Alma Mater Studiorum · Università di Bologna

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Triennale in Informatica

Il protocollo di routing B.A.T.M.A.N.

Tesi di Laurea in Reti di Calcolatori

Relatore:
Chiar.mo Prof.
VITTORIO GHINI

Presentata da:
LUCA TOSI

Seconda Sessione
Anno Accademico 2010/11

ai miei genitori e a tutta la mia famiglia, in particolare a Redea

Indice

Elenco delle figure

Capitolo 1

Introduzione

Le Wireless Mesh Network sono una tecnologia di reti wireless che sta diventando sempre più utilizzata in ambiti cittadini o luoghi di difficili copertura. La capacità di coprire ampi spazi senza l'ausilio di numerosi access point utilizzando i nodi per propagare il segnale rendono queste reti meno costose delle attuali reti wireless e, sopravvivendo anche ai possibili guasti dei router, altrettanto stabili nella spedizione dei dati.
Il protocollo attuale che permette il routing tramite i singoli nodi della rete è un protocollo progettato ormai svariati anni fa chiamato *Optimized Link State Routing* (O.L.S.R.), pensato per risolvere le esigenze di quel tempo che si limitavano a piccole reti casalinghe o aziendali.
Al giorno d'oggi l'impiego delle Wireless Mesh Network si è sviluppato fino a essere usate per le ampie reti metropolitane, rendendo O.L.S.R. obsoleto in prestazioni, calcolo dei percorsi e utilizzo delle risorse.
La necessità di un protocollo che gestisca agilmente reti Mesh di grandi dimensioni ha portato alla nascita dell'oggetto di questa tesi: B.A.T.M.A.N., acronimo di *Better Approach To Mobile Ad-hoc Networking*.

1.1 Obbiettivo della tesi

Scopo di questa tesi è comporre un quadro generale dettagliato sul protocollo, ponendo particolare attenzione agli aspetti fondamentali del protocollo

e alla valutazione delle prestazioni nei confronti di O.L.S.R.
Saranno analizzate caratteristiche tecniche quali l'algoritmo, il formato dei pacchetti, le strutture dati e la sicurezza del protocollo in modo da capire bene quali sono i vantaggi che stanno facendo di B.A.T.M.A.N. il candidato migliore a sostituire O.L.S.R. in un futuro prossimo molto vicino.

1.2 Struttura della tesi

La tesi è suddivisa in tre capitoli principali. I capitoli 2 e 3 hanno lo scopo di fornire un contesto chiaro di utilizzo del protocollo e un accenno alla storia delle reti che ha portato allo sviluppo e all'utilizzo delle reti Mesh. Sarà analizzato il funzionamento di tale tecnologia in modo dettagliato in modo da avere le basi per comprendere al meglio l'ambiente in cui B.A.T.M.A.N. opera. Inoltre, allo scopo di classificare il protocollo nella folta foresta di quelli attuali, saranno studiate tutte le tipologie di instradamento dei pacchetti di utilizzo comune.
Il capitolo 4, invece, discuterà nello specifico di tutti gli aspetti di B.A.T.M.A.N., nell'ordine:

- i principi alla base che hanno dato vita all'idea
- il formato dei principali pacchetti utilizzati e le strutture dati necessarie al funzionamento
- l'algoritmo inteso come ragionamento vero e proprio nella scelta del percorso e il meccanismo di flooding dei messaggi necessario a consegnare a tutti i nodi le informazioni necessarie al routing
- la possibilità di dotare un nodo di più interfacce per agevolare il routing
- la sicurezza del protocollo e i possibili attacchi a cui potrebbe essere sottoposto

- una valutazione pratica delle prestazioni nei confronti del suo rivale diretto O.L.S.R., confrontando overhead, throughput, packet loss e carico di CPU e memoria
- le esperienze reali in cui B.A.T.M.A.N. è già stato reso operativo.

Capitolo 2

Tipologie di Reti

A partire dal ventesimo secolo l'uomo ha cominciato a sfruttare la potenza data dall'informazione di massa, sviluppando quelle che sono state denominate Tecnologie dell'Informazione. Sono infatti state realizzate le prime trasmissioni radio regolari, reti telefoniche, reti televisive, fino all'invenzione del primo calcolatore elettronico, in grado (all'epoca solo potenzialmente) di gestire grandi quantità di informazioni in modo automatizzato.
Con l'arrivo della Guerra Fredda e la corsa alla tecnologia spaziale è diventato essenziale lo sviluppo di un'unica tecnologia trasversale atta a integrare e spostare velocemente tutte le conoscenze che si stavano accumulando.
Fino al 1970 i sistemi di calcolo erano un limitato numero di elaboratori di grandi dimensioni con tecnologia proprietaria nei quali le periferiche si limitavano a essere stampanti, dischi e nastri rimovibili. Il solo modo per trasferire dati era spostare fisicamente nastri o dischi da un calcolatore all'altro con una visione del lavoro master/slave.
L'evoluzione dei sistemi ha portato ad abbandonare lo schema master/slave per dirigersi verso un modello di lavoro che prevede molti piccoli o medi elaboratori autonomi interconnessi tra loro, dando cosi vita alle prime reti di calcolatori.
Negli ultimi 30 anni l'impiego di reti di calcolatori ha avuto uno sviluppo esponenziale fino ad arrivare alla possibilità di creare una rete globale, ossia internet. Questo sviluppo, nella pratica, ha dato vita a molte tipologie di

reti, tra le quali le più utilizzate e quelle che interessano per creare l'ambito dell'argomento di questa tesi sono le reti cablate e le reti wireless.

2.1 Reti Cablate

Le reti cablate sono la prima tipologia di rete nata per risolvere il problema dell'interconnessione tra elaboratori. L'idea di base è stabilire la connessione tra più sistemi con l'utilizzo di un cavo fisico. Questo sistema presenta molti vantaggi che vanno dalla stabilità del collegamento (intesa come indipendente da fattori esterni), al basso costo e alle buone prestazioni, ma conservano un problema intrinsecamente irrisolvibile: la *staticità* della topologia della rete.
I Link infatti sono fisici e decisi al momento della progettazione della rete, senza la possibilità di apportare modifiche a tale topologia nel caso la rete dovesse subire cambiamenti nello schema strutturale.
L'aumento esponenziale degli utenti che fanno uso di calcolatori comporta la questione di due fondamentali problemi da risolvere: la *scalabilità*, intesa come possibilità di ampliare la grandezza e il numero di nodi che la compongono e la *dinamicità*, ossia la possibilità di apportare cambiamenti nella topologia senza dover cambiare il progetto iniziale. In questo senso si muovono buona parte degli sviluppi tecnologici nei vari campi di questo settore. I più evidenti traguardi in questo sviluppo sono il lento passaggio all'utilizzo di reti wireless che liberano i progettisti dal vincolo del cavo e la nuova versione del protocollo IP che in un futuro molto prossimo passerà da IPv4 a IPv6.

2.2 Reti Wireless

La necessità di rendere le reti dinamiche e scalabili, unite alla tendenza che negli ultimi anni ha portato a rendere gli accessori tecnologici, come palmari o notebook, portabili (utilizzabili in qualsiasi luogo) ha dato vita alle reti wireless (dall'inglese senza fili).
Lo standard più diffuso per l'implementazione di reti wireless è lo Standard

Wi-Fi 802.11, nato nel 1997 e, nel corso degli anni, aggiornato a diverse versioni, che tendono via via a migliorare le prestazioni (vedi fig. 2.1).

Standard	Frequenza	Velocità di trasferimento (Mbit/s)
802.11 legacy	FHSS, 2,4 GHz, IR	1, 2
802.11a	5,2, 5,4, 5,8 GHz	6, 9, 12, 18, 24, 36, 48, 54
802.11b	2,4 GHz	1, 2, 5.5, 11
802.11g	2,4 GHz	1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48, 54
802.11n	2,4 GHz, 5,4 GHz	1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48, 54, 125

Figura 2.1: Standard per reti wireless

La tecnologia wireless si accosta al concetto di Local Area Network (LAN), infatti appare a livello LLC (Logical Link Layer) come una tradizionale rete ethernet. Cambia, però, il mezzo trasmissivo del segnale, che passa dal cavo fisico a onde radio spedite in broadcast [1] nell'ambiente circostante o da una unità centrale chiamata Access Point (AP) oppure direttamente da tutti i nodi che compongono la rete in modo paritario.
Questa distinzione dà vita alle due configurazioni principali con cui le reti wireless sono impiegate nella pratica: *Config. con Infrastruttura* e *Config. Ad-Hoc*.
Con l'avvento di reti wireless occorre introdurre il concetto di *stabilità*, infatti, a differenza delle reti cablate in cui non si presentano fluttuazioni nelle performance a parte i casi di guasti, la qualità dei collegamenti wireless dipende da un insieme di variabili. I fattori che possono incidere sulla stabilità del link possono essere condizioni del segnale radio, distanza tra i nodi, presenza di ostacoli lungo il percorso e interferenze. Si può affermare che le reti cablate obbediscono a una logica *vero-falso*, mentre le reti wireless obbediscono a una logica sfumata (*fuzzy*).
Oltre al problema della stabilità occorre porre attenzione anche allo scarso raggio che un segnale wireless può percorrere (circa 100 metri), che è il motivo per cui si sono create le diverse configurazioni atte a rendere disponibile il servizio su zone più ampie del raggio a disposizione (vedi fig. 2.2).

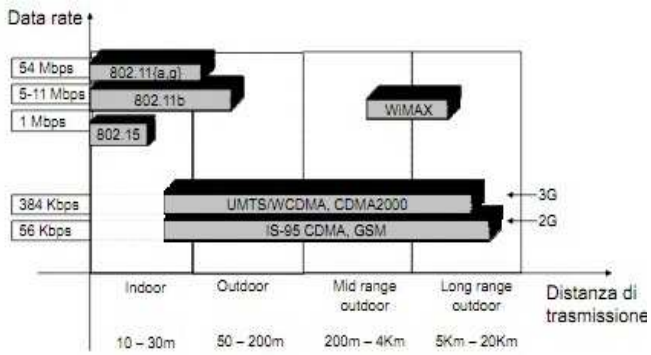

Figura 2.2: Copertura reti senza fili

2.2.1 Reti Wireless, Configurazione con Infrastruttura

Una rete wireless con infrastruttura (o *Strutturata*) è divisa in celle chiamate BBS (*Basic Service Set*), ogni cella è controllata da un unità centrale (l'*Access Point*). L'Access Point collega l'intera cella ad un altra rete ethernet che fornisce i servizi, la DS (*Distribution System*). Tutto il sistema è visto hai livelli più alti ISO/OSI come un'unica rete standard 802 e prende il nome di ESS (*Extended Service Set*). Ogni ESS è individuato da un ESSID (*Service Set Identifier*), un identificativo di 32 caratteri ASCII che serve da nome della rete.

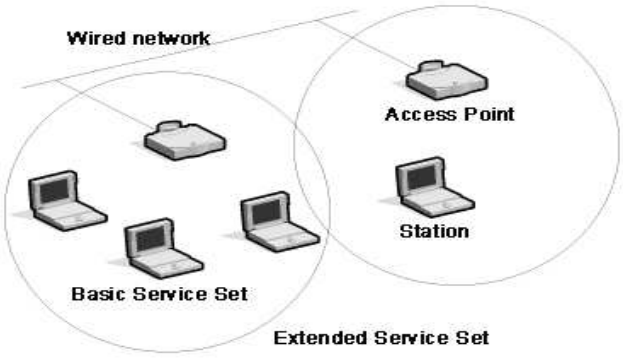

Figura 2.3: Rete wireless strutturata

L'Access Point svolge il ruolo fondamentale in questa configurazione, in quanto ha il compito di fornire la connessione sia tra i vari client, sia tra i

client e la rete esterna, dando forma a una struttura gerarchica.

2.2.2 Reti Wireless, Configurazione Ad-Hoc

Una rete wireless Ad-Hoc [2] è composta da soli terminali wireless, si crea spontaneamente e non necessita della presenza di un punto di accesso. A differenza della struttura gerarchica delle reti strutturate, non esiste una architettura statica, i client si connettono gli uni agli altri per costruire spontaneamente una rete *peer-to-peer*, ossia una rete in cui ogni terminale ha allo stesso tempo sia il ruolo di client, sia quello di punto di accesso.
I diversi nodi che compongono la struttura è chiamato IBSS (*Indipendant Basic Service Set*).

Figura 2.4: Rete wireless Ad-Hoc

A differenza della configurazione strutturata, dove l'Access Point, aveva maggiore importanza rispetto a tutti gli altri componenti della rete, qui tutti i terminali sono collegati tra loro in modo paritario, formando una struttura non gerarchica e il traffico è, per la maggior parte User-To-User.
Si tratta quindi di un sistema distribuito che rende la topologia della rete estremamente variabile.
Solitamente questa configurazione è usata quando i client sono pochi in modo simile a una connessione con cavo crossed tra 2 computer, ma non è adatta per una rete numerosa per via della possibile sovrapposizione dei segnali e il conseguente calo di affidabilità.

L'applicazione di maggiore successo delle reti wireless ad-hoc sono le *Mobile Ad-hoc NETwork* (*M.A.N.E.T.*) e le *Wireless Mesh Network* (WMN).

M.A.N.E.T.

La definizione data dall'IETF (Internet Engineering Task Force) [3] per questa applicazione è: *Una M.A.N.E.T. è un sistema autonomo di router ,mobili e dei loro host associati, connessi con collegamenti di tipo wireless che sono uniti formando un grafo di forma arbitraria. Tali router sono liberi di muoversi casualmente e di auto organizzarsi arbitrariamente, sebbene la topologia wireless vari rapidamente ed in modo imprevedibile. Tale rete può operare da sola oppure essere connessa a internet.*
In pratica si tratta di una rete creata per connettere dispositivi in continuo movimento, in cui tutti i nodi del sistema collaborano per creare una rete di collegamenti più generale.
Il traffico si muove attraverso una *nuvola* di apparecchi wireless, che possono sia trasmettere che ricevere dati senza però concorrere all'instradamento sei pacchetti.
Nella vita reale, l'utilizzo di questa tecnologia si limita soprattutto a scopi militari e situazioni di disastri, ma si tratta di un ulteriore passo evolutivo verso le reti Mesh, ossia l'ambito in cui il protocollo B.A.T.M.A.N. viene utilizzato.

Capitolo 3

WMN - Wireless Mesh Network

Una *Wireless Mesh Network* (rete magliata senza fili) è una particolare tipologia di rete wireless ad-hoc ideata per fornire una copertura radio che consente accessi wireless a banda larga sia in ambienti limitati, come abitazioni o uffici, sia in aree estese come campus o città, con la particolarità di agevolare l'accesso per zone difficilmente raggiungibili o scarsamente popolate.

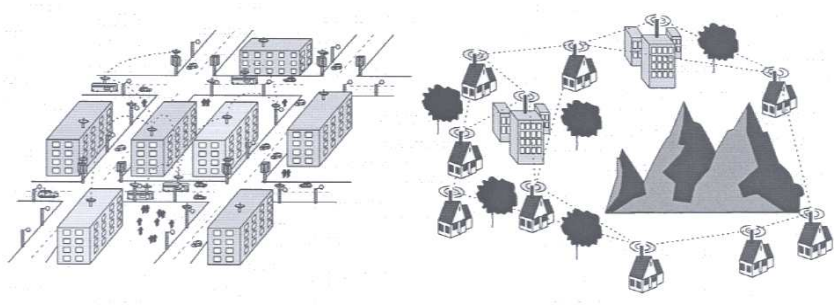

Figura 3.1: Wireless Mesh Network in ambito urbano e in ambienti di difficile accesso.

Generalmente queste reti sono realizzate con tecnologia WiFi secondo gli standard 802.11b/g mediante l'utilizzo di una frequenza di 2,4 GHz.

3.1 Caratteristiche principali

Una WMN è una rete *decentralizzata* (non esistono server che gestiscono il traffico) e *cooperativa* (tutti i mesh point svolgono un ruolo fondamentale per il corretto funzionamento dell'intera rete), in cui ogni nodo, sia esso un router o un client, può comunicare direttamente qualsiasi altro nodo.
E' costituita da un gran numero di nodi, ognuno dei quali svolge sia il ruolo classico di host, sia funzionalità di routing, effettuando operazioni di forwarding dei pacchetti verso altri nodi.
La caratteristica principale è la *dinamicità*: se un nodo per qualsiasi motivo va down la rete si comporta in modo da compensare questa mancanza, riorganizzando il routing e trovando percorsi alternativi per inoltrare ugualmente i dati evitando il nodo non funzionante. Questo concetto è definito *fault-tolerant*.
Per la connessione a internet sfruttano un limitato numero di access point che danno il servizio a tutti i nodi della rete mesh.

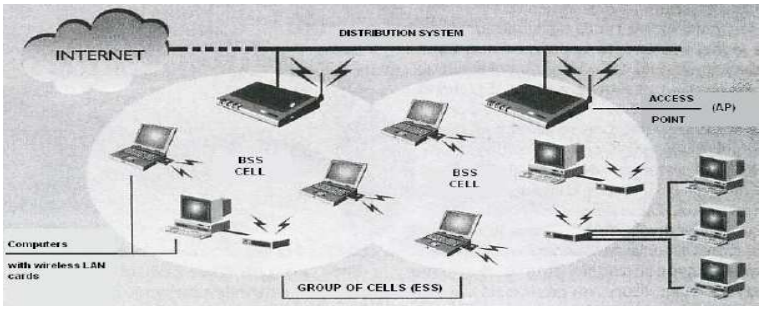

Figura 3.2: Struttura generica di una Wireless Mesh Network

I nodi di questa rete sono di due tipi: *mesh router* e *mesh client*.
Un *mesh router*, oltre a svolgere tutte le funzioni dei router classici, ha funzionalità aggiuntive. In pratica esegue un costante monitoraggio dell'attività di rete grazie al mantenimento e aggiornamento delle liste dei dispositivi attivi nelle vicinanze. Essendo una rete dinamica, in qualsiasi momento si può verificare la presenza di nuovi nodi (appena connessi) o la mancanza di nodi prima presenti. Quando un dispositivo compare, manda un messaggio di

broadcast che permette al router di aggiornare la sua lista con uno specifico timeout, se entro questo tempo non riceve un altro messaggio di broadcast il router lo cancella dalla lista. Questo procedimento permette di avere sempre un immagine aggiornata della rete dinamica.
Poiché in una rete mesh tutti i nodi svolgono funzione di router, la capacità di trasmissione ha un ruolo meno importante rispetto a un router tradizionale, essi più che altro rappresentano la backbone (spina d'orsale) per i mesh client.
Il costo di questi router è più alto di quello dei router classici per via delle funzionalità aggiuntive implementate in essi.
Un *mesh client*, come abbiamo detto può svolgere sia funzioni di host che funzioni di router, ma grazie hai mesh router il loro compito è più semplice, i protocolli di comunicazione sono più leggeri e non possono svolgere il ruolo di gateway o di bridge. Essi possono, inoltre, essere sia fissi che mobili, permettendo cosi alle WMN di offrire un valido supporto alla mobilità.

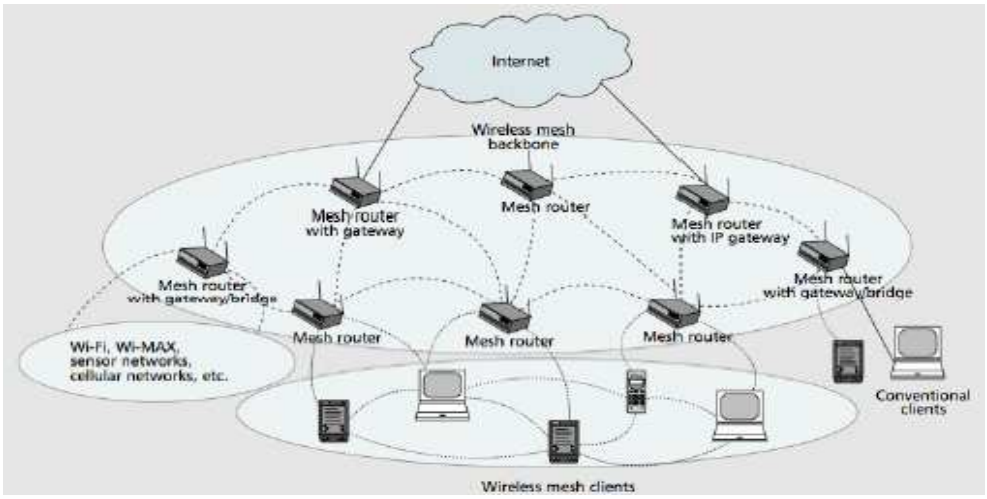

Figura 3.3: Backbone (spina d'orsale) di una rete mesh

Caratteristica importante delle WMN è che tutti i nodi sono collegati tra loro, direttamente o indirettamente. Quando la connessione è diretta si parla di *single-hop*, altrimenti di *multi-hop*. Poiché esiste un link tra ogni nodo i percorsi possono essere diversi, nel senso che anche se la distanza tra due nodi è di 2 hop, c'è sicuramente un'altra strada (probabilmente più lunga di

2 hop) che collega i due nodi. Questo fatto gioca un ruolo fondamentale per l'*affidabilità*, se un nodo viene meno alla rete, a causa di problemi hardware o per qualunque altro motivo i nodi vicini semplicemente cercano altri percorsi per trasmettere il segnale.
Una WMN è un modello particolare di rete wireless in configurazione ad-hoc. A differenza di quest'ultima, però, oltre a usufruire della maggiore robustezza data dai mesh routers, può contenere anche nodi mobili, per supportare la mobilità, si appoggiano a una infrastruttura e spediscono la maggior parte dei dati in modo User-To-Gateway.

Vantaggi e Svantaggi

I vantaggi delle WMN sono molti, alcuni derivano direttamente dai vantaggi delle reti wireless generiche, altri sono proprie delle Wireless Mesh Network.
I *costi di installazione* sono bassi, sia nei confronti di una rete cablata, sia nei confronti di una normale rete wireless. Per quanto riguarda la rete cablata la minor spesa riguarda la necessità di infrastruttura meno onerosa rispetto alla necessità di installazione di cavi di quest'ultima. Per quanto riguarda le reti wireless, nonostante il più alto costo dei routers, le reti mesh, per la copertura di grandi zone, permettono l'installazione di un numero minore di mesh router rispetto agli access point della configurazione strutturata, la costante connessione tra i nodi permettere di espandere il segnale a tutti i nodi della rete idealmente anche da un solo router (fig. 3.4).

Come già spiegato le WMN godono di alta *affidabilità* grazie a percorsi ridondanti tra endpoints e backbone wireless anche utilizzando dispositivi mobili, inoltre la gestione dei guasti e il setup della rete sono automatici, la rete in pratica funziona in *autogestione*, e necessita meno manodopera di qualunque altro tipo di rete.
L'altro grande vantaggio delle WMN risiede nella *scalabilità*, ossia nello *sviluppo su larga scala*. L'aggiunta di un nodo (o più nodi) infatti non causa nessun cambiamento nel progetto della rete, al massimo necessità di più

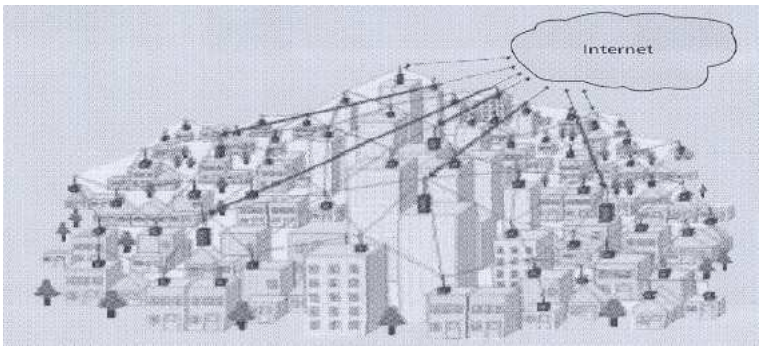

Figura 3.4: Accesso broadband a internet con limitato num. di access point

memoria per le tabelle di routing. Essa è completamente dinamica e con l'aggiunta di nuovi nodi si estende anche la copertura totale. Inoltre con la tecnica multi-hop per le comunicazioni su lunghe distanze non si va a intaccare la velocità di trasmissione, in quanto a ogni hop (un singolo hop è di solito molto breve) viene rispedito il messaggio mantenendo la velocità pressoché costante.
Essendo una tecnologia sostanzialmente nuova, ancora presenta anche alcuni svantaggi.
Innanzitutto l'*assenza di uno standard* costringe a usare standard vecchi come 802.11 o 802.15 e tecnologie proprietarie, causando una forte dipendenza dal vendor che integra tali tecnologie.
In secondo luogo, il routing dinamico e le operazioni multi-hop implicano che un nodo convogli non solo il proprio traffico, ma anche quello degli altri. In pratica la banda effettiva di un nodo può essere minore di quella disponibile a livello radio sottostante. Questo problema è chiamato *throughput*.
Infine la natura dinamica e adattiva rende difficile garantire la *QoS* (Quality Of Service).

3.2 Impieghi

Inizialmente le WMN sono nate per scopi militari, che necessitavano di reti facili da creare in posti difficili da raggiungere. La natura poco costosa di questo progetto e l'affidabilità hanno portato a una notevole estensione

degli impieghi di questa tecnologia:

- Collegamento ad internet di aree topografiche o geografiche periferiche
- Realizzazione di reti urbane a basso costo e a basso impatto ambientale
- Reti di sensori per rilevamenti di vario genere
- Reti dinamiche costituite da nodi in movimento per organizzazione di soccorsi in situazioni di catastrofi naturali
- Messa a disposizione di risorse socialmente utili in situazioni di svantaggio ambientale (digital divide)
- Sorveglianza di punti sensibili mediante l'impiego di telecamere o altri dispositivi
- ecc.

3.3 Protocolli di routing

I protocolli di instradamento per reti cablate, caratterizzate da percorsi rigidi e statici, e i protocolli per internet(O.S.P.F., B.G.P., R.I.P.v2, ecc.) sono progettati con l'assunzione di rari cambiamenti nei link senza significative modifiche alla topologia della rete, non sono adatti alla dinamicità delle WMN.
I protocolli per reti wireless in configurazione ad-hoc, invece, sono progettati per tassi di cambiamento più alti, e con opportune ottimizzazioni possono essere usati per le WMN.
L'organizzazione delle connessioni consiste essenzialmente in 3 passaggi: quando due nodi (qualunque) non hanno un collegamento diretto tra loro, le funzionalità di routing devono stabilire una connessione, ridurre al minimo il traffico indotto (overhead) e rilevare velocemente eventuali interruzioni dei link allo scopo di minimizzare perdite nella trasmissione dei pacchetti.

3.3.1 Protocolli di Routing Tradizionali

Volendo dare una definizione, possiamo dire che per routing si intende il processo di scambio di informazioni da un nodo a un altro nella rete.
Storicamente ne esistono di due tipologie: *Link State* e *Distance Vector*.

Protocolli Link-State

I protocolli link-state (routing basato sullo stato del collegamento) utilizzano il concetto di *mappa distribuita*, ossia un elenco di tutti i link della rete con relativo costo. Tutti i routers hanno una copia di tale mappa che viene aggiornata in continuazione. Ogni nodo, periodicamente, usa la tecnica del *flooding*, chiamata in questo caso *Link State Broadcast*, per inviare a tramite tutti i suoi link diretti un messaggio *Link State Packet* (LSP), contenente tutte le informazioni sui link tra il mittente del messaggio e tutti i suoi vicini. Alla ricezione di tale messaggio, ogni router, aggiorna la propria *routing table* e lo rispedisce a tutti i suoi vicini diretti tranne quello da cui arriva tale messaggio.
Più in dettaglio quando viene ricevuto un LSP si confronta il numero di sequenza del pacchetto con quello dell'ultimo LSP ricevuto da quello stesso nodo e se il numero di sequenza appena arrivato è più recente, allora il pacchetto viene memorizzato e inoltrato tutti i nodi collegati tranne quello mittente. Se invece il numero è invariato o più vecchio, il messaggio viene, rispettivamente, scartato o rimandato al mittente.
In questo modo tutti i routers hanno sempre memorizzata la mappa della rete aggiornata in una struttura ad albero ed il cammino più conveniente si ottiene con un algoritmo *shortest path*, che in genere è l'algoritmo di Dijkstra.
Link-State è particolarmente adatto per gestire reti complesse con un elevato numero di nodi, in quanto essendo sempre presente e aggiornata l'intera topologia riesce a convergere rapidamente sul cammino minimo nonostante la complessità della rete stessa evitando, tranne che in rari casi, di generare cammini ciclici. Inoltre, inviando aggiornamenti soprattutto per cambiamenti nelle tabelle di routing e non periodicamente non va a diminuire la capacità di banda della rete, mantenendo sempre una velocità elevata.

Lo svantaggio di questi protocolli sta nella complessità di realizzazione e nella notevole quantità di memoria (RAM) richiesta nei routers per mantenere l'intera struttura della rete salvata.

Protocolli Distance Vector

I protocolli Distance Vector si basano sull'algoritmo di Bellmann-Ford. Ogni nodo mantiene un database con le distanze minime tra sè stesso e tutte le possibili destinazioni. A intervalli regolari invia ai nodi adiacenti un *distance-vector*, che è un insieme di coppie *indirizzo-distanza*, chiamate annunci. La distanza è espressa come numero di hop o con criteri più generali che tengono conto di velocità, carico e affidabilità dei collegamenti.
A partire da tali dati, utilizzando l'algoritmo di Bellman-Ford, il router costruisce una tabella che associa ad ogni destinazione conosciuta la distanza che lo separa dalla destinazione e il primo passo del percorso calcolato. Quando riceve il distance-vector, un nodo può usare queste informazioni per ricalcolare la sua tabella di routing e, a differenza dei Link-State, questo messaggio non viene forwardato.
Un router ricalcola le proprie tabelle se:

- cade una linea attiva direttamente connessa
- riceve da un router vicino un annuncio per una destinazione non conosciuta
- riceve da un router vicino un annuncio per una destinazione già nota, ma a costo più basso rispetto a quello memorizzato
- riceve da un router vicino un annuncio per una destinazione che lo stesso router aveva già annunciato precedentemente con costo più elevato
- scade il tempo massimo di vita (TTL) per una destinazione in tabella

Si può affermare che pregi e difetti di questi tipi di protocolli siano opposti a quelli dei protocolli Link-State. Infatti sono più adatti a reti piccole senza requisiti stringenti di prestazioni poiché trovare il percorso migliore richiede

più tempo non avendo l'intera topologia a disposizione per il calcolo e questa complessità si traduce nella pratica in velocità di convergenza più bassa. Inoltre vengono inviati periodicamente gli aggiornamenti, non solo quando si individua un cambiamento nella rete e ciò influisce sulla capacità di banda della rete. Il problema più grande di questi protocolli sta nella possibilità di creare *routing-loop*, ossia quando un pacchetto rimane vincolato a muoversi fra due o più router senza trovare una via d'uscita.
D'altra parte la configurazione dei router risulta molto più semplice e la necessità di memoria (RAM) in essi è bassa poiché ogni router deve memorizzare solo le informazioni sui collegamenti da se stesso verso gli altri nodi e non sull'intera rete, questo si traduce in un minor costo.

3.3.2 Protocolli di Routing per Reti Wireless Ad-Hoc

Possiamo classificare i protocolli di routing per reti ad-hoc in base al momento in cui avviene l'elaborazione dei cammini.
Se le rotte vengono calcolate a priori, controllando tutti i possibili percorsi senza sapere se poi verranno effettivamente utilizzati, allora il protocollo è *Proattivo*.
Se invece le rotte vengono calcolate solo se effettivamente richieste al momento dell'effettivo instradamento del pacchetto (on-demand), allora il protocollo è *Reattivo*.
Esistono anche dei protocolli *ibridi*, che usano metodologia proattiva per i nodi vicini e metodologia reattiva per i nodi più lontani. Questi protocolli non sono di interesse per l'argomento di questa tesi quindi non verranno trattati in profondità.

Protocolli Proattivi (o *table driven*)

Questo tipo di protocollo deriva da quelli tradizionali e permette di avere a disposizione in ogni momento il percorso da una qualunque sorgente ad una qualunque destinazione. Tutti i possibili percorsi di routing vengono, infatti, calcolati e modificati senza sapere se verranno in futuro utilizzati.
Il vantaggio sta nell'aver sempre disponibile velocemente l'informazione di

instradamento ad ogni richiesta di routing, a scapito, però, di un elevato overhead (traffico di supporto) all'interno della rete, che può venire impegnata anche senza nessuna richiesta di trasmissione dati, e di una grande utilizzo delle risorse (sia di memoria che di CPU).
Per queste caratteristiche di velocità e occupazione di banda sono particolarmente adatti per applicazioni che richiedono una bassa latenza come, ad esempio, quelle interattive.
Ogni nodo è associato a una tabella di routing che mantiene informazioni su ogni nodo della rete, se un nodo ne modifica una il cambiamento deve essere propagato per tutta la rete, in modo da avere sempre una visione della rete consistente.

Protocolli Reattivi (o *source-initiated on-demand driven*)

In questo caso il percorso viene calcolato solo al momento in cui ci sono pacchetti da trasmettere, riducendo in questo modo il traffico di overhead della rete, ma impiegando più tempo per inviare effettivamente un pacchetto. Un approccio del genere risulta adatto in casi di banda limitata.
In ogni protocollo reattivo si possono individuare tre fasi: *Route Discovery*, *Route Maintenance* e *Route Delection*.
Si parla di Route Discovery quando, a seguito di una richiesta di trasmissione, il servizio di routing inizia la ricerca della destinazione. Il nodo mittente invia un messaggio di query a tutti i vicini che reagiscono facendo altrettanto, diffondendo in questo modo il messaggio nella rete. Quando questo pacchetto arriva a destinazione o ad un nodo che sa come raggiungere tale destinazione viene ritrasmessa al mittente la risposta ripercorrendo a ritroso il cammino seguito dal messaggio di query.
Costruito il cammino, a causa della mobilità degli host, è necessario controllarlo periodicamente e, se serve, ricalcolarlo. Questa fase prende il nome di Route Maintenance.
Se la fase di Route Maintenance ha avuto esito negativo occorre liberare le risorse occupate nei nodi intermedi (Route Deletion).

3.4 O.L.S.R. Optimized Link State Routing

Siccome il protagonista di questa tesi è un protocollo "nuovo" di routing per WMN, diventa necessario fare una panoramica sul protocollo "attuale" per WMN. Questo "protocollo attuale" è chiamato *Optimized Link State Routing* [4] (RCF-3626).
O.L.S.R. è un protocollo di routing dinamico *proattivo* e *Link-State*, che al giorno d'oggi rappresenta lo standard per Wireless Mesh Networks e M.A.N.E.T. La topologia di rete, quindi, viene mantenuta, aggiornandola a intervalli fissi di pochi secondi in tutti i nodi, facilitando l'instradamento dei pacchetti in ogni punto della rete ed evitando alti tempi di attesa.
Dal concetto di Link-State eredita la stabilità, ma migliora tale concetto con un'ottimizzazione dell'utilizzo della banda, minimizzando il flooding broadcast. Nell'algoritmo Link-State classico infatti lo scambio di informazioni avviene tramite broadcast da parte di *tutti* i nodi. Secondo O.L.S.R. però non è necessario questo flooding venga inviato da tutti i nodi, ma basta che a inviarlo siano un ristretto numero di vicini del nodo comunicante. Questo set di nodi prende il nome di MPR (*Multipoint Relays*) set.
La tecnica MPR nasce dall'osservazione che in una situazione di broadcast non ottimizzato ogni nodo riceve più volte le stesse informazioni causando un notevole spreco di banda e di potenza di calcolo. E' però possibile migliorare questa situazione scegliendo un sottoinsieme di nodi vicini con link bidirezionale che possono ritrasmettere le informazioni.

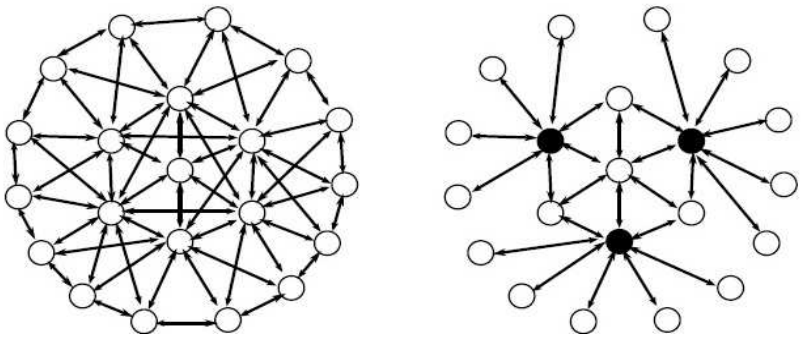

Figura 3.5: Broadcast completo Vs Broadcast MPR

E' stato dimostrato che questa è una scelta NP-completa, infatti l'RCF descrive un semplice algoritmo euristico per calcolare il sottoinsieme di MPR. O.L.S.R. utilizza pacchetti UDP per trasferire le informazioni di controllo. Un pacchetto di informazioni di controllo generico è descritto in figura 3.6.

Dimensione pacchetto		Numero di sequenza
Tipo del messaggio	Tempo di validità	Dimensione messaggio
Indirizzo del mittente		
Tempo di vita	Numero di salti	Numero di sequenza
Messaggio		
Tipo del messaggio	Tempo di validità	Dimensione messaggio
Indirizzo del mittente		
Tempo di vita	Numero di salti	Numero di sequenza
Messaggio		

Figura 3.6: Pacchetto O.L.S.R. generico

I messaggi richiesti per le funzionalità di base sono:

- HELLO - messaggi inviati a intervalli regolari che servono per il rilevamento dei vicini, il link sensing e la comunicazione dei nodi MPR
- MID - usati dai nodi con più interfacce per dichiararne l'esistenza al resto della rete
- TC (Topology Control) - comunicano le informazioni topologiche dal punto di vista di ogni nodo

Le funzionalità di base di questo protocollo possono essere riassunte in tre fasi: *Link Sensing*, *Optimized Flooding (MPR)* e *Link-State Messaging*.

Link Sensing

Questa fase ha il compito di effettuare un *probing* della rete tramite l'uso di messaggi HELLO in modo che ogni nodo venga a conoscenza di quale sia il suo vicinato (Neighborhood). Le informazioni sul vicinato sono mantenute in due tabelle chiamate rispettivamente *Link Set* e *Neighbor Set* che prevedono anche la possibilità per un link di essere sia simmetrico che asimmetrico.

Un link viene detto simmetrico quando un nodo riceve da un altro nodo già presente nella propria tabella di routing un messaggio HELLO che contenga il proprio indirizzo. Con questa tecnica è possibile capire quali siano i link bidirezionali.
Se un nodo A riesce a ricevere i messaggi di un nodo B, ma B non è in grado di ricevere messaggi da A, significa che il link è asimmetrico (o monodirezionale), in questo caso l'algoritmo tenta di trovare un MPR che possa mettere in comunicazione i due nodi in modo bidirezionale.

Optimized Flooding

Le informazioni sulla topologia della rete, come in tutti i protocolli Link-State è distribuita tramite pacchetti di controllo, che in O.L.S.R. sono chiamati *Topology Control* (TC). Per evitare che la rete venga inondata da questi pacchetti, l'Optimized Flooding fa si che un nodo elegga tra i suoi vicini (neighbor) il minimo insieme che consente di raggiungere tutti i *2-hop-neighbor* (tutti i vicini a distanza di due hop). Serve quindi capire quanti MPR deve avere ogni nodo (*tuning*) e quali informazioni topologiche far distribuire ai propri MPR.
Il tuning consente da un lato, scegliendo un parametro basso, cioè inoltrando i propri TC a solo pochi MPR, di diminuire il traffico sulla rete, dall'altro, scegliendo un alto numero di MPR, minimi di aumentare l'efficienza e la reattività della rete alle variazioni topologiche, avendo a disposizione Relay (instradatori) ridondanti.
Le informazioni topologiche da distribuire nei pacchetti TC consentono una scelta analoga: scegliendo di inviare poche informazioni in ogni TC fa si che il traffico nella rete sia basso, al contrario invece si aumenta la velocità di reazione della rete.

Link-State Messaging

Con tutte le informazioni ottenute dalle prime due fasi, ora non rimane che eseguire la ricerca della strada meno costosa, tramite Dijkstra.

Durante il funzionamento, ogni nodo, in base al contenuto dei messaggi che riceve, mantiene un certo numero di tabelle. Le più importanti sono:

- Neighbor Set - insieme dei vicini, ossia quei nodi da cui è stato ricevuto un messaggio HELLO
- 2-Hop Neighbor Set - insieme dei nodi vicini a distanza di 2 hop
- MPR Set - insieme dei nodi che agiscono come MPR per il nodo in questione
- MPR Selector Set - insieme dei nodi che hanno scelto il nodo attuale come MPR
- Topology Set - tabella contenente l'elenco di tutti i nodi esistenti sulla rete, con associato l'indirizzo IP del nodo e l'indirizzo IP del nodo che lo ha pubblicizzato come vicino

Il protocollo O.L.S.R. fa parte dei protocolli Link-State e presenta quindi i tipici svantaggi di questa tipologia di protocolli. Il problema principale risiede nella scalabilità della rete, infatti ogni router mantiene informazioni sull'intera topologia nelle sue tabelle, ciò comporta che più la rete è grande, più le tabelle necessiteranno di memoria e più saranno possibili incongruenze tra i database di ampie dimensioni.
Le reti odierne diventano via via più grandi e le reti mesh iniziano a trovare sviluppo anche per coperture metropolitane (quindi di grandi dimensioni), in questo contesto O.L.S.R. crea problemi di costi e di prestazioni. E' questo uno dei motivi principali per cui B.A.T.M.A.N. sta prendendo velocemente piede ed è destinato in un futuro prossimo a diventare il nuovo standard per Wireless Mesh Networks.

3.5 A.O.D.V - Ad-hoc On-demand Distance Vector

Oltre a O.L.S.R., che è di tipo Link-State, un altro algoritmo di routing usato per WMN, anche se con minore successo è A.O.D.V. [5].

Si tratta di un algoritmo di tipo reattivo, che, quindi, costruisce il cammino ogni volta che un pacchetto deve essere inviato. Questo provoca un ritardo iniziale nella comunicazione, per il tempo necessario ad ottenere dal resto della rete una *route* valida. Una volta che il cammino è stato stabilito viene mantenuto nella cache di tutti i nodi intermedi con l'informazione di quale sia il nodo successivo da contattare. Quando uno di questi collegamenti punto a punto viene a mancare, l'ultimo nodo raggiungibile dalla sorgente rispedisce indietro un messaggio di servizio che causa la rimozione dalla cache del cammino non più valido e innesca una nuova ricerca a partire dalla sorgente.
Senza stare a soffermarsi tanto su questo protocollo a causa dello scarso utilizzo nelle reti attuali, mi limiterò a fare una lista dei vantaggi e, soprattutto, degli svantaggi di questo protocollo, che hanno portato a passare tecniche più performanti come O.L.S.R. o B.A.T.M.A.N.
I vantaggi si limitano ad un basso utilizzo della banda (occorre ricordare che con lo svilupparsi di nuove tecnologie il bandwidht è ampliamente aumentato quindi il vantaggio è relativo) e a un efficiente sfruttamento delle risorse dei router, che in quanto devono memorizzare poche informazioni non necessitano di grandi quantità di memoria RAM.
I difetti invece sono più significativi, partendo da un consistente ritardo (parliamo di qualche secondo) necessario ad ogni inizio di nuove comunicazioni verso nodi ancora inesplorati, per arrivare a un imponente uso di CPU per il calcolo di ogni percorso.

Capitolo 4

B.A.T.M.A.N. - Better Approach To Mobile Ad-hoc Networking

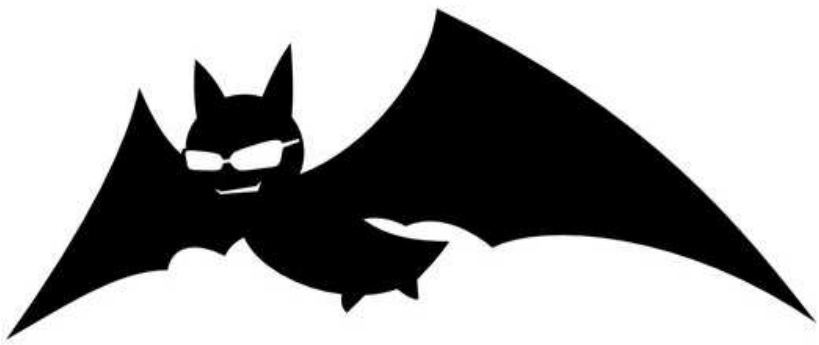

Figura 4.1: Logo ufficiale di B.A.T.M.A.N., reso pubblico in data 8/10/2011

Fino ad ora, lo scopo è stato quello di creare un contesto chiaro per capire precisamente i campi di utilizzo e i problemi che hanno posto le basi alla nascita di B.A.T.M.A.N.[6]. L'aumentare di Wireless Mesh Network in larga scala ha messo in risalto un calo di prestazioni notevole del protocollo O.L.S.R. e si è reso necessario sviluppare un protocollo che partisse da un idea di base diversa.
Mentre O.L.S.R. usa un approccio Link-State, ossia calcola indipendentemente dall'effettivo utilizzo tutti i percorsi e li distribuisce a tutti i nodi della rete, B.A.T.M.A.N. ha un approccio Distance Vector, in pratica non cerca di determinare l'intero percorso, ma utilizzando dei messaggi speciali (Originator Messages) trova solo la prima tappa da ogni nodo verso la destinazione.

4.1 Breve storia del protocollo

A differenza di molti altri protocolli di routing, B.A.T.M.A.N. ha subito una implementazione pratica e una ampia fase di sperimentazione fin dall'inizio del progetto, migliorando via via l'algoritmo per adattarlo ai nuovi problemi della vita reale.
Nel 2005 *Corinna Aichele* e *Thomas Lopatic* riconsiderarono O.L.S.R. per la sua complessità e per tutte le modifiche apportate negli anni all'algoritmo originale per farlo funzionare ad un certo livello di realismo in reti wireless e ebbero l'idea di un nuovo protocollo con approccio molto più semplice.
Da allora possiamo distinguere tre generazioni del protocollo che lo hanno portato ad essere praticamente utilizzabile solo da pochi anni.
La prima generazione (B.A.T.M.A.N.-I)non verificava le condizioni del collegamento bidirezionale per l'inoltro pacchetti. Si trattava di un difetto di progetazione ovvio, ma la prima generazione aveva il solo scopo di testare l'algoritmo e i risultati sono stati comunque promettenti.
Nella seconda generazione (B.A.T.M.A.N.-II) viene implementato l'algoritmo base per il controllo dei link bidirezionali per nodi mesh con *una sola* interfaccia. Un link specifico viene considerato *bidirezionale* per un certo periodo di tempo se la risposta (*re-broadcast*) a un messaggio da sè stesso

inviato viene ricevuta dal vicino corrispondente all'invio. Per distinguere un messaggio ricevuto tramite un link unidirezionale da uno ricevuto tramite un link bidirezionale è stato introdotto il flag UDF (*Unidirectional Link*).
La terza generazione (B.A.T.M.A.N.-III) è molto simile a quella attuale (a cui ovviamente sono stati apportati ulteriori miglioramenti). Qui viene introdotto il concetto di *best-ranking neighbor* (nodo vicino con ranking migliore). Tutti questi concetti saranno spiegati dettagliatamente in seguito.
All'inizio del 2007 vi fù l'ultimo grande cambiamento nell'algoritmo. Inizialmente il protocollo lavorava, come grandissima parte dei protocolli in circolazione a livello 3 della pila ISO/OSI, ossia a livello Network. Alcuni sviluppatori hanno iniziato, con successo, a sperimentare l'idea di *Routing a livello Data Link* (Livello 2). Questa si è rivelata una ulteriore evoluzione per il protocollo e per distinguerlo dalle versioni precedenti (Routing a livello Network) è stato chiamato *B.A.T.M.A.N.-adv* (advanced). Al principio furono sviluppate sia un interfaccia virtuale userspace che un modulo del kernel, ma l'interfaccia userspace ha imposto un significativo overhead (carico di lavoro) per i nodi wireless di fascia bassa e per questo motivo è stato rimosso. Oggi quando si parla di B.A.T.M.A.N.-adv ci si riferisce solo al modulo kernel, che è infatti diventato parte del kernel ufficiale di linux dalla versione 2.6.38, rilasciata all'inizio del 2011.
L'attuale versione di B.A.T.M.A.N.-adv è la 2011.3.0. ed è disponibile solo per piattaforma Linux.

4.2 Principi di Base

Il problema con i classici protocolli di routing è che spesso non sono adatti per le reti wireless ad-hoc e quindi neanche per le reti mesh poiché tali reti si modificano dinamicamente nella topologia e si basano su un mezzo (la wireless) intrinsecamente instabile. O.L.S.R., il protocollo attualmente più usato per tali scenari, pur sottoposto a numerose modifiche per affrontare le sfide imposte dalle reti mesh cittadine (di ampie dimensioni) presenta dei limiti dovuti alla necessità dei protocolli Link State di calcolare l'intero grafico/topologia per ogni nodo. Per fare un esempio un router integrato con una

piccola CPU embedded per calcolare l'intera topologia di una rete di circa 450 nodi impiegherebbe diversi secondi.
L'approccio di B.A.T.M.A.N. consiste nel dividere la conoscenza tra tutti i nodi della rete e renderli tutti partecipanti attivi alla rete stessa.
Ogni router mantiene solo le informazioni riguardanti al miglior *next-hop* verso tutti gli altri nodi della rete. In questo modo non è più necessaria una conoscenza globale riguardo ai cambiamenti della topologia.
Una rete mesh basata su B.A.T.M.A.N. viene inondata a intervalli regolari da Originator Messages fino a che non vengono ricevuti dal destinatario almeno una volta o finché il pacchetto non viene perso causa fine del TTL o problemi di comunicazione. Il protocollo ritiene significativa ogni perdita di pacchetti per trovare il percorso migliore. Il numero di OGM ricevuti da un determinato Originator dai vicini viene usato per stimare la qualità di un tragitto (sia single-hop che multi-hop). Per fare questa stima B.A.T.M.A.N. conta gli OGM ricevuti e mantiene questa informazione in una tabella (Originator List) in cui calcola il miglior next-hop per tutta la rete.
Il protocollo è, quindi studiato, per cogliere gli effetti delle fluttuazioni e dei malfunzionamenti della rete e compensare tale instabilità.
B.A.T.M.A.N. comunica usando UDP e la porta 4305 è stata assegnata da I.A.N.A. (Internet Assigned Numbers Authority) per uso esclusivo di questo protocollo.

4.3 Formato Pacchetti

4.3.1 Pacchetto B.A.T.M.A.N. Generico

Ogni pacchetto è incapsulato in un singolo *UDP data packet* ed è formato da un *Originator Message* (OGM) e zero o più estensioni HNA (*Host and Network Association*) [7].

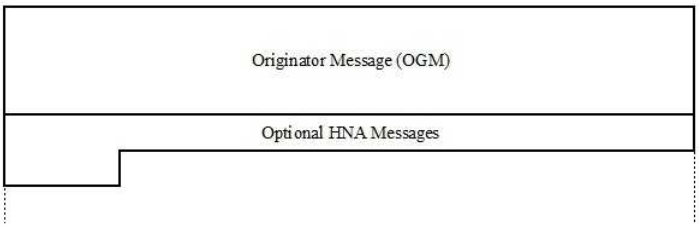

Figura 4.2: Pacchetto B.A.T.M.A.N. generico

4.3.2 Originator Messages

Gli Originator Messages sono messaggi piccoli, tipicamente di 52 byte comprese le informazioni su IP e UDP. Essi devono contenere almeno l'indirizzo del mittente, l'indirizzo del nodo che sta attualmente trasmettendo il pacchetto, il TTL (Time To Live) e un Sequence Number.

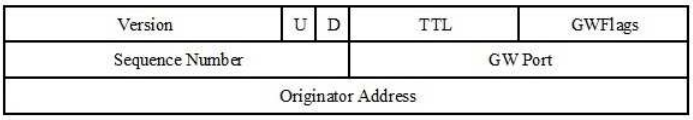

Figura 4.3: Formato di un Originator Message (OGM)

- *Version*: Campo obbligatorio, ogni OGM ricevuto di una versione differente sarà ignorato
- *Unidirectional Flag* (U): flaggato se il link è unidirezionale (o asimmetrico)
- *Is-Direct-Link Flag* (D): flaggato se si tratta di un link diretto, quindi se composto da un solo *hop*
- *TTL* (Time To Live): Limite temporale superiore al massimo numero di *hop* con cui un OGM può essere trasmesso
- *GWFlags* (Gateway Flag): Codifica la larghezza di banda approssimativa in kbit/sec nel caso il nodo abbia accesso a internet e si renda disponibile a fare da gateway per gli altri nodi della rete

- *Sequence Number*: Numero che identifica l'OGM in questione. Ogni nodo tiene memorizzato una cronologia degli ultimi numeri di sequenza ricevuti per essere sicuro di non aver già ricevuto un OGM dallo stesso mittente.
- *Originator Address*: Indirizzo IPv4 dell'interfaccia di rete che ha originato l'OGM

4.3.3 HNA Messages

Le estensioni HNA servono quando un nodo vuole annunciarsi (inviare un OGM) a reti non B.A.T.M.A.N.

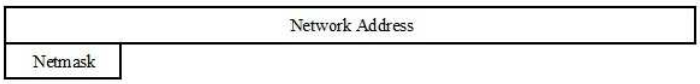

Figura 4.4: Formato di un HNA Extension Message

- *Network Address*: Indirizzo IPv4 della rete a cui vuole annunciarsi
- *Netmask*: Numero di bit che presenta la grandezza della rete

4.4 Strutture Dati

4.4.1 Originator List

Ogni nodo mantiene informazioni sugli altri mittenti conosciuti della rete nella Originator List. Questa tabella mantiene una voce per ogni mittente dal quale è stato ricevuto un OGM. Poiché ogni nodo-B.A.T.M.A.N. può avere più di un'interfaccia, può succedere che da uno stesso vicino arrivino OGM da Originator diversi, in questo caso la tabella deve tener conto di ogni singola interfaccia.
Per ogni Originator vengono mantenute le seguenti informazioni:

- Originator IP Address

- Last Aware Time: un timestamp che deve essere aggiornato ad ogni OGM ricevuto dall'Originator dato
- Bidirect Link(s) Sequence Number: il numero di sequenza dell'ultimo OGM autoinizializzato e accettatodalla lista dei vicini diretti
- Current Sequence Number: il più nuovo numero di sequenza che è stato accettato da un OGM di un dato Originator
- HNA List: tutte le reti degli Originator con netmask e IP-range
- Gateway Capability: se l'Originator offre anche un gateway
- Neighbor Information List: per ogni link diretto con verso i vicini del nodo in questione vengono mantenute Sliding Window, Packet Count, Last Valid Time e il TTL dell'ultimo OGM ricevuto da un dato vicino. Lo Sliding Window è un controllo del numero di sequenza ricevuto attualmente con quelli ricevuti in precedenza per capire se sia già stato ricevuto. Il Packet Count è il numero dei numeri di sequenza registrati in una Sliding Window ed è usato come metrica del percorso dall'Originator tramite il vicino in questione. Last Valid Time è il timestamp del momento di ricezione dell'ultimo OGM tramite il vicino dato.

4.4.2 Sequence Number

B.A.T.M.A.N. è *Sequence Number Oriented*, in quanto l'informazione chiave che viene trasmessa da ogni OGM è, appunto, il numero di sequenza. Questi numeri sono registrati in appositi Sliding Windows fino a quando non sono considerati Out-Of-Range, in modo che la Sliding Window contenga sempre l'insieme dei più recenti numeri di sequenza ricevuti. L'ammontare dei Sequence Number per ogni Originator viene utilizzato per determinare la qualità dei link dei percorsi individuati.
Il range dei numeri di sequenza è limitato e va da 0 a $2^{(16-1)}$, che comporta l'obbligo di eseguire le operazioni aritmetiche in modulo 2^{16}.

4.5 L'algoritmo

Ogni rete è modellata come un grafo $G = (N, E)$, dove N rappresenta l'insieme dei nodi e E rappresenta l'insieme degli archi che connettono i vari nodi.
Per ogni nodo $i \in N$ in B.A.T.M.A.N. esiste l'insieme dei *one-hop neighbor*, ossia l'insieme di tutti i vicini del generico nodo i raggiungibile tramite un solo *hop*. Chiamiamo questo insieme l'insieme K.
Il messaggio dalla sorgente $s \in N$ alla destinazione $d \in N$ viene trasmesso lungo il collegamento $(s, d) \in E$ solo se d è un elemento del sottoinsieme K. Quindi se d è un vicino one-hop di s.
Altrimenti viene trasmesso lungo un tragitto *multi-hop* composto da un link (s, i) e un percorso $[i, d]$ dove i è un nodo scelto appartenente all'insieme K. Il percorso $[i, d]$ rappresenta il tragitto dal nodo i al nodo d attraverso la sottorete S.

$$S = (N - \{s\}, A - \{(s, i) : i \in K\}) \tag{4.1}$$

Possiamo considerare l'algoritmo di B.A.T.M.A.N. divisibile in 4 step:

- Step 1: dato un messaggio m diretto dal mittente s al destinatario d nella rete/grafo G , si possono escludere tutti i link (s, i) per ogni $i \notin K$, ossia tutti gli archi che terminano in un nodo che non è un vicino *one-hop* di s

- Step 2: a ogni arco non escluso nello Step 1 viene associato un peso $\omega_{s,i}$ che indica il numero di OGM ricevuto dalla destinazione d attraverso il vicino i di s, senza il passaggio corrente

- Step 3: dati i possibili archi (s, i) e il loro peso $\omega_{s,i}$, trovare l'arco nel sottografo con $\omega_{s,i}$ più alto e inviare il messaggio m attraverso l' (s, i) scelto

- Step 4: se $i \neq d$ ripetere gli Step da 1 a 4 finché dal corrente i non si arriva a d nel sottografo S.

Per capire meglio l'algoritmo si propone il seguente esempio.
Si consideri la rete/grafo composta di nodi wireless G di figura 4.5.

Il nodo *1* vuole inviare un messaggio al nodo *6*. Come soli link utilizzabili considererà solo i vicini *one-hop* del nodo *1*, che compongono l'insieme $K = \{2, 3, 4\}$(figura 4.6).
Supponendo che il link con maggior $\omega_{s,i}$ sia $(1, 2)$, verrà scelto per il prossimo nodo del percorso del tragitto che farà il messaggio il nodo *2* (figura 4.7).
Dato che il nodo *2* non è la destinazione, riduciamo il grafico G nel più piccolo grafo S, escludendo i nodi che non fanno parte del nuovo insieme K, e ripetiamo gli Step dell'algoritmo (figura 4.8).
Supponendo che il link $(2, 5)$ abbia $\omega_{s,i}$ maggiore, verrà scelto come prossima tappa nel percorso e ripetendo un ultima volta gli Step si avrà che la tappa successiva sarà la destinazione, terminando l'algoritmo.

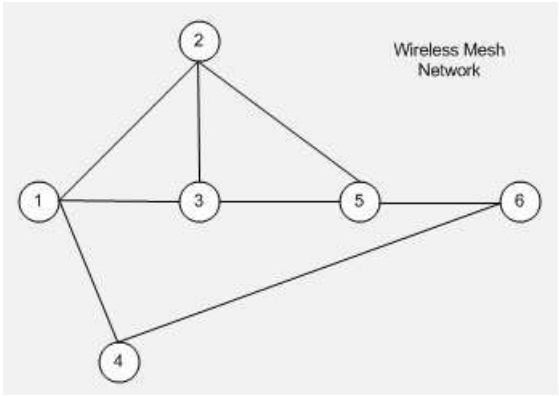

Figura 4.5: Algoritmo: Rete/grafo iniziale G

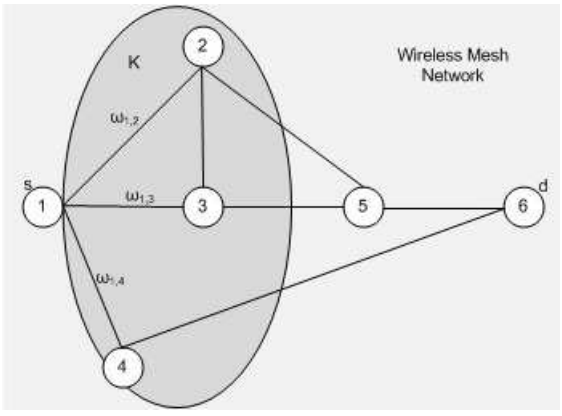

Figura 4.6: Algoritmo: Sottografo K

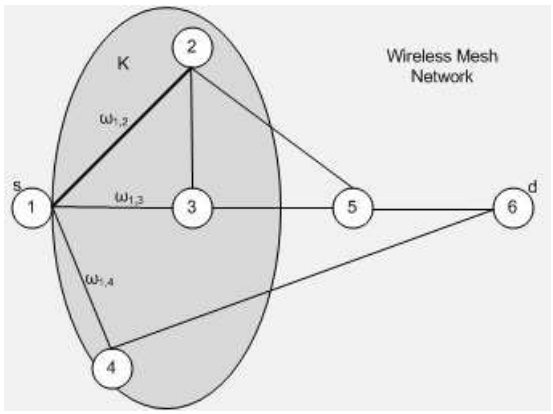

Figura 4.7: Algoritmo: Scelta dell'arco con peso maggiore

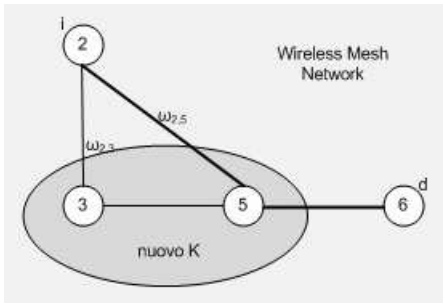

Figura 4.8: Algoritmo: Seconda e terza iterazione degli Step

4.5.1 Meccanismo di Flooding

Il meccanismo di flooding è composto da cinque parti:

- generare e trasmettere in broadcast un OGM
- ricevere e valutare un OGM
- *Bidirectional Link Check*
- *Neighbor Ranking*: stimare un ranking degli OGM per rilevare il best-link
- re-broadcast dell'OGM

Generazione e trasmissione in broadcast di un OGM

Ogni nodo, quindi, deve generare e trasmettere periodicamente un OGM per ogni B.A.T.M.A.N.-interface. Questi messaggi devono essere trasmessi ogni *ORIGINATOR-INTERVAL*. E' prevista anche la possibilità che un nodo voglia annunciarsi a una rete non-B.A.T.M.A.N., per fare questo deve appendere un HNA-extension per ogni rete in cui vuole annunciarsi.

Ricezione e valutazione di un OGM

Alla ricezione di un pacchetto B.A.T.M.A.N. generico un nodo deve eseguire dei *controlli preliminari* prima che il pacchetto sia ulteriormente trasformato e trasferito:

- *Version Check*: se un OGM contiene una versione diversa dalla propria versione interna, allora il pacchetto deve essere scartato
- *Source Check*: se l'indirizzo del mittente contenuto nell'OGM appartiene a una delle sue interfacce, allora il pacchetto deve essere scartato
- *Destination Check*: se l'indirizzo della destinazione è un indirizzo unicast, il pacchetto deve essere scartato
- se il flag Unidirectional Link è settato, allora il pacchetto deve essere scartato
- se, invece, l'OGM è stato ricevuto tramite un link bidirezionale e contiene un Sequence Number nuovo e non duplicato, allora questo OGM deve essere processato e ritrasmesso.

Bidirectional Link Check

Questa fase è utilizzata per verificare se il link rilevato da un vicino può essere utilizzato in entrambe le direzioni. Pertanto il numero di sequenza di ogni OGM auto-originato e ritrasmesso (re-broadcast) da un vicino diretto per ogni interfaccia deve essere salvato solo se tale numero di sequenza corrisponde a quello inviato con l'OGM in broadcast, se il flag di link bidirezionale

è settato e se è stato ricevuto tramite l'interfaccia per il quale l'OGM è stato generato.

Neighbor Ranking

Consiste in una serie di operazioni da eseguire al momento della ricezione di un OGM.
Dopo aver aggiornato il Packet Count, se il numero di sequenza dell'OGM è più recente del numero di sequenza attuale, il numero di sequenza attuale viene settato con quello appena ricevuto e viene aggiornato il TTL. Inoltre le sliding windows di tutti i link conosciuti verso l'Originator dell'OGM devono essere updatate per riflettere i nuovi limiti superiori e inferiori del Ranking Range.
Se invece la sliding window del link tramite il quale lìOGM è stato ricevuto contiene il numero di sequenza più alto, allora tale collegamento è il *Best Link* verso l'Originator dell'OGM.

Re-broadcast dell'OGM

Quando un OGM deve essere ritrasmesso alcuni campi del messaggio devono essere cambiati:

- il TTL deve essere decrementato di 1, se dopo il decremento assume valore 0 allora il pacchetto non sarà ritrasmesso ma verrà scartato
- il flag Is-Direct-Link deve essere settato se l'OGM è stato ricevuto da un Direct Link Neighbor (vicino diretto) e se viene ritrasmesso attraverso il link da cui è stato ricevuto.
- il flag Unidirectional Link deve essere settato se non è possibile ritrasmetterlo direttamente al vicino da cui è stato ricevuto l'OGM.

4.5.2 Routing

Al fine di mantenere aggiornata la tabella di routing di un nodo-B.A.M.A.N., il demone di routing registra traccia dei nuovi OGM in entrata e mantiene la

lista di tutti gli Originator che hanno spedito un OGM. Ogni voce definisce l'interfaccia di uscita e l'indirizzo IP del *next-hop direct-link neighbor* (vicino con link diretto a distanza di un hop) verso il prossimo Originator corrispondente.
Se un OGM proviene da un Originator sconosciuto o è diretto verso un host/rete sconosciuto, allora viene aggiunto alla tabella di routing e il vicino con ranking migliore sarà selezionato come gateway per la destinazione.
Nel caso un nodo non riceve un OGM da un Originator conosciuto entro i tempi di *WINDOW-SIZE* e *PURGE-TIMEOUT* il percorso è considerato scaduto e viene rimosso dalla tabella di routing. B.A.T.M.A.N. deve comportarsi in modo opportunistico quando viene eliminata una rotta, considerandone immediatamente una alternativa. Il tempo *PURGE-TIMEOUT* è relativamente breve e questo fa si che le informazioni di instradamento nella tabella vadano in overflow.

4.5.3 Gateway

Un nodo-B.A.T.M.A.N. con accesso a internet e routing capability può agire da gateway internet, annunciandolo nella GWFlags dell'OGM. Ovviamente se non dotato di accesso a internet il flag deve essere obbligatoriamente settato a 0, altrimenti la capacità di bandwidth, codificata come in figura 4.9.

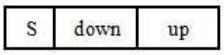

Figura 4.9: Campi del GWFlags

GWFlags è composto da 2 campi, uno di downstream e uno di upstream, interpretati come numeri binari e calcolati come segue:

$$downstreambandwidth = 32 * (S + 2) * 2^{down}\ kbit/sec \tag{4.2}$$

$$upstreambandwidht = ((up + 1) * (downstreambandwidth))/8\ kbit/sec \tag{4.3}$$

Un nodo può decidere quale gateway (tra quelli disponibile con GWFlags attivo) utilizzare sulla base di molti parametri, quali la velocità di download o qualità di connessione. Sarebbe utile che un utente potesse decidere una combinazione di velocità e qualità di connessione a lui idonea, ma questo permetterebbe agli utenti di accettare, ad esempio, connessioni pessime solo per avere una connessione particolarmente veloce, ma instabile.
Per questo motivo è stato deciso di non includere nel protocollo la possibilità di scegliere il gateway, ma di lasciare al protocollo stesso una scelta per non incorrere in problemi di *gateway unreachable*.
I pacchetti provenienti da internet verso un nodo non vengono sottoposti a nessun tipo di incapsulamento. Al contrario invece, i client incapsulano i dati internet in un datagram UDP e li manda al gateway selezionato. Quest'ultimo identifica i pacchetti in base al numero di porta dell'header UDP e lo manda verso la destinazione originale in una procedura è completamente StateLess.
Per l'incapsulamento, un client deve impostare l'header IP esterno e l'indirizzo di destinazione come come, rispettivamente il suo Originator Address e l'Originator Address del gateway che sta utilizzando, mentre la sorgente UDP esterna deve essere settata con il numero di porta del gateway (CHIEDERE).

4.6 Supporto a Interfacce Multiple

Il protocollo B.A.T.M.A.N. prevede il supporto di interfacce multiple per nodo. Spesso queste interfacce consistono in due schede di rete Wi-Fi che possono lavorare insieme o separatamente.
In presenza di un nodo di questo tipo il protocollo è in grado di ottimizzare il flusso di traffico per ottenere il massimo delle prestazioni. Le modalità di funzionamento possibili sono due: interface alternating mode e bonding mode.
Interface alternating mode è la modalità di default, adatta per la maggior parte delle situazioni e consiste nello switchare interfaccia WiFi con ogni hop per evitare lo *store & forward*.

Bonding mode, invece, usa tutte le interfacce nello stesso momento per spedire e ricevere dati contemporaneamente.
Con la presenza di interfacce multiple, inoltre, nella ricezione di un messaggio di re-broadcast un nodo deve essere in grado di capire se tale messaggio viene ricevuto sulla stessa interfaccia oppure no. Per ottenere questa informazione è stato introdotto il flag Is-Direct-Link negli OGM, che serve solo nelle trasmissioni one-hop e indica se il messaggio di ritorno di re-broadcast viaggia sulla stessa interfaccia che lo ha spedito inizialmente.
Ogni nodo, infine, può modificare i TTL di default per gli OGM delle sue interfacce allo scopo di limitare il range di hops a cui queste possono spedire OGM. Per fare un esempio, un nodo con 3 interfacce può essere configurato per spedire OGM con TTL alto solo da una di esse, mentre con TTL basso (es. TTL = 2) per le altre. In questo modo il nodo è ancora raggiungibile tramite l'IP dell'interfaccia con TTL alto ma non sovraccarica i nodi oltre due hops con la ritrasmissione delle altre due.

4.7 Sicurezza in B.A.T.M.A.N.

I protocolli di routing, in particolare quelli progettati per reti wireless, devono fare affidamento su informazioni provenienti da altri nodi della rete che viaggiano su canali non sicuri. Senza tecniche di crittografia e di autentificazione sarebbe essenzialmente un protocollo non sicuro.
Per aumentare la sicurezza tutti i dati che viaggiano nello strato wireless sono criptati, ma questo non risolve il problema se la rete è un WMN di grandi dimensioni non autenticata, come le reti cittadine, anche se il protocollo limita intrinsecamente molte delle modalità di attacco possibili.

4.7.1 Overflow delle Tabelle di Routing

Un host malintenzionato potrebbe inviare OGM che annunciano l'esistenza di nodi inesistenti per causare un overflow nelle tabelle di routing degli altri nodi e un eccessivo carico di CPU. Questo attacco può essere intercettato da semplici controlli di integrità sulle tabelle: se le voci oltrepassano un

tetto massimo, gli Originator con un basso numero di OGM devono essere eliminati.

4.7.2 Manipolazione delle Rotte

Un utente malintenzionato può anche effettuare il re-broadcast di OGM da un Originator esistente con un numero di sequenza valido continuativo che però in realtà non ha mai ricevuto tali OGM in fase di flooding al fine di manipolare il routing e reindirizzare il traffico verso se stesso.
Poiché le decisioni di routing sono basate su analisi statistiche sugli OGM ricevuti piuttosto che sulle informazioni contenute nei pacchetti, per effettuare questo attacco è necessario generare un alto numero di messaggi falsi allo scopo di vincere il ranking dei possibili nodi per quella destinazione. Vincere tale ranking però è veramente difficile e già questo fatto basta a limitare la maggior parte di attacchi di questo tipo. In più, inviare messaggi falsi con un numeri di sequenza differenti dalla dimensione delle Sliding Window causerà l'eliminazione dalle tabelle di routing, rendendo cosi questo attacco quasi impossibile in situazioni reali.
L'ultima possibilità per effettuare la manipolazione delle rotte consiste nell'inviare OGM fasulli per Originator esistenti con numeri di sequenza di poco superiori ai Sequence Number inviati da altri nodi, in modo da non uscire dalla Sliding Window e di avere alte probabilità di vincere il ranking. La possibilità di indovinare un numero di sequenza leggermente più alto di quelli reali ma non tanto alto da uscire alla Sliding Window però è veramente molto bassa e per vincere il ranking occorrerebbe mandarne in grande quantità. Anche in questo caso quindi, le possibilità di successo di questo attacco sono quasi nulle.

4.8 Valutazione Prestazioni

Dopo aver concluso l'effettiva descrizione di tutti gli aspetti del protocollo, passiamo ad analizzarne le effettive prestazioni allo scopo di capire se è veramente destinato a diventare uno standard per reti wireless ad-hoc e

mesh. L'analisi sarà sempre rivolta a capire se B.A.T.M.A.N. sia superiore al suo attuale rivale O.L.S.R., analizzando in dettaglio le caratteristiche principali che determinano le prestazioni di un protocollo: *Overhead*, *Throughput*, *Carico di CPU e Memoria* e *Packet Loss*.
Allo scopo di avere un dato oggettivo per valutare tali aspetti del protocollo ci rifacciamo a un esperimento eseguito da David Johnson, Ntsibane Ntlatlapa e Corinna Aichele descritto nell'articolo *A simple pragmatic approach to mesh routing using B.A.T.M.A.N.*[8] su una rete mesh composta da 49 nodi.

4.8.1 Overhead

La capacità di un protocollo di routing di sostenere reti di grandi dimensioni (*scalabilità*) dipende dal controllo del traffico di *overhead*. Questo termine indica quella parte di trasmissione composta dalle informazioni addizionali che i router devono scambiarsi per potere eseguire i loro algoritmi. Tali dati non sono quindi quelli che un mittente spedisce a un destinatario, ma sono quelli che i router della rete si scambiano per capire come fare arrivare ciò che un nodo vuole spedire al destinatario scelto. Essendo però lo stesso canale trasmissivo, questo overhead si somma alle normali trasmissioni incidendo sulla quantità di banda utilizzabile dai componenti della rete per lo scambio di dati vero e proprio soprattutto in reti wireless con capacità limitate.
Il meccanismo di flooding di OGM di B.A.T.M.A.N. è, a prima vista, poco scalabile. In una rete composta da N nodi a intervalli regolari si compie un flooding di esattamente N messaggi, poiché ogni nodo spedisce il suo OGM. Alla ricezione degli OGM, poi, ogni nodo effettua il re-broadcast, portando, in una situazione di perdita dei pacchetti (*Packet Loss*) ideale nulla, il numero di tutti gli OGM spediti in un intervallo ammonta, circa, al numero di N^2,in quanto ogni nodo inoltra gli OGM ricevuti ad altri $N-1$ nodi (non N).
Tuttavia in una situazione wireless e con possibilità di gestione di host mobili, le perdite di pacchetti sono piuttosto frequenti e non tutti gli OGM vengono ricevuti. La dimensione di tali OGM, inoltre, in questo protocollo

è molto più piccola che nei protocolli tradizionali, di conseguenza occupano una quantità molto più limitata di banda.
O.L.S.R., pur utilizzando un metodo simile per il flooding dei pacchetti di controllo, invia, anziché un pacchetto dalla grandezza media di 52 byte (gli OGM), il pacchetto HALLO e i pacchetti TC.
In figura 4.10 e 4.11 vengono messi a confronto la quantità di pacchetti dei due protocolli sia in uscita che in entrata. Per leggere correttamente i grafici occorre tener conto che il vero peso per un protocollo è l'overhead di uscita.
Dai grafici si può notare come pur essendo l'overhead in entrata lievemente

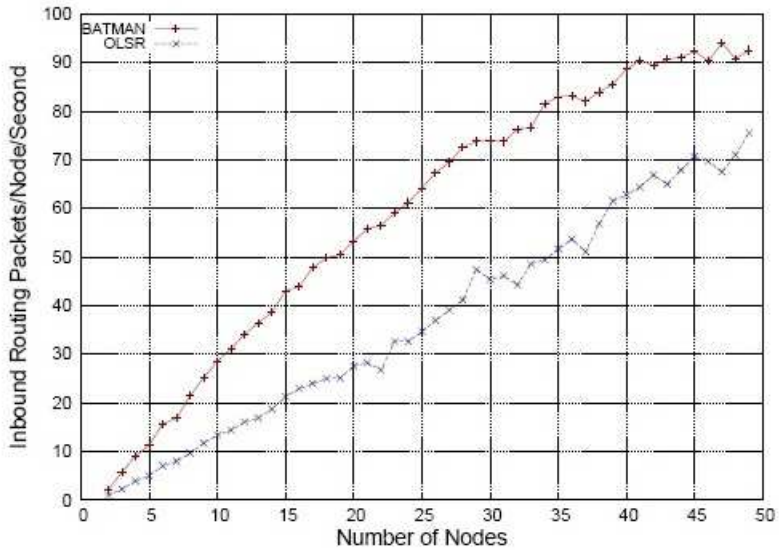

Figura 4.10: Overhead in entrata in relazione al numero di nodi della rete

maggiore in B.A.T.M.A.N., quello in uscita è molto alto per un basso numero di nodi ma si stabilizza subito da 15 host in su, arrivando a superare in prestazioni O.L.S.R. in reti composte da più di 45 nodi.
Ciò è la conferma di una buona riuscita dell'intento del protocollo, per quanto riguarda l'overhead, che sin dall'inizio si è sempre proposto per reti di grandi dimensioni.
Inoltre per conoscere la vera quantità di traffico di controllo di un protocollo occorre analizzare anche, come detto in precedenza la grandezza dei pacchetti di controllo. Al crescere delle dimensioni della rete O.L.S.R. necessita di

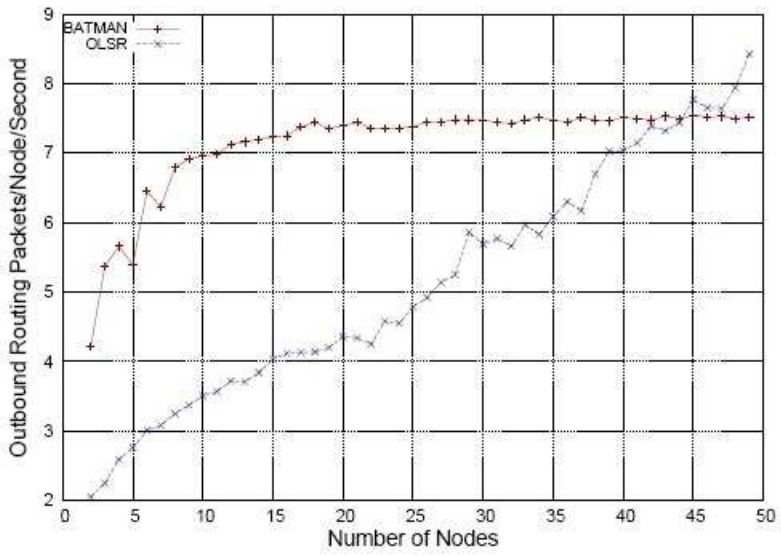

Figura 4.11: Overhead in uscita in relazione al numero di nodi della rete

includere nei TC e inviare tutte le volte l'intera mappa di routing, che verosimilmente crescerà all'aumentare dei nodi. B.A.T.M.A.N. incorpora solo le informazioni sul migliore next-hop, che è un'informazione che non dipende dalla grandezza della rete. Per calcolare l'overhead totale occorre moltipli-

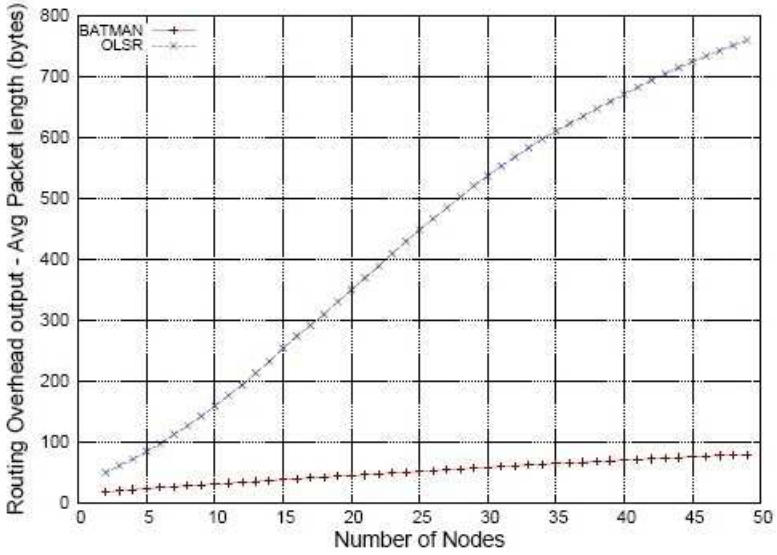

Figura 4.12: Grandezza dei pacchetti di controllo in relazione alle dimensioni della rete

care la lunghezza di ogni pacchetto per il numero di messaggi di controllo che escono da un nodo in un secondo. Nell'esperimento, tramite tale calcolo, già in una rete di piccole dimensioni come quella del test eseguito l'overhead totale di B.A.T.M.A.N. è di dieci volte inferiore a quello di O.L.S.R.

4.8.2 Throughput

Con il termine *Throughput* si intende la capacità di trasmissione effettivamente utilizzata. Occorre fare attenzione alla differenza tra throughput e capacità trasmissiva del link (che si esprimono tutti in bit/sec): il primo esprime la frequenza trasmissiva massima alla quale i dati possono viaggiare e dipende esclusivamente dalla quantità di informazione che è immessa nel canale di trasmissione. Il secondo è la quantità di dati media che circola nel canale trasmissivo. In parole semplici lo throughput è l'indice di effettivo utilizzo della capacità totale trasmissiva del link.
Potremmo definire questo aspetto come il rendimento di una rete. Nell'esperi-

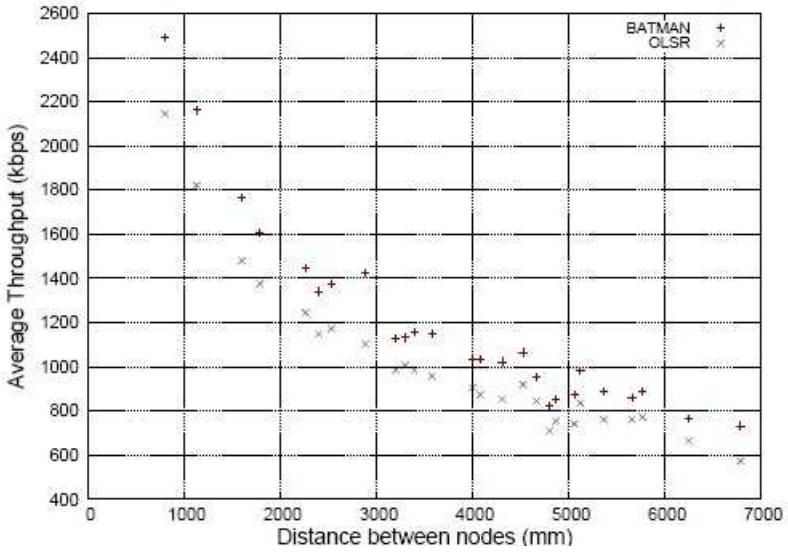

Figura 4.13: Throughput in relazione alla distanza tra i nodi

mento, B.A.T.M.A.N. migliora lo throughput di O.L.S.R. di circa il 15%, indipendentemente dalle dimensioni della rete. Questo dato rivela, in parole semplici, che B.A.T.M.A.N. trova percorsi migliori rispetto al suo rivale.

4.8.3 Packet Loss

La perdita dei pacchetti, nonostante i due protocolli lavorino su reti instabili quali le WMN, è relativamente bassa. Benché se la differenza è minima, in questo aspetto, O.L.S.R. batte il suo rivale diretto di circa un 1%.

Bisogna ricordare, però, che in B.A.T.M.A.N. il Packet Loss è un fattore

Routing Protocol	Forward hop count	Symm links (%)	Seconds per Route change	Packet loss (%)	Delay (ms)	Throughput (kbps)	No link (%)
BATMAN	1.88	28	25.64	2.63	7.61	1378.35	1.11
OLSR	2.26	61	12.20	1.68	17.39	1177.92	0.60

Figura 4.14: Tabella riassuntiva con attenzione sul Packet Loss

basilare per calcolare la velocità e la qualità del percorso, fattore questo che, ne da conferma lo Throughput, si è rivelati vincente per il calcolo di percorsi migliori rispetto a quelli calcolati da O.L.S.R.

4.8.4 Carico CPU e Memoria

Un grande vantaggio delle WMN è il suo basso costo di installazione, questo fatto fa si che i router che compongono la rete siano dotati in genere di poca potenza di CPU e poca memoria RAM.
In questo scenario, quindi, l'utilizzo delle risorse diventa un fattore molto importante per la valutazione oggettiva di un protocollo di routing.
Il carico di CPU deriva sostanzialmente da due fattori: la complessità dell'algoritmo che calcola i percorsi ottimali e dal numero di pacchetti di cui il processo di routing ha bisogno per avere dati sufficienti con cui fare funzionare l'algoritmo.
Nei test B.A.T.M.A.N. ha esibito un maggior numero di pacchetti in uscita in una rete inferiore ai 45 nodi, ciò porterebbe ad aspettarsi un miglioramento rispetto a O.L.S.R. solo in WMN composte da più di 45 nodi. L'algoritmo B.A.T.M.A.N. però è molto più semplice rispetto a quello del rivale e infatti il carico di CPU risulta vantaggioso già in reti di dimensione maggiore ai 6 nodi. In generale, già in una rete piccola di 49 nodi, O.L.S.R. usa il 44% in

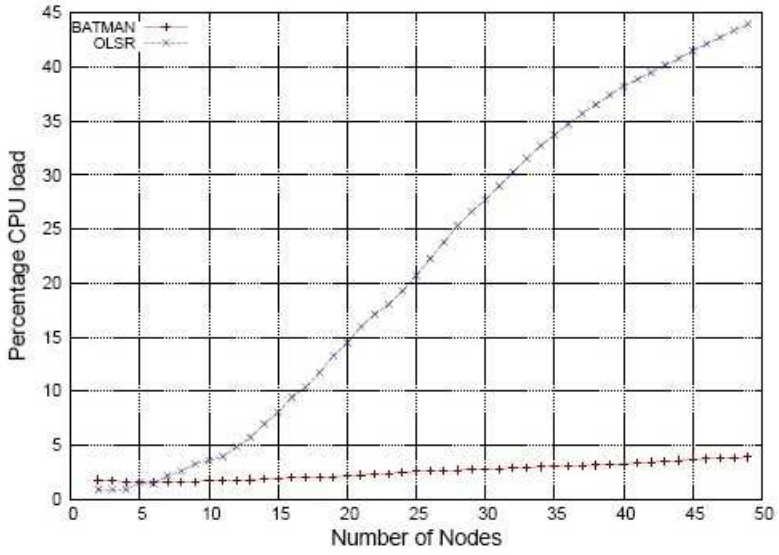

Figura 4.15: Carico di CPU in relazione al numero di nodi

più di CPU rispetto a B.A.T.M.A.N., possiamo quindi capire quale vantaggio porterebbe il nostro protocollo in una rete di grandi dimensioni cittadina.
Occorre notare, inoltre, come l'utilizzo di CPU e la grandezza dei pacchetti (figura 4.12) crescano essenzialmente di pari passo, ciò è dovuto al fatto che la CPU necessaria per elaborare informazioni grandi come i pacchetti O.L.S.R. che tengono traccia di tutta la topologia della rete, è maggiore rispetto all'elaborazione di pacchetti di piccola dimensione come gli OGM.
La grandezza dei pacchetti è anche un fattore decisivo per quanto riguarda il consumo di memoria. O.L.S.R. infatti, oltre a dover memorizzare sempre l'insieme dei tutti percorsi della rete, deve anche memorizzare in ogni router i pacchetti di controllo. Nonostante ciò, a causa di un flooding più frequente, B.A.T.M.A.N. si rivela la scelta migliore solo in reti composte da più di 30 nodi, ma si rivela comunque vantaggiosissimo rispetto al rivale in reti di grandi dimensioni, rispettando le prospettive di scalabilità che sono alla base del protocollo (figura 4.16).

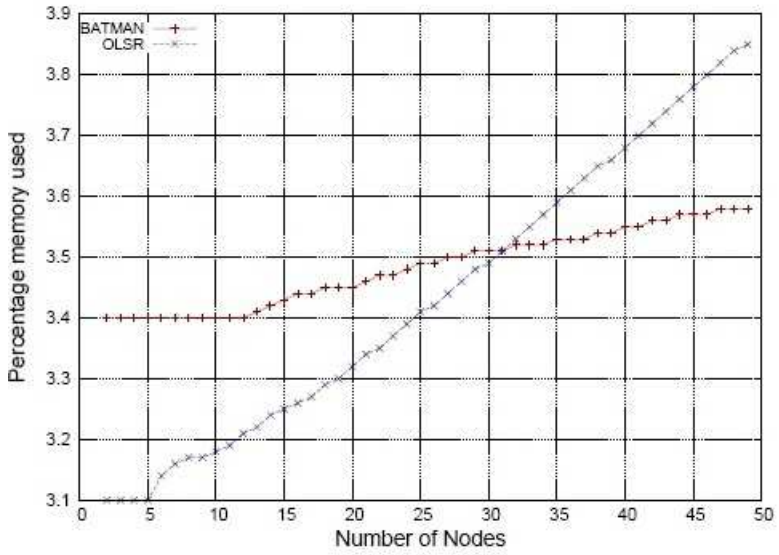

Figura 4.16: Carico di Memoria in relazione al numero di nodi

4.8.5 Conclusione sulle Prestazioni

Come si è potuto osservare, per reti di dimensione superiori a 50 nodi B.A.T.M.A.N. si rivela superiore a O.L.S.R. [9] in quasi tutti i parametri di rendimento. La semplice filosofia di non raccogliere più informazioni di quante se ne possano utilizzare, ma solo sui nodi vicini rende la computazione più efficiente.
O.L.S.R. ha mostrato solo due piccoli vantaggi: la perdita di pacchetti inferiore dell'1% e la presenza di link guasti inferiore dello 0.5%.
Per quanto riguarda i link simmetrici, B.A.T.M.A.N. è riuscito a usare solo il 28% dei link unidirezionali presenti, contro il 61% del rivale.
La minore necessità di risorse, infine, è di buon auspicio per le reti mesh attuali, composte in gran parte da nodi mobili o situati in posti scomodi alimentati a batteria. Il fine futuro sarà quello di usare fonti di energia rinnovabili come pannelli solari, per ricaricare tali batterie.
Infine, in tutti i test proposti, non si sono mai verificati casi di Routing Loop.

4.9 Esperienze Reali

B.A.T.M.A.N., oltre a essere un protocollo ancora sostanzialmente nuovo e ancora in via di evoluzione, ha anche la caratteristica di portare grandi vantaggi soprattutto se impiegato in reti di grandi dimensioni come reti cittadine o reti universitarie. Per reti piccole, per quanto si comporti comunque meglio di O.L.S.R., il miglioramento delle prestazioni non è sufficiente per giustificare un cambiamento di tecnologia che comporterebbe una spesa e un consumo di tempo notevoli.
Per questo motivo le reali applicazioni del protocollo nella realtà sono ancora poche. Le esperienze più significative sono in reti cittadine tedesche in città come Berlino, Goerlitz, Halle, Magdeburgo, Lipsia e Weimar.
Alla festa di rilascio di B.A.T.M.A.N.-0.2 (versione 2011.2.0) nel Giugno 2011 al C-base di Berlino gli amministratori delle WMN delle città sopra citate hanno riferito la loro esperienza. La conclusione è stata che, in confronto a O.L.S.R., B.A.T.M.A.N. funziona talmente bene che le versioni di O.L.S.R. che inizialmente funzionavano in parallelo a B.A.T.M.A.N. in fase di test non saranno più abilitate.
Anche l'Italia, infine, da circa un anno ha fatto la sua prima esperienza con il protocollo nella città di Pisa.

Capitolo 5

Conclusioni

Era nelle intenzioni di questo lavoro esporre un quadro dettagliato di questo nuovo protocollo che con molta probabilità diventerà il nuovo standard per il routing su reti mesh e cercare la spiegazione di tale futuro cambiamento di protocolli in un analisi delle prestazioni nei confronti dell'attuale standard. Come si è potuto osservare B.A.T.M.A.N. nasce da un idea diversa di routing basata sull'intuizione che le informazioni necessarie per connettere due host, ad ogni nodo, possono essere limitate alla sola conoscenza della successiva tappa nella strada verso il destinatario, permettendo l'utilizzo di router meno preformanti e quindi meno costosi. Unito al fatto che si tratta di un protocollo Open Source si capisce come nella realtà odierna sia un ottimo incentivo alla futura larga distribuzione per le nuove reti di tipo mesh.
Nonostante sia ancora in fase di sviluppo, B.A.T.M.A.N. ha un algoritmo sicuro e performante, con un funzionamento intrinseco che agevola la sicurezza nella rete. Inoltre si è visto come, per reti grandi, le prestazioni, in tutti i parametri di misura, sono superiori a quelle dei protocolli utilizzati attualmente.
In un mondo dove le reti wireless stanno man mano trovando utilizzo non solo in ambito domestico, ma anche sempre più spesso reti metropolitane, l'ideologia di B.A.T.M.A.N. si pone, con successo, come scopo di risolvere i problemi di scalabilità che si sono sempre incontrati fin'ora in network di grandi dimensioni. La valutazione delle prestazioni ha provato che su reti

grandi il miglioramento rispetto a O.L.S.R. è tale da giustificare il pronostico che lo pone come suo successore in un futuro veramente prossimo.
La conclusione di questa tesi risiede proprio in questo pronostico che è già in fase di attuazione in città estere come Berlino o italiane come Pisa con risultati eclatanti.

Bibliografia

[1] Y. Yang, J. Wang, and R. Kravets, "Designing routing metrics for mesh network", *In WiMesh*, 2005

[2] W. Kiess and M. Mauv,e "A survey on real-world implementation of mobile ad-hoc networks", *Ad Hoc Networks, vol. 5, no. 3, pp. 324-339*, 2007.

[3] IETF Draft, [online], http://datatracker.ietf.org/doc/draft-ietf-manet-dlep/

[4] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," *Multi Topic Conference. IEEE INMIC 2001. Technology for the 21st Century. Proc. IEEE International*, pp. 62–68, 2001.

[5] C.E. Perkins and E.M. Royer, "Ad-hoc on-demand distance vector routing," *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, vol. 2, pp. 90–100*, 1999.

[6] Open-Mesh.net "B.A.T.M.A.N. (better approach to mobile ad-hoc networking)". [Online]. Available: http://www.open-mesh.net/

[7] Neumann A., Aichele C., Lindner M and Wunderlich S. "Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.), Internet-Draft", *http://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00*, 2008

[8] D. Johnson, N. Ntlatlapa, and C. Aichele, "A simple pragmatic approach to mesh routing using BATMAN," *2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries, CSIR, Pretoria, South Africa*, 2008.

[9] L.Barolli, M. Ikeda, G. De Marco, A. Durresi and F. Xhafa, "Performance Analysis of OLSR and BATMAN Protocols Considering Link Quality Parameters", *2009 Conference on Advanced Information Networking and Applications, Bradford*, 2009.