

Capitolo 1

Test e valutazioni sperimentali

Dopo aver illustrato il funzionamento e l'implementazione del TED, passiamo ora ad analizzare i dati relativi alla trasmissione di pacchetti tramite la nostra applicazione. Lo scopo dei test effettuati è quello di analizzare la QoS del segnale e come può variare in diversi scenari. Questo ci è utile in quanto, in base ai risultati ottenuti, si può decidere se cambiare NIC per l'invio di determinati pacchetti.

Andremo ora a mostrare quali test sono stati effettuati. In particolare mostreremo quali dispositivi sono stati utilizzati per fare le prove, i parametri di valutazione ed i risultati ottenuti.

Lo scopo di questi test è di andare ad osservare la QoS del segnale e come può variare in scenari diversi.

1.1 Dispositivi

Per effettuare delle prove sperimentali, sono stati utilizzati diversi dispositivi. In particolare tre per simulare il client (o nodo mobile) ed altri per creare traffico nella rete, in modo da impegnare il canale per avere condizioni simili ad un tipico scenario di utilizzo. Per ogni dispositivo è interessante mostrare per cosa è stato utilizzato. Nel caso di un nodo della rete, mostreremo anche il sistema operativo, il kernel, il processore e scheda di rete.

ZyXEL NBG4615 v2 Access point utilizzato durante i test. È stata impostata la modalità Wi-Fi 802.11b/g/n.

LB-LINK BL-WN151 Adattatore Wireless USB. Velocità fino a 150Mb/s, supporta 802.11b/g/n. È stata utilizzata principalmente sui Raspberry, visto che non dispongono di wireless integrato.

NETGEAR WG111 Adattatore Wireless USB. Velocità fino a 54Mb/s, supporta 802.11b/g. Questo adattatore è più lento ed è stato usato sul computer HP quando il wireless integrato dava dei problemi.

HP Pavilion dv6 Entertainment PC Questo notebook è stato utilizzato come client. Abbiamo montato un kernel versione 4.0.1, modificato tramite la procedura illustrata precedentemente. Le specifiche tecniche sono:

- Kernel: Linux versione 4.0.1 modificata.
- Processore: Intel Core i5 CPU M 430.
- Sistema operativo: Ubuntu 14.04 LTS 64-bit.
- Scheda di rete: Broadcom BCM43225 802.11b/g/n.

DELL Latitude E6400 Anche questo notebook è stato utilizzato come client, e vi è quindi montato un kernel versione 4.0.1 modificato. Le specifiche tecniche sono:

- Kernel: Linux versione 4.0.1 modificata.
- Processore: Intel Core 2 Duo.
- Sistema operativo: Ubuntu 14.04 LTS 32-bit.
- Scheda di rete: Intel Corporation WiFi Link 5100 802.11a/g/n.

Raspberry Pi Model 2 Abbiamo utilizzato questo raspberry per creare traffico sulla rete. Le specifiche tecniche sono:

- Kernel: Linux versione 3.18.0-20-rpi2.
- Processore: 900MHz quad-core ARM Cortex-A7.
- Sistema operativo: Raspbian.

UDOO Quad Abbiamo utilizzato questo raspberry per creare traffico sulla rete. Le specifiche tecniche sono:

- Kernel: Kernel Linux 3.0.35.
- Processore: Freescale i.MX 6 ARM Cortex-A9 CPU Dual/Quad core 1GHz.
- Sistema operativo: UDOObuntu.

Raspberry Pi Model B Abbiamo utilizzato questo raspberry per creare traffico sulla rete. Le specifiche tecniche sono:

- Kernel: Kernel Linux 3.18.7+
- Processore: 700 MHz single-core ARM1176JZF-S
- Sistema operativo: Raspbian Wheezy

1.2 Parametri di valutazione

Per poter analizzare i test in modo ottimale e per avere dei dati su cui lavorare abbiamo deciso di controllare alcuni parametri.

I parametri che ci interessano maggiormente sono:

- **Id:** l'Id del pacchetto inviato.

- **ACK:** se è stato ricevuto un ACK o un NACK da parte dell'AP.
- **Data:** è dato dalla data e dall'orario di invio del pacchetto.
- **Tempo:** è il tempo in millisecondi tra l'invio del pacchetto e la ricezione della notifica da parte dell'access point.
- **Retry count:** è il numero di tentativi di invio di un determinato pacchetto
- **Versione IP:** IPv4 o IPv6
- **Configurazione:** è la configurazione dei dispositivi utilizzati durante l'esperimento.
- **Wait:** indica se la recv è bloccante.

Abbiamo scelto questi parametri perché ci permettono di poter giudicare in maniera chiara l'andamento dei pacchetti e la situazione della rete. In particolare sono molto significativi il tempo, l'ACK ed il retry count. Grazie a questi dati si può analizzare in modo dettagliato la situazione di ogni singolo pacchetto. L'applicazione può leggere l'ACK e successivamente decidere di rimandare il pacchetto in base ai millisecondi passati prima di ricevere la notifica. Il numero di retry count risulta rilevante per confrontare differenti situazioni di traffico, oppure per notare cosa succede in caso di trasmissione in movimento. Gli altri parametri che abbiamo deciso di utilizzare hanno un valore più trascurabile per un singolo pacchetto, ma possono diventare eloquenti per analizzare i dati a posteriori. In particolare si potrebbe notare in base all'orario se c'è un evidente rallentamento della trasmissione. Ad esempio si potrebbe notare come in una zona industriale la QoS migliori durante la sera/notte. Un altro dato che può essere utilizzato per esaminare i dati raccolti è la versione IP, si può controllare se c'è un differenza notevole tra IPv6 e IPv4 a parità di condizioni. Le configurazioni, invece, riguardano i dispositivi utilizzati durante un test e lo scenario applicativo. Andremo a mostrare quali configurazioni sono state provate in modo più dettagliato

successivamente. Per quanto riguarda la wait abbiamo deciso di fare sia una recv bloccante che una non bloccante. Abbiamo fatto dei test con entrambe e abbiamo analizzato le differenze, che andremo a descrivere più avanti.

Si potrebbero utilizzare anche altre informazioni (ad esempio la bitrate) per analizzare meglio i risultati, che saranno approfondite negli sviluppi futuri.

1.3 Configurazioni

Per ottenere dei risultati che potessero rispecchiare un reale utilizzo da parte di un nodo mobile abbiamo creato diverse configurazioni di dispositivi. In particolare abbiamo deciso di tenere conto di alcuni possibili scenari di utilizzo, che sono:

- Dispositivo client in movimento oppure fermo.
- Utilizzo indoor o outdoor.
- Trasmissione in linea diretta oppure trasmissione con un ostacolo tra nodo mobile ed AP.
- Assenza di traffico sulla rete in contrapposizione ad uno o più host wireless a creare traffico.
- Nel caso di presenza di nodi sulla rete, abbiamo utilizzato anche la distanza e la velocità come parametri.
- Trasmissioni ad un host della stessa sottorete oppure ad una rete esterna.

Dati questi possibili utilizzi, abbiamo creato alcune configurazioni per valutare la qualità del collegamento. Il nostro interesse si è focalizzato sulla costruzione di un prodotto cartesiano tra tutte le opzioni. Abbiamo quindi provato col nodo mobile fermo, da solo nella rete ed in linea diretta con l'AP.

In questa configurazione abbiamo anche valutato le possibili differenze in velocità tra recv bloccante e non bloccante. La scelta tra queste due opzioni è stata implementata nell'applicazione di prova che abbiamo creato e che abbiamo descritto precedentemente.

In contrapposizione a questa prima configurazione, abbiamo testato un nodo mobile solitario nella rete, fermo ma con un ostacolo tra lui e l'AP. Come ostacolo è stato scelto un muro, di larghezza di circa 15 centimetri.

Per completare la raccolta dati da analizzare abbiamo fatto altre due configurazioni, andando a modificare quelle precedenti usando il nodo mobile in movimento.

Queste prime configurazioni ci permettono già di raccogliere importanti dati, che però non possono riflettere un reale utilizzo di una applicazione. Questo perché difficilmente la trasmissione VoIP avverrà con il dispositivo da solo sulla rete, ma la rete potrà essere più o meno congestionata in base al luogo o all'orario.

Per rispondere a questa esigenza abbiamo deciso di fare delle prove con uno o più host attivi sulla rete, in modo che andassero a causare traffico per rallentare il nodo mobile.

Abbiamo quindi creato altre configurazioni, andando in ognuna a modificare il numero di host ed altri dati. Per quanto riguarda gli host li abbiamo lasciati sempre fermi, ma abbiamo modificato la distanza in diverse prove. In questo modo gli host trasmetteranno più o meno velocemente e vogliamo andare a verificare al client dia più fastidio avere host lenti o veloci. Per controllare le velocità è stata creata una applicazione in grado di stabilire la bitrate. Questa applicazione sarà descritta successivamente.

Analizziamo ora i settaggi che sono stati implementati. In particolare avremo il nodo mobile fermo ed un host attivo, entrambi in linea diretta con l'access point. Da questa configurazione base abbiamo allontanato l'host, fino a variare la velocità in modo significativo (anche di un fattore $1/20$).

Queste ultime configurazione sono state ampliate anche con l'uso della recv prima bloccante, e poi non bloccante. Invece per quanto riguarda tutte le

configurazioni a seguire abbiamo deciso di testare l'applicazione solo con la modalità della recv non bloccante. Questo è stato fatto perché i primi dati empirici erano stati raccolti in quel modo e non si voleva quindi andare ad alterare il risultato. Avendo comunque analizzato separatamente il comportamento bloccante e non, possiamo dare una congettura di quello che può succedere in caso di comportamento non bloccante.

1.4 Dettagli implementativi

Qui spiegheremo come misuriamo la velocità degli hosts (applicazione), dove abbiamo salvato i dati (json) e come li manipoliamo (script python)

1.5 Risultati

Andiamo ora ad analizzare i risultati ottenuti.

Tabella 1.1: Prima tabella

	Da solo	Un host	Due hosts	Tre hosts
prima	1	1	1	456
seconda	1	1	1	1
terza	1	1	1	54
quarta	1	1	1	54
Media	1	1	1	54