Politecnico di Milano

V Facoltà di Ingegneria

Corso di laurea in Ingegneria delle Telecomunicazioni

Dipartimento di Elettronica e Informazione


# Progetto e implementazione di un protocollo di multicast per reti wireless mesh

Relatore: Prof. Antonio CAPONE

Correlatore: Ing. Alberto POLLASTRO

Tesi di Laurea di:

Lorenzo INVERNIZZI Matr. 740051

**Anno Accademico 2010-2011**

# Sommario

L' innovazione tecnologica ha permesso la diffusione sempre maggiore delle reti wireless a banda larga; partendo dal modello cablato si è passati allo sviluppo wireless della sezione d' accesso per giungere poi all' infrastruttura di backbone, dando luogo così a reti puramente senza fili dette reti wireless mesh.

Queste reti permettono di offrire servizi a banda larga ad utenti mobili, senza i vincoli di staticità di una rete tradizionale. Esse hanno bisogno di conseguenza di meccanismi che garantiscano *fast handover* e mobilità a livello di rete.

Per i servizi di rete più classici, già da alcuni anni sono state studiate e implementate diverse soluzioni a questa necessità; altri servizi, relativamente meno diffusi, non sono ancora supportati e per essi il problema resta aperto.

Uno di essi è il servizio di inoltro di flussi dati in modalità multicast; esso prevede che un dispositivo sorgente invii traffico IP non ad una tradizionale destinazione IP, ma ad un gruppo, per il quale i client interessati possono sottoscrivere un' iscrizione. Non si creano perciò tanti flussi quanti sono i client, ma un unico flusso che viene replicato dai router solo quando necessario, ovvero nei punti in cui l' albero di inoltro si dirama.

Obiettivo di questa tesi è la progettazione e l' implementazione di un' architettura che supporti il servizio di inoltro in multicast, tenendo contemporaneamente presente i requisiti di gestione della mobilità di una rete completamente wireless.

Per ottenere questo risultato, a partire dagli standard già esistenti è stato sviluppato un protocollo di comunicazione che permettesse ai vari elementi della

rete di coordinarsi. In seguito è stato implementato il processo che rende di fatto operativo il protocollo proposto.

L' architettura è stata infine impiegata in un testbed per verificare l' effettivo funzionamento e le prestazioni del protocollo implementato.

# Indice

# Elenco delle figure

# Capitolo 1

# Introduzione

Negli ultimi anni nel settore delle telecomunicazioni senza fili hanno preso sempre più piede le reti wireless di tipo mesh, che permettono di offrire ai terminali degli utenti la comodità dell' accesso a banda larga con mobilità, mantenendo i costi dei servizi offerti significativamente più bassi rispetto a quelli di una rete cablata. Le *Wireless Mesh Network* (WMN) sono reti auto-configuranti completamente wireless che si basano sull' utilizzo di nodi connessi a maglia tra loro. In queste reti il backhaul è quindi realizzato tramite collegamenti radio ed il traffico viene instradato in modalità multi-hop.

In particolare, in questo lavoro di tesi si è fatto riferimento alla rete *Mobi-MESH*, realizzata presso il Politecnico di Milano. Essa presenta una struttura innovativa che prevede due livelli logici distinti: il primo è una rete di backbone costituita da nodi in modalità ad hoc, mentre il secondo è una classica rete di accesso wireless. Su di essa è presente un algoritmo che permette di coordinare le informazioni di routing tra i due domini, e un sistema di gestione della mobilità che garantisce il *seamless roaming* all'interno di tutta la rete.

Le reti mesh hanno il vantaggio di una veloce e semplice dislocazione degli apparati, che per via del mezzo trasmissivo interamente wireless non richiede scavi o lavori molto onerosi, tuttavia non dispongono di risorse elevate, sia in

termini energetici che di banda.

Una delle soluzioni per ovviare alla scarsità di risorse sfruttando in maniera efficiente quelle disponibili è l' uso di un servizio di inoltro di tipo *multicast*: esso prevede che, nel caso in cui ci siano più utenti interessati ad un certo contenuto simultaneamente, ad esempio una multiconferenza o la trasmissione di un evento real-time, il terminale sorgente inoltri un solo flusso dati, indipendente dal numero dei destinatari, e che i router della rete provvedano a moltiplicare le informazioni solo quando necessario, ossia in corrispondenza delle diramazioni dell' albero di inoltro, fino a raggiungere gli utenti che hanno richiesto quel contenuto.

I più diffusi protocolli di routing multicast presenti in letteratura, tuttavia, non prevedono di fatto la gestione della mobilità dei client: un utente che si sposti da un access point ad un altro deve sottoscrivere ogni volta la sua partecipazione al gruppo di riceventi del traffico multicast.

Scopo di questo lavoro di tesi è l' implementazione sulla rete MobiMESH di una variante di protocollo per l' inoltro multicast con supporto alla mobilità dei client, in modo da garantire il minor ritardo possibile nella ricezione del flusso in caso di handover di un utente. Un altro obiettivo è quello di mantenere una certa interoperabilità con eventuali reti su cui è presente la versione standard del protocollo adottato, in modo che l' inoltro non sia vincolato alla sola area servita dalla rete MobiMESH.

Il lavoro di tesi è così organizzato. Nel Capitolo 2 sono illustrate le generalità delle *Wireless Mesh Network*, ponendo particolare attenzione allo standard della tecnologia Wi-Fi comunemente impiegata nelle implementazioni, e viene presentata la rete MobiMESH e la soluzione che propone per la gestione della mobilità dei client.

Nel Capitolo 3 viene offerta una panoramica sui protocolli disponibili per l' implementazione di servizi di multicast; inizialmente è descritto il funzionamento della segnalazione IGMP che viene usata dai client di multicast per comunicare con i router, in seguito vengono esposti i protocolli di routing multicast più

presenti in letteratura, evidenziando per ciascuno i pregi, i difetti ed il tipo di scenario in cui è più adatto.

All' interno del Capitolo 4 viene esposta l' idea generale della soluzione al problema affrontato in questa tesi; vengono descritte le ragioni per cui si è scelto di adottare il protocollo di PIM-SM, e le varianti e le procedure che sono state introdotte rispetto allo standard.

Il Capitolo 5 comprende l' analisi nel dettaglio dell' implementazione vera e propria del protocollo progettato; si descrive l' architettura software necessaria al funzionamento dell' algoritmo, le strutture dati e dei pacchetti di segnalazione, in particolare di quella introdotta per affrontare il problema di gestione della mobilità.

Le misurazioni, il parco software e l'architettura di testbed adottati per verificare l' effettivo funzionamento del protocollo implementato sono illustrati nel Capitolo 6, in cui vengono inoltre sottolineati alcuni aspetti degli esperimenti di verifica di cui tenere conto nell' analisi dei risultati ottenuti.

Infine il Capitolo 7 trae le conclusioni sui risultati ottenuti e propone possibili sviluppi futuri atti a migliorare la soluzione proposta in questa tesi.

# Capitolo 2

# Le reti mesh e MobiMESH

Negli ultimi anni il concetto classico di rete cablata è stato soppiantato dalla più moderna visione senza fili. Le reti wireless, più flessibili di quelle tradizionali, hanno conquistato grandi fette di mercato e sono tuttora in costante espansione. Un sottoinsieme di queste reti in forte ascesa è quello delle reti wireless di tipo mesh, dette Wireless Mesh Networks (WMN).

Questo capitolo illustrerà i principali aspetti delle WMN. Inizialmente ne verrà presentata l' architettura, per poi passare alla descrizione della tecnologia più diffusa ed utilizzata nella loro implementazione: il Wi-Fi (IEEE 802.11). Nell' ultima sezione, infine, verrà esposta una particolare realizzazione di tali reti (la rete *MobiMESH*) e le modalità con cui sono gestiti il routing e la mobilità all' interno di essa.

## 2.1 Le reti mesh wireless

In questo capitolo verranno esposte le caratteristiche salienti delle reti mesh wireless; verranno inizialmente presentati gli apparati tipici utilizzati nella realizzazione di una rete, ed saranno di seguito esposte le peculiarità che differenziano maggiormente le WMN dalle reti tradizionali. Saranno infine descritti gli scenari

di maggiore applicazione di questa tecnologia.

### 2.1.1 Struttura ed apparati

La struttura più diffusa di una WMN è quella ibrida, costituita cioè da una rete backbone multi-hop che collega tra di loro varie reti di accesso con tecnologie potenzialmente differenti, fornendo servizi di rete ad una gran varietà di dispositivi. La natura stessa di queste reti permette il supporto ad una vasta gamma di apparecchiature (può in pratica farne parte un qualsiasi dispositivo dotato di interfaccia wireless ed equipaggiato delle opportune funzionalità); è comunque possibile suddividere questo insieme eterogeneo in due macro gruppi:

- Mesh router (MR): dispositivi dotati tipicamente di bassa mobilità, che in genere non originano traffico dati ma vengono utilizzati come supporto ai servizi di rete. Una delle loro principali operazioni è la gestione dell' instradamento, necessaria a garantire l' inoltro di informazioni;
- Mesh client (MC): elementi più semplici dei precedenti, in genere possiedono maggiore mobilità, protocolli di comunicazione elementari, software/hardware semplificato e non supportano funzionalità avanzate per l' inoltro dei flussi dati. Sono le entità che generano e ricevono il traffico dati della rete.

Ogni MR è collegato attraverso più interfacce a diversi altri nodi e la connettività viene garantita attraverso la creazione di cammini multipli. L' integrazione delle WMN è facilmente attuabile; è possibile realizzare svariate soluzioni ibride, è semplicemente necessario che alcuni nodi siano equipaggiati con le più disparate tecnologie fornendo così il supporto, ad esempio, per reti di sensori, LAN e WLAN, reti ad-hoc, WiMAX, reti telefoniche cellulari e per la rete Internet.

### 2.1.2 Caratteristiche

Vi sono diverse caratteristiche che differenziano una WMN dalle normali reti LAN e WLAN, le principali sono le seguenti:

#### Affidabilità

L' elevato numero di cammini che esiste tra ciascuna coppia di nodi permette di ovviare in maniera relativamente agevole ad eventuali malfunzionamenti. Inoltre le capacità di self-healing (protezione automatica dai guasti) ne fanno aumentare sensibilmente la stabilità.

#### Ridotti costi di installazione

La possibilità di avere un' estesa copertura senza dover far fronte a dispendiose opere di allestimento (scavi, cablaggio, infrastrutture dedicate ecc.) mantiene ragionevolmente contenuti i costi in fase di avviamento. Inoltre la vasta scelta di dispositivi, presenti sul mercato ed adatti alle WMN, permette un' installazione della rete con investimenti il più ridotti possibile.

#### Ridotti costi di funzionamento e manutenzione

Le capacità di self-healing e self-forming (configurazione automatica della rete) consentono di limitare il numero di interventi da parte di personale specializzato; inoltre, la possibilità di ottenere la connettività creando celle di piccole dimensioni determina un sostanziale abbassamento delle potenze di trasmissione delle interfacce wireless portando, allo stesso tempo, vantaggi economici e risparmi energetici.

#### Facilità di integrazione

I MR sono dispositivi che possono essere integrati con molte delle tecnologie esistenti. Questo fa sì che le WMN possano operare sinergicamente con la quasi

totalità delle architetture presenti sul mercato.

### 2.1.3 Scenari applicativi

Inizialmente impiegate solo per la creazione di reti temporanee (come per esempio quelle utilizzate in situazioni di emergenza o in contesti bellici), le WMN hanno col tempo acquistato importanza ed hanno visto crescere la varietà di scenari in cui può risultare conveniente il loro utilizzo. I motivi principali che spingono alla scelta di una WMN sono l' opportunità di utilizzarla per il tempo strettamente necessario (tipico delle reti ad-hoc), la dislocazione in tempi relativamente rapidi ed il fatto che non è necessario ricorrere alla stesura di cavi per garantire un' adatta copertura della zona interessata. Come conseguenza si ha che le WMN vengono impiegate nei casi in cui il costo ed i tempi di realizzazione dell' equivalente cablato sarebbero superiori oppure nei casi in cui si vuole garantire una maggiore flessibilità. Il consistente numero di scenari utilizzativi in cui è possibile impiegare le WMN meriterebbe una trattazione a parte. Ci si limita ad illustrarne una piccola porzione.

#### Reti domestiche ed aziendali

Per supportare i più recenti servizi Internet, le utenze domestiche possono sfruttare le WMN ottenendo bande ragionevoli, maggior copertura e continuità del servizio. Le normali WLAN, attualmente utilizzate, forniscono l' accesso ad elevata velocità di trasmissione in aree però, talvolta, limitate. Non di rado si hanno delle zone parzialmente o completamente non coperte e l' installazione di ulteriori AP implicherebbe costi considerevoli in fase di configurazione e di installazione soprattutto per quanto riguarda la loro interconnessione. Uguali considerazioni possono essere fatte per uffici e contesti aziendali in cui le apparecchiature elettroniche sono molto più concentrate ed una modifica della struttura fisica degli

edifici comporterebbe la ristesura dei cavi ed il ripristino delle infrastrutture. Con l' utilizzo, invece, di MR-AP si potrebbe ovviare a questo problema.

### Condivisione della rete di distribuzione

Considerando una rete che fornisce servizi ad un comune o quartiere composto da diverse abitazioni, si può notare che soltanto l' ultimo tratto verso l' utente è puramente wireless, la rete di distribuzione è generalmente condivisa e cablata. In questo scenario, al crescere del traffico, si ha un degrado delle risorse di rete così come delle prestazioni. I cammini sono forzati e la possibilità che si verifichino dei rallentamenti o addirittura dei guasti è molto alta. Sostituendo questa architettura con una WMN si eliminerebbero gli svantaggi citati, garantendo inoltre una più diffusa connettività.

### Rapido supporto in situazioni di emergenza

In caso di emergenza, non è pensabile l' installazione di una rete, nell' accezione classica del termine, sia per un' eventuale impossibilità di raggiungere i luoghi interessati sia perchè non si ha a disposizione tempo a sufficienza. Una rete ad-hoc, mantenuta attiva, per il solo tempo necessario, potrebbe risultare estremamente utile ai soccorritori. Si pensi, per esempio, ad un edificio in fiamme in cui sono intrappolate delle persone. Anzichè setacciare i vari piani si potrebbe avviare una comunicazione (resa possibile anche in condizioni estreme dalla robustezza delle reti mesh) in modalità manuale o automatica, per una più celere ed accurata individuazione.

### Digital divide

Anche se al giorno d' oggi si è soliti pensare che l' accesso ad Internet sia di dominio pubblico, esistono paesi, comunità o intere regioni in cui questo non accade. I motivi possono essere i più disparati: difficoltà di essere raggiunti dalla

rete di distribuzione, reddito insufficiente o assenza di infrastrutture dedicate. Le WMN potrebbero venire incontro a queste esigenze. Dal punto di vista economico perchè garantiscono accesso condiviso, con conseguente ammortizzamento delle spese ed apparecchiature reperibili a buon mercato e dal punto di vista strutturale perchè non richiedono interventi sul territorio come scavi o collegamenti fissi.

### Automazione degli edifici e sistemi di sorveglianza

All' interno degli edifici (domestici e non) si ha un numero sempre crescente di apparecchiature informatiche. La possibilità di connettere fra loro tutti questi dispositivi aprirebbe orizzonti vastissimi. Si pensi infatti ad un controllo remoto dell' impianto di riscaldamento o alla gestione di sistemi di videosorveglianza e sicurezza. Le WMN sarebbero il supporto ideale per arrivare a questo obiettivo. Certo, la strada da percorrere è lunga, la comunicazione tra apparati diversi deve prevedere un protocollo che riesca ad unificare lo scambio di informazioni, ma poter disporre di uno strumento che getti le basi per lo sviluppo di questi scenari è già un grosso vantaggio.

### Reti metropolitane

L' installazione di WMN in aree metropolitane porterebbe numerosi benefici. L' area di copertura potrebbe essere più capillare (al limite anche nel sottosuolo), i costi di installazione notevolmente ridotti e si avrebbero i presupposti per una maggiore scalabilità. Inoltre si potrebbero fornire servizi a valore aggiunto come il videocontrollo o servizi informatici di pubblica utilità ed amministrazione altrimenti difficilmente realizzabili.

## 2.2 Standard IEEE 802.11

Le WMN definiscono un' architettura di rete e sono quasi totalmente slegate dalla tecnologia implementativa, infatti esistono diversi standard che possono essere impiegati. Tra questi i più diffusi sono Wi-Fi (IEEE 802.11 [1]), High Performance Radio LAN (HIPERLAN [2]) e Worldwide Interoperability for Microwave Access (WiMAX [3]). Nelle prossime sezioni si focalizza l' attenzione sullo standard Wi-Fi che è ad oggi quello più robusto, completo ed è di gran lunga il più utilizzato per la realizzazione di WMN.

### 2.2.1 Modalità operative

L' unità base di una WLAN è costituita dal Basic Service Set (BSS), il quale è definito all' interno di una Basic Service Area (BSA). Tutte le stazioni (ossia i terminali mobili) che si trovano in una BSA possono comunicare tra loro. La tipologia di comunicazioni ammesse tra le stazioni che compongono un determinato BSS è legata alla modalità con cui è stato creato:

- Independent BSS: Le stazioni appartenenti all' IBSS possono comunicare direttamente, purchè si trovino all' interno del raggio di corretta ricezione. In caso contrario possono ugualmente farlo ma devono utilizzare un' opportuno protocollo di routing. Tipicamente, gli IBSS sono formati da poche stazioni allestite per specifici scopi e per periodi di tempo limitati. Per questo motivo, spesso, si fa riferimento a queste tipologie di rete con il termine di reti ad-hoc. Questa modalità è spesso utilizzata nella sezione di backbone delle reti mesh implementate con 802.11.

- Infrastructure BSS: Si distinguono dalle precedenti per la necessità che sia presente un access point (AP). Le comunicazioni avvengono solo ed esclusivamente attraverso l' aP, anche se richieste tra due stazioni appartenenti alla stessa BSA. Questo fa sì che ogni scambio dati consti di almeno due

hop. Questo modello centralizzato limita l' estensione della BSA alla sola copertura e posizione dell' aP. Questo tipo di trasmissione ha lo svantaggio di richiedere maggiori risorse ma permette un maggior controllo della rete stessa. In questo caso una stazione, per poter usufruire dei servizi offerti dalla rete, deve innanzitutto avviare una procedura di associazione e solo dopo averla portata a termine con successo appartiene a quel determinato BSS e può comunicare con gli altri terminali. Per ampliare la copertura è possibile fare in modo che diversi BSS sovrappongano parzialmente le loro BSA e tramite il collegamento ad un Distribution System (DS) diano luogo ad un Extended Service Set (ESS). Tutte le stazioni all' interno dell' ESS possono comunicare tra loro come se appartenessero alla stessa WLAN.

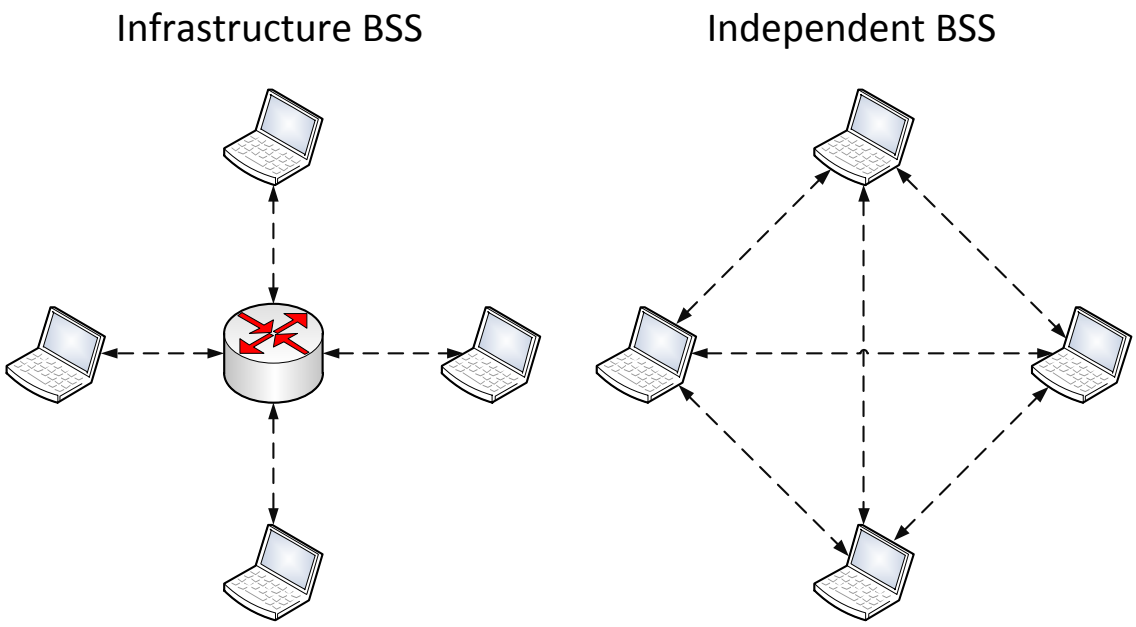



Figura 2.1: Modalità operative delle stazioni 802.11

### 2.2.2 Stack protocollare

Il protocollo 802.11 definisce solo i livelli più bassi della della pila Open System Interconnection (OSI) ([4]), vale a dire il Physical Layer (PHY) ed il Medium Access Control (MAC), il quale è una sottosezione del livello due, detto Data Link Layer (DLL).

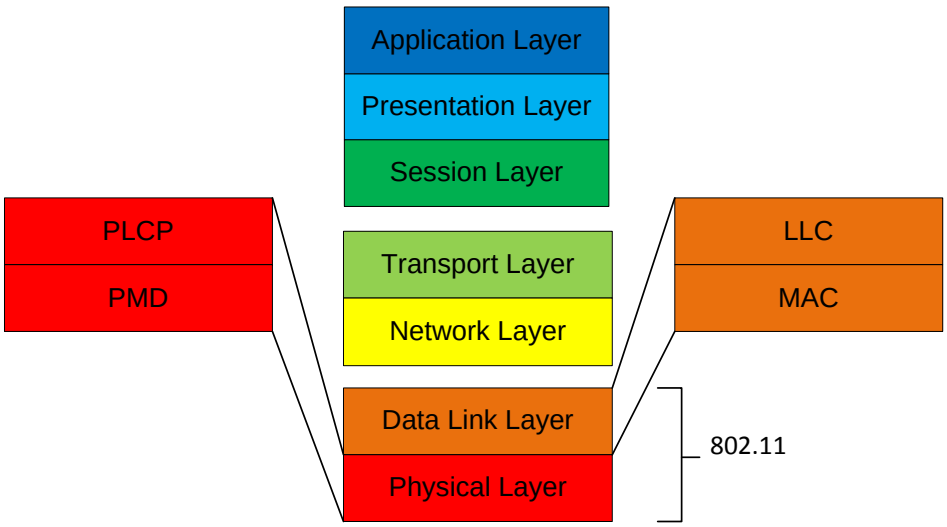



Figura 2.2: Stack protocollare OSI

#### Livello fisico

è il livello che si interfaccia direttamente con il mezzo wireless. Esso viene ulteriormente suddiviso in due sottolivelli:

- Physical Medium Dependent (PMD): è il sottolivello inferiore, realizza i meccanismi di trasmissione, ricezione e quelli necessari ad individuare quando il canale è libero. Il nome stesso indica che è strettamente connesso alla soluzione tramissiva per cui si è optato: IrDA, FHSS oppure DSSS.

- Physical Layer Convergence Procedure (PLCP): è la parte più alta del PHY. Questo sottolivello è stato pensato in modo da unificare le diverse PMD e presentarle come un unico livello fisico allo strato superiore. La principale funzione svolta è la traduzione della Medium Access Control Protocol Data Unit (MPDU) nel formato opportuno per la trasmissione, per esempio inserendo un header all' inizio della trama in base al mezzo fisico selezionato.

**Livello MAC**

è il livello che si occupa di regolare l' accesso multiplo al canale. Si presenta come supporto per due diversi tipi di rete, quelle ad-hoc e quelle infrastructure. Il meccanismo basilare con cui viene controllato l' accesso al mezzo condiviso è il sensing del canale che verrà esposto dettagliatamente nella seguente sezione.

### 2.2.3 Accesso al mezzo fisico

Il canale wireless è per sua stessa natura un mezzo condiviso. Per poter permettere l' accesso a diversi utilizzatori è indispensabile un meccanismo di coordinazione. Le tecniche di Collision Detection (CD) già esistenti ed ampiamente utilizzate nelle reti LAN non sono sufficienti, in quanto le caratteristiche di attenuazione del mezzo radio sono diverse da quelle del caso cablato. Ogni stazione sente le proprie trasmissioni con una potenza di gran lunga superiore a quelle delle altre, e il risultato è l' oscuramento di quest' ultime senza la possibilità di individuare eventuali collisioni. In aggiunta si ha poi l' insorgere di nuove problematiche tipiche del caso wireless, la più nota delle quali è la situazione del "terminale nascosto". In questo caso due stazioni lontane che non possono rilevarsi a vicenda ma che appartengono alla stessa BSA cominciano a comunicare con lo stesso AP, collidendo, senza possibilità di accorgersene, su quest' ultimo. Lo standard 802.11 propone diverse funzionalità a livello MAC per poter far fronte a questo e ad altri problemi. Tra queste, una delle più diffuse è la Distributed Coordination

Function (DCF), basata sul Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). Secondo questo sistema di coordinamento, una qualunque stazione che vuole trasmettere per prima cosa deve verificare se ce ne sono altre che stanno trasmettendo, e, se riconosce la presenza di trasmissioni, si mette in attesa. Quando il mezzo si libera attende che rimanga tale per un intervallo di tempo minimo detto Distributed Inter Frame Space (DIFS), dopodiché inizia una fase di contesa per l' utilizzo del mezzo: la stazione sceglie un secondo intervallo, casuale, chiamato backoff, al termine del quale, se il mezzo è ancora libero, inizia la trasmissione. L' intervallo di backoff serve a ridurre la probabilità di collisione quando, alla fine di una trasmissione, ci sono molte stazioni in attesa che il mezzo si liberi. Se una stazione, in attesa che termini il periodo di backoff, sente che il mezzo non è più libero, congela il tempo residuo. Quando poi il mezzo torna libero per un tempo pari ad un DIFS, la stazione non sceglie un nuovo intervallo di backoff ma termina il precedente. Il meccanismo di backoff non esclude la possibilità di trasmissioni contemporanee, e quindi di collisioni. Per prevenire questa eventualità e realizzare la Collision Avoidance (CA), il protocollo prevede anche un CS virtuale. Per poterlo realizzare si utilizzano i messaggi Request To Send (RTS) e Clear To Send (CTS). Quando una stazione trova libero il mezzo allo scadere del tempo di backoff, non invia subito il dato, bensì una trama di tipo RTS. Se riceve dal destinatario una risposta CTS allora procede con l' invio del messaggio, altrimenti suppone che si sia verificata una collisione e si mette in attesa per riprovare. Per evitare che durante l' invio di questi due particolari messaggi ci si trovi nel mezzo di un tempo di attesa di un' altra comunicazione si utilizzano intervalli minori del DIFS detti Short Inter Frame Space (SIFS). Un ultimo controllo viene poi effettuato al termine della trasmissione. La stazione che ha ricevuto i dati, invia un messaggio di Acknowledge (ACK) per confermare la buona riuscita dell' operazione. Quando è in corso una trasmissione, secondo il protocollo, tutte le stazioni non interessate dovrebbero sentire il mezzo occupato. Tuttavia, a causa della bassa affidabilità della trasmissione, e della lontananza

tra i terminali, una stazione potrebbe non ricevere i messaggi e iniziare una comunicazione con conseguente collisione. Nelle trame RTS e CTS allora, vengono incluse informazioni sulla durata della trasmissione successiva, in modo tale che le stazioni non interessate alla ricezione possano caricare, in un registro detto Network Allocation Vector (NAV). Tale registro viene via via decrementato ed ogni stazione ne attenderà l' azzeramento prima di cominciare la procedura di trasmissione. Dal momento che il CTS viene trasmesso dalla stazione di destinazione, le informazioni sulla durata raggiungono sia le STA vicine alla destinazione che quelle vicine alla sorgente. L' utilizzo delle trame RTS/CTS ha però due controindicazioni: innanzi tutto, se la trama dati è corta, l' overhead introdotto può essere eccessivo; inoltre, non è applicabile nel caso dei comunicazioni multicast e broadcast in quanto più di una stazione potrebbe rispondere con il corrispondente CTS. Esiste pertanto la possibilità (obbligatoria per trame al di sotto di una certa dimensione definibile a priori) di effettuare la trasmissione dati immediatamente allo scadere del tempo di backoff, se il mezzo è ancora libero. In questo caso è naturalmente possibile che una collisione impedisca la corretta trasmissione dei dati.

## 2.3 La rete MobiMESH

Come diverse altre soluzioni, quella sviluppata al Politecnico di Milano con il nome MobiMESH ([5], [6]) rappresenta una WMN ibrida. Essa è fondamentalmente costituita da un backbone al quale vengono interconnesse più reti di accesso, le quali permettono l' interfacciamento con un eterogeneo insieme di dispositivi, dotati delle più diverse tecnologie. MobiMESH è stata implementata in un testbed utilizzando lo standard per le comunicazioni radio maggiormente diffuso sul mercato: 802.11. In particolare si utilizza 802.11b/g ([7]) per la sezione di accesso e 802.11a ([8]) per la parte backbone. Questa rete non è legata ad una particolare specifica di livello due in quanto le funzionalità richieste sono esclusivamente l'

abilità di associarsi e disassociarsi ai punti di accesso, perciò un qualsiasi standard che garantisca queste funzionalità può essere utilizzato per implementarne l' architettura. Nel prosieguo verrà presentata l' architettura di rete; successivamente si descriveranno le tecniche che vengono attuate per garantire la mobilità dei terminali wireless all' interno di questa rete ed il protocollo di routing che si occupa della determinazione delle rotte.

### 2.3.1 Architettura di rete

Caratteristica primaria della rete MobiMESH è quella di apparire a tutti gli effetti come un' unica grande rete 802.11. Un client che vi si affaccia per la prima volta non percepisce alcuna differenza rispetto ad una normale rete WLAN; la trasparenza è infatti il principio base su cui è stata costruita. Come ogni rete puramente wireless non dispone di una topologia definita entro precisi confini fisici. È comunque sempre possibile distinguere due distinte sezioni separate anche nello spazio di indirizzamento IP: la sottorete mesh backbone e la sottorete di accesso. Nel prosieguo verrano descritte dettagliatamente le tipologie di nodo che popolano questi due segmenti di rete. Infine verrà presentata anche una terza entità che si occupa di attuare il coordinamento della rete stessa.

#### 2.3.1.1 Terminali mobili

I terminali mobili, detti anche Mobile Host (MH), rappresentano la stazione utente. Per accedere ai normali servizi di rete, è sufficiente quindi un qualsiasi dispositivo conforme con le specifiche implementative della sezione di accesso. I MH non partecipano alle operazioni di gestione della rete, di cui si occupa la sezione di backbone, ma fanno parte esclusivamente della sottorete di accesso; per farlo è sufficiente che ottengano, tramite le classiche richieste DHCP, un indirizzo IP appartenente allo spazio di indirizzamento dedicato. Tutte le funzionalità di gestione della mobilità hanno sede nella parte backbone, in modo da garantire

la mobilità senza aumentare la complessità dei MH, tenendo anche conto che tipicamente questi non sono sotto il controllo diretto di chi gestisce la rete (a meno di porre vincoli aggiuntivi sulla trasparenza e sulla disponibilità dei servizi, a seconda della tipologia di terminale che li richiede).

#### 2.3.1.2 Mesh router

I nodi della sottorete di backbone appartengono tutti allo stesso spazio di indirizzamento, differente da quello utilizzato dai terminali mobili (MH). L' instradamento dei pacchetti all' interno del dominio backbone avviene a livello tre, mediante la creazione di una rete multi-hop magliata. La mobilità dei router, a differenza di quella dei client, è praticamente nulla ed in genere non è necessario che effettuino alcuno spostamento. Il protocollo di routing monitora continuamente la topologia e garantisce il riarrangiamento a seguito di modifiche, come l' aggiunta o la rimozione di nodi. Tale protocollo permette a ciascun nodo, dopo la fase iniziale di avvio in cui si inizializza e si autoconfigura acquisendo un proprio indirizzo IP, di essere in grado di raggiungere tutti gli altri nodi della rete. L' algoritmo di routing unicast tuttora utilizzato è OLSR, ma il meccanismo di gestione dell'handover è completamente indipendente da tale protocollo, e nulla vieta di usare una qualsiasi altra soluzione per l' instradamento.

#### 2.3.1.3 Gestore dei mappaggi

Un determinato nodo, detto Mapping Server (MS), svolge particolari operazioni di fondamentale importanza per il funzionamento della rete stessa. Oltre ad includere le funzionalità tipiche di instradamento comuni a tutti i MR, questo nodo ospita un server DHCP; è infatti compito di MS assegnare ogni indirizzo presente in MobiMESH, sezione di accesso inclusa. Le modalità con cui questo avviene sono differenti in base alle sezione in cui si trova il nodo che ne fa richiesta. Un MR richiede al MS il proprio indirizzo IP nel momento in cui la rete viene creata

e non necessita di doverlo cambiare per tutto il tempo in cui resta operativo. Un MH non può invece contattare direttamente MS, ma si deve appoggiare alle funzionalità di relay presenti nei MR-AP (mesh router con funzionalità di access point). Ciascun MH si vede assegnare l' indirizzo IP dallo specifico MR-AP cui è associato senza alcuna percezione che vi sia un controllo di queste informazioni da parte di MS. Inoltre la validità delle lease rilasciate agli host della rete di accesso è di durata inferiore rispetto a quelle dei nodi del backbone; questo perchè la maggior mobilità ed il minor tempo di permanenza all' interno di MobiMESH fanno sì che il tempo medio di vita di un MH e quindi dell' indirizzo IP ad esso assegnato sia di gran lunga inferiore a quello di un qualsiasi MR. In questo modo è possibile un riuso di questi indirizzi con una maggiore flessibilità e controllo della rete. Oltre all' assegnamento di questi indirizzi a livello di rete, MS si occupa anche di tracciare le corrispondenze tra questi indirizzi di livello tre (IP) e quelli di livello due (MAC), propri dello specifico dispositivo che ne ha fatto richiesta. Per poterle utilizzare dinamicamente, ad ognuna di esse è legato un timer allo scadere del quale vengono ritenute non valide. In ogni istante dunque, all' interno di un opportuno database di MS sono presenti tutte le coppie MAC-IP di ogni nodo appartenente a MobiMESH.

### 2.3.2 Gestione della mobilità

Un client (MH) accede alla rete tramite l' opportuno router che gestisce l' area in cui è presente (MR-AP). Dopo aver instaurato un collegamento di livello due, l' host richiede un indirizzo IP per poter utilizzare i servizi di rete. Il router risponde appoggiandosi al server dedicato all' assegnamento degli indirizzi (MS) e fornisce all' utente tutte le informazioni necessarie per operare all' interno della rete. Il router è inoltre responsabile affinchè vi sia piena visibilità dell' utente da parte di ogni altro nodo (MR) della rete. Tale operazione viene attuata mediante l' invio di annunci di associazione all' interno della sezione backbone. In questi annunci

viene comunicato il mappaggio MAC-IP dei client. Questi messaggi giungono anche ad opportuni gateway (MG), che si occupano di garantire la connessione dei vari client e dei relativi router ad Internet. L' utente, completamente inconsapevole di questa struttura di alto livello, può comunicare liberamente come fosse collegato ad una classica rete locale. In aggiunta però è libero di muoversi all' interno di tutta l' area in cui è presente la copertura MobiMESH. Nel momento in cui il client, spostandosi, si associa ad un differente nodo della sezione backbone, innesca la procedura che garantisce la mobilità in rete. Il nuovo router si accorge che un utente ha effettuato un' associazione e recupera il corrispondente mappaggio tra gli indirizzi di livello due e livello tre. Questa informazione, grazie agli annunci precedenti, viene ritrovata con un costo pressochè nullo, all' interno del proprio database locale. Il nuovo router comincia così ad annunciare il client come proprio associato, provocando un aggiornamento in tutti i nodi della rete, fra i quali vi è anche il router che aveva visto per primo la sua associazione. Questa operazione fa sì che vengano modificate tutte le rotte che avevano come destinazione quell' utente reindirizzandole attraverso il tunnel che porta al router appropriato. L' utente continua così a ricevere ed inviare pacchetti attraverso il nuovo nodo di accesso, mantenendo attive tutte le connessioni precedentemente stabilite. La mobilità viene gestita a livello distribuito dai router stessi, non c' è bisogno di appoggiarsi ad alcun altro nodo. Tuttavia dopo che il roaming è stato portato a termine con successo, il router chiede conferma al nodo centrale per estendere la validità delle informazioni in suo possesso legate al client.

### 2.3.3 Protocollo di routing (OLSR)

Tra i protocolli proattivi sviluppati per reti ad-hoc merita particolare attenzione quello ideato in Francia dal gruppo HINRIA con il nome di Optimized Link State Routing (OLSR) ([9]). Questo protocollo si basa sulle classiche caratteristiche dei normali link state presentando però due fondamentali ottimizzazioni. Nei proto-

colli puramente link state le informazioni sulla topologia vengono scambiate tra i vari nodi che si informano su tutti i collegamenti a disposizione con i loro vicini. In OLSR invece vengono dichiarati solamente quei collegamenti che ogni nodo ha con un relativo sottoinsieme di questi vicini, detti Multipoint Relay (MPR). In questa maniera si ottiene una forte diminuzione delle dimensioni dei messaggi di controllo. In secondo luogo viene ottimizzata anche la diffusione di questi messaggi in rete senza affidarsi al classico flooding. Ogni nodo invia questi messaggi in broadcast solo al primo hop dopodichè utilizza unicamente i suoi MPR per effettuarne l' inoltro ottenendo una sostanziale riduzione dell' overhead del traffico. Un apposito algoritmo determina quali siano i nodi che devono far parte dell' insieme MPR in modo che tramite essi sia possibile raggiungere ogni parte della rete evitando la creazione di cicli. Le informazioni sulla topologia vengono scambiate periodicamente senza la generazione di traffico extra in risposta alla creazione o alla scomparsa di link. In questa maniera ogni elemento della rete ne ha a disposizione una completa descrizione e può dunque calcolare, tramite i classici algoritmi, la via migliore per giungere ad una qualsiasi destinazione. Una volta trovata, la rotta viene inserita nella tabella di routing che in questo modo rimane costantemente aggiornata.

### Messaggi di controllo

I messaggi di controllo vengono inviati in broadcast, utilizzando UDP sulla porta 698, assegnata ad OLSR dalla Internet Assigned Number Authority (IANA). Essi sono di tre tipi:

- *Hello*: viene utilizzato per ricavare i dati sul canale, individuare quali collegamenti sono simmetrici, quali sono i nodi limitrofi ed eleggere quali di questi faranno parte del MPR;
- *Topology Control*: è un messaggio di tipo link state. La sua funzione è quella di diffondere in rete le informazioni sulle caratteristiche dei collegamenti e

sull' identità dei nodi vicini al mittente. Grazie alla tecnica di inoltro MPR viene ricevuto da ogni nodo della rete stessa;

- *Multiple Interface Declaration*: questo messaggio viene inviato solamente nel caso in cui OLSR stia funzionando su più interfacce. In questo messaggio si elencano gli indirizzi delle varie interfacce posseduti dal nodo. Con modalità analoga al precedente raggiunge ogni nodo della rete.

### Ricostruzione della topologia

La prima azione svolta dai nodi è quella di inviare dei messaggi vuoti di Hello ad intervalli regolari con l' intento di annunciare la propria presenza. Quando ricevuti per la prima volta, questi messaggi causano l' aggiunta del nodo nell' insieme dei vicini del ricevente. Come conseguenza di ciò il nodo inizia ad inviare messaggi di Hello includendo anche la lista di questi vicini. Questa operazione fa sì che ogni nodo sia consapevole dei collegamenti simmetrici che lo circondano. I messaggi di Hello ricevuti non vengono inoltrati. Per fare in modo che la topologia di rete sia nota a tutti, ogni nodo invia messaggi di Topology Control, nei quali viene inclusa la lista dei vicini attivi. Questi messaggi vengono inoltrati secondo l' algoritmo definito in fase di configurazione. Quando ricevuti, questi messaggi, causano l' aggiornamento delle informazioni sulla topologia in ogni nodo della rete. Un' estensione standard di OLSR permette inoltre ai nodi di annunciare STA appartenenti ad una rete esterna o comunque di annunciare delle sottoreti IP in cui non viene utilizzato OLSR. Queste informazioni viaggiano nei messaggi di Host and Network Association (HNA) che includono l' indirizzo della sottorete e la sua netmask. In questo modo, utilizzando OLSR, è possibile estendere il dominio della rete.

# Capitolo 3

# Multicast IPv4

Considerando la limitata disponibilità di risorse radio che caratterizza in genere le reti di tipo wireless, è sempre apprezzabile l'ottimizzazione dell'uso della banda. In quest'ottica risulta particolarmente interessante la distribuzione di pacchetti in modalità multicast: essa prevede l'inoltro simultaneo di un flusso di dati da una sorgente ad un insieme (dinamico) di destinazioni, inviando un solo pacchetto e replicandolo laddove risulta essere necessario.

I campi più tipici di applicazione del multicast sono ricezione di streaming audio/video, multiconferenze, resource discovery, diffusione di contenuti multimediali e di dati, ed in generale situazioni in cui vi è necessità di distribuire contemporaneamente la medesima informazione o lo stesso contenuto a più destinatari.

In IPv4, i client in una rete che desiderino ricevere un certo flusso di traffico multicast devono sottoscrivere una registrazione per quel flusso, identificato da un indirizzo IP multicast (classe D degli indirizzi IP). A lato rete, un apposito algoritmo di routing implementato sui nodi si occupa della creazione e del mantenimento di un albero di distribuzione, in cui il flusso multicast viene duplicato solo quando un ramo si biforca.

Nel paragrafo seguente descriverò lo spazio di indirizzamento riservato al traf-

fico multicast in IPv4; seguirà la descrizione del protocollo IGMP usato dai client per sottoscrivere la registrazione ad un gruppo multicast, ed infine una panoramica dei protocolli di routing multicast più diffusi.

## 3.1 Indirizzi Multicast

Al traffico multicast è stata riservata una classe apposita di indirizzi IP, chiamata "classe D": questa comprende tutti gli indirizzi la cui sequenza di bit inizia con "1110" (224.0.0.0/4). All'interno di una rete, ciascun gruppo di multicast è associato ad uno di questi indirizzi. Esistono, nella classe D, alcuni indirizzi riservati ad applicazioni specifiche; i più noti sono:

- 224.0.0.1 – corrisponde al gruppo "tutti gli host"; se si invia un ping verso questo indirizzo, tutti gli host della rete (se abilitati a farlo) risponderanno all'echo request;
- 224.0.0.2 – corrisponde al gruppo "tutti i router" della subnet; ogni router (multicast) della rete deve iscriversi a questo indirizzo su ciascuna delle sue interfacce su cui è abilitato il traffico multicast;
- 224.0.0.4 – corrisponde al gruppo di nodi che implementano il protocollo di routing multicast *Distance Vector Multicast Routing Protocol* (DVMRP);
- 224.0.0.13 – indica tutti i router della rete che supportano il protocollo *Protocol Independent Multicast* (PIM) versione 2;
- 224.0.0.22 – è usato dai client che implementano il protocollo *Internet Group Management Protocol* (IGMP) versione 3 per inoltrare le richieste di join/leave verso un qualsiasi altro gruppo di multicast;
- 224.0.1.241 – usato per la discovery degli indirizzi dei gatekeeper H.323;

eccetera. Il range di indirizzi 224.0.0.0/24 è in ogni caso riservato per scopi locali, amministrativi e di manutenzione, ed i pacchetti destinati a questi indirizzi non vengono inoltrati dai router multicast.

A livello data-link, lo spazio di indirizzamento riservato al multicast è ottenuto dalle permutazioni dei 23 bit meno significativi dell' indirizzo MAC 01-00-5E-00-00-00. Si presenta quindi il problema del mappaggio di un indirizzo multicast di livello 3 sul livello 2: visto che il numero indirizzi riservati nei due livelli non è la stesso (28 bit disponibili per lo spazio di indirizzamento multicast di livello 3, 23 bit per il livello 2), ad ogni indirizzo MAC corrispondono 32 indirizzi IP. La soluzione adottata consiste nel mappare i bit meno significativi del livello 3 sui bit disponibili per l'indirizzamento multicast del livello 2. Inevitabilmente, si può verificare che un client riceva ed elabori un pacchetto MAC destinato ad un gruppo di cui non fa parte; è compito del kernel del client, una volta esaminato l'header IP, scartare il pacchetto.

## 3.2 IGMP

Qualunque sia lo specifico algoritmo di routing multicast utilizzato, affinchè i nodi della rete creino gli opportuni alberi di inoltro dei flussi è necessario che i client comunichino al nodo/access point a cui sono associati la propria iscrizione ad un dato gruppo [10]. Ciò avviene, in IPv4, tramite il protocollo *Internet Group Management Protocol* (IGMP). Esso prevede che un IGMP querier (il router di una sottorete designato a gestire il protocollo) invii periodicamente interrogazioni (IGMP query) per conoscere i gruppi a cui i client (gli host a cui è direttamente connesso) vogliono iscriversi. I client comunicano quindi il proprio "interesse" verso un gruppo inviando un primo annuncio di join per quel gruppo, e poi rispondendo con degli IGMP report alle query periodiche dell' IGMP querier. Indipendentemente dallo specifico tipo di pacchetto o versione del protocollo, i

messaggi IGMP non vengono inoltrati, sono scambiati solo a livello di rete locale tra i router e gli host a loro direttamente connessi.

Sebbene dalla fine del 2002 sia stato proposto lo standard IGMPv3 [11] per la sottoscrizione a gruppi da parte dei client, le funzionalità aggiuntive introdotte rispetto alla versione 2 [12] del protocollo non sono ancora molto sfruttate e la gran parte delle applicazioni implementa ancora il vecchio IGMPv2. Nei paragrafi 3.2.1 e 3.2.2 descriverò quindi entrambe le versioni del protocollo evidenziandone le differenze, particolarmente rilevanti per comprendere alcune scelte che sono state operate in questo lavoro di tesi.

### 3.2.1 IGMPv2

In questa versione del protocollo sono previste tre tipologie di messaggio: query, report e leave. La struttura dei pacchetti è comunque sempre la stessa, e dalla dimensione fissa di 8 byte; a seconda del tipo di messaggio trasportato in un pacchetto, i campi *max response time* e *group addresses* vengono utilizzati oppure settati col valore 0. Di seguito saranno descritti il significato dei tre differenti messaggi citati e le modalità del loro utilizzo.

#### Query

Le interrogazioni sono inviate dai querier IGMP a tutti gli host della sottorete a cui sono assegnati, e si dividono nelle sottocategorie di *general query* e *group specific query*.

Le general query servono a richiedere a tutti gli host della sottorete di rendere noti al proprio router di riferimento gli IP dei gruppi di multicast di cui desiderano fare parte. Poichè non sono riferite a dei client di un gruppo in particolare, devono essere ricevute ed elaborate da tutti gli host direttamente connessi al querier; hanno quindi come indirizzo di destinazione del pacchetto IP che le contiene il gruppo multicast *all systems group* 224.0.0.1, ed il campo *group address* del

pacchetto IGMP viene lasciato con i bit settati a zero. Il valore di *max response time*, può venire impostato dall'amministratore della rete, di default è di 10 secondi.

Le group specific query sono rivolte solo ai client di un dato gruppo, pertanto l'indirizzo IP di destinazione coincide con quello del campo di *group address*, che chiaramente indica il gruppo multicast che il querier desidera interrogare. Il valore di *max response time*, anche in questo caso modificabile, è di default di un secondo.

### Report

I messaggi di *report* sono inviati da un client appena questo decide di entrare a far parte di un qualche gruppo multicast, oppure in risposta ad un messaggio di general query o di group specific query per un gruppo di cui il client fa già parte. Per segnalare la propria adesione ad un certo gruppo il mittente del messaggio riempie il campo *group address* del pacchetto e il *destination address* del datagramma IP con l'indirizzo del gruppo di cui vuole far parte; si noti che con un messaggio di IGMPv2-report viene sottoscritta la partecipazione ad uno ed un solo gruppo, visto che il pacchetto è di lunghezza fissa e per il campo *group address* sono disponibili solo 32 bit.

### Leave

I messaggi di *leave* sono inviati da un client quando questo vuole annullare una precedente sottoscrizione ad un un gruppo multicast. Tale gruppo è indicato nel campo *group address*, ed il pacchetto IP che contiene la leave è destinato al gruppo multicast *all-routers-group* (224.0.0.2), in modo che possa essere ricevuto e letto da tutti i router multicast connessi al client.

### Aspetti fondamentali del protocollo

Appena un client decide di entrare a far parte di un gruppo, lo comunica al proprio router di riferimento tramite l'invio di un messaggio di report, senza attendere di ricevere una query che lo solleciti. In maniera speculare, se un membro di un gruppo di multicast non desidera più farne parte, spedisce immediatamente un messaggio di leave. Quando un router designato elabora una notifica di leave, invia agli host della LAN una group specific query per verificare se restano comunque altri membri del gruppo segnalato.

E' importante rimarcare che tutti i messaggi IGMP sono incapsulati in datagrammi IP che presentano un indirizzo di destinazione della classe D: non sono dunque pacchetti inviati in unicast da un querier ad un singolo host o viceversa, ma pacchetti multicast che vengono ricevuti ed elaborati da tutte le macchine che fanno parte del gruppo indicato. Questo è fondamentale per il meccanismo di report suppression presente in IGMPv2: quando un router invia general o source specific query, al massimo un client per ogni gruppo risponderà: ricevuta una query, ogni host interessato inizializza un timer con un valore estratto a caso tra 0 ed il max response time; se allo scadere del tempo non avrà ricevuto alcun messaggio di report da altri client per quel gruppo, procederà con l'invio del proprio. Se invece prima del timeout vedrà il report di un altro client per il medesimo gruppo, azzererà il timer e non invierà alcun pacchetto. Questo meccanismo è necessario per evitare effetti di *packet storm* in seguito ad una query: senza di esso, in risposta ad una general query si avrebbero tanti report quanti fossero i gruppi sottoscritti da ogni client, per ogni client della sottorete, con probabili effetti di collisione dei pacchetti. D'altra parte, ciò comporta che ciascun querier sappia soltanto se esiste almeno un client a lui connesso interessato a ricevere i pacchetti indirizzati ad un certo gruppo multicast, e non quali client nello specifico vogliano riceverli. Come si vedrà in seguito, questo inconveniente ha inciso in maniera determinante sulla scelta del protocollo di sottoscrizione ai gruppi usato

nella soluzione proposta di multicast con mobilità.

### 3.2.2 IGMPv3

IGMP versione 3 introduce principalmente due novità rispetto alla versione precedente: la possibilità per i client di specificare un sottoinsieme di sorgenti per un dato gruppo da cui vuole ricevere traffico, e la rimozione del meccanismo di *report suppression*. Tali aggiunte comportano un incremento delle dimensioni dei pacchetti e alcuni cambiamenti nel protocollo. Per implementare la possibilità di selezione delle sorgenti, nei messaggi di query e di report sono stati introdotti campi appositi: oltre alle general query e group specific query i router possono inviare *source specific query*, elencando le sorgenti del gruppo per le quali desiderano ricevere report. Analogamente, nei pacchetti di report i client possono inserire per ogni gruppo riportato una lista di sorgenti (*Group record*) da cui desiderano ricevere traffico per quel gruppo, o che desiderano al contrario filtrare dalla propria lista di sorgenti per quel gruppo; nel campo *Record type* di ciascun record si dichiara appunto quale delle due modalità si sta usando ("includi sorgenti" o "escludi sorgenti").

Questi cambiamenti nei pacchetti di report hanno reso inutili i messaggi di leave di IGMPv2: anche nel caso in cui un client desideri interrompere la ricezione di traffico multicast da tutte le sorgenti di un gruppo, basta che all'interno del group record riferito ad esso setti il campo *Record type* a "mode is include" e non inserisca alcuna sorgente nell'elenco di inclusione. Il meccanismo di report suppression non è più necessario: con i cambiamenti nei pacchetti appena descritti, ogni client multicast invia soltanto un report in risposta ad una query, contenente tanti record quanti sono i gruppi a cui intende associarsi o che intende lasciare. In questo modo, è sufficiente il meccanismo di timer prima di rispondere alle query per evitare l'effetto di packet storm.

## 3.3 Protocolli di Routing

Nei prossimi paragrafi seguirà una panoramica dei principali protocolli di routing multicast [13]. Descriverò prima l'algoritmo *Center Based Tree* (CBT) [14], classico protocollo ad albero condiviso, ed in seguito *Multicast-OSPF* [15] (MOSPF,di tipo link-state), *Distance Vector Multicast Routing Protocol* (DVMRP) [16] e *Protocol Indipendent Multicast – Dense Mode* (PIM-DM) [17], esempi di algoritmi "densi"; infine chiuderà una sezione su *Protocol Indipendent Multicast – Sparse Mode* (PIM-SM) [18] [19] [20], che è di fatto il più usato per l'inoltro di trame multicast all'interno di una rete.

### 3.3.1 CBT

L'idea fondamentale del Center Based Tree, come di tutti gli algoritmi core-based, è di creare all'interno della rete di un albero che colleghi con cammini minimi i router di last-hop dei ricevitori ad un core-router pre-determinato; questo sarà quindi la radice di un albero usato per la distribuzione del traffico multicast da tutte le sorgenti del gruppo ai client che ne fanno parte (shared tree). Il core-router viene scelto tra i nodi della rete tramite una procedura di elezione, oppure è impostato manualmente; in ogni caso, ciascun router in qualsiasi momento deve essere a conoscenza dell'indirizzo del core. Quando un client vuole segnalare che desidera ricevere traffico multicast, comunica il gruppo scelto al proprio router di riferimento tramite pacchetti IGMP; il router, ricevuto il messaggio, invia una *join request* al next-hop per il core designato. La join viene propagata hop per hop, finchè non raggiunge un nodo che faccia già parte dell'albero di distribuzione multicast oppure il core stesso. A questo punto, il nodo raggiunto propaga nello stesso modo un *join ack* in senso inverso; quando l'ack arriva al router di partenza, il suo "allacciamento" all'albero di inoltro è completato e può iniziare a ricevere il traffico multicast. Periodicamente le entry su ogni router per i gruppi ricevuti vengono riconfermate tramite messaggi di *echo request* ed *echo reply*; in caso

una entry venga lasciata scadere, il router interessato invia un messaggio di *quit notification* upstream (verso il core) ed uno di *flush tree* downstream (verso i nodi-foglia), in modo da notificare l'interruzione del ramo. I flussi multicast sono quindi inoltrati dai router di riferimento delle sorgenti verso il core, e da questo distribuiti ai router di last-hop dei client che ne hanno fatto richiesta. Si hanno perciò tanti cammini minimi dalle sorgenti al core-router quante sono le sorgenti, ma un solo shared tree dal core ai router di last hop, comune a tutti i flussi del gruppo di multicast. Questo permette di distribuire il traffico multicast di un gruppo ai client che ne fanno parte, senza che questi debbano occuparsi del problema di discovery delle relative sorgenti; d'altra parte, optare per un albero di distribuzione condiviso implica, rispetto alla scelta di cammini minimi da ciascuna sorgente al (o ai) last-hop router, un peggioramento in termine di ritardi tra le sorgenti ed i client. Questo è particolarmente sconveniente nel caso di diffusione di contenuti audio/video real time, che rappresentano una buona fetta del tipico traffico multicast. Risulta critico anche il problema di posizionamento del core, che a seconda dei casi potrebbe portare alla costruzione di alberi inefficienti. Per di più, il core è la radice di tutti gli alberi di multicast per ogni gruppo a cui è assegnato, e questo comporta una grande concentrazione di carico su di esso.

### 3.3.2 MOSPF

*Multicast Open Shortest Path First* è la variante multicast del protocollo di routing unicast OSPF. Come quest'ultimo, utilizza messaggi di *link state advertisement* (LSA), opportunamente modificati, per permettere ad ogni router di conoscere completamente la topologia della rete e di costruire l'albero di inoltro. Le informazioni contenute negli annunci di link state di OSPF sono integrate con quelle relative ai partecipanti ad ogni gruppo multicast, che i router ottengono tramite IGMP dai client a loro direttamente connessi; i messaggi che ne risultano, chiamati *group membership link state advertisement* (GM-LSA), vengono invia-

ti in flooding agli altri router della rete. In qualsiasi momento, quindi, ciascun router possiede il database topologico dell'intera area, comprese le posizioni dei partecipanti ad i gruppi di multicast. Quando riceve un pacchetto destinato ad un certo gruppo, se non ha già informazioni riguardo ad esso, ciascun router è in grado di calcolare dinamicamente l'albero di inoltro: conoscendo quali altri nodi desiderano ricevere i pacchetti per quel gruppo, calcola l'albero dei cammini minimi dalla sorgente a tutte le destinazioni con l'algoritmo di Dijkstra, potando i rami che non portano a sottoreti con partecipanti. Viene così costruita una tabella di *forwarding cache*, usata per inoltrare i pacchetti, che va aggiornata ad ogni cambiamento della topologia o della popolazione dei partecipanti ai gruppi. Avendo a disposizione l'intera topologia, ogni router può memorizzare il numero di hop necessari per raggiungere una certa destinazione, in modo da non inoltrare i pacchetti con *time to live* inferiore ad esso. MOSPF presenta principalmente due limiti: per prima cosa è un protocollo di routing di tipo denso, ossia richiede un consumo di risorse difficilmente accettabile in reti con un numero di partecipanti per gruppo basso rispetto al numero di host presenti. In particolare, è significativo l'overhead necessario a costruire e mantenere un albero per ogni sorgente. In secondo luogo, per poter usare questo protocollo è necessario che nella rete sia presente OSPF come algoritmo di routing unicast.

### 3.3.3 DVMRP

*Distance Vector Multicast Routing Protocol* è un protocollo di routing "denso", derivato da RIP, che si basa sul principio di flood and prune e che usa come metrica il numero di hop. Quando una sorgente inizia ad inviare traffico multicast, il suo router di riferimento inoltra i pacchetti su tutte le interfacce tranne che su quella da cui li riceve. A loro volta, gli altri nodi della rete ricevono e ritrasmettono il traffico su tutte le interfacce d'uscita disponibili. Gli unici limiti a questo meccanismo di flooding sono il check di *Reverse Path Forward* (RPF)

e il pruning. Il primo consiste nel fatto che, ogni volta che riceve un pacchetto multicast, ciascun router controlla che l'interfaccia da cui gli è arrivato sia quella verso cui inoltrerebbe il traffico indirizzato alla sorgente del pacchetto; in caso ciò non sia verificato, questo viene scartato. I messaggi di *prune* invece sono inviati dai router di last-hop verso l'interfaccia di arrivo dei pacchetti qualora non sia presente alcun client interessato al gruppo multicast ricevuto, e risalgono hop per hop l'albero di flooding finchè non arrivano ad un nodo con almeno un'interfaccia d'uscita non "pruned". I router che ricevono un messaggio di prune su un'interfaccia disabilitano l'inoltro dei pacchetti verso di essa per il gruppo indicato. Dunque, un pacchetto multicast che passa l'RPF check viene inoltrato verso tutte le interfacce, tranne quella di ingresso e quelle in stato "pruned" per il gruppo. I messaggi di prune vengono inviati periodicamente dai nodi non interessati al dato flusso multicast; se lo stato "pruned" di un'interfaccia non viene riconfermato da questi messaggi, allo scadere di un timer l'interfaccia riprenderà a far parte dell'albero di flooding. Qualora un router voglia riallacciarsi all'albero, non è necessario che attenda lo scadere del timer di prune del suo vicino: può inviare messaggi di *graft*, che forzano l'aggiornamento dello stato dell'interfaccia e vengono riscontrati con *graft ack*. DVMRP prevede inoltre l'utilizzo di tunnel IP (da impostare manualmente) nel caso in cui il traffico multicast debba attraversare router che non lo supportino, ed è usato ad esempio per mettere in comunicazione i nodi della *multicast backbone* con le varie "isole di multicast". L'utilizzo di questo algoritmo può essere giustificato in reti con poche sorgenti e molti ricevitori vicini tra loro, ma a causa del meccanismo di flooding periodico esso è decisamente poco scalabile.

### 3.3.4 PIM-DM

*Protocol Indipendent Multicast – Dense Mode* è molto simile al DVMRP; la differenza sta nel fatto che PIM utilizza l'algoritmo di routing unicast presente sulla

macchina per il calcolo delle metriche, qualunque esso sia, mentre DVMRP, come detto, usa un suo proprio algoritmo derivato da RIP con metrica in hop count. PIM-DM risente quindi degli stessi limiti di DVMRP, ma non è vincolato ad una metrica in particolare.

### 3.3.5 PIM-SM

*Protocol Indipendent Multicast – Sparse Mode* è la modalità "distribuita" di PIM: diversamente dai protocolli di routing multicast "densi", parte dal concetto che il traffico multicast debba essere inoltrato solo ai nodi che ne facciano esplicitamente richiesta; per realizzare ciò, esso prevede la coesistenza di shared tree condivisi tra più sorgenti (dello stesso tipo di quelli utilizzati nell'algoritmo CBT) con alberi ottimizzati per ogni sorgente (shortest path tree). Si distinguono quindi due fasi di instaurazione dell'albero:

#### Shared tree

In fase di setup della rete viene scelto, tramite un meccanismo di elezione oppure impostandolo manualmente, un router che faccia da *rendez-vous point* (RP), con funzioni simili a quelle del core del CBT. Quando una sorgente inizia ad inviare traffico multicast al proprio router di riferimento (*designated router*, DR), questo incapsula i pacchetti multicast in pacchetti IP unicast (*PIM-register*) destinati al rendez-vous point, che sarà così informato dell'esistenza di una nuova sorgente per il gruppo indicato. Se l'RP ha già ricevuto delle richieste di partecipazione a quel gruppo può inoltrare direttamente i pacchetti incapsulati verso i nodi interessati, in caso contrario risponde con un messaggio di *PIM-stop-register* al DR, che interrompe l'invio dei messaggi di registrazione della sorgente. La registrazione della sorgente viene comunque ripetuta periodicamente, in caso contrario dopo un certo tempo il rendez-vous point cancellerà l'informazione relativa ad essa dalla propria memoria. A lato client, i terminali inviano le proprie sottoscrizioni

per i gruppi multicast di cui desiderano fare parte ai rispettivi designated router (sempre tramite messaggi IGMP). I DR, quando ricevono una richiesta per un nuovo gruppo, inoltrano verso l'RP un messaggio di *PIM-join* per segnalare la propria partecipazione a quel gruppo; via via che tale messaggio si propaga hop per hop verso il rendez-vous, i nodi intermedi creano una entry nella propria multicast routing table che tenga traccia del gruppo e dell'interfaccia di ingresso e di uscita, costruendo così ramo per ramo l'albero di distribuzione per quel gruppo, con radice nell'RP (*shared tree*). Una volta che la join raggiunge il rendez-vous oppure un nodo che fa già parte dell'albero, il traffico multicast richiesto può iniziare a fluire verso la nuova foglia.

### Shortest path tree

Quando un designated router inizia a ricevere i pacchetti di multicast per il gruppo richiesto, può leggere dagli header gli indirizzi IP delle sorgenti attive per quel gruppo. Può quindi decidere di inviare richieste di PIM-join direttamente alle sorgenti, in modo da ricevere il flusso multicast attraverso degli *shortest path tree*, ciascuno con radice nella propria sorgente specifica. Il nuovo albero (o i nuovi alberi, a seconda del numero di sorgenti) viene costruito nello stesso modo dello shared tree, che, quando non è più necessario, viene abbandonato dal DR tramite l'invio di messaggi di *PIM-prune*.

La fase di shared tree è quindi inizialmente necessaria in quanto i router della rete non conoscono gli indirizzi delle sorgenti dei vari gruppi, ma soltanto quello del rendez-vous point. Affidarsi soltanto ad esso per la distribuzione dei contenuti vuol dire però ricadere nelle stesse problematiche presenti nell'algoritmo CBT; si rischia cioè di concentrare attorno all'RP un carico eccessivo, e di costruire alberi di distribuzione center-based poco efficienti, a seconda della topologia della rete. Da qui il passo in avanti costituito dalla possibilità di costruire degli shortest path tree per ogni sorgente, in modo da distribuire il traffico in rete in maniera

più omogenea e da avere cammini ottimizzati tra sorgente e client multicast. Il PIM-SM risulta perciò un algoritmo più scalabile dei precedenti, oltre ad avere, come il PIM-DM, il vantaggio di non essere legato ad un particolare algoritmo di routing unicast.

# Capitolo 4

# Soluzione Proposta

Nel capitolo precedente sono stati illustrati diversi algoritmi ed architetture di rete presenti in letteratura per la gestione del routing di traffico multicast. Alcune risultano essere più semplici anche se poco scalabili, altre garantiscono una maggior ottimizzazione delle risorse a scapito dei requisiti di qualità del servizio.

In questo lavoro di tesi si è proposto un sistema di routing multicast volto principalmente a migliorare la gestione della mobilità del client, che nei protocolli illustrati di fatto non è stata presa in considerazione. Più precisamente, si è scelto di partire dal PIM-SM, per via della sua maggior scalabilità rispetto agli altri protocolli esistenti e del fatto che non richiede la presenza di un protocollo di routing unicast in particolare, e di introdurre delle aggiunte atte a garantire una gestione quanto più veloce possibile del processo di handover.

Nella prossima sezione verrà descritta l'architettura di rete in cui opera la soluzione proposta; successivamente verrà esposto il protocollo stesso e saranno poi esaminati i messaggi scambiati dai router della rete a seconda degli eventi che si possono verificare.

## 4.1 Architettura

Con riferimento all'architettura di rete definita nella sezione 2.3.1, si prenderanno ora in esame le entità che interagiscono per il corretto funzionamento della soluzione proposta; un esempio di realizzazione di tale architettura è mostrato in figura 4.1.

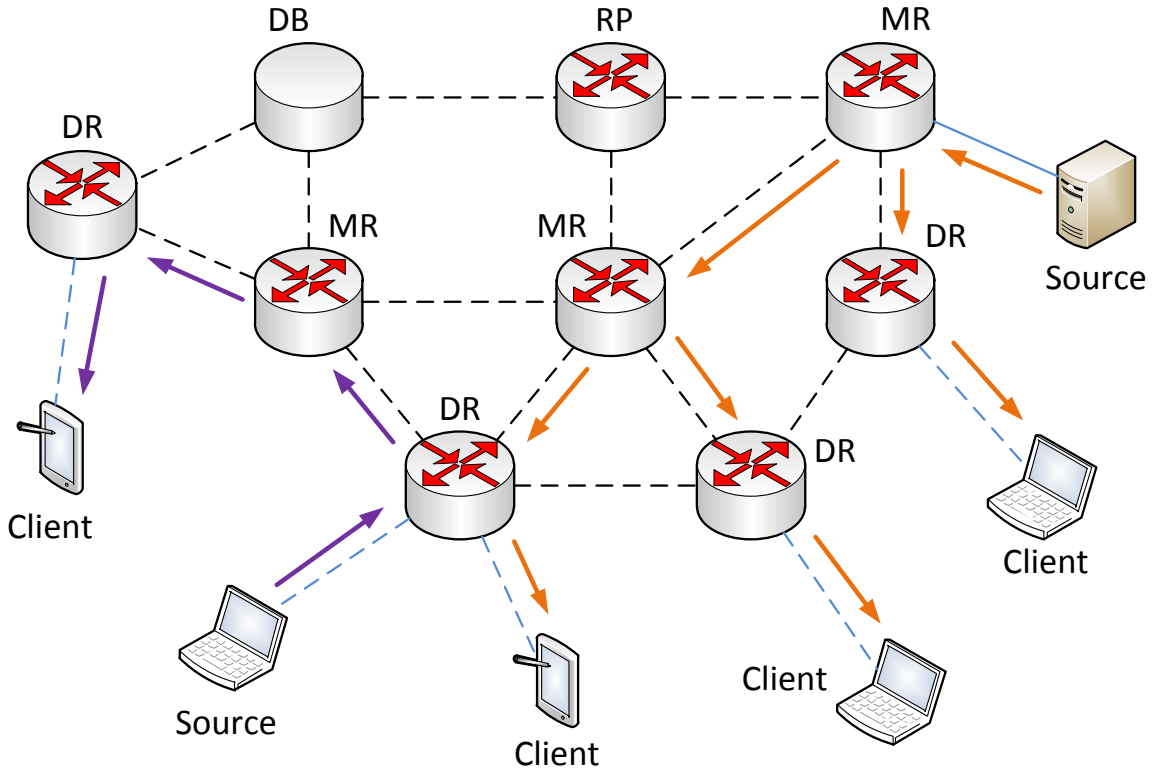



Figura 4.1: Scenario di una rete multicast con l' architettura proposta

### 4.1.1 Client multicast

Un client di multicast è un qualsiasi mobile host (descritti nella sezione 2.3.1.1) che abbia abbia sottoscritto almeno una richiesta di partecipazione ad un gruppo; perchè un client possa avvalersi dei servizi di multicast forniti dalla soluzione

proposta, è necessario che il kernel del sistema operativo supporti il protocollo IGMP in versione 3. Sebbene questo standard non sia ancora usato spesso (i demoni di routing usano tipicamente query di IGMPv2), è implementato su quasi tutti i sistemi operativi attualmente in circolazione, quindi il requisito aggiuntivo che si richiede non risulta particolarmente stringente.

### 4.1.2 Sorgenti multicast

Una sorgente di traffico multicast è un qualsiasi terminale, connesso alla rete, che origina un flusso di multicast. Una sorgente, quindi, genera pacchetti IP che hanno il suo indirizzo nel campo *Source* dell'header e l'indirizzo di un gruppo di multicast nel campo *Destination*; tipicamente una sorgente è costituita da un server che distribuisce contenuti audio/video in streaming, o un dispositivo partecipante ad una videoconferenza. Anche un mobile host può essere una sorgente di multicast, e a differenza del caso di un multicast client non è necessario che implementi il protocollo IGMP: sono sufficienti le sole funzionalità di rete di base.

### 4.1.3 Multicast router

Tutti i router della rete che supportano il traffico multicast sono *multicast router* (MR); un multicast router è quindi in grado di ricevere, elaborare ed inoltrare pacchetti di multicast. Su tutti gli MR è presente il demone di routing implementato in questo lavoro di tesi, che si occupa di inviare le interrogazioni IGMP agli eventuali client, di elaborare i relativi report, ed in generale di scambiare con gli altri componenti della rete i vari messaggi necessari al funzionamento della rete di multicast e della gestione della mobilità dei client. L'insieme degli MR comprende anche dispositivi con funzioni aggiuntive, quali sono i *designated router* ed i *rendez-vous point*.

#### 4.1.3.1 Designated router

Rientrano nella categoria dei designated router (DR), o last-hop router, tutti i multicast router che sono direttamente connessi ad almeno un client multicast; data la natura particolarmente dinamica di una rete wireless, un multicast router può facilmente acquisire lo status di designated router e viceversa. Il primo caso si verifica quando un MR riceve una richiesta di partecipazione ad un gruppo da un host a lui associato, oppure quando un multicast client effettua un handover associandosi all' MR, che diventerà quindi designated router. La situazione contraria avviene quando un DR perde tutti i suoi client di multicast (perchè si spostano o perchè annullano le sottoscrizioni).

I designated router, oltre ad occuparsi di fornire accesso al traffico multicast ai client, tengono in memoria una tabella che associa l'IP dei propri client a quelli dei rispettivi gruppi multicast.

#### 4.1.3.2 Rendez-vous point

Il rendez-vous point (RP) è un multicast router con funzionalità di discovery delle sorgenti. Esso si occupa di registrare in memoria locale gli IP delle sorgenti di traffico multicast presenti nella rete e quelli dei relativi gruppi, e di inviare le medesime informazioni ad un nodo di *database* centralizzato. Nell'architettura di rete della soluzione proposta deve esistere almeno un rendez-vous point; in fase di configurazione si provvede a definire per ogni RP un sottoinsieme di indirizzi della classe D a cui esso è assegnato.

### 4.1.4 Database centrale

Un determinato nodo di database tiene traccia delle informazioni sui mappaggi gruppo-sorgenti raccolte dal (o dai) rendez-vous point, e di quelli client-gruppi raccolti da tutti i designated router. Il database centrale dispone quindi di due tabelle in cui saranno annotate tutte le associazioni di questo tipo, e il suo compito

è quello restare a disposizione delle interrogazioni dei router, distribuendo quando è opportuno le informazioni che ha memorizzato.

## 4.2 Descrizione del protocollo

Come è stato anticipato, la soluzione proposta prende spunto dal protocollo di PIM-SM e ne modifica alcuni aspetti per tenere conto del problema di gestione della mobilità dei client.

Nella prossima sezione sarà descritta la procedura per mappare a lato rete tutte le associazioni client-gruppo, passaggio necessario per supportare l'handover dei client di multicast; nei paragrafi seguenti saranno esposte le interazioni tra i vari elementi dell'architettura necessarie al supporto per il multicast e per la mobilità dei client.

### 4.2.1 Creazione e gestione del database

Per un meccanismo di gestione della mobilità di un client è necessario che si abbiano a disposizione in ogni momento le informazioni riguardanti i gruppi di cui questo faccia parte. Si è deciso quindi di mantenere ed aggiornare un database centralizzato, situato su un nodo della backbone, che tenga traccia dei gruppi di cui fa parte ciascun client della rete d'accesso.

Quando un designated router riceve un messaggio di IGMP-report che comunica l'ingresso o l'uscita di un client in un gruppo, aggiorna nella propria memoria la lista di gruppi di cui fa parte il client e, se a seguito dell'aggiornamento questa subisce dei cambiamenti, li comunica al nodo di database (con messaggi di DR-Update, descritti nel dettaglio nella sezione 5.2.2.2) che a sua volta aggiorna la propria tabella di associazioni client - gruppi (si veda la figura 4.2).

Similmente, quando l' MR di una sorgente invia all' RP pacchetti di register contenenti un nuovo flusso di dati multicast, l'RP aggiorna la propria tabella

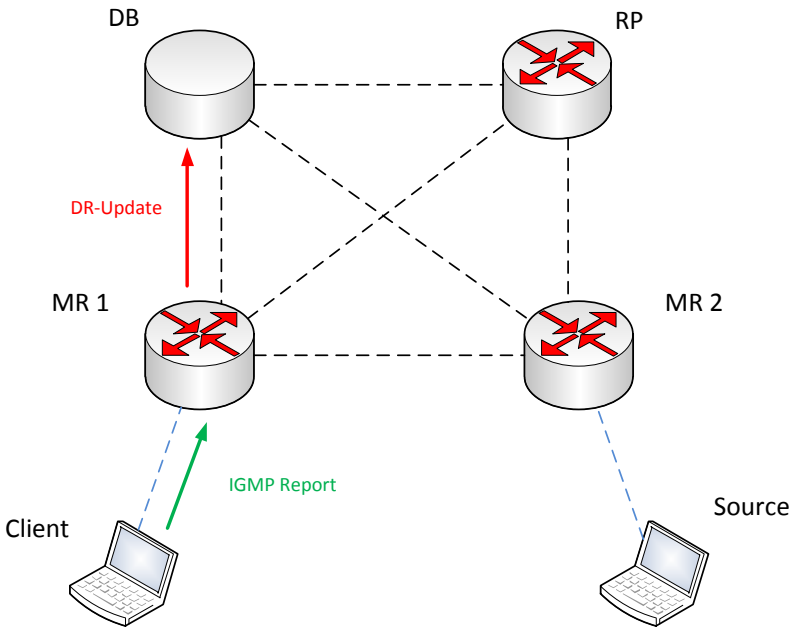



Figura 4.2: Aggiornamento delle tabelle per un nuovo client

locale che indica quali sono le sorgenti presenti per ogni gruppo; come nel caso dei DR, ad ogni modifica della tabella invia messaggi di segnalazione (RP-Update, sezione 5.2.2.3) al database perchè anch'esso tenga traccia dei cambiamenti (figura 4.3).

Ogni elemento del database è associato ad un timer, allo scadere del quale viene cancellato. Per prolungarne la validità, vengono inviati periodicamente dai designated router e dal rendez-vous point dei messaggi che contengono l'intero contenuto delle tabelle locali, che viene scandito e confrontato con quello del database, affinchè sia garantita la coerenza dei dati registrati.

### 4.2.2 Procedura di join

Se un terminale mobile desidera entrare a far parte di un gruppo multicast, segnala la propria richiesta al suo designated router tramite un messaggio di IGMP-report (senza dover attendere query); come detto, il DR aggiorna il proprio

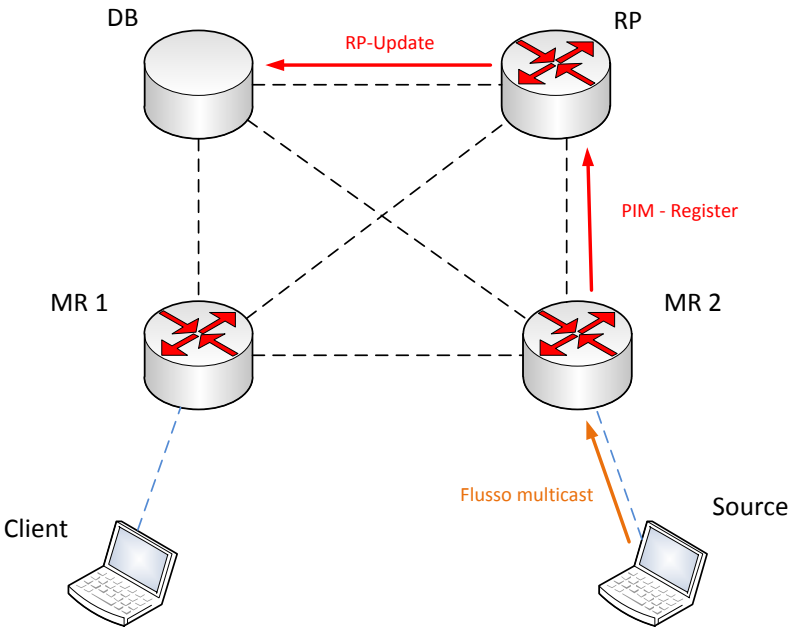
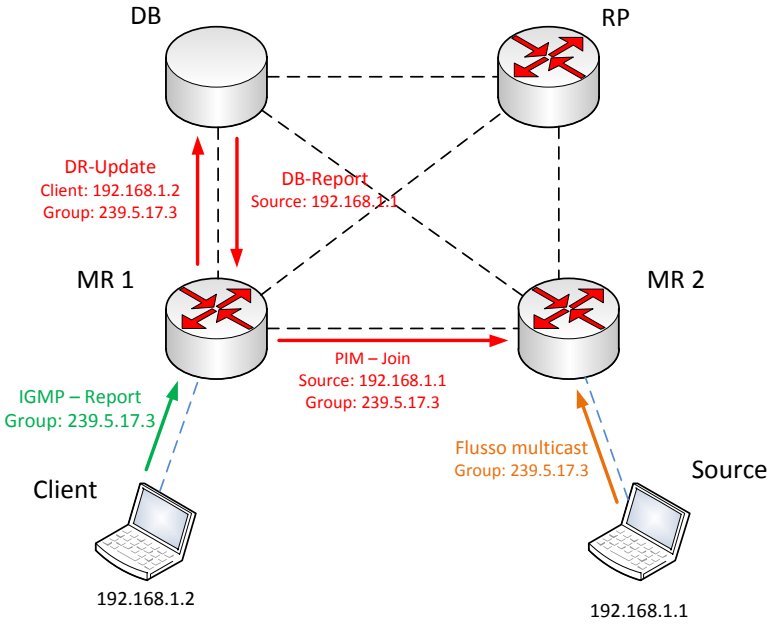


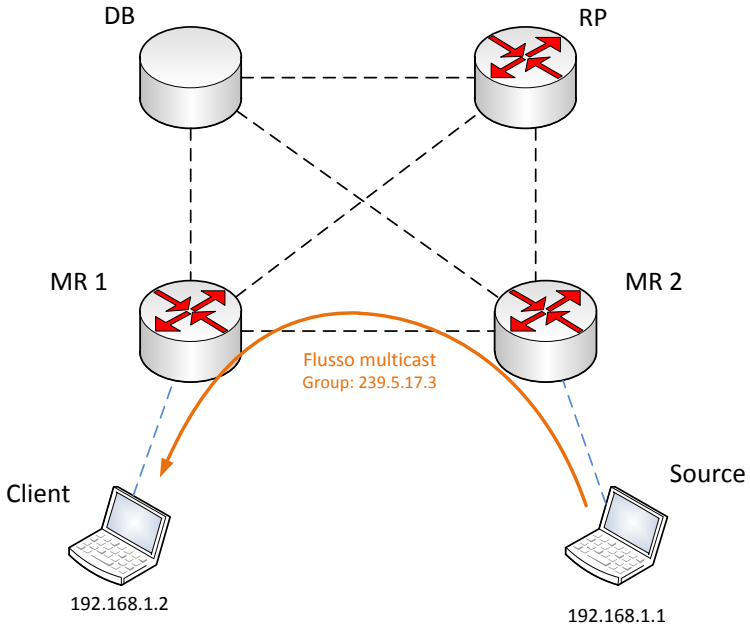

Figura 4.3: Aggiornamento delle tabelle per una nuova sorgente

database locale con l'IP del nuovo client e quello del gruppo di cui questo vuole entrare a far parte. Contemporaneamente, comunica le stesse informazioni al database centrale ed attende risposta da quest'ultimo. Il database informa di rimando il DR sulle eventuali sorgenti esistenti per il gruppo richiesto, con un messaggio contenente gli IP delle varie sorgenti attive.

In questo modo, il designated router può inviare pacchetti di PIM-join destinati direttamente ad esse, senza doversi affidare al rendez-vous point per l'individuazione delle sorgenti ed evitando la fase iniziale di costruzione dello shared tree prevista dal PIM-SM originale. Dopo lo scambio di messaggi di notifica/interrogazione e risposta tra DR e database, viene quindi instaurato subito lo shortest path tree da ciascuna sorgente al nuovo membro del gruppo, che può iniziare a ricevere pacchetti dai vari cammini minimi creati. Un esempio di procedura di join di un gruppo è mostrato in figura 4.4.




(a)




(b)

Figura 4.4: Procedura di join

### 4.2.3 Procedura di handover

In caso di handover di un mobile host della rete mesh, il demone di gestione della mobilità provvede a segnalare al nuovo router di riferimento (MR o DR) del terminale la nuova associazione; il router però non dispone sulla sua tabella locale di informazioni riguardo ad eventuali partecipazioni a gruppi di multicast del nuovo host. Il MR quindi, ogni volta che riceve una notifica di nuova associazione, invia una interrogazione al database centrale per sapere di quali gruppi (se ce ne sono) l'host fa parte.

Il database risponde fornendo l'elenco degli indirizzi IP dei gruppi e delle relative sorgenti; il router può così inserire le informazioni ricevute nella propria memoria locale ed inviare alle sorgenti i messaggi di PIM-join necessari, nel caso in cui non stesse già ricevendo i corrispondenti flussi di multicast (figure 4.5 e 4.6).

Le join si propagheranno verso le sorgenti fino a raggiungere un nodo che già ne riceve il traffico multicast (o fino alla sorgente stessa), riallacciando così il client agli shortest path tree a cui era precedentemente collegato (figura 4.7).

### 4.2.4 Inoltro di traffico da una nuova sorgente

Quando si attiva una nuova sorgente di un gruppo, come previsto dalle specifiche del PIM-SM, il suo MR di riferimento inizialmente incapsula i pacchetti di multicast in datagrammi IP unicast e li indirizza al rendez-vous point; questo, ricevutili, estrae i pacchetti di multicast, legge l'indirizzo IP della sorgente e del gruppo a cui sono destinati e li aggiunge alla propria tabella locale delle sorgenti. Contestualmente, come mostrato in figura 4.8, invia al database le stesse informazioni perchè questo aggiorni la tabella generale delle sorgenti (che raccoglie i dati dei rendez-vous point relativi ai vari gruppi). In questo modo, quando riceve da un designated router un messaggio che notifichi l'ingresso di un client in un gruppo multicast (o la richiesta inviata da un DR a seguito dell' handover di un

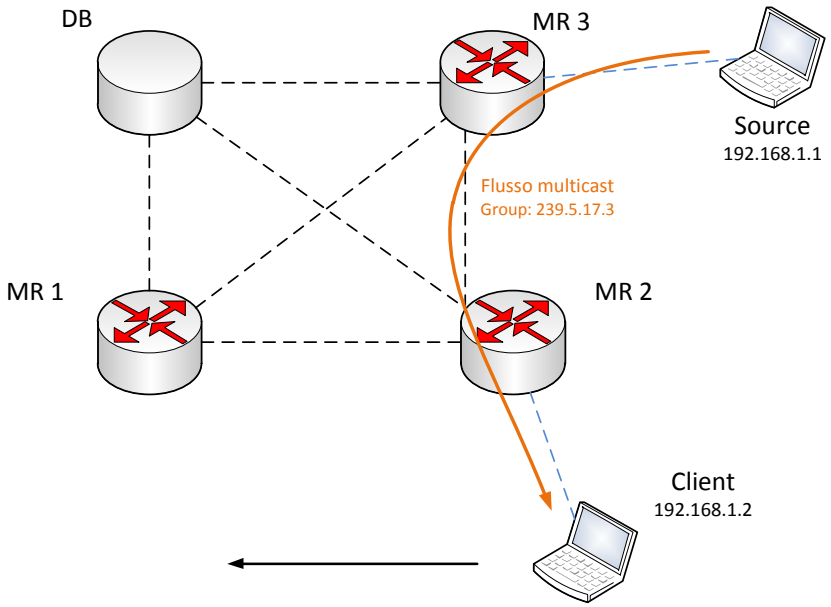



Figura 4.5: Handover - situazione di partenza

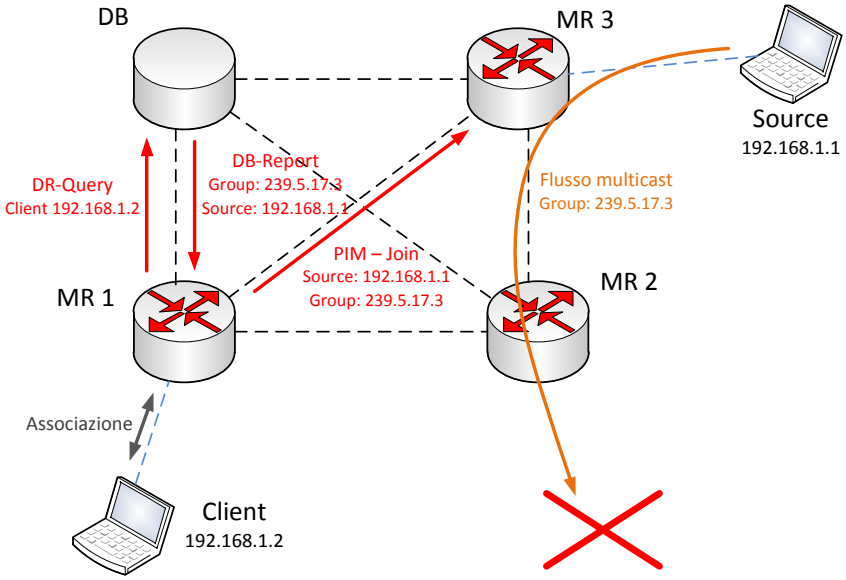



Figura 4.6: Handover - segnalazione nella fase transitoria

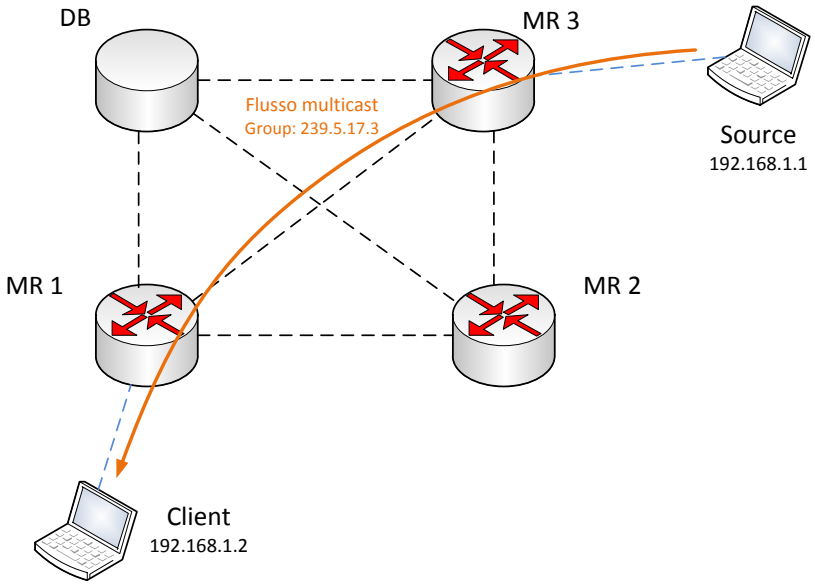



Figura 4.7: Handover - riallacciamento ad SPT

client), il database può fornire una lista completa delle sorgenti attive per quel gruppo.

Il database, inoltre, quando riceve da un RP la notifica di una nuova sorgente per un dato gruppo, controlla nella propria tabella dei client se ce n'è qualcuno che sta già partecipando al gruppo stesso. In caso affermativo, invia al DR del client un pacchetto per avvisarlo che si è attivata una nuova sorgente per un gruppo a cui è interessato, indicando ovviamente gli IP della coppia sorgente-gruppo. Il designated router può così inviare un messaggio di PIM-join verso la nuova sorgente per allacciarsi all'albero (SPT) relativo ed inoltrare ai client interessati il nuovo flusso (figure 4.9 e 4.10).

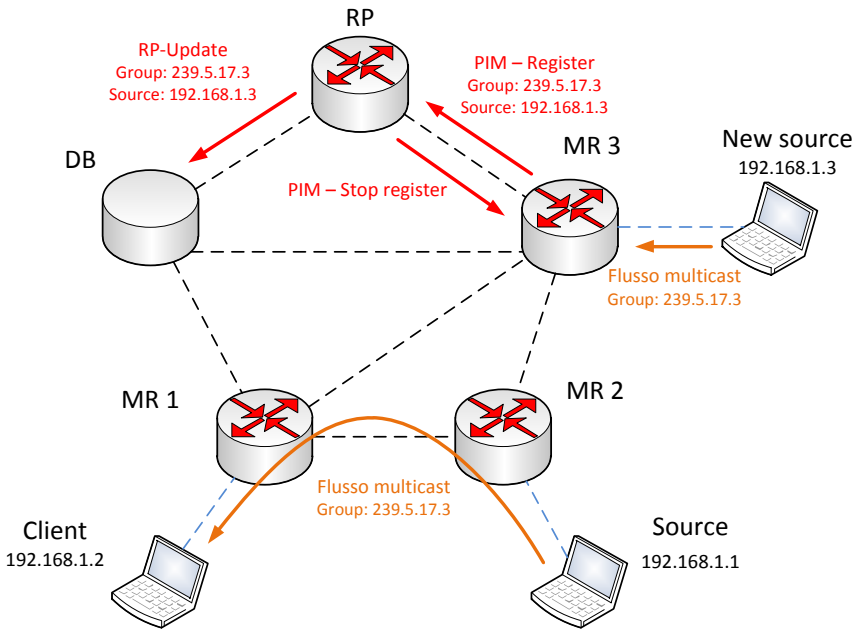



Figura 4.8: Segnalazione tra il router di una nuova sorgente e il DB

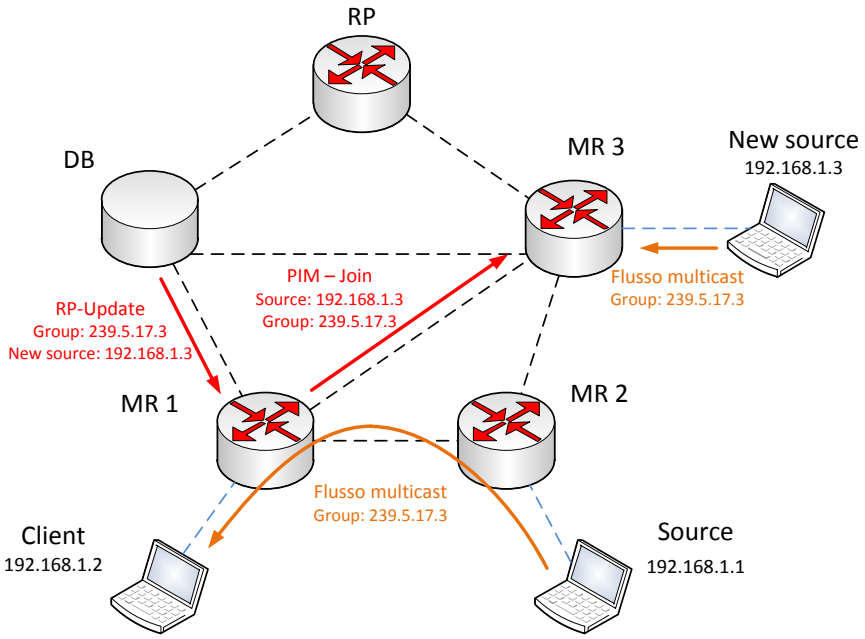



Figura 4.9: Segnalazione tra DB e DR e join verso la nuova sorgente

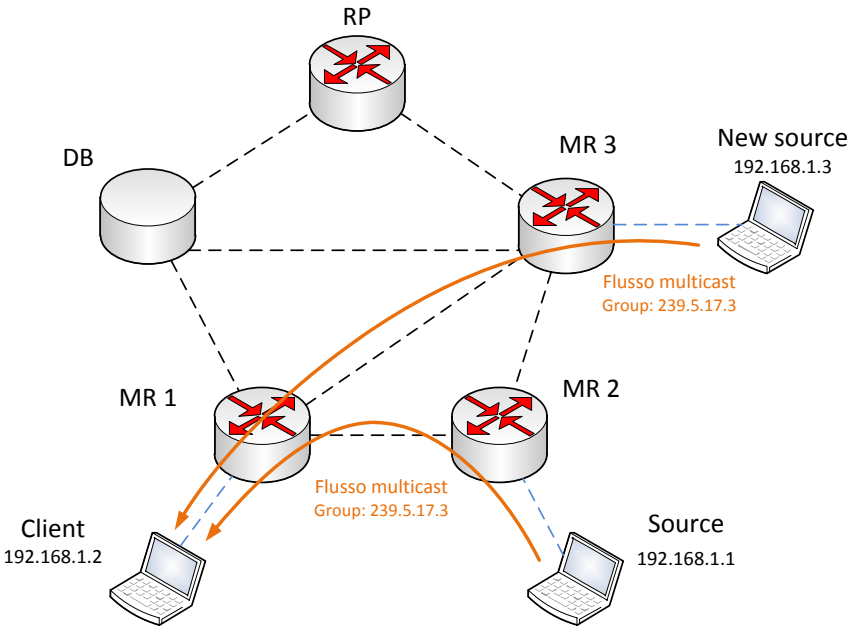



Figura 4.10: Ricongiungimento con SPT della nuova sorgente

# Capitolo 5

# Implementazione

La soluzione proposta, come detto, ricalca il Protocol Indipendent Multicast - Sparse Mode cambiandone diversi aspetti per tenere conto della possibilità dei client multicast di spostarsi all'interno della rete di accesso. Per l'implementazione della stessa, quindi, si è partiti dal demone di routing multicast PIMD ([21]), *open source* ed operante su Linux, che a lato accesso prevede lo scambio di messaggi IGMPv2 tra designated router e client per le sottoscrizioni ai gruppi, ed a lato rete implementa lo standard di PIMv2 in modalità *sparse*. Da questa base, sono state apportate le modifiche al protocollo esposte nel capitolo precedente.

Nelle prossime sezioni saranno inizialmente esposti i componenti dell'architettura software necessari al funzionamento del protocollo stesso, seguirà poi una descrizione delle modalità e dei messaggi con cui i componenti medesimi comunicano tra loro, ed infine verranno illustrati i meccanismi di mantenimento delle informazioni che servono ai processi per operare correttamente.

## 5.1 Architettura software

L' architettura software è composta da tre processi. Il processo Hostapd [22] amministra la sezione di accesso dei router wireless gestendone le interfacce ope-

ranti in modalità infrastrutturata. PIMD è il demone di routing multicast vero e proprio che costituisce il cuore dell'architettura ed agisce su tutti i router di multicast. Vi è poi una singola istanza di database server, che mantiene e coordina le informazioni presenti sui singoli nodi della rete. Saranno di seguito illustrati i compiti specifici di ognuno di essi.

### 5.1.1 Hostapd

Hostapd è un processo che non è stato sviluppato in questo lavoro di tesi e che svolge funzionalità di access point 802.11. Esso è presente su ogni mobile router della rete MobiMESH che si interfacci con la sottorete di accesso, ed al suo interno sono incluse numerose funzionalità di amministrazione come le autenticazioni IEEE 802.1X ([23]), Wi-Fi Protected Access (WPA, WPA2), Extensible Autentication Protocol (EAP) ed un server Remote Access Dial-In User Service (RADIUS).

Per il corretto funzionamento del protocollo esposto in questo elaborato, è necessario che ogni volta che su una interfaccia wireless 802.11 di un Multicast Router con funzionalità di accesso viene rilevata una nuova associazione, questa venga notificata al demone di routing multicast (PIMD) che si occupa di gestire la mobilità dei client. Questa funzione è svolta appunto da Hostapd, che invia la segnalazione di associazione/disassociazione di un Mobile Host tramite un socket UNIX sul quale PIMD rimane in ascolto (il formato di questi messaggi è discusso nella sezione 5.2.1.2).

Non si è comunque strettamente vincolati all'utilizzo di Hostapd, l'unico requisito necessario per garantire il supporto agli handover dei client di gruppi multicast è di fare in modo che vi sia un qualche processo che monitori le interfacce di accesso alla rete e generi un pacchetto di notifica di associazione o disassociazione, inviandolo sull'opportuno socket UNIX su cui PIMD è in ascolto.

### 5.1.2 PIMD modificato

Per l'implementazione del protocollo di routing multicast vero e proprio, e delle funzionalità necessarie sugli MR per il suo funzionamento, si è utilizzato come base il demone PIMD, sviluppato sotto licenza open source per Linux e BSD e liberamente scaricabile. Esso segue le specifiche della versione originale del protocollo PIM-SM v2, in accordo con lo standard descritto nell' RFC 2362. Partendo dal codice del demone originale, sono state effettuate le aggiunte e le modifiche previste dalla variante del protocollo esposta nel capitolo precedente, principalmente atte a tener conto della natura mobile dei client, che nella versione standard del protocollo (e della sua implementazione) non viene di fatto considerata. Il demone di routing ottenuto è costituito da una singola applicazione scritta in linguaggio di programmazione C per sistemi Linux compatibili, in particolare ne è stato verificato il funzionamento sotto sistemi operativi OpenWrt ed Ubuntu. Le impostazioni di configurazione di tale applicazione possono essere controllate e modificate tramite un opportuno file dedicato. Il processo PIMD, come detto, è attivo su tutti i multicast router della rete, e svolge le principali operazioni necessarie al supporto al multicast ed alla mobilità previste dalla soluzione proposta in questo lavoro.

A lato rete di backbone, esso si occupa di mantenere aggiornate le informazioni relative al vicinato (gli altri MR a cui è direttamente connesso), all' identità del router di bootstrap, dei rendez-vous point e del range di gruppi a cui sono associati; si occupa inoltre di inviare la segnalazione necessaria (PIM-join e prune) per sottoscrivere l'ingresso in un gruppo multicast o comunicare il proprio abbandono dello stesso, e di scambiare i messaggi con il server che contiene il database centrale dei client, dei gruppi e delle sorgenti e che coordina la gestione della mobilità.

A lato rete di accesso, il demone gestisce lo scambio di messaggi IGMP con i client direttamente connessi al nodo su cui è attivo e controlla ed aggiorna la

tabella locale con le informazioni sui gruppi di cui i client fanno parte; resta inoltre in ascolto su un socket UNIX per ricevere la segnalazione locale inviata da Hostapd, come anticipato nel paragrafo precedente.

Poichè lo standard (esposto nel capitolo 3.3.5) prevede la presenza di un qualsiasi protocollo di routing unicast sottostante su cui PIM si appoggi per ottenere le rotte di instradamento, PIMD si occupa anche di interrogare la routing table unicast ogni volta che è necessario.

Sulle macchine configurate per agire come rendez-vous point, PIMD svolge infine la funzione di discovery delle sorgenti attive, ricevendo i pacchetti di register dai rispettivi designated router, e segnalandone gli indirizzi al database server perchè questo mantenga aggiornate le proprie informazioni di mappaggio dei gruppi con le sorgenti.

### 5.1.3 Database server

Il server centrale risiede su un solo nodo dedicato. E' stato implementato, come gli altri componenti, in linguaggio di programmazione C per piattaforme Linux.

Il suo compito è, come già illustrato nel capitolo precedente, quello di raccogliere e mantenere consistenti le informazioni provenienti da tutti i designated router/rendez-vous point sui client dei gruppi di multicast e sulle relative sorgenti, restando in attesa di interrogazioni da parte dei router; il database, per fare ciò, dispone di due tabelle (descritte nella sezione 5.3.3), che tiene aggiornate e consulta ad ogni richiesta che riceve.

Il database tiene inoltre traccia di quali designated router stiano ricevendo traffico multicast per un dato gruppo; in questo modo può svolgere un suo secondo compito, ovvero quello di notificare a tali router l'eventuale presenza in rete di una nuova sorgente per il gruppo stesso, così che questi possano immediatamente inviare le richieste di join verso di essa.

## 5.2 Scambio di informazioni

Affinchè i componenti dell' architettura della rete mantengano consistenti le informazioni necessarie al corretto supporto del traffico multicast, è necessario che comunichino costantemente tra loro, sia per notificare il verificarsi di nuovi eventi o di cambiamenti nella topologia, sia per rinfrescare le informazioni già presenti nei terminali (che se non vengono periodicamente riconfermate, allo scadere di un timer sono considerate obsolete e cancellate). I messaggi che vengono scambiati tra i diversi componenti usano differenti protocolli, e a seconda dei casi seguono degli standard già maturi e definiti negli RFC di riferimento oppure sono stati pensati specificamente per questo lavoro di tesi. In particolare, i pacchetti di segnalazione tra client e router della rete d' accesso e quelli tra i router della rete di backbone sono definiti dai rispettivi standard (IGMP e PIM-SM) presenti da tempo e discussi nei relativi RFC, mentre i messaggi che vengono scambiati tra i router ed il database server sono stati definiti per gli scopi specifici descritti nel capitolo 4 e nelle sezioni 5.2.2.2 e 5.2.2.3.

Nelle prossime sezioni verranno descritte le modalità e i protocolli usati per lo scambio di informazioni nella rete di accesso e in quella di backbone, sottolineando quando si è scelto di non implementare completamente gli standard esistenti e spiegandone i motivi.

### 5.2.1 Segnalazione nella rete di accesso

I messaggi di segnalazione presenti nel lato accesso della rete sono di due tipi: i pacchetti IGMP, come esposto nella sezione 3.2, vengono usati dai client per notificare l' interesse a partecipare a dei dati gruppi di multicast, e dai router per sollecitare l'invio di tali notifiche da parte dei client; i messaggi inviati da Hostapd, invece, non viaggiano fisicamente in rete, ma sono usati da questo processo (presente su ogni router multicast che svolga funzioni di access point) per interagire con il processo PIMD che gira sulla medesima macchina. Nelle sezioni

5.2.1.1 e 5.2.1.2 verranno descritti nel dettaglio i messaggi e gli accorgimenti usati nell' implementazione degli stessi.

#### 5.2.1.1 Messaggi IGMPv3

Il demone PIMD nella sua versione di partenza usa messaggi di segnalazione IGMPv2 per comunicare con i client e richiedere le notifiche di ingresso o uscita da un gruppo. Il protocollo IGMPv2, come descritto nel capitolo 3.2.1, prevede un meccanismo di report suppression per evitare che vengano generati troppi pacchetti di report in risposta ad una query; per l'implementazione del protocollo esposto in questo lavoro di tesi, però, è necessario che tutti i client che fanno parte di un certo gruppo rispondano alle interrogazioni che lo riguardano, in modo che il router che le ha generate possa tenere aggiornata la propria tabella locale delle associazioni di ogni client con i gruppi di cui intende fare parte. Per questo motivo si è deciso di implementare a lato accesso la versione 3 del protocollo IGMP: grazie all'utilizzo di record aggregati, il numero di pacchetti in risposta alle query diminuisce significativamente rispetto al caso precedente, ed il meccanismo di report suppression non è più necessario (viene scongiurato il pericolo di packet storming).

I report dello standard IGMPv3, di dimensioni variabili e molto maggiori rispetto a quelli, di lunghezza fissa (8 byte), usati in IGMPv2, contengono una catena di record con alcuni campi destinati alla trasmissione di informazioni necessarie al funzionamento di una variante del protocollo PIM-SM, il *Protocol Indipendent Multicast - Source Specific Multicast* (PIM-SSM, [24]). In particolare, con IGMPv3 ciascun client può segnalare al router non solo il proprio interesse per un dato gruppo multicast, ma anche specificare di quali sorgenti per quel gruppo vuole ricevere il flusso e quali invece desidera ignorare. Si è deciso, per semplicità e perchè PIM-SSM non specifica in che modo i client debbano poter venire a conoscenza degli indirizzi delle sorgenti attive, di non implementare questa

variante del protocollo PIM; questo implica che, nel caso in cui un client segnali con IGMPv3 di non voler ricevere traffico da alcune sorgenti di un gruppo di cui fa parte, la sua richiesta sia ignorata da PIMD e gli vengano comunque inoltrati i flussi generati da tutte le sorgenti per quel gruppo. E' stato cioè necessario ricondursi al caso di semplici messaggi di Report e di Leave della versione 2 del protocollo IGMP, pur utilizzando in realtà quella successiva. I record ricevuti, che si riferiscono alle sorgenti, vengono perciò scanditi e tradotti in semplici join o leave riferite ad un un gruppo, a seconda dei casi:

- i record di tipo include con la lista di sorgenti vuota vengono considerati come dei messaggi di leave per il gruppo indicato;
- i record di tipo include con la lista di sorgenti contenente almeno un indirizzo vengono considerati delle join per il gruppo;
- i record di tipo exclude vengono in ogni caso considerati delle join per il gruppo riportato.

#### 5.2.1.2 Notifiche su socket locale (Hostapd)

Sui nodi di rete che svolgono funzioni di access point è presente il processo Hostapd che si occupa di rilevare le nuove associazioni dei *mobile host*. Tali eventi vengono notificati al processo PIMD tramite un socket dedicato, come detto precedentemente nella sezione 5.1.1. La struttura dei pacchetti usati da Hostapd per comunicare l'avvenuta associazione (o disassociazione) è illustrata in figura 5.1.

### 5.2.2 Segnalazione nella rete di backbone

Nel lato backbone della rete si possono classificare i messaggi usati in tre gruppi:

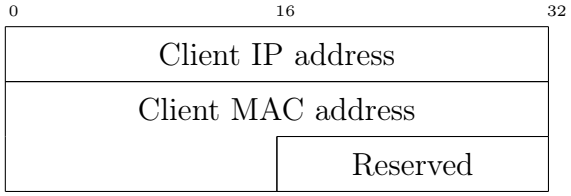



Figura 5.1: Messaggio di notifica di host discover

- i messaggi trasportati dai pacchetti di segnalazione definiti nell' RFC per il PIM-SM, usati dai nodi della rete per il mantenimento del vicinato, per eleggere i rendez-vous point e per la costruzione e la potatura degli alberi alberi di inoltro;

- i messaggi scambiati tra i designated router e il database server, usati per l'aggiornamento delle informazioni memorizzate nei componenti, quali l' associazione di un nuovo client ad un AP, l' handover di un client già presente in rete, la notifica di una nuova sorgente per un gruppo multicast che un dato AP desidera ricevere;

- i messaggi scambiati tra i rendez-vous point e il database server, usati per mantenere consistenti le informazioni necessarie al database server sulle sorgenti dei vari gruppi presenti in rete.

Gli ultimi due gruppi di messaggi non sono definiti in alcuno standard, e sono stati implementati in modo specifico per questo lavoro di tesi; essi vengono trasportati nel campo dati di pacchetti TCP, in modo da garantire che giungano a destinazione in maniera corretta.

Le strutture dei pacchetti definiti per questi messaggi contengono alcuni campi sovradimensionati in maniera evidente; questo perchè, in ogni caso, i bit che si sarebbero eventualmente risparmiati definendo campi più piccoli sarebbero comunque andati sprecati come bit di padding, a causa dei requisiti di allineamento dei processori ARM presenti sui nodi della rete MobiMESH.

I primi 32 bit di tali pacchetti sono sempre occupati da un *header* che contiene i campi di *Protocol ID* e *Packet code*, usati dai router che li ricevono per verificare che il messaggio faccia parte di questo protocollo specifico (Protocol ID) e per distinguere i vari tipi di pacchetto che sono stati definiti per tale protocollo (Packet code). Di seguito verranno esposte nel dettaglio le tipologie di messaggio usate nei tre gruppi appena descritti.

#### 5.2.2.1 Segnalazione PIM-SM

I messaggi di segnalazione previsti dallo standard del PIM-SM sono contenuti nel payload di datagrammi IP aventi il campo *protocol* dell' header impostato con valore 103. La struttura di un generico pacchetto PIM-SM comprende i campi:

- *Version*, che indica la versione del protocollo in uso (in questo caso è implementata la versione 2);
- *Type*, che indica il tipo di messaggio trasportato. I diversi valori che può assumere tale campo in questa implementazione saranno descritti di seguito in questa sezione, insieme con la descrizione dei messaggi specifici.
- *Reserved*, impostato sempre a 0;
- *Checksum*, calcolato come il complemento a uno della somma in complemento a uno dell' intero messaggio;
- *Data*, di lunghezza variabile a seconda del tipo di messaggio in uso.

Le tipologie di messaggio (contenuto nel campo Data del pacchetto) previste dal protocollo PIM-SM che sono usate in questo lavoro di tesi sono 6:

- *Hello* (*type* 0): messaggi scambiati periodicamente da tutti i router di multicast per ottenere informazioni sul vicinato;

- *Register* (*type* 1): messaggi inviati dal DR di una sorgente all' opportuno RP, con incapsulati i pacchetti di multicast, come descritto nella sezione 3.3.5;

- *Register-stop* (*type* 2): messaggi inviati dall' RP in risposta ai pacchetti di *register*, per confermare l' avvenuta registrazione della sorgente e fermare l' incapsulamento dei pacchetti multicast;

- *Join/Prune* (*type* 3): messaggi inviati dai DR verso i rispettivi MR di next-hop per un dato flusso di multicast, con lo scopo di allacciarsi (nel caso di una *join*) o di staccarsi (inviando una *prune*) dall'albero di multicast relativo quel flusso, come esposto nei capitoli 3.3.5 e 4.2;

- *Bootstrap* (*type* 4): messaggi usati dall' MR designato a svolgere le funzioni di bootstrap, ossia a comunicare agli altri MR le informazioni relative ai rendez-vous point attivi (indirizzi IP degli RP e relativi range di indirizzi multicast di competenza);

- *Candidate-RP-advertisement* (*type* 8): messaggi usati dagli MR per candidarsi a svolgere le funzioni di RP; contengono l' indirizzo IP del mittente, il range di indirizzi di classe D che è abilitato a servire e la priorità che gli è stata assegnata in fase di setting. Tali messaggi vengono poi elaborati dal router di bootstrap, che si occupa di eleggere un rendez-vous point per ogni sottogruppo di indirizzi multicast e di comunicare tali informazioni agli altri MR.

#### 5.2.2.2 Segnalazione DR e DB

In questa sezione saranno descritti i messaggi scambiati tra i designated router (DR) ed il server centrale che opera da database (DB). Come anticipato all' inizio del capitolo (sezione 5.2.2), questi messaggi sono contenuti in pacchetti TCP, e i

primi 32 bit sono riservati ai campi, comuni a tutti, di *protocol ID* e *packet code*, di 16 bit ciascuno.

Oltre ai messaggi di DR-Query, DR-Update, DR-Report e DB-Report, di seguito esposti, possono essere inoltrati dal DB ai DR messaggi di RP-Update, nel caso in cui avvenga l' handover di una sorgente oppure una nuova sorgente inizi a trasmettere un flusso verso un gruppo già esistente (caso descritto nel capitolo 4.2.4). Tali messaggi, essendo originati da un RP, saranno descritti nella prossima sezione (capitolo 5.2.2.3).

### DR-Query

Quando PIMD riceve dal processo Hostapd un messaggio che segnala l'associazione di un nuovo client sull'interfaccia di accesso, non conoscendo gli eventuali gruppi (con le relative sorgenti) di cui il client fa parte, invia verso il database server un pacchetto di segnalazione (DR-Query), che ha un duplice scopo: far aggiornare le informazioni presenti sul database a proposito della posizione del client nella rete, e richiedere al database stesso informazioni su quel client. I pacchetti usati per tale fine, oltre ai campi di header comuni agli altri pacchetti usati dal protocollo, hanno solo il campo *Client*, di 32 bit, che specifica appunto l'IP del client che ha eseguito la nuova associazione sull'interfaccia governata da Hostapd.

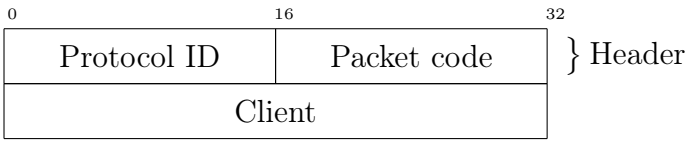



Figura 5.2: Messaggio di DR-Query

### DR-Update

I messaggi di DR-Update vengono inviati dai DR verso il DB ogni volta che l' arrivo di un pacchetto IGMP (da parte di un client) comporti qualche cambiamento nel database locale del DR che lo riceve; vale a dire, ogni volta che un client notifichi al proprio router di riferimento il proprio ingresso in un nuovo gruppo di multicast, o l'abbandono di un gruppo a cui prima partecipava.

I DR-Update hanno due scopi: mantenere aggiornate la tabella del DB che contiene le informazioni sui gruppi di cui fa parte ogni client e richiedere implicitamente al DB un messaggio di DB-Report in risposta, contenente l' elenco delle sorgenti per il gruppo (o i gruppi) segnalati nel DR-Update.

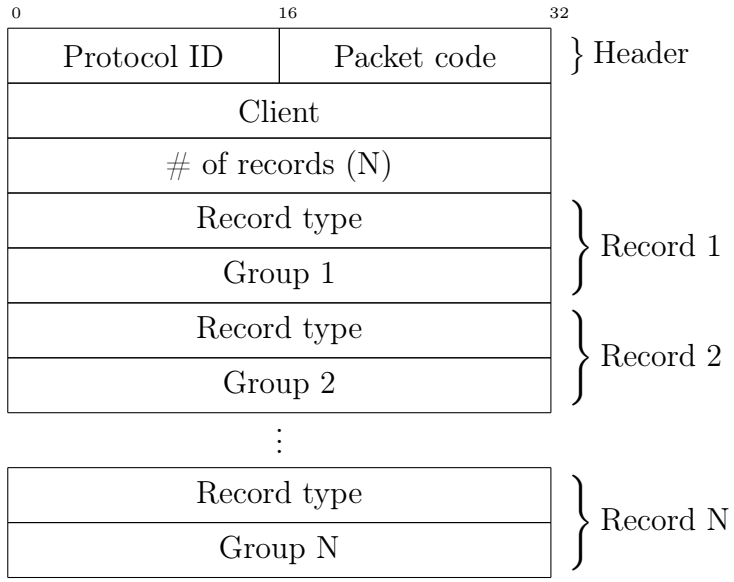



Figura 5.3: Messaggio di DR-Update

Il payload di un DR-Update è mostrato in figura 5.3 e contiene i seguenti campi:

- *Client*: il client a cui si riferisce il messaggio di update;
- *Number of records*: il numero di strutture (*record*) contenute nel messaggio, una per ogni gruppo segnalato;

- *Record type*: il tipo di record, indica se il record corrente è una leave o una join;
- *Group*: l' indirizzo IP del gruppo segnalato nel record corrente.

### DR-Report

I messaggi di DR-Report vengono inviati periodicamente (una volta ogni minuto all' incirca, secondo le impostazioni di default) da ogni DR al DB, per mantenere coerenti le informazioni contenute nel database centrale ed evitare che i timer relativi alle varie entry della tabella scadano. Un DR-Report contiene tutte le informazioni presenti nella tabella locale del DR, nel momento in cui questo lo spedisce.

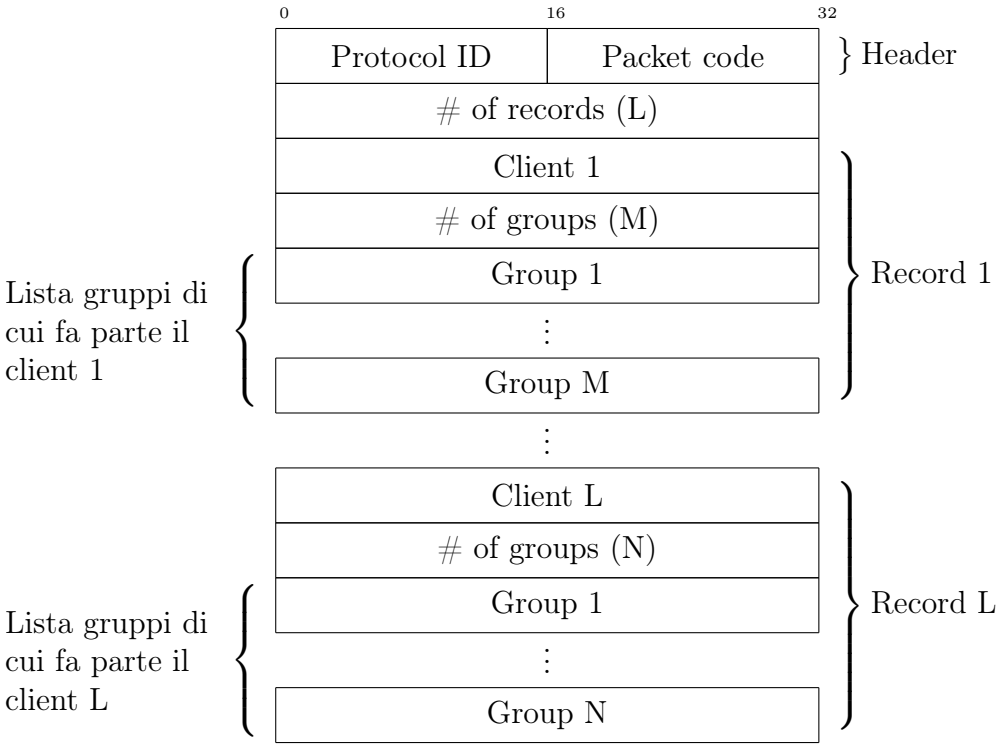



Figura 5.4: Messaggio di DR-Report

In figura 5.4 sono è mostrato il payload di un messaggio di DR-Report, che comprende i campi:

- *Number of records*: indica il numero di strutture di record contenute nel messaggio;
- *Client*: questo campo contiene l' indirizzo IP del client a cui si riferisce il record presente;
- *Number of groups*: il numero di gruppi riportati nel record presente;
- *Group*: in questi campi vengono segnalati gli indirizzi IP dei gruppi riportati nel DR-Report.

### DB-Report

I DB-Report vengono inviati dal DB in risposta a messaggi di DR-Query o di DR-Update dei DR. Essi servono quindi a trasmettere le informazioni richieste da un router a proposito dei gruppi (e relative sorgenti) di cui fa parte un dato client, ad esso associato.

Osservando il payload di un pacchetto di DB-Report, descritto in figura 5.5 si distinguono i campi:

- *Client*: in questo campo viene indicato l' indirizzo IP del client a cui il report si riferisce;
- *Number of records*: il numero di record contenuti nel messaggio;
- *Group*: l' indirizzo IP del gruppo a cui si riferisce il record corrente;
- *Number of sources*: il numero di sorgenti elencate nel record presente;
- *Source*: gli indirizzi IP delle sorgenti conosciute dal DB per il gruppo indicato nel record presente.

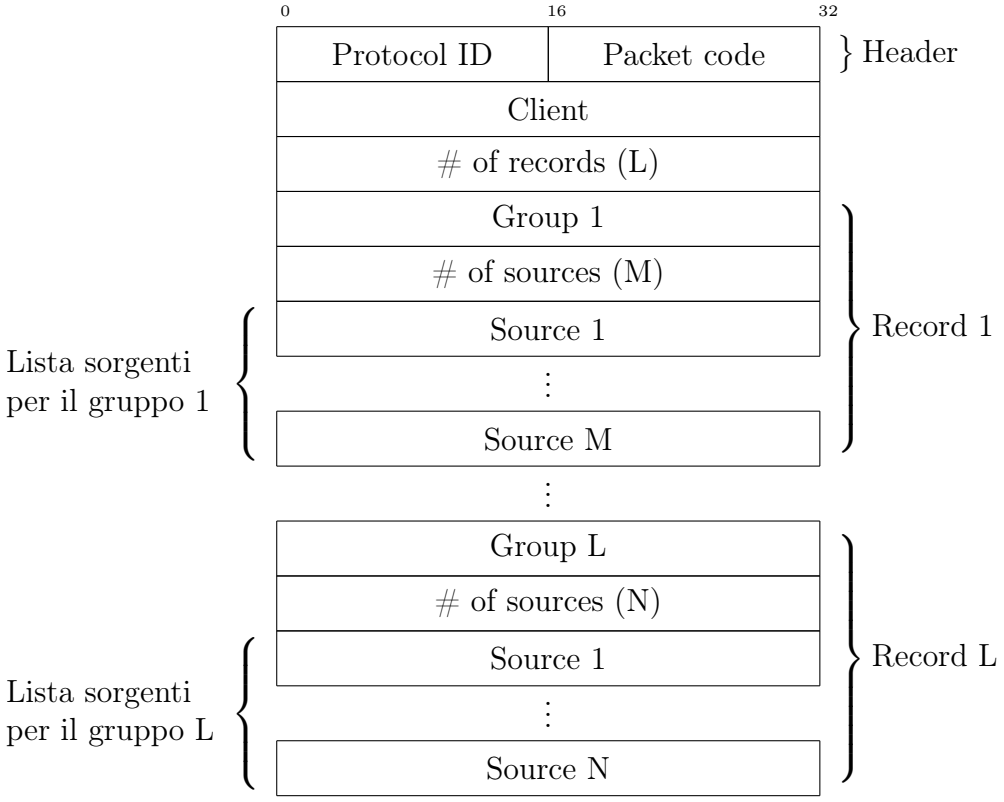



Figura 5.5: Messaggio di DB-Report

#### 5.2.2.3 Segnalazione RP - Database server

In questa sezione verranno esposti i messaggi scambiati tra rendez-vous point (RP) e database server (DB). Essi sono necessari per aggiornare e mantenere coerenti le informazioni presenti sul DB a proposito delle sorgenti dei vari gruppi multicast.

**RP-Update**

Gli RP-Update vengono inviati da un RP al DB in due tipi di occasioni: quando si attiva una nuova sorgente (per un gruppo multicast già presente in rete o meno) o quando l' RP si rende conto che una sorgente si è associata ad un nuovo DR (ossia ha eseguito un handover). In entrambi i casi, l' RP-Update, dopo essere ricevuto ed elaborato dal DB, viene inoltrato verso tutti i DR che in quel momento stanno

ricevendo i flussi multicast relativi al gruppo in questione, in modo che possano inviare una join verso la nuova sorgente.

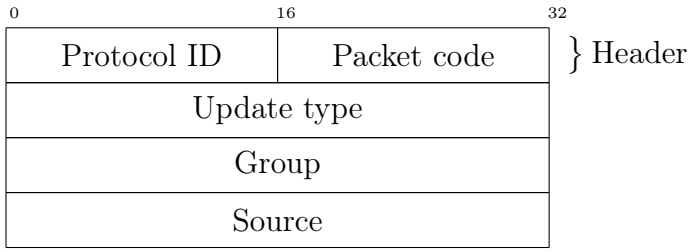



Figura 5.6: Messaggio di RP-Update

Un messaggio di RP-Update, mostrato in figura 5.6, oltre alla parte di header contiene le seguenti informazioni:

- *Type*: questo campo indica il tipo di RP-Update contenuto nel messaggio, ossia specifica se la sorgente indicata è nuova o era già presente in rete ed ha eseguito un handover;
- *Group*: indica l' indirizzo IP del gruppo verso cui la sorgente segnalata trasmette;
- *Source*: indica l' indirizzo IP della sorgente oggetto del messaggio.

### RP-Report

I messaggi di RP-Report sono inviati periodicamente dagli RP al DB, che li elabora e resetta i timer relativi alle entry della propria tabella centrale corrispondenti alle sorgenti segnalate attive nel messaggio. Questi messaggi servono quindi a riconfermare delle informazioni che sono state precedentemente comunicate al DB, ossia tutte quelle presenti nella tabella locale del rendez-vous point che trasmette il messaggio.

La struttura di un messaggio di RP-Report è mostrata in figura 5.7, e comprende i campi:

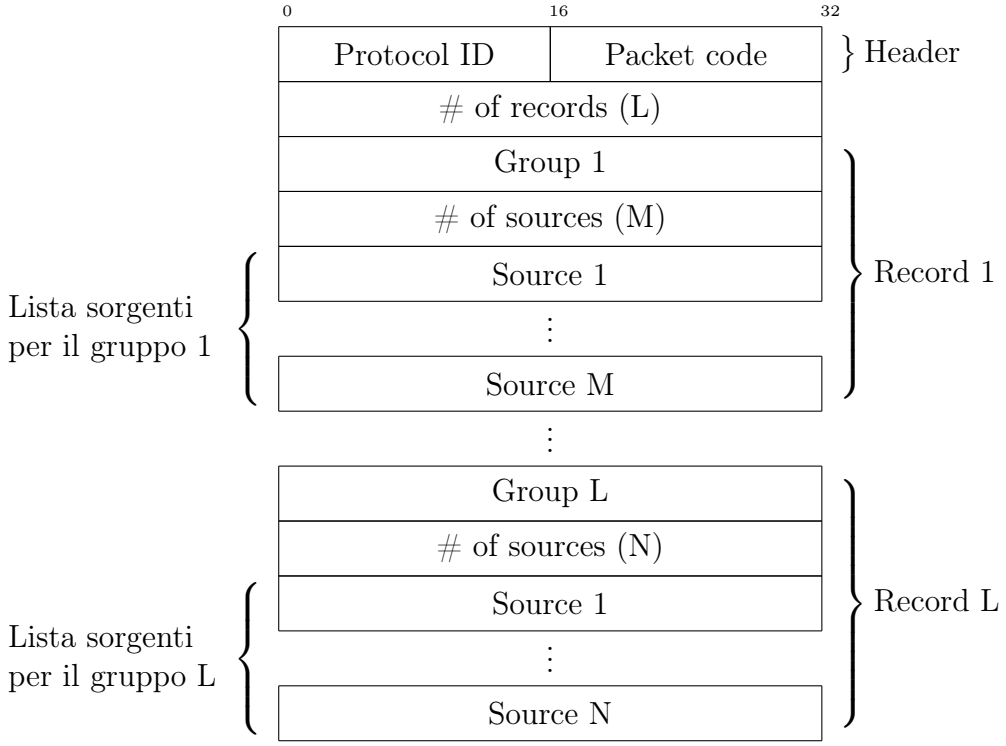



Figura 5.7: Messaggio di RP-Report

- *Number of records*: questo campo indica la cardinalità delle strutture di record contenute nel messaggio;
- *Group*: l' indirizzo IP del gruppo a cui si riferisce il record corrente;
- *Number of sources*: la cardinalità delle sorgenti elencate nel record corrente, ossia le sorgenti attive per il gruppo corrente di cui l' RP è a conoscenza;
- *Source*: in questi campi vengono segnalati gli indirizzi IP delle sorgenti attive conosciute per il gruppo corrente.

## 5.3 Gestione delle informazioni

Le informazioni scambiate in rete tra i vari nodi vengono organizzate dai diversi componenti sotto forma di database. Sui router multicast e sui rendez-vous

point sono presenti solo le informazioni che si riferiscono ai terminali di rispettiva competenza (client e sorgenti multicast direttamente collegati ad essi), aggiornate e mantenute consistenti tramite i vari messaggi descritti nella sezione 5.2.2, mentre sul database server viene mantenuta una copia di tutte le informazioni riguardanti ogni componente della rete multicast (host, DR, RP).

Si è scelto di organizzare i dati raccolti in *hash table*, ottenute mediante una libreria esposta nella sezione seguente; seguirà la descrizione dei database previsti dal protocollo, quelli locali ed il database centrale.

### 5.3.1 Uthash

Come detto, le informazioni necessarie ai componenti della rete sono organizzate in hash table; si è optato per questa soluzione invece che per l'uso di veri e propri *database management system* per una maggiore semplicità e portabilità del risultato finale. La libreria utilizzata per creare e manipolare le hash table è chiamata *Uthash* ([25]), un progetto open source, liberamente scaricabile sotto licenza BSD e presente nei repository di Debian/Ubuntu. Uthash consiste in realtà in un singolo file di header, scritto in linguaggio C, che implementa le macro per la gestione degli hash, le quali possono poi essere usate nel codice della propria applicazione; non è quindi necessaria l'interazione con programmi aggiuntivi per l'organizzazione delle tabelle, le hash table sono create e aggiornate direttamente dal demone PIMD. Sono supportate diverse operazioni, ma per l'implementazione delle tabelle nei componenti della rete sono stati usati solo i semplici comandi di *add*, *delete*, *find*, *count* e *iterate*, necessari rispettivamente per aggiungere o cancellare un oggetto all' hash table (gli hash vengono ottenuti dalle chiavi tramite l'algoritmo di Jenkins), trovare un oggetto tra quelli presenti, contare gli hash già generati ed eseguire più volte una data operazione sugli hash.

### 5.3.2 Database locali

Verranno ora descritte le strutture delle tabelle che contengono le informazioni salvate sui router di multicast (DR e RP), specificando l'uso del contenuto dei vari campi.

#### 5.3.2.1 Database dei DR

La hash table presente sui designated router viene usata da ciascun nodo per tenere traccia dei gruppi di cui ogni client associato al DR fa parte; ogni entry della tabella comprende i campi:

- *Client*, contiene l' indirizzo IP del client corrispondente alla entry. Per ogni client di multicast associato al DR viene creato un elemento di questo tipo;
- *Timer*, tempo di validità della entry. Allo scadere del timer, l' hash relativo ad essa viene cancellato ed l' host associato non viene più considerato client di multicast. Il timer viene resettato ad ogni ricezione di messaggio di IGMP Report da parte del client corrispondente;
- *Group List*, lista dei gruppi di cui il client corrente fa parte, per ogni voce della lista ci sono due elementi:
  - *Group*, indica l' IP di un gruppo di cui il client fa parte;
  - *Timer*, è il timer che indica il tempo di validità dell' associazione del gruppo indicato con il client presente, viene resettato ad ogni ricezione di IGMP Report da parte del client per quel gruppo. Allo scadere del timer, il gruppo viene considerato lasciato dal client e viene cancellata la voce corrispondente dalla lista dei gruppi.

#### 5.3.2.2 Database degli RP

La hash table presente sui rendez-vous point serve per tenere traccia delle sorgenti attive in rete per i vari gruppi multicast, e per rilevare i casi in cui una data sorgente esegue un handover. Le entry dell' hash table comprendono i campi:

- *Group*, indica l' indirizzo IP del gruppo corrispondente alla entry;
- *Source list*, contiene la lista delle sorgenti attive per il gruppo corrente, ogni voce della quale si suddivide negli elementi:
  - *Source*, indica l' indirizzo IP di una sorgente per il gruppo corrente;
  - *Router*, indica l' indirizzo IP del router al quale la sorgente è attualmente associata (ossia l' ultimo router che ha inviato messaggi di PIM-register per quella sorgente). Viene usato dall' RP per rilevare un eventuale handover della sorgente corrente;
  - *Timer*, indica il tempo di validità della voce corrente, rinnovato ogni volta che viene ricevuto un messaggio di PIM-register relativo alla sorgente corrente.

### 5.3.3 Database centrale

Sul database centrale sono mantenute due hash table, che ricalcano quelle presenti su DR ed RP, con alcune differenze che verranno di seguito esposte.

In primo luogo, la hash table relativa ai client contiene informazioni su tutti i client presenti in rete (a differenza di quelle presenti sui DR, che tengono traccia solo dei client locali). Oltre ai campi presenti sulle tabelle dei DR, inoltre, è stato aggiunto per ogni client indicizzato l' indirizzo IP del router a cui è associato, che viene aggiornato ad ogni ricezione di DR-Update; in questo modo il DB può scartare i messaggi di DR-Report in cui il router mittente non corrisponde, evitando di aggiornare la tabella con informazioni vecchie o non corrette.

La hash table che indicizza i gruppi multicast e le relative sorgenti, oltre ai campi presenti nelle tabelle degli RP, contiene per ogni entry l' elenco dei router che stanno ricevendo flussi destinati al gruppo corrente. Questo è necessario perchè, in caso l' RP rilevi una nuova sorgente attiva per un gruppo già esistente, il DB possa inoltrare verso ciascuno dei router in lista un RP-Update che la segnali (in modo che il DR possa inviare una PIM-join verso di essa).

# Capitolo 6

# Testing

Come anticipato, la variante del protocollo esposta in questo lavoro di tesi è stata implementata prevedendo un suo utilizzo su piattaforme Linux, ed in particolare su router ARM con sistema operativo OpenWRT. Si è adoperata una macchina di questo tipo per verificare che il demone di routing (PIMD) risultasse effettivamente funzionante, inviasse pacchetti con le strutture di header e payload descritte precedentemente ed elaborasse correttamente le informazioni ricevute, aggiornando le varie hash table in maniera opportuna.

Accertato il corretto funzionamento del programma in termini di svolgimento delle operazioni come invio, ricezione ed elaborazione di messaggi di notifica, update, query e report, aggiornamento delle tabelle ed inoltro dei flussi multicast, si è predisposta un' architettura di testbed per misurare un' altra componente vincolante per il risultato finale: gli effettivi tempi di handover nel caso in cui un client multicast migri da un access point (con funzioni di multicast) ad un altro. Nelle prossime sezioni verranno perciò descritti l' architettura di rete predisposta per le prove, il parco software utilizzato ed infine saranno esposti i risultati delle misurazioni.

## 6.1 Architettura di testbed

Per eseguire le prove di misurazione dei tempi di handover è stata predisposta l' architettura descritta in figura 6.1. Pur volendo ridurre al minimo i componenti della rete adoperati per i test, erano necessari almeno un RP, un database, una sorgente di multicast, due router per simulare l' handover ed il client che lo effettuasse. Si è deciso perciò di realizzare l' architettura necessaria tramite delle macchine virtuali collegate in rete nel modo illustrato, usando alcuni accorgimenti descritti nelle prossime sezioni.

Il dispositivo utilizzato per ospitare le virtualizzazioni è un PC desktop Acer Aspire M5400 dotato di processore AMD Phenom II X6 1055T a 2.8 GHz. Non essendo strettamente necessario ai fini delle misurazioni di handover che il database risedesse su un apparecchio dedicato, per risparmiare risorse il processo che gestisce l'hash table centrale e quello di PIMD designato ad operare come rendez-vous point sono stati posti sulla stessa macchina; per ciascun router (sia i due DR che l' RP/DB) sono stati riservati 512 MByte di memoria RAM e uno dei processori della macchina ospite, mentre per le macchine con ruolo di sorgente e client multicast è stata assegnata la memoria di 1 GByte RAM.

## 6.2 Software Utilizzati

Per la realizzazione della rete di virtualizzazioni e per generare ed analizzare i flussi di traffico su di essa sono stati adoperati diversi software, che saranno brevemente presentati nel prosieguo.

### Virtualbox

Virtualbox [26] è un' applicazione disponibile per sistemi Windows, MacOS e Linux per la realizzazione di macchine virtuali che operino sopra al sistema ospite. È possibile definire, per ogni macchina creata con Virtualbox, un set di interfacce

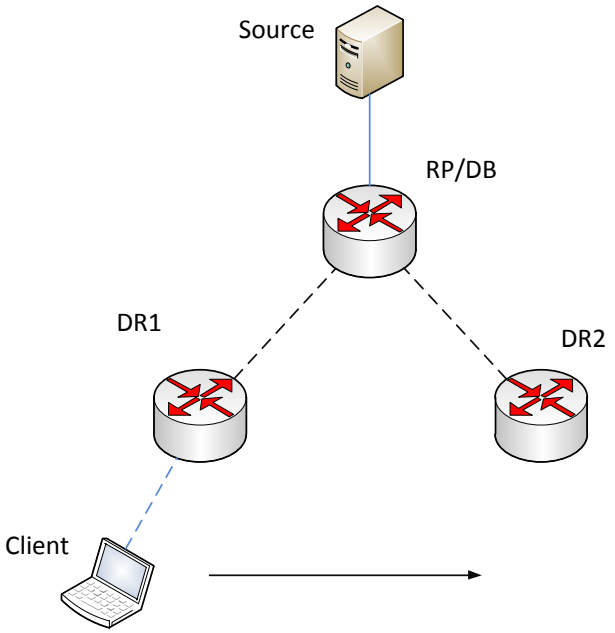



Figura 6.1: Scenario dell' architettura di test

(ovviamente virtuali) da cui inoltrare e ricevere traffico dati, proprio come se fosse dotata di schede ethernet o wi-fi, ed impostare i tipi di collegamenti simulati tra le varie virtualizzazioni.

Per la predisposizione dell' architettura di testbed, sono state create due macchine-client e tre macchine-router; su tutte opera un sistema operativo Ubuntu 10.04 LTS, con kernel 2.6.32 (compatibile con le applicazioni sviluppate in questa tesi), abilitato a scambiare messaggi di segnalazione IGMPv3.

## Netem

Netem è un tool presente su Ubuntu che offre numerose funzionalità volte a riprodurre il più fedelmente possibile il comportamento di una rete fisica in un ambiente controllato. Questa applicazione consente cioè di emulare il ritardo (costante o distribuito secondo una certa funzione di probabilità) nelle trasmissioni, la perdita e la duplicazione casuale di pacchetti e altri scenari, tipici di una

rete vera e propria.

In questo lavoro di tesi Netem è stato utilizzato principalmente per riprodurre il ritardo inevitabile introdotto dal mezzo trasmissivo wireless delle reti mesh wi-fi, stimato (per la rete MobiMESH a cui si fa riferimento) in media di circa 1,5 ms per ogni tratta tra un router e il suo successivo.

## VLC

VLC media player (VLC) [27] è un programma scaricabile liberamente, disponibile per tutti i sistemi, usato per la riproduzione di contenuti multimediali. È stato usato nell' ambito di queste prove di misurazione come tipico esempio di applicazione client e sorgente multicast, in quanto dispone di un' interfaccia apposita per impostare la trasmissione di flussi audio/video in pacchetti RTP o UDP, destinati (volendo) anche a gruppi multicast; in modalità client permette chiaramente di specificare di quale indirizzo della classe D si desidera entrare a far parte, in modo che poi il kernel possa occuparsi di comunicarlo al router con opportuni messaggi IGMP.

## Wireshark

Wireshark [28] è uno strumento che permette di catturare il traffico a pacchetti presente in una rete, di salvarlo e di analizzarlo (*packet sniffer*); il suo funzionamento si basa sull' uso di un filtro che riceve dai driver delle interfacce monitorate copie dei pacchetti inviati e ricevuti, li salva in memoria e tiene traccia dei tempi di interarrivo. Permette inoltre di analizzare i pacchetti stessi a livello di byte, supportando nativamente i protocolli più diffusi e dividendo automaticamente i flussi di varia natura. È quindi l' applicazione che ha reso possibile l' analisi dei tempi di handover nei test effettuati.

## 6.3 Misure di handover

Come precedentemente esposto, l' architettura di testbed è stata realizzata interamente tramite virtualizzazioni; questo ha comportato l' uso di qualche accorgimento per la simulazione dell' handover che il client effettua dal DR1 al DR2. Nella situazione di partenza, la macchina Source si registra presso il router che svolge le funzioni di RP e database come sorgente di un gruppo multicast, la macchina Client sottoscrive con messaggi IGMP al DR1 la propria adesione a tale gruppo, e terminata la procedura di join comincia a ricevere dalla sorgente il flusso multicast (uno stream video trasmesso con VLC).

A causa della natura della rete realizzata, non è possibile spostare fisicamente il client in modo che esegua realmente un handover. La soluzione adottata è stata quella di attivare uno script sul router DR2, che inviasse una notifica di associazione (descritta nella sezione 5.2.1.2) indicando l' indirizzo IP del client; contemporaneamente, venivano disattivate le funzionalità di rete della macchina Client, senza però inviare segnalazione di IGMP leave o disassociazione. Quando riceve la notifica di associazione il router DR2 incomincia lo scambio di segnalazione descritto nella sezione 4.2.3, e la procedura di handover ha inizio.

Per valutare le prestazioni del protocollo e del demone che lo implementa sono state fatte 7 prove di handover nelle modalità appena esposte, i cui risultati sono illustrati nella tabella 6.1. Le latenze illustrate in tabella sono tali da garantire

| esperimento | latenza (ms) |
|---|---|
| primo handover | 17,498 |
| secondo handover | 21,997 |
| terzo handover | 13,027 |
| quarto handover | 15,145 |
| quinto handover | 15,828 |
| sesto handover | 17,545 |
| settimo handover | 17,29 |
| tempo medio | 16,9 |

Tabella 6.1: Tempi di handover

la continuità nella ricezione del traffico multicast da parte del client, evitando la vera e propria interruzione del servizio a cui si andrebbe incontro usando il protocollo di PIM-SM classico implementato dal demone PIMD originale; esso infatti, come descritto nel capitolo 5.1.2, di fatto non prevede la mobilità dei client nella rete. Nel caso dell' esperimento proposto, il protocollo non modificato implicherebbe che il client, spostatosi dal DR1 al DR2, per continuare a ricevere il flusso multicast dovesse attendere una delle query IGMP periodiche del DR (tipicamente inviate con intervalli di 60 secondi), rispondere segnalando in un report il gruppo, ed aspettare almeno il ricongiungimento del nodo con lo shared tree (con radice nell' RP).

È importante sottolineare che il tempo di handover risultante dagli esperimenti effettuati non comprende però la latenza dovuta alla procedura di associazione tra il cient e il nuovo router: per via degli accorgimenti appena descritti, il calcolo del ritardo parte dall' istante in cui il DR riceve la notifica di Hostapd, e non da quello in cui il client incomincia fisicamente l'handover; un' ulteriore incremento delle latenze potrebbe essere introdotto dall' utilizzo di nodi meno performanti delle macchine usate nel test, caso che facilmente si verifica in reti mesh reali, in cui il risparmio energetico (con conseguente riduzione della capacità computazionale) gioca un ruolo fondamentale.

Si osserva inoltre che il testbed utilizzato presenta un' architettura piuttosto semplice, con una topologia ridotta, ed i router comunicano direttamente con il nodo di database. In uno scenario più complesso, come quello di una rete reale, verrebbero introdotti ritardi aggiuntivi dipendenti dalla distanza che separa i router interessati dall' handover ed il DB, a causa delle connessioni TCP che vengono instaurate per l' aggiornamento delle hash table sui nodi.

Un passo ulteriore verso un protocollo più distribuito che garantirebbe probabilmente ritardi inferiori potrebbe essere quello di sfruttare le caratteristiche del demone di gestione della mobilità di MobiMESH per costruire all'interno di ogni nodo una tabella di routing multicast che comprendesse le informazioni riguar-

danti tutti i client e le sorgenti presenti in rete. Poichè il meccanismo di routing di MobiMESH prevede già che ciascun router conservi in memoria delle tabelle con tutte le associazioni tra gli host e gli access point della rete, aggiungere per ogni host (client o sorgente) indicato nelle tabelle l' elenco dei gruppi multicast dei quali fa parte o verso i quali trasmette non dovrebbe essere eccessivamente oneroso, sia in termini di memoria necessaria che di overhead di segnalazione.

# Capitolo 7

# Conclusioni

Con la diffusione sempre più pervasiva delle reti wireless nelle città, sul posto di lavoro e nei luoghi pubblici, si è resa sempre più necessaria una tecnologia che consenta il rapido dispiegamento di reti che permettano il collegamento ad Internet in banda larga, mantenendo bassi i costi e i tempi di installazione e garantendo agli utenti la possibilità di spostarsi durante la fruizione del servizio senza una eccessiva perdita della qualità durante la migrazione tra gli access point (*seamless roaming*).

In questo scenario prendono sempre più piede le *Wireless Mesh Network* (WMN), reti caratterizzate da collegamenti realizzati completamente su mezzo radio, sia a lato accesso che nel backhaul, e da una topologia magliata con inoltro in modalità multi-hop.

La rete *MobiMESH* è un' implementazione di questo tipo di network; essa comprende un sistema di gestione della mobilità a livello 3 che permette agli utenti di spostarsi da un access point all' altro all' interno dell' area di copertura della rete in maniera completamente trasparente, ossia senza avvertire ritardi significativi o interruzioni del servizio a causa della perdita di pacchetti.

Poichè il risparmio e l' ottimizzazione nell' uso delle risorse sono sempre aspetti fondamentali nelle reti in generale e nelle WMN in particolare, si è pensato

di realizzare un servizio di inoltro multicast, riferendosi in particolare alla rete MobiMESH, che tenesse conto delle esigenze di gestione della mobilità degli utenti.

I protocolli di routing multicast più diffusi in letteratura permettono una grande ottimizzazione delle risorse nell' inoltro di contenuti fruiti contemporaneamente da più utenti (tipicamente audio/video), inviando un solo flusso per ciascun l' albero di distribuzione, che viene replicato dai nodi all' occorrenza (ossia quando l' albero si divide in più rami). I protocolli standard, però, di fatto non prendono in considerazione alcun requisito di mobilità da parte degli utenti/client, che in caso di handover sprimentano un' interruzione del servizio che può durare svariati secondi.

In questa tesi si è quindi proposta ed implementata una variante di uno dei protocolli di multicast standard più diffusi (*Protocol Independent Multicast - Sparse Mode*, PIM-SM) che tenesse conto di tale esigenza di mobilità. Nell' architettura prevista dall' implementazione sono stati introdotti e modificati alcuni elementi rispetto a quella tradizionale del PIM-SM, e sono stati aggiunti vari messaggi di segnalazione necessari al mappaggio della posizione dei client e delle sorgenti multicast su un database centrale, in modo che questo possa venire interrogato dai router interessati da un handover per un veloce riallacciamento all' albero di distribuzione.

I risultati sperimentali ottenuti testando il protocollo implementato, seppur con alcune limitazioni, mostrano che in ogni caso i tempi morti nell' aggiornamento dell' albero di distribuzione presenti del protocollo originale sono stati di molto ridotti (pur dipendendo dalla topologia della rete, sono comunque dell' ordine delle decine di millisecondi). Un altro aspetto da sottolineare è che l' interoperabilità con la versione originale del protocollo PIM-SM è stata mantenuta, in modo che il traffico multicast non sia necessariamente limitato all' area geografica della rete che lo ospita, ma possa essere inoltrato verso l' esterno (o dall' esterno essere diffuso nella rete in esame).

Possibili sviluppi futuri del protocollo proposto potrebbero andare nella direzione di un algoritmo maggiormente distribuito, in modo da eliminare la necessità di consultare un database centrale da parte dei router, evitando effetti collo di bottiglia presso di esso e riducendo le latenze dovute allo scambio dei messaggi di segnalazione.

# Bibliografia


[1] J. Fakatselis and C. F. Andren. Wireless LAN Medium Access Control and Physical Layer specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band. IEEE- SA Standards Board, September 1999.

[2] Radio Equipment and Systems; High-Performance Radio Local Area Network (HIPERLAN) Type 1; Functional specification. ETS, April 1996.

[3] R. B. Marks, K. Stanwood, and D. Chang. Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems. IEEE- SA Standards Board, October 2004.

[4] Information Processing System - Open System Interconnection, Basic Reference Manual. ISO/IEC-ITFF, November 1989.

[5] A. Capone, M. Cesana, S. Napoli, A. Pollastro, and Politecnico Di Milano. MobiMESH: a Complete Solution for Wireless Mesh Networking. IEEE MASS 2007 (4th IEEE International Conference on Mobile Ad Hoc and Sensor Systems), Pisa, Italy, 8-11 October 2007.

[6] S. Napoli and A. Pollastro. MobiMESH: Design and Implementation of a New Wireless Mesh Architecture with Mobility Support. Master Degree, Politecnico di Milano, October 2005.

[7] M. B. Shoemake, J. Terry, C. F. Andren, and K. Smart. Wireless LAN Medium Access Control and Physical Layer specifications: Further High-

Speed Physical Layer Extension in the 2.4 GHz Band. IEEE-SA Standards Board, June 2003. URL `http://standards.ieee.org/getieee802/download/802.11g-2003.pdf`.

[8] N. Chayat and H. Takanashi. Wireless LAN Medium Access Control and Physical Layer specifications: High-Speed Physical Layer in the 5 GHz Band. IEEE-SA Standards Board, September 1999. URL `http://standards.ieee.org/getieee802/download/802.11a-1999.pdf`.

[9] T. Clausen and P. Jacquet. RFC 3626: Optimized Link State Routing Protocol (OLSR), October 2003. URL `http://www.rfc-editor.org/rfc/rfc3626.txt`. Status: EXPERIMENTAL.

[10] S.E. Deering. RFC 1112: Host extensions for IP multicasting, August 1989. URL `http://www.rfc-editor.org/rfc/rfc1112.txt`. Updated by RFC 2236 [12], Status: STANDARD.

[11] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan. RFC 3376: Internet Group Management Protocol (IGMP), Version 3, October 2002. URL `ftp://ftp.rfc-editor.org/in-notes/rfc3376.txt`. Obsoletes RFC 2236 [12], Updated by RFC 4604 [24]. Status: PROPOSED STANDARD.

[12] W. Fenner. RFC 2236: Internet Group Management Protocol (IGMP), Version 2, November 1997. URL `ftp://ftp.rfc-editor.org/in-notes/rfc2236.txt`. Updates RFC 1112 [10], Obsoleted by RFC 3376 [11]. Status: PROPOSED STANDARD.

[13] P. Savola. RFC 5110: Overview of the Internet Multicast Routing Architecture, January 2008. URL `http://www.rfc-editor.org/rfc/rfc5110.txt`. Status: INFORMATIONAL.

[14] A. Ballardie. RFC 2189: Core Based Trees (CBT version 2) Multicast Routing, September 1997. URL `http://www.rfc-editor.org/rfc/rfc2189.txt`. Status: EXPERIMENTAL.

[15] J. Moy. RFC 1584: Multicast Extensions to OSPF, March 1994. URL `http://www.rfc-editor.org/rfc/rfc1584.txt`. Status: HISTORIC.

[16] D. Waitzman, C. Partridge, and S. E. Deering. RFC 1075: Distance Vector Multicast Routing Protocol (DVMRP), November 1988. URL `fhttp://www.rfc-editor.org/rfc/rfc1075.txt`. Status: EXPERIMENTAL.

[17] A. Adams, J. Nicholas, and W. Siadak. RFC 3973: Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised), January 2005. URL `http://www.rfc-editor.org/rfc/rfc3973.txt`. Status: EXPERIMENTAL.

[18] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. RFC 2362: Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification, June 1998. URL `http://www.rfc-editor.org/rfc/rfc2362.txt`. Obsoletes RFC 2117, Obsoleted by RFC 4601 [19]. Status: EXPERIMENTAL.

[19] Bill Fenner, Mark Handley, Hugh Holbrook, and Isidor Kouvelas. RFC 4601: Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised). RFC 4601, IETF, August 2006. URL `http://www.rfc-editor.org/rfc/rfc4601.txt`. Obsoletes RFC 2362 [18], Updated by RFC 5059, Status: PROPOSED STANDARD.

[20] N. Bhaskar, A. Gall, J. Lingard, S. Venaas. RFC 5059: Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM), January 2008. URL `http://www.rfc-editor.org/rfc/rfc5059.txt`. Obsoletes RFC 2362 [18], Updates RFC 4601 [19], Status: PROPOSED STANDARD.

[21] Ahmed Helmy, George Edmond Rusty Eddy, and Pavlin Ivanov Radoslavov. PIMD, 1998-2011. URL `http://vmlinux.org/jocke/pimd.shtml`.

[22] J. Malinen. Hostapd: IEEE 802.11 AP, IEEE 802.1X, WPD, 2002-2010. URL `http://hostapd.epitest.fi/hostapd/`.

[23] T. Jeffree, N. Jarvis, and M. Seaman. Local and Metropolitan Area Networks: Port-Based Network Access Control. IEEE- SA Standards Board, June 2001. URL `http://standards.ieee.org/getieee802/download/802.1X-2001.pdf`.

[24] H. Holbrook, B. Cain, and B. Haberman. RFC 4604: Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast. RFC 4604, IETF, August 2006. URL `http://www.rfc-editor.org/rfc/rfc4604.txt`. Updates RFC 3376 [11], Status: PROPOSED STANDARD.

[25] Troy D. Hanson. Uthash: a hash table for C structures, 2005-2011. URL `http://uthash.sourceforge.net`.

[26] Oracle Corporation. Virtualbox, 2004-2011. URL `http://www.virtualbox.org/`.

[27] VideoLAN Organization. VLC media player. URL `http://www.videolan.org/`.

[28] G. Combs. Wireshark: Network Protocol Analyzer, 1998-2011. URL `http://www.wireshark.org`.

[29] D.E. Comer. *Internetworking con TCP/IP*. Number v. 1 in Internetworking con TCP/IP. Pearson Education Italia, 2006. ISBN 9788871922805. URL `http://books.google.com/books?id=cVP6VhwgHkkC`.