ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione

"Guglielmo Marconi"

-DEI-

CORSO DI LAUREA IN

INGEGNERIA DELLE TELECOMUNICAZIONI

TESI DI LAUREA

in

Laboratorio di Telecomunicazioni M

SPERIMENTAZIONE DI COMUNICAZIONI VEICOLARI IEEE 802.11p PER L'OFFLOADING DI RETI CELLULARI

CANDIDATO

Luca Gaetano Nisi

RELATORE:

Prof. Ing. **Alessandro Bazzi**

CORRELATORI

Dott.sa Ing. **Barbara Mavì Masini**

Dott. Ing. **Alberto Zanella**

Anno Accademico 2013/14

Sessione II

"Se ho fatto qualche scoperta di valore, ciò è dovuto più ad un'attenzione paziente che a qualsiasi altro talento"

Isaac Newton

Sommario

Introduzione

Siamo nell'era moderna, figli del petrolio e ormai assuefatti al progresso. Statistiche alla mano, in Italia 610 persone su 1000 hanno un'auto; considerando che tra quei 390 sono anche incluse le persone al di sotto dei 18 anni, così come chi non ha la patente, praticamente ogni persona in grado di guidare possiede un'automobile. Si è deciso quindi di trovare immediate soluzioni riguardanti i problemi connessi alla crescente densità automobilistica, quali, ad esempio, il maggiore rischio di incidenti, la congestione delle strade, l'inquinamento atmosferico…

La direzione verso cui gli enti dei vari Paesi hanno deciso di muoversi è quella della sperimentazione di comunicazioni inter-veicolari. Si è deciso quindi di cambiare rotta rispetto al passato. Negli anni precedenti la ricerca si è concentrata sul miglioramento dei sistemi utilizzati per ridurre il rischio di incidenti (o il trauma fisico che ne conseguiva), come gli airbag (che si sono evoluti da airbag esclusivamente frontali ad airbag frontali e laterali), o come l'assistenza alla frenata (ABS), o ancora il miglioramento della stabilità e dell'assetto del veicolo. Adesso, invece, la parola chiave è "smart", intelligenza, veicoli intelligenti capaci di comunicare tra di loro, di scambiarsi informazioni che in alcune occasioni potrebbero anche rivelarsi vitali. Un veicolo che segue a distanza ravvicinata chi lo precede sarà quindi informato, ad esempio, della velocità a cui viaggia chi gli sta davanti, tramite messaggio sonoro e/o visivo gli sarà comunicato inoltre di aumentare la distanza di sicurezza, così come gli saranno segnalati eventuali ostacoli sulla carreggiata, nonché incidenti stradali avvenuti nelle vicinanze. Questo sistema permetterà inoltre una corretta gestione del traffico, evitando quindi congestioni nelle grandi arterie stradali e smaltendo il flusso dei veicoli su vie secondarie, permettendo quindi agli automobilisti di risparmiare tempo (e denaro) e di diminuire l'inquinamento atmosferico.

Lo standard che è stato sviluppato con l'intento di risolvere i problemi sopra citati è l'IEEE 802.11p, una variante del ben noto IEEE 802.11 (relativo alle comunicazioni wireless a breve raggio). In questo ambito, gli Stati Uniti hanno regolamentato la porzione di spettro da adoperare per le comunicazioni inter-

veicolari, definendo anche l'ampiezza di banda da utilizzare. La Federal Communication Commission ha allocato 75 MHz di banda nell'intervallo di frequenze che va da 5.85 GHz a 5.925 GHz, con un canale di controllo alla frequenza 5.89 GHz, e 6 canali di servizio [22]. Anche l'Europa si è mossa in quest'ambito, infatti la European Conference of Postal and Telecommunications (CEPT) ha allocato 50 MHz di banda nell'intervallo di frequenze che va da 5.855 GHz a 5.905 GHz, con un canale di controllo a 5.9 GHz e solamente 4 canali di servizio [22].

Le VANET possono anche essere sfruttate per liberare parte delle risorse altrimenti occupate dalle reti cellulari, che risultano essere sempre più congestionate. Si parla quindi di Offloading dalla rete cellulare. Si utilizzeranno tecnologie di altre reti di comunicazione (Wi-Fi, VANET…) per recapitare a destinazione dati programmati per essere inviati tramite rete cellulare. In questo ambito, sono state svolte alcune misure in laboratorio, in cui i pacchetti venivano dapprima trasmessi tramite rete infrastruttura (la rete LAN della Scuola di Ingegneria e Architettura di Bologna), in seguito, a partire da un istante scelto da input, si effettuava l'offloading su rete Ad-Hoc.

In questa tesi saranno descritte le caratteristiche principali delle reti VANET (Vehicular Ad-hoc Networks), includendo alcuni possibili scenari di applicazione, fornendo inoltre alcuni dettagli riguardanti lo standard 802.11p e la competizione internazionale GCDC (Grand Cooperative Driving Challenge) che si ripropone di fornire un sistema in grado di far cooperare veicoli ed infrastrutture. Successivamente descriverò i dispositivi ALIX utilizzati per effettuare le misure in laboratorio, compiendo anche un excursus sul codice Java utilizzato e sulle possibili modalità con cui effettuare l'offloading da rete cellulare, per poi concludere con i risultati sperimentali.

I capitoli sono organizzati come segue:

- Capitolo 1: Reti Vanet, panoramica introduttiva con elenco degli scenari di applicazione e descrizione della tecnologia utilizzata.
- Capitolo 2: Descrizione dello standard IEEE 802.11p, focalizzando l'attenzione sul livello fisico e MAC.
- Capitolo 3: Offloading e varie modalità con cui effettuarlo.

- Capitolo 4: Descrizione del GCDC, dei dispositivi ALIX utilizzati per le misure, delle modalità di configurazione di tali dispositivi e della relativa connessione al computer. Modifiche ai dispositivi ALIX per supportare i due canali di trasmissione, centrati sulle frequenze 5.88 GHz e 5.9 GHz, con relativa descrizione della configurazione necessaria all'utilizzo delle due schede wireless.
- Capitolo 5: Excursus sul codice Java utilizzato, sulle modifiche apportate per effettuare le misure, sulle classi Java aggiunte per supportare tale codice.
- Capitolo 6: Risultati sperimentali.

1. RETI VANET

1.1 Panoramica introduttiva

A partire dagli ultimi vent'anni, a causa del sempre crescente numero di veicoli, si sono venuti a creare numerosi gruppi di ricerca, i cui obiettivi erano il miglioramento delle condizioni di viaggio degli automobilisti: organizzazione efficiente del traffico, sicurezza stradale, riduzione dell'inquinamento; questi erano i tre punti chiave delle loro ricerche. Uno dei più famosi gruppi di studio fu l'IVHS (Intelligent Vehicle Highway System), oggi conosciuto come ITS (Intelligent Transportation System). Questo gruppo nacque dalla necessità di gestire i problemi sopracitati attraverso le reti di comunicazione. Successivamente, nel 1996, il Dipartimento dei Trasporti degli Stati Uniti, l'ITSA (Intelligent Transportation Society of America) insieme ad altre associazioni creò una struttura chiamata NITSA (National Intelligent Transportation System Architecture), con lo scopo di pianificare e sviluppare i servizi di trasporto intelligente. All'epoca, l'unica porzione di banda utile per questo tipo di applicazioni era quella compresa tra i 902 e i 928 MHz. Purtroppo tale banda si rivelò insufficiente, quindi poco funzionale. La FCC (Federal Communication Commission) assegnò allora nel 1999 una porzione di banda pari a 75 MHz, compresa tra i 5.85 GHz e i 5.925 GHz, con l'obiettivo di utilizzare tale risorsa spettrale per il DSRC (Dedicated Short Range Communication). L'ultimo passo verso la nascita di un nuovo standard si compì nel 2002, quando vennero poste le basi per l'IEEE 802.11. Un paio di anni dopo si studiarono alcune possibili varianti di tale standard, in modo da renderlo applicabile alle reti VANET. Nacque così l'802.11p.

1.2 Le reti

Le reti VANET (Vehicular Ad-hoc Network) sono reti ad hoc, costituite da un insieme di nodi mobili, altamente dinamiche, in cui i vari nodi possono comunicare via wireless senza la presenza di strutture preesistenti. Una peculiarità di queste reti è l'implementazione del multi-hop, tecnica secondo cui un determinato nodo può svolgere anche funzioni di routing per messaggi destinati ai suoi vicini [19]. La VANET non ha una topologia fissa, i nodi devono periodicamente inviarsi messaggi con l'obiettivo di conoscere i propri vicini e creare così una mappa della rete. Esistono due tipologie di comunicazione nelle reti Vanet:

- V2V: (Vehicle-to-Vehicle) comunicazione tra veicolo e veicolo;
- V2I: (Vehicle-to-Infrastructure) comunicazione tra il veicolo e l'infrastruttura, ovvero le road side units (RSU).

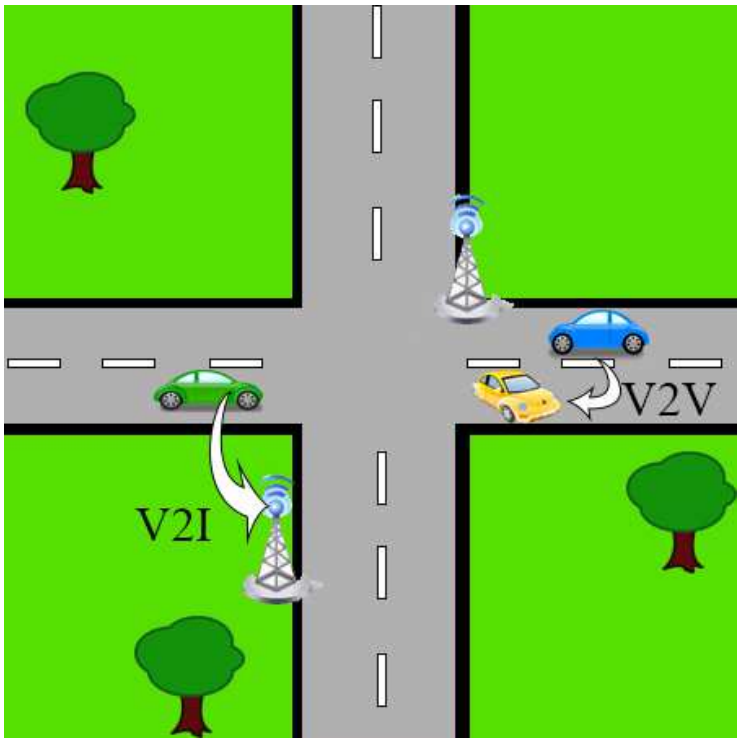

Figura 1: Esempio di comunicazione V2I e V2V

I dispositivi radio lavorano nelle bande non licenziate [1], più specificatamente nell'intervallo spettrale compreso tra 5.850 - 5.925 GHz [3].

Gli scenari in cui tale tecnologia può essere utilizzata sono molto differenti tra di loro: reti autostradali, vie a scorrimento veloce, strade urbane o vie suburbane, ognuna ha una diversa caratteristica di mobilità. Su strade ad alto scorrimento le elevate velocità dei veicoli sono fonte di frequenti disconnessioni dalla rete, con conseguente riconnessione, pertanto la comunicazione avviene in un ambiente che muta a seconda della densità veicolare o di particolari ostruzioni che ostacolano la comunicazione diretta. Sono necessari quindi diversi algoritmi per l'instradamento dei messaggi, single-hop o multi-hop. Infine, essendo assenti le infrastrutture fisse, la gestione e il controllo della rete è affidato ai nodi che la compongono, così come la sicurezza della comunicazione.

Un ulteriore problema è dato dalla mancata visibilità diretta di due nodi della rete, noto come *problema del terminale nascosto*: due nodi provano a usare contemporaneamente il canale di comunicazione, causando collisioni di pacchetti e quindi perdita d'informazione. Dualmente, esiste anche il problema del *terminale esposto*: i nodi vorrebbero comunicare con altri nodi esterni, ma uno dei nodi non può iniziare la comunicazione, perché il canale risulta occupato dalla comunicazione dell'altro. Si deve avviare quindi una procedura di *backoff*.

Un altro aspetto da gestire è l'interferenza: le reti VANET, realizzate mediante mezzo wireless, risentono del fading e delle interferenze con i segnali trasmessi da altri veicoli.

Altri aspetti che bisogna tenere in considerazione sono:

1. La possibilità di creare nuovi algoritmi per massimizzare le prestazioni;
2. Il raggiungimento di un buon valore della QoS (Quality of Service) tramite accorgimenti quali la riduzione di pacchetti ridondanti, l'efficiente gestione della banda e la riduzione al minimo del numero di pacchetti persi.

I veicoli devono essere messi in grado di comunicare gli uni con gli altri, ciò è reso possibile dai dispositivi wireless montati a bordo delle automobili, le OBU (On Board Unit), e da quelli installati lungo le strade, le RSU. Con lo standard

802.11 la bit rate può essere dell'ordine delle decine di Mbps [1], utilizzando hardware a basso costo.

1.3 Possibili applicazioni

L'ambito di applicazione di maggiore interesse è indubbiamente la sicurezza stradale, anche se sono possibili anche applicazioni legate alla gestione del traffico o all'ambito commerciale (collegamento al web, messaggi pubblicitari…) [2]. Il primo tipo mira ad evitare rischi di incidenti e rendere la guida sicura tramite l'invio di messaggi riguardo rischi ed ostacoli. La seconda categoria si focalizza sul ridurre la congestione stradale, migliorando la viabilità, riducendo il tempo di viaggio. La terza categoria, infine, si concentra su un aspetto puramente commerciale e di intrattenimento.
Alcune esempi:

1. richiesta di stop o rallentamento a causa di incidenti/lavori stradali;
2. informazioni riguardanti la situazione del traffico (ciò diminuirebbe l'inquinamento atmosferico e migliorerebbe l'efficienza della rete stradale);
3. pagamento dei caselli autostradali senza necessità di sosta;
4. prevenzione degli incidenti tramite invio di informazioni sulle condizioni del manto stradale o sulla visibilità in un determinato tratto;
5. accesso al web, quindi intrattenimento, acquisti online, pubblicità;
6. Post Crash Networking (PCN) [3]: in tale scenario, un veicolo che ha avuto un incidente può inviare in broadcast alcuni messaggi di avvertimento, in cui inserisce la propria posizione (cosicché gli altri guidatori possano aver tempo necessario per dirigersi su strade alternative). Il PCN può implementare sia le comunicazioni V2V sia le V2I. Nel primo caso, si ha un vantaggio in termini di velocità di informazione, senza contare la possibilità che il veicolo che ha appena ricevuto il messaggio, può inoltrare tale comunicazione ai veicoli dietro di lui, disseminando la rete con tali messaggi;
7. vision enhancement: il compito di tale applicazione è dare ai guidatori una chiara visuale dei veicoli e degli ostacoli [2];

8. segnalazione di eventuali stazioni di rifornimento, hotel, attrazioni turistiche;
9. applicazioni anti-collisione: le cause tipiche di tamponamento sono la distrazione dei conducenti, il mancato rispetto della distanza di sicurezza, una frenata improvvisa. Questi pericoli possono essere evitati, utilizzando veicoli che comunicano tra di loro durante la marcia, che quindi possono scambiarsi informazioni su posizione, velocità, distanza di sicurezza. Se il software rileva che, in base ai parametri monitorati, il rischio di incidenti aumenta, segnala prontamente la situazione al guidatore in modo da poterlo mettere in guardia;
10. gestione efficiente del traffico: queste applicazioni hanno lo scopo di informare il guidatore sulla situazione della rete di trasporto automobilistica, evitando congestioni, diminuendo i tempi di spostamento, riducendo l'inquinamento, ma anche quello di fornire informazioni utili ai gestori della rete. I dati raccolti dai vari veicoli, e quindi dalle OBU, vengono poi trasmessi alle RSU che hanno il compito di elaborarli e quindi ritrasmettere i risultati ai viaggiatori;
11. onda verde: informazioni sulla velocità da mantenere per far sì che ogni semaforo che l'utente attraverserà possa essere verde. Tutto ciò favorisce un maggiore smaltimento del traffico;
12. navigazione in internet: Esiste anche la possibilità di navigare sul web dalla propria auto, attraverso le comunicazioni V2I, e basando tali connessioni sul protocollo IP. Le RSU mantengono una connessione stabile con il veicolo, e fungono da gateway per collegarsi con il mondo esterno. Ciò è possibile in quanto viene eseguito un instradamento multi-hop dei pacchetti: il veicolo interagisce con diverse RSU durante il suo percorso, ma tutto ciò è trasparente ai livelli protocollari superiori, che vedono solo una connessione stabile. Con un'applicazione di questo tipo, le RSU possono trasmettere ai veicoli qualsiasi tipo di informazione (anche commerciali o pubblicitarie);

13. avaria del veicolo: l'applicazione può trasmettere una richiesta di soccorso all'officina più vicina, al fine di minimizzare i tempi di intervento.

1.4 Tecnologia

La principale novità utilizzata in quest'ambito è lo standard 802.11p. Il protocollo 802.11p descrive i protocolli dei livelli fisico e MAC del sistema Wireless Access in Vehicular Network (WAVE). Tale sistema utilizza le frequenze dedicate al DSRC (Dedicated Short Range Communication), per permettere trasmissioni con bassa latenza e alta velocità. La banda prevista negli Stati Uniti per le comunicazioni (70 MHz) è suddivisa in 7 diversi canali, ognuno con un'ampiezza di banda pari a 10 MHz [3]. Il primo canale è utilizzato per la comunicazione V2V, il secondo e il terzo sono canali privati, utilizzati per la sicurezza pubblica a medio raggio. Il quarto canale è un canale di controllo, mentre quinto e sesto canale sono utilizzati per la sicurezza pubblica a breve raggio. L'ultimo è utilizzato per gestire la sicurezza negli incroci.

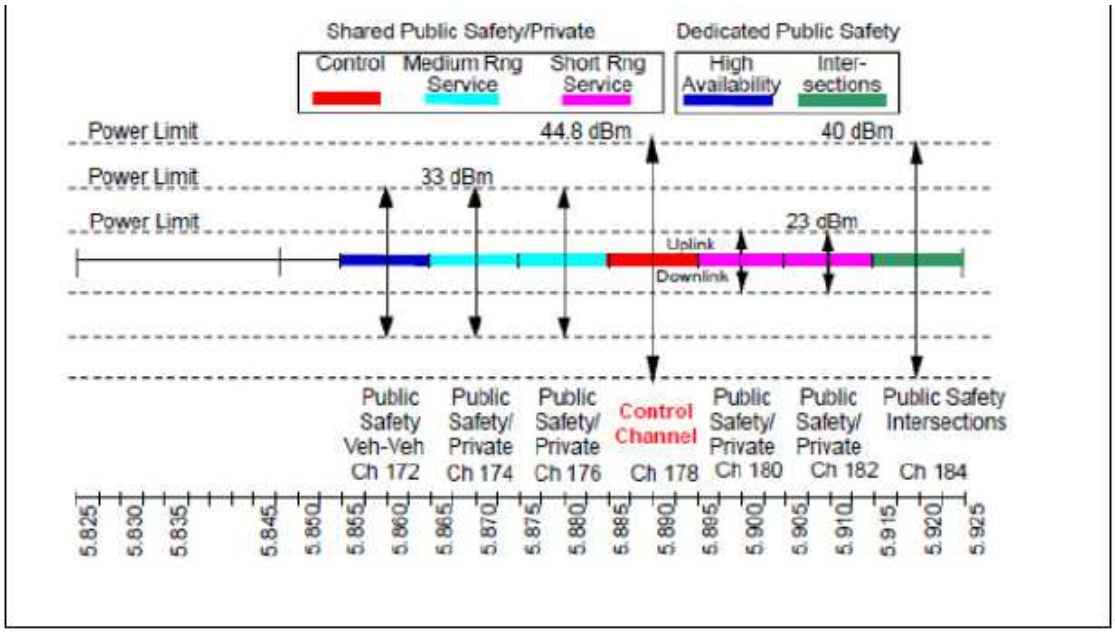

Figura 2: Canali del DSRC [18]

1.5 Protocolli di Routing

Le VANET cercano di estendere la mobilità dei nodi senza far uso di strutture fisse, delegando ai nodi i compiti che normalmente verrebbero svolti dai router (packet forwarding, oltre che semplice routing). Essendo la VANET una rete per definizione mobile, la sua topologia non è certamente prefissata, ma muta al variare dei veicoli che la compongono. Sono stati proposti diversi protocolli di routing:

1. **Topology Based** (in tali protocolli si utilizzano link esistenti tra i nodi per effettuare il routing);
2. **Localization Based** (utilizzano informazioni sulle posizioni geografiche dei nodi per effettuare il routing dei pacchetti).

Un servizio di localizzazione viene normalmente utilizzato dal mittente per determinare la posizione del destinatario. I protocolli basati sulla localizzazione non richiedono quindi salvataggio di tabelle di instradamento.

1.6 Modalità di comunicazione

Un aspetto da tenere in considerazione è la velocità dei veicoli. Esistono diverse situazioni di studio, dal caso in cui i veicoli sono fermi, ad esempio, ad un semaforo, fino al caso in cui i veicoli si muovono a velocità elevata, addirittura anche in sensi opposti, in autostrada. In quest'ultima situazione il tempo a disposizione per lo scambio di messaggi risulta essere particolarmente ridotto. Due automobili che viaggiano in direzione opposta, a 90 Km/h, con un'antenna che ha raggio di copertura pari a 300m, hanno 12 secondi utili per lo scambio di informazioni [8]. Bisogna tenere in conto anche lo scostamento Doppler legato alla mobilità dei veicoli. La frequenza utilizzata per la trasmissione, a causa di questo effetto, si discosta da quella utilizzata per il collegamento di un valore che dipende dalla velocità dei veicoli:

$$\Delta f = \pm f_0 \frac{v}{c}$$

dove f_0 è la frequenza utilizzata per la trasmissione, *v* è la velocità del veicolo, *c* la velocità della luce, e il segno + o - dipende dal fatto che il veicolo si avvicini o si allontani.

Se i veicoli sono pressoché fermi, o si muovono molto lentamente, la topologia della rete è molto stabile (anche se in questo caso bisognerebbe caratterizzare problemi legati all'interferenza, dovuta all'inevitabile aumento della densità veicolare che causa il rallentamento). Diventa utile in questo ambito sfruttare una particolarità delle reti VANET: il vincolo che i veicoli sono obbligati a rimanere nel percorso stradale, eccezion fatta per incroci o rotatorie. Si distinguono quindi tre possibili tipologie di vie: autostrade, strade cittadine e strade rurali.

Le autostrade sono caratterizzate da traffico variabile ad alta velocità, in cui i veicoli seguono un'unica direzione. Questa situazione favorisce la formazione di una rete tra i nodi che viaggiano nello stesso senso di marcia. Le strade cittadine invece sono costituite da un traffico molto intenso, un elevato numero di incroci e possibili cause di percorsi multipli (palazzi). Infine, le strade rurali sono caratterizzate da una bassa densità di automobili, quindi è difficile costituire una rete veicolare.

In quest'ultimo caso, quando i veicoli sono limitati, un messaggio può essere continuamente trasmesso fintanto che non si incontrerà un altro veicolo. Nelle zone urbane invece, a causa dell'alta densità veicolare, tale operazione è da impedire per evitare l'intasamento del canale di comunicazione. Un altro fattore discriminante, tenendo fermo uno dei tre scenari presentati adesso, può essere l'orario della giornata in cui vengono effettuate le comunicazioni. In aree urbane si potrebbe riscontrare una maggiore densità veicolare in determinati momenti della giornata piuttosto che in altri.

Le applicazioni per la sicurezza stradale e la gestione del traffico sono numerose, come ho descritto nel paragrafo precedente, ma le varie caratteristiche appena elencate le rendono difficili da realizzare. Sostanzialmente, si possono individuare alcuni metodi di base per possibili sviluppi futuri, metodi che richiedono comunicazioni broadcast o unicast.

Information Dissemination

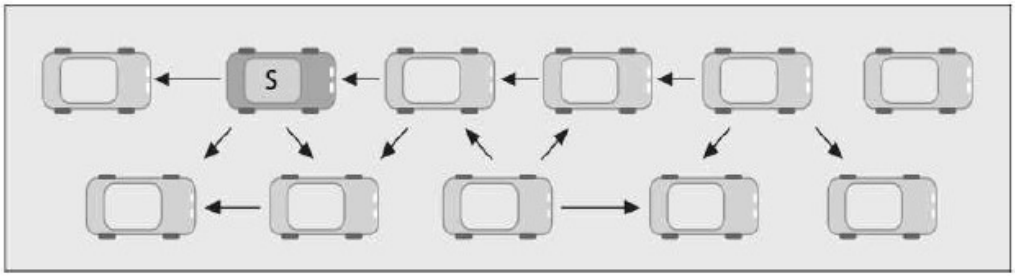

Figura 3: Information Dissemination [7]

Con questo tipo di informazione si mantiene vivo uno scambio di informazioni per diversi periodi di tempo, rendendo tali messaggi disponibili anche ai veicoli che sopraggiungeranno in seguito. Ogni messaggio viene trattato con la tecnica store-and-forward, memorizzandolo in un buffer e inoltrandolo in seguito.

Information Aggregation

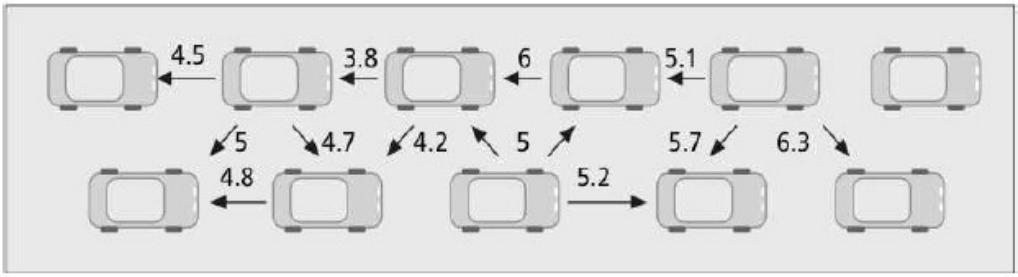

Figura 4: Information Aggregation [7]

I dati provenienti da altri veicoli non vengono memorizzati e inviati, ma elaborati ed aggregati da ogni veicolo che li riceve. Questo metodo ha l'obiettivo di ridurre il sovraccarico del canale trasmissivo quando un evento viene notificato da più di un utente. La collaborazione fa sì che ogni messaggio ricevuto si arricchisce di informazioni aggiuntive ed aggiornate. L'invio di queste informazioni può avvenire a intervalli temporali definiti, o su richiesta. Ovviamente applicazioni sensibili al delay temporale non possono utilizzare tale metodo di comunicazione.

Beaconing

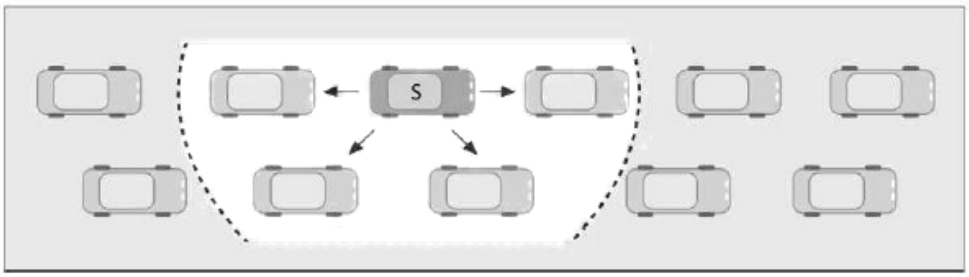

Figura 5: Beaconing [7]

Questa tipologia di messaggi viene inviata da tutti i veicoli nelle vicinanze, ad intervalli temporali predefiniti. Il contenuto di tali messaggi è l'identità del veicolo, la posizione, la velocità, informazioni derivanti da sensori. Prima di inviare nuovi messaggi, ogni veicolo elabora i dati ricevuti da quelli adiacenti, modificando i parametri del messaggio successivo. I beacon vengono immessi nel canale sotto la modalità broadcast, ovvero diretti a tutti, e hanno bassa priorità, essendo messaggi con informazioni aggiuntive. Aumentando la densità veicolare aumenta l'occupazione del canale trasmissivo, quindi i messaggi di sicurezza vengono ostacolati. La potenza di tutti i veicoli deve quindi, per ovviare a ciò, essere regolata in modo da garantire ad ogni mezzo la percezione dei suoi vicini.

Geobroadcast

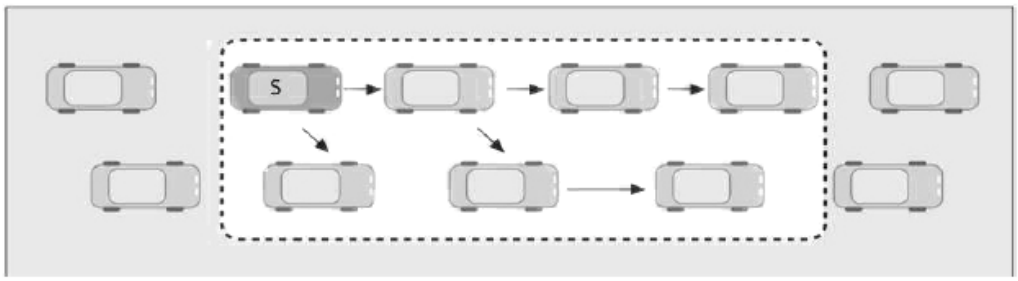

Figura 6: GeoBroadcast [7]

I messaggi di questo tipo vengono inviati agli utenti che si trovano solamente in una determinata area, per metterli al corrente di eventi improvvisi o particolari situazioni. Il mittente sceglie la zona ove inviare le informazioni, inserendo

questo dato nel messaggio che verrà inviato in modalità broadcast. Tutti i ricevitori coinvolti (appartenenti all'area designata), inoltreranno il messaggio.

Unicast

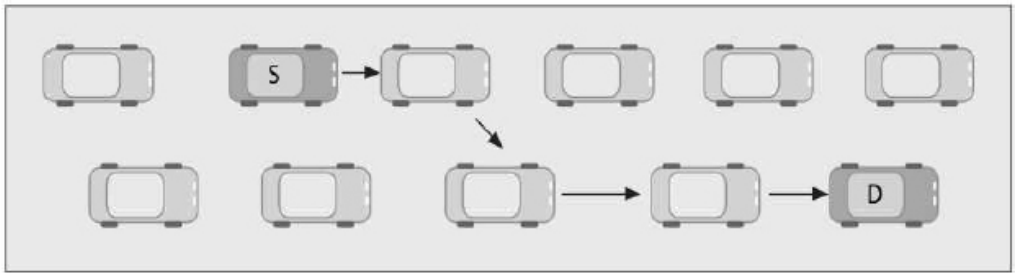

Figura 7: Unicast [7]

Questi messaggi sono destinati a un unico ricevitore, l'obiettivo è instaurare un collegamento punto-punto tra due veicoli. Il mittente deve conoscere la posizione del destinatario. Ogni auto dovrà mantenere in memoria una tabella dei vicini, registrando di volta in volta le identità dei veicoli che *percepisce*. Tali tabelle saranno aggiornate di continuo in base agli aggiornamenti ricevuti dai messaggi beacon.

2. STANDARD 802.11p WAVE

Il processo di standardizzazione dell'IEEE 802.11p WAVE ha origine dall'intento di definire la banda da allocare e la tecnologia da utilizzare per il Dedicated Short Range Communications (DSRC). Nel 1999 la Federal Communication Commission decise di allocare negli Stati Uniti una banda pari a 75 MHz nell'intorno dei 5.9 GHz da usare esclusivamente per la comunicazione veicolare. L'obiettivo principale era ovviamente aumentare la sicurezza stradale. Come anticipato nel capitolo precedente, lo spettro DSRC è suddiviso in 10 canali da 7 MHz ciascuno. Come si può vedere dalla figura 2, il canale 178 (CCH, Control Channel) è riservato solamente alle comunicazioni di sicurezza. La banda utilizzata dal DSRC è libera ma licenziata. La Federal Communication Commission ha definito anche delle regole riguardo lo sviluppo di altre tecnologie radio da utilizzare nella banda del DSRC, limitando la potenza in emissione a 33 dBm.

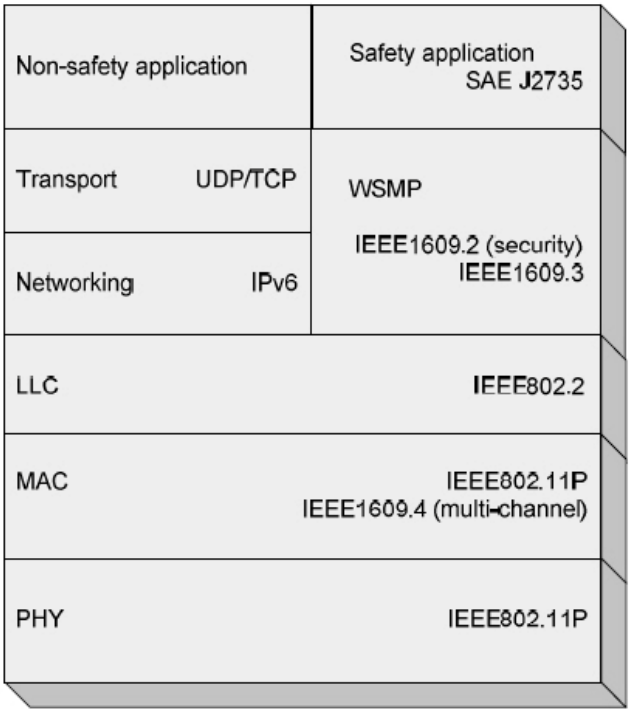

Figura 8: Stack protocollare del WAVE[5]

In figura 8 è mostrato lo stack protocollare del DSRC WAVE, come si può notare, l'IEEE 802.11p si occupa del livello MAC e FISICO.

2.1 Livello Fisico

Ci sono parecchi vantaggi nell'utilizzare lo standard 802.11. E' innanzitutto uno standard stabile, e la stabilità è un requisito fondamentale per garantire l'interoperabilità tra veicoli prodotti da diverse aziende costruttrici. Tuttavia, è richiesta una versione differente rispetto all'802.11 perché bisogna supportare un lungo raggio di operazione, tener conto dell'alta velocità dei veicoli e caratterizzare un ambiente affetto da percorsi multipli. La versione 802.11p implementa funzionalità che garantiscono le comunicazioni in un ambiente altamente mobile, utilizzando canali con ampiezza di banda pari a 10 MHz, un canale di controllo e sei canali di servizio, in aggiunta anche i controlli di potenza. Lo standard 802.11p, come si può vedere in figura 8, opera insieme ai protocolli IEEE 1609 che definiscono architettura, modelli di comunicazione, protocolli di sicurezza e accesso fisico al mezzo radio [6]. Le modifiche a livello fisico rispetto alle altre versioni dell'IEEE 802.11 riguardano una traslazione della frequenza di lavoro, dai 5 GHz dell'IEEE 802.11a fino ai 5.9 GHz dell'IEEE 802.11p. Quest'ultimo è basato su una modulazione OFDM (Orthogonal Frequency Division Multiplexing), utilizzando 64 sotto-portanti di cui 48 sono utilizzate per i dati, 4 sono sotto-portanti pilota, e le restanti 12 sono virtuali [7]. I canali hanno larghezza di banda pari a 10 MHz (a differenza dei 20 MHz dell'802.11). In aggiunta, l'IEEE 802.11p richiede una rapida decaduta dello spettro per ridurre le interferenze con i canali adiacenti. Nella tabella in basso è possibile notare le differenze tra i parametri della modulazione OFDM nel caso WAVE e nel caso dell'IEEE 802.11.

PARAMETRI	IEEE 802.11p	IEEE 802.11 a/g
Frequenza della portante	5.9 GHz	5 GHz / 2.4 GHz
Ampiezza di banda dei canali	10 MHz	20 MHz
Data Rate supportata (Mbps)	3, 4.5, 6, 9, 12, 18, 24, 27	6, 9, 12, 18, 24, 36, 48, 54
Modulazioni	BPSK, QPSK, 16QAM, 64QAM	BPSK, QPSK, 16QAM, 64QAM
Sottoportanti di dati	48	48
Sottoportanti virtuali	12	12
Sottoportanti pilota	4	4
Δf tra le sottoportanti	0.15625 MHz	0.3125 MHz
Intervallo di un simbolo OFDM	8μs	4μs

Tabella 1

La tabella è stata riformulata da [5].

2.2 Livello Mac

I cambiamenti a livello MAC (rispetto all'802.11) richiedono solamente un aggiornamento del software, garantendo connessioni in modo rapido [6]. Nelle precedenti versioni IEEE 802.11 sono definiti tre tipi di Service Set: Basic Service Set (BSS), Independent BSS (IBSS), Extended BSS (EBSS). Un BSS include gli Access Point (AP), mentre gli ESS sono le unioni di uno o più BSS connessi da un servizio di distribuzione (DS).

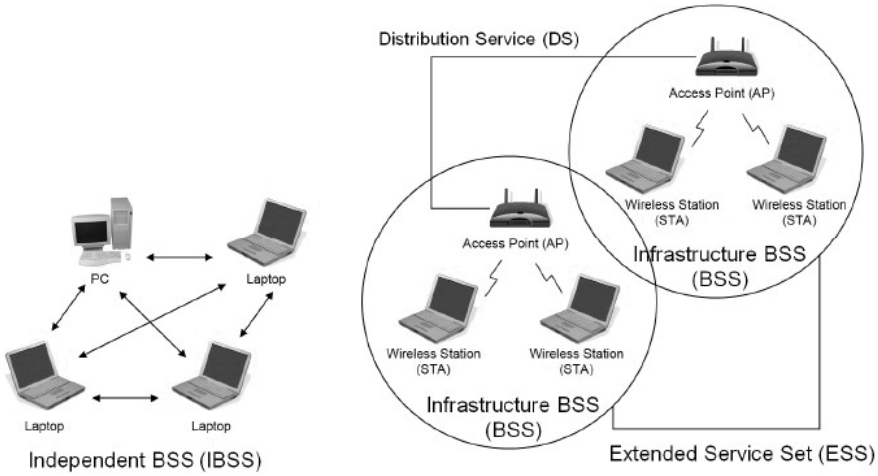

Figura 9: BSS tratta da [4]

Nel BSS, una stazione (STA) invia in broadcast, periodicamente, dei messaggi (beacon) contenenti, tra le varie informazioni, il Service Set ID (SSID). Le altre stazioni all'interno del BSS ricevono il beacon e sincronizzano tempo e frequenza con quelli indicati nel beacon inviato. Le stazioni possono però comunicare le une con le altre solo ed esclusivamente se hanno lo stesso SSID. Quest'architettura può essere utilizzata anche nello standard 802.11p, tuttavia, la creazione di un BSS richiede operazioni (invio dei messaggi, sincronizzazione…) molto lunghe, rispetto ai tempi stringenti richiesti dalla sicurezza stradale. Due veicoli in movimento potrebbero essere rispettivamente visibili per un intervallo di tempo pari a meno di un secondo. Per ovviare a tali limitazioni, è stata introdotta una modifica, con l'introduzione di un nuovo tipo di BSS, il WBSS (Wave BSS). Una stazione crea un WBSS trasmettendo all'inizio un beacon di richiesta. Tale segnale contiene tutte le informazioni necessarie affinché le stazioni riceventi capiscano i servizi offerti in quel WBSS. Questo approccio offre la possibilità di ridurre notevolmente l'overhead rispetto a quello necessario per la creazione di un BSS, eliminando tutti i processi di autentificazione e associazione. Una qualunque stazione radio può entrare a far parte del WBSS utilizzando il BSSID (Basic Service Set IDentification). Prima della modifica WAVE, all'interno del BSS potevano comunicare tra di loro solamente le stazioni che avevano uguale BSSID. Adesso, utilizzando il wildcard BSSID, una stazione può comunicare con le altre anche se non fa parte

della WBSS [6]. Una stazione in un WBSS è ancora in modalità WAVE e può quindi trasmettere o ricevere pacchetti utilizzando appunto tale wildcard BSSID.

3. OFFLOADING DI RETI CELLULARI

Con il crescente numero di smartphones in circolazione, le reti 3G sono sempre più sovraccariche. Secondo [11], il traffico telefonico negli ultimi tre anni è aumentato del 5000%. A causa di questa esplosione di traffico telefonico, ben presto è sorto il problema di garantire una discreta qualità di servizio agli utenti. Una possibile soluzione è l'utilizzo di femtocelle o tecnologia Wi-Fi in aggiunta alla rete cellulare.

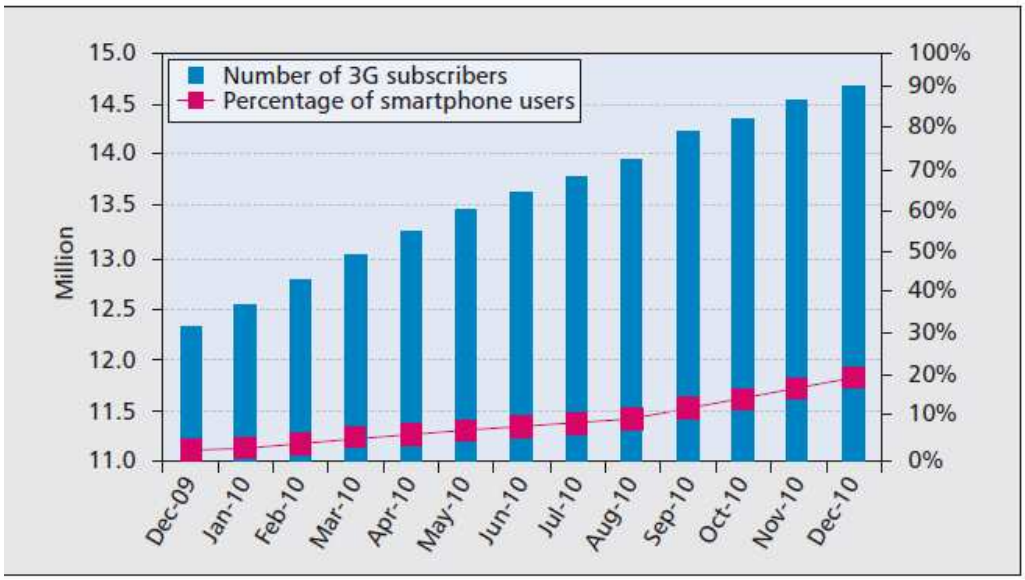

Figura 10: Utenti 3G e Utenti Smartphone [15]

In figura 10 è visualizzato il grafico relativo all'aumento degli utenti che utilizzano il 3G rispetto a quelli che utilizzano gli smartphones (relativo all'anno 2010). In figura 11 invece, sempre relativo a quell'anno, il volume del traffico per rete.

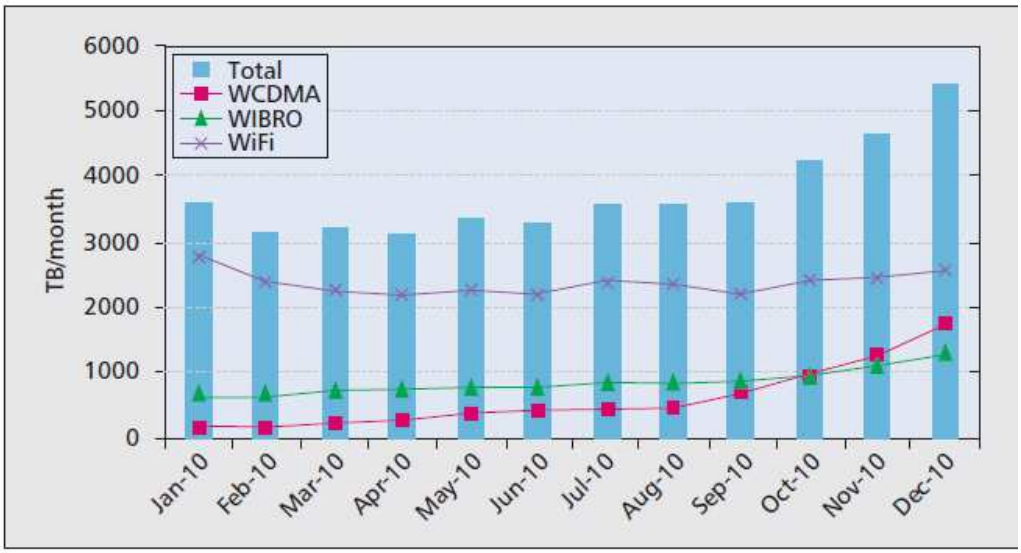

Figura 11: Traffico per rete [15]

Nelle figure sottostanti invece, si può notare (i grafici risalgono sempre all'anno 2010) il numero di utenti che hanno sottoscritto un abbonamento internet (attraverso la relativa rete), e il traffico mensile per utente.

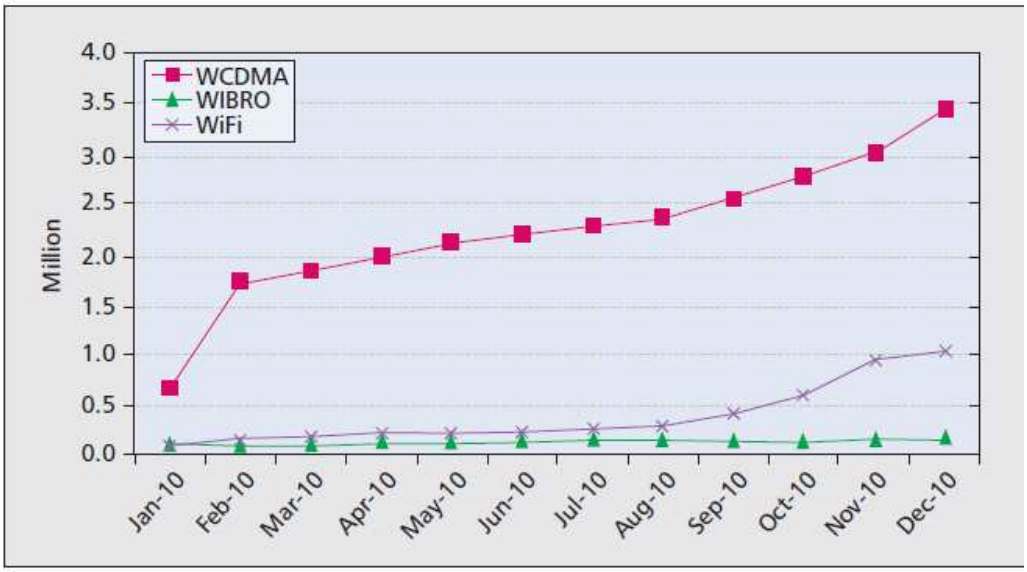

Figura 12: Numero di utenti che hanno stipulato un abbonamento internet [15]

Figura 13: Traffico mensile per utente [15]

Per *offloading della rete cellulare* si intende quindi l'utilizzo di tecnologie di altre reti di comunicazione (WI-FI, VANET…) per recapitare a destinazione dati che originariamente furono programmati per essere inviati tramite rete cellulare. Una tra le possibili soluzioni per ridurre il carico della rete cellulare è l'offloading su femtocelle, che sono state proposte per migliorare il servizio in un ambiente indoor. Per utenti quindi all'interno delle mura domestiche, gli operatori di telefonia mobile possono decidere di effettuare l'offloading del traffico dalle macrocelle alle celle più piccole. Un'alternativa sarebbe data dall'offloading su rete Wi-Fi (che lavora su bande di frequenza non licenziate e che quindi non interferisce con la rete cellulare).

Parlando dell'aumento del numero degli smartphone, non si può non sottolineare un aspetto di cui si potrebbe trarre beneficio, ovvero le multiple interfacce radio. Per fare offloading su reti Wi-Fi, ad esempio, il dispositivo mobile invierà diversi frammenti di dati alle stazioni radio-base associate, includendo nel messaggio informazioni aggiuntive quali la potenza del segnale, la posizione geografica dell'utente, la velocità di movimento e la direzione. Ovviamente è richiesto che le stazioni radio-base siano a conoscenza degli Access Point (AP) utilizzati per fare offloading. Basandosi su questi dati, la stazione radio-base

potrà decidere se inviare i dati attraverso rete cellulare, oppure tramite offloading su rete Wi-Fi.

3.1 Offloading tramite Wi-Fi

Wi-Fi (Wireless Fidelity) è una tecnologia basata sull' IEEE 802.11. Confrontata con altre tecnologie, come LTE o HSPA, Wi-Fi offre una maggiore bit-rate, con limitata copertura e possibilità di movimento. Si presenta come soluzione naturale per l'offloading, grazie anche alla capacità degli smartphones di utilizzare tale tecnologia. A causa del degrado della Quality of Service, dovuta al largo numero di apparati mobili che utilizzano la rete cellulare, molti utenti si vedono costretti a dover utilizzare la connessione Wi-Fi per accedere al web. Un ulteriore punto a favore di tale tecnologia è la possibilità di trasferire traffico utilizzando le bande non licenziate (2,4 GHz), con conseguente risparmio economico da parte dei providers. Secondo [12] esistono tre diversi approcci per eseguire l'offload su reti Wi-Fi, a seconda del livello di integrazione tra le due reti. Il primo metodo è il *network bypass*, in cui il traffico dati è *spostato*, in maniera totalmente trasparente, dalla rete cellulare alla rete Wi-Fi, qualora l'utente si trovi nel raggio di copertura Wi-Fi. Questo metodo presenta tuttavia alcuni lati negativi, ad esempio, il provider perde la visibilità dell'utente quando questi si trova sotto copertura Wi-Fi. Nonostante gli svantaggi, questo metodo può essere una soluzione immediata da utilizzare per l'offloading, grazie alla sua semplicità realizzativa.

Un secondo metodo è il *managed data offloading*, adottabile dai providers che non vogliono perdere la visibilità dei propri utenti. In questo caso non è richiesta la completa integrazione tra le due reti.

Infine, il terzo metodo è il *integrated data offloading*, che dà al provider pieno controllo sui propri abbonati, offrendo anche la possibilità di recapitare qualsiasi tipo di contenuto in abbonamento. Ciò è reso possibile dall'integrazione tra la rete cellulare e quella Wi-Fi, facendo sì che possa crearsi un ponte tra le due reti, per poter effettuare l'offloading.

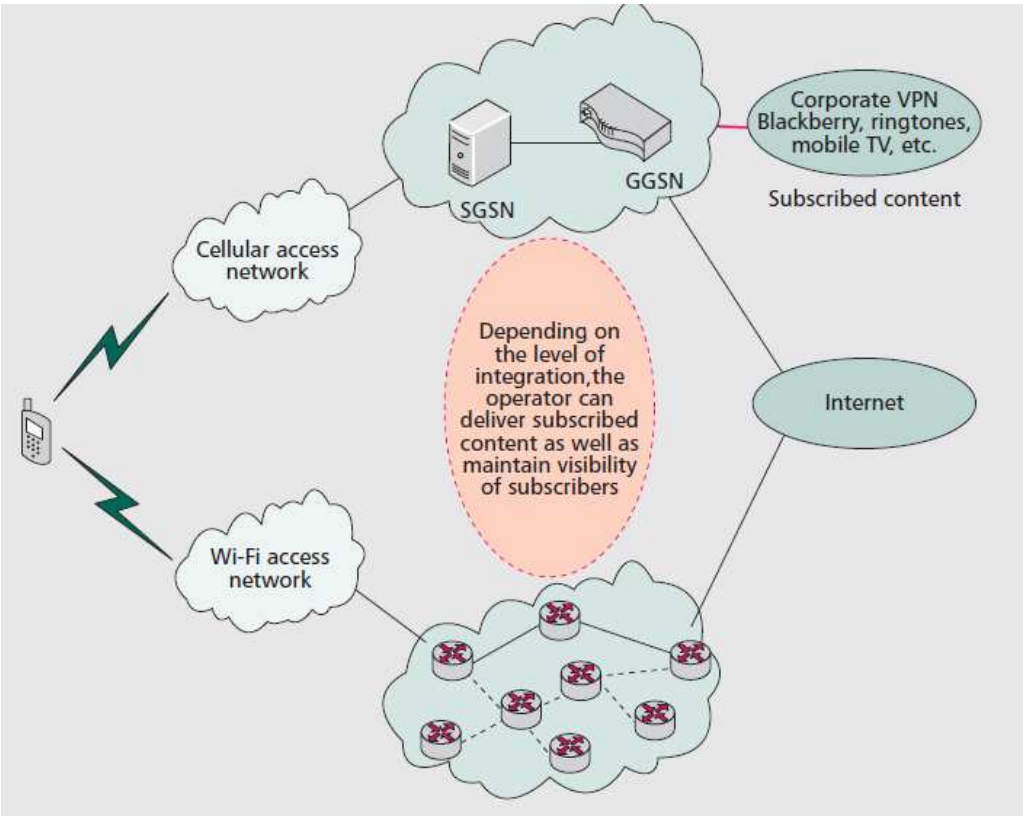

Figura 14: Offloading via Wi-Fi [21]

Per incrementare l'efficienza dell'offloading, gli Access Point (AP) devono essere installati nei luoghi di maggior richiesta di traffico telefonico (un esempio fra tanti, i centri commerciali). Considerando un'ampia zona cittadina, bisogna tenere in considerazione che gli utenti che fanno richiesta di traffico dati possono muoversi, il punto da cui è partita la richiesta potrebbe essere totalmente differente da quello in cui attualmente si trova l'utente. Inoltre, l'accesso al web potrebbe anche avvenire solamente in certe fasce orarie.

3.2 Offloading tramite femtocelle

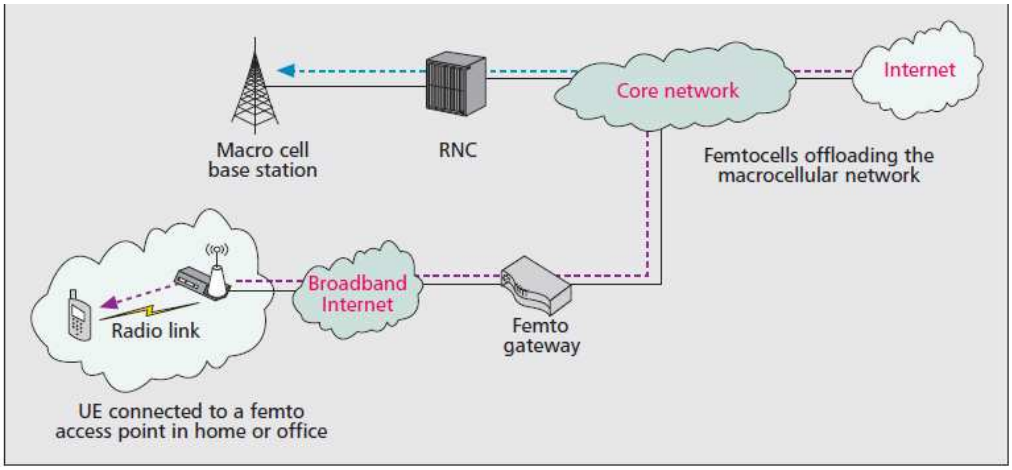

Figura 15: Offloading tramite femtocelle [21]

Una femtocella è una piccolo stazione radio-base, utilizzata principalmente per usi indoor (casa o ufficio). Connette con la rete del providers e permette a tali providers di estendere la copertura del servizio, specialmente in aree in cui gli accessi sarebbero altrimenti limitati o non disponibili. Le femtocelle sono applicabili a tutti gli standard, GSM, CDMA, WIMAX ed LTE; inoltre forniscono un metodo efficace per facilitare lo smaltimento del traffico cellulare. Fare offloading tramite femtocelle può rivelarsi efficace per diversi motivi, tra i quali:

1) sono utilizzate in ambienti indoor (secondo uno studio, il 55% del traffico dati viene smaltito in ambiente domestico, il 26% invece in ufficio) [14].

2) sono facilmente installabili, al contrario delle macro-celle.

Il traffico passa dall'interfaccia radio del cellulare alle femtocelle, poi attraverso il web fino alla rete dell'operatore (o a uno specifico sito web).

Passo ad analizzare adesso gli svantaggi e i vantaggi offerti da Wi-Fi e dalle femtocelle.

Il Wi-Fi lavora, come detto in precedenza, su bande non licenziate. I provider hanno la possibilità di utilizzare una porzione maggiore di spettro. Le femtocelle invece richiedono un'accurata pianificazione, e lavorano su bande di frequenza licenziate. Tuttavia le femtocelle, a differenza del Wi-Fi, catturano la totalità del traffico cellulare, che sia esso voce o dati. Le femtocelle inoltre non influiscono sulla durata della batteria del terminale mobile, a differenza delle

apparecchiature Wi-Fi. Parlando invece di bit-rate, ovviamente il Wi-Fi permette una maggiore velocità di trasmissione di dati rispetto alle femtocelle. Questo aspetto non è da sottovalutare, in quanto molte volte gli utenti scaricano dal web file anche di grosse dimensioni.

3.3 Offloading tramite comunicazioni opportunistiche

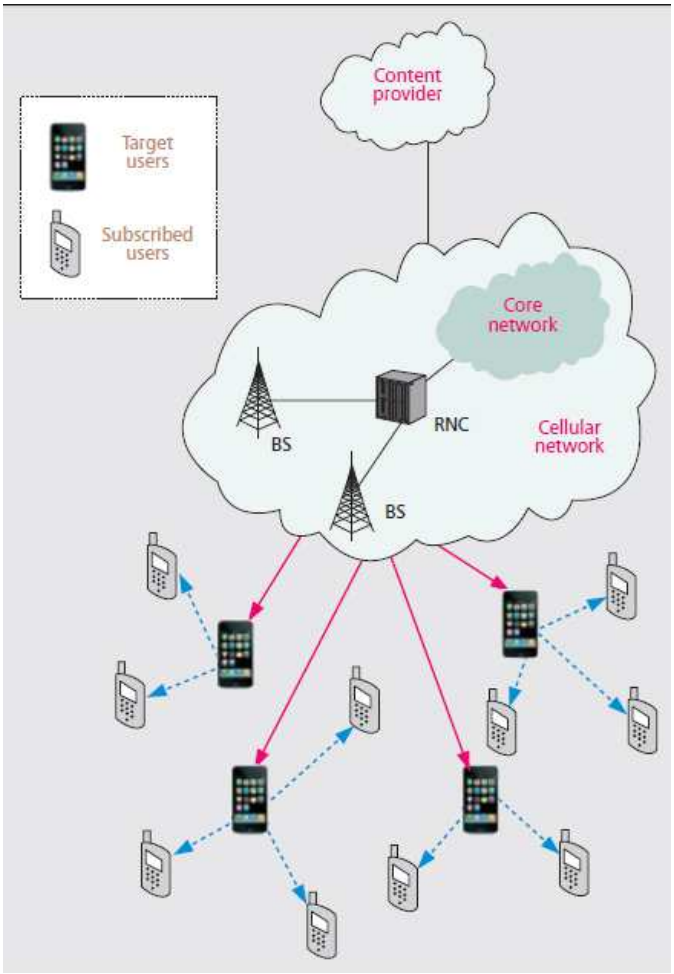

Figura 16: Offloading tramite comunicazioni opportunistiche [21]

Un ulteriore approccio all'offloading è dato dalle comunicazioni opportunistiche. Buona parte del traffico consegnato tramite rete cellulare proviene da provider di servizi, e può includere le varie applicazioni disponibili oggigiorno sul mercato, come giochi, multimedialità e informazioni in tempo reale. Queste applicazioni sono insensibili ai ritardi, e il provider può trarre

beneficio da ciò, inviando l'informazione solamente a un determinato gruppo di utenti. Tale gruppo poi diffonderà l'informazione agli utenti che ne hanno fatto richiesta, quando il loro terminale si troverà in prossimità e quindi in grado di comunicare.

3.4 Offloading tramite DSRC

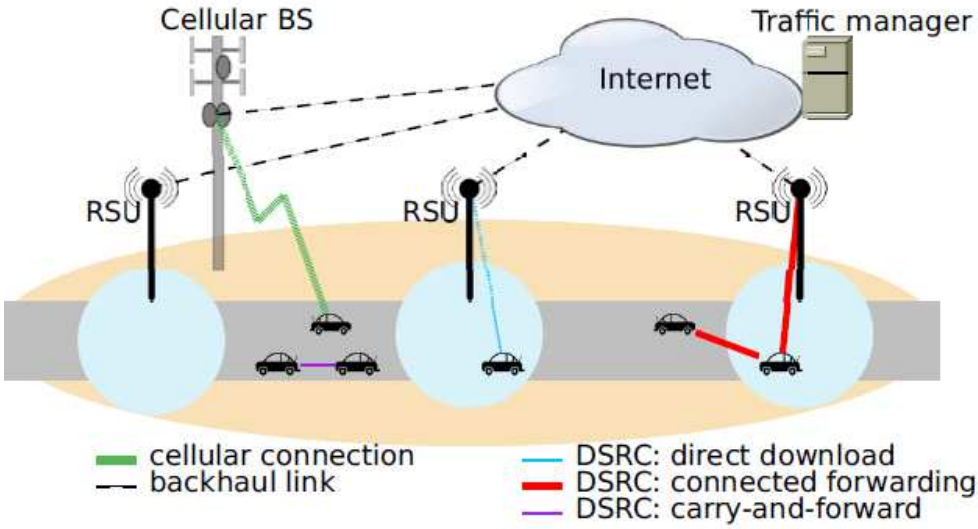

Figura 17: Rete [17]

Come si può vedere in figura 17, consideriamo una rete composta utenti in movimento (quindi le automobili, con le OBU a bordo), e infrastrutture fisse (le RSU ai margini della carreggiata), in uno scenario che comprende anche copertura di rete cellulare. Si suppone inoltre che i veicoli abbiano un'unica interfaccia DSRC, e che comunicazioni veicolo-veicolo e veicolo-infrastruttura avvengano su frequenze differenti. Gli utenti mobili potrebbero decidere in qualsiasi istante di scaricare contenuti dal web. Dando per scontato che i veicoli siano dotati di interfacce che garantiscono sia la comunicazione V2I, che la V2V, sia la comunicazione su rete cellulare, è possibile sviluppare diversi algoritmi di trasferimento dati. Questi utenti possono essere quindi in grado di scaricare tali contenuti o dalla RSU ai margini della carreggiata, oppure possono decidere di farsi assistere dai veicoli che compongono la rete VANET per assemblare i frammenti di dati ricevuti e ricreare il file originale (quindi utilizzando algoritmi

di multi-hop, carry and forward). Un utente che desidererà scaricare un qualsiasi tipo di dato dal web, genererà una richiesta ad un sistema di gestione, tramite RSU o rete cellulare. Alla RSU spetta poi il compito di fornire correttamente i dati per il download. Le Road Side Unit saranno quindi in grado di scaricare il contenuto richiesto dai server che lo conservano, e consegnarlo direttamente all'utente che ne ha fatto richiesta.

4. GCDC, OPENWRT E LE SCHEDE ALIX

In laboratorio, per implementare lo standard IEEE 802.11p, sono state utilizzate le schede ALIX, della PCEngines, schede a basso costo, quindi con buona scalabilità. Il firmware utilizzato per gestirle si chiama OpenWrt, versione di linux adattata per questi scopi. In aggiunta a tale firmware, il software java utilizzato per la ricezione e la trasmissione dei pacchetti è stato fornito dal GCDC (Grand Cooperative Driving Challenge).

4.1 GCDC: Grand Cooperative Driving Challenge

Il GCDC è una competizione a livello mondiale, che interessa team di diverse nazioni, che si ripropone di fornire un sistema in grado di far cooperare i veicoli e le infrastrutture. Il GCDC mira quindi ad accelerare l'attuazione di tali sistemi, in modo da dare un contributo significativo al miglioramento delle condizioni di viaggio degli automobilisti, così come a unire mondo accademico e realtà industriale, per far sì che possano realizzare veicoli capaci di comunicare tra di loro. Tale organizzazione ha anche fissato delle regole da seguire, in modo da garantire l'interazione veicolare anche nel caso in cui le automobili fossero state costruite da diverse aziende automobilistiche.

Per i primi due livelli dello stack protocollare si decise di utilizzare lo standard IEEE 802.11p, mentre per i livelli superiori furono avviati alcuni progetti dai gruppi di studio CVIS e SAFESPOT, i cui risultati vennero utilizzati per GCDC. Il problema fu che CVIS decise di utilizzare hardware proprietario, quindi non garantiva il carattere open source del progetto. Alcune novità cominciarono a presentarsi quando Eric Koenders decise di collaborare a tale progetto. Fu creata quindi una distribuzione linux IEEE 802.11p, totalmente gratuita ed open source, costituita da un Daemon, una versione di OpenWrt, e il programma java che avrebbe garantito, come detto in precedenza, l'invio e la ricezione dei pacchetti.

4.1.1 Stack Protocollare

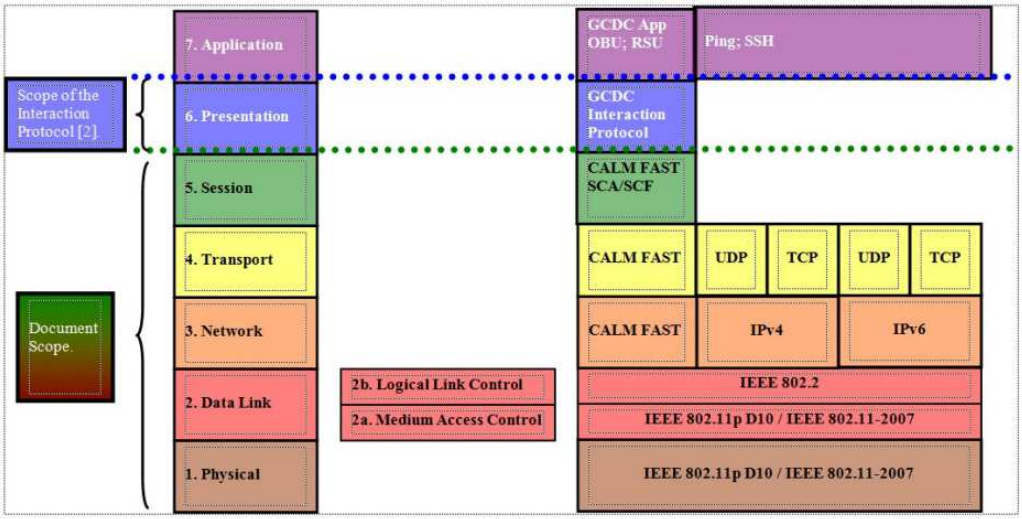

Figura 18: Stack Protocollare del GCDC [9]

Lo strato fisico e di collegamento sono gestiti dall'IEEE 802.11p. Nel caso in cui un veicolo debba utilizzare un canale diverso di comunicazione (rispetto al canale CCH di controllo), quindi si venga a trovare nella necessità di utilizzare un canale SCH di servizio, il cambio di banda avviene prima dell'instaurazione di una nuova comunicazione. Il codice sorgente dei moduli (Daemon, codice Java e OpenWrt) è composto da:

1. Librerie CALM-FAST: libfast, libcalmmedia, libcalmfast, libcam;
2. I moduli del kernel aggiornati in modo da poter supportare lo standard IEEE 802.11p, ovvero i driver ath5k, mac80211e wireless;
3. Il software CALMFAST;
4. Un aggiornamento dell'user-space "IW", utile per configurare i driver wireless modificati.

L'implementazione del CALMFAST segue le specifiche dello standard ISO/DIS 29281, che definisce lo scambio di messaggi tra router ed host. Durante la trasmissione, i messaggi dovranno essere instradati dal router, che utilizzerà una tabella di routing. Tale modalità presenta dei problemi, in quanto sono necessari dei tools di supporto per configurare la rete wireless. Il protocollo FAST è implementato quindi in un DAEMON, che comunica con il driver wireless tramite una raw socket.

4.2 Schede Alix

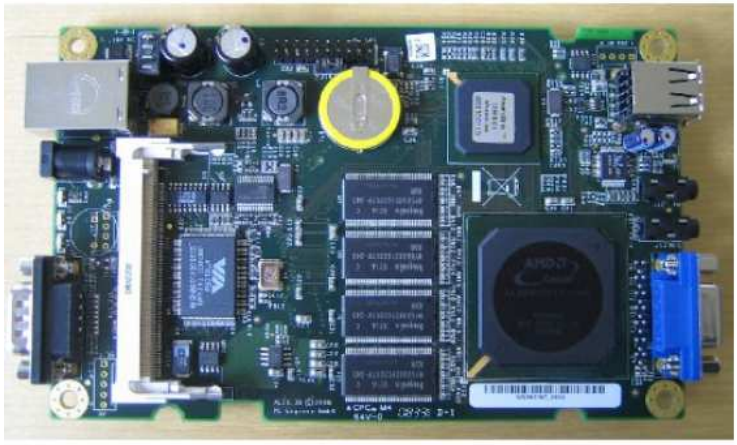

Figura 19: Scheda Alix (1)

Figura 20: Scheda Alix (2)

Figura 21. Scheda Alix (3)

- Chipset: Atheros 5414
- DRAM: 256 MB DDR DRAM
- Storage: CompactFlash socket
- Power: DC jack or passive POE, min. 7V to max. 20V
- Three LEDs
- Expansion: 2 miniPCI slots, LPC bus
- Connectivity: 1 Ethernet channel (Via VT6105M 10/100)
- I/O: DB9 serial port, dual USB, VGA, audio headphone out / microphone in
- RTC battery
- Board size: 100 x 160 mm
- Firmware: Award BIOS

Figura 22: Scheda tecnica della Alix [6]

In figura 19 si può vedere uno dei due lati della scheda Alix, mentre in figura 23 è visualizzato il lato opposto. Come si può notare in figura 19, lo scompartimento a sinistra è utilizzato per la scheda wireless, mentre la flash card contenente il sistema operativo, più un'ulteriore (opzionale) scheda wireless possono essere inserite nei due scomparti mostrati in figura 20. In figura 22 sono invece mostrate le caratteristiche della scheda [6]. Il consumo di potenza della ALIX è ridotto a

5W. Di seguito sono mostrate le immagini relative alla prima scheda wireless e le sue caratteristiche tecniche.

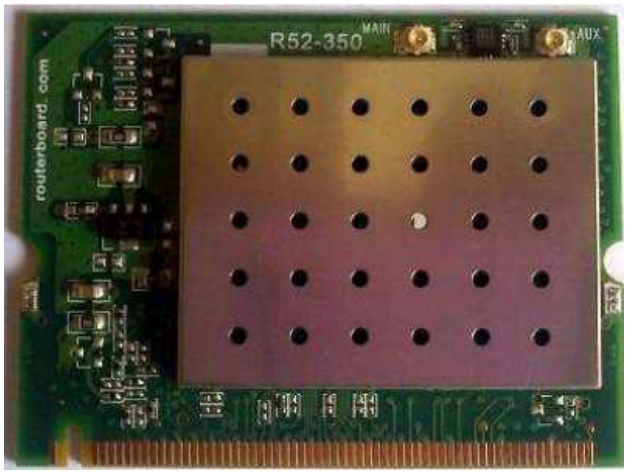

Figura 23: Scheda wireless della Alix

Chipset:	Atheros AR5414
Standards:	IEEE802.11a, IEEE802.11b, IEEE802.11g
Media Access:	CSMA/CA with ACK architecture 32-bit MAC
Security:	Hardware 64/128 bit WEP, TKIP and AES-CCM encryption, WPA, WPA2, 802.1x
Modulation:	802.11b+g: DSSS, OFDM for data rate >30Mbps
	802.11a: OFDM
Connectors:	Two U.fl connectors
Certifications:	FCC, EC
Powering:	3.3V +/- 10% DC; 800mA max (600mA typ.)
Frequencies: 802.11b/g	2.192 – 2.507 (5 MHz step); 2.224 – 2.539 (5MHz step)
Frequencies: 802.11a	4.920 – 6.100 (5 MHz step)
Operating temperature	-20 .. +70C
Standard	Output power / Receive Sensitivity
802.11a	17dBm/-88dBm @ 6Mbps
	13dBm/-71dBm @ 54Mbps
802.11b	19dBm/-95dBm @ 1Mbps
	19dBm/-90dBm @ 11Mbps
802.11g	18dBm/-90dBm @ 6Mbps
	15dBm/-73dBm @ 54Mbps

Figura 24: Riassunto tecnico

In figura 25 e 26 sono mostrate invece l'immagine e le caratteristiche tecniche della seconda scheda wireless.

Figura 25: Seconda scheda wireless

Chipset	Atheros AR5414A-B2B
Frequency Range	5.85 – 5.925GHz
Channel Bandwidth	40MHz, 20MHz, 10MHz, and 5MHz
Frequency Tolerance	transmitted center frequency deviation ± 10ppm max.
Interface	32-bit mini-PCI Type IIIA
Operation Voltage	3.3 ± 5% VDC
Modulation Technique	OFDM with BPSK, QPSK, 16-QAM, and 64-QAM
Data Rate	• 20MHz BW: 6, 9, 12, 18, 24, 36, 48, 54Mbps • 10MHz BW: 3, 4.5, 6, 9, 12, 18, 24, 27Mbps • 5MHz BW: 1.5, 2.75, 3, 4.5, 6, 9, 12, 13.5Mbps
Transmit Power	

	5860MHz Ave./Peak	5920MHz Ave./Peak
1.5/3/6Mbps	24/29dBm	24/29dBm
2.75/4.5/9Mbps	24/29dBm	24/29dBm
3/6/12Mbps	24/29dBm	24/29dBm
4.5/9/18Mbps	24/29dBm	24/29dBm
6/12/24Mbps	24/29dBm	24/29dBm
9/18/36Mbps	23/28dBm	23/28dBm
12/24/48Mbps	20.5/25.5dBm	20.5/25.5dBm
13.5/27/54Mbps	19/24dBm	18.5/23.5dBm

Remark: The default transmit power setting of DCMA-86P2 is as above. If ASTM E2213-03 Clause 8.9.1.10 Class-C power mask conformance required, please make sure to change the power setting of DCMA-86P2 to be lower than 20dBm.

Figura 26: Riassunto tecnico

La scheda funziona grazie a una versione del software OpenWrt (un insieme di Makefiles utili alla compilazione del software che farà funzionare il sistema) che viene installato nella compact flash da 4 GB.

4.3 Configurazione della connessione

Per poter comunicare con le schede Alix è necessario eseguire alcune procedure preliminari per configurare la connessione ai dispositivi. Per prima cosa bisogna collegare il dispositivo al computer tramite cavo ethernet, in seguito impostare i seguenti valori del protocollo IPv4 (la procedura per Windows7 è questa: click destro sull'icona della connessione > Apri centro connessioni di rete e condivisione > Modifica Impostazioni scheda > Selezionare la connessione lan e con il tasto destro cliccare su Proprietà > Scorrere il menu a tendina fino alla voce Protocollo Internet versione 4 (TCP/IPv4) e cliccare, dopo averla selezionata, su Proprietà > cliccare sulle voci "Utilizza il seguente indirizzo ip" e "Utilizza i seguenti indirizzi server DNS" per poter inserire i valori dell'immagine seguente).

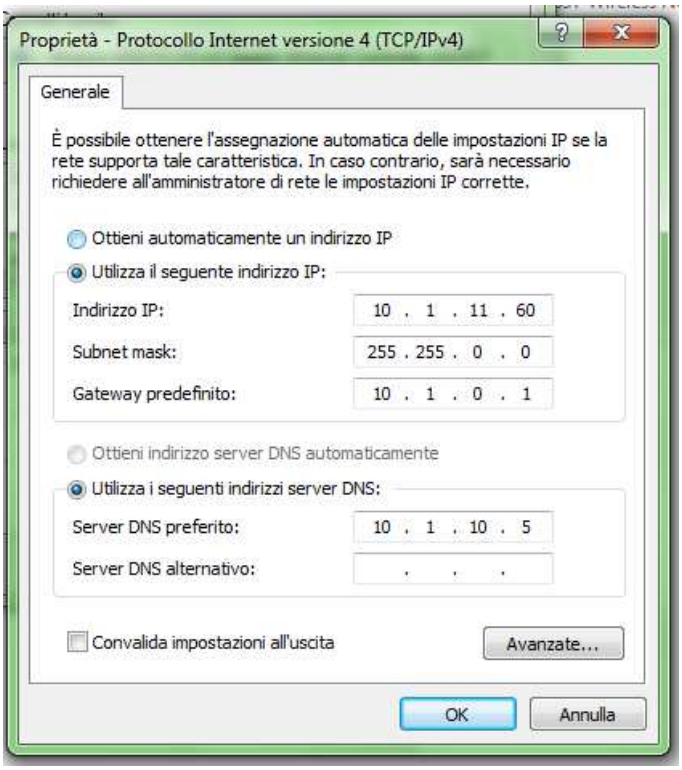

Figura 27: Parametri di Connessione

Una volta convalidata la scelta, bisogna modificare il file host del proprio computer. Il percorso per raggiungere tale file è *C:|windows|system32\drivers\etc\hosts.* Copiare il file host sul desktop, aprirlo con il blocco note e inserire le seguenti righe in fondo al file:

127.0.0.1 localhost

::1 localhost

10.1.6.ZZ wilabalixZZ

dove, nell'ultima delle tre righe, ZZ sta per il numero della Alix con cui ci si vuole connettere (quindi, se ad esempio si vuole utilizzare la Alix 45, bisogna scrivere 10.1.6.45 wilabalix45). Tale riga va replicata (cambiando il numero finale) per ogni dispositivo che si vuole utilizzare. A questo punto è possibile scaricare il programma Putty e connettersi al dispositivo.

4.4 Putty

Nella figura sottostante è mostrata l'interfaccia del software.

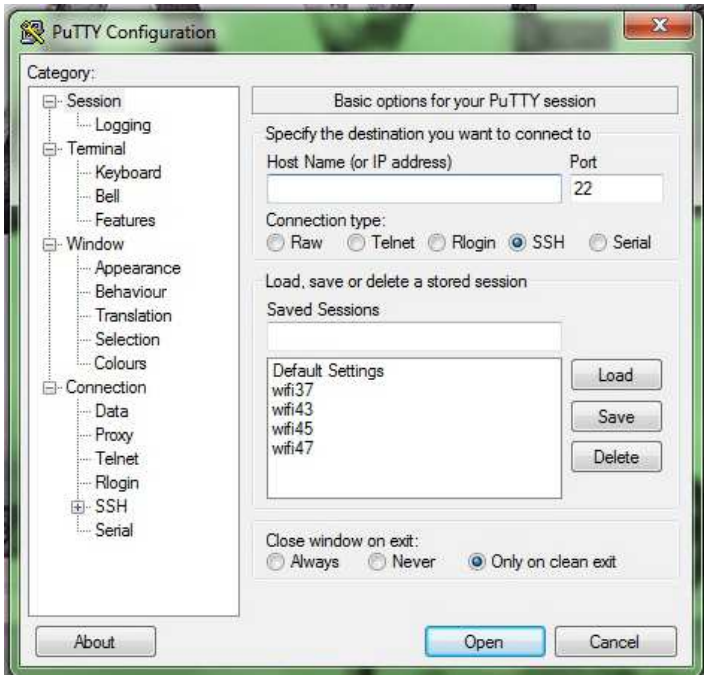

Figura 28: Putty

Inserire nel campo Host Name l'indirizzo Ip del dispositivo (come porta utilizzare la 22). Per ogni dispositivo è inoltre possibile salvare i parametri di

ingresso (memorizzati, come visibile, nei file Wi-Fi37 per il dispositivo 37, Wi-Fi43 per il dispositivo 43 ecc.). Cliccare su Open. Digitare "root" come nome utente, e "gcdc2011" come password.

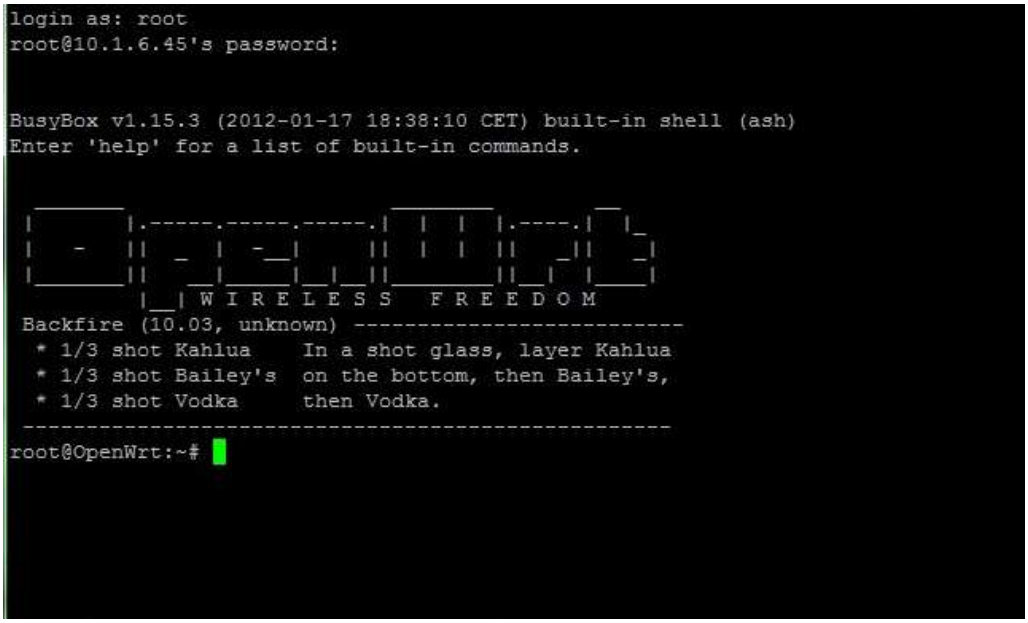

Figura 29: OpenWrt

In figura 29 è mostrata l'interfaccia di OpenWrt una volta entrati nel dispositivo. Da terminale è possibile adesso utilizzare i comandi del sistema operativo.

I principali comandi utilizzabili sono:

1. *iwconfig*, con il quale si ottengono le informazioni sulle interfacce wireless della scheda;
2. *ifconfig*, che permette di ottenere le informazioni su tutte le interfacce della scheda;
3. *vi*, seguito dal nome del file, che permette di modificare tale file.

Inoltre, tramite un tool di OpenWrt chiamato Iperf, è possibile inviare dati UDP o TCP per misurare il throughput del collegamento.

Per configurare bene le schede, bisogna modificare un paio di files all'interno di Openwrt: il file network, e quello wireless (il primo permette di definire indirizzo ip, netmask e gateway, il secondo consente di modificare l'indirizzo mac della scheda wireless).

Digitare *vi* seguito dal percorso del file da modificare. Per il primo file, bisogna quindi digitare *vi /etc/config/network*, ottenendo tale schermata:

Figura 30: File network

Digitare *i* per inserire del testo. Bisogna innanzitutto commentare, inserendo il carattere # all'inizio, la riga *option 'type' bridge.* Aggiungere poi la parola *'static'* alla riga *option 'proto'*. Posizionandosi sulla relativa voce, è possibile modificare netmask, indirizzo Ip e gateway (le voci sotto la categoria 'interface' 'lan_wilab'). Inserire quindi l'indirizzo Ip della scheda utilizzata, e i valori di Netmask e Gateway mostrati in figura 27. Posizionarsi in fondo al file, premere ESC e ZZ per salvare, digitare *reboot* per riavviare la scheda.

Per modificare invece l'indirizzo mac della scheda wireless, digitare *vi /etc/config/wireless*. Nella figura sottostante è mostrato l'output della schermata.

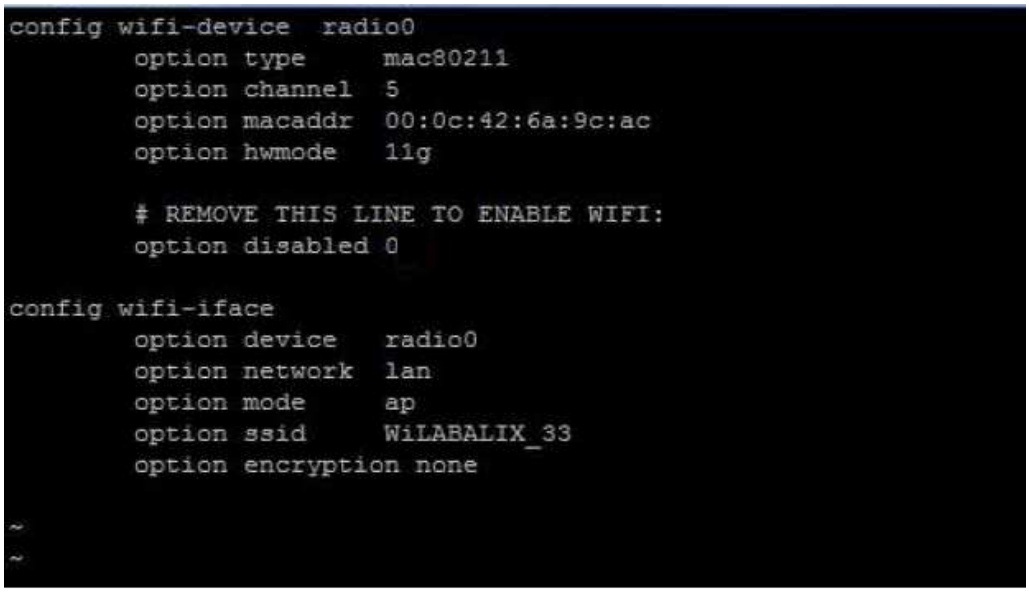

Figura 31: File wireless

Per permettere al dispositivo l'uso del Wi-Fi bisogna inserire 0 alla voce *option disabled*, altrimenti impostare il valore 1. La voce *option macaddr* contiene il mac della scheda wireless utilizzata dal dispositivo, bisogna cambiarlo se la scheda viene sostituita. Invece, in *option ssid* digitare il nome WILABALIX_ZZ dove ZZ sta per il numero del dispositivo. Posizionarsi in fondo al file, digitare ESC e poi ZZ, *reboot* per riavviare.

L'operazione successiva da eseguire è la modifica del file */etc/resolv.conf*, aggiungendo la linea *nameserver 10.1.10.5*. Tramite questa modifica, la scheda riuscirà a scaricare alcuni aggiornamenti dal web. Collegare quindi il dispositivo al web tramite cavo ethernet, e digitare il comando *opkg update* per eseguire gli aggiornamenti. Digitare (anche facendo copia-incolla, quindi con un'unica digitazione) i comandi:

opkg install wpad
opkg install kmod-madWi-Fi
opkg install kmod-ath9k
opkg install nano
opkg install subversion-client
opkg install libpciaccess

opkg install make
opkg install libelf
opkg install iperf
opkg install atk
opkg install cairo
opkg install coreutils-stty
opkg install gpsd
opkg install gtk2 48
opkg install jamvm
opkg install kismet-client
opkg install kismet-drone
opkg install kismet-server
opkg install kmod-nls-base
opkg install kmod-usb-core
opkg install kmod-usb-ohci
opkg install kmod-usb-serial
opkg install kmod-usb-serial-pl2303
opkg install kmod-usb-storage
opkg install libtdl
opkg install libpopt
opkg install pciutils
opkg install tcpdump
opkg install usbutils

Digitare i comandi *Wi-Fi down* e *Wi-Fi up*. Sono state quindi create le interfacce wireless/lan chiamate mon.wlan0 e wlan0.
Installati gli aggiornamenti, l'ultima operazione da eseguire è la creazione di uno script *iw_startup* da inserire nella cartella *root.* Tale file conterrà le impostazioni che il dispositivo caricherà all'avvio, rendendo possibile l'utilizzo delle frequenze situate intorno ai 5,9 GHz piuttosto che quelle a 2,4 GHz. Per creare il file bisogna utilizzare il software WinScp, di cui è mostrata la schermata principale nella figura sottostante.

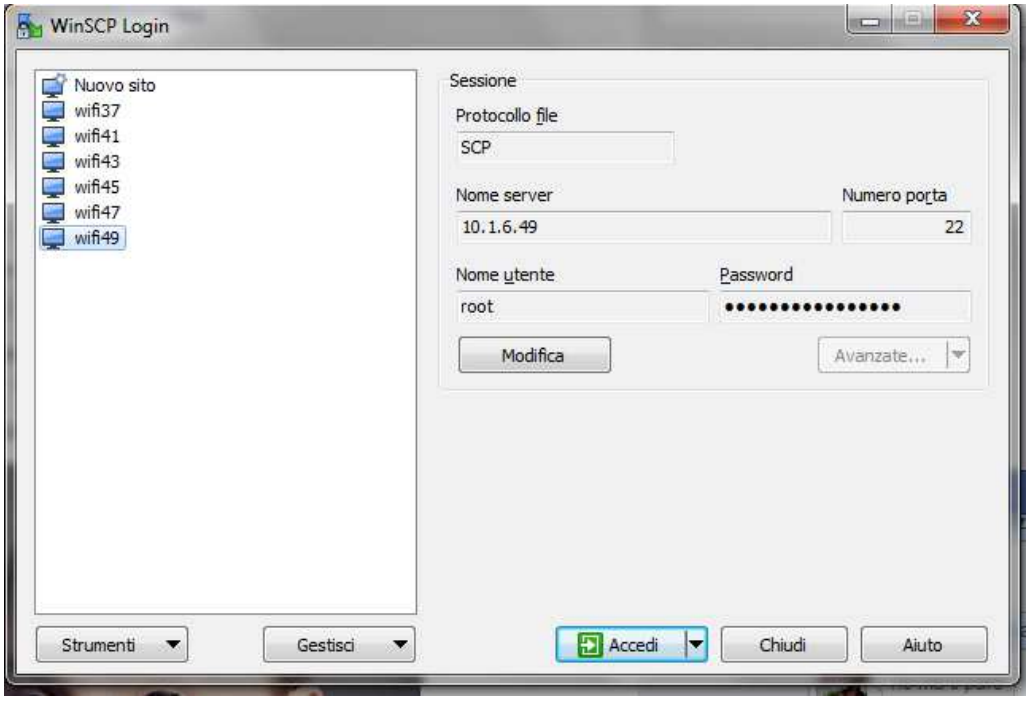

Figura 32: WinScp

Per accedere a un dispositivo, collegarsi innanzitutto via ethernet ad esso, avviare il software e scegliere SCP come protocollo file. Inserire l'indirizzo IP del dispositivo utilizzato nel campo Nome Server, e 22 su Numero Porta. Utilizzare *root* come nome utente, e *gcdc2011* come password. Anche qui, come in Putty, è possibile salvare le sessioni di login (visualizzate sulla sinistra dell'immagine).

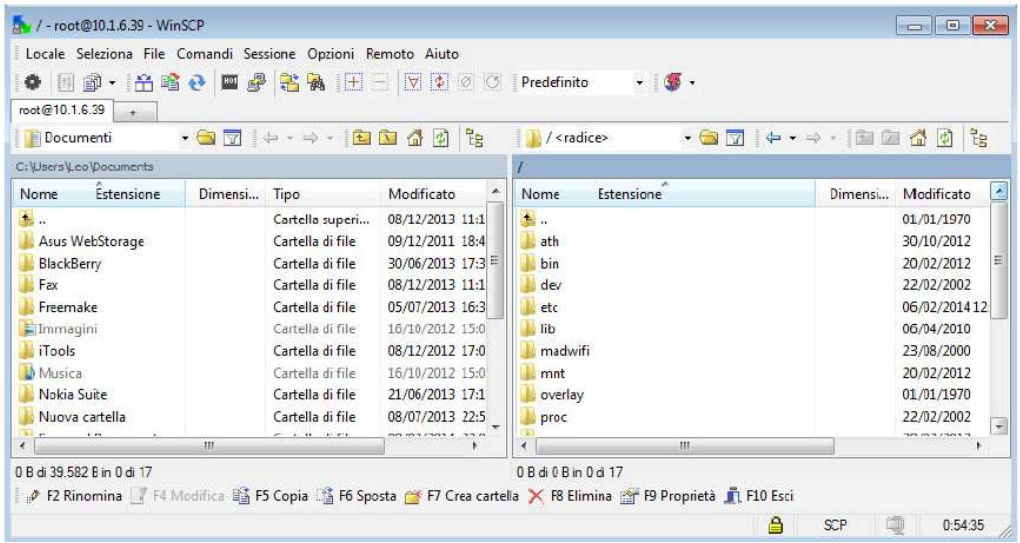

Figura 33: WinScp (2) tratta da [6]

In figura 33 è mostrata la schermata d'accesso al dispositivo. Sul lato sinistro sono visualizzate le cartelle dell'utente, sul lato destro, invece, quelle del dispositivo. Entrare quindi nella cartella *root* (non visibile in figura) e creare un nuovo file, nominandolo *iw_startup*.

Ci sono due possibili modi per modificare tale file (così come anche i file */etc/config/wireless* ed */etc/config/network*). Si può utilizzare Putty, o lo stesso software WinScp. Tramite WinScp basta aprire il file appena creato, e inserire il seguente testo:

```
#!/bin/sh
echo "Wi-Fi Down"
Wi-Fi down
sleep 1
echo "Wi-Fi Up"
Wi-Fi up
sleep 1
echo "Netherlands Frequency Register Set"
iw reg set NL
sleep 1
ifconfig wlan0 down
sleep 1
echo "Set Mode Ad-Hoc"
iwconfig wlan0 mode ad-hoc
ifconfig wlan0 up
sleep 1
echo "Set Frequency 5890MHz"
iw dev wlan0 set freq 5890
iw dev wlan0 ibss leave
iw dev wlan0 ibss join ITS49 5890 fixed-freq ff:ff:ff:ff:ff:ff beacon 2
echo "IP address"
ifconfig wlan0 10.10.6.59 netmask 255.255.0.0
```

echo "Set Rate 6M and Power 27dBm"
iwconfig wlan0 rate 6M
iwconfig wlan0 txpower 27

Nella riga *ifconfig wlan0 10.10.6.59* netmask *255.255.0.0* inserire come indirizzo IP il valore 10.10.6.YY, dove stavolta YY è il numero finale del dispositivo + 10.

Tale script permette, come detto in precedenza, di lavorare nella banda di frequenze intorno ai 5,9 GHz, utilizzando 10 MHz di banda piuttosto dei 20 MHz dell'IEEE 802.11. Salvare il file e riavviare il dispositivo.

Utilizzando Putty, invece, basta digitare su terminale *vi iw_startup* (solo dopo averlo creato tramite WinScp), e modificare il file come descritto sopra. Posizionarsi infine in fondo al file, digitare ESC + ZZ, e poi *chmod +x iw_startup* per rendere eseguibile il file all'avvio. Riavviare il dispositivo digitando *reboot*, eseguire nuovamente l'accesso, e digitare *./iw_startup* per caricare le impostazioni salvate nel file. Digitando adesso *iwconfig*, si può vedere come le impostazioni siano state caricate correttamente.

```
root@OpenWrt:~# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11abg  ESSID:"ITS33"
          Mode:Ad-Hoc  Frequency:5.89 GHz  Cell: Invalid
          Tx-Power=27 dBm
          RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

mon.wlan0  IEEE 802.11abg  Mode:Monitor  Frequency:5.89 GHz  Tx-Power=27 dBm
          RTS thr:off   Fragment thr:off
          Power Management:off

root@OpenWrt:~#
```

Figura 34: iwconfig di Putty

Per rendere il calmd funzionante all'avvio del dispositivo, e quindi per permettere al software Eclipse di interfacciarsi con la scheda, basta modificare (tramite WinScp o Putty, allo stesso modo descritto in precedenza) il file */etc/rc.local* inserendo il seguente script:

/root/iw_startup

iwconfig wlan0 txpower 27

calmd

gpsd -bG /dev/ttyUSB0

Riavviare quindi il dispositivo.

Per eseguire tali passaggi tramite Linux, si rimanda a [6].

Un ulteriore comando utilizzabile tramite Putty per impostare i valori di trasmissione delle schede è

iwconfig wlan0 rate X

dove *X* indica il modo di trasmissione.

X	MODO	MBit/s
6M	1	3
9M	2	4.5
12M	3	6
18M	4	9
24M	5	12
36M	6	18
48M	7	24
54M	8	27

Tabella 2: Valori dei modi

Il comando

iwconfig wlan0 txpower Z

rende possibile impostare il valore della potenza di trasmissione (sostituire *Z* con un valore compreso tra 1 dBm e 27 dBm). Tali comandi sono utili quando si vuole utilizzare il tool Iperf per fare misure di throughput.

4.5 Copia di una flash card

Per poter copiare il contenuto di una flash card in un'altra, bisogna utilizzare Linux. Aprire il terminale e digitare *sudo su*, inserendo la password, in modo da avere privilegi di amministratore. Inserire nel lettore la flash card di cui si vuole copiare il contenuto, creare una cartella (supponiamo il nome sia *copy*) in cui andrà inserita l'immagine copiata. Posizionarsi su tale cartella inserendo nel terminale la riga di comando *cd* seguita dal percorso della cartella, e dare il comando *dd if=/dev/sdf of=Copia*
dove la stringa *Copia* è il nome che verrà dato all'immagine copiata, può quindi essere scelto a piacimento. Inserire adesso nel lettore la compact flash nuova.
Digitare *sudo gparted* nel terminale, selezionare (dal menu a tendina in alto a destra) la flash card, e formattare entrambe le partizioni come estensione ext2. Se non è possibile farlo, cliccare sulla partizione, poi su *unmount*, e formattarla. Digitare di nuovo il comando *cd* seguito dal percorso della cartella *copy* in cui è stata salvata l'immagine, e digitare *dd if =Copia of=/dev/sdf* (in entrambi i comandi, scegliere sdf oppure sdb a seconda dell'estensione data alla compact flash da gparted).

4.6 Doppia scheda wireless

Le schede Alix, come si può vedere nelle figure 19 e 20, presentano due alloggiamenti per le schede wireless. Ciò permette di utilizzare due canali diversi di comunicazione scindendo le informazioni di controllo (inviate tramite un canale) dalle informazioni di servizio (inviate tramite l'altro). E' possibile configurare i dispositivi per l'utilizzo di entrambe le interfacce radio grazie ai file presenti attualmente in laboratorio. Per prima cosa bisogna eseguire la procedura di cross-compilazione di OpenWrt (si rimanda al capitolo 3.5 di [6] per tale procedura), effettuando però alcune modifiche.
La prima operazione da eseguire è, una volta effettuato l'accesso a Linux, copiare sul desktop il file presente nella cartella *dl*, estrarlo, entrare nella cartella estratta e sostituire il file calmd.c con la versione che si trova all'interno della

cartella contenente il calmd.c modificato per supportare la trasmissione su doppio canale. Sostituire quindi tale file, e ricompattare la cartella, facendo attenzione a rinominarla *calmd-1.1.tar.gz.*

Adesso è possibile iniziare la procedura descritta da Benedetti in [6]. Seguendo tale procedura, si arriva in un punto in cui bisogna copiare i file presenti nella cartella *gcdc-backfire-10.03-V3/feed/files* all'interno della cartella *backfire_10.03/dl*. Prima di copiare tali file, sostituire l'archivio creato in precedenza con il suo omonimo presente nella cartella *gcdc-backfire-10.03-V3/feed/files*. La procedura prosegue per come è stata descritta da Benedetti. Una volta effettuata la cross-compilazione, bisogna inserire l'immagine del sistema operativo all'interno della flashcard. Collegare quindi la flashcard al computer tramite l'apposito lettore, aprire il terminale e digitare *sudo gparted* (inserire la password *gcdc2011* se richiesta). Formattare quindi la flashcard come file system ext2 (in caso non si riesca a formattarla, cliccare prima su *unmount* e dopo su *format as ext2*). Adesso digitare sul terminale il comando *cd* seguito dal percorso fino alla cartella *...gcdc/backfire_10.03* per posizionarsi in tale cartella, digitare *sudo su* nel terminale per avere privilegi di amministratore, e dare quindi il comando *dd if=bin/x86/OpenWrt-x86-squashfs.image of=/dev/sdb*. In questo modo l'immagine del sistema operativo viene trasferita sulla flashcard (non verranno visualizzate stampe a video durante l'esecuzione del comando).

4.6.1 Parametri e script da impostare

Una volta effettuata l'operazione di cui sopra, bisogna modificare alcuni script presenti all'interno del dispositivo. Dal momento che la Alix non possiede ancora un indirizzo IP (e quindi non è possibile effettuare l'operazione descritta nel paragrafo 3.4), bisogna collegarsi via porta seriale. Collegare quindi il dispositivo e il pc tramite cavo seriale, e il dispositivo a Internet tramite cavo ethernet. Aprire il software Putty, e scegliere la modalità di connessione *Serial*, scegliendo la porta seriale attuale (di norma è la COM1, in caso contrario basta cliccare col tasto destro su Risorse del Computer > Proprietà > Gestione dispositivi, e verificare la porta seriale utilizzata), e inserire 38400 alla voce

speed. Effettuare quindi l'accesso a OpenWrt (è possibile che si renda necessario avviarlo digitando il comando), ed eseguire la procedura descritta nel paragrafo 3.4.

Quando le procedure di aggiornamento sono state portate a termine e gli indirizzi IP, la netmask e il resto dei parametri sono stati correttamente configurati, bisogna modificare alcuni file del dispositivo.

Non appena viene assegnato al dispositivo un indirizzo IP, è possibile accedervi anche tramite cavo ethernet. Aprire quindi WinScp, digitando l'indirizzo IP del dispositivo, e modificare i seguenti script come descritto qui sotto:

File *iw_startup*:

```
#!/bin/sh
echo "Wi-Fi Down"
Wi-Fi down
sleep 1
echo "Wi-Fi Up"
Wi-Fi up
sleep 1
echo "Netherlands Frequency Register Set"
iw reg set NL
sleep 1
ifconfig wlan0 down
ifconfig wlan1 down
sleep 1
echo "Set Mode Ad-Hoc"
iwconfig wlan0 mode ad-hoc
iwconfig wlan1 mode ad-hoc
ifconfig wlan0 up
ifconfig wlan1 up
sleep 1
echo "Set Frequency 5890MHz"
iw dev wlan0 set freq 5900
iw dev wlan1 set freq 5880
```

```
iw dev wlan0 ibss leave
iw dev wlan1 ibss leave
iw dev wlan0 ibss join ITS 5900 fixed-freq ff:ff:ff:ff:ff:ff beacon 2
iw dev wlan1 ibss join ITS 5880 fixed-freq ff:ff:ff:ff:ff:ff beacon 2
echo "IP address"
ifconfig wlan0 10.10.6.49 netmask 255.255.0.0
ifconfig wlan1 10.10.6.50 netmask 255.255.0.0
echo "Set Rate 6M and Power 27dBm"
iwconfig wlan0 rate 6M
iwconfig wlan1 rate 6M
iwconfig wlan0 txpower 27
iwconfig wlan1 txpower 27
```

In questo modo sia per la wlan0 che per la wlan1 (ultime 4 righe) viene utilizzato il Modo 1 con potenza di trasmissione pari a 27 dBm. Inoltre, nella riga ***iw dev wlan0 set freq 5900*** e in ***iw dev wlan1 set freq 5880*** si impostano le frequenze utilizzate dalle due schede wireless (5,9 GHz e 5,88 GHz). Eventuali fenomeni di interferenza sono trattati in [6]. Un altro script da modificare è il *etc/config/wireless.* Oltre alla configurazione per wlan0 bisogna aggiungere:

```
config Wi-Fi-device radio1
        option type mac80211
        option channel 5
        option macaddr (mac riportato sulla scheda wireless)
        option hwmode 11g
        # REMOVE THIS LINE TO ENABLE WI-FI:
        option disabled 0

config Wi-Fi-iface
        option device radio1
        option network lan
        option mode ap
       option ssid WiLABALIX_XX
```

option encryption none

sostituendo il numero del dispositivo in WILABALIX_XX.

Tali modifiche rendono possibile l'utilizzo di entrambe le schede wireless. L'ultima operazione da effettuare è il collegamento della seconda antenna, come mostrato nelle figure sottostanti.

Figura 35: Collegamento della seconda antenna (1)

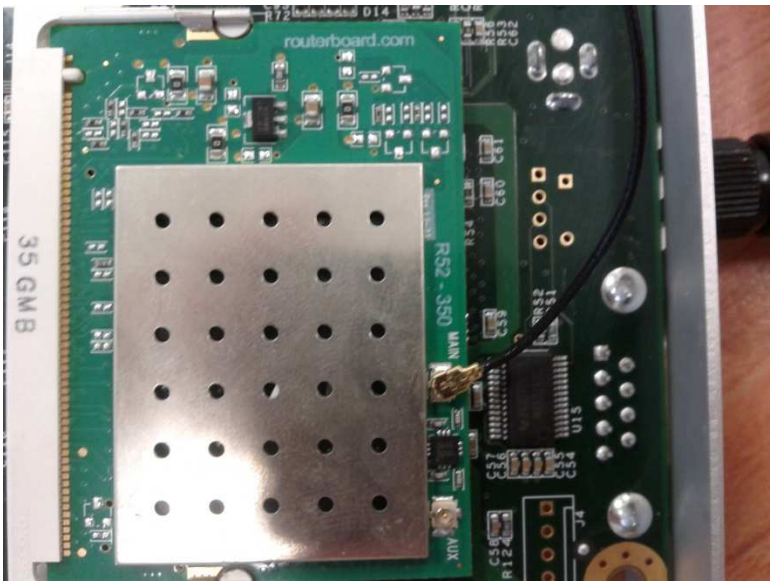

Figura 36: Collegamento della seconda antenna (2)

Figura 37: Collegamento della seconda antenna (3)

5. CODICE JAVA

5.1 Main

Il codice è stato implementato dalle Università della Germania, dell'Olanda e, ovviamente, di Bologna. Il progetto italiano si trova all'interno della cartella *GCDC_Test3x/src/it/unibo*. Aprire innanzitutto Eclipse e scegliere una cartella come Workspace. Cliccare su *File*, poi su *Import*, e nella finestra che si viene ad aprire selezionare la voce *Existing Project into Workspace*. Selezionare quindi la cartella contenente i file del progetto Java, premere *Next* e importarli. La figura sottostante rappresenta l'inizio della classe Main.

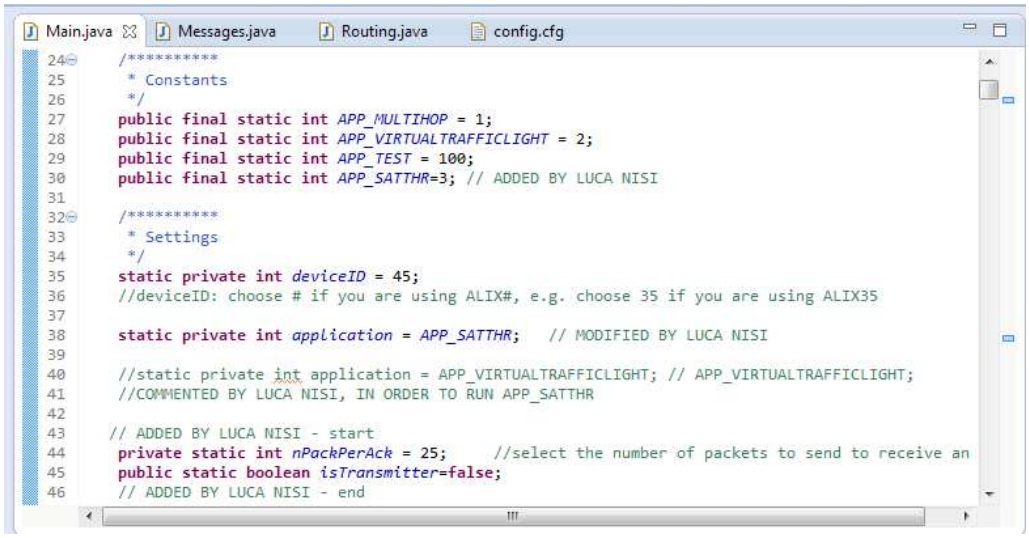

Figura 38: Classe Main

I parametri da impostare sono *deviceID* nel file di configurazione *config.cfg*, inserendo il numero del dispositivo che si sta utilizzando. Non è necessario modificare tale parametro anche nella classe *Main* in quanto se tale parametro viene inserito nel file *config.cfg*, sovrascrive il numero scritto nella classe *Main.*

```
25 #
26 ## Starting point zona via Pasubio
27 ## Starting point Bologna
28 #
29 # from NORTH-EAST
30 startingLat = 44.499895
31 startingLon = 11.319737
32 ## Route file
33 virtualGPSroute = RoutePasubioNEtoW
34 #virtualGPSroute = RoutePasubioNEtoS
35 #
36 # from SOUTH-WEST
37 # startingLat = 44.499498
38 # startingLon = 11.314523
39 ## Route file
40 #virtualGPSroute = RoutePasubioSWtoN
41 #virtualGPSroute = RoutePasubioSWtoE
42
```

Figura 39: File config.cfg

La figura 39 mostra una porzione del file *config.cfg*. Qualora si decidesse di utilizzare le coordinate virtuali basta de-commentare le righe relative alla posizione desiderata. Scorrere tale file e impostare a *false* il parametro *uploadToServerEnabled.*

Per utilizzare l'applicazione che verrà descritta in seguito, la *APP_SATTHR*, bisogna impostare (come si può vedere in figura 38) il parametro *application* con il valore *APP_SATHR* (che viene fissato pari a 3). Il codice prosegue, fino a quando uno *switch* (visibile in figura 40) permette di scegliere il percorso in base all'applicazione selezionata.

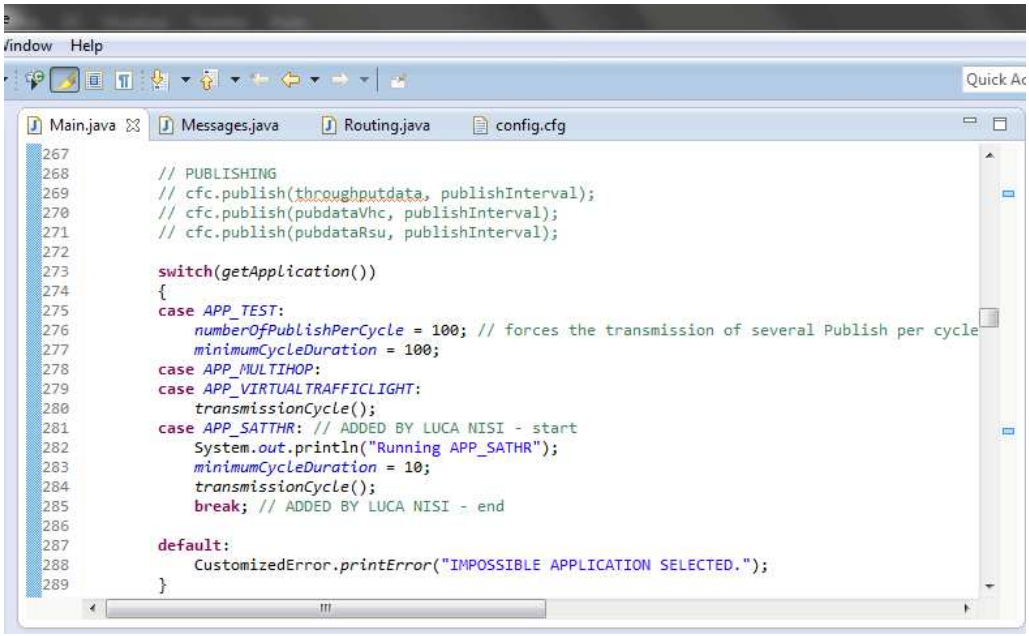

Figura 40: Switch

L'applicazione selezionata è la *APP_SATHR*, quindi lo *switch* porta direttamente al metodo *transmissionCycle().*

5.2 TransmissionCycle()

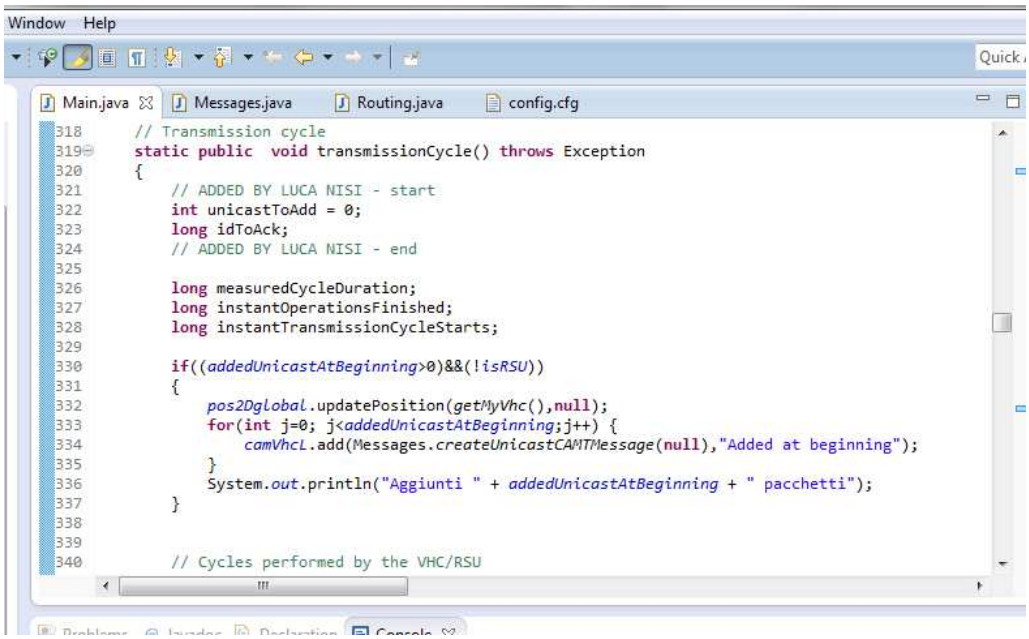

Figura 41: transmissionCycle()

Il transmissionCycle() altro non è che un metodo che si occupa della trasmissione dei pacchetti. La prima operazione che tale metodo compie è l'update della posizione del veicolo (solo ed esclusivamente se l'istanza di Eclipse non è settata per essere una RSU tramite il relativo parametro booleano *isRSU*).

L'operazione successiva, ovvero quella nella riga n. 334 di figura 41 è l'aggiunta nella coda di trasmissione dei pacchetti aggiungi all'inizio, gli *addedUnicastAtBeginning* (utili per non avere una coda di trasmissione già vuota all'inizio del ciclo). In caso l'applicazione selezionata sia la *APP_SATTHR*, l'update della posizione è effettuato tramite il metodo *Routing.checkRSUVisible(),* che verrà descritto nel seguito. Eseguiti tali comandi, il codice prosegue.

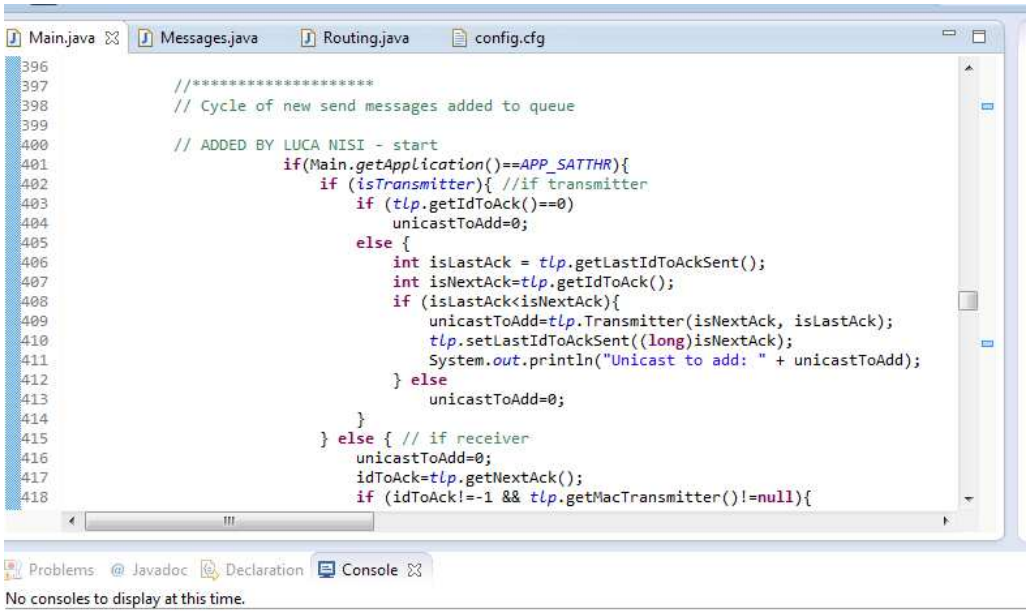

Figura 42: Oggetto di classe Transport Level Packet Manager

In figura 42 si può vedere l'inizio della porzione di codice utile a calcolare gli *unicastToAdd* da aggiungere alla coda di trasmissione. Tale algoritmo permette di avere una coda di trasmissione che abbia in ogni momento dei dati da trasmettere, senza che vada in saturazione né che debba attendere l'invio effettivo dei pacchetti. L'oggetto *tlp* presente nella riga 403 di figura 42 appartiene alla classe *Transport Level Packet Manager*.

5.3 Transport Level Packet Manager

Lo scopo di tale classe da me creata è fornire, rispettivamente al dispositivo utilizzato come ricevitore e a quello impostato come trasmettitore, l'ID dell'ack da inviare (parametro che verrà calcolato sulla base del numero di sequenza dei pacchetti ricevuti) e il numero di pacchetti da aggiungere alla coda di invio. Sono stati creati due costruttori per ambo i lati di trasmissione, mostrati nella figura sottostante.

```
/********** CONSTRUCTOR FOR RECEIVER **********/
public TransportLevelPacketManager(boolean isTransmitter, int nPackPerAck){

    this.isTransmitter=isTransmitter;
    this.nPackPerAck = nPackPerAck;
}


/********** CONSTRUCTOR FOR TRANSMITTER **********/
public TransportLevelPacketManager(int nPackPerAck){

    this.nPackPerAck=nPackPerAck;
}
```

Figura 43: Costruttori

Il primo, per il ricevitore, prende in ingresso i parametri:

1. *isTransmitter* - un booleano (i cui possibili valori sono *true* o *false)*, con il quale si comunica alla classe se l'apparecchio in questione è un ricevitore o un trasmettitore;
2. *nPackPerAck* - un intero che stabilisce quanti pacchetti devono essere stati ricevuti prima di inviare un ack.

Il secondo costruttore invece, relativo al lato trasmissione, prende in ingresso solo il parametro *nPackPerAck*. Utilizzare due costruttori con numero di parametri in ingresso diversi è necessario al corretto funzionamento del codice java.

Successivamente sono stati sviluppati i metodi *Receiver* e *Transmitter*, con i quali vengono calcolati i parametri oggetto di ricerca.

```
/********** METHOD FOR RECEIVER **********/
public void Receiver(int lastPacketReceived) {
    lastAckSnt=localLastAckSnt;

 if (isTransmitter==false){
     int nToBeAcked = lastPacketReceived-lastAckSnt*nPackPerAck;
     if ( nToBeAcked >= nPackPerAck ) {
         nextAck = (int) Math.floor(lastPacketReceived/nPackPerAck);
         localLastAckSnt=nextAck;
     } else {
         nextAck=lastAckSnt;
         localLastAckSnt=nextAck;
     }
 }
```

Figura 44: Metodo del ricevitore

```
/********** METHOD FOR TRANSMITTER **********/
public int Transmitter(int actualAckRcv, int previousAckRcv) {

  int packetToTransmit=-1;

  if(actualAckRcv>previousAckRcv){
      packetToTransmit=(actualAckRcv-previousAckRcv)*nPackPerAck;
      }

  return packetToTransmit;
}
```

Figura 45: Metodo del trasmettitore

Il metodo *Receiver* memorizza nella variabile *lastAckSnt* l'ID dell'ack inviato in precedenza, poi esegue una verifica sul parametro *isTransmitter*. Superata tale verifica si determina il valore di *nToBeAcked*, ovvero il numero di pacchetti ricevuti dopo l'invio del precedente ack. Se tale valore è più grande di quello del parametro *nPackPerAck*, allora viene calcolato l'ID del nuovo ack da inviare (variabile *nextAck).*

Il metodo *Transmitter* invece calcola il numero di pacchetti che andranno aggiunti alla coda di trasmissione, basandosi sui due ack ricevuti, quello attuale e quello precedente.

Sono stati definiti ulteriori metodi: *getLastAckSent(), getNextAck(), getIdToAck()* che permettono di leggere i valori dei rispettivi parametri dall'esterno della classe *TransportLevelPacketManager* (dualmente a questi, sono stati creati gli stessi metodi *set*(input parameter)*, che permettono di cambiare il valore del rispettivo parametro con quello inviato in input). Infine, i metodi *getMacTransmitter()* e *setMacTransmitter(input parameter)* permettono di leggere (o di settare) l'indirizzo MAC del nodo che ha trasmesso il pacchetto.

5.4 Integrazione con il codice sorgente

5.4.1 Classe Main

Il parametro *isTransmitter* nella classe *Main* va impostato a *true* se ci si trova dal lato trasmettitore, viceversa a *false*. Il booleano *isRsu* va settato a *true* se il dispositivo è utilizzato come RSU, altrimenti a *false.*

```
//SETTING FOR THE APP_SATHR
if (Main.getApplication()==APP_SATTHR){
    routingProtocol=Routing.ROUTING_TRANSPORTLEVELP;
    tlp = new TransportLevelPacketManager(nPackPerAck);
}
```

Figura 46: Protocollo di routing

Come visualizzato in figura 46, in caso l'applicazione in uso sia l'APP_SATTHR (quella che ho realizzato con il Transport Level Packet Manager) il valore di *routingProtocol* (utilizzato nella classe *Routing)* viene forzato al valore *Routing.ROUTING_TRANSPORTLEVELP* (definito anch'esso nella classe *Routing*), in seguito viene creato l'oggetto *tlp* appartente alla classe descritta nel paragrafo precedente.

Una volta all'interno del metodo *TransmissionCycle()* della classe *Main*, viene eseguito un update delle posizioni degli altri nodi tramite il metodo *Routing.checkRSUVisible* (che sarà descritto in seguitò) nel caso il dispositivo funga da trasmettitore, altrimenti l'update delle posizioni viene eseguito tramite il metodo *Routing.update*, come visualizzato nella figura sottostante.

```
// Cycles performed by the VHC/RSU
for (int i = 0; i < transmissionCycles; i++)
{
    if (Main.getApplication()==APP_SATTHR){                    //
        if (isTransmitter){
            Routing.NextNode=Routing.checkRSUVisible(gethNodeL());
        } else {
            Routing.update(Routing.ROUTING_GF,getMyVhc(),gethNodeL());
            }
        }
```

Figura 47: Update della posizione

Successivamente, si calcolano i pacchetti (variabile *unicastToAdd*) da aggiungere alla coda di trasmissione, tramite l'oggetto *tlp*, come mostrato nella figura sottostante,

```
if(Main.getApplication()==APP_SATTHR){
    if (isTransmitter){ //if transmitter
        if (tlp.getIdToAck()==0)
            unicastToAdd=0;
        else {
            int isLastAck = tlp.getLastIdToAckSent();
            int isNextAck=tlp.getIdToAck();
            if (isLastAck<isNextAck){
                unicastToAdd=tlp.Transmitter(isNextAck, isLastAck);
                tlp.setLastIdToAckSent((long)isNextAck);
                System.out.println("Unicast to add: " + unicastToAdd);
            } else
                unicastToAdd=0;
        }
    } else { // if receiver
        unicastToAdd=0;
        idToAck=tlp.getNextAck();
        if (idToAck!=-1 && tlp.getMacTransmitter()!=null){
            Messages.sendThisMessage(Messages.createUnicastCAMTMessage(tlp
        } else {
            if (idToAck==-1)
                System.out.println("No ack to send");
            if (tlp.getMacTransmitter()==null)
```

Figura 48: Calcolo degli unicastToAdd

poi si aggiungono gli *unicastToAdd* nella coda di trasmissione, pronti per l'invio.

5.4.2 Classe Messages

Il seguente codice è stato scritto all'interno del metodo *camUnicastProcess*, in cui vengono processati i pacchetti ricevuti.

```
// CODE FOR THE APP_SATTHR                    // ADDED BY LUCA NISI - start
if (Main.getApplication()==Main.APP_SATTHR){

    if (Main.isTransmitter){ // if transmitter
        Main.tlp.setIdToAck(camT.getCam().getHeader().getMessageToSend().getValue());
    } else { // if receiver

        Main.tlp.setMacTransmitter(camT.getSourceUrl());
        Main.tlp.Receiver(camT.getCam().getHeader().getSequenceNumber());
```

Figura 49: Porzione del camUnicastProcess

In caso il dispositivo sia usato come trasmettitore, il metodo *setIdToAck* dell'oggetto *tlp* permette di impostare l'ID dell'ack da inviare tramite il valore letto nell'header del pacchetto stesso, altrimenti, in caso di ricevitore, il metodo *setMacTransmitter* salva l'indirizzo MAC del trasmettitore, poi il metodo *Receiver* calcola l'id dell'ack da inviare. Il numero di sequenza dell'ultimo pacchetto ricevuto è letto nell'header, tramite il metodo *getSequenceNumber()*.

5.4.3 Classe Routing

In questa classe viene definito il parametro *ROUTING_TRANSPORTLEVELP*. Se, dal *main*, viene invocata questa classe per eseguire l'update delle posizioni dei nodi, il metodo *update* fa un controllo sul protocollo di routing memorizzato nella variabile *RoutingProtocol*. Se tale variabile è uguale a *ROUTING_TRANSPORTLEVELP*, viene invocato il metodo *checkRSUVisible*, altrimenti si usa il metodo *getNearestNodeTowardRsu.*

```
public static void update(int routingProtocol, HeardNode myVhc, HeardNodesList HNodeL) {
    if(routingProtocol==ROUTING_TRANSPORTLEVELP){          //ADDED BY LUCA NISI
        NextNode=Routing.checkRSUVisible(HNodeL);         //ADDED BY LUCA NISI
    } else {
    NextNode=Routing.getNearestNodeTowardsRSU(routingProtocol, myVhc, getLonRSU(), getLatRSU(),HNodeL);
    }
}
```

Figura 50: Routing

Nella figura 51 mostro invece il metodo *checkRSUVisible.* Tale metodo prende in ingresso la lista dei nodi presenti (*HNodeL),* effettua una scansione su tale lista e restituisce il nodo che risulta essere la RSU.

```
public static HeardNode checkRSUVisible(HeardNodesList HNodeL){ // ADDED BY LUCA NISI - start

    HeardNode nodeToReturn=null;
    if (Main.isTransmitter){
      for (int k=0; k<HNodeL.getNumOfElements(); k++){
        //if (HNodeL.get(k).getNodeTypeIntForm()==1){ // RSU INT FORM IS EQUAL TO 1
          if (HNodeL.get(k).getNodeType()==HeardNode.EnumType.RSU){
              nodeToReturn=HNodeL.get(k);
              break;
          }
        }
      if (nodeToReturn==null)
        System.out.println("RSU not found.");
      }
    return nodeToReturn;
}                                                                    //ADDED BY LUCA NISI - end
```

Figura 51: checkRSUVisible()

6. RISULTATI SPERIMENTALI

6.1 Misure di Throughput in condizioni di non visibilità tramite Iperf

Sono state effettuate alcune misure di throughput utilizzando il tool Iperf di OpenWrt, che consente di inviare flussi di dati UDP e TCP per misurare, appunto, il troughput del collegamento e i pacchetti persi. Un dispositivo viene settato in modalità server, e l'altro in modalità client. Il secondo trasmette datagrammi UDP al server che è in ascolto in una determinata porta. Una volta ricevuti tali datagrammi, il server invia l'ack.
Sono stati utilizzati due dispositivi Unex per una prima misura (n. 43 e n. 47). Una seconda prova è stata effettuata con altri due dispositivi (MicroTik), il n. 37 e il n. 39
Lo scenario in cui tali misure sono state effettuate (insieme al laureando Michele Livini [10]) è quello di non visibilità, situandoci nel parcheggio della Scuola di Ingegneria. La frequenza utilizzata è 5.89 GHz, con singolo canale radio e Modo 1 (3 Mbit/s), a distanza variabile dall'angolo del muro. Dopo aver impostato i parametri degli indirizzi IP dei due portatili come spiegato nei precedenti capitoli, abbiamo collegato i dispositivi a delle batterie. Una volta effettuato l'accesso ad OpenWrt, abbiamo dato i comandi (per entrambi i dispositivi):

iwconfig wlan0 txpower 15
iwconfig wlan0 rate 6M

Il primo serve a impostare una potenza in trasmissione pari a 15 dBm (in modo da ottenere in uscita una potenza pari a 20 dBm, aggiungendo i 5 dBm di guadagno dell'antenna) , mentre il secondo imposta l'utilizzo del Modo 1.
Per settare un dispositivo in modalità server, viene dato il comando

iperf -s -u -i <TEMPO> -l <PAYLOAD>

dove:
-s indica che si utilizza il dispositivo in modalità server

-u indica l'utilizzo di datagrammi UDP

-i <TEMPO> indica l'intervallo di tempo (misurato in secondi) di visualizzazione del throughput (noi abbiamo scelto 10 secondi, quindi abbiamo inserito *10* al posto di <TEMPO>)

-l <PAYLOAD> indica che si utilizza il payload UDP

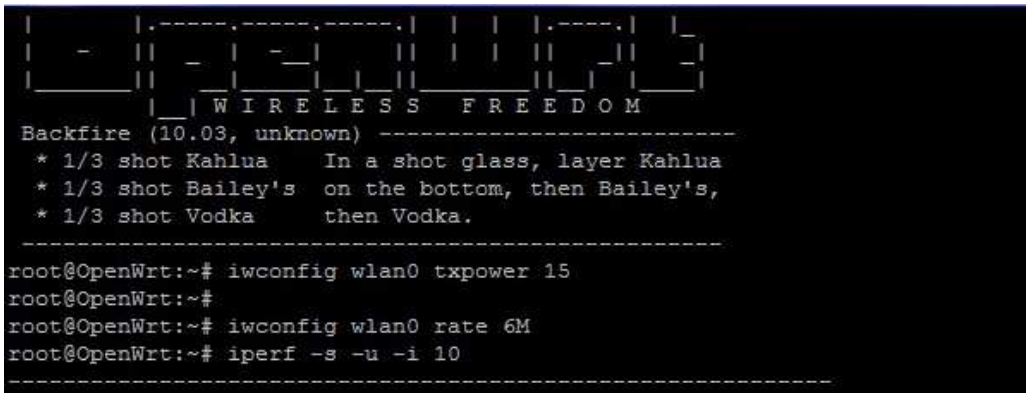

Figura 52: Putty (1)

Per impostare invece un dispositivo in modalità client, bisogna digitare il comando:

iperf –c <802.11p_IP-ADDR_SERVER> -u –t <TEMPO> -i <TEMPO> -b <RATE> -l <PAYLOAD>

dove:

-c indica che il dispositivo è impostato in modalità client

-<802.11p_IP-ADDR_SERVER> va modificato inserendo l'IP della scheda impostata come server, aggiungendo 10 al numero finale (ad esempio, 10.1.6.47 per la scheda ALIX 37)

-u indica l'utilizzo di datagrammi UDP

-t <TEMPO> imposta la durata totale (in secondi) della trasmissione

-i <TEMPO> indica l'intervallo di tempo di visualizzazione (in secondi)

-b <RATE> indica la bit rate in Mbit/s

-l <PAYLOAD> indica il payload UDP

Figura 53: Putty (2)

Riporto adesso le stampe a video relative al lato server. Per ogni intervallo temporale vengono visualizzate le informazioni riguardanti la bandwith (throughput), il jitter (arrivo dei pacchetti in tempi diversi rispetto all'ordine di partenza), il rapporto tra pacchetti persi e inviati.

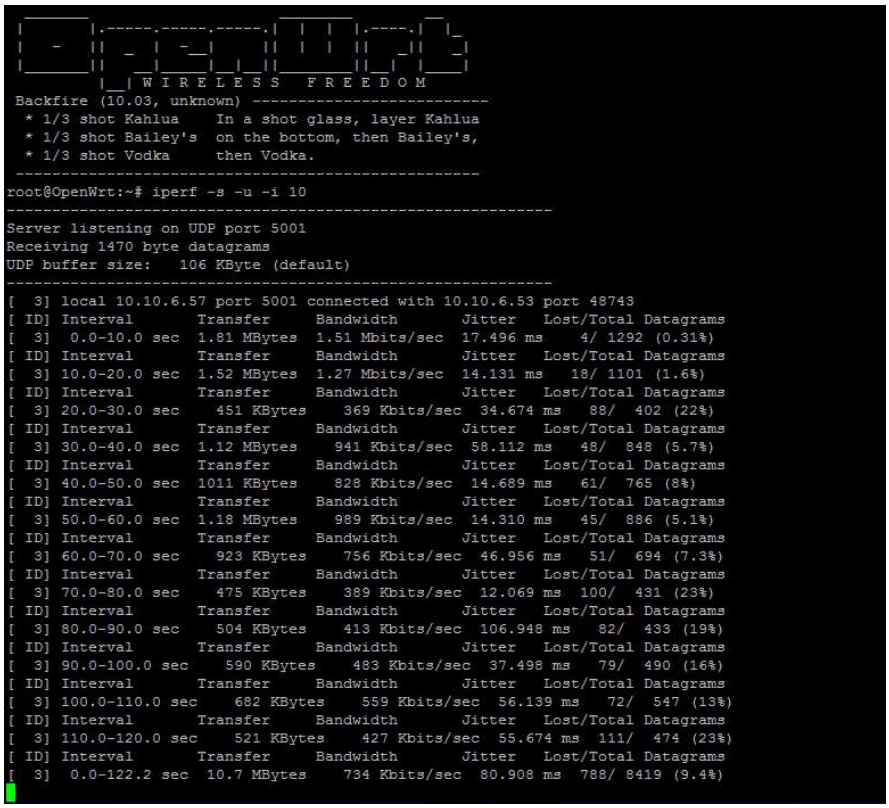

Figura 54: Putty (3)

Come si può interpretare dalla schermata, la misura è durata 122.2 secondi (impostata a 120 secondi quindi, dal lato client), e ogni 10 secondi vengono visualizzate le informazioni sulla trasmissione. La schermata precedente è riferita a una misura con schede Unex, distanti (entrambe) 15 metri dall'incrocio. La schermata successiva mostra invece lo stesso lato server, a distanza 5 metri, utilizzando però i dispositivi MicroTik.

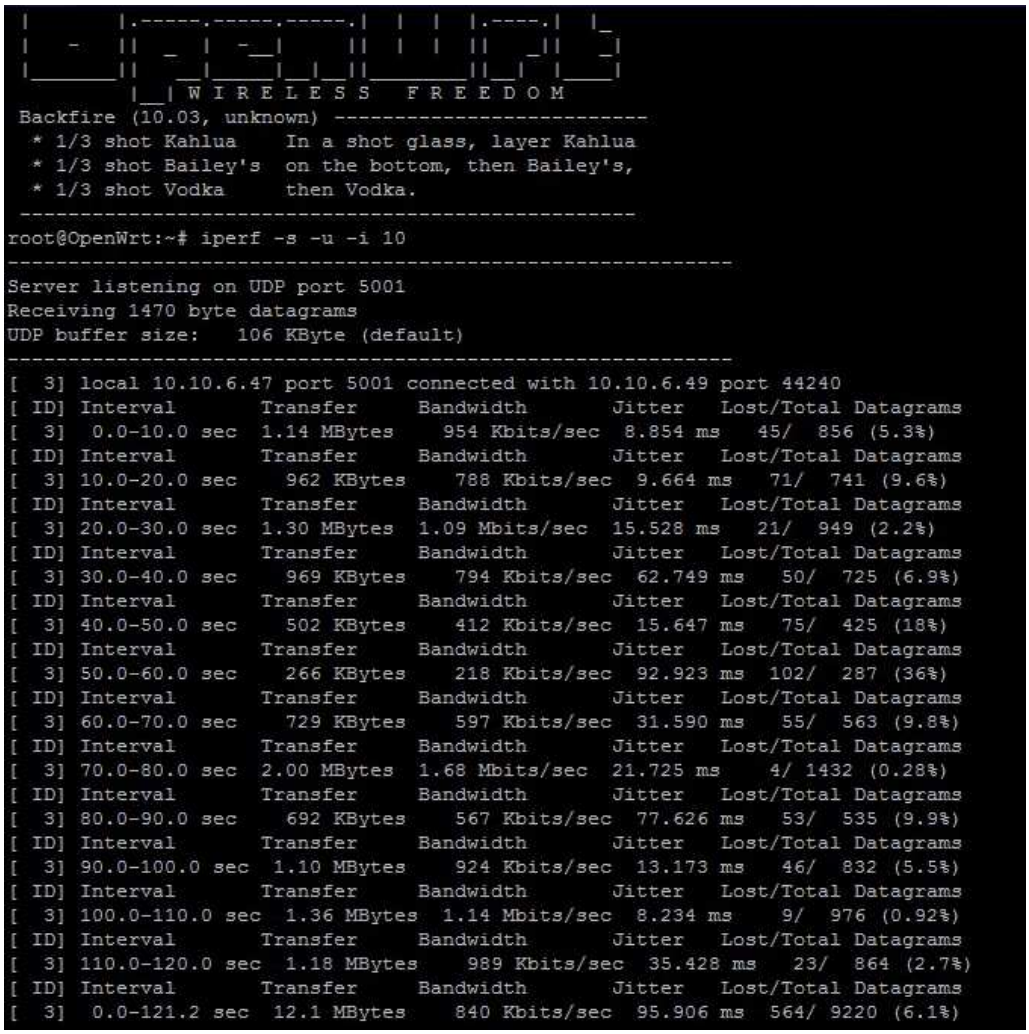

Figura 55: Putty (4)

Si può notare come i valori di bandwith siano simili a quelli ottenuti utilizzando i dispositivi Unex (posizionando però i dispositivi MicroTik a distanza pari a 1/3 rispetto all'altra coppia).

Adesso mostro le immagini di Iperf lato client.

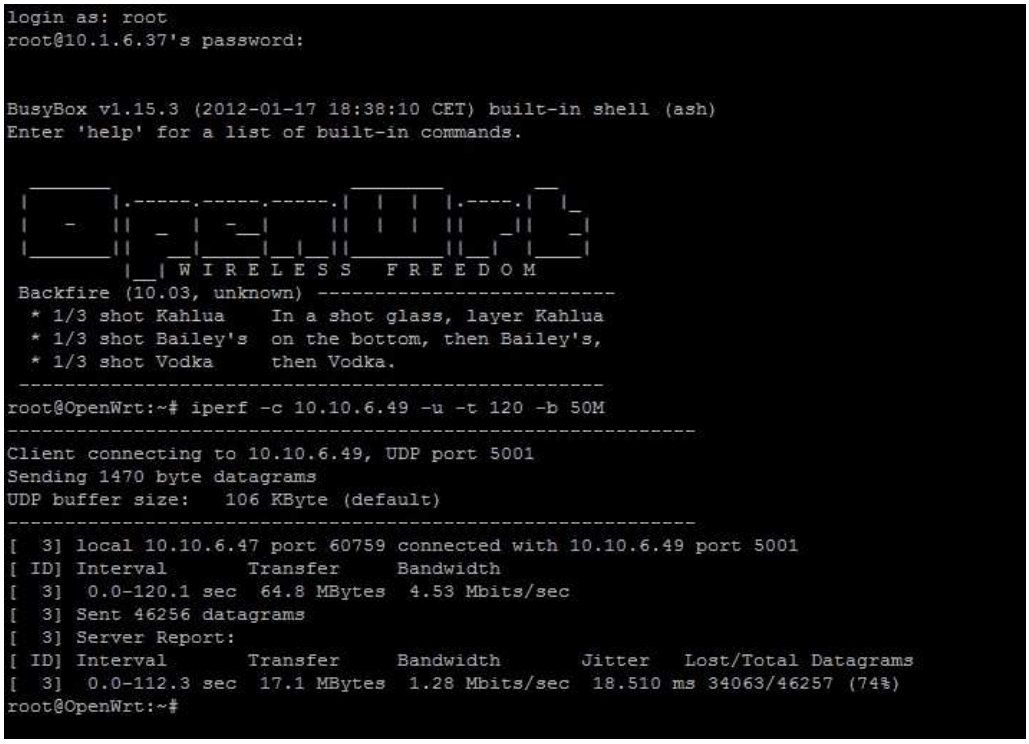

Figura 56: Putty (5)

La schermata dà informazioni sulla durata della trasmissione, sulla quantità di dati trasferiti, sulla bandwith, sul jitter e sui pacchetti persi rispetto a quelli totali. Tale misura è stata effettuata con dispositivi MicroTik a distanza di 5 metri dall'incrocio. Mostro adesso lo stesso tipo di immagine, relativa all'utilizzo di dispositivi Unex, a distanza pari a 5 metri.

Figura 57: Putty (6)

Si può notare come, con i dispositivi Unex, venga perso solo un pacchetto sui 18085 totali inviati, mentre, mantenendo ferma la distanza, coi dispositivi MicroTik se ne perde il 74% di quelli inviati.

Mostro adesso alcune immagini relative agli scenari utilizzati. I dispositivi disponevano di un cavo d'antenna corto (di lunghezza pari a 145 cm) con perdita di circa 5 dB.

Le misure sono state effettuate a distanza 5, 7, 10, 15, 20 e 25 metri.

Figura 58: DISTANZA 7 METRI

Figura 59: Posizionamento delle antenne

Mostro adesso un grafico relativo al throughput.

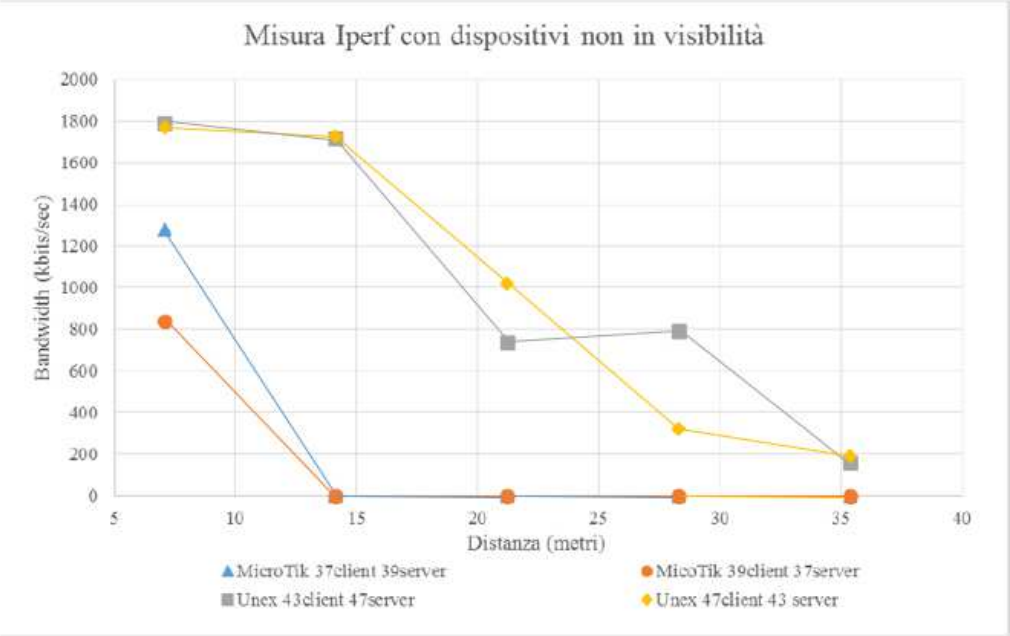

Figura 60: Grafico di Throughput

Ciò che abbiamo concluso, una volta effettuate le misure, è che le schede MicroTik riescono a ricever e fino ad una distanza di 7 metri (a distanza superiore non riuscivano proprio a collegarsi l'una all'altra), con una perdita di pacchetti elevata (pari al 60/70%).

Le schede Unex invece sono più affidabili, sono riuscite a trasmettere fino ad una distanza pari a 35 metri, sebbene a tale distanza ovviamente la bit rate diminuisce notevolmente. A distanza 7 metri, confrontando le Unex e le MikroTik si nota come le Unex raggiungano una bandwith di 1,70 Mbit/s, con una perdita di pacchetti prossima allo zero.

6.2 Misure di Throughput effettuate in laboratorio tramite codice Java

A differenza delle misure descritte nel paragrafo precedente (effettuate utilizzando il tool Iperf), il grafico di throughput mostrato alla fine del suddetto paragrafo è stato ottenuto effettuando misure tramite codice Java ed Eclipse.
Ho collegato il mio computer e i due dispositivi allo switch tramite cavo ethernet e ho aperto due istanze di Eclipse (una per dispositivo), adattando il file *config.cfg* al rispettivo ID del dispositivo in questione. In un'istanza di Eclipse ho impostato il parametro *isRSU* a *false*, nell'altra è stato impostato a *true*.
Si definiscono le variabili:

1. *addedUnicastAtBeginning*=1000 ;
2. *numberOfSendPerCycle*=30;
3. *nPackPerAck* = 50

che rappresentano, rispettivamente, i pacchetti *unicast* aggiunti all'inizio della trasmissione, il numero di pacchetti *send* inviati ad ogni ciclo e il numero di pacchetti necessario per l'invio di un ack.
Il throughput effettivo misurato risulta essere pari a poco più di 500 kbit/s, rispetto ai 3 Mbit/s nominali del MODO 1. Calcolandolo analiticamente, (come riportato in [20]) il throughput risulta pari a 1.4 Mbit/s. Si può vedere quindi come si è al di sotto del massimo throughput analitico raggiungibile.
Mostro adesso il grafico che rappresenta il troughput relativo a una delle misure effettuate. Sull'asse delle ascisse sono visualizzati circa 2500 ms, su quello delle ordinate invece 200 kbyte.

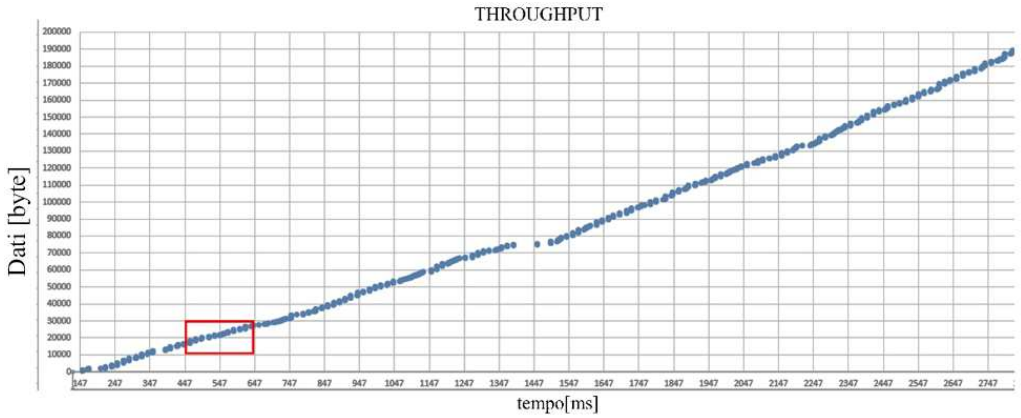

Figura 61: Throughput

Nella figura successiva invece, faccio uno zoom sulla porzione in rosso. Sull'asse delle ascisse sono visualizzati 200 ms, invece sull'asse delle ordinate sono rappresentati 11,5 kbyte.

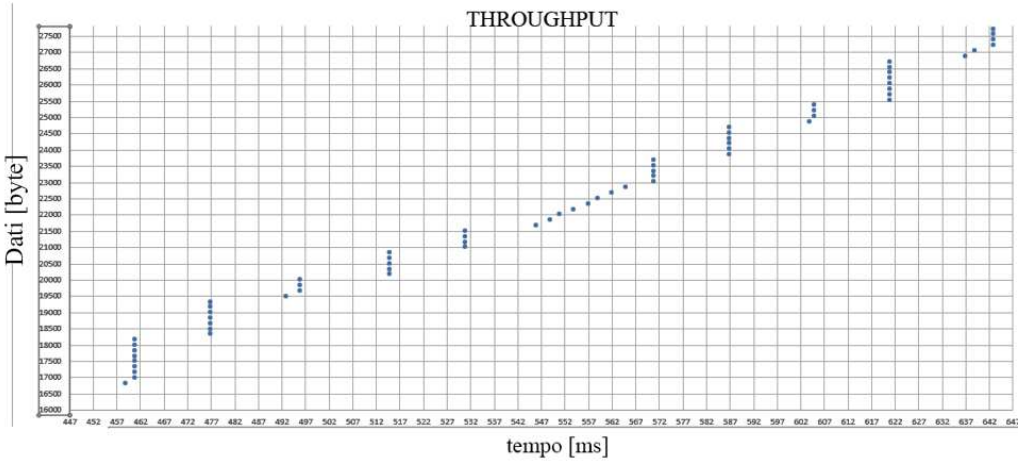

Figura 62: Porzione del grafico di Throughput

6.3 Sperimentazioni di Offloading

Si è cercato di riprodurre l'offloading da una rete infrastruttura ad una rete ad hoc 802.11p. I test effettuati in laboratorio hanno previsto una trasmissione attraverso rete LAN, con offloading su rete IEEE 802.11p a partire da un determinato istante impostato da input. Il codice Java è stato adattato per supportare i due tipi di trasmissione, in modo da aprire una socket per la trasmissione attraverso rete struttura, ed un'altra per la trasmissione su rete 802.11p. Tre dei parametri che gestiscono l'offloading sono:

1. la variabile *NOUSEOFOBU,* un booleano, impostata a *false* fa sì che la trasmissione dei pacchetti è effettuata solo tramite rete LAN, altrimenti, impostandola a *true,* rende possibile l'invio dei dati tramite rete 802.11p;
2. la variabile *SENDOVERNETENABLED,* un booleano, impostata a *false* (in concomitanza con *NOUSEOFOBU=false)* fa sì che la trasmissione dei dati avvenga tramite rete 802.11p. Impostandola a *true* (in concomitanza con *NOUSEOFOBU=true)* permette la trasmissione dei pacchetti tramite rete struttura. Infine, settando *SENDOVERNETENABLED=true* e *NOUSEOFOBU=false*, è possibile effettuare l'offloading;
3. *timerOffloading* è il parametro che da input definisce l'istante in cui viene effettuato l'offloading;
4. la variabile *throughNetServerAddress* serve a settare l'indirizzo IP da utilizzare nelle misure con offloading. Se impostata a *localhost* fa sì che un singolo pc riesca a gestire le due istanze di Eclipse.

I parametri di configurazione sono stati impostati a:

1. *addedUnicastAtBeginning = 1000;*
2. *numberOfSendPerCycle = 30;*
3. *nPackPerAck = 50*.

Dal grafico che mostro nella figura 64, è possibile ricavare il throughput effettivo misurato, che risulta pari a 800 kbit/s, ancora al di sotto di 1.4 Mbit/s (risultato

analitico calcolato secondo [20]), e ben sotto i 3 Mbit/s nominali del MODO 1, come si può vedere dalla figura 63.

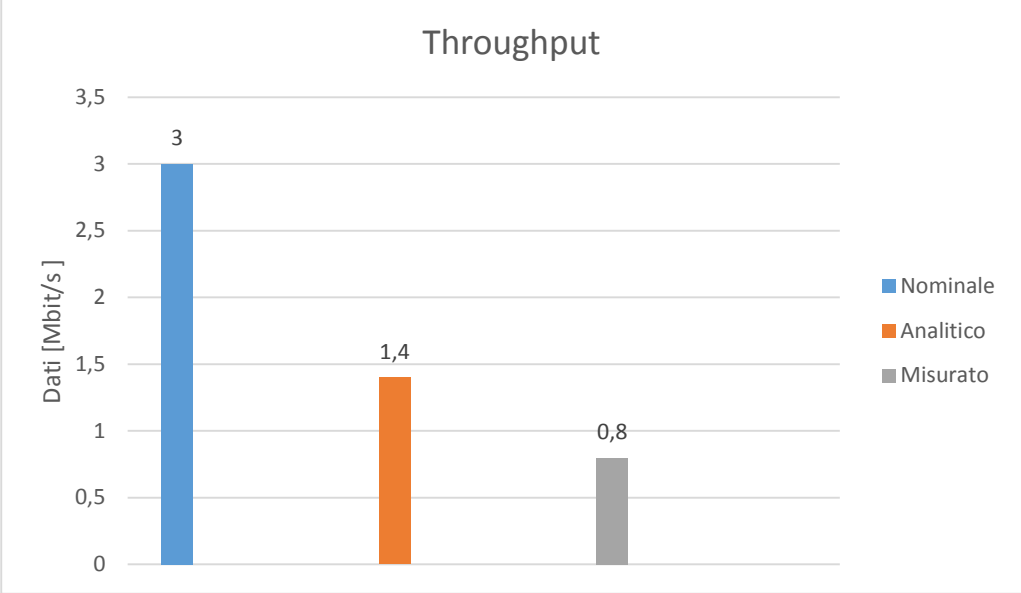

Figura 63: Differenze tra throughput nominale, analitico e misurato

Mostro il grafico di throughput relativo a tali misure. Sull'asse delle ascisse sono visualizzati 2200 ms, sull'asse delle ordinate sono visualizzati 1.9 Mbyte.

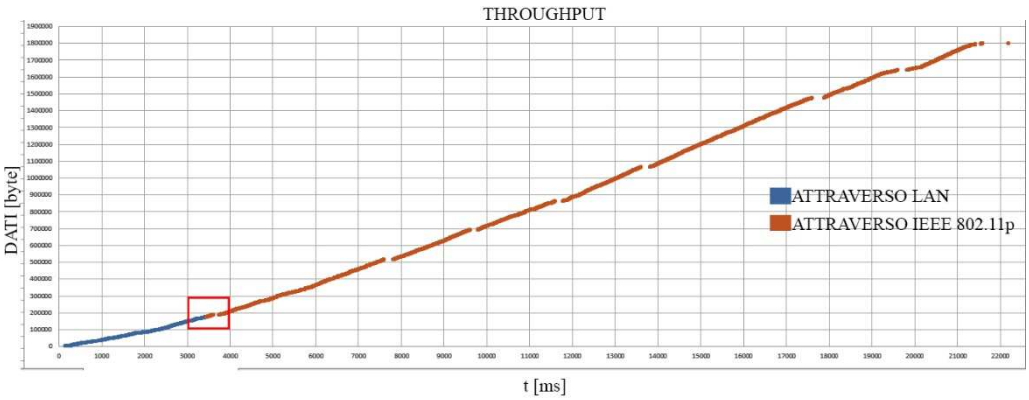

Figura 64: Throughput con offloading

La prima parte dei pacchetti viene trasmessa tramite rete LAN, la seconda invece è trasferita dopo aver effettuato l'offloading su rete 802.11p. Il grafico successivo mostra il dettaglio delimitato dal rettangolo rosso in figura 64.

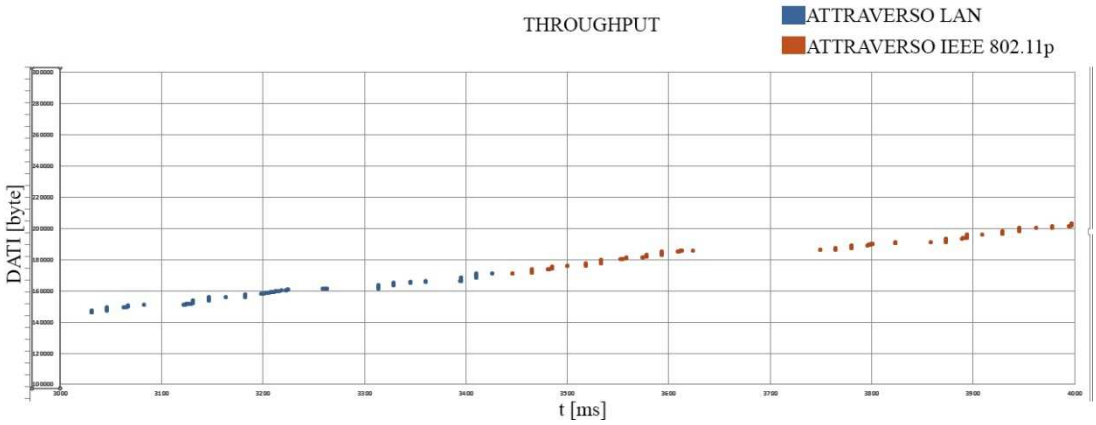

Figura 65: Throughput con offloading - dettaglio

In questo grafico, sull'asse delle ascisse sono rappresentati 1000 ms, sull'asse delle ordinate sono visualizzati invece 200 kbyte. Si può notare il passaggio da rete infrastruttura a trasmissione 802.11p.

Errori nelle misure di Offloading

Sono state effettuate misure di offloading dalla rete Wi-Fi della Scuola di Ingegneria e Architettura di Bologna alla rete 802.11p, impostando la variabile *throughNetServerAddress* con un adeguato indirizzo IP, coinvolgendo quindi più di un computer. Tuttavia ci sono elevate possibilità che, in un istante del tutto casuale, il software decodifichi in maniera errata un pacchetto e, piuttosto che scartarlo, chiude la socket e quindi il collegamento. Si sta tuttora studiando il modo per evitare tale blocco.

CONCLUSIONI

Lo scopo di questa tesi era quello di sperimentare il funzionamento dei dispositivi ALIX con standard IEEE 802.11p, realizzando trasmissioni continue di pacchetti, cercando di evitare che la coda di trasmissione si svuotasse o, viceversa, andasse in saturazione. A tal fine è stato sviluppato il codice Java discusso nei capitoli precedenti. Ulteriore fine della trattazione, era effettuare alcuni test in cui da una trasmissione di pacchetti su rete strutturata si effettuava l'offloading su rete 802.11p, iniziando a percorrere la strada che porta a possibili sviluppi futuri in ambito di offloading da rete cellulare a rete IEEE 802.11p.

La prima parte della trattazione si è focalizzata sul fornire una prospettiva teorica riguardo il panorama delle reti VANET, degli standard utilizzati, del GCDC e degli strumenti utilizzati in laboratorio. La seconda parte si è occupata di discutere degli aspetti tecnici legati alla corretta configurazione dei dispositivi e ai modi in cui ad essi è possibile connettersi. In più, è stata fornita una panoramica sul codice Java utilizzato ed implementato per poter eseguire le misure, mostrandone poi i risultati nell'ultimo capitolo.

Si è visto, durante le misure effettuate in condizioni di non visibilità, come i dispositivi Unex offrano una maggiore copertura e data-rate rispetto ai dispositivi MicroTik.

Sono state effettuate parecchie misure, con i dispositivi in visibilità diretta, per potere individuare dei limiti superiori ai valori assumibili dai parametri fondamentali per le misure, e si è visto come aumentare il valore dei pacchetti aggiunti alla coda di trasmissione all'inizio della misura causi una chiusura della socket.

Infine, sono state effettuate alcune misure eseguendo l'offloading da una rete infrastruttura a una rete 802.11p, analizzando, anche in questo caso, il limite massimo a cui i parametri potevano essere portati, in modo da garantire un'efficiente trasmissione.

Bibliografia

[1] Gayathri Chandrasekaran - VANETs: The Networking Platform for Future Vehicular Applications.

[2] Kamini Rakesh Kumar - VANET Parameters and Applications: A Review.

[3] Anna Maria Vegni, Mauro Biagi, Roberto Cusani - Smart Vehicles, Technologies and Main Applications in Vehicular Ad hoc Networks.

[4] Daniel Jiang, Luca Delgrossi - IEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments.

[5] Yunxin (Jeff) Li - An Overview of the DSRC/WAVE Technology.

[6] Leonardo Benedetti - Sperimentazione di comunicazioni veicolari tramite tecnologia IEEE 802.11p.

[7] Gianluca Spinelli - Caratterizzazione di sistemi IEEE 802.11p per comunicazioni veicolo-veicolo.

[8] Jeremy J. Blum, Azim Eskandarian, Lance J. Hoffman - Challenges of Intervehicle Ad Hoc Networks.

[9] Giovanni Toscano - Studio e analisi del protocollo 802.11p

[10] Michele Livini - Caratterizzazione di dispositivi IEEE 802.11p per comunicazione veicolo-veicolo.

[11] A. Balasubramanian - Augmenting Mobile 3G Using Wi-Fi.

[12] A. Ghosal - Mobile Data Offload: Can Wi-Fi Deliver

[13] Eyuphan Bulut, Boleslaw K. Szymanski - WiFi Access Point Deployment for Efficient Mobile Data Offloading.

[14] Informa Telecoms - Mobile Broadband Access at Home.

[15] Yongmin Choi, Hyun Wook Ji, Jae-yoon Park, Hyun-chul Kim, John A. Silvester - A 3W Network Strategy for Mobile Data Traffic Offloading.

[16] A. Olivia - IP Flow Mobility: Smart Traffic Offload for Future Wireless Networks.

[17] Francesco Malandrino, Claudio Casetti, Carla-Fabiana Chiasserini - Offloading Cellular Networks through ITS Content Download.

[18] Michele Weigle - Standards: Wave / DSRC / 802.11p

[19] http://comp.ist.utl.pt/~rmr/WSN/CaseStudies2007no/WSN_Transportation

[20] Mauro Macchia - Implementazione e Misure di Prestazioni di Sistemi WAVE/IEEE 802.11p

[21] Adnan Aijaz, Hamid Aghvami, Mojdeh Amani - A Survey on Mobile Data Offloading: Technical and Business Perspectives.

[22] Claudia Campolo, Antonella Molinaro - Multichannel Communication in Vehicular Ad Hoc Networks: A survey.

Dediche e Ringraziamenti

Dedico questa tesi:

- *Alla mia famiglia, che ringrazio per avermi dato la possibilità di crearmi un futuro, permettendomi di studiare all'università e sostenendomi fino alla fine, con il loro supporto, i loro sforzi e i loro sacrifici, ma anche con gli innumerevoli "Buonanotte, ciao, buonanotte, ciao!" parlando al telefono prima di andare a dormire, con i "Copriti, che fa freddo", con i "Mangi sempre la pasta col tonno?". Grazie anche a mia sorella, che a sua insaputa ho nutrito con dosi maggiori di pasta con il tonno, che mi è stata vicina in momenti difficilissimi, aiutandomi a star meglio. Grazie di cuore, vi voglio bene.*
- *Ai miei amici, soprattutto Ilenia, Silvia, Alberto, Gemma, con i quali ho condiviso bei momenti (e che sono stati anche vittime del mio essere stonato), compagni d'avventura e di battute (più o meno) divertenti. Grazie anche ad Alberto, per gli aiuti stilistici nella stesura della tesi.*
- *A te, Gaia. Grazie per i bei momenti, per le tue risate alle mie battute idiote (precisiamo, risate dovute al fatto che io ridessi a tali battute senza senso), grazie per la gioia che mi hai fatto provare, soprattutto grazie per avermi fatto capire cosa vuol dire Voler Bene a una persona, a prescindere dal rapporto che si ha con lei. Ti voglio bene, e sempre te ne vorrò. Grazie Doc.*
- *Ciccia, come dimenticarmi di te? Questa tesi la dedico anche ad Ambra, mia fedele Amica, la mia Timon di "Timon e Pumbaa" (indovinate chi è Pumbaa?). Mi hai fatto ridere tutte le volte che ne avevo bisogno, mi hai tirato su dal pozzo ogni volta che era necessario farlo. Grazie Amica mia, per le conversazioni quasi sempre monotematiche su una certa essenza, per le reciproche prese in giro, grazie semplicemente per esserci stata. Ti voglio bene.*

Luca.