Wifimon report

Daniele Paolini[1], Marco Venturini[2], Lorenzo Vannucci[3]

Abstract
In questo articolo sono spiegate le fasi di progettazione e realizzazione del tool open source *wifimon*: un'applicazione scritta in linguaggio C (POSIX standard) orientata al monitoraggio di reti, utilizzabile per filtrare e analizzare *beacon frame* in modalità live e offline, al fine di determinare la qualità istantanea e nel tempo delle reti WLAN IEE 802.11.

Keywords
WLAN Networks — Monitoring Tool — Beacon Frames

[1] *Corso di studi di Informatica, Università di Pisa, Pisa, Italia, daniele.paolini@hotmail.it*
[2] *Corso di studi di Informatica, Università di Pisa, Pisa, Italia, alexander00@hotmail.it*
[3] *Corso di studi di Informatica, Università di Pisa, Pisa, Italia, ucci.dibuti@gmail.com*

Contents

Introduzione

L'analisi qualitativa di una rete WLAN parte dalla cattura di alcuni pacchetti che circolano al suo interno, in questo caso ciò che ci interessa sono dei frames particolari, i *beacon frames*, i quali vengono inoltrati periodicamente dagli access point di una rete WLAN al resto dei dipositivi presenti nel loro raggio di azione, al fine di annunciarsi e rendersi disponibili per l'eventuale utilizzo.

1. Beacon Frames

Sono pacchetti di dimensione variabile, la quale dipende dalla quantità di informazioni che l'access point include al loro interno. Sono composti dall'*header* e dal *beacon frame body*.

1.1 Header

Contiene tre campi di dimensione standard: *revision, pad, lenght e present flag*. Quest'ultima è una bitmap che specifica quali sono i campi variabili presenti nel pacchetto, in genere troviamo *data rate, channel frequency e ssi signal* che di fatto sono quasi uno standard in quanto, almeno nei nostri test, sono sempre stati inclusi dagli access point all'interno dei frames.

1.2 Beacon Frame Body

E' il pacchetto vero e proprio, al suo interno sono immagazzinate le informazioni sul mittente (il MAC address dell'access point che vuole annunciarsi all'interno del suo raggio di azione) ed il destinatario (MAC address broadcast ff:ff:ff:ff:ff:ff in modo da essere recepito da tutti i dispositivi nel raggio di azione). Un'altra informazione importante contenuta all'interno di questa sezione è l'SSID, ovvero il nome della rete in formato *human readable*.

2. Realizzazione ed esecuzione

2.1 Generalità

Wifimon è stato appositamente progettato per intercettare ed analizzare i beacon frames. Può funzionare in *live mode*, previa selezione dell'interfaccia di rete wi-fi da utilizzare, in modo da analizzare istantaneamente tutti gli access points e le reti nel raggio di azione dell'utente. Inoltre è in grado di funzionare anche in modalità *offline*, previa selezione di un file in formato pcap contenente una cattura del traffico di rete. L'applicativo è stato progettato ed ideato principalmente per essere eseguito su macchine con sistema operativo POSIX, GNU-LINUX in primis, poichè è l'ambiente esecutivo e di sviluppo con il quale abbiamo maggiore esperienza. L'applicativo è in grado di funzionare anche su macchine con S.O. Apple OS-x©, ma solamente in modalità *offline*, per quanto riguarda O.S. Windows©la compatibilità è, al momento, totalmente negativa, in futuro però abbiamo intenzione di effettuare un porting completo del tool su tutte le piattaforme.

2.2 Preparazione dell'ambiente di esecuzione

Per compilare il codice sorgente, oltre ad un compilatore C standard (ad esempio GCC), è necessario prima aver installato

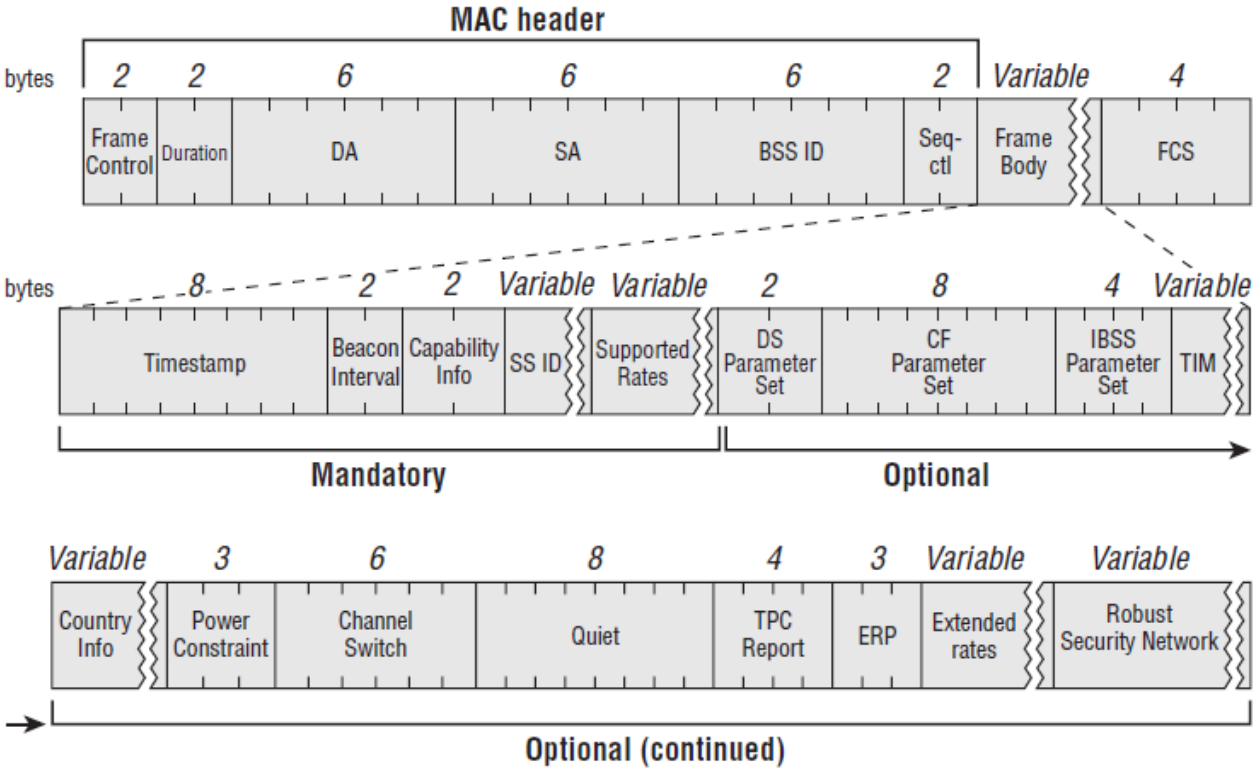

Figure 1. Struttura di un *Beacon Frame*

la libreria *libpcap* che fornisce il fondamentale supporto al programma.
Per far funzionare il programma in modalità *live* è necessario avere i permessi di root sulla macchina in modo da poter impostare la modalità *monitor* sulla scheda di rete, inoltre è necessario avere installato una scheda di rete che supporti tale modalità. Per il funzionamento *offline* invece non sono necessari i permessi di root o schede di rete che supportino la modalità *monitor*. Il pacchetto software include anche un *Makefile* per l'installazione e la rimozione automatizzata dell'applicazione: basta eseguire

```
$ make install <installazione>
$ make clean <disinstallazione>
```

Monitor mode setting Per configurare la scheda di rete eseguire i seguenti comandi in una shell root:

```
$ sudo su
$ <inserire la propria password>
$ iwconfig <scegliere la periferica>
$ ifconfig <nome scheda> down
$ iwconfig <nome scheda> mode monitor
$ ifconfig <nome scheda> up
```

Managed mode setting Per ripristinare la scheda di rete in modalità *managed* eseguire i seguenti comandi in una shell root:

```
$ sudo su
$ <inserire la propria password>
$ iwconfig <scegliere la periferica>
$ ifconfig <nome scheda> down
$ iwconfig <nome scheda> mode managed
$ ifconfig <nome scheda> up
```

Opzioni Il tool può essere eseguito con diverse opzioni. Per la modalità *live* basta seguire le seguenti istruzioni:

```
$ ./wifimon -l <nome scheda>
```

Il programma stamperà a video le statistiche temporanee (relative a tutte le reti WLAN presenti nell'area interessata) ogni 100 beacon frames intercettati ed analizzati. Terminando il programma con CTRL+C (SIGTERM) verranno visualizzate invece le statistiche finali relative a tutto il periodo di osservazione.
Per eseguire l'applicazione in modalità *offline* basta digitare in una shell i seguenti comandi:

```
$ ./wifimon -i <nome file>
```

Il programma eseguirà l'analisi nei beacon frames presenti nel pacchetto e stamperà a video direttamente le statistiche

finali.

In ambedue le modalità, le statistiche finali sono organizzate gerarchicamente: prima sono raggruppati i diversi SSID intercettati e per ognuno di essi sono raggruppati gli Access Point che forniscono l'accesso alla WLAN con le relative caratteristiche:

1. mac address
2. data rate medio (Mbps)
3. ssi signal medio (Dbm)
4. canale
5. data rate massimo e relativo timestamp
6. data rate minimo e relativo timestamp
7. ssi signal massimo e relativo timestamp
8. ssi signal minimo e relativo timestamp

n.b. il data rate espresso in Mbps non è da considerarsi effettivo per la rete analizzata, ma solo indicativo, in quanto, dai test effettuati da noi in diversi ambienti, i beacon frame sono recepiti con una velocità media molto inferiore (circa 5-6 volte) rispetto agli altri protocolli di comunicazione WLAN.

Sia in *live mode* che *offline* oltre alla classica opzione "-h" che stampa un messaggio di aiuto e le istruzioni, è possibile selezionare due opzioni per esportare i pacchetti catturati in un file di testo (opzione -o nome-file), oppure esportare le statistiche finali nel file "report.json" in formato JSON (opzione -j) per essere utilizzate in un secondo momento per produrre grafici o ulteriori statistiche.

2.3 Main flow

Libreria pcap Wifimon, come già detto, utilizza la libreria *pcap* al fine di:

1. Filtrare i pacchetti di tipo beacon utilizzando le funzioni pcap_compile() e pcap_setfilter()
2. Utilizzare l'interfaccia di rete predisposta con pcap_open_live()
3. Ottenere la propria netmask ed il proprio indirizzo ip con pcap_lookupnet()
4. Tramite pcap_loop() eseguire in loop la funzione packet_callback() che è il motore vero e proprio del sistema di cattura.
5. Scansionare i dispositivi di rete mediante la funzione pcap_findalldevs()

Capture engine Il cuore dell'applicazione è senza ombra di dubbio la funzione packet_callback(), che viene chiamata dalla pcap_loop(), la quale provvede ad estrarre le informazioni citate nella sezione precedente localizzando i campi grazie alla lunghezza del *radiotap header* reperibile in un campo standard del *beacon header*.

Funzionamento In base alle opzioni scelte dall'utente il programma segue due flussi diversi. Se è stata scelta la modalità *live* l'applicazione ricava l'indirizzo IP e la Netmask della scheda di rete utilizzata, imposta i filtri per catturare unicamente i beacon frames, installa un handler per gestire correttamente il SIGTERM e lancia la pcap_loop() indicando di gestire i pacchetti in arrivo con la funzione packet_callback() che stampa a video le statistiche temporanee ogni 100 pacchetti, una volta ricevuto il segnale con la pcap_breakloop() viene interrotto il ciclo e vengono stampate le statistiche finali. Nella modalità *offline* invece della scheda di rete viene fornito alla pcap_loop() l'indirizzo del file contenente la cattura dei pacchetti, indicando sempre di gestire i pacchetti in arrivo con la funzione packet_callback() che però non esegue il calcolo delle statistiche temporanee ma stampa direttamente le statistiche finali una volta raggiunta la fine del file.

2.4 Disclaimer

Questo software open source è stato sviluppato per analizzare la qualità della connessione e l'intensità del segnale delle reti WLAN, sia istantaneamente, sia a fini statistici in un arco temporale esteso (con i test ci siamo spinti fin'ora a nove ore di monitoraggio continuo senza riscontrare alcun problema). Il pacchetto software è distribuito sotto licenza GNU GPL, non viene fornita alcuna garanzia assieme ad esso, l'utilizzatore è libero di modificare, in parte o totalmente ogni parte del codice sorgente citando gli autori nel proprio codice. Per proporre estensioni o modifiche nella versione ufficiale si prega di contattare gli autori agli indirizzi email citati nell'intestazione dell'articolo. Gli sviluppatori non sono responsabili per un eventuale utilizzo improprio di questo tool.

Talk is cheap, show me the code. L.Torvalds

Riconoscimenti

Un ringraziamento speciale da/a tutti i membri del team[1] di sviluppo, ed al prof. Luca Deri - Università di Pisa per le lezioni teoriche e pratiche.

References

[1] Lorenzo Vannucci. Daniele Paolini, Marco Venturini. Progettazione e realizzazione dell'applicazione "wifimon" - wlan monitoring tool. *Unipi*, 2015.