

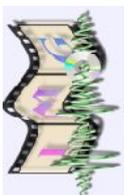


Progetto WiSafety

Deliverable finale

Internet Media Group

1. Licenza del presente documento.....	2
2. Licenza del software.....	3
3. Executive Summary.....	4
4. Descrizione tecnica del software.....	5
5. Note per l'installazione.....	10
6. Riferimenti.....	13
7. Contatti.....	14



Paolo Bucciol - Ph. D. Student, Research Engineer (paolo.bucciol@polito.it)

Politecnico di Torino - Dipartimento di AUTomatica INformatica
Corso Duca degli Abruzzi, 24 - 10129 Torino - Italia





POLITECNICO DI TORINO

Dipartimento di Automatica Informatica

1. Licenza del presente documento



**Attribuzione - Non commerciale - Condividi allo stesso modo 2.5
Italia**

Tu sei libero:

- di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare quest'opera
- di modificare quest'opera

Alle seguenti condizioni:



Attribuzione. Devi attribuire la paternità dell'opera nei modi indicati dall'autore o da chi ti ha dato l'opera in licenza.



Non commerciale. Non puoi usare quest'opera per fini commerciali.



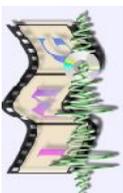
Condividi allo stesso modo. Se alteri o trasformi quest'opera, o se la usi per crearne un'altra, puoi distribuire l'opera risultante solo con una licenza identica a questa.

- Ogni volta che usi o distribuisi quest'opera, devi farlo secondo i termini di questa licenza, che va comunicata con chiarezza.
- In ogni caso, puoi concordare col titolare dei diritti d'autore utilizzi di quest'opera non consentiti da questa licenza.
- Il testo della licenza integrale è reperibile all'indirizzo

<http://creativecommons.org/licenses/by-nc-sa/2.5/it/legalcode>

Paolo Bucciol - Ph. D. Student, Research Engineer (paolo.bucciol@polito.it)

Politecnico di Torino - Dipartimento di AUTomatica INformatica
Corso Duca degli Abruzzi, 24 - 10129 Torino - Italia





POLITECNICO DI TORINO

Dipartimento di Automatica Informatica

2. Licenza del software

Il software è distribuito sotto licenza GPLv2.

È possibile reperire il testo integrale della licenza all'indirizzo

<http://www.gnu.org/licenses/gpl.txt>

Internet Media Group



Paolo Bucciol - Ph. D. Student, Research Engineer (paolo.bucciol@polito.it)

Politecnico di Torino - Dipartimento di AUTomatica INformatica
Corso Duca degli Abruzzi, 24 - 10129 Torino - Italia





POLITECNICO DI TORINO

Dipartimento di Automatica Informatica

3. Executive Summary

WiSafety: Sicurezza degli incroci grazie alle reti senza filo

Grazie all'adozione delle nuove tecnologie di trasmissione senza fili Wi-Fi e di programmi liberi, è possibile adottare nuove tecniche per aumentare la sicurezza stradale con costi contenuti.

Il progetto

Molti dispositivi portatili di larga diffusione in ambito veicolare, sia navigatori satellitari sia palmari (agende elettroniche, personal organizer etc...), dispongono ormai di un'interfaccia senza filo wireless 802.11 (WiFi). Questi dispositivi potrebbero dunque essere efficacemente sfruttati per la ricezione senza fili di messaggi di pericolo o di semplice avvertimento emessi da opportuni trasmettitori. Il progetto WiSafety prevede quindi di sfruttare la presenza e diffusione di tali dispositivi, equipaggiati di interfaccia senza filo, per incrementare la sicurezza stradale.

Il progetto è stato seguito dall'Internet Media Group, che fa capo al professor Angelo Raffaele Meo e che si occupa di media e di software libero, oltre che, dal punto di vista tecnologico, di elaborazione e trasmissione di servizi multimediali.

Scenario

Un ipotetico scenario prevede la collocazione di un dispositivo elettronico, dotato di connessione senza fili, in un punto "a rischio" della rete stradale. Tale dispositivo, che funge da *trasmettitore*, ha il compito di inviare un opportuno messaggio di pericolo che viene trasmesso a tutti i veicoli che si trovano abbastanza vicini da poterlo ricevere e che sono dotati di appositi *ricevitori* senza fili. Scelta una intersezione stradale considerata "a rischio", il progetto WiSafety prevede quindi di installarvi un dispositivo trasmettitore, che periodicamente invii dei messaggi di avvertimento circa la pericolosità di tale incrocio a dispositivi ricevitori, su cui è stato installato un apposito programma sviluppato all'interno del progetto.

Un dispositivo ricevitore, posizionato all'interno di un veicolo che si accinga ad attraversare l'incrocio, riceverà automaticamente il messaggio vocale di avvertimento riguardante il potenziale pericolo dell'incrocio stesso dal dispositivo trasmettitore. I dispositivi ricevitori utilizzati sono di facile reperibilità commerciale e di costo limitato, per consentire l'utilizzo del sistema realizzato all'interno del progetto anche su scala più ampia.

Inoltre, il programma che sta alla base di WiSafety è stato realizzato utilizzando *software libero*. Questa scelta rende possibile la modifica e la personalizzazione del sistema senza limitazioni e con le più ampie possibilità di personalizzazione.

Paolo Buccioli - Ph. D. Student, Research Engineer (paolo.buccioli@polito.it)

Politecnico di Torino - Dipartimento di AUTomatica INformatica
Corso Duca degli Abruzzi, 24 - 10129 Torino - Italia



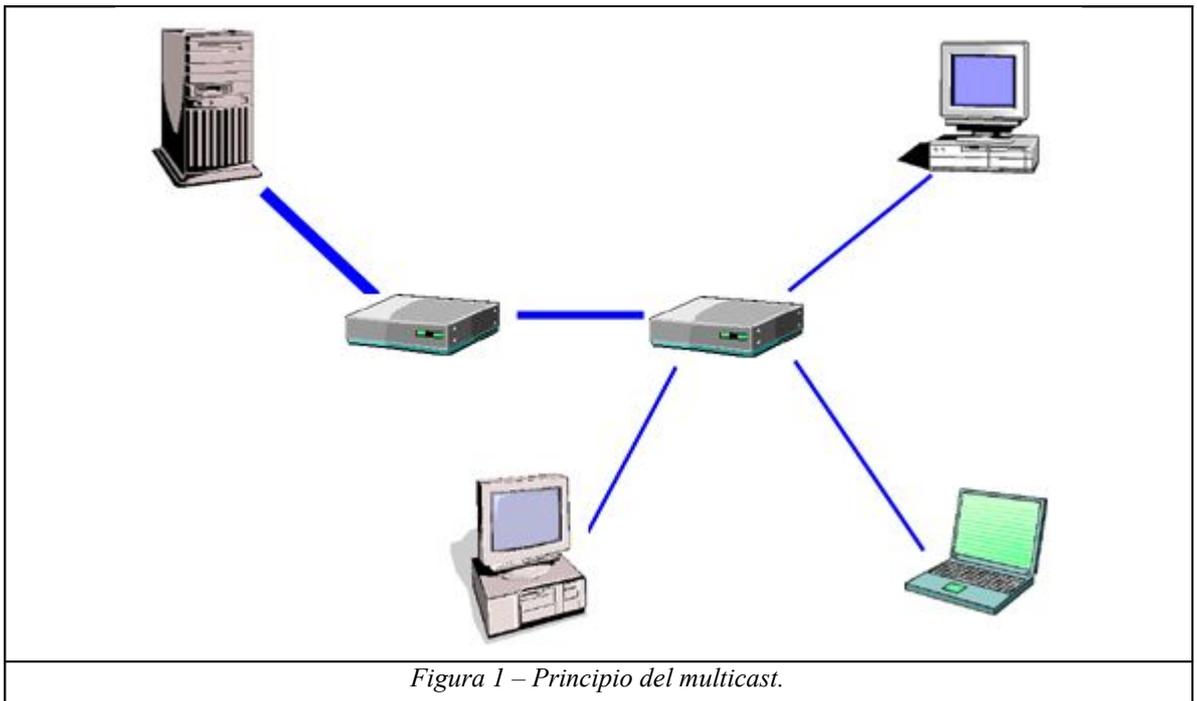


4. Descrizione tecnica del software

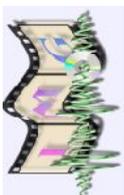
Principi di funzionamento

Il software è stato progettato seguendo la filosofia *open source*, ed è stato sviluppato in linguaggio ANSI C per massimizzarne la portabilità.

Il principio di funzionamento su cui si basa il software sviluppato è il concetto del *multicast* (rappresentato in figura 1), secondo cui ad un dispositivo che trasmette informazioni (*server*) possono essere associati un numero indefinito di dispositivi riceventi (*client*) che vogliono ricevere le informazioni da quel *server*. Per questo motivo su server e client è necessario installare due diverse parti del software.



Tutti i nodi che fanno parte della rete sono configurati secondo la filosofia *ad hoc*. I dispositivi client verranno installati all'interno delle autovetture,

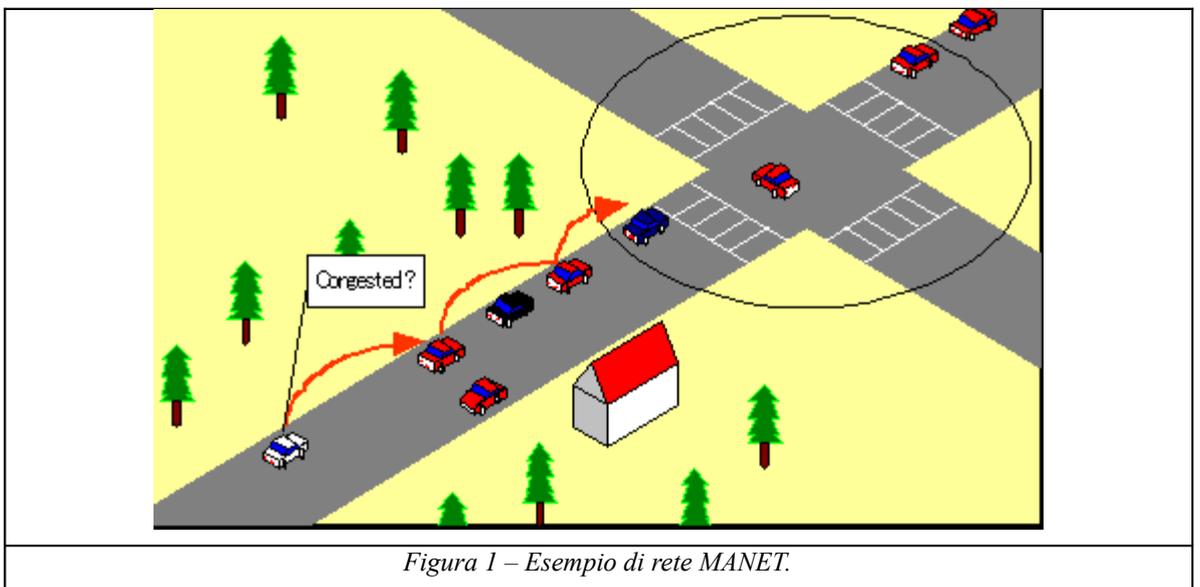




POLITECNICO DI TORINO

Dipartimento di Automatica Informatica

mentre i dispositivi server verranno posizionati negli incroci considerati a rischio. Si verrà così a formare una **rete ad-hoc mobile** (detta MANET, **Mobile Ad-hoc NETwork**)¹ [WikiAdHoc]. Un esempio di tale rete è rappresentato in figura 1.

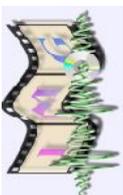


Ad ogni nodo (sia *server* che *client*) viene associato un indirizzo IP univoco all'interno della rete, in maniera tale che due nodi qualsiasi possano parlare tra di loro senza bisogno di nodi intermediari.

Il software si divide in due parti: parte *client*, da installare nel dispositivo palmare; parte *server*, da installare all'interno dell'*access point*.

Il software è stato progettato per essere eseguito in ambiente *Linux*. È possibile installare ed eseguire con successo il software su ogni dispositivo su cui possa essere installato un ambiente *Linux*. In fase di scrittura del codice si sono utilizzate delle *utility* di autoconfigurazione (autoconf), in maniera da automatizzare la riconfigurazione del software nel caso in cui sia necessario cambiare l'architettura di destinazione.

¹ Una MANET è definita come un sistema autonomo di router (detti anche nodi) mobili connessi mediante collegamenti wireless. Tutti i nodi del sistema collaborano al fine di instradare i pacchetti nel modo corretto. A causa della mobilità imprevedibile dei nodi, la topologia di rete può cambiare costantemente. Le reti Ad-Hoc vengono costruite all'occorrenza ed utilizzate in ambienti estremamente dinamici, non necessariamente con l'aiuto di una infrastruttura già esistente.





POLITECNICO DI TORINO

Dipartimento di Automatica Informatica

Descrizione del software lato server

Il programma lato server ha il compito di inviare in *multicast* uno stream MP3.

Esempio di lancio del programma lato server

```
mp3streamer -p 1234 -a 224.124.0.1 -f nomefile.mp3 [-l]
```

I principali parametri:

- **mp3streamer** è il nome dell'eseguibile;
- **-a** indica l'indirizzo (solitamente di multicast) su cui inviare il flusso mp3;
- **-p** indica la porta da utilizzare;
- **-f** specifica il file mp3 da utilizzare per lo streaming;
- **-l** (opzionale) indica di trasmettere in *loop* il file specificato con l'opzione -f.

Installazione su computer portatile

Architettura su cui è stato compilato ed eseguito il software:

- Architettura Pentium, 32-bit (386,486,Pentium I, II, III, IV, Centrino)

Ecco il dettaglio dei dispositivi su cui è stato installato il software:

- PC portatile – Pentium IV 2.4GHz
- PC portatile – Centrino 750 (1.86GHz)





POLITECNICO DI TORINO

Dipartimento di Automatica Informatica

Descrizione del software lato client

Il software da installare sul *client* è stato progettato tenendo in considerazione alcuni requisiti fondamentali. Prima di tutto, deve essere eseguito su architetture molto differenti. In secondo luogo, si richiede velocità ed efficienza nell'esecuzione; infine, l'ultimo requisito è la *stabilità*.

Il compito del software lato client è quello di ricevere il flusso multimediale inviato dal server e di riprodurlo nella maniera più fluida possibile. Sono stati effettuati diversi test per l'ottimizzazione della riproduzione, anche in collaborazione con il CSP.

Esempio di lancio del programma lato client

```
mp3receiver -p 1234 -a 224.124.0.1
```

I principali parametri:

- **mp3receiver** è il nome dell'eseguibile;
- **-a** indica l'indirizzo (solitamente di multicast) su cui è stato inviato il flusso mp3;
- **-p** indica la porta da utilizzare.

Installazione su computer portatile

Architettura su cui è stato compilato ed eseguito il software:

- Architettura Pentium, 32-bit (386,486,Pentium I, II, III, IV, Centrino)

Ecco il dettaglio dei dispositivi su cui è stato installato il software:

- PC portatile – Pentium IV 2.4GHz
- PC portatile – Centrino 750 (1.86GHz)

Installazione su dispositivo palmare

Architettura su cui è stato compilato ed eseguito il software client:

- Architettura ARM (PXA250,PXA255)



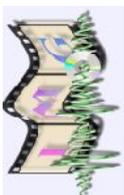
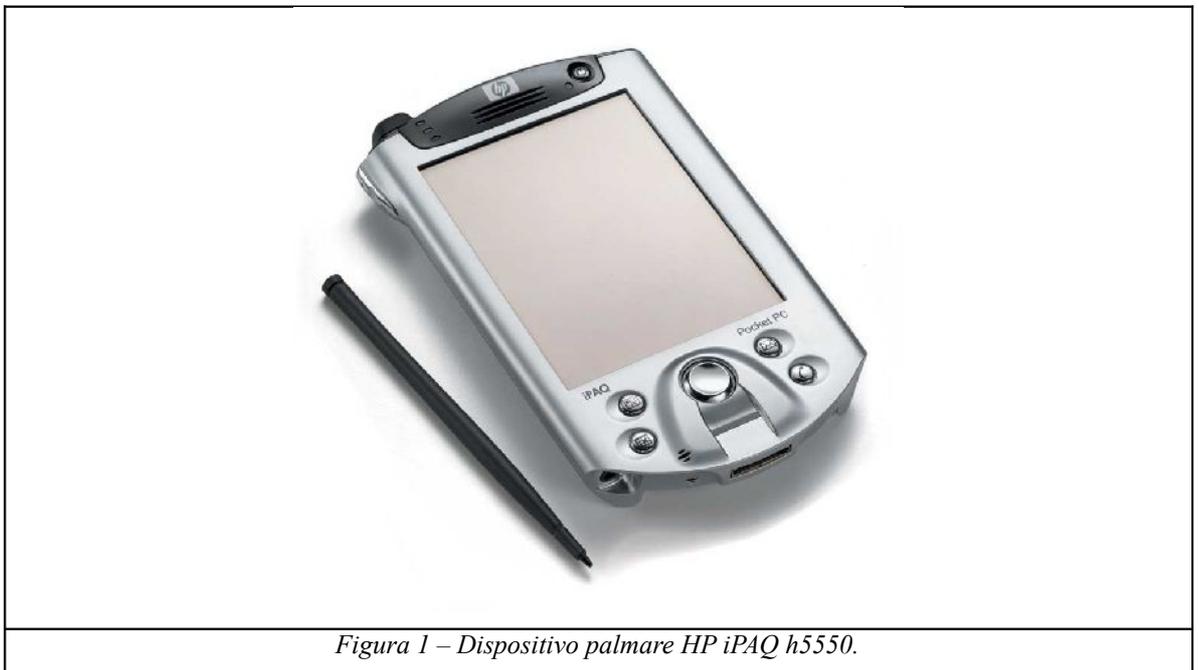


POLITECNICO DI TORINO

Dipartimento di Automatica Informatica

Questo il dettaglio del dispositivo su cui è stato installato il software client (rappresentato in figura 1):

- Palmare HP iPAQ h5550 – PXA 255 (RISC @ 400MHz)





POLITECNICO DI TORINO

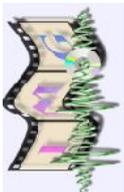
Dipartimento di Automatica Informatica

5. Note per l'installazione

Compilazione del software per pc portatile

I pacchetti da cui dipende la corretta compilazione del software, da installare sul PC su cui si effettuerà la compilazione, sono i seguenti:

- libao
- libao-dev
- alsa-oss
- alsa-base
- libgtk2.0-dev
- libgtk2.0-common
- libgtk2.0-0
- libgtkspell0
- pthread
- libmad0
- libmad0-dev
- autoconf tools > 1.7
- automake
- m4
- make
- gcc > 3.2





POLITECNICO DI TORINO

Dipartimento di Automatica Informatica

Crosscompilazione

Per installare il software su dispositivo palmare, è caldamente consigliato il metodo della *crosscompilazione*. Con questa tecnica, mediante la quale è possibile compilare su personal computer il programma per palmare, si semplifica la procedura di compilazione e si velocizzano considerevolmente le operazioni.

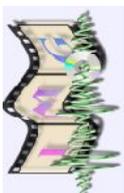
Si consiglia a questo scopo di installare l'ambiente di compilazione SCRATCHBOX versione 1.0 o successiva [Scratchbox].

In questo caso, i pacchetti richiesti sono:

- scratchbox-core
- scratchbox-devkit-debian
- scratchbox-libs
- scratchbox-toolchain-arm*

Si noti che è possibile scaricare una *toolchain* completa per ARMv5, senza quindi la necessità di costruire *toolchain* ad hoc.

N.B.: I palmari serie HP iPAQ supportati da questa toolchain sono i seguenti: h38xx, h39xx, h54xx, h55xx, h22xx





POLITECNICO DI TORINO

Dipartimento di Automatica Informatica

Software scaricabile dal sito del progetto

Sul sito del progetto <http://media.polito.it/WiSafety> è possibile scaricare i seguenti pacchetti:

- wisafety_v1.tar.gz sorgenti di tutta la suite client-server
in mp3streamer/ si trova il server
in mp3receiver/ si trova il client
- mp3receiver_386 client compilato per Intel 386 e compatibili
- mp3streamer_386 server compilato per Intel 386 e compatibili
- mp3receiver_arm client compilato per ARMv5 e compatibili
- mp3streamer_arm server compilato per ARMv5 e compatibili
- gpe_wisafety_image_5550 immagine (jffs2) funzionante per iPAQ h5550
basata su Linux Familiar 0.8.3 con GPE 2.6,
su cui è stato installato il client mp3receiver

Note sull'installazione di Linux su iPAQ

Per l'installazione di Linux sui palmari HP iPAQ, si rimanda al sito della distribuzione Familiar [Handhelds], quella su cui è stato testato il software lato client.





POLITECNICO DI TORINO

Dipartimento di Automatica Informatica

6. Riferimenti

- [WikiAdHoc] http://it.wikipedia.org/wiki/Rete_ad-hoc_mobile
Fonte: Wikipedia (edizione italiana)
- [Scratchbox] <http://scratchbox.org> (Sito ufficiale)
www.scratchbox.org/debian/ (Repository apt per debian)
- [Handhelds] <http://handhelds.org> (Sito ufficiale)





POLITECNICO DI TORINO

Dipartimento di Automatica Informatica

7. Contatti



Sito del progetto:

<http://media.polito.it/WiSafety>



Ing. Federico Ridolfo

federico.ridolfo@polito.it

Ing. Paolo Bucciol

paolo.bucciol@polito.it

+39 328 2498106

Prof. J. C. De Martin

demartin@polito.it

+39 011 564 7065

Internet Media Group



Paolo Bucciol - Ph. D. Student, Research Engineer (paolo.bucciol@polito.it)

Politecnico di Torino - Dipartimento di AUTomatica INformatica
Corso Duca degli Abruzzi, 24 - 10129 Torino - Italia

