IEEE 802.21: Media Independent Handover

Francesco Soncina

Bologna, 18 Marzo 2014

Tesi di Laurea in Informatica

Relatore:
Chiar.mo Prof.
Vittorio Ghini

Presentata da:
Francesco Soncina

Indice

Tipi di handovers

L'*Handover* è l'atto di cambiare l'*access point* al fine di mantenere la connettività e può essere catalogato secondo due differenti criteri:

1. Tecnologia prima e dopo il passaggio
   - Horizontal handover (intra-rat)
   - Vertical handover (inter-rat)
2. Stato della sessione prima e dopo il passaggio
   - Hard handover (break-before-make)
   - Soft handover (make-before-break)

Graficamente

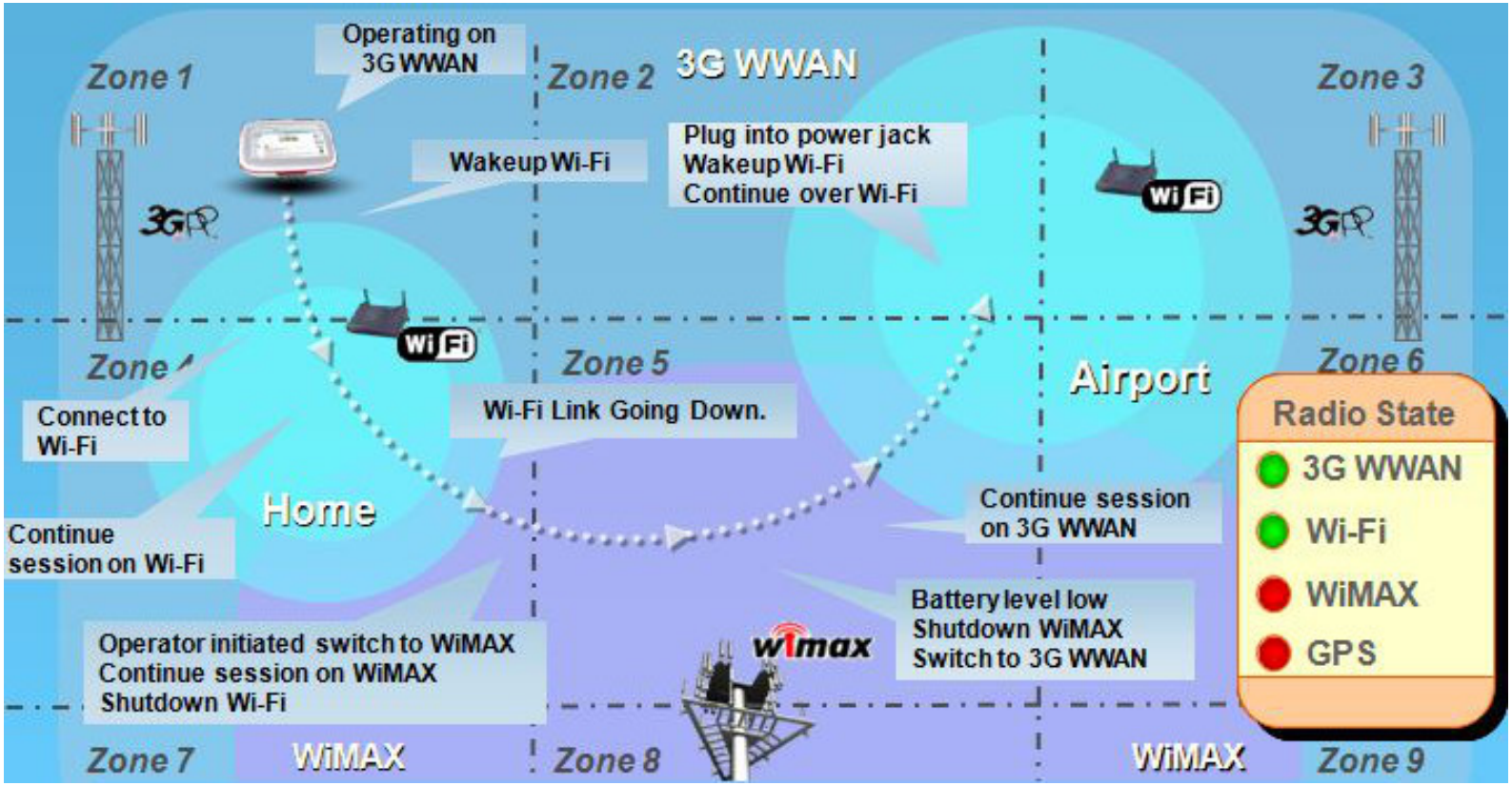

Figura: Esempi di *handovers*

Problematiche

Ogni giorno ci sono sempre più *devices* dotati di più interfacce di rete e quindi la gestione del processo di *handover* è diventata sempre più importante. I principali problemi da gestire sono:

1. Decisioni di handover (IEEE 802.21)
2. Mantenimento sessione in handovers nella stessa rete (IEEE 802.11r-2008)
3. Mantenimento sessione in handovers tra reti diverse (IETF Mobile IPv4/IPv6)

Finalità

Lo standard IEEE 802.21 si assume l'onere di risolvere il primo dei precedenti problemi, ovvero come debbano avvenire le decisioni di *handover*, in particolare:

- raccolta delle informazioni necessarie sullo stato dei vari collegamenti disponibili
- propagazione di eventi e comandi alle entità interessate
- logica di funzionamento *media-independent*

Architettura

Le entità previste dallo standard sono:

1. *Media Independent Handover Function* (MIHF): è il *core* dello standard
   - Media Independent Event Services (MIES): gestisce propagazione eventi
   - Media Independent Command Services (MICS): gestisce propagazione comandi
   - Media Independent Information Services (MIIS): gestisce propagazione *queries* e relative risposte
2. *Link_Sap*: forniscono dei servizi ad un MIHF (e.g. generazione di eventi, esecuzione di comandi, etc)
   - *media-specific*: gestiscono solo una tipologia di interfaccia
   - *media-independent*: forniscono un'astrazione generale per ogni tipo di interfaccia
3. *MIH-User*: è l'entità che sfrutta i servizi offerti da un MIHF.

Funzionalità implementate

ODTONE è una implementazione dello standard IEEE 802.21 *open-source* (LGPLv3), resa *cross-platform* attraverso la libreria Boost (eseguibile su sistemi GNU/Linux, Windows, Android ed OpenWrt) e fornisce:

- MIHF: fornisce un'implementazione del core e supporta la propagazione di ogni tipo di evento e comando
- Link_Sap generico: è compatibile con ogni famiglia di interfacce, ma supporta solo eventi *Link_Up* e *Link_Down*
- Link_Sap specifico per 802.3/802.11: supporta più eventi e comandi, ma solo per 802.3 e 802.11
- MIH-User: fornisce un esempio di applicazione utente che possa comunicare con l'MIHF
- libodtone: è la libreria ufficiale per interagire correttamente con ODTONE a livello applicativo

Pro e contro

Pro

- cross-platform
- open-source
- facilmente espandibile
- il core supporta la propagazione di tutti gli eventi e comandi

Contro

- il SAP generico genera solo gli eventi *Link_Up* e *Link_Down*
- sono forniti SAP specifici solo per 802.3 e 802.11
- i SAP specifici non implementano tutti gli eventi e comandi previsti
- non sono forniti SAP per nessuna tecnologia della famiglia 3GPP

Obbiettivi

Per poter testare sia lo standard IEEE 802.21 sia l'implementazione ODTONE è stata realizzata un'applicazione in grado di rimanere in ascolto degli eventi generati dall'MIHF al fine di realizzare un proxy ad alta affidabilità, i.e. in grado di utilizzare più interfacce di rete per gestire una singola connessione, ed è stato realizzato in due versioni:

- **Simplex**: è in grado di gestire una comunicazione di sola uscita, sfruttando le interfacce disponibili secondo la politica di *scheduling* adottata. (il destinatario non deve considerare l'IP sorgente)
- **Full-duplex**: è in grado di gestire una comunicazione bidirezionale, sfruttando le interfacce disponibili sia per ricevere sia per inviare, per entrambi i *peers*.

Graficamente

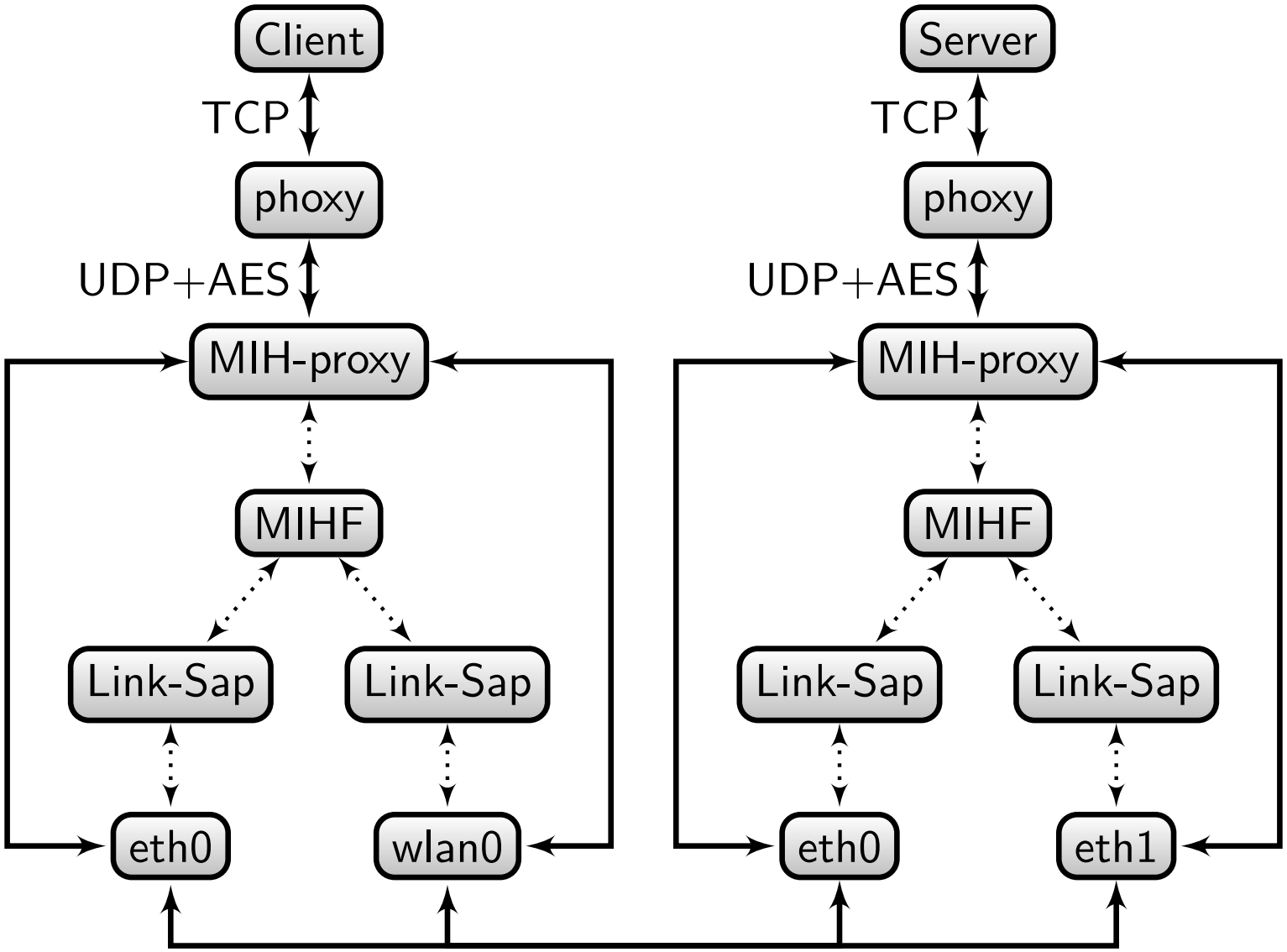

Figura: MIH-proxy bidirezionale con phoxy

Internals

- **Unidirezionale**: rimane in ascolto di eventi dell'MIHF per stabilire quali interfacce sono disponibili e invia tutto ciò che riceve dal client verso il server utilizzando potenzialmente anche più interfacce contemporaneamente, a seconda della politica di *scheduling* implementata. Gestisce solo pacchetti UDP, la comunicazione è in un sol verso e quindi basta una sola istanza dell'MIH-proxy.
- **Bidirezionale**: rappresentato in figura 2, per poter trasferire traffico bidirezionale sono necessarie due istanze dell'MIH-proxy, i.e. una per *peer*. Questa versione è in grado di poter inviare i pacchetti a più destinatari attraverso più interfacce. Per sapere lo stato delle interfacce locali utilizza lo standard IEEE 802.21 mentre per lo stato delle interfacce del destinatario utilizza un sistema di *heartbeats* temporizzati. Per abilitare il supporto per flussi TCP è necessario aggiungere un altro programma, *phoxy*, incaricato di trasformare lo *stream* in datagrammi, gestire il loro rinvio in caso andassero persi, il loro riordinamento in caso arrivassero disordinati e può, a richiesta, cifrare i datagrammi con l'algoritmo a chiave simmetrica *AES256*.

References

Repository di questo lavoro reperibile con i seguenti comandi:

```
$ git clone https://github.com/phra/802_21.git
```

oppure

```
$ wget https://github.com/phra/802_21/archive/master.tar.gz
```

IEEE 802.21 Working Group (2008)

IEEE 802.21: Media Independent Handover Services

http://standards.ieee.org/getieee802/download/802.21-2008.pdf

ODTONE Team (2009-2014)

ODTONE

http://atnog.github.io/ODTONE/