Alma Mater Studiorum · Università degli Studi di Bologna
SCUOLA DI SCIENZE
Corso di Laurea in Informatica

Transmission Error Detector per Wi-Fi su kernel Linux 4.0

Relatore:
Chiar.mo Prof.
Vittorio Ghini

Presentata da:
Gabriele Di Bernardo

Sessione I

Anno Accademico 2014/2015

Scenario

- Mobile device multi-homed
- Wi-Fi

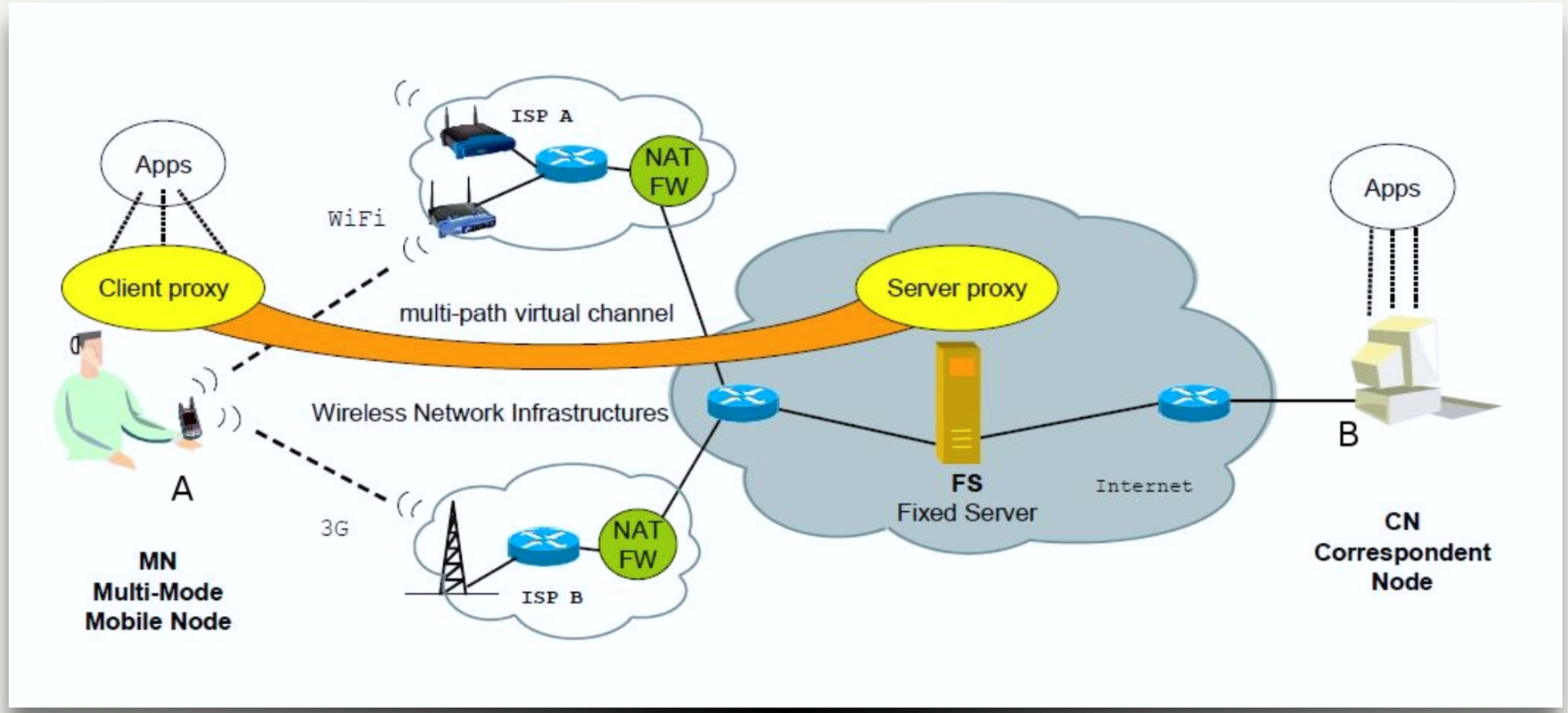
- ***Real-time*** multimedia oriented app (ad esempio un app VoIP, solitamente implementate over UDP)

Always Best Packet Switching

- Architettura distribuita per consentire ad un app in esecuzione su un device mobile multi-homed di sfruttare tutte le sue interfacce di rete

- Ciascun datagram UDP inviato sull'interfaccia di rete più *vantaggiosa*

Always Best Packet Switching



- Client proxy
- Server proxy

Transmission Error Detector

- Modulo fondamentale ABPS client proxy
- Implementato in modo cross-layer lungo il kernel del nodo mobile
- Monitora ciascun datagram UDP in invio attraverso una certa scheda wireless
- Notifica al proxy client lo stato di consegna di ogni frame trasmesso al first-hop (First-hop Transmission Notification)

Obiettivo

- Estendere kernel Linux 4.0 con un modulo Transmission Error Detector per Wi-Fi

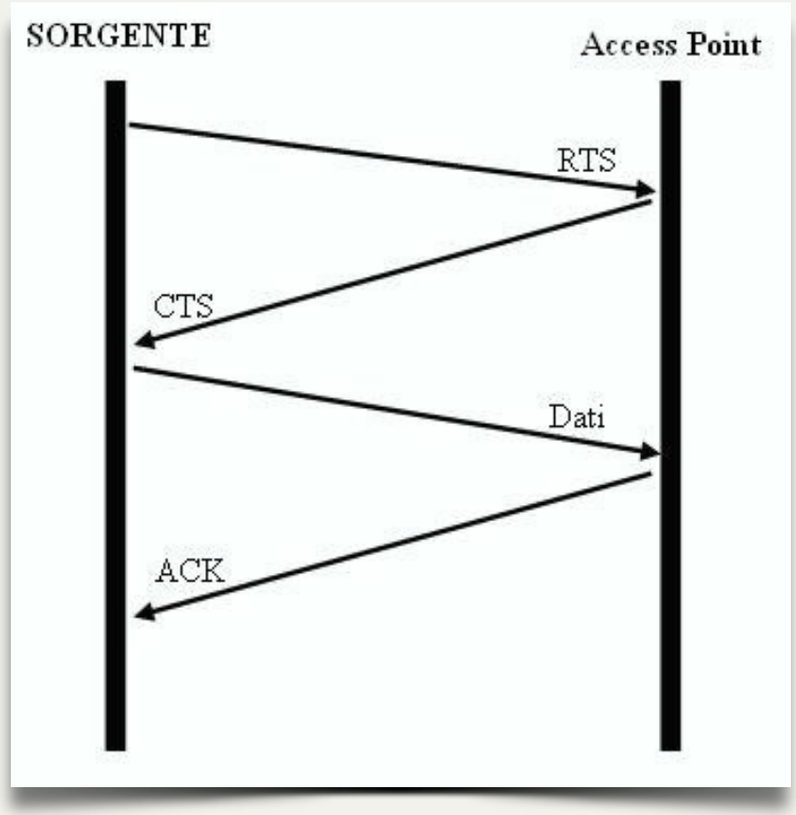
- Definire un’interfaccia che consenta ad una qualsiasi app in esecuzione sul sistema ottenuto di sapere se i pacchetti trasmessi sono stati consegnati o meno al primo AP

- Modulo TED precedentemente sviluppato per kernel Linux 2.6.30-rc5

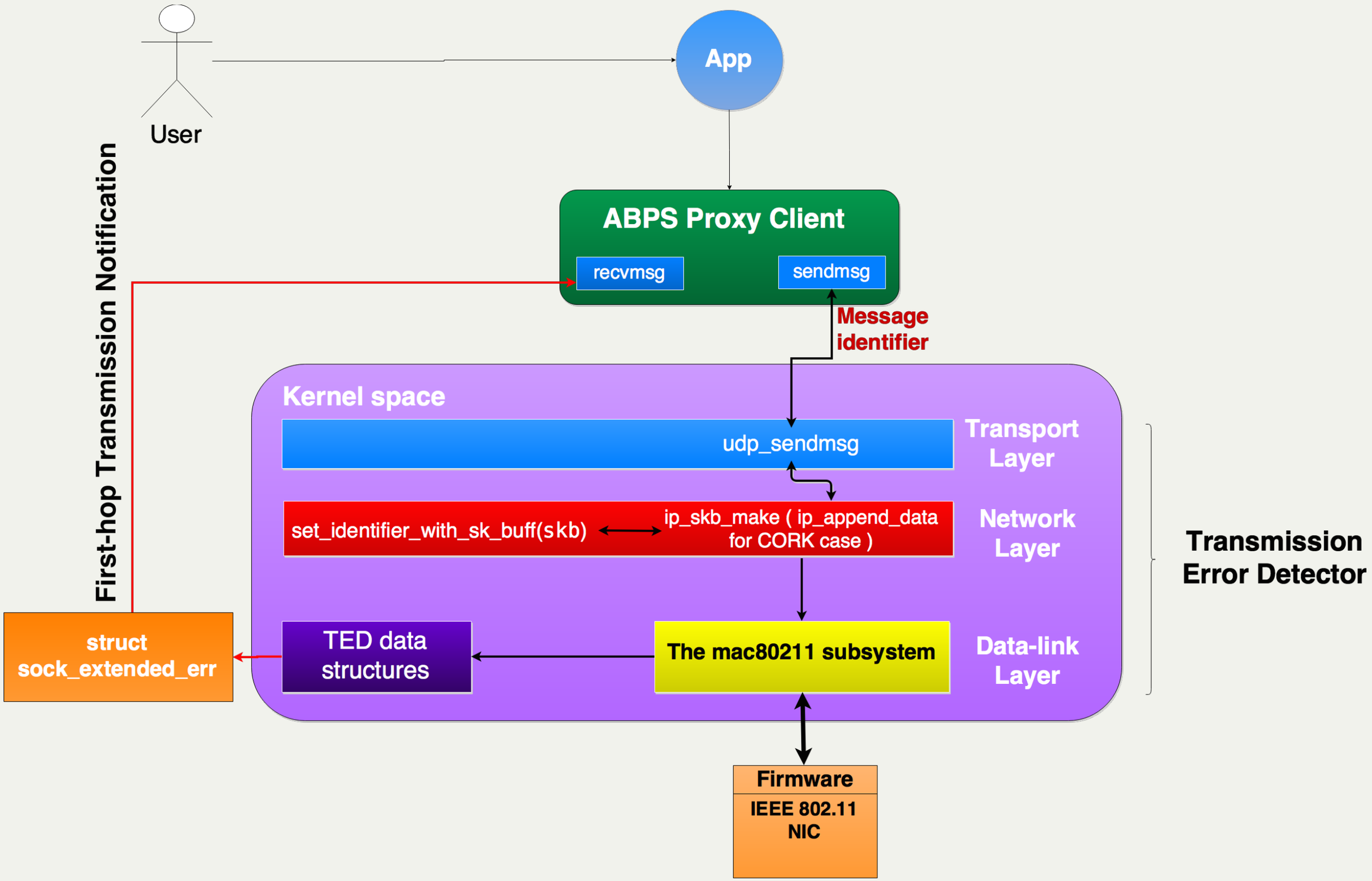
IEEE 802.11

Nelle trasmissioni Wi-Fi l'accesso al mezzo trasmissivo è regolamentato dal protocollo CSMA/CA

A trasmissione dati completata il nodo sorgente si pone in attesa di ricezione di un ACK da parte della stazione ricevente.



Implementazione



Implementazione - user space

- App necessita di un identificativo per monitorare ogni messaggio in invio
- Identificativo messaggio utile a contestualizzare notifiche provenienti da TED
- Utilizzo system call sendmsg
- App specifica in ancillary data puntatore di memoria in user space

Implementazione - kernel space

Struttura socket buffer

```
struct sk_buff
{
 ..
 ..
 /* new field added */
 uint32_t sk_buff_identifier;
};
```

Implementazione - Transport layer

- Trasmissione UDP controllo ceduto a udp_sendmsg

- Parsing ancillary data

- Allocazione struttura **sk_buff** moduli livello rete

- Identificativo passato ad app richiedente attraverso indirizzo di memoria specificato negli ancillary data

Implementazione - Network layer

- Introduzione contatore per identificare ciascun messaggio in invio

- Contatore incrementato per ogni messaggio in invio

- Valore contatore assegnato a socket buffer

- Frammentazione, ciascun frammento mantiene l’identificativo pacchetto originario

Implementazione - mac80211 subsystem

- Sottosistema mac80211 realizza logica di creazione e gestione frame 802.11 per device softMAC

mac80211 subsystem - Invio frame

- TED mantiene informazioni principali frame in invio in apposite strutture dati

- Informazioni associate a campo *sequence control* header IEEE 802.11

mac80211 subsystem - Ricezione ACK

- Sottosistema mac80211 notificato sullo status di consegna frame

- Accesso header IEEE 802.11 ed utilizzo sequence control come chiave di ricerca all'interno strutture dati mantenute da TED

- Sollevata First-hop Transmission Notification contente informazioni memorizzate da TED e informazioni relative allo stato di consegna del frame (ACK/NACK, retry count)

First-hop Transmission Notification

- App può ricevere First-hop Transmission Notification come qualsiasi messaggio di tipo ICMP

- App può leggere (tramite system call recvmsg) dalla coda degli errori del socket abilitando flag IP_RECVERR

- Messaggio di errore allocato nel kernel tramite struttura struct sock_extended_err

- TED identifier, informazioni frammentazione, informazioni trasmissione

First-hop Transmission Notification

```
struct sock_extended_err
{
  __u32     ee_errno;
  __u8      ee_origin;
  __u8      ee_type;
  __u8      ee_code;
  __u8      ee_pad;
  __u32     ee_info;
  __u32     ee_data;
  /* Transmission Error Detector new field for retry count */
  __u8    ee_retry_count;
};

#define SO_EE_ORIGIN_NONE  0
#define SO_EE_ORIGIN_LOCAL 1
#define SO_EE_ORIGIN_ICMP  2
#define SO_EE_ORIGIN_ICMP6 3
#define SO_EE_ORIGIN_TXSTATUS 4

/* Transmission Error Detector new value for ee_origin field*/
#define SO_EE_ORIGIN_LOCAL_NOTIFY 5
```

ACK/NACK

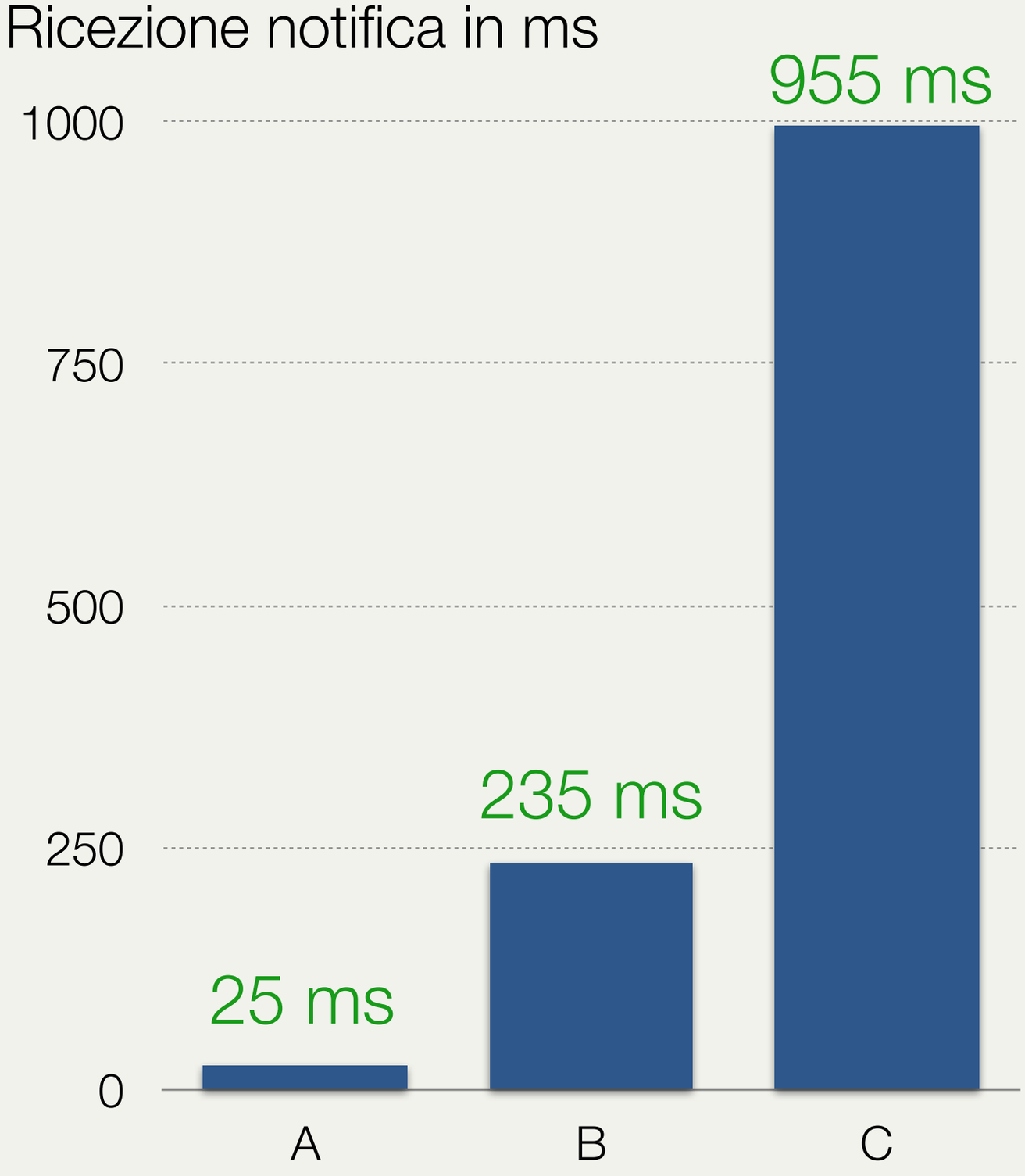
more fragment

message identifier

fragment length & offset

retry count

Valutazioni sperimentali



Traffico generato client

5000 pacchetti 1096 Byte ciascuno

- (A) 71MB trasmessi da Raspberry Pi a 40Mbps
- (B) Due host trasmettono 36MB rispettivamente a 10Mbps e 15Mbps
- (C) Tre host trasmettono 23MB rispettivamente a 10Mbps, 11Mbps e 15Mbps

Conclusioni & sviluppi futuri

Realizzazione modulo Transmission Error Detector per Wi-Fi su kernel Linux 4.0

È stata sviluppata un'app che sfrutta modulo TED su cui è stato possibile condurre test e valutazioni sperimentali

Sviluppi futuri:

- Porting piattaforma Android
- Estensione First-hop Transmission Error Notification con bit rate trasmissivo frame