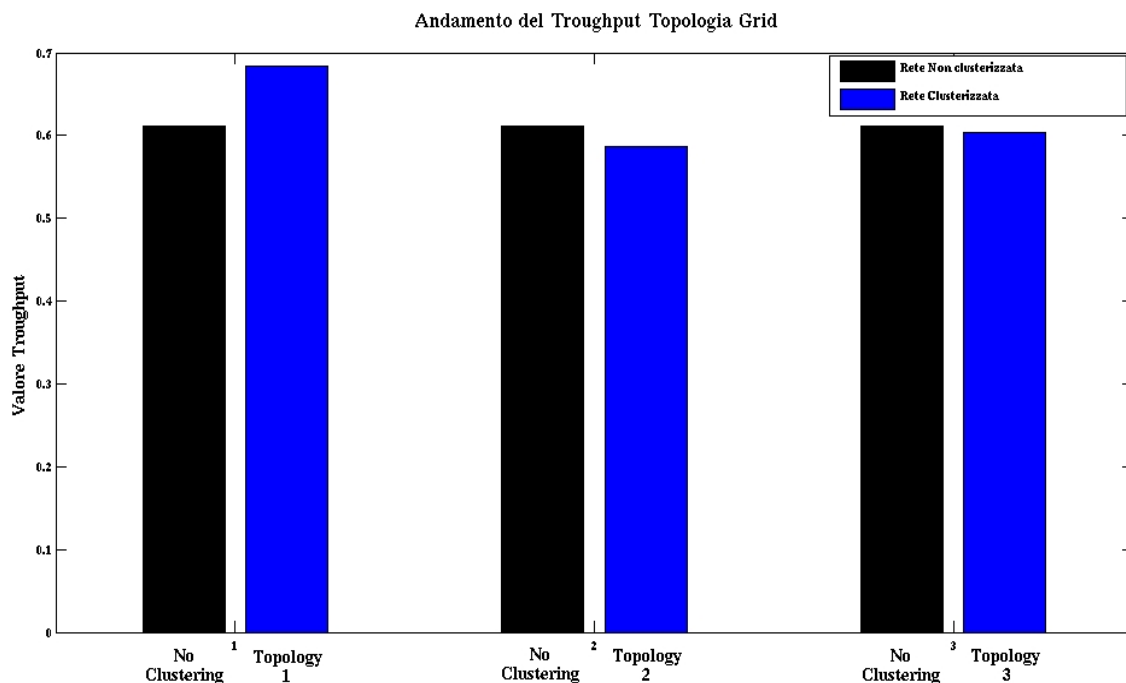
1200s (20m) per la GRID e 3200s (circa 53m) per quella RANDOM: questa differenza è data dalla maggiore o minore rapidità con cui i grafici ottenuti dalle simulazioni raggiungevano un andamento asintotico pressochè costante.

Nel proseguio di questo capitolo saranno, quindi, prima trattate tutte le statistiche inerenti alla topologia GRID e successivamente quelle per la topologia Random.

I valori graficati sono ottenuti per mezzo di una funzione (*Statistiche()*) presente nel file *aodv.cc*, attraverso la quale ad intervalli di 50s (per la topologia GRID) o 160s (per la topologia RANDOM) per un periodo totale rispettivamente di 1200s e 3200s, sono stampati i valori dei parametri precedentemente citati su un file (*statData.dat*); in entrambe le topologie sono state eseguite 5 run e si è effettuata una media per ottenere la quantità numerica finale da inserire nei grafici.

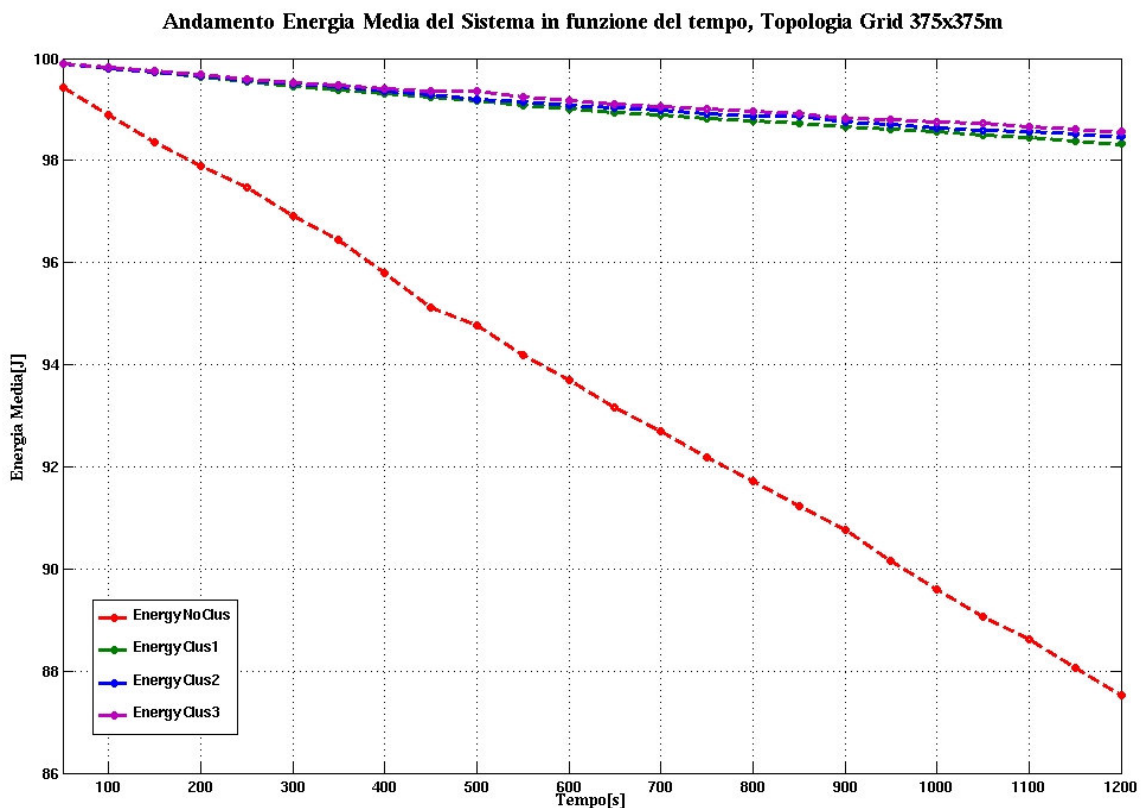
## 5.1 ANALISI TOPOLOGIA GRID

Per quanto concerne questa prima topologia, dai risultati ottenuti si può considerare come la clusterizzazione non sempre migliori le prestazioni globali del sistema se non effettuata con criterio; dal punto di vista del throughput (come mostrato in Figura 5.1, in cui si raffronta il valore percentuale di throughput senza clusterizzazione con i tre diversi modelli di clustering) si ha un miglioramento solo con la soluzione 1, in cui i quattro CH sono equidistanti gli uni dagli altri, mentre con le altre topologie, in cui i CH assumono posizione diverse, non si hanno miglioramenti ma addirittura le cose peggiorano.



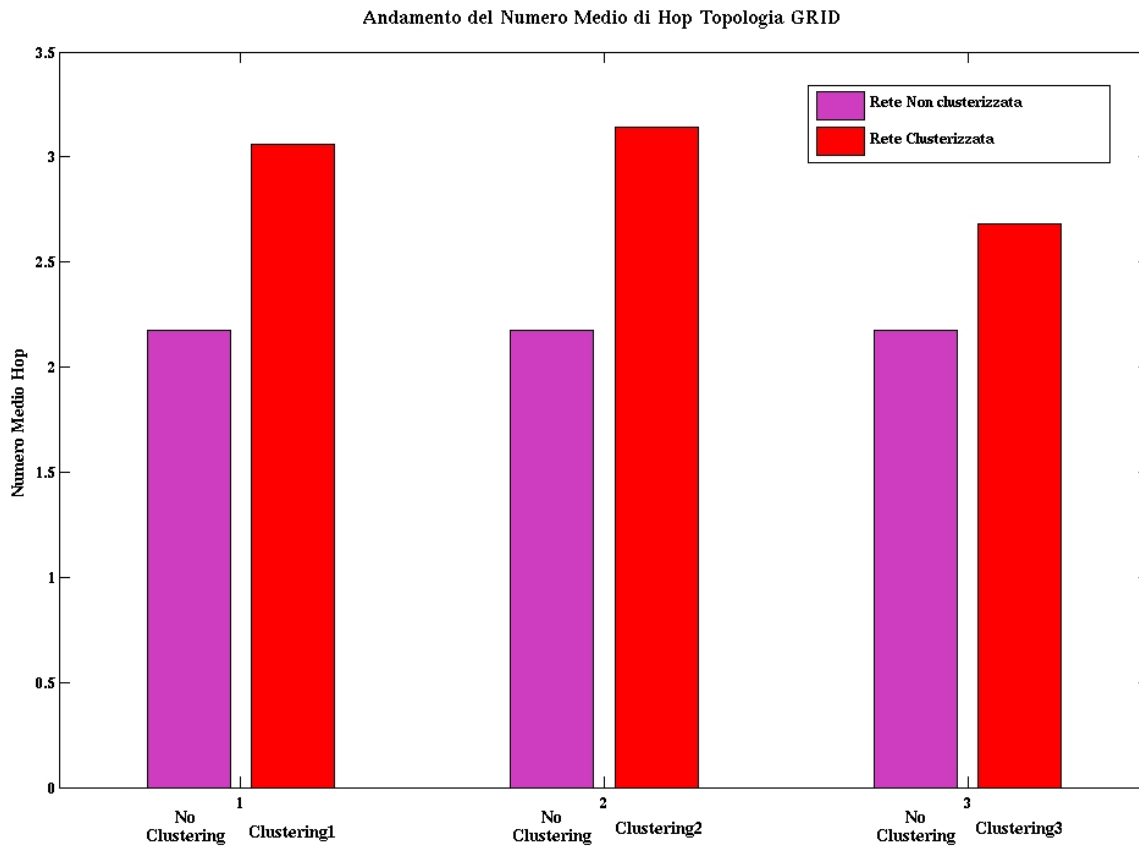
**Fig. 5.1: Andamento del Troughput in Topologia GRID.**

Per quanto riguarda il consumo medio di energia dell'intero sistema, invece, la clusterizzazione offre migliori consumi in tutte e tre le topologie, andando ad allungare di molto il tempo di vita dell'intero sistema (come mostrato in Figura 5.2).

**Fig. 5.2: Andamento dell'Energia media del Sistema in funzione del tempo in Topologia GRID.**

Per quanto riguarda il numero medio di hop del sistema, invece, la clusterizzazione “forza” l'instradamento dei pacchetti in quanto la comunicazione tra due nodi appartenenti a cluster differenti, anche se vicini tra di loro, può avvenire soltanto per mezzo dei CH e quindi saranno impiegati come minimo tre hop per lo scambio di informazioni; in tutti e tre i casi, dunque, come mostrato in figura 5.3, attuando la clusterizzazione si ha un incremento del numero di hop medio rispetto ad uno scenario libero.



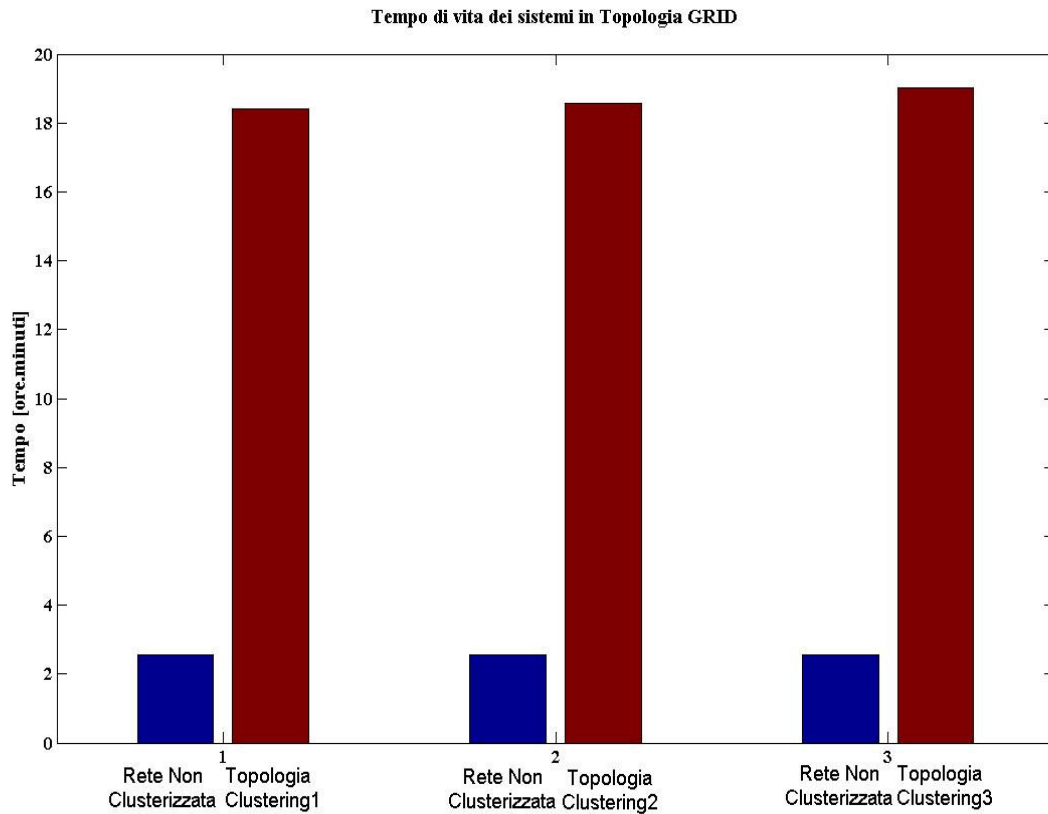


**Fig 5.3: Numero medio di Hop del sistema in Topologia GRID.**

A conclusione di questo capitolo è molto importante analizzare come la clusterizzazione dia vantaggi soprattutto dal punto di vista energetico, in quanto, mentre si sforzano i CH a smaltire una quantità di traffico maggiore, e quindi ad un maggiore dispendio di energia, gli altri nodi sono meno esposti a questo pericolo a vantaggio dell'economia dell'intero sistema, considerando soprattutto la tipologia di instradamento e di calcolo di dispendio energetico applicato al simulatore: un messaggio destinato ad un nodo, infatti, una volta raggiunti Host diversi viene comunque immagazzinato e rilanciato, contribuendo a sottrarre energia al terminale in cui avviene questa operazione.

In Figura 5.4, quindi, si può notare come tutti e tre i tipi di clustering (con andamento molto simile) apportino un miglioramento in termini di tempo di vita del sistema,

inteso come il momento in cui il primo nodo cessa di agire a causa dell'azzeramento del suo livello energetico.



**Fig. 5.4:** Confronto del tempo di vita del sistema, Topologia GRID, con e senza clusterizzazione.

## 5.2 ANALISI TOPOLOGIA RANDOM

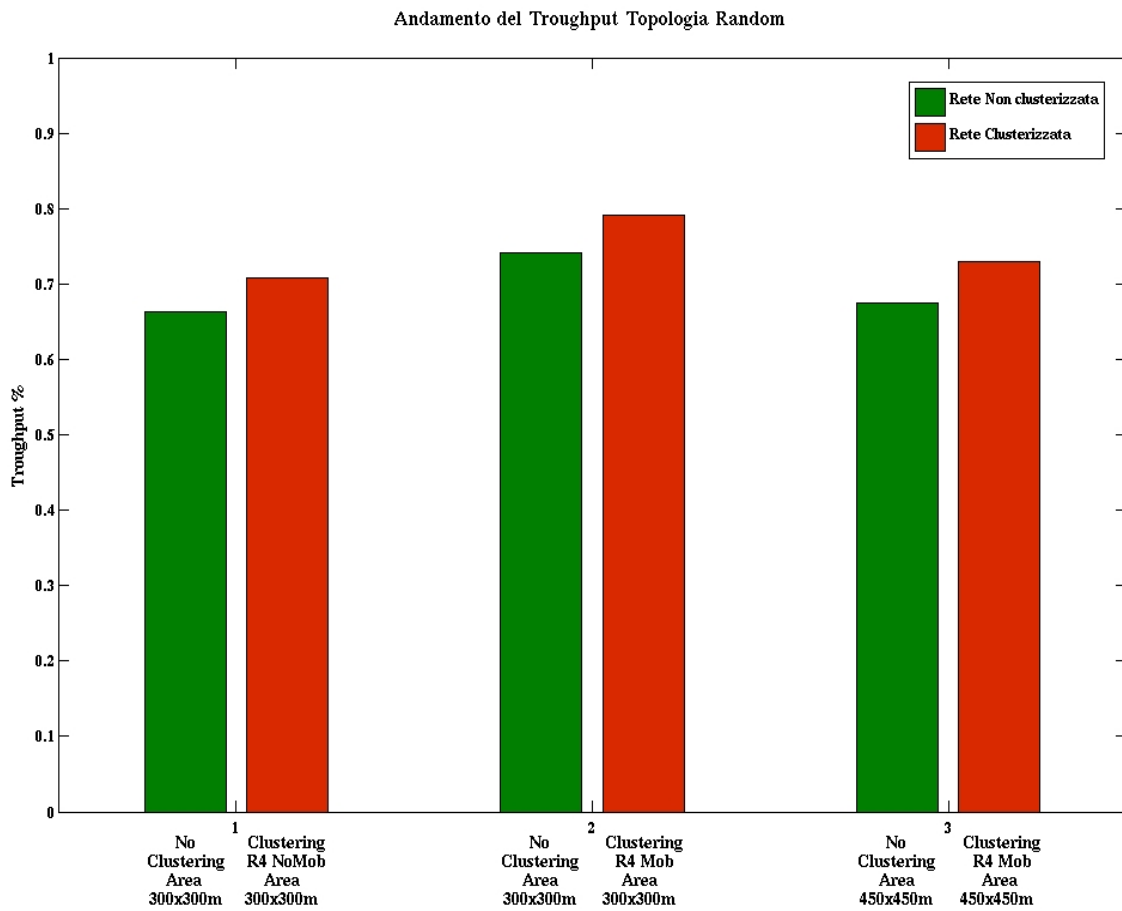
La topologia RANDOM è stata oggetto di un numero maggiore di simulazioni rispetto alla precedente in quanto le sue caratteristiche la rendono molto più flessibile e applicabile in moltissimi ambienti reali; i due algoritmi di clusterizzazione

analizzati (EEUC e in base al numero di trasmissioni, come spiegato nel Capitolo 4) sono stati applicati in assenza di mobilità e con mobilità RANDOM WAYPOINT, in cui i nodi si muovono ad una velocità che varia tra 1m/s e 0,5m/s.

Le dimensioni dello scenario, inoltre, variano tra 300x300m e 450x450m: nel secondo caso, per la maggiore distanza presente tra i singoli nodi ed il possibile cambio, con il passare del tempo, della formazione dei clusters, si introduce un refresh che ogni 200s (stesso tempo dopo il quale, durante una iniziale simulazione senza clusterizzazione, subentrano gli algoritmi di clustering) aggiorna la procedura di elezione dei CH e la successiva composizione dei clusters ad essi associati.

Nel primo slot di simulazioni sono messe a confronto le prestazioni del sistema nel momento in cui si attua una clusterizzazione di tipo R4, senza mobilità in 300x300m e con mobilità sia in 300x300m che in 450x450m: ogni valore è raffrontato con il corrispettivo in assenza di clusterizzazione.

Per quanto riguarda il throughput, come mostrato in Figura 5.5, si può notare come in ogni singola topologia ci sia un miglioramento di circa 4 punti percentuali; si noti come in area più ristretta i valori con mobilità siano più elevati rispetto a quelli senza mobilità, (grazie probabilmente ancora alla vicinanza degli apparati) mentre aumentando il raggio d'azione il throughput diminuisce.



**Fig. 5.5: Andamento del Troughput in Clustering R4.**

Dal punto di vista energetico, in Figura 5.6 sono graficati gli andamenti del clustering R4 nelle tre diverse condizioni: in tutti e tre i casi, rispetto allo scenario non clusterizzato, si ottiene un risparmio energetico abbastanza evidente che contribuisce ad aumentare la vita media del sistema.

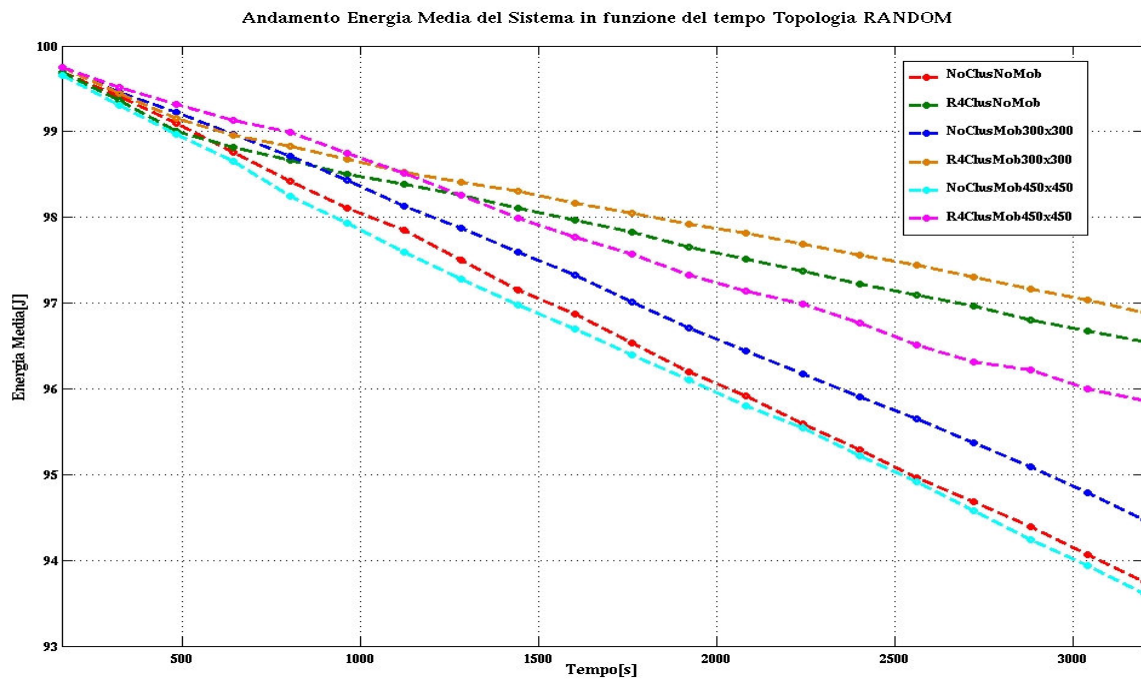


Fig. 5.6: Andamento dell'Energia in funzione del Tempo nel Clustering R4.

In Figura 5.7, infine, è graficato l'andamento del numero di hop medio del sistema:

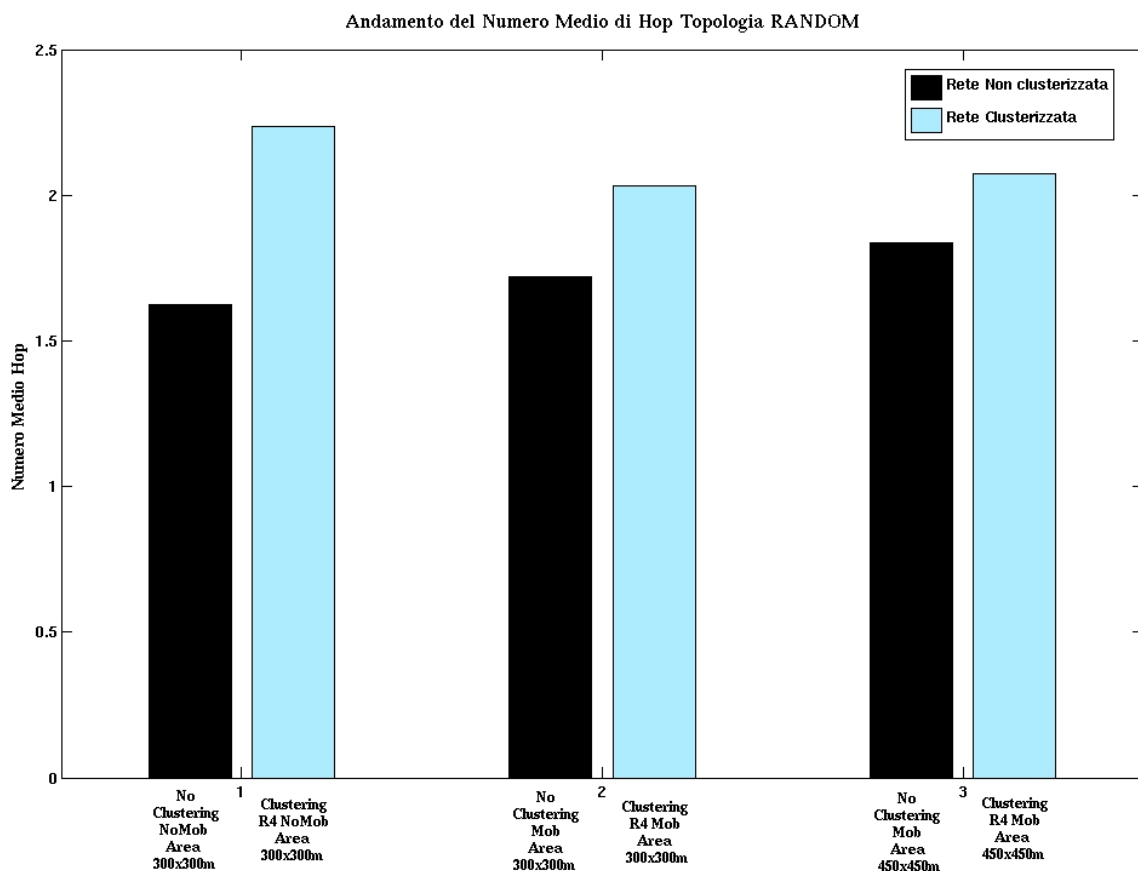
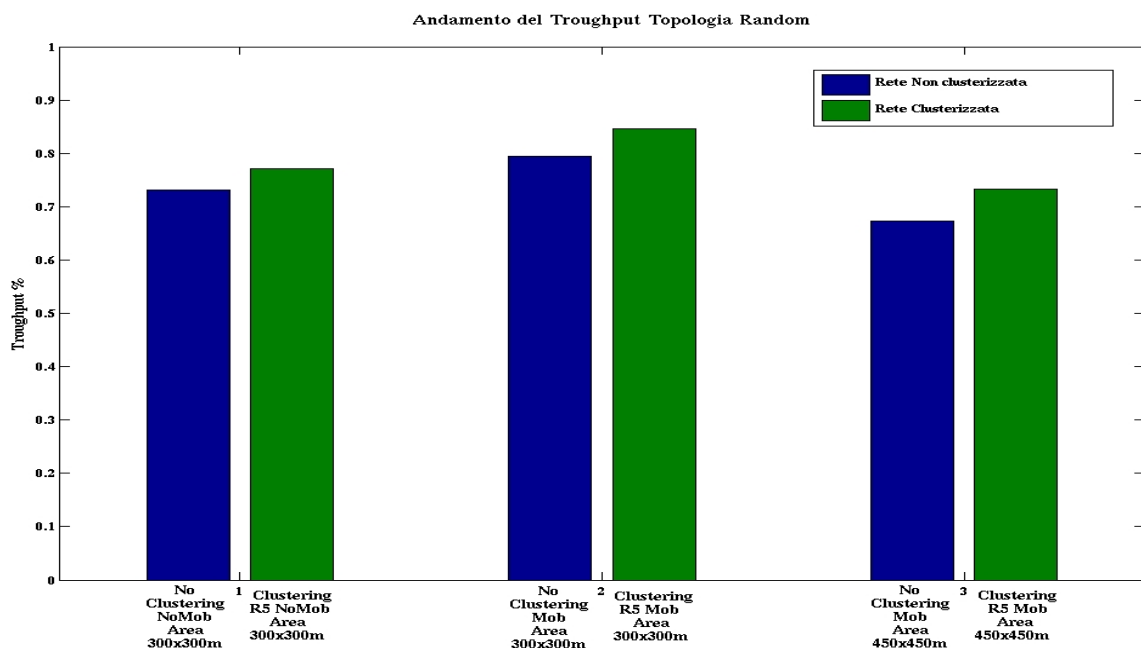


Fig 5.7: Andamento del numero medio di Hop in Topologia R4.

dall'analisi dei valori riportati si può notare come nel primo caso, senza mobilità, in una situazione più rigida, aumenta il numero di hop medio in modo più evidente rispetto agli altri; negli altri due casi, invece, grazie alla variazione della posizione dei nodi ed al raggio d'azione più ristretto, il numero di hop medio aumenta in modo più misurato.

Nel secondo slot di simulazioni sono state analizzate le prestazioni del sistema in clusterizzazione R5, nelle stesse condizioni di assenza e presenza di mobilità e variazione delle dimensioni dell'area di interesse che hanno caratterizzato lo slot precedente; in questo caso, però, la differenza rispetto all'algorithm precedente è data dalla scelta preventiva di quattro nodi-pozzo (Host [0], [4], [8], [12]) come unici apparati riceventi, verso i quali trasmettono gruppi distinti e prescelti di tre nodi.

Dal punto di vista del throughput (Figura 5.8) si può notare come, anche in questo caso, attuando l'algorithm di clusterizzazione, il sistema tragga dei benefici:

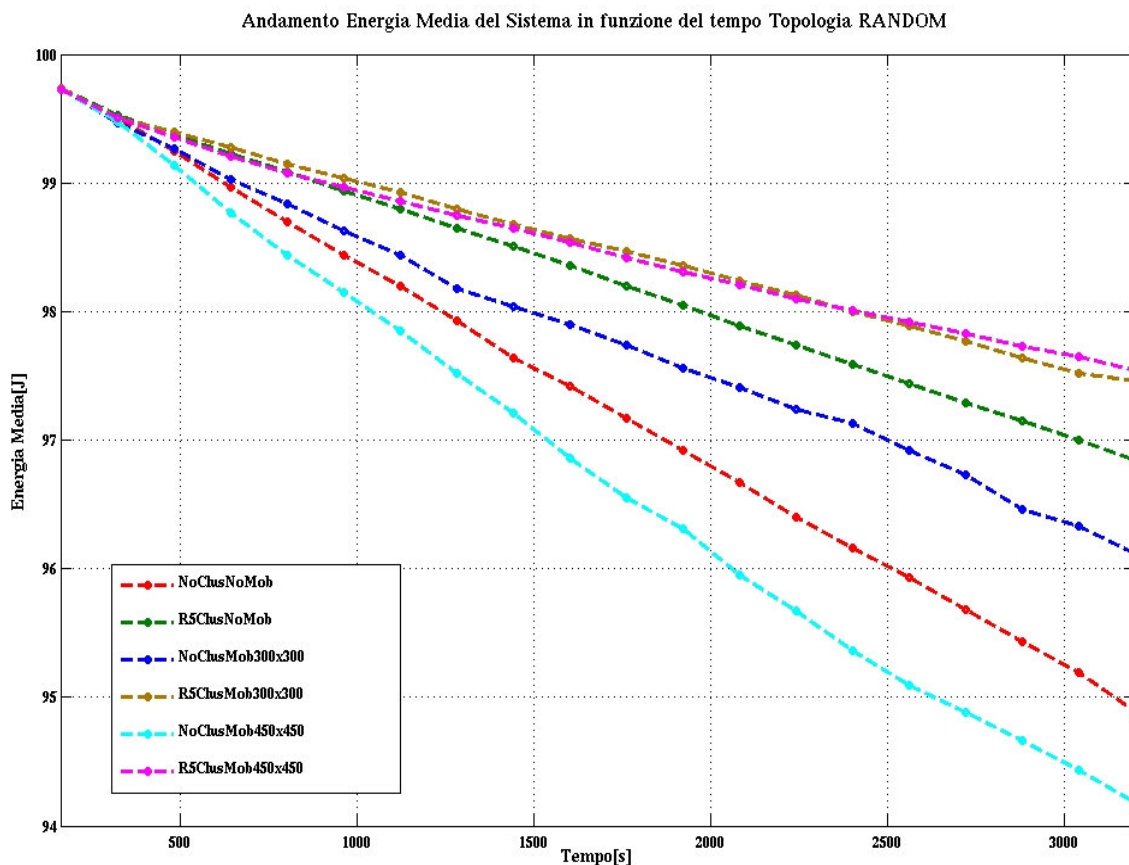


**Fig 5.8: Andamento del Troughput in Clusterizzazione R5.**

il maggiore incremento si ottiene con l'aggiunta della mobilità, (indipendentemente dall'area scelta per lo sviluppo del sistema) e può dipendere dal fatto che i nodi,

variando la loro posizione, non intaccano comunque la preventiva creazione dei clusters e quindi lo scambio di informazioni all'interno di essi.

Anche dal punto di vista energetico si possono notare, in tutti e tre i casi, prestazioni migliori per un minor consumo atto ad allungare il tempo di vita del sistema (Figura 5.9):

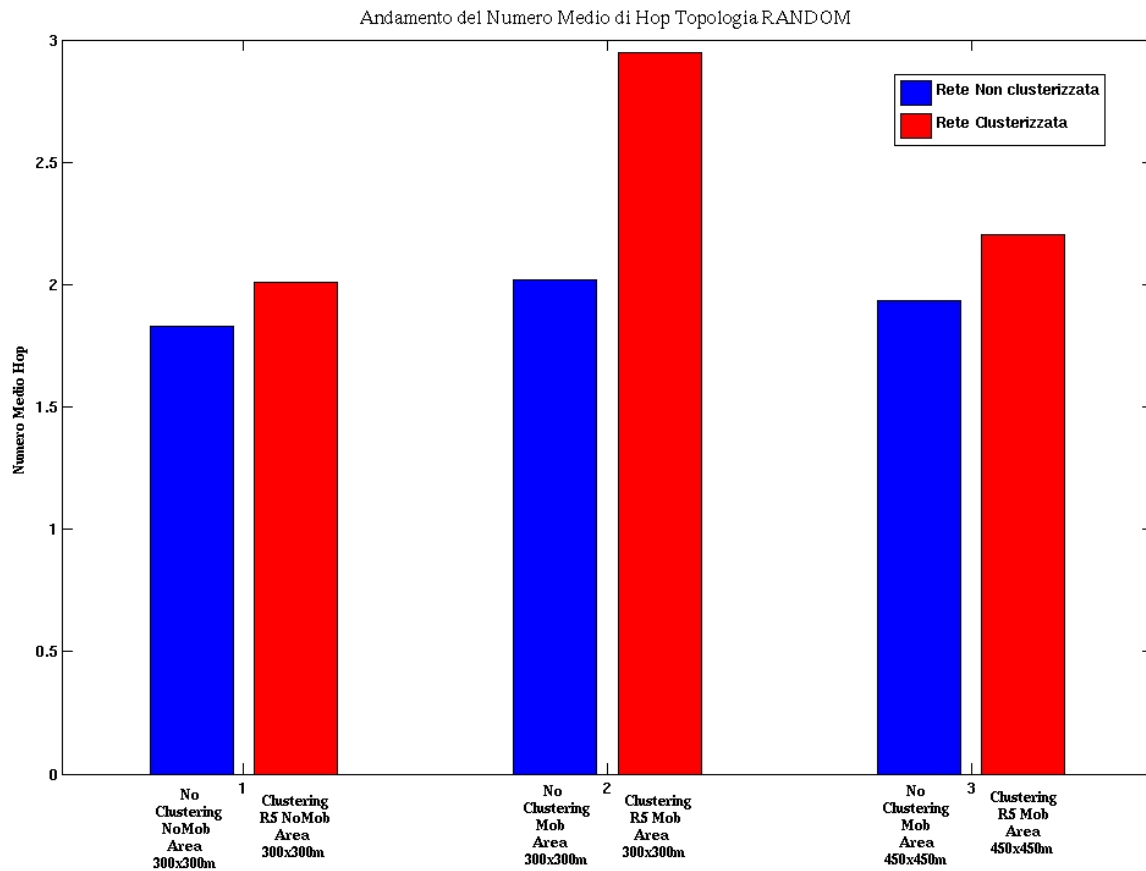


**Fig. 5.9: Andamento dell'Energia in funzione del Tempo in Clustering R5.**

come si poteva prevedere il maggior consumo si ha con mobilità in assenza di clusterizzazione, ma in questo caso si ottiene anche un miglioramento più grande rispetto alle altre topologie, nelle quali il decremento appare comunque abbastanza sostenuto.

Per quanto concerne il numero medio di hop, infine, in Figura 5.10 è evidenziato come, proprio come accadeva nel caso precedente, applicando la clusterizzazione aumenti il numero medio di hop. La struttura rigida di questa rete altera in qualche

modo il cammino dei messaggi scambiati a vantaggio della clusterizzazione.

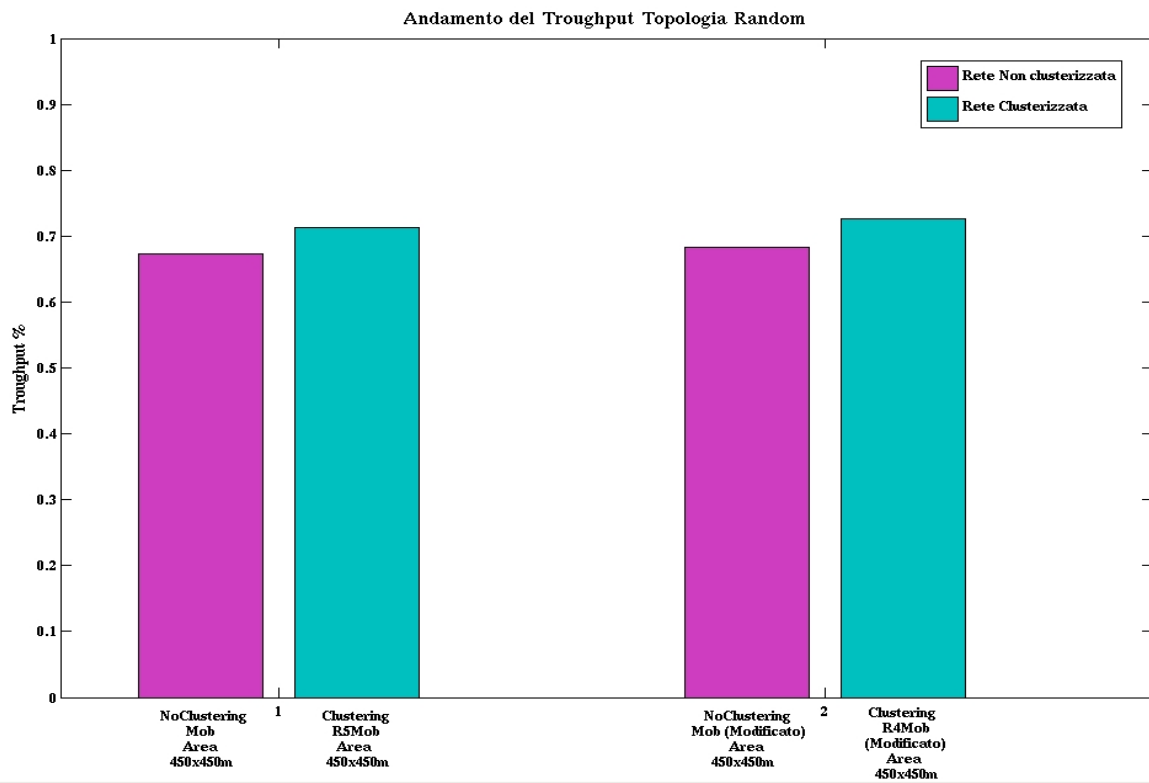


**Fig. 5.10: Andamento del Numero di Hop medio in Topologia R5.**

Nel terzo slot di simulazioni sono messe a confronto le prestazioni del sistema con e senza clusterizzazione, in topologia R4 ed R5, invertendo i modi in cui si considera lo scambio dei messaggi tra i vari apparati: si fanno ricevere tutti i nodi in topologia R5, mentre in R4 si inseriscono gli Host pozzo.

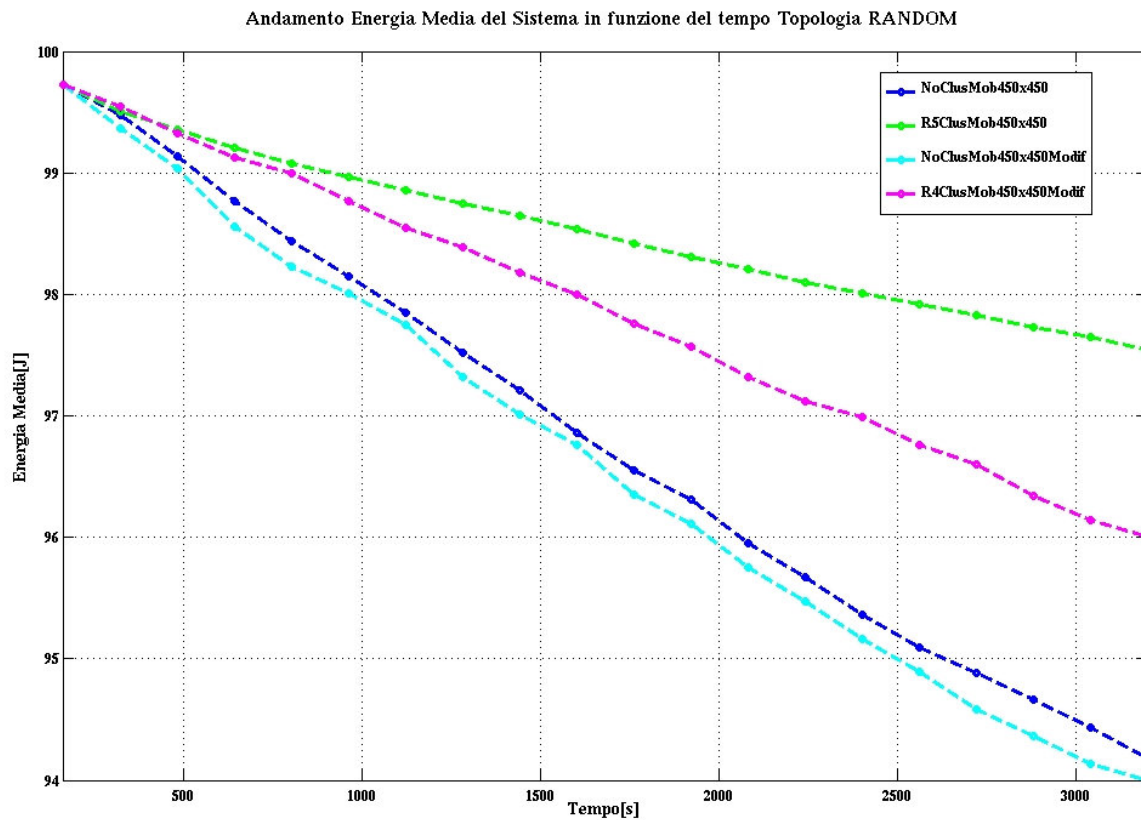
Dal punto di vista del throughput si può notare come la clusterizzazione apporti sempre miglioramenti e nel caso modificato dia comunque una buona prestazione dell'intero sistema. Analizzando i valori numerici, infatti, si ha un incremento quasi uguale di punti percentuali.





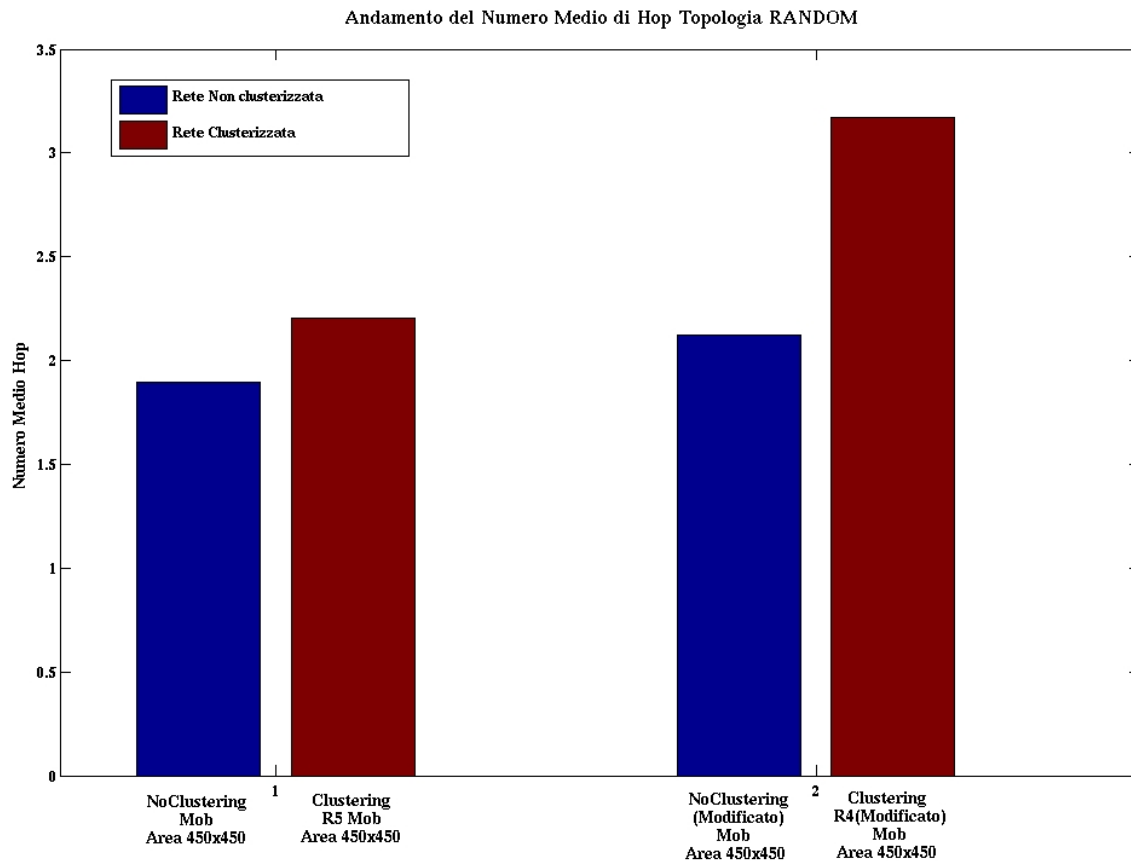
**Fig. 5.11: Andamento del Troughput in Topologie R5 ed R4 modificata.**

Dal punto di vista energetico si può notare come entrambi gli scenari non clusterizzati abbiano un andamento quasi simile, mentre il miglioramento maggiore si ha applicando la clusterizzazione R5 (come mostrato in Figura 5.12).



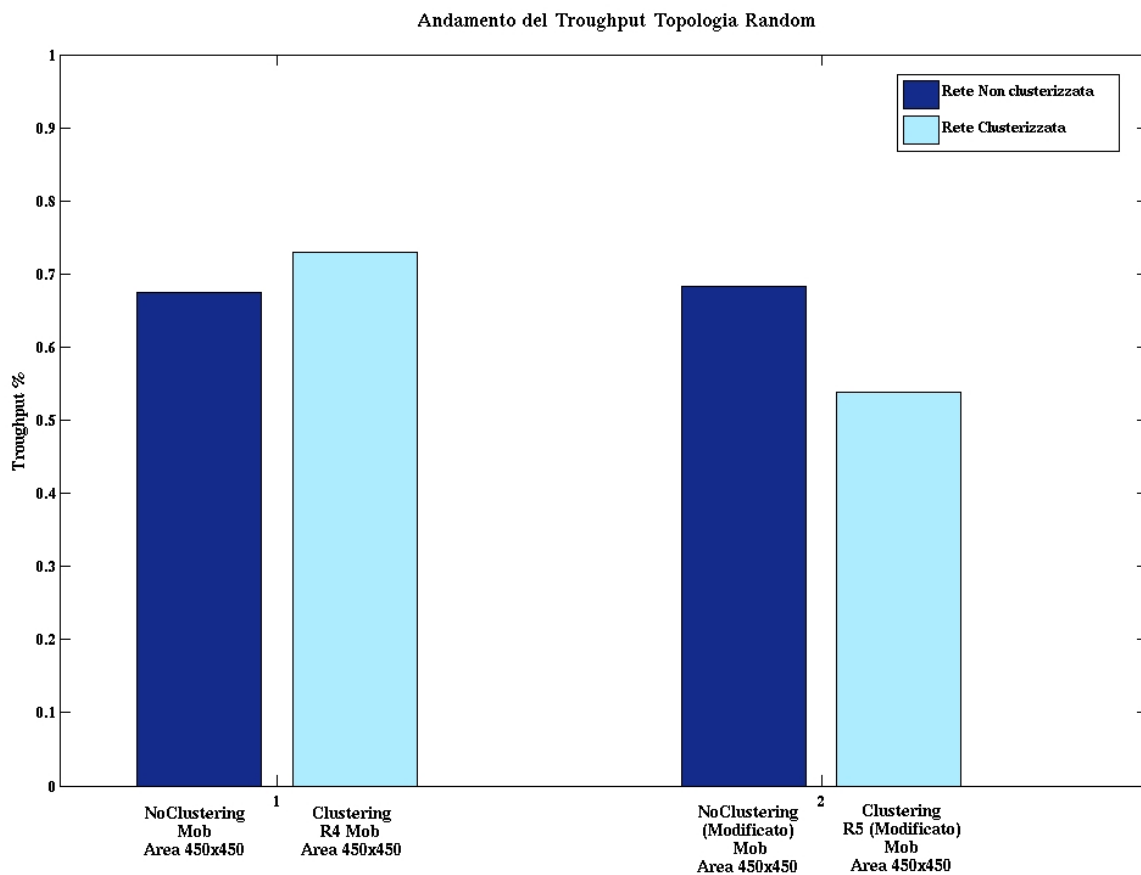
**Fig. 5.12: Andamento Energetico in Topologia R5 ed R4 modificata.**

Per quanto concerne il numero medio di hop, infine, (come mostrato in Figura 5.13), così come era accaduto per le simulazioni precedenti, anche in questo caso si ha un incremento conseguente all'applicazione della clusterizzazione, giustificabile con le stesse motivazioni precedentemente descritte.



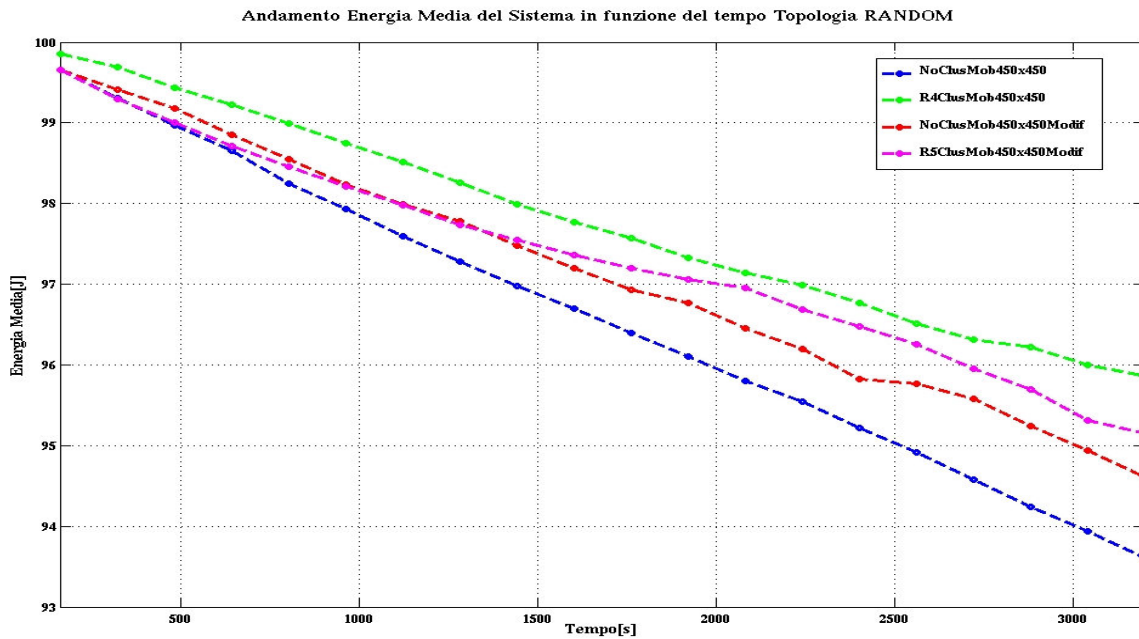
**Fig. 5.13: Andamento del Numero Medio di Hop in topologia R5 ed R4 modificata.**

Per quanto concerne l'ultimo slot di simulazioni, sono stati analizzati i valori ottenuti dall'inversione del modo di clusterizzazione rispetto al caso precedente; dal punto di vista del throughput (Figura 5.14) si può notare come l'algoritmo R5 non dia più risultati soddisfacenti ma introduca un peggioramento rispetto al caso non clusterizzato: senza la scelta dei nodi pozzo, infatti, le possibili connessioni tra tutti i nodi portano ad una maggiore perdita dei pacchetti durante la loro trasmissione e propagazione.



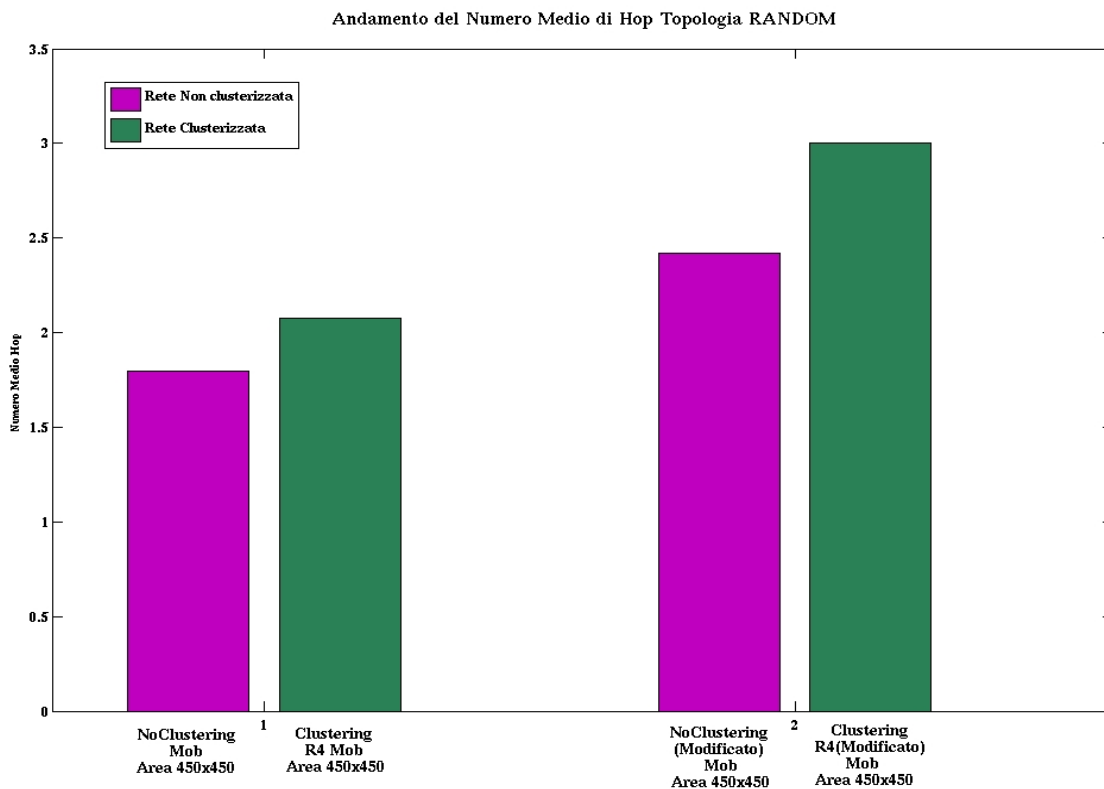
**Fig. 5.14: Andamento del Troughput in Topologia R4 ed R5 modificata.**

Dal punto di vista energetico, invece, come si può notare dall'analisi della Figura 5.15, la clusterizzazione apporta dei miglioramenti in entrambi i casi, ma in topologia R5 modificata, al di fuori di un iniziale andamento simile, non si hanno grandi benefici; questo problema è strettamente legato, infatti, all'andamento del throughput analizzato in precedenza.



**Fig. 5.15: Andamento dell'Energia in funzione del Tempo in Topologia R4 ed R5 modificata.**

Per quanto concerne il numero medio di hop (come mostrato in Figura 5.16), infine, nonostante i peggioramenti introdotti dal clustering R5 modificato, si ha un aumento del numero di hop:



**Fig 5.16: Andamento del Numero Medio di Hop in topologia R4 ed R5 modificata.** rispetto agli altri casi, infatti, la clusterizzazione peggiora le prestazioni del sistema,

contribuendo ulteriormente ad aumentare il numero medio di hop per la comunicazione tra i nodi che compongono l'ambiente di simulazione.

Successivamente alle analisi introdotte fin ora e strettamente legato al risparmio energetico che si ottiene nelle due diverse topologie di rete, è interessante notare anche in questo caso come la clusterizzazione aumenti il tempo di vita dell'intero sistema (Figura 5.17) per tutti e due gli algoritmi analizzati, R4 ed R5, sia in mobilità che in assenza di mobilità: è importante notare come in mobilità non ci sia un maggiore dispendio energetico (anche perchè le analisi son fatte in base ai messaggi scambiati e non all'energia impiegata per far muovere i terminali), ma un miglioramento che riguarda l'intero sistema.

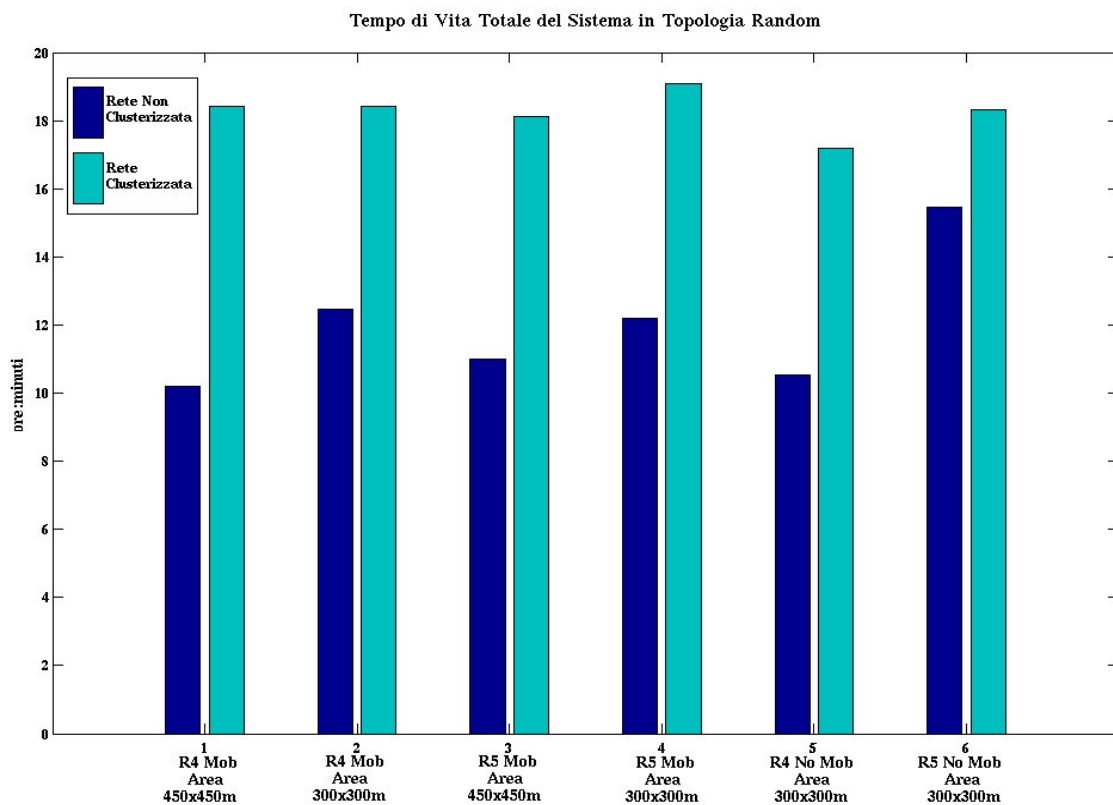


Fig 5.17: Tempo di Vita del Sistema in Topologia RANDOM in diverse specifiche di sistema.

## CONCLUSIONI FINALI

Lo scopo principale di questo lavoro è stato quello di applicare diversi algoritmi di clusterizzazione a due differenti scenari: uno con topologia GRID (più rigido) ed un altro con topologia RANDOM (più flessibile).

Nella parte iniziale si è introdotto l'ambiente su cui operavano i vari apparati ricetrasmittenti e gli standard utilizzati, cercando di analizzare le loro caratteristiche principali e di introdurre il concetto fondamentale su cui si basano tutti i discorsi affrontati: i miglioramenti sulle prestazioni del sistema che un algoritmo di clusterizzazione può apportare.

Nel Capitolo 5 sono stati analizzati tutti i risultati delle simulazioni e si è giunti alla conclusione che l'applicazione del clustering in una rete ad hoc caratterizzata dalle specifiche descritte nei capitoli introduttivi porta dei miglioramenti soddisfacenti dal punto di vista del throughput e delle buonissime prestazioni dal punto di vista energetico.

Le applicazioni di tali sistemi possono essere diverse:

- *apparati per lo studio di fenomeni atmosferici*, in cui il risparmio energetico è basilare per il loro posizionamento (a volte) in ambienti non facilmente raggiungibili (appartenenti alla topologia GRID o Random senza mobilità);
- *dispositivi di sicurezza*, in cui, con l'aggiunta di algoritmi di posizionamento, si possono studiare le condizioni in cui devono operare, ad esempio, forze dell'ordine o pompieri in caso di necessità (topologia RANDOM con mobilità);
- *dispositivi di monitoring*, importanti per poter monitorare un ambiente per lungo periodo, senza dover intervenire assiduamente sugli apparecchi ricetrasmittenti (topologia GRID e RANDOM).

Una possibile evoluzione può essere quella di migliorare le prestazioni del sistema aggiungendo degli algoritmi per la correzione degli errori: in questo caso, infatti, si assume la condizione più pessimistica possibile per cui anche solo un bit errato corrompe il pacchetto informativo, portando ad un conseguente scarto di esso. Con

l'aggiunta di codici FEC per la correzione degli errori, ad esempio, si possono ottenere miglioramenti nel numero di pacchetti persi con un conseguente aumento percentuale del throughput e quindi del consumo totale energetico del sistema.

## **APPENDICE A**

OMNeT++ è un simulatore di rete modulare orientato agli oggetti basato su eventi discreti, usato soprattutto per studiare i comportamenti di svariate topologie di rete.

Tale simulatore può essere utilizzato per modellare:

1. traffico in reti di telecomunicazioni;
2. protocolli di rete;
3. reti di code;
4. qualsiasi altro sistema che sia basato su un approccio ad eventi discreti.

Un modello di simulazione in OMNeT++ consiste in un insieme di moduli annidati



gerarchicamente. La profondità dell'annidamento dei moduli non è limitata, il che permette di riflettere la struttura logica dei modelli di comunicazione reali.

I moduli comunicano tramite scambio di messaggi che possono contenere anche una struttura dati complessa.

I moduli, inoltre, possono inviare i messaggi sia direttamente al loro destinatario, sia attraverso un percorso predefinito, tramite gates e connessioni; possono avere dei parametri propri che possono essere utilizzati per caratterizzare il comportamento del modulo e per parametrizzare la topologia dei moduli. I moduli che si trovano nel livello più basso della gerarchia incapsulano il comportamento dei moduli a livello più alto e vengono detti *simple modules*: essi sono programmati in C++ utilizzando un'appropriata libreria di simulazione.

## A.1 La gerarchia dei moduli e Module Types

Un modello OMNeT++ consiste in un insieme di moduli annidati gerarchicamente che comunicano tramite scambio di messaggi l'uno con l'altro. Generalmente si fa riferimento ai modelli OMNeT++ con il nome di *networks*. Il modulo di più alto livello è il *system module*: esso contiene diversi *submodules*, che possono a loro volta contenere altri sottomoduli (Figura A.1).

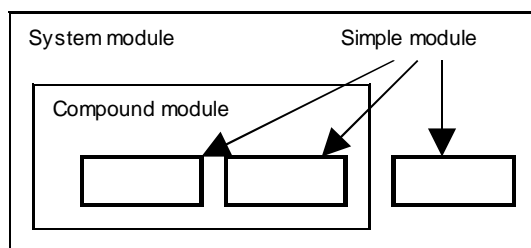


Figura A.1: Simple e Compound module

La struttura della rete è descritta tramite il linguaggio NED.

I moduli che contengono dei sottomoduli sono denominati *compound modules*, mentre i moduli che formano il livello più basso della gerarchia sono denominati *simple modules*. Gli utenti implementano gli algoritmi di rete nei *simple modules* utilizzando il linguaggio di programmazione C++ e la libreria di simulazione di OMNeT++.

I moduli semplici (*simple*) ed anche i moduli composti (*compound*) sono istanze di *modules types*. Durante la descrizione del modello di rete, il programmatore definisce i tipi di moduli che saranno utilizzati, ed delle istanze di questi tipi saranno utilizzate per costruire dei tipi di moduli più complessi. Inoltre il programmatore può costruire

il modulo di sistema complessivo, come insieme di istanze dei tipi di moduli definiti; tutti i moduli della rete sono istanziati come sottomoduli del modulo complessivo del sistema.

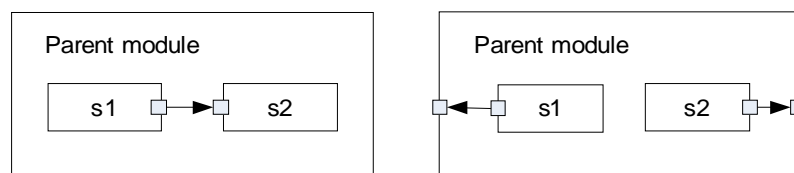
Quando un tipo è istanziato per costruire un modulo non vi è distinzione se il modulo è semplice o composto: questo consente al programmatore di separare un modulo in diversi moduli semplici e aggregarli all'interno di un modulo composto, oppure, viceversa, di aggregare le funzionalità di un modulo composto in un unico modulo semplice. I tipi di moduli definiti possono essere memorizzati in un file diverso da quello in cui sono utilizzati e ciò consente di creare vere e proprie librerie di componenti.

## **A.2 Messaggi, gates, links, parametri e trasmissione dei pacchetti**

I moduli comunicano tramite scambio di *messaggi*, i quali possono contenere strutture dati complesse.

Il “*local simulation time*” di un modulo avanza quando il modulo riceve un messaggio; il messaggio può provenire dal modulo stesso o da un altro modulo (i *self-messages* sono utilizzati per implementare i timer). I *gates* rappresentano le interfacce di ingresso e d’uscita dei moduli, i messaggi sono inviati tramite i gates di output e sono ricevuti tramite i gates di input.

Ogni connessione (detta anche *link*), è generata all’interno di un singolo livello della gerarchia dei moduli: all’interno di un modulo composto si possono connettere i gates di due sottomoduli o il gate di uscita di un sottomodulo e il gate di ingresso del modulo composto (Figura A.2).



**Figura A.2: Connessioni**

A causa della struttura gerarchica del modello, i messaggi attraversano una serie di connessioni per transitare da un modulo semplice ad un altro. I moduli composti possono essere visti come delle scatole nere che consegnano i messaggi in modo trasparente tra il mondo interno ed esterno.

Alle connessioni possono essere assegnati tre parametri opzionali, che facilitano la modellazione delle reti di comunicazione: *propagation delay*, *bit error rate*, *data rate*. Si possono specificare i parametri delle connessioni individualmente per ciascuna connessione, oppure si possono definire dei tipi di links che possono essere poi utilizzati in tutto il modello.

I moduli possono avere dei propri parametri che possono essere assegnati nei files NED o nel file di configurazione omnetpp.ini.

I parametri possono essere utilizzati per caratterizzare i moduli semplici, o per parametrizzare la topologia. Essi possono essere di tipo *string*, *numeric*, *boolean*. I

parametri di tipo numeric possono essere utilizzati per costruire le topologie in modo flessibile. All'interno di un modulo composto i parametri possono essere utilizzati per definire il numero di sottomoduli, il numero di gates, ed il modo con cui sono generate le connessioni interne.

## **A.3 Descrizione della topologia ed uso di OMNet++**

In OMNeT++ il programmatore può definire la topologia della rete che vuole costruire utilizzando il linguaggio di descrizione NED (Network Description). Il linguaggio NED facilita la descrizione modulare della rete, permettendo di descrivere la rete come un insieme di componenti (canali, simple/compound module types). I files che contengono la descrizione della rete hanno il suffisso *.ned*, e sono traslati in linguaggio C++ dal compilatore NED.

Per maggiori dettagli sul linguaggio NED è possibile far riferimento al Capitolo 3 di [26].

Si può descrivere a grandi linee, inoltre, come lavora OMNet++ dal punto di vista pratico.

Un modello OMNeT++ è formato dai seguenti componenti:

- la descrizione della topologia in linguaggio NED (*.ned* files), che descrivono la struttura dei moduli, i parametri ed i gates.
- la definizione dei messaggi (*.msg* files): si possono definire vari tipi di messaggi, che saranno traslati da OMNet++ in classi C++.
- i codici sorgenti dei moduli semplici: questi sono files C++ con estensioni *.cc* e *.h*.

Il sistema di simulazione fornisce invece i seguenti componenti:

- il kernel di simulazione, che contiene il codice per la gestione della simulazione: esso è scritto in C++.
- le interfacce utente, che possono essere utilizzate per effettuare il debug durante la simulazione, per rendere visibile all'utente l'interno della simulazione, e per modificare i parametri a tempo di esecuzione.

Il programma di simulazione viene costruito a partire dai componenti precedenti. Per primo si traslano i files *.msg* in classi C++, utilizzando il compilatore di messaggi *opp\_msgc*, quindi si compilano tutte le altre classi C++, e si collegano al kernel di simulazione. I files NED possono essere traslati in C++ utilizzando il compilatore di files NED, *nedtool*.

In conclusione si può affermare che le informazioni date in tale paragrafo su OMNeT++ sono solo sommarie e volte a fornire una panoramica generale su tale tool di sviluppo. La costruzione di un simulatore in OMNeT++ richiede una più elevata conoscenza dello strumento e uno studio dettagliato delle sue caratteristiche; per una descrizione più completa è possibile far riferimento a [26] ed al sito ufficiale di OMNeT++ [25].

## BIBLIOGRAFIA

- [1] ZigBee Standards Organization, “ZigBee Specification”, ZigBee Document 053474r06 74r06, Version 1.0, 2005.
- [2] Patrick Kinney, “ZigBee Technology: Wireless Control that Simply Works”, Communications Design Conference, 2 October 2003.
- [3] IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs), 7 Marzo 2006.
- [4] IEEE P802.15 Task Group 4a for Wireless Personal Area Networks (WPANs),” 802.15.4a Dallas Plenary Meeting Minutes”, Minutes of Task Group 4a in Dallas, 18 Novembre 2006.
- [5] M.G. Di Benedetto, L. De Nardis, G. Giancola and D. Domenicali, “The Aloha access (UWB)<sup>2</sup> revisited for IEEE 802.15.4a”, to appear in ST Journal of Research, <http://www.st.com/stonline/press/magazine/stjournal/index.htm>, 2006.
- [6] [www.distar.it](http://www.distar.it), “ZigBee Wireless Communications Facile con Freescale”, Distar Focus Magazine, freescale semiconductor, n°13-GIU 2006 [df13@distar.it](mailto:df13@distar.it).
- [7] Università degli Studi di Siena, “Le Tecniche di Comunicazione Wireless”, Moduli professionalizzanti, Corso di reti di Telecomunicazioni, 2006.
- [8] M. G. Di Benedetto and G. Giancola, Understanding Ultra Wide Band Radio Fundamentals. Prentice Hall, 2004.
- [9] M. G. Di Benedetto and B. R. Vojcic, “Ultra Wide Band (UWB) Wireless Communications: A Tutorial,” Journal of Communication and Networks, Special Issue on Ultra-Wideband Communications, vol. 5, no. 4, pp. 290–302, December

2003.

[10] M. L. Welborn, "System considerations for ultrawideband wireless networks," in *IEEE Radio and Wireless Conference*, August 2001, pp. 5–8.

[11] I. Guvenc and H. Arslan, "On the modulation options for UWB Systems," in *IEEE Military Communications Conference*, vol. 2, October 2003, pp. 892–897.

[12] I. Rossini, "Clustering", Corso di *Metodologie per Sistemi Intelligenti*, Lezione2, Politecnico di Milano, 2006.

[13] E. Mantovani, "Clustering e Vector Quantization", Corso di *Sistemi di Telecomunicazioni*, Lezione6, Facoltà di Ingegneria di Genova, 2006.

[14] Cabrera, J.B.D., Gutierrez C. and Mehra R.K., "Infrastructures and algorithms for distributed anomaly-based intrusion detection in mobile ad-hoc networks", [Military Communications Conference, 2005. MILCOM 2005. IEEE](#) 17-20 Oct. 2005 Page(s):1831 - 1837 Vol. 3.

[15] Yanlei Shang and Shiduan Cheng, "A stable clustering formation in mobile ad hoc network", [Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on](#) Volume 2, 23-26 Sept. 2005 Page(s): 714 – 718.

[16] Ting-Chao Hou and Tzu-Jane Tsai, "An Access-Based Clustering Protocol for Multihop Wireless Ad Hoc Networks", *IEEE Journal on Selected Areas in Communicatios*, Vol. 19, NO.7, July 2001.

[17] P.Basu, N.Khan, and D.C.Little, "A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks", *Workshop on Distributed Computing Systems*, 2001, pp. 413 –418.

[18] S.Sivavakeesar, G.Pavlou, A.Liotta, "Stable Clustering Through Mobility Prediction for Large-Scale Multihop Intelligent Ad Hoc Network", *IEEE Wireless Communications and Networking Conference (WCNC'04)*, (2004) 1488–1493.

[19] B.McDonald, and F.Znati, "A Mobility-Based Framework for AdaptiveClustering in Wireless Ad Hoc Networks", *IEEE Journal on Selected Areas in Communications*, vol. 17, August 1999, pp 1466 –1487.

[20] V. Shah, H. Deng and D. P. Agrawal, "Parallel Cluster Formation for Secured Communication in Wireless Ad Hoc Networks", [Networks, 2004. \(ICON 2004\)](#).



[Proceedings. 12th IEEE International Conference on](#) Volume 2, 16-19 Nov. 2004  
Page(s):475 - 479 vol.2 .

[21] A. M. Yip, C. Ding and T. F. Chan, “Dynamic Cluster Formation Using Level Set Methods”, IEEE TRANSACTIONS OF PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 28, NO. 6, JUNE 2006.

[22] Mohammed S. Al-kahtani and Hussein T. Mouftah, “A Stable Clustering Formation infrastructure Protocol in Mobile Ad Hoc Networks”, [Wireless And Mobile Computing, Networking And Communications, 2005. \(WiMob'2005\), IEEE International Conference on](#) Volume 3, 22-24 Aug. 2005 Page(s):406 - 413 Vol. 3.

[23] Chengfa Li, Mao Ye, Guihai Chen and Jie Wu, “An energy-efficient unequal clustering mechanism for wireless sensor networks”, [Mobile Ad hoc and Sensor Systems Conference, 2005. IEEE International Conference on](#) 7-10 Nov. 2005  
Page(s):8 pp.

[24] T. Voigt, A. Dunkels, J. Alonso, H. Ritter and J. Schiller, “Solar-aware Clustering in Wireless Sensor Networks”, Conference in Swedish Institute of Computer Science (SICS), Stoccolma, 2005.

[25] <http://www.omnetpp.org> .

[26] Varga A., OMNeT++ User Manual version 3.2, disponibile a <http://www.omnetpp.org>.

[27] <http://www.cs.unibo.it/~concer/>.

## Ringraziamenti

*E' dal primo giorno in cui sono entrato nel laboratorio ACTS per iniziare a lavorare alla mia tesi che penso a quanto sarebbe stato bello, ed allo stesso tempo difficile, scrivere queste ultime pagine!!*

*La paura più grande è quella di dimenticare qualcuno, ma in fondo al mio cuore le persone importanti rimarranno indelebili per sempre.*

*Vorrei ringraziare tutte le persone che mi hanno circondato in questi anni e che per un motivo o per un altro hanno contribuito a cambiarmi ed a rendermi quello che*

*sono.....l'ordine che seguirò è logico, non d'importanza, perchè ogni singola persona ha un ruolo speciale.....*

*Volevo dire grazie a.....*

*Alla Prof. Di Benedetto, che mi ha dato la possibilità di fare questa fantastica esperienza!per tutti i consigli preziosi che mi ha dato in ambito professionale, per l'esempio di dedizione al lavoro ed allo stesso tempo alla famiglia che tutti i giorni mi ha mostrato!per tutti i racconti sulle sue esperienze (traumatiche o meno) all'estero, i sui viaggi, i suoi corsi di Yoga che hanno contribuito a distrarmi e staccare dal lavoro anche nei momenti più stressanti.*

*A Luca,.....che dire!!nonostante tutte le battute e gli scherzi, è stato davvero il migliore correlatore che si potesse mai avere!sempre pronto a darmi una mano nelle pause tra le sue diecimila scadenze, sempre disponibile, col sorriso sulle labbra!per tutti i suoi mesi in California sono stato la prima persona che sentiva quando si svegliava, prima di sua moglie, e nonostante questo mi ha sempre risposto con un "Buongiorno!" piuttosto che con un più sensato "Che altro vuoi da me!!", riuscendo sempre a risolvere tutti i miei problemi!Compagno nerazzurro di vittorie e sconfitte, resterà sempre per me l'icona del "Genio Compreso"!!!!*

*A Dome, una persona eccezionale!anche se non impegnato nel mio progetto, mi ha sempre aiutato, spronato e consolato nei momenti di crisi!le sue battute in toscano son state sempre il momento più comico della giornata (da qui la nascita del "Dome docet..!")!Grande sostenitore di Ubuntu è sempre riuscito a rallegrarmi la giornata, facendomi notare quanto fosse maggiore la sua mole di lavoro rispetto alla mia!:-)*

*A Mauro, compagno di mille avventure, collega universitario per tutti questi anni, amico fraterno, sempre disponibile, sincero, pronto ad aiutarmi!senza di lui (e dei suoi gentilissimi genitori che mi hanno fatto sempre sentire a casa, trattandomi come un figlio!) non so se sarei riuscito in questa impresa!!per tutte le serate passate a divertirci, per quelle passate a studiare, per le risate nei giorni tranquilli e le litigate in quelle tesi!!!per tutte le partite alla play nel segno dell'anti-fairplay, per gli sfottò sulle nostre diverse fedi calcistiche!!abbiamo lottato fino all'ultimo giorno per questa tesi, ci siamo aiutati, spronati ed alla fine ci siamo riusciti!!*

*Ai tesisti dell'ACTS (Moulay, Luis, che mi ha insegnato lo spagnolo..., e Giorgio), ai mitici TaralloBoys e tutte le persone della mailing List Calcetto, perchè quelle serate in un campo da gioco son sempre servite a scaricare stress e stanchezza accumulati durante il lavoro!*

*Ai miei Genitori, le persone senza le quali non sarei mai diventato quello che sono, sempre attenti, affettuosi, disponibili!!hanno sempre creduto in me, non mi hanno*

*mai fatto mancare niente e non mi hanno fatto mai sentire solo!mi hanno dato la migliore educazione che si potesse mai avere e che spero, un giorno, di tramandare ai miei figli!!*

*A Nonna, la mia seconda madre, la persona più saggia e colta che conosca; la cuoca migliore del mondo!!l'esempio vivente che la forza di volontà e la determinazione possono tutto!si è sempre interessata a me ed alle mie attività; i suoi pranzi domenicali son sempre stati motivo di gioia e allegria!*

*Ad Angela e Piera, due sorelle fantastiche!!mi han sempre fatto sentire il loro affetto, in un modo o nell'altro,:-), mi han sempre coccolato e viziato, preoccupandosi per ogni mio problema, ogni mia incertezza, sempre pronte e disposte a darmi il loro totale aiuto!!!Le mie mamme quando ero lonato da casa, le mie amiche quando avevo problemi.....grazie di cuore!!!*

*A Massimo, il fratello che ho sempre voluto, la persona più educata ed affettuosa che abbia mai conosciuto!tutti i momenti con lui son stati indimenticabili, mi ha sempre trattato come un vero fratellino e si è sempre fatto in quattro per me!*

*A Nicoletta, la mia tata di sempre!mi ha sempre coccolato e viziato fin da quando son nato!le sue omelette mi hanno nutrito per anni e mi hanno dato la forza per fare qualsiasi cosa!!*

*A zio Dino, il mio "Padrino" e padre adottivo!la persona che mi ha dato la possibilità di vivere a Roma con "il privilegio" di avere sempre tutto quello di cui avevo bisogno!I suoi metodi non sono sempre stati il massimo, :-), ma a modo suo non mi ha mai fatto mancare il suo grandissimo affetto!!!mi ha fatto capire tante cose della vita e non dimenticherò mai tutti i suoi preziosissimi insegnamenti.*

*A zia Cinzia, Vale, Pietro e Carlotta, una mamma sempre premurosa e affettuosa la prima, dei cugini fraterni su cui poter sempre contare gli altri!!!spero di lasciare un buon ricordo della permanenza a casa vostra!!*

*Ad Alice, la mia sorellina Romana, la mia migliore amica, la mia confidente....!!per tutti i discorsi e i progetti fatti insieme, le chiacchierate sui nostri problemi alla fine di interminabili giornate di studio, le risate, i pianti....mi mancheranno tanto!!*

*A zio Franco e Zia Rosella, altri due veri e propri genitori nel corso della mia vita, sempre presenti e disponibili nei momenti importanti!!anche in questo giorno avete fatto di tutto per esser qui e non lo dimenticherò mai!!*

*A Riccardo, compagno di mille avventure, amico e cugino fidato, compagno di sventure (neroazzurre) ma sempre con allegria; Nicola, il cugino Inglese, Luisa e Claudia, sempre affettuose e disponibili in qualsiasi momento!!*

*A Francesca, perchè in questi mesi non mi ha mai fatto mancare il suo affetto, per avermi supportato e sopportato con tanta pazienza, per tutte le sere che si è fatta 50 Km da sola soltanto per coccolarmi e consolarmi!! Per avermi fatto credere in qualcosa che pensavo non esistesse più, per avermi fatto ridere anche quando ero triste, avermi calmato quando ero teso, avermi capito anche quando i miei comportamenti erano incomprensibili!!! non dimenticherò mai quel che hai fatto per me.....!!*

*A Eddy e Rita, perchè anche loro mi hanno aiutato nella vita di tutti i giorni ed han sempre fatto di tutto per farmi stare bene.*

*Agli amici di Castellaneta, compagni di giochi prima e consiglieri fidati poi, a Giuseppe e Rossella su tutti (senza sminuire l'affetto degli altri) perchè sono due persone fantastiche che mi son sempre state vicine e si son sempre fatte in quattro per me!!*

*Agli amici di università, la mia famiglia in questi sei anni!!! per tutte le serate folli passate insieme, tutte le feste, le passeggiate, i compleanni, i tornei di Play... siete gli amici che tutti sognano di avere!! A Daniele F., Massy P., Paolo, Lorenzo, Giorgia, Simone su tutti, perchè è anche grazie a loro se oggi sono qui!!!*

*Agli amici di Terlizzi, Anna, Sara, Carmine, Ornella, compagni di liceo, fondamentali e ancora punti fermi della mia vita!! sempre vicini e presenti nel mio cuore anche se lontani fisicamente!!!*

*A tutti gli amici del calcetto, perchè fare tutti i tornei che ho fatto insieme a loro è stato un piacere ed un onore!! perchè han contribuito a farmi distrarre ed a mantenermi in forma....!!!*

*Spero davvero di non aver dimenticato nessuno e mi scuso con chi si sente trascurato da questi ringraziamenti, ma non è stato facile riassumere in queste righe tutto quello che sento per le persone più importanti della mia vita.....*

*Ci tengo tanto a "ringraziare" anche componenti fondamentali di questa mia avventura durante il periodo di tesi.....*

*All'INTER, perchè dopo 18 anni ha vinto lo scudetto, sul campo, stabilendo record su record e riuscendo a farmi dimenticare quel tragico 5 Maggio Romano.....*

*A Ubuntu, perchè senza lui quando avrebbe funzionato Omnet++?!!?!!?!!*

*Alla Playstation 2 di Mauro ed all'inventore di Pro Evolution Soccer, perchè serate*

*come il torneo 2vs2 sul 42” appena comprato, il mondiale in notturna in 8, tutti con l'Italia, a Fregene, non si dimenticano facilmente.....*

*Alla macchinetta del caffè “Lavazza Methodo” del laboratorio, perchè in questi mesi ha lavorato senza sosta e non si è mai lamentata, ed allo “spacciatore di cialde” del dipartimento, perchè è sempre venuto immediatamente quando eravamo a corto di “droga”!*

*Al Sig. Rinaldo dello Spaccio, perchè i suoi panini mi hanno nutrito in tutti questi anni, perchè la “pausa Spaccio” è sempre stato un buon motivo di cazzeggio e di ricche mangiate....*

*Ai “Romanisti”, non intesi come tifosi, (si intenda!!!), ma come copisteria universitaria, per le loro fotocopie “formato esame” e per le loro battute, perchè se entri incavolato e depresso per qualcosa in un posto dove trovi i macchinari con i nomi dei giocatori della Roma e lo stemma della Lazio sulle fotocopiatrici guaste, non puoi non ridere.....*

*Al “Bar, non il Mosè, ma il secondo a destra!”, perchè la pausa caffè dopo pranzo lì è stato sempre un rito a cui non si poteva rinunciare!perchè vederci tutti i giorni seduti in dieci, occupando tre tavolini, con solo tre caffè e due cappuccini, non deve esser stato facile, ma siamo sempre stati trattati come dei signori e con il sorriso sulle labbra!!*

*A Kit, perchè se non lo facevo Daniele F. mi toglieva il saluto!*

*A FattinationalChannel, perchè mi ha riportato il sorriso in tutti i momenti tristi, di totale sconforto!! FattinationalChannel deve vivere!!!!*

*Al padre di Simone che fa il “TassiDriver” ed a tutti i suoi colleghi Valenciani che ci hanno scarrozzato in giro per la città!!!!*

*Alla RyanAir, perchè senza i suoi voli low-cost non saremmo mai andati da nessuna parte....*

*Alla “Laurea”, il nostro ristorante preferito, perchè se mangi antipasto, pizza e caffè, con vino, acqua e cocacola a soli 9 euro ed il giorno dopo sei ancora forte come un leone, perchè se hai sempre e comunque una sala riservata solo per te per i grandi eventi, non puoi non essere felice, e ad Erisa, ex cameriera, che ci dava da bere gratis per vederci movimentare la serata!!!!*