Università degli Studi di Padova

Dipartimento di Ingegneria
dell'Informazione
Corso di Laurea in Ingegneria delle
Telecomunicazioni

Algoritmi di routing su reti mesh wireless

Laureando:
Ibrahim Yankine

Relatore:
Prof. Andrea Zanella

Anno accademico 2012/2013

Alla mia famiglia e, a
Sonia che mi sostiene
sempre!

Indice

Capitolo 1

Introduzione

In questi ultimi anni si è assistito progressivamente all'aumento di dispositivi elettronici capaci di effettuare collegamenti veloci a reti wireless. Di pari passo è seguito l'aumento della domanda d'utenza da parte di un numero sempre più elevato di persone. Si capisce subito che implementare studi ed ideare metodi che portano alla realizzazione di una tipologia di rete wireless ottimale, e capace di supportare un traffico di dati enorme, a velocità sempre più elevate pur mantenendo una certa affidabilità in termini si sicurezza, diventa una sfida fondamentale per il prossimo futuro. Una delle sfide in concreto che pongono diversi limiti alla realizzazione di reti è rappresentato dalla posa di cavi su vaste aree. Aree in cui è praticamente impensabile realizzare strutture, come montagne e foreste, dove il costo sia economico che ambientale sarebbero insopportabili o contro producenti. Soprattutto se ci affida alle tecnologie attualmente di maggiore utilizzo.

1.1 Obbiettivo della tesi

Nell'ambito di questa tesi ci si concentra nel definire una soluzione per reti wireless, attualmente argomento di numerosi studi e ricerca. Le Wireless Mesh Network (WMNs) spiegando cosa sono, quali sono in vantaggi che offrono rispetto alle reti attuali, gli utilizzi, si discuterà del lavoro che sta svolgendo il gruppo IEEE802 per la definizione di uno standard per tali reti. Soprattutto si fornirà l'analizzi di alcuni degli algoritmi di routing più comunemente utilizzati quali: OLSR, AODV, enfatizzando maggiormente HWMP [di IEEE802.11s] e B.A.T.M.A.N [di Freifunk]. Sara in fine fornito un esempio di casi di studio di reti WMNs: la rete Roofnet (MIT).

1.2 Breve panoramica

La Wireless Mesh Network (WMNs) è una rete a maglia decentralizzata composta principalmente da nodi (router mesh e client mesh) connessi reciprocamente da un link, cioè non esiste più la relazione del tipo server/cliente presente nella topologia a stella delle reti wireless tradizionali. Essa è invece cooperativa ed ogni nodo è in grado di comunicare direttamente con qualsiasi altro nodo, vicino (one-hop) o lontano (multi-hop). Questo tipo di rete ha un comportamento dinamico auto-configurante e robusto, i nodi infatti stabiliscono e man-

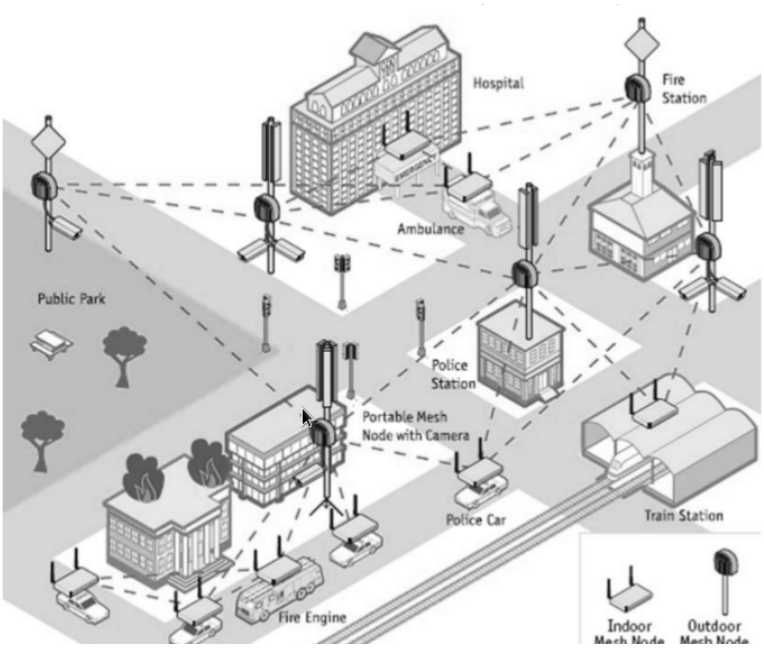


Figura 1.1: Rete Wireless Mesh

tengono automaticamente la connettività al loro interno, trovando se necessario una soluzione in modo autonomo quando parte delle rete è danneggiata. Le funzionalità gateway/bridge presenti in alcuni routers mesh permettono anche l'integrazione delle WMNs con molte tipologie di reti: Wi-Fi, Wi-Max, HiperLAN, reti di sensori e reti cellulari. Permette cosi la realizzazione di alcune applicazioni come la banda larga su wi-fi, sistemi di sorveglianza e svariate reti wireless metropolitane (università, trasporto, negozi, ecc). Le caratteristiche principali di una WMN devono dunque essere:

- Fornire accesso continuo ai propri utenti.
- Mantenere le connessioni degli utenti ed effettuare rapidamente il passaggio da un punto di accesso all'altro quando gli utenti si spostano nell'area di copertura della rete.
- Essere facile da sostituire, spostare e espandere.
- Essere robusto e continuare a funzionare anche se una parte della rete non è disponibile.
- Essere conveniente, cioè, deve funzionare bene utilizzando i routers wireless a basso costo già presenti in commercio.

Grazie a queste caratteristiche la rete WMN è scalabile, flessibile ed affidabile e può anche essere costruito integrando una struttura cablata, aggiungendo un router per volta, cosi da aumentare la flessibilità della struttura stessa. Cosa di grande rilevanza, la vasta diffusione di strumenti conformi allo standard IEEE 802.11 fanno sicché i costi per la realizzazione di una WMN siano di molto contenuti. Purtroppo attualmente non esistono standardizzazioni che permettano di sfruttare i vantaggi delle reti WMNs e nascono diversi problemi per quanto riguarda i protocolli in uso nella copertura di vaste aree. L'istituto IEEE

802. tramite una Task Group sta cercando di realizzare una standardizzazione anche per le WMNs denominata IEEE 802.11s ma siamo ancora lontani dalla versione definitiva.

Capitolo 2

Tipologie di Reti

Il secolo scorso fu l'inizio di una grande presa di conoscenza per quello che riguarda la comunicazione di massa attraverso la manipolazione dell'informazione. Nascono o si specializzano cosi le tecnologie dell'informazione. Difatti vengono introdotti le trasmissioni radio regolari, le reti telefoniche, le prime reti televisive fino all'introduzione dei primi calcolatori. La necessità di trasferire dati velocemente e in modo sicuro ha spinto ad una rapida evoluzione di quest'ultimi portandogli da strutture monolitiche, enormi e statici a strutture più piccoli facilmente manovrabili e soprattutto alla portata di moltissimi. A differenza dei primi calcolatori dove per trasferire i dati si doveva smontare e trasportare fisicamente le unità dati come nastri e dischi da un calcolatore all'altro, oggi i calcolatori sono interconnessi formando delle reti. La più grande di queste è internet che realizza interconnessioni al livello globale. Grazie questo sviluppo continuo si sono realizzate diverse tipologie di reti. Di queste parleremo delle reti cablate e le reti wireless.

2.1 Reti Cablate

La rete cablata è caratterizzato da una grande velocità di comunicazione, grande affidabilità e sicurezza. Si realizza con costi abbastanza ridotti e garantisce buone prestazioni. Infatti per moltissimo tempo ed ancora oggi è il mezzo più utilizzato per interconnettere dispositivi statici soprattutto al livello aziendale, amministrativo e quello che viene chiamato ultimo miglio (il tratto di rete che connette l'utenza domestica alla rete). Con l'andare del tempo, la crescita esponenziale della tecnologia dei dispositivi mobili e del numero di utenti che ha necessita di accedere ad una rete internet connessa, scalabile e soprattutto flessibile, questo tipo di rete ha presto svelato i propri limiti. Si pensi ad esempio al caso estremamente limitante di accedere ad internet con il proprio computer portatile oppure dal cellulare collegandosi alla rete cablata i.e Ethernet. La rete cablata è caratterizzato da un staticità già in fase di progettazione, costringendo ad una struttura rigida che rende difficile sia il mantenimento che aggiornamento. Se poi ci si concentra sul fattore ambientale è chiaro che bisogna trovare soluzioni meno invasive ma sicuramente più flessibili e scalabili. La soluzione allora arriva dall'utilizzo delle reti wireless.

2.2 Reti Wireless

I problemi riscontrati nelle reti cablate, scarsa flessibilità, immobilità della struttura, poco scalabilità e difficoltà nello scavare in certe zone, hanno reso necessario l'introduzione di una rete che non facesse più affidamento al cavo fisico o almeno in parte. L'enorme diffusione delle apparecchiature elettroniche come portatili, PDA, notebook, tablet, cellulari hanno fatto del proprio cavallo di battaglia la mobilità inducendo cosi molti ricercatori e aziende ad investire nella tecnologia della rete wireless.

La rete wireless (dall'inglese senza fili) sfrutta come mezzo di trasmissione le onde radio a determinate frequenze, solitamente quella a 2,4 Ghz, permettendo cosi all'utenza di connettersi senza bisogno di un supporto fisico ma bensì restando nell'area di copertura radio (hot spot) generato da un dispositivo wireless chiamato punto di accesso (Access Point). La tecnologia più diffusa per l'implementazione di una rete wireless è definita nello standard IEEE 802.11 che si occupa di definire al livello fisico (PHY) e Data Link le caratteristiche che tale rete deve avere. Si tratta di uno standard in evoluzione con continui aggiornamenti che [3] a partire dal 1997 con la versione 802.1y specificava velocità di trasmissione comprese tra 1 e 2 Mb/s e utilizzava raggi infrarossi o onde radio a frequenza di 2,4 GHz per la trasmissione. Ad ora siamo alla versione 802.11n che include la possibilità di utilizzare la tecnologia MIMO (multiple-input multiple-output). Questo consentirà di utilizzare più antenne per trasmettere e più antenne per ricevere incrementando la banda disponibile e permettendo di raggiungere velocità di trasmissione di 100 Mb/s. Le reti wireless si possono distinguere in due grosse categorie: reti wireless managed e reti wireless Ad-hoc.

2.2.1 Reti Wireless Managed

Una rete wireless managed (in italiano rete strutturata) è formata da una struttura gerarchica con topologia a stella dove il nodo centrale svolge il ruolo di server controllando e gestendo la connessione tra gli altri nodi del suo insieme; i clients. L'insieme del nodo centrale che solitamente è un *Access Point* (AP) e dei client viene chiamato *Basic Service Set* (BBS). Questo insieme tramite l'AP è collegato ad un rete cablata che fornisce i servizi, la *Distribution System* (DS). La rete che raggruppa tutti i BBS prende il nome di *Extended Service Set* (ESS) e viene identificato da un *Service Set Identifier* (SSID), un nominativo di rete. Il principale svantaggio di questa rete risiede nella sua topologia in quanto il mancare dell'AP impedisce completamente la connessione tra i clients di tale sottoinsieme ma anche con i restanti nodi della struttura.

2.2.2 Reti Wireless Ad-Hoc

La realizzazione di questo tipo di rete wireless avviene in modo spontaneo e si caratterizza dall'estrema mobilità dei nodi al suo interno. Infatti a differenza del tipo managed questa rete non fa affidamento ad un supporto cablato. Tutti i suoi nodi sono dispositivi wireless interconnessi in modo paritario, non è più necessario la presenza di un AP in quando ogni nodo si comporta come server o come client a seconda delle necessità della rete, stabilendo cosi una

connessione *peer-to-peer*. Qui l'insieme dei nodi è definito *Indipendant Basic Service Set* (IBSS). Una rete siffatta è adatto al collegamento di poche utenze perché le onde radio emesse da tutti i nodi si sovrappongono causando interferenza che peggiora il throughput. Le rete Ad-Hoc tradizionali hanno dei limiti caratterizzanti: vincolo di consumo di energia nei nodi, forte interferenza tra nodi, scarsa capacità computazionale. si possono superare questi limiti implementando la più recente tecnologia delle reti mesh wireless.

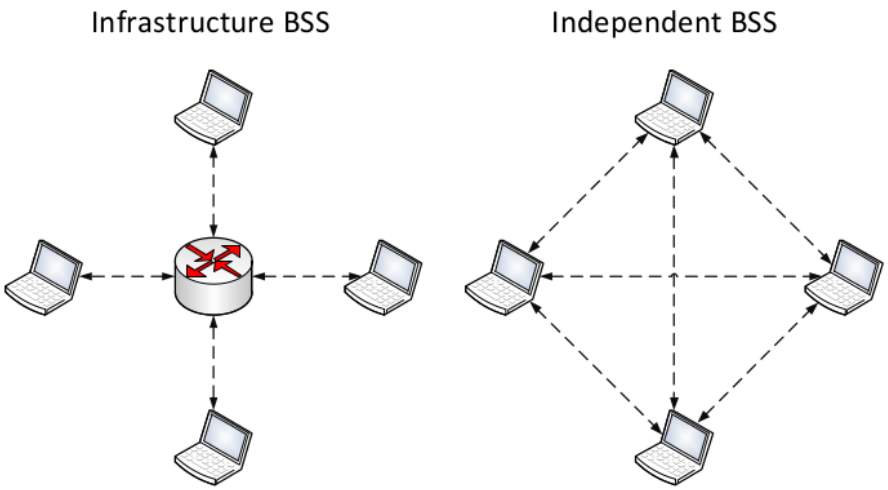


Figura 2.1: Reti wireless tradizionali

Capitolo 3

Reti Mesh Wireless (WMNs)

Una rete WMN si può considerare come una generalizzazione tecnologica e strutturale di una rete wireless Ad-Hoc col vantaggio che molto del traffico dati viene gestito dai routers alleggerendo il carico sui clients, sfrutta la comunicazione multi-hop, la capacità di riadattamento in automatico del routing ai cambiamenti che avvengono nella topologia, costituisce una forte base per un sistema scalabile ed è inoltre compatibile con diverse altre tecnologie. Dal punto di vista topologico questa rete sembra essere un misto tra una rete managed ed una rete Ad-hoc che implementa una diversa tecnologia e consiste nella interconnessione di routers mesh statici o quasi che formano la struttura centrale (backbone) a cui possono connettersi i clients ed eventuali gateway per il ponte con internet. Cosi facendo la copertura radio all'interno della struttura aumenta notevolmente grazie all'aggregarsi di nuovi dispositivi.

3.1 Architettura

In base a come sono connessi tra di loro i vari nodi della rete si possono distinguere tre forme di architettura mesh [1]:

- WMNs STRUTTURATA/BACKBONE: In quest'architettura i routers mesh formano una infrastruttura connessa dove i routers possono essere dotati di funzionalità gateway/bridge e possono eseguire connessioni sia cablate che wireless. La particolarità in questo caso consiste nel fatto che la rete formata dai routers (backbone) funge da supporto per i clients che per potersi connettere devono agganciarsi ad un hot-spot. Tra i routers c'è un rapporto reciproco ed essi in base alle informazioni che ricevono dalla rete implementano le funzionalità di shelf-healing, shelf-managing e shelf-configuring.

- WMNs DI CLIENTS: L'architettura basata sull'interconnessione dei clients è priva di routers mesh e gateway. La connessione avviene in modalità peer-to-peer dove i dati viaggiano da nodo a nodo in modalità uni-hop o multi-hop. I clients sono solitamente: telefoni cellulari, PDA, GPS mobile, tablet, notebook e varie. Dovendo comportarsi come strumenti che a loro volta instradano i dati verso altri nodi i requisiti di tecnologia dei clients aumenta leggermente rispetto a quelli tradizionali ed essendo una rete distribuita i costi per le installazioni diminuiscono notevolmente

rispetto alle reti cablate. Di fatti questa si può considerare una rete Ad-hoc, un esempio è MANET.

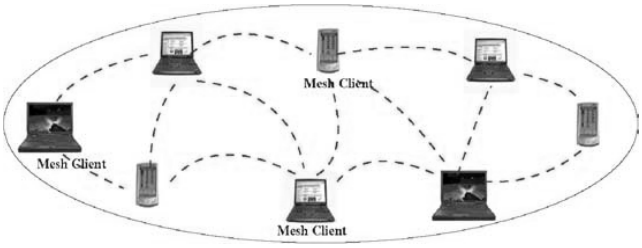


Figura 3.1: Mesh fatto sullo di clienti

- WMNs ibrida: È la combinazione tra le prime due, dove i clients possono connettersi sia tra di loro che con i routers mesh della rete arrivando al nodo radice (gateway) tramite connessione multi-hop. La parte infrastrutturale/backbone permette supporto a diverse tecnologie (WI-FI, WiMAX, HiperLAN ecc) fornendo forte scalabilità. Invece i clients contribuiscono ad ingrandire la copertura radio via via che ne arrivano nuovi.

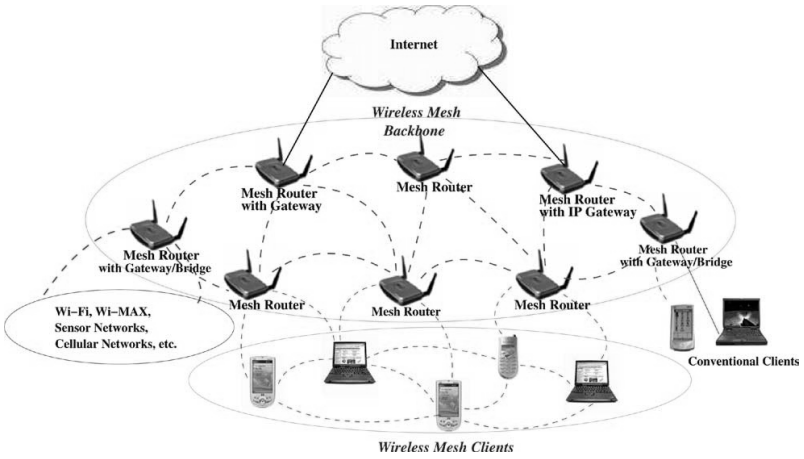


Figura 3.2: Mesh ibrida, configurazione abituale

3.2 Applicazioni

- Uso militare o civile (ricerca) per connessione di veicoli e strumentazione mobile in aree isolate
- Collegamento alla rete internet per zone rurali di difficile accesso
- Reti di sensori per la rivelazione di varie misure
- Realizzazione di infrastrutture a basso costo economico ed ambientale
- Riduzione del Digital Divide tramite messa a disposizione di risorse socialmente utili (OLPC – One Laptop Per Child)

- Sorveglianza di punti sensibili mediante l'impiego di telecamere o altri dispositivi
- Reti dinamiche per un pronto intervento in situazioni di catastrofe naturali Altri

3.3 Vantaggi e svantaggi

Vantaggi

- Auto-organizzazione e auto-configurazione. Le WMNs hanno una architettura robusta e flessibile che non dipende dall'implementazione e dei protocolli. Questo riduce di molto i tempi di set-up e costi di mantenimento oltre ad incrementare le performance della rete. Caratteristiche che permettono a chi (Provider) realizza la struttura, di poter modifica ed adattare la rete in base a diverse necessità.
- I costi di installazione, si riducono notevolmente grazie soprattutto alla capacità dei routers mesh di implementare trasmissione di tipo multi-hop. Si pensi ai costi d'installazione di router mesh dislocati su un area vasta che si interconnettono tramite trasmissione radio, si auto-configurano e calcolano in maniera efficiente una soluzione ad un guasto nella topologia e ai costi per la posatura di cavi per connettere diversi AP nelle rete wireless tradizionali per rendersi conto dei benefici economici delle WMNs rispetto a quest'ultimi.
- L'affidabilità è garantito dalla possibilità di poter operare su differenti percorsi all'interno della rete per raggiungere un stesso nodo. Questo permette di attuare un meccanismo di bilanciamento che interviene in caso di bottleneck e in generale di congestione del traffico. La rete interviene autonomamente per sopperire ai guasti interni.
- La scalabilità nelle reti wireless mesh è molto importante in quanto l'aumentare di nodi implica anche l'aumentare dei percorsi che si possono scegliere. Difatti questo può comportare una crescita delle performance. Diversamente dalle reti tradizionali dove un aumento solitamente ne comporta un peggioramento.
- L'interoperabilità, le WMNs hanno collegamenti multi-point-to-multi-point (MTM) che sfruttando la compatibilità con diversi standard già esistenti come: WiMAX, HiperLAN, Wi-Fi, Zigbee, Bluetooth, Sensor, MANET, etc.

Svantaggi

- Lo svantaggio principale di questa tecnologia è la mancanza di un protocollo definito, il che attualmente costringe ad utilizzarne altri, senza raggiungere il pieno potenziale di questo tipo rete. Le tecnologie a cui si appoggia sono IEEE 802.11, 802.15
- Aumentano le interferenze radio in quanto i nodi sono relativamente vicini.

- La natura adattiva e dinamica della rete rende molto difficile implementare e garantire servizi QoS (Quality of Service).
- L'implementazione per la gestione della sicurezza non è tuttora definito a pieno.

Capitolo 4

IEEE 802.11/s

La tecnologia WLAN più diffusa attualmente è senza dubbio rappresentato dall'IEEE 802.11 che definisce tutte le caratteristiche che devono avere i livelli PHY e Data Link dell'Open System Interconnection (OSI). Nel paragrafo wireless managed le gerarchie WLAN sono organizzate partendo dal basso da delle piccole unità o celle denominate BBS. Le BBS sono controllate da punti centrali detti AP che hanno una caratteristica molto limitante quella della restrizione sulla potenza di trasmissione. Questo implica immediatamente un limite di mobilità dei clients all'interno del BBS in quanto la loro connessione dipende dalla copertura degli AP. Come già detto la copertura può essere estesa collegando fra loro gli AP tramite i mezzi DSM (Distribution System Medium) che sfruttano tecnologia 802.3 (Ethernet) formando cosi gli ESS. Lo schema appena descritto comporta pero l'uso di cavi nella DSM che per la loro natura risultano costosi e impraticabile in certe condizioni ambientali. Da questa considerazione il gruppo IEEE tramite una TaskGroup (TGs) ha deciso di realizzare una estensione dello standard 802.11 con lo scopo di definire in modo appropriato la tecnologia da applicare per le reti WMN. Il progetto di standardizzazione che nasce dal gruppo viene chiamato 802.11s. Il principio motore dell'estensione 802.11s deriva da esigenze di costi, flessibilità ed interoperabilità assenti nelle reti attuali. Il progetto viene svolto nell'ottica di trovare una soluzione di tipo wireless per sopperire alle gravi limitazioni delle Distribution Service (DS) che attualmente sono tutte basate su standard 802.3 (Ethernet). In modo da creare quello che viene definito Wireless Distribution Service (WDS). La sfida principale che deve affrontare l'estensione è la definizione di varie soluzioni volte a dare risposta alle seguenti domande:

1. In che modo i vari nodi si rilevano.
2. Come avviene il calcolo del miglior percorso dal nodo X al nodo Y.
3. Come deve avvenire interconnessione fra i nodi nella formazione della rete mesh.
4. La gestire della sicurezza nelle connessioni.

Grazie allo sforzo collettivo il gruppo rilasciò una versione dell'estensione che definisce i protocolli e le funzionalità per le reti mesh wireless nel 2011.

4.1 Definizioni

station (STA): Tutti di dispositivi conformi allo standard IEEE 802.11.
AP: Entità avente funzionalità di una station e che da accesso ad una DS ad altre STA collegate ad essa tramite wireless.
mesh facility o mesh point: L'insieme di funzioni avanzate, regole di accesso al mezzo, formato dei pacchetti, metodi di autenticazioni mutue utilizzati per fornire trasferimento di dati tra station operanti autonomamente.
mesh station: Una STA quality-of-service (QoS) che implementa la mesh facility.
mesh gate o mesh AP: Entità avente funzionalità di una station e che da accesso a uno o piu DS alle mesh basic service set (MBSS).
mesh BSS (MBSS): Una basic service set (BSS) contenente mesh stations (STAs) con la differenza che i mesh STA hanno una connessione in peer2peer e autonomi. Una MBSS può contenere zero o più mesh gates.
mesh portal (MPP): Il punto logico dove avviene l'integrazione di vari servizi.
precursor: Il mesh STA precedente nella scelta del percorso che conduce alla destinazione.

4.2 Struttura

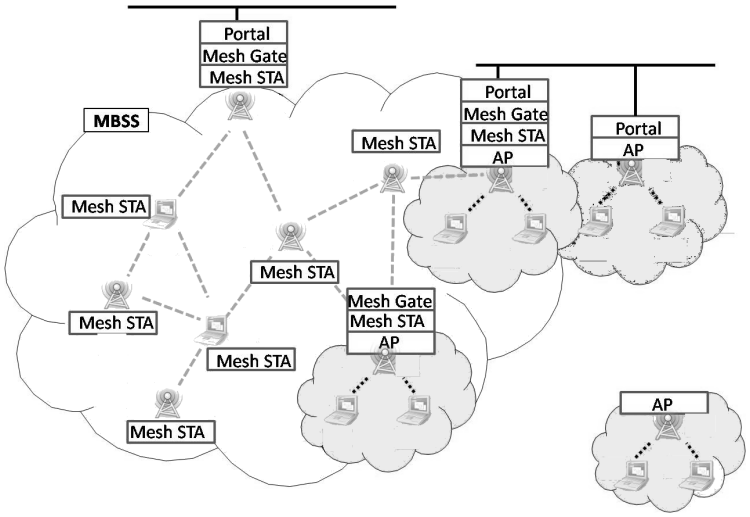


Figura 4.1: Rete wireless mesh con compatibilità ad altre reti

Le MBSS come da definizione sono delle BSS dove i nodi (mesh STA) sono connessi in modo paritario e possono scambiarsi dati mutualmente. In quest'ottica sembrano essere più delle Indipendant Basic Service Set (IBSS). Con la differenza che nelle MBSS la maggior parte dei componenti (mesh routers) è statica. Nell'estensione 802.11s i mesh STA non sono componenti ne dei tradizionali BSS ne dei IBSS ma solo una distinzione logica per indicare che le mesh STA non hanno le stessa funzionalità degli STA tradizionali ossia non sono conformi a 802.11. Essi potrebbero però integrare tali funzionalità per interfacciarsi sia alle celle BSS che alle IBSS ma a tale proposito vengono introdotte le mesh gate che si possono pensare come delle AP con funzionalità mesh facilty, che si occupano di interconnettere le MBSS alle celle tradizionali

tramite *Wireless Distribution System* (WDS). Per quanto riguarda invece la possibilità di collegare le MBSS ad altri servizi lo standard definisce il mesh portal, ancora una volta, questo componente può essere visto come un gateway. Con il mesh portal si possono connettere le varie MBSS alle DS tramite conformità 802.3.

4.3 Medium Access Control (MAC)

Il Medium Access Control (MAC) è un protocollo che appartiene alla sotto categoria del Data Link Layer. Il suo ruolo consiste nella gestione dell'accesso al mezzo trasmissivo tra i differenti nodi della la rete. I classici protocolli MAC non sono sufficiente quando implementate per le reti mesh. Questa differenza è dovuto alla topologia, alle caratteristiche di potenza e alla modalità di instradare i pacchetti nelle reti mesh in cui i collegamenti avvengono per lo più in multi-hop. La topologia delle mesh per definizione non possiede strutture centralizzate per questa ragione il MAC in uso è di tipo distribuito e cooperativo. Le funzionalità di shelf-healing, shelf-configuring, shelf-managed fanno si che il protocollo deve aver una buona conoscenza della rete per poter implementare al meglio gli algoritmi per la gestione degli accessi. Come si sa lo sviluppo di soluzione per le rete mesh da integrare nello standard 802.11s è fortemente legato a tecnologie già esistenti nello standard 802.11. Cosi L'algoritmo di livello MAC di riferimento attualmente usato dallo standard 802.11s è CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) con molti adattamenti poiché com'è adesso non va assolutamente bene.

4.3.1 CSMA/CA

Nel CSMA/CA un nodo che vuole trasmettere, deve prima ascoltare il canale per una certo tempo allo scopo di verificare le trasmissioni di altri nodi. Se dopo tale tempo il nodo ottiene conferma di canale libero esso può iniziare a trasmettere. Questo approccio viene detto "Listen Before Talk" (LBT) e il Collision Avoidance CA è il meccanismo usato per evitare trasmissioni multiple. Per ascoltare il canale il protocollo fa uso di due meccanismi, uno appartenente allo strato PHY "Physical Carrier Sense" (P-CS) e l'altro appartiene al MAC "Virtual Carrier Sense" (V-CS). Con P-CS viene rilevato la presenza di potenza "Energy Detection" (ED) nel canale, una potenza (energia) sopra una determinata soglia implica che il canale è occupato. Mentre con il V-PC è possibile ottenere informazioni sullo stato di trasmissione di pacchetti in corso oppure pianificate. Nel caso V-PC i nodi catturano i pacchetti e ricavano informazioni relative a pianificazione, tali informazioni sono solitamente custoditi nell'header in un campo chiamato "duration" , dal valore della di tale campo i nodi aggiornano il "Network Allocation Vector" (NAV) che indica sostanzialmente il tempo per cui sarà occupato il canale. Il NAV può essere aggiornato in ogni momento.
Il CA Viene usato per evitare trasmissioni multiple e quindi ridurre le collisioni tra pacchetti. Per fare questo vengono introdotti due timer:

- 1. Short Interframe Space (SIFS) il tempo sufficiente perché il ricevitore di un pacchetto muti in trasmettitore e che il trasmettitore muti in ricevitore.
- 2. aSlotTime tempo necessario ad eseguire ED e inoltrare il risultato al livello MAC dal livello PHY.

In base a questi tempi un nodo può accedere al canale solo se esso è in Idle per un tempo pari a:

$$SIFS + aSlotTime$$

Altrimenti deve aspettare un tempo:

$$SIFS + 2 * aSlotTime$$

detto Distributed Coordination Function Interframe Space (DIFS). Il nodo che riceve invia un ACK di conferma in caso di corretta ricezione del pacchetto dopo un tempo

$$SIFS + aSlotTime$$

Se ACK non viene ricevuto, il ricevitore tenta nuovamente l'invio di ACK dopo un tempo

$$DIFS + BACKOFFTIME(random() * aSlotTime)$$

4.3.2 Enhanced Distributed Channel Access (EDCA)

Il CSMA/CA viene utilizzato dal meccanismo EDCA che implementa QoS con piorità di accesso al livello MAC. EDCA supporta otto tipi di priorità di accesso classificati secondo quattro categorie differenti "Access Categories" (AC): Best effort, Background, Voice, Video. Queste a loro volta sono caratterizzate da diversi parametri tra cui: Arbitration Interframe Space (AIFS), Minimum Contention Window (CWmin), Maximum Contention Window (CWmax) e Transmission Opportunity limit (TXOP: Intervallo di tempo contention-free in cui una mesh STA può trasmettere pacchetti) che servono a differenziare l'accesso a canale. AIFS è l'intervallo di tempo che un dato pacchetto appartenente ad una AC deve aspettare dopo che il canale entra in stato idle per settare il backoff oppure trasmettere. Le classi possono settare il valore dei parametri determinando il diverso grado di priorità. Tipicamente gli AC video e voce hanno un grado di priorità più elevato delle altre. Ogni AC gestisce una coda con un flusso di pacchetti che viene organizzato in modo prioritario. Una volta scelto dalla coda il pacchetto da inviare si utilizza un meccanismo "Distributed Coordination Function" (DCF) per condividere il canale tra i diversi AC. Se nella trasmissione del pacchetto avvengono delle collisione si ritrasmette. 802.11s viene realizzato a partire da EDCA che è un meccanismo adattato alla trasmissione single-hop di conseguenza realizza performance limitate quando è utilizzato direttamente per le reti mesh. Per le risolvere il problema della natura asincrona nella gestione delle priorità ed garantire buone perfomance per comunicazioni multi-hop end-to-end ad EDCA vengono associati alcuni miglioramenti, tra cui un meccanismo di congestion control, un protocollo opzionale multi-channel e il protocollo MAC Mesh Deterministic Access.

4.3.3 Mesh Deterministic Access (MDA)

Mesh Deterministic Access è un meccanismo di accesso al mezzo opzionale elaborato per dare supporto a EDCA quando si verificano problemi di congestione. Esso per la sua natura deterministica implementa la sincronizzazione fra i nodi (mesh STA) che vogliono accedere al canale. In MDA un intervallo mesh Delivery Traffic Indication Message (DTIM) (contenente informazioni riguardanti il traffico sulla rete, la selezione del percorso, le associazioni ed altro) è definito tra due pacchetti di beacon consecutivi. Ogni DTIM e suddiviso in slot da 32 microsecondi dove posso essere riservate dalle MDA opportunity (MDAOP) per trasmettere i dati. Prima che una mesh station possa trasmettere deve mandare un MDA Setup request con informazioni sulla durata dell'MDAOP e l'offset all'interno del DTIM al ricevitore per chiedere una connessione, se il ricevitore può ricevere, risponde alla richiesta con un MDA Setup reply. Il meccanismo è completato e il mesh STA trasmittente può inviare periodicamente dati tramite MDAOP. Tramite caratteristiche del MDAOP i nodi mandano info in broadcast ad altri nodi per mettergli al corrente dello stato in cui si trovano. In base a queste informazioni contenuti nei MDAOP gli altri nodo determinano se possono trasmettere o restare inibiti. MDA come meccanismo end-to-end ha lo svantaggio di produrre un forte overhead nella rete, ci sono rischi di sovrapposizione tra gli MDAOP e frammentazione dei DTIM. Nel IEEE802.11s non tutti i mesh Point sopportano il meccanismo MDA che è solo integrativo.

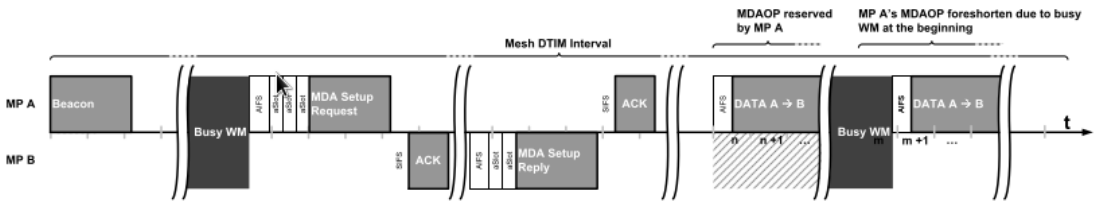


Figura 4.2: Accesso al mezzo tramite MDA: *trasmissione tra nodo A e nodo B ricevente.*

4.4 Problemi

Nella tecnologia wireless i principali problemi cause di molti effetti negativi sono da ricondursi al basso tasso di trasmissione di pacchetti ed elevati ritardi nella connessione end-to-end. Ci sono due fattori che causano questi problemi, ma il più distruttivo in termini di performance è certamente il problema del terminale nascosto. In base alla configurazione della topologia e alla posizione, i nodi si possono distinguere in due categorie, quella dei nodi nascosti e quella dei nodi non nascosti (visibili). Il secondo dei fattori invece riguarda la capacità di memorizzare dei nodi. Infatti una memoria piena tende a rifiutare nuovi pacchetti in arrivo, contribuendo in generale a varie complicazioni. Un esempio, in situazione di memoria piena i nodi non possono aggiornare le proprie tabelle di routing, e questo comporta calcoli errati nella determinazione di rotte. In questa sessione si spiegherà solo il meccanismo del terminale nascos-

to fornendo la descrizione di una delle soluzione maggiormente utilizzate per contrastare il fenomeno.

4.4.1 Il terminale nascosto

Il nodo o terminale nascosto (Hidden Terminal Problem) è un problema fondamentale in quasi tutti i sistemi formati con reti wireless. I terminali non possono sentirsi pure essendo relativamente vicini, questa situazione impedisce hai terminali nascosti di ricevere pacchetti di controllo e/o aggiornamenti. I pacchetti inviati ai nodi visibili dunque transitano indipendentemente, anche se altri nodi stanno già trasmettendo. Questo porta a collisioni all'interno del mezzo e perdita di dati. Ci sono più o meno tre situazioni in cui un terminale può essere considerato nascosto. Nella prima di queste, la peggiore in termini di throughput, tutti i nodi sono nascosti. Nella seconda i nodi sono tutti visibili e si contendono una risorsa, come AP, questo è il caso di miglior throughput. Infine nella terza sia i nodi visibili che nascosti appaiono insieme. Il problema del terminale nascosto nasce in ogni caso in modalità upstream, cioè nella fase in cui sono i clients a generare dati e trasmettere verso l'access point AP e questo è dovuto al range radio dei terminali. Viceversa il problema non si pone perché tutti nodi possono sentire l'AP. La figura fa capire meglio, il

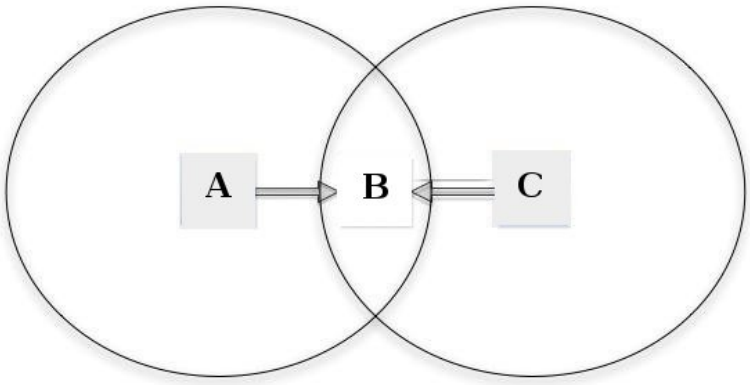


Figura 4.3: Hidden terminal.

classico scenario del terminale nascosto, consiste in tre nodi, A, B, C. in questo caso A può sentire B ma non C, B può sentire sia A che B e C può sentire B ma non A. se A e C dovessero trasmettere verso B allo stesso momento ci sarebbe collisione impedendo a B di ricevere. Un altro tipo di problema è il terminale esposto che descrive la conseguenza del fatto che quando due nodi vogliono trasmettere ad un medesimo destinatario, uno di essi si arresta bloccando la propria trasmissione concludendo erroneamente di poter interferire con l'altro nodo. Questo implica un spreco di risorse e una diminuzione del throughput.

4.4.2 RTS/CTS

La più comune delle soluzione a questi problemi proposta nelle reti wireless è l'utilizzo del meccanismo di handshacking tramite invio di pacchetti RTS/CTS per evitare le contese nell'accesso al mezzo. Un terminale che deve inviare un pacchetto, verifica innanzitutto che il canale sia libero e se questo è vero invia un pacchetto RTS (Request To Send) che viene captato dai terminali vicini a sé, i quali inibiscono le loro trasmissioni di nuovi pacchetti nel periodo specificato dall'RTS. Il destinatario del pacchetto invia alla sorgente un pacchetto CTS (Clear To Send) che inibisce i terminali vicini dal trasmettere. I pacchetti RTS e CTS sono molto piccoli e permettono di minimizzare le collisioni garantendo che la trasmissione vera e propria dei dati sia possibile senza collisioni nel mezzo condiviso.

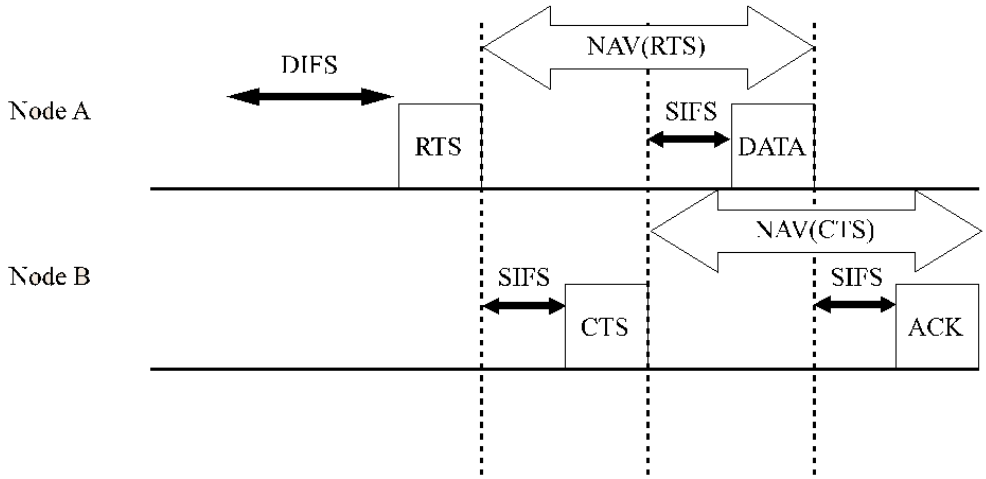


Figura 4.4: RTS/CTS

Capitolo 5

Algoritmi di Routing su Reti Mesh Wireless

Ad oggi la maggiore parte degli algoritmi di routing per le reti mesh sono derivati o estensione di algoritmi già esistenti per reti wrireless tradizionali. Le tecniche di routing che trovano ancora molta difficoltà nell'applicazione alle reti mesh. Difficoltà dovute alla stessa natura di tali reti.

5.1 Tipi di protocollo

I protocolli di routing su reti mesh e in generale su reti wireless Ad-Hoc vengono classificati in base al loro comportamento al momento della scelta del percorso (cammino, path) che lega i nodi della rete: in protocolli proattivi, protocolli reattivi e protocolli ibridi. Se i percorsi sono calcolati anticipatamente si sta facendo uso del *protocollo proattivo*, i cammini vengono stabiliti a priori con tutti i nodi della rete, indipendentemente se essi servono a formare o no un determinato percorso. Spesso chiamato table driven, in questo protocollo i nodi sono costantemente in cerca di percorsi, mantengono la consistenza e l'aggiornamento delle informazioni riguardanti i vari collegamenti nelle tabelle di routing. Il sapere ad ogni momento la configurazione e la topologia della rete permette a questo protocollo di essere molto veloce nei tempi di calcolo dei cammini, ma l'alto traffico di pacchetti di controllo fa si che ci sia bisogno di più spazio nei nodi, e il consumo di energia diventa un problema da considerare. Quando invece i percorsi sono calcolati al momento della richiesta di collegamento, si sta usando *il protocollo reattivo*, presenta il principale svantaggio rispetto al protocollo proattivo di essere molto lento nel decidere il percorsi, in quanto ogni nodo nella creazione del collegamento richiede in modalità On-demand la connessione ai nodi successivi. Presenta però il vantaggio per cui viaggiano meno pacchetti di controllo (overhead), c'è meno rischio di congestione e i nodi possono gestire in maniera efficiente i consumi di energia. Questo protocollo è spesso impiegato nelle reti con banda limitata dove l'overhead peggiora di solito le performance. I meccanismi su cui si basa sono il Route discovery per imparare i nuovi cammini, il Route maintenance per mantenere i cammini già calcolati e il Route delection per eliminare i cammini non più validi. Un esempio di rete che utilizza questo protocollo è la rete cellulare. *Il protocollo ibrido*, è un misto tra i primi due, dove

viene sfruttato il protocollo reattivo per piccole reti (Ad-Hoc) e il protocollo proattivo per reti più ampie (strutturata/backbone).

5.1.1 Link State Routing

Il Link State Routing è un protocollo proattivo in cui tutti i nodi originano informazioni riguardanti se stessi e lo stato del link con i nodi a se adiacenti. Queste informazioni vengono salvati in pacchetti chiamati Link State Packet (LSP). Tramite una tecnica di trasmissione chiamato flooding o link state broadcast gli LPS vengono mandati a tutti nodi vicini che a loro volta aggiornano la propria tabella di routing e rimandano gli LPS ai propri nodi vicini tranne quello da cui hanno ricevuto l'ultimo LPS. Ogni LPS ha un campo, il sequence number da cui i nodi capiscono se l'LPS è di aggiornamento, un duplicato o semplicemente un LPS arrivato fuori sequenza. Il sequence number permette di evitare confusione tra i vari LPS che transitano nel canale. Il meccanismo descritto permette a tutti i nodi di avere in memoria le informazione su tutti i nodi e i link della rete. A partire da queste informazioni si crea una mappa o grafo della rete, su cui puoi si applica degli algoritmi di selezione di cammini ottimali. L'algoritmo shortest path di E. W. Dijkstra è solitamente quello più utilizzato. Link State permette una forte convergenza nel costruire i cammini ed è adatto a reti complesse con un numero di nodi mediamente alto. Necessità però che i nodi (routers) abbiano abbastanza memoria per contenere le tabelle di routing e rispetto al Distance Vector è di più complessa da realizzazione.

Optimized Link State Routing (OLSR)

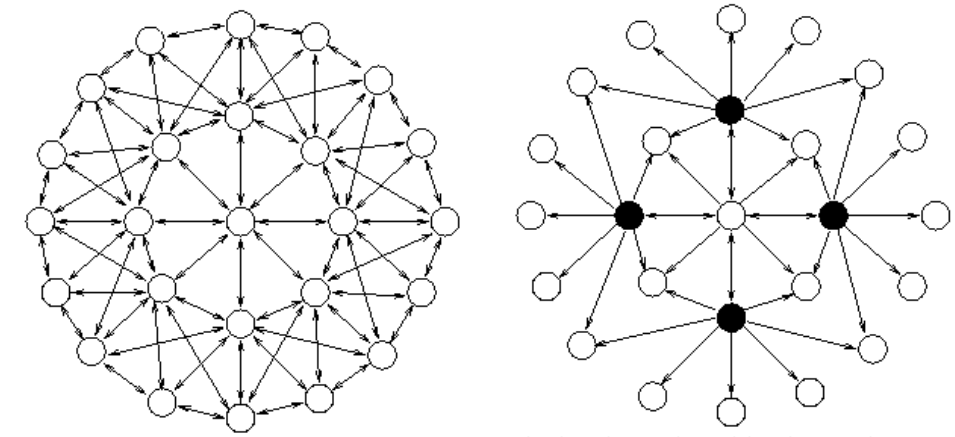
OLSR è un protocollo proattivo basato su tecnica link state routing, creato con lo scopo di aumentare le performance attraverso un meccanismo che riduce l'invio dei LPS nella rete. Si è capito infatti che dovendo ogni nodi della rete conoscere la topologia, l'invio indiscriminato dei LPS da parte di tutti i nodi portava ad una forte ridondanza che aumentava in numero dei duplicati, riducendo ovviamente le performance in termini di throughtput. Concettualmente il protocollo funziona implementando tre meccanismi diversi: Neighbor Sensing, Optimized Flooding (MPR) e Topology information.

- *Neighbor Sensing*: Questo meccanismo consiste nell'invio periodico da parte di un nodo di piccoli messaggi detti "hello", che contengono informazioni su l'indirizzo del nodo e la lista dei nodi vicini a sé, incluso lo stato del link. Ogni nodo ottiene cosi informazioni sul proprio vicinato (neighborhood) a 1 o 2 hop di distanza. Da tali informazioni sono estratti caratteristiche dei percorsi. In particolare si differenzia un percorso simmetrico o bidirezionale, quando un nodo A vede nel messaggio hello di B il proprio indirizzo e viceversa, da un percorso asimmetrico o mono-direzionale in cui solo B vede il proprio indirizzo nel messaggio di A.

- *Optimized Flooding (MPR)*: Nel risolvere il problema della trasmissione o ricezione di messaggi di controllo duplici si introduce il concetto di

Multipoint Relays (MPRs). In cui un nodo manda un pacchetto di controllo a tutti i suoi vicini, di questi solo un sottoinsieme detto MPRs con percorsi simmetrici ha il compito di rimandare ad altri nodi il pacchetto di controllo. In questo modo nell'intera rete solo alcuni nodi MPRs hanno la facoltà di ritrasmette gli LPS. Per definizione quando un nodo agisce come MPR l'insieme dei nodi che inviano LPS attraverso di esso è detto MPR Selector. Si riduce di molto la ridondanza e trasmissione di duplicati, cosa essenziale per le performance.

- *Topology information*: Ottenuti i primi due meccanismi, si raccoglie un insiemi sufficiente di informazioni riguardante la topologia della rete. Tutti i nodi MPR generano periodicamente una Topology Control message (TC-message) contenente il loro indirizzo e gli indirizzi del MPR selector. Cosi facendo comunicano alla rete da quale insieme MPR selector possono essere raggiunti. Il risultato finale è che ogni nodo riceve un grafo parziale della topologia della rete, fatto da tutti nodi raggiungibili e dall'insieme dei links tra nodo e MPRs della rete. Usando il grafo parziale si può calcolare tramite l'algoritmo shortest path un percorso ottimo da un nodo a qualunque destinazione raggiungibile. Le informazioni topologiche custoditi nei nodi hanno durata limitata, per rimanere validi devono essere periodicamente aggiornate. Quando non aggiornate vengono scartate.



Flooding dei pacchetti in una link state normale.
Le frecce rappresentano tutte le connessioni.
Grafo totale memorizzato nei nodi.

Flooding dei pacchetti dal nodo centrale tramite
nodi MPRs (in nero).
Grafo parziale memorizzato nei nodi.

Figura 5.1: Illustrazione ottimizzazione OLSR

5.1.2 Distance Vector Routing

Nell'Algoritmo di routing Distance Vector (noto anche come algoritmo di Bellman-Ford) i nodi (routers) conoscono il proprio indirizzo e la metrica che le lega ai propri vicini. A intervalli regolari inviano ai nodi adiacenti un *distance-vector*, che è un pacchetto contenente una coppia (indirizzo, distanza). La distanza indica il numero di hop o tiene conto di carico, peso, velocità e affidabilità dei collegamenti (metrica). A partire da tali dati, utilizzando l'algoritmo di Bellman-Ford, i nodi calcolano una tabella di routing che

associa ad ogni destinazione conosciuta, una stima della distanza e il primo hop del percorso calcolato. Quando avvengono cambiamenti i nodi inviano un *distance-vector* aggiornato che altri nodi usano per ricalcolare le tabelle di routing. Solitamente il ricalcolo avviene per i seguenti motivi:

- cade una linea attiva
- si riceve un *distance-vector* da un nodo non conosciuto
- si riceve da un nodo vicino un *distance-vector* che indica una metrica più conveniente rispetto a quello memorizzato

Il routing con distance vector ha il pregio di essere semplice da realizzare, poco costoso in quanto i nodi non richiedono particolare consumo di energia e la necessità di memoria risulta bassa. Il problema principale riguarda il formarsi cicli e la possibilità di avere una situazione di count-to-infinity quando si interrompono i collegamenti. Presenta una convergenza lenta o lunghissima nel caso si verificassero cicli o count-to-infinity.

Ad hoc On Demand distance Vector Routing (AODV)

Ad-hoc On-demand Distance Vector è un protocollo reattivo di tipo Distance Vector progettato per reti Ad-hoc mobile. Fornisce percorsi di routing "on demand" ai nodi che lo richiedono, abbattendogli quando non necessari. Ha un basso overhead sia in messaggi di rete che in elaborazione computazionale. Il protocollo permette ai nodi mobili di rispondere in modo tempestivo alle variazioni di topologia e quando una link cade (un nodo entra o lascia la rete). AODV si basa principalmente sul meccanismo d'invio di tre tipi di pacchetti quali:

- *Route request (RREQ)*, Quando è necessario una nuova rotta, il nodo (sorgente) trasmette un pacchetto RREQ in broadcast con l'indirizzo della destinazione e il campo per la metrica uguale 0. I nodi che ricevono RREQ creano un percorso verso la sorgente o aggiornano il percorso se nel RREQ è presente un Sequenze Number (SN) più grande o se SN è uguale ma la metrica è migliore del link in memoria. In tali casi i nodi rinviano l'RREQ aggiornato, in broadcast, comportando spesso la ricezione multipla di uno stesso RREQ. un percorso viene determinato quando RREQ raggiunge o la destinazione, o un nodo intermedio con un percorso aggiornato verso la destinazione.

- *Route reply (RREP)*, Con questo pacchetto il destinatario o un nodo intermedio comunicano alla sorgente l'accordo per la formazione della rotta. I nodi intermedi possono replicare con RREP verso la sorgente solo nel caso in cui l'RREQ sorgente ha settato il flag "D = 0", altrimenti solo il destinatario lo può fare. Attingendo alla propria cache i nodi intermedi creano un link unicast verso la sorgente permettendo a quest'ultimo di ricevere la RREP del destinatario.

- *Route error (RERR)*, Tutti i nodi monitorano il proprio collegamento col nodo vicino facente parte dei link attivi verso le destinazioni salvate.

Quando un link tra di essi cade, i nodi coinvolti inviano immediatamente verso gli altri nodi un pacchetto RERR comunicando il cambiamento. RERR indica la perdita delle rotte tra quelle possibili verso una destinazione.

Il vantaggio di AODV è che non genera traffico di controllo quando ci sono percorsi già stabiliti. Il livello computazionale risulta leggero e richiede poca memoria. Permette di mantenere performance elevate in reti con poca banda. Presenta però tutti gli svantaggi dei protocolli reattivi.

5.2 Metriche

La metrica di un link è il costo in termine di risorse che un canale necessità per trasmettere dei dati da un punto all'altro. É uno dei parametri fondamentali nell'applicazione di algoritmi di routing sia a livello di rete (layer 3) che al livello Data Link (layer 2). La metrica racchiude in sé le informazione sulla qualità del link, informazione che i protocolli usano per stabilire se un collegamento è ottimale o no. Definire un costo dei link di tipo wireless, quindi di reti mesh wireless è un compito particolarmente arduo. Sopratutto se si confronta con quelli delle rete con cavi. Non esiste una definizione assoluta del link nelle reti wireless in generale. Le proprietà di un link tra due nodi qualunque possono variare rapidamente in ogni momento, rendendo imprevedibile l'affidabilità della stessa. L' implementazione di una tecnica per sentire la simmetria e la gestione della mobilità dei nodi nelle mesh è indispensabile. La qualità radio di un collegamento dipende ovviamente da vari fattori come rumore, ostacoli vari, fading e comunque da interferenza (collisioni) quando il canale è sede di trasmissioni simultanee. Calcolare una metrica per un link deve quindi tenere conto di questi diversi fattori per poter implementare soluzioni che garantiscono una certa performance nei vari algoritmi. La metrica è un misura locale derivata direttamente dalle proprietà e dal comportamento del mezzo (quindi al livello PHY). In base queste proprietà e a parametri di trasmissione: potenza, modulazione, codifica e tempi vengono derivate diversi metriche. In seguito alcuni sono descritti.

5.2.1 Hop count

la metrica hop count è stata concepita per essere semplice, si basa sul calcolare il minimo numero di hop pensati come distanze da una sorgente ad una destinazione, assumendo l'infallibilità o no del link. Hop count è indipendente dalle caratteristiche di qualità e di trasmissione del link la sua implementazione è immediata ed è molto usato nelle reti cablate per la natura non mobile dei componenti ivi presenti e la forte affidabilità. Implicitamente o esplicitamente viene spesso usato anche nelle reti wireless. Nel caso delle reti mesh hop count può potare ad avere performance povere se si pensa che aumentare il numero di nodi è proprio una delle sfide di una rete mesh wireless.

5.2.2 ETX

L'*Expected Transmission Count* (ETX) è una metrica sviluppata specificatamente per la rete MANET, sulla base dell'osservazione per cui l' hop count non era ottimale applicato alle reti wireless. Basato sulla percezione di simmetria del link, ETX stima il numero di trasmissioni (inclusi re-trasmissioni) necessari affinché si possa inviare un pacchetto con successo. ETX si calcola tramite un semplice calcolo probabilistico, consideriamo due nodi A e B collegati da un link S, sia ti il tasso medio di trasmissioni da A verso B e tr il tasso medio di ACK inviati da B ad A per conferma di corretta ricezione. La probabilità che un pacchetto arrivi correttamente con conferma è uguale a $ti * tr$. Assumendo le trasmissioni variabili *iid* di Bernoulli e Il numero di trasmissioni affinché il pacchetto sia ricevuto correttamente una variabile geometrica. Il numero medio di trasmissioni necessario ad una corretta ricezione è:

$$ETX = \frac{1}{ti * tr} \tag{5.1}$$

Trovato ETX i cammini minimi su una rete mesh, vengono calcolati dall'algoritmo di routing utilizzando tecniche Shortest Path con metrica sui link dato dall'ETX. Il cammino minimo tra due nodi della rete risulterà quello con la somma ETX minore. A parità di ETX sarà implementato Hop count.

5.2.3 Airtime link metric (ALM)

Lo standard IEEE802.11s raccomanda come metrica di default Airtime Link. Rappresenta la misura di occupazione delle risorse di un canale quando vengono trasmessi dei dati. Il costo Airtime per ogni link si calcola tramite la formula:

$$Ca = [Oca + Op + \frac{Bt}{r}] * \frac{1}{1 - ef} \tag{5.2}$$

Dove Oca è overhead di accesso al mezzo, Op overhead del protocollo MAC, Bt è una constante e i parametri di ingresso r misurati in Mb/s e ef il tasso di errore.

5.3 HWMP

La *Hybrid Wireless Mesh Protocol* (HWMP) è un protocollo di routing ad implementazione obbligatoria secondo IEEE 802.11s. Esso combina la flessibilità dell'on-demand routing e proactive tree-based routing. La combinazione di elementi reattivi e proattivi di HWMP consente di selezionare il percorso ottimale ed efficiente in una vasta gamma di reti mesh (con o senza infrastrutture). HWMP utilizza un insieme di protocolli e regole di trasformazione tratti dal protocollo di routing AODV adattato al livello Data Link Layer (MAC) e ALM come metrica (*par.* 5.2.3). AODV costituisce la base per la ricerca on-demand di rotte all'interno di una rete mesh, mentre protocolli supplementari vengono utilizzati per impostare in modo proattivo collegamenti ad albero, fino al nodo MP radice (Gateway/Portal). La funzione che consente di costruire una topologia ad albero è opzionale e configurabile nei MPs. HWMP supporta due modalità di funzionamento a seconda della configurazione. Queste modalità sono:

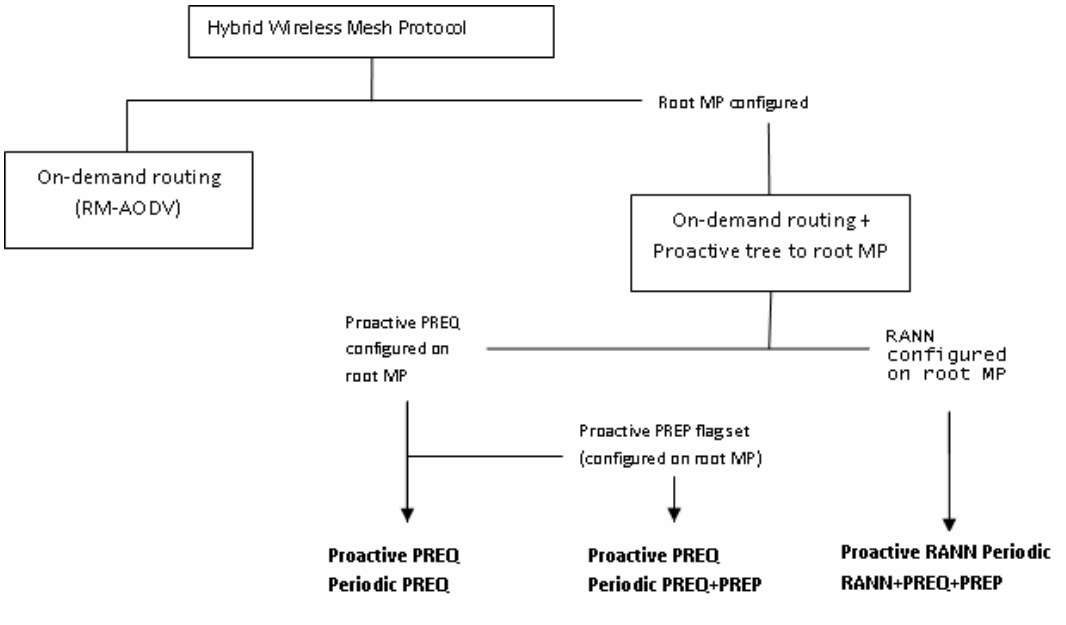


Figura 5.2: Schema delle modalità per implementazione di HWMP

- modalità on demand path selection: questa modalità permette ai MPs di comunicare utilizzando percorsi peer-to-peer. La modalità viene utilizzata in situazioni in cui non c'è presenza di una radice configurata (Ad-hoc). È utilizzato anche in determinate circostanze, ove invece esiste una radice configurata (strutturale).

- modalità proactive tree-based: usato quando si vuole raggiungere un nodo radice. Può essere eseguita utilizzando il modo proactive PREQ o modo proactive RANN. Queste modalità non sono esclusivi: su richiesta la modalità On-Demand e la modalità proattiva possono essere utilizzati contemporaneamente.

In HWMP tutte le modalità di funzionamento utilizzano regole di elaborazione comuni e i pacchetti di controllo sono simili al caso AODV ma considerando che il protocollo agisce al livello MAC si chiamano: *Path request* (PREQ), *Path reply* (PREP), *Path error* (PERR) e *Root Announcement* (RANN). La metrica di default, ALM determina quali percorsi HWMP costruisce. L 'uso del Sequenze Number permette di mantenere la connettività riducente la presenza di cicli.

5.3.1 Modalità on demand path selection

Uno dei vantaggi di On demand è che il metodo usato per la selezione di rotte tiene sempre conto del più aggiornato delle informazioni sulle rotte e permette di riduce di molto l' overhead introdotto dal protocollo. *Radio Metric AODV* (RM-AODV) è un estensione di AODV che usa come metrica per il routing Airtime Link Metric (ALM) definito in IEEE 802.11s draft 1.0.3 sotto la sezione "On demand path selection mode". In RM-AODV un MP sorgente che vuole trovare una rotta per una destinazione, invia in broadcast un pacchetto PREQ ai Mps specificati nella proprio lista destinazioni e setta il campo metric a 0 essendo esso l'iniziatore. Quando un MP riceve un PREQ, crea come nel caso AODV, una rotta verso la sorgente o aggiorna il rotta se nel PREQ è presente un Sequenze Number (SN) più grande o se il SN è uguale ma la metrica ALM è migliore di quella in memoria. Quando un pacchetto PREQ viene rinviato da un MP, il campo metrica viene aggiornato per includere le variazioni della rotta verso la destinazione. Dopo aver creato o aggiornato un percorso verso la destinazione, il MP destinatario invia in maniera unidirezionale un pacchetto PREP alla sorgente a conferma della ricezione del PREQ.

Quando il MP sorgente comincia a calcolare una rotta solitamente può indicare ai nodi intermedi che possiedono informazioni sulla destinazione di inviale pacchetti PREP. In questo caso il flag "Destination Only" (DO) viene posto a 0. quando invece tale flag è posto a 1 valore di default solo il destinatario può rispondere con PREP. In ogni caso quando il flag "Reply and Forward" (RF) è settato a 1 il nodo intermedio deve comunque ritrasmettere il paccheto PREQ con $DO = 1$ per evitare che gli altri nodi intermedi rispondano tutti con PREP. Se $RF = 0$ i nodi sono dispensati dal ritrasmettere PREQ.

Il meccanismo di DO e RF permette ai nodi di stabilire in tempi più brevi possibili le rotte, risparmiando in memoria e comunicando subito disponibilità di una rotta ottimale alla sorgente. Quando la sorgente riceve il PREP dal destinatario crea un percorso verso di esso. Appena il destinatario riceve un altro PREQ migliore in metrica di quello attuale aggiorna la sorgente di questa rotta mandando un PREP aggiornato con le informazioni sulla nuova rotta. In pratica si forma una rotta bidirezionale End-to-End tra sorgente e destinazione con la metrica migliore. Finché la sorgente invia dati la rotta è considerata attiva e viene mantenuta. Altrimenti le informazioni contenute nei nodi intermedi sulla rotta vengono cancellati. Se per errore, per caduta di linea o per qualche altro motivo il collegamento si interrompe su una rotta attiva i nodi che sentono il cambiamento devono segnalare alla sorgente che la destinazione è diventata irraggiungibile inviando un pacchetto PERR. Ricevuto questo pacchetto la sorgente rin-inizia il processo di calcolo per una nuova rotta.

Pacchetto PREQ è utilizzato per il path discovery di una o più destinazioni, per la costruzione in modo proattivo di un cammino ad albero verso un MP radice infine per confermare il percorso di una destinazione (facoltativo).

Octets:1	1	1	1	1	4	6	4	0 or 6	4
Element ID	Length	Flags	Hop-count	Time to Live	PREQ ID	Originator Address	Originator Sequence Number	Proxied Address	Lifetime

4	1	1	6	4	...	1	6	4
Metric	Destination Count	Per Destination Flags #1	Destination Address #1	Destination Seq. Num. #1	...	Per Destination Flags #N	Destination Address #N	Destination Seq. Num. #N

Figura 5.3: Pacchetto PREQ

- *Element ID*, identificativo del pacchetto.
- *Length*, lunghezza del pacchetto.
- *I Flags*:
  - Bit 0: Portal Role (0 = non-portal, 1 = portal),
  - Bit 1: (0 = group addressed, 1 = individually addressed),
  - Bit 2: Proactive PREP (0 = off, 1 = on),
  - Bit 3 – 5: Reserved,
  - Bit 6: Address Extension (AE) (1= (destination count ==1 proxied device address present), 0 = otherwise),
  - Bit 7: Reserved.
- *Hop Count*, è un numero intero senza segno ed indica il numero di hop a partire dalla sorgente fino al nodo intermedio MP che ritrasmettere il pacchetto. è incrementa di un ad ogni trasmissione.
- *Time to Live*, è un numero intero senza segno ed indica il tempo di permanenza in rete di questo pacchetto.
- *PREQ ID*, è un ID univoco per questo PREQ.
- *Originator Address*, è rappresentato come un indirizzo MAC a 48 bit ed è destinato all'indirizzo MAC della sorgente.
- *Originator Sequence Number*, è codificato come un intero senza segno ed è destinato a un numero di sequenza specifico della sorgente.
- *Lifetime*, indica il tempo per cui un MP ricevente il PREQ lo considera valido.
- *Metric*, tiene conto della metrica totale accumulato dalla sorgente al nodo intermediario.
- *Destination Count N*, indica il numero delle destinazioni (N) contenute nel presente PREQ.

- *Destination Flags*:
  - Bit 0: DO (Destination Only): se DO = 0, un MP intermedio può risponde alla PREQ con un PREP unicast, se DO = 1, solo il destinatario può rispondere con un PREP unicast. Il valore predefinito è 1.
  - Bit 1: RF (Reply-and-Forward): il flag RF controlla la ritrasmissione di PREQ ai nodi intermedi. Quando DO = 0 e il MP intermedio ha informazioni di inoltro alla destinazione, PREQ viene inoltrata solo se RF = 1. Il valore predefinito è 1. Quando DO = 1, il flag RF non ha nessun effetto.
  - Bit 2-7: Riservato.
- *Destination Address*, rappresentano indirizzo MAC a 48 bit delle destinazioni.
- *Destination Sequence Number*, è l'ultimo numero di sequenza ricevuto dalla sorgente per un qualsiasi percorso verso la destinazione.

Pacchetto PREP è utilizzato per confermare un percorso richiesto con PREQ.

Octets: 1	1	1	1	1	6	4	0 or 6	4
ID	Length	Flags	Hopcount	Time to Live	Destination Address	Destination Seq.Num.	Destination Proxied Address	Lifetime

4	6	4	1	6	4		6	4
Metric	Originator Address #1	Originator Seq. Num.	Dependent MP Count N	Dependent MP MAC Address #1	Dependent MP DSN #1	...	Dependent MP MAC Address #N	Dependent MP DSN #N

Figura 5.4: Pacchetto PREP

- *Element ID*, identificativo del pacchetto.
- *Length*, lunghezza del pacchetto.
- *Flags field*:
  - Bit 0-5: Reserved.
  - Bit 6: Address Extension (AE) (1=proxied address present, 0=otherwise).
  - Bit 7: Reserved.
- *Hop Count*, è un numero intero senza segno ed indica il numero di hop a partire dalla sorgente fino al nodo intermedio MP che ritrasmettere il pacchetto.

- *Time to Live*, è un numero intero senza segno ed indica il tempo di permanenza in rete di questo pacchetto.
- *Destination MP Address*, indirizzo MAC della destinazione per cui è calcolato il percorso.
- *Destination Sequence Number (DSN)*, indica la DSN del MP destinatario del PREQ.
- *Lifetime field*, se presente, riflette il Lifetime del pacchetto PREQ a cui sta rispondendo questo PREP.
- *Metric*, tiene conto della metrica totale accumulato dalla sorgente al nodo intermediario.
- *Originator Address*, è rappresentato come un indirizzo MAC a 48 bit ed è destinato all'indirizzo MAC della sorgente del PREQ.
- *Originator Sequence Number* il sequence number della sorgente del PREQ.
- *Dependent MP Count N*, indica il numero di MPs (N) dipendenti.
- *Dependent MP MAC Address* , indica l'indirizzo MAC del MP dipendente.
- *Dependent MP DSN* , rappresenta il Destination Sequence Number associato all'indirizzo MAC del MP dependente.

Pacchetto RERR è usato per segnalare un link non più disponibile a tutti i nodi che hanno in tabella informazione su tale link.

Octets: 1	1	1	1	6	4
ID	Length	Mode Flags	Num of Destinations	Destination Address	Destination MP Seq. Num

Figura 5.5: Pacchetto RERR

- *Element ID*, identificativo del pacchetto.
- *Mode Flags*, è riservato.
- *Number of Destinations*, indica il numero di nodi destinatari del pacchetto PERR (Destination address and Destination MP sequence number).
- *Destination Address*, indica l'indirizzo MAC del nodo non più raggiungibile.
- *Destination Sequence Number*, indica il sequence number del nodo non più raggiungibile.

5.3.2 Modalità proactive tree-based

Nella modilità proactive tree-based ci sono due meccanismi utilizzati per raggiungere il nodo radice (MPP/Gateway). Il primo modo usato è proactive Path Request (P-PREQ) che consente di creare rotte che collegano la radice a tutti i nodi della rete con protocollo proattivo. Il secondo modo usa la tecnica Root Annoucement (RANN) che consente invece di raggiungere il nodo radice con protocollo on demand.

5.3.2.1 Metodo PREQ proattivo

Il modo PREQ proattivo comincia con il MP radice che invia un pacchetto PREQ in broadcast, settando il $DO = 1$, $RF = 1$. PREQ contiene un campo (metrica) distanza settato inizialmente a 0 e una SN. Questo pacchetto viene mandato periodicamente incrementando SN. Tutti i MPs che ricevono il PREQ creano o aggiornano le proprie informazioni di invio verso la radice, aggiornano la metrica se necessario. Creano un nuovo PREQ aggiornato e lo rinviano agli altri nodi. Cosi le informazioni sulla presenza e la distanza da una radice vengono registrati da tutti i MPs della rete.

5.3.2.2 Metodo RANN proattivo

La radice trasmette periodicamente un pacchetto RANN nelle rete. L'informazione contenuta nel pacchetto viene usato per comunicare metriche sulla rotta verso di essa. Appena ricevuto il pacchetto, ogni nodo che deve creare o aggiornare una rotta manda un pacchetto PREQ in modo unidirezionale e con modalità on demand verso la radice attraverso il nodo precursor da cui a ricevuto il RANN. La radice allora invia un PREP in risposta ad ogni PREQ, cosi facendo il pacchetto PREQ viene usato per creare una rotta dalla radice al nodo richiedente mentre il pacchetto PREP per creare una rotta dal nodo alla radice.

Pacchetto RANN è utilizzato per la presenza di un nodo configurato come radice MP (root). Questo pacchetto è trasmesso periodicamente dal nodo radice.

Octets: 1	1	1	1	1	6	4	4
Element ID	Length	Flags	Hopcount	Time to Live	Originator Address	Destination Sequence Number	Metric

Figura 5.6: Pacchetto RANN

- *Element ID*, identificativo del pacchetto.
- *I Flags*: Bit 0: Portal Role (0 = non-portal, 1 = portal). Bit 1 – 7: Reserved

- *Hop Count*, indica il numero di hops dalla radice (root MP) al MP che trasmette la richiesta.
- *Time to Live*,indica il numero di volte rimanenti per la ritrasmissione del pacchetto RANN.
- *Originator Address*, l'indirizzo MAC della radice.
- *Destination Sequence Number*, il sequence number specifico della radice (root MP).
- *Metric*, indica la metrica totale dalla radice al nodo che da cui è stato trasmesso RANN.

5.4 B.A.T.M.A.N

B.A.T.M.A.N. è un protocollo di routing proattivo risultato da una rivisitazione del protocollo OLSR per le wireless Mobile Ad-hoc Networks (MANET) e wireless Mesh Networks. È stato realizzato dalla comunità di Freifunk con sede a Berlino sulla base di una propria esperienza. Attraverso diversi test del protocollo OLSR, si è concluso che vi era la necessità di un nuovo algoritmo di routing, cosi nasce BATMAN. La ragione principale di questa decisione è la dimensione di reti con oltre 400 nodi. Poiché il protocollo OLSR, sfrutta il protocollo di routing link state, diffonde le informazioni sulla topologia della rete a tutti i nodi e richiede che essi calcolino il percorso più breve conoscendo intera topologia, questo comporta un elevato costo computazionale nel mantenere ed aggiornare le tabelle di routing. La perdita di tempo per il calcolo del cammino minimo può portare a incoerenze nella topologia e quindi produrre cicli, che ovviamente sono indesiderati. Per questo motivo si è optato per un approccio che funzioni senza che ogni nodo debba conoscere le informazioni sulla topologia del intera rete ma mantenga la costruzione del link in modo proattivo.

Evoluzione del protocollo A differenza di molti altri protocolli di routing, B.A.T.M.A.N ha subito una ampia fase di sperimentazione fin dall'inizio del progetto, migliorando l'algoritmo per i problemi di vita reali. L'evoluzione dell'algoritmo si può suddividere in generazioni tutt'ora in corso di miglioramento:

- Generazione I servì solo come implementazione sperimentale per testare l'inizio del progetto. La qualità dei link è misurata in base alla quantità di "Originator Messages" (OGM) ricevuti. Non verificava le condizioni di link bidirezionale per l'inoltro del pacchetto.
- Generazione II introduce la classificazione dei link in bidirezionale e unidirezionale, implementando il meccanismo per cui i nodi non trasmettono informazioni su link non bidirezionali.
- Generazione III introduce alcune regole per la ritrasmissione dei OGMs, sono ritrasmessi solo se provenienti da link ritenti migliori il best-ranking

neighbor (nodo vicino con ranking migliore), limitando la probabilità di avere dei cicli. Il riconoscimento di un link bidirezionale ora dipende anche dal tempo. In più l'algoritmo sopporta interfacce multiple per i nodi, cosi come un valore TTL configurabile nei OGMs.

- Generazione IV cambia il concetto del meccanismo di neighbor ranking. Le generazioni precedenti basano le loro decisioni di routing su quanti OGM vengono ricevuti, la IV generazione introduce la metrica transmit quality (TQ), che misura la probabilità di successo nella trasmissione di un pacchetto su un determinato link.

- Generazione V è attualmente in via di sperimentazione.

- B.A.T.M.A.N Advance (Batman-adv) Il protocollo realizzato in forma di modulo nel kernel linux (dal 2.6) è applicato al livello 2 ISO/OSI. Utilizza l'indirizzo MAC per l'indirizzamento ai nodi vicini. Il risultato è che Batman-adv è in grado di emulare un bridge/Ethernet, in modo che tutti i nodi sembrano collegati da un link diretto (virtuale). Come conseguenza, tutti i livelli superiori dell'ISO/OSI vedono il livello Data Link come un solo link, ignari di collegamenti multi-hop.

5.4.1 Principio base

Batman è un algoritmo semplice e robusto creato per fornire migliori performance nelle creazione di percorsi multi-hop nelle reti Ad-hoc e mesh wireless. I nodi in batman non mantengono le informazioni riguardanti tutta la topologia della rete, bensì ogni nodo memorizza il next-hop migliore che porta alla propria destinazione. L' obbiettivo è quello di massimizzare la probabilità di consegnare un pacchetto, per fare questo il protocollo non calcola direttamente la qualità del link [metrica] ma piuttosto ne cerca l'esistenza. Per sentire se il link esiste ogni nodo periodicamente manda in broadcast pacchetti hello ai propri vicini (ad una hop), questi pacchetti vengono chiamati *Originator Messages* (OGM). Quando un nodo riceve un OGM ci sono due possibilità, che nella sua tabella di routing è presente la sorgente del pacchetto OGM oppure no. Se la sorgente non è presente si crea una entry in cui viene aggiunta, il nodo ricevente è indicato come un suo nodo vicino e sequenze number incrementato. Se la sorgente c'è e il nodo è nuovo lo si segna come nodo vicino e si incrementa il sequenza number altrimenti si incrementa solo il sequenze number, il nodo non è nuovo. Ogni nodo compara i suoi link in base al numero di OGM ricevuti entro lo sliding windows, questo metodo rappresenta una metrica usato da batman chiamato *transmission quality* (TQ)(*par.* 5.4.3.3) introdotto nella versione IV.

5.4.2 Algoritmo

Il principio fondamentale dell'algoritmo di B.A.T.M.A.N si riassume semplicemente: dopo un determinato intervallo di tempo ogni nodo trasmette un pacchetto originator message (OGM) a tutti i nodi vicini, per informare della propria esistenza. Al fine di comunicare agli altri nodi fuori portata radio dell'esistenza del nodo sorgente, questo pacchetto viene ritrasmesso nella rete dai nodi vicini in base a determinate regole. Tutti i nodi memorizzano le informazioni sul vicino dal quale ricevono l'OGM e ricavano il TQ di tutti i link diretti a cui sono connessi. In base a tale metrica ogni nodo decide quale dei suoi link vicini è il migliore per trasmettere dei dati verso una destinazione. Si illustra con un modello:

Consideriamo un grafo G = (N, E), dove N rappresenta l'insieme dei nodi e E l'insieme dei link (archi) tra i nodi. Per ogni nodo $i \in N$, esiste un sottoinsieme K di nodi vicini (single-hop). Il pacchetto 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. Altrimenti viene trasmesso lungo un tragitto multi-hop composto da un percorso *(s,i)* e una rotta [i, d] dove i è un nodo scelto dall'insieme K e *(s,i)* un link di E. la rotta [i, d] rappresenta il tragitto dal nodo i al nodo d attraverso la sottografo S.

$$S = (N - \{s\}, E - \{(s, i) : i \in K\}) \tag{5.3}$$

L'obbiettivo di BATMAN è quello di massimizzare la probabilità di consegna del pacchetto, considerando la metrica sui vari link e implementando questi 4 Step:

- Step 1. dato un pacchetto m diretto dalla sorgente s al destinatario d nel 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 single-hop di s

- Step 2. ad ogni arco non escluso nello Step 1 viene associato una metrica (peso) $\omega_{s,i}$ che rappresenta il numero di originator messages ricevuti dalla destinazione attraverso un nodo vicino i, presenti nello sliding window.

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

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

Esempio di Algoritmo B.A.T.M.A.N: Il nodo 1 vuole inviare un messaggio m al nodo 6. I link utilizzabili sono solo quelli con i nodi vicini (single-hop), che formano l'insieme K = {2, 3, 4} figura 5.7. Per primo, determina il link da cui ha ricevuto il maggior numero di OGM per il nodo 6.
Supponendo che il link *(1,2)* abbia il più alto $\omega_{s,i}$, esso sarà scelto per inoltrare m fino al nodo 2 figura 5.8.
Dato che il nodo 2 non è il destinatario, si riduce il grafico G nel più piccolo grafo S, escludendo i nodi che non fanno parte dell'insieme K del nodo 2, e si

ripete l'algoritmo dallo Step 1 figura 5.9.
Supponendo ora che il link con maggiore $\omega_{s,i}$ sia *(2,5)*, si sceglie come prossimo nodo per inoltrare m il nodo 5. Dal nodo 5 si giunge al nodo 6 il destinatario, termina l'algoritmo.

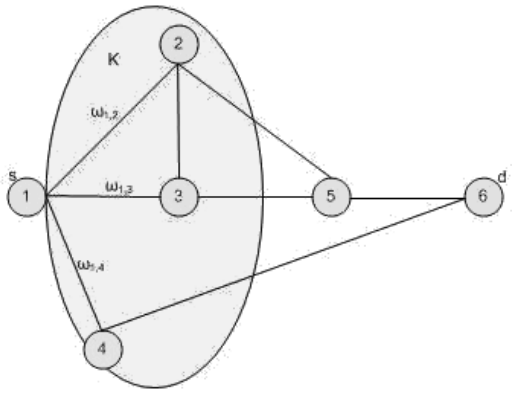


Figura 5.7: Grafo con insieme K e metrica $\omega_{s,i}$

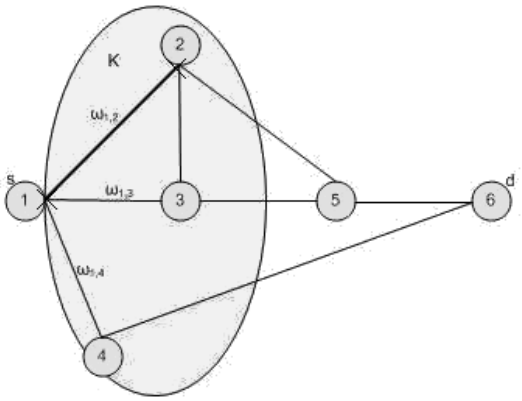


Figura 5.8: Selezione del migliore link e trasmissione del messagio m

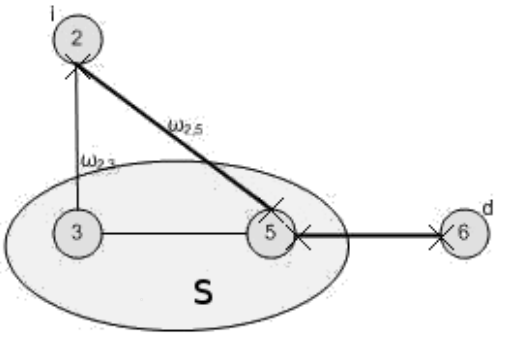


Figura 5.9: Sottografo S e continuazione dell'algoritmo

5.4.3 Protocollo *ver.IV*

5.4.3.1 Formatto pacchetti

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

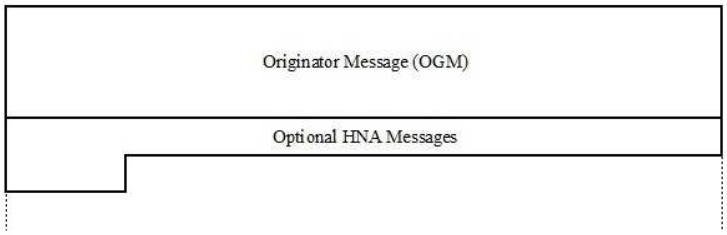


Figura 5.10: Pacchetto BATMAN generico con campo HNA per la compatibilità con altre reti

5.4.3.2 Originator Message Format

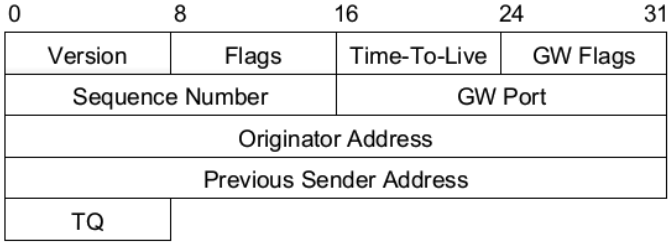


Figura 5.11: Pacchetto Originator message

Gli Originator Messages (*riferendosi alle ver.IV di B.A.T.M.A.N*) sono messaggi piccoli, tipicamente di 17 byte. Considerando headers IP e UDP sono circa 57 byte e Contengono almeno:

- *Version*: campo obbligatorio, se OGM viene ricevuto con versione differente sarà ignorato
- *Is-Direct-Link Flag*: indica un link diretto, single-hop
- *TTL (Time To Live)*: imposta un Limite di tempo superiore al massimo numero di hop con cui un OGM può essere ritrasmesso
- *GWFlags (Gateway Flag)*: gestisce la banda in kbit/sec nel caso il nodo abbia accesso ad internet e faccia da gateway nella rete
- *Sequence Number*: numero che identifica ogni OGM. Ogni nodo tiene memorizzato in una finestra gli ultimi Sequence Number ricevuti, per evitare duplicati ed aggiornare TQ

- *Originator Address*: indirizzo IPv4 o MAC dell'interfaccia di rete che ha originato l'OGM
- *Transmit Link Quality (TQ)*: la metrica che descrive la probabilità di successo nella trasmissione di un pacchetto su un determinato link
- *Previous Sender Address*: l'indirizzo del nodo precedente al nodo che sta ritrasmettendo il pacchetto

Sequenze Number B.A.T.M.A.N. È Sequence Number Oriented, L'insieme di SN possibili è determinata dalla lunghezza del campo SN. Avendo 16 bit si ottiene un SN massimo di $2^{16} - 1 = 65535$. I Sequenze Numbers sono suddivisi in due aree detti, in-window e out-of-range di uno sliding window. Il meccanismo di base memorizza i Sequenze Numbers più recenti ricevuti da ciascun Originator nell'in-window, dove vengono utilizzati per determinare la qualità dei link vicini, il Neighbor Ranking. Ogni volta che un Sequenze Number già memorizzato cade nell'area out-of-range viene scartato cosi da mantenere le informazioni più aggiornate.

5.4.3.3 Neighbor Ranking

Quando un nodo riceve un OGM riconosce l'esistenza del suo originator. Inoltre, una rotta verso tale originator viene creato attraverso i nodi intermedi. Nel caso in cui esistono diverse rotte verso l'originator, può succedere che gli OGM siano ricevuti da più di un vicino single-hop. Pertanto, il nodo ricevente per ogni originator conosciuto, mantiene una lista di tutti i vicini single-hop, da cui l'originator è raggiungibile. Al fine di prendere la decisione di routing, un nodo esegue una classifica dei suoi vicini single-hop in base ai valori del TQ globale ricevuto, dopo aver preso in considerazione il TQ locale e l'asymmetric penalty del ultimo hop prima di sé stesso. Il vicino con il più alto valore di TQ globale è considerato quello che fornisce il percorso con la più alta probabilità di trasmissione con successo e che privilegia i percorsi più brevi, frequenti e non asimmetrici. I valori del TQ globale sono memorizzati all'interno di una sliding window per ogni voce nella neighbor list di un nodo. Il valore desiderato per il confronto si ottiene facendo la media delle voci presenti nella sliding window . Gli OGMs con TQ pari a zero non sono presi in considerazione, in quanto pacchetti non ricevuti.

Transmit Link Quality (TQ) Nella versione IV di B.A.T.M.A.N la qualità del link (link quality) viene determinata da due valori:

1. transmit link quality (TQ)
2. receive link quality (RQ): la probabilità di una corretta ricezione di un pacchetto da un nodo vicino (figura).

Generalmente un nodo non può misurare direttamente la quantità di pacchetti trasmessi con successo, verso un altro nodo senza alcuna indicazione. Per questo motivo il protocollo esegue un semplice calcolo per cercare di approssimare il valore locale di TQ.

Figura 5.12: Illustrazione: Transmit Link Quality (TQ)

Figura 5.13: Illustrazione: receive link quality (RQ)

Per fare questo il protocollo tiene conto anche di un terzo valore definito come Echo Link Quality (EQ) che indica la probabilità di una corretta ricezione di un OGM proveniente un nodo A e ritrasmesso da un nodo B verso A figura 5.14. Utilizzando i valori di RQ e di EQ, un nodo può calcolare il valore locale

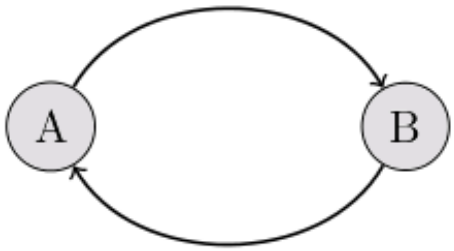


Figura 5.14: Illustrazione: Echo Link Quality (EQ)

di TQ verso un nodo vicino eseguendo l'equazione seguente:

$$EQ = RQ * TQ_{locale} \Longrightarrow TQ_{locale} = \frac{EQ}{RQ} \tag{5.4}$$

Nel OGM generato da un nodo, il campo TQ viene impostato con lunghezza massima (255) prima di essere trasmesso. Il vicino single-hop ricevente calcola il proprio TQ_{locale} e lo aggiunge al valore TQ ricevuto prima di ritrasmettere l'OGM con il nuovo valore TQ. Il nuovo valore TQ , rimpacchettato nell'OGM da ritrasmettere viene calcolato come:

$$TQ = TQ_{ricevuto} * TQ_{locale} \tag{5.5}$$

Un esempio di calcolo di TQ viene raffigurato nella figura sottostante.

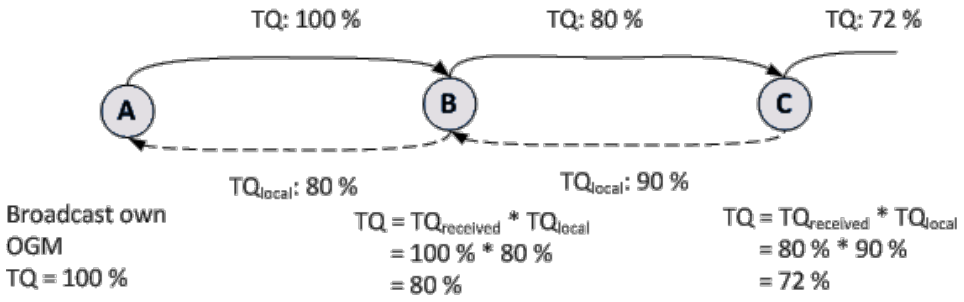


Figura 5.15: Esempio di come viene calcolo TQ nei vari nodi. Un TQ di *100%* indica 255 il valore massimo.

5.4.3.4 Bidirectional Link Check

BATMAN esegue un controllo bidirezionale del link tra nodi vicini, considerando il valore TQ dell'OGM. Questo è essenziale, in quanto deve essere garantito il funzionamento del link in entrambe le direzioni, nel caso in cui vengano usati ACK del livello MAC . La simmetria del link da e per l'originator viene testato calcolando il nuovo valore TQ per il percorso. Questo viene già fatto dall'equazione *5.7*, fatta eccezione per la hop penalty, che viene applicato appena prima che l'OGM venga ritrasmesso. Se il valore locale di TQ verso l'ultimo hop si dimostra troppo lento, il valore risultante globale TQ risulterà sotto una determinata soglia, che serve come limite per considerare il link bidirezionale. Di default questa soglia e impostato a 1. Una soglia più alta non cambia le decisioni di routing, ma riduce comunque il campo di propagazione di un OGM in una rete afflitto da link con perdite.

Asymmetric Penalty e Hop Penalty Per assicurare che il miglior link bidirezionale sia scelto dal protocollo, un parametro *Asymmetric Penalty* è stato introdotto per penalizzare quei link che hanno un scarso RQ. Questo parametro è un valore ponderato e si trova con la seguente formula:

$$f_{asym} = (1 - (1 - RQ)^3) \tag{5.6}$$

Questo parametro ha una grande influenza sul valore di TQ per i link in cui è forte la perdita di pacchetti. Un nodo in BATMAN conosce solo il miglior link verso i propri vicini next hop e non dell'intero percorso. Questo fa si che, il nodo non sappia di quanti hops è costituito il link verso una determinata destinazione multi-hop. Altri protocolli usano il meccanismo del shortest path per individuare il minimo numero di hops per una destinazione cosi da ridurre la latenza e risparmiare in banda. BATMAN utilizza un altro parametro chiamato, *Hop Penalty* (HP) che ad ogni hop attraversato da un OGM diminuisce il valore TQ di un valore fisso. Sia asymmetric penalty che la hop penalty vengono aggiunti al valore TQ finale dall'equazione:

$$TQ = TQ_{ricevuto} * TQ_{locale} * f_{asym} * HP \tag{5.7}$$

5.4.4 Prestazioni

Le prestazione che sono fornite in questa tesi prendono riferimento dal lavoro svolto da [19]. Per capire meglio le performance fornite da BATMAN è opportuno mostrare i pregi o difetti rispetto a un altro protocollo almeno di pari livello, nel testo [19] questo confronto è stato fatto con OLSR che rappresenta il diretto concorrente. Si tiene conto dei ben noti parametri per la valutzione delle perfomance di un algoritmo di routing: THROUGHPUT, OVERHEAD, CONSUMO CPU, PACKET LOSS e MEMORIA. La sperimentazione viene fatto su 49 nodi montati in griglia con la possibilità di modificare disposizioni e parametri [potenza, frequenza, velocità, posizione]. Gli strumenti di misura sono tools presenti nel sistema unix utilizzato.

5.4.4.1 Routing Overhead

L'abilità di un algoritmo di routing di essere scalabile dipende essenzialmente dalla sua capacità di gestire i pacchetti di controllo (overhead) al crescere dei nodi nella rete. In BATMAN questo viene fatto trasmettendo OGM's mentre in OLSR si utilizza pacchetti hello e Topology Control (TC). La quantità di pacchetti che transita in un nodo definisce il traffico in esso. In base a tale traffico e alla grandezza dei pacchetti si pongono a confronto i due protocolli per capire quale dei due offre migliore gestione del overhead. Il traffico in uscita è la chiave per il confronto in quanto dato più incisivo sulle performance. Ovviamente i due protocolli gestiscono questa fase in modo diverso come descritto nelle sessioni precedenti. Il risultato di questa analizzi viene presentata in figura. Dove viene valutato il traffico di pacchetti in uscita in

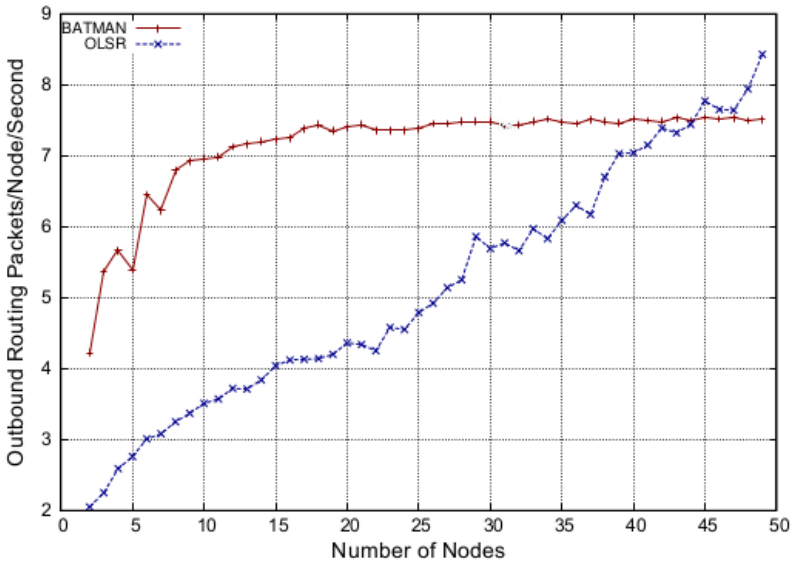


Figura 5.16: Overhead in uscita dai nodi rispetto al *num* di nodi della rete

base al numero di nodi, la superiorità di OLSR per numero di nodi inferiori a 45 è netta. Oltre, BATMAN prende la rileva mostrando una significativa

costanza. L'overhead è fortemente dipende dalla dimensione dei pacchetti. In figura 5.17 viene mostrato l'overhead totale espresso dalla media pesata ottenuta moltiplicando la dimensione di ogni pacchetto uscente dal nodo al secondo per il numero di nodi delle rete. Qui BATMAN prevale almeno di 10 volte su OLSR, in principio la natura proattiva di OLSR fa si che ogni nodo debba memorizzare informazioni su tutta la topologia della rete, provocando un forte overhead spesso dovuto alla trasmissione di pacchetti inutili alla decisione dei percorsi. Mentre in BATMAN viene memorizzato solo informazioni sul migliore link verso il proprio vicino, inoltre molti pacchetti vengono abbandonati (non ritrasmessi) per rispettare questa scelta.

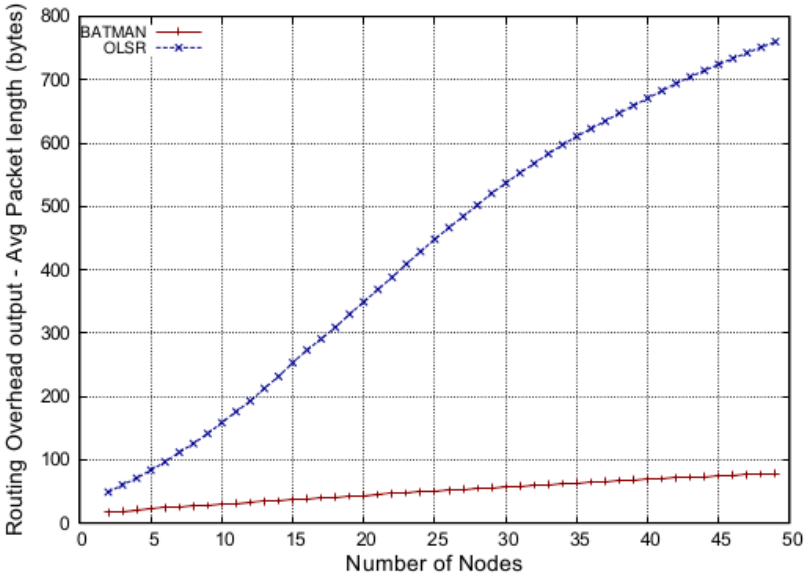


Figura 5.17: Lunghezza media dei pacchetti al crescere della rete

5.4.4.2 Throughput & Packet loss

Il Throughput è la capacità di trasmissione effettivamente utilizzata ed indica la porzione di banda utilizzata rispetto alla banda totale offerta da un canale trasmissivo. Come tale il Throughput fornisce indicazioni indispensabile nella valutazione delle performance di un algoritmo. Dal grafico 5.18 si vede che il throughput decresce quasi linearmente in entrambi gli algoritmi al crescere dei nodi. BATMAN però fornisce performance leggermente superiori circa del 15% rispetto al rivale.

Come detto in parte alla fine del paragrafo (5.4.4.1) ed illustrato nel grafico 5.19, sotto la voce, *packet loss* BATMAN ha una forte attitudine ad abbandonare pacchetti rispetto ad OLSR, questo per migliorare il throughput che dipende direttamente dalle decisioni per la scelta ottimale dei percorsi. Ci mette più tempo ad adattarsi ai cambiamenti di topologia e in genere trova meno percorsi verso una destinazione. Questi però non rappresentano un

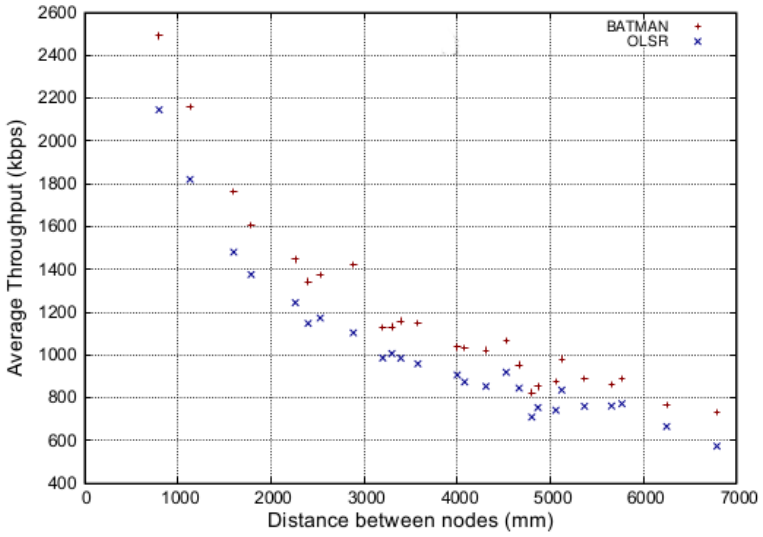


Figura 5.18: Throughput medio in base alla distanza tra i nodi della rete

particolare svantaggio in quanto statisticamente non molto rilevanti per le performance più importanti.

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 5.19: Tabella riassuntiva di vari parametri

5.4.4.3 Carico CPU & Memoria

È importante che un algoritmo tenga conto del carico su processore e del consumo di memoria che ogni nodo della rete deve sopportare affine di evitare drastiche riduzione di performance ed eccessivo consumo di energia. Il carico della CPU dipende da traffico in transito in ogni nodo e dalla complessità di calcolo dell'algoritmo nel determinare i percorsi migliori. Nell'ambito delle rete wireless mesh è molto importanti la questione delle risorse. Questo perché i mesh sono stati ideati con la caratteristica peculiare di ridurre i costi d'installazione ed è possibile farlo se i componente nella rete sono disponibili a costi accessibili. Si dimostrata da questo esperimento che BATMAN pure producendo un maggiore overhead per reti con meno di 45 nodi esibisce un carico sulla CPU di gran lunga inferiore di circa 40% rispetto a OLSR già a partire 6 nodi, figura 5.20. BATMAN si conferma come scelta per le reti di grandi dimensione, e comunque per reti che puntano molto sulla scalabilità. A conferma di tale scelta guardando la figura 5.21 BATMAN riduce di molto il consumo di memoria nei nodi rispetto al rivale già quando si superano 30 nodi. OLSR può

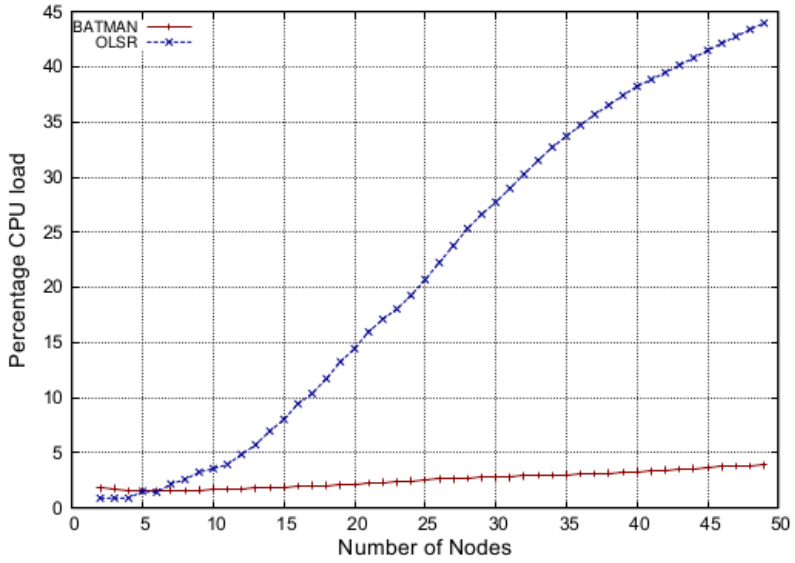


Figura 5.20: Percentuale di carico CPU rispetto al *num* di nodi della rete

tranquillamente essere utilizzato per reti domestiche, mentre per reti cittadine e aree metropolitane BATMAN si considera la soluzione migliorare.

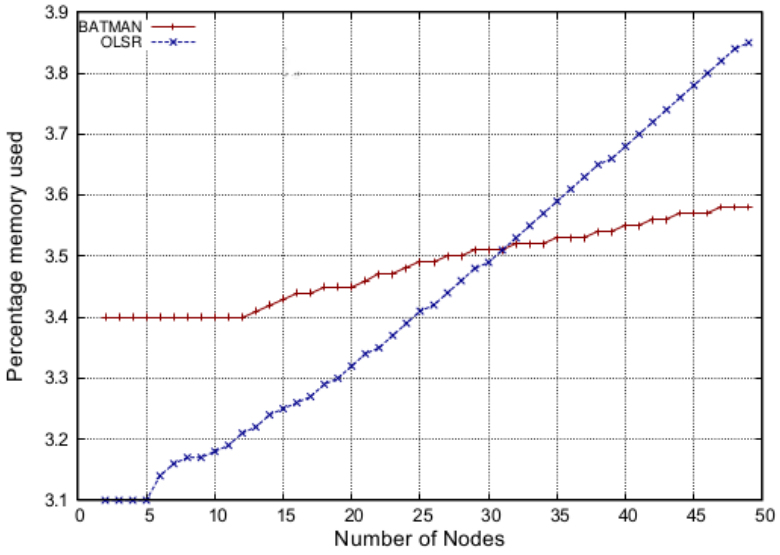


Figura 5.21: Percentuale di memoria in consumo rispetto al *num* di nodi della rete

5.5 Sicurezza

La sicurezza è fondamentale nelle reti basate su tecnologia wireless. Gli attacchi a reti wireless mesh, sono numerosi e variegati, ma molti di loro sono essenzialmente Denial-of-Service (DoS), attacchi che non distruggono i dati in sé, ma consumano e compromettono le risorse a disposizione di nodi nella rete. Tuttavia, in questo ambito si tratterà di attacchi che vengono perpetuati

al livello MAC dandone un breve elenco per poi specificare alcuni metodi di difesa.

5.5.1 Attacchi

- Wormhole attack: è un attacco in cui due nodi maliziosi creano un link che utilizzano per ingannare la rete. In figura 5.22 tra s e d sono presenti i nodi di attacco $M1$ e $M2$. I due nodi maligni fanno capire alla rete di avere il percorso migliore. Tutte le informazioni vengono fatte transitare da tale link. L'aggressore a completo controllo su tale link e di conseguenza sui dati che vi transitano.

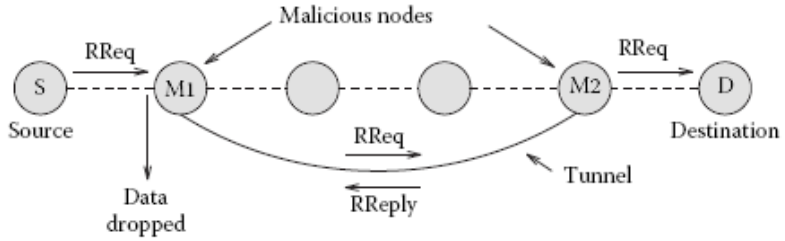


Figura 5.22: I nodi $M1$ e $M2$ effettuano un attacco wormhole

- Black Hole attack: se la maggiore parte dei nodi single hop attuano link wormhole, l'informazione da trasmettere potrebbe andare persa.

- Routing tables Overflow: l'aggressore segnala percorsi verso nodi inesistenti. Lo scopo è riempire le routing tables con informazioni errate, compromettendo il protocollo.

- Eavesdropping: nelle WMNs, tipicamente, prende di mira il livello fisico e MAC quando vengono immessi dati non protetti. Un aggressore può ricavare informazioni riguardanti la topologia della rete, le relazioni tra i diversi nodi e il tipo di routing che si sta adottando. Questo attacco è difficile da eliminare poiché un ascoltatore passivo non inserisce dati nella rete. Ci si può proteggere da questo attacco implementando un buon algoritmo di crittografia dei dati.

- Flooding attack: l'aggressore tenta di inondare l' intera rete con messaggi MAC di controllo o pacchetti contenenti dati inutili ai propri vicini (attacco DoS) sfruttando la natura concorrenziale dell'accesso al mezzo. L'obiettivo è quello di consumare l'energia dei nodi nella rete, la larghezza di banda portando a rotture e peggioramenti di throughput.

- Routing Loop Attack: è un percorso ciclico che attraversa li stessi nodi più e più volte. Come conseguenza, questo tipo di attacco riduce le risorse di ogni nodo nel ciclo causando il loro isolamento.

- Packet Dropping: un nodo aggressore può bloccare la ricerca di un percorso, non inoltrando un messaggio e non replicando i pacchetti.

- ROUTE ERROR INJECTION: un nodo aggressore può ingannare il routing iniettando falsi pacchetti RERR facendo cadere i link. Questo attacco è particolarmente facile da attuare e non richiede conoscenze avanzate del routing.

5.5.2 Meccanismi di difesa

HWMP definisce un modello di sicurezza per link tra i nodi di una rete mesh Mesh security association (MSA), senza tuttavia considerare il collegamento end-to-end. B.A.T.M.A.N invece non specifica modelli di sicurezza tranne per il fatto che intrinsecamente il protocollo è immune ad alcuni attacchi. Comunque l'approccio utilizzato attualmente consiste nel riuso di meccanismi di sicurezza adattati. Ad esempio 802.11i, SEAD, ARN, SAODV. Si può affermare che la metodologia più usata è quella su soluzioni di crittografia e autenticazione dei messaggi.

5.5.2.1 Mesh security association (MSA)

IEEE 802.11s assicura la protezione dei link utilizzando il meccanismo Mesh Security Association (MSA). Questo meccanismo è una estensione del concetto, di autenticazione gerarchica tramite chiavi, implementato dallo standard 802.11i e utilizza per le autenticazioni lo standard 802.1X. Come funzionamento, MSA definisce una funzione detto mesh key holder, che può essere implementato dai MPs della rete mesh. ogni mesh key holder viene suddiviso in due tipi: mesh authenticator (MA) e mesh key distributor (MKD). un nodo MP può implementare entrambi, uno solo o nessuno dei due tipi. Come primo

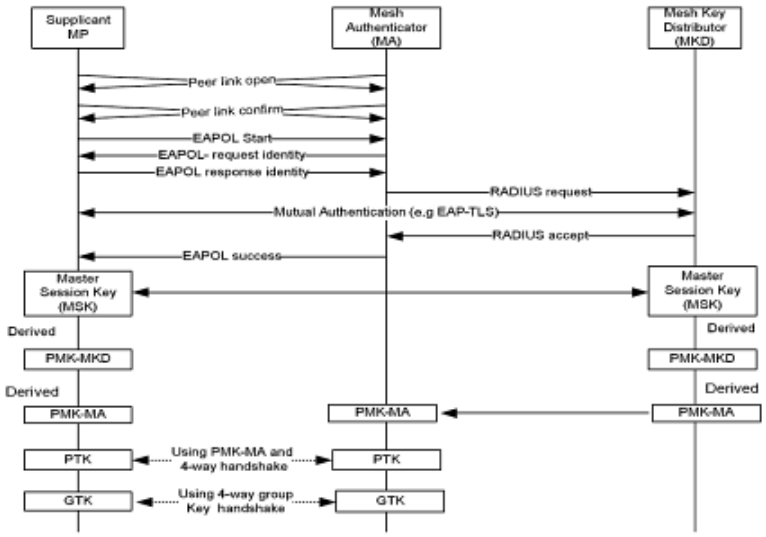


Figura 5.23: Procedura di Key establishment in IEEE 802.11s

stadio per l'impostazione di sicurezza un supplicant MP (nodo che accede) e un MKD devono ottenere un Pairwise Master Key(PMK)-MKD dal Master Session Key (MSK) creato nella fase iniziale di autenticazione tra il supplicant MP e MKD o da un pre-shared key (PSK) tra supplicant e MKD, se esiste. il secondo stadio consiste nell'ottene un PMK-MA da parte ancora di supplicant

MP e MKD. MKD allora invia il PMK-MA al MA permettendoli di iniziare la procedura MSA 4-way handshake, che come risultato estrae un Pairwise Transient Key (PTK) di 512 bits tra supplicant MP e MA. Durante MSA 4-way handshake, il MA riceve un Group Temporal Key (GTK) dal supplicant MP. Al completamento del MSA 4-way handshake, un handshake di groupo viene utilizzato per inviare il GTK del MA al supplicant MP. il GTK è una chiave condivisa fra tutti i supplicant MPs connessi allo stesso mesh autenticator (MA).

5.5.2.2 Secure Ad Hoc On-Demand Distance Vector (SAODV)

SAODV è una variante sicura di AODV. Utilizza delle catene hash per rendere sicura il campo hop count e firme digitali per la protezione dei campi non mutabili dei pacchetti RREQ e RREP. L'estensione difende da attacchi di impersonificazione, modifica dell' hop count e del sequenze number. SAODV funziona nel seguente modo. Durante il route discovery, il nodo sorgente genera un numero random detto *seed* e il massimo valore di hop count (MHC) viene settato dal valore di TTL presente nell'header IP. A partire dal seed il nodo crea un codice $hash = h(seed)$ e un $TOPHash = h_{MHC}(seed)$ che include nel pacchetto. un nodo che riceve il pacchetto verifica prima se TOPHash è uguale a $h_{MHC-Hopcount}(hash)$. Se sono uguale significa che il campo hop count non è stato alterato. A verifica effettuata il nodo incrementa di uno l'hop count, crea un nuovo codice $hash = h(hash)$ e ritrasmette il pacchetto ai nodi vicini. Eccetto il campo hop count niente dovrebbe cambiare nei pacchetti e tutti gli altri campi sono protetti da chiavi (firme) digitali. Il destinatario al riceve del RREQ crea un RREP nella maniera appena descritta e lo trasmette alla sorgente.

Capitolo 6

Casi di studio

6.1 Roofnet (MIT)

Figura 6.1: ROOFNET

Roofnet è un progetto di rete wireless sperimentale realizzato da una piccola comunità al MIT basato su tecnologia mesh. La loro rete è costituita da 50 nodi collocati in appartamenti a Cambridge, Massachusetts. Tutti i nodi sono equipaggiati con schede wireless 802.11b basati su chipset Intersil Prism 2.5 che lavorano alla frequenza di 2.422 GHz, con una potenza in trasmissione di 200mW. Ricevono il segnale radio da antenne omni-direzionale con 8 dB di guadagno. la disposizione dei nodi si può vedere nella mappa sopra. La topografia è tipicamente quella di una rete mesh e alcuni dei nodi fungono dai gateways che forniscono service DSL in sharing. La copertura radio è molto variabile ma tipicamente è di 100 m. tutti i nodi sono allacciati in modo

automatico ed auto configurante, e fanno uso di software unix con il protocollo di routing già caricato al livello kernel.

6.1.1 Accesso ad Internet

In roofnet c'è il principio per cui alcuni nodi forniscono volontariamente la loro connessione ad internet (tramite cavo). Cosi inizialmente, ogni nodo controlla se può raggiungere internet, tramite la propria porta ethernet chiedendo un indirizzo IP come client DHCP. Se può accedere, il nodo lo comunica alla rete roofnet, che lo vedo immediatamente come gateway. Se no può invece, cerca un percorso migliore per un gateway nella rete attraverso l'algoritmo di routing. Roofnet lavora al livello MAC, non ha quindi necessita di indirizzamento tramite IP, se no per una connessione esterna (con internet). Tutti i gateway implementano la funzione NAT che permette ai nodi di Roofnet di condividere un unico IP. Mentre nella rete stessa i nodi utilizzano indirizzi privati tipo 10.XX.XX.XX che con le funzionalità del routing permette di essere indipendenti dai Providers. Quando un nodo riceve tramite ethernet un pacchetto IP destinato a internet, lo incapsula nel proprio indirizzo 10.XX.XX.XX, e poi lo inoltra tramite il pacchetto di routing vero il gateway scelto. Il gateway de-capsula il pacchetto, e lo incapsula di nuovo nel proprio pacchetto IP che manda verso internet. Questo viene fatto per nascondere ad internet i nodi interni a roofnet.

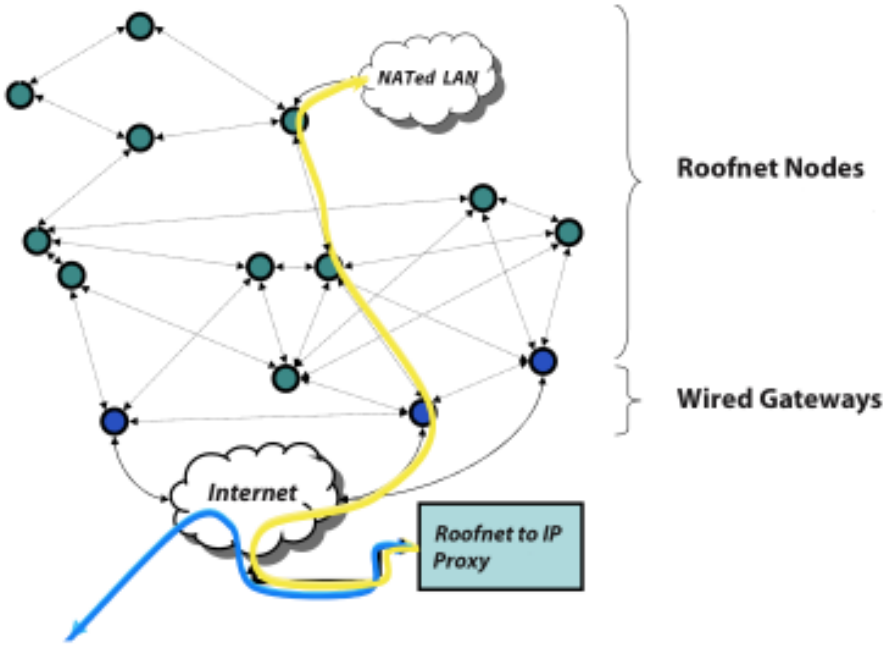


Figura 6.2: Roofnet e modalità di accesso ad internet

6.1.2 Protocollo

Roofnet usa un nuovo protocollo chiamato ScrRR, che cerca di trovare la rotta col piu alto throughput disponibile. ScrRR è un algoritmo inspirato a DSR dove un nodo n_0 crea un percorso verso un altro nodo n_d, inviando in broadcast pacchetti detti *query*. Tutti i nodi n_i che ricevono la query lo ritrasmettono aggiungendo il proprio ID e la metrica ETX per i propri link. Quando n_d riceve la query, manda un messaggio di conferma a n_0 tramite il percorso stabilito dai nodi intermedi che hanno il proprio ID nel source route della query. Il nodo n_0 (come i nodi nei vari percorsi) salva le informazioni dei link presenti nel messaggio di risposta in un database. Da questo database, in cui viene salvato anche la metrica dei link, si calcola il percorso usando l'algoritmo di Dijkstra. Se il percorso si rompe o comunque viene perso mentre il nodo n_0 sta trasmettendo pacchetti dati, n_0 se ne accorge in quanto dopo 10 insuccessi consecutivi (non ricezione ACK) riceve un metrica peggiore.

6.1.3 Valutazioni

Alcuni test di valutazione sono state effettuate su roofnet per capire le sue performance. A partire da [21] si riportano alcuni dei risultati ottenuti dalle misurazione.
Come metodo per la misurazione del throughput si stabilisce un trasferimento attraverso un collegamento TCP di 15 secondi tra ogni paio di nodi. come paramento per la misura si utilizza la quantità di dati in byte recapitati al livello applicazione di ogni nodo (in roofnet sono dei PC). Ogni collegamento è preceduto da un intervallo di 30 secondi, in cui il nodo sorgente invia un ping di 84 bytes al secondi per stabilire la rotta e misurare la latenza. Si considera anche che nella zona della misurazione ci sono altri nodi che possono allacciarsi per ottenere accesso ad internet da roofnet, che potrebbero fare parte del traffico. Il risultato ottenuto è presentato in figura 6.3. Dove si vede la funzione

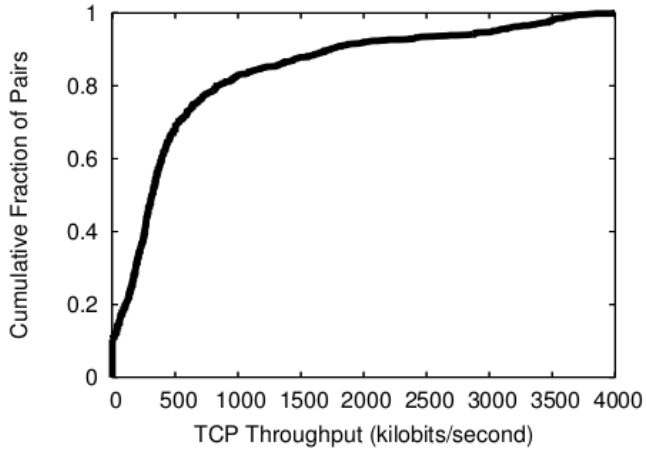


Figura 6.3: CDF del throughput TCP tra ogni paia di nodi della rete. (multi-hop TCP)

di distribuzione (CDF) del throughput attraverso il collegamento di ogni paio

di nodi della rete. é semplice notare che quando aumenta la probabilità di formare coppie di nodi sopratutto vicini, il throughput aumenta raggiungendo diversi kilobyte al secondo. Ed è ancora più semplice da capire dalla tabella in figura 6.4 che mette in evidenza sia il throughput che la latenza [tempo medio

Hops	Number of Pairs	Throughput (kbits/sec)	Latency (ms)
1	158	2451	14
2	303	771	26
3	301	362	45
4	223	266	50
5	120	210	60
6	43	272	100
7	33	181	83
8	14	159	119
9	4	175	182
10	1	182	218
no route	132	0	–
Avg: 2.9	Total: 1332	Avg: 627	Avg: 39

Figura 6.4: throughput TCP e round-trip ping latency tra i nodi della rete, relazionati al numero di hop. (multi-hop TCP)

che un pacchetto trascorre nella rete prima di arrivare a destinazione, dovuto principalmente a tempi di processo, back-off, ritrasmissione] in base al numero di hop. Questo risultato deriva dal fatto che la maggiore parte di nodi tende a collegarsi ai gateways per accedere ad internet, gateways che in roofnet sono collocate centralmente. Questo fa si che generalmente i nodi raggiungono tali gateways in pochi hops (solitamente 5) figura sotto.

Hops	Number of nodes	Throughput (kbits/sec)	Latency (ms)
1	12	2752	9
2	8	940	19
3	5	552	27
4	7	379	43
5	1	89	37
Avg: 2.3	Total: 33	Avg: 1395	Avg: 22

Figura 6.5: throughput e round-trip ping latency relazionato al numero hop prima di arrivare al gateway prescelto. (multi-hop TCP)

Un dato particolarmente importante riguarda la densità di nodi in un area, infatti è presentato il risultato del throughput del link tra tutti i nodi della rete grafico 6.6 (sopra) e il throughput dei link che il protocollo ScrRR sceglie più spesso grafico 6.6 (sotto). Dal risultato si vede che ScrRR opta per i nodi relativamente vicini con un alto throughput. È intuibile che nelle reti mesh più dense, sia ha più possibilità di interconnessioni e la sfida sta nel come i diversi protocollo implementano la scelta del percorso migliore.

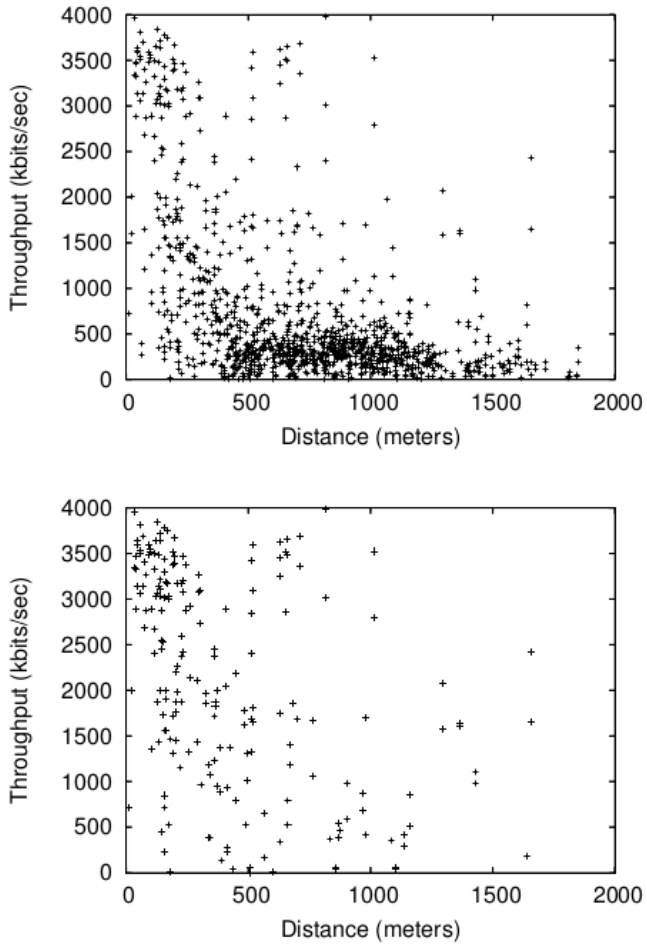


Figura 6.6: throughput di tutti link della rete in base alla distanza tra i nodi (sopra) e throughput dei link scelti da ScrRR (sotto) in base alla distanza. *ScrRR usa di più i link brevi con alto traffico. sia per single-hop TCP che per multi-hop TCP*

Capitolo 7

Conclusioni

La rete Wireless Mesh Network è sicuramente una delle candidate a diventare, la rete wireless standard nel prossimo futuro. Le peculiarità (basso costo, efficienza energetica, mobilità ed auto configurazione) di questa rete hanno spinto sia ricercatori che aziende a investire uno sforzo importante per permettere il suo utilizzo in situazioni reali e quotidiane. La strada da intraprendere per arrivare ad un risultato efficiente è difficile e laboriosa ma i benefici sono tangibili. Infatti molte zone rurali, città e aree metropolitane stanno realizzando reti di questo tipo. Città come San Francisco, Philadelphia ed alcune città italiane [24] con Ninux, comunità wireless mesh networks in Minnesota, Berlino, Cambridge, e New Mexico etc, e infine aree rurali in Sud Africa [22] e in Scozia [23] riportano ogni giorno risultati promettenti. La sfida maggiore per l'interconnessione che possa ritenersi davvero scalabile, è riuscire a creare uno standard che riesca ad unificare la grande eterogeneità delle WMN a tutti i livelli dell'ISO-OSI. In questa tesi si è cercato di capire e spiegare le diverse problematiche che colpiscono l'instradamento dei dati; il routing. Nello specifico gli algoritmi di routing, analizzando alcuni tra i più comuni, BATMAN e HWMP. Come detto, un forte tentativo della task group IEEE 802.11s è quello di ottenere con HWMP un protocollo efficiente, flessibile che sia in grado di adattarsi alle variabili caratteristiche delle reti mesh. Al con tempo la comunità di Freifunk con continui esperimenti ed esperienze reali conduce BATMAN a livelli di performance sempre più elevati e robusti. Roofnet è un importante banco di prova per quanto riguarda l'uso domestico di queste reti in ambiente cittadino. Si conclude facendo rimarcare che il successo delle WMNs dipende fortemente dagli algoritmi di routing, molti dei quali non possono funzionare con efficienza in ambienti in cui moltissime utenze utilizzano le stesse frequenza [ISM band]. L'effetto dell'interferenza deve essere trattato con particolare attenzione e i meccanismi di accesso al mezzo (MAC) devono realizzare soluzioni di allocazioni e riuso della banda di frequenze a disposizione con obbiettivo di garantire performance sufficienti a sopportare svariati tipi di dati. Questo aspetto enfatizza l'importanza di una metrica che segua in qualche modo le proprietà del canale e che ne riporti in tempi brevi, le misure sulla qualità in modo accurato. Porre l'attenzione su dati di tipo video e voce in questa nostra epoca permette una integrazione socio-educativa non di poco conto. Lavorare per mantenere un throughput elevato è un obbiettivo cruciale per l'espansione della rete mesh wireless.

Bibliografia

[1] Ian F. Akyildiz, Xudong Wang, Weilin Wang "Wireless mesh networks: a survey"

[2] W. Kiess and M. Mauv,e "A survey on real-world implementation of mobile ad-hoc networks"

[3] http://it.wikipedia.org/wiki/IEEE_802.11

[4] Joseph D. Camp and Edward W. Knightly, Electrical and Computer Engineering, Rice University, Houston, TX, "The IEEE 802.11s Extended Service Set Mesh Networking Standard"

[5] BWN lab wireless mesh networks research project, "http://www.ece.gatech.edu/research/labs/bwn/mesh/work.html"

[6] Yong Ding, Kanthakumar Pongaliur, Li Xiao, "Hybrid Multi-Channel Multi-Radio Wireless Mesh Networks"

[7] IEEE Std 802.11^{TM}-2012, IEEE Standard for Information technology, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 4.3.15 Mesh BSS: IEEE 802.11 wireless mesh network

[8] IEEE P802.11sTM/D1.07, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications

[9] http://www.ieee802.org/802_tutorials

[10] http://www.cwnp.com/wp-content/uploads/pdf/802.11s_mesh_networking_v1.0.pdf

[11] MeshTech'07 – "First IEEE International Workshop on Enabling Technologies and Standards for Wireless Mesh Networking"

[12] Zhifeng Tao and Shivendra Panwar, "Throughput and Delay Analysis for the IEEE 802.11e Enhanced Distributed Channel Access"

[13] Can Emre Koksal, "Quality-Aware Routing Metrics in Wireless Mesh Networks"

[14] http://git.open-mesh.org/batman-adv-doc.git

[15] http://www.open-mesh.org/projects/batman-adv/wiki/ELP

[16] http://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00

[17] http://tools.ietf.org/html/rfc3626

[18] http://tools.ietf.org/html/rfc3561

[19] David Johnson, Ntsibane Ntlatlapa, Corinna Aichele, "A simple pragmatic approach to mesh routing using BATMAN"

[20] Axel Neumann, Corinna, "Elektra" Aichele, Marek Lindner, "B.A.T.M.A.N Status Report"

[21] John Bicket, Daniel Aguayo, Sanjit Biswas, Robert Morris, "Architecture and Evaluation of an Unplanned 802.11b Mesh Network"

[22] Wireless Africa Home Page, "http://wirelessafrica.meraka.org.za/wiki/index.php/Wireless_Africa_Home_Page"

[23] Giacomo Bernardi, Peter Buneman, Mahesh K. Marina Tegola, "Tiered Mesh Network Testbed in Rural Scotland"

[24] http://wiki.ninux.org/PaginaPrincipale