



1506
UNIVERSITÀ
DEGLI STUDI
DI URBINO
CARLO BO

UNIVERSITÀ DEGLI STUDI DI URBINO CARLO BO

Dipartimento di Scienze Pure e Applicate
Corso di Laurea in Informatica Applicata

Tesi di Laurea

COS'È BLAZOR?

Relatore:
Chiar.mo Prof. Emanuele Lattanzi

Candidato:
Francesco Belacca

Anno Accademico 2018-2019

A tutti quelli che mi hanno detto almeno una volta di non mollare.

Indice

1	Introduzione	1
1.1	Contesto	1
1.2	Organizzazione	3
2	Wireless LAN	5
2.1	Stato Attuale delle Wireless Network	5
2.2	Dynamic Power Management	5
A	Prova	8
A.1	Pippo	8
A.2	Ciu Ciu	8
	Bibliografia	9
	Ringraziamenti	10

Elenco delle figure

2.1	Consumo di energia senza power management	6
2.2	Consumo di energia con power management	6
2.3	Visione beacon e Ps-Poll	6

Elenco delle tabelle

1.1 Numeri riepilogativi	4
------------------------------------	---

Capitolo 1

Introduzione

1.1 Contesto

A seguito della crescita esponenziale del web in questo secolo e dell'abituarsi di coloro che ne usufruiscono ad un livello grafico sempre migliore e ad una esperienza mano a mano più interattiva e vicina all'utente medio, i siti web e le tecnologie utilizzate si sono adattati per permettere uno sviluppo sempre più rapido di codice più facilmente testabile e mantenibile.

Di conseguenza nel frontend si sono susseguiti una serie di framework a partire da JQuery nel 2006, che per primo si è occupato di risolvere il problema della compatibilità tra browsers, permettendo ai developers di scrivere una volta, e poter eseguire su tutti i browsers. AngularJS nel 2010 è stato il primo MVC framework ad offrire in un unico pacchetto, two-way data binding, dependency injection, routing facilitato e altri strumenti utili per rendere più standard lo sviluppo nel frontend. Dopo la riscrittura di questo framework nel 2013 che è diventato Angular 2 (e recentemente Angular) senza mantenere retrocompatibilità e senza offrire un modo preciso per migrare alla nuova versione agli utilizzatori di AngularJS, React, un nuovo framework più leggero e modulare sviluppato dagli sviluppatori di Facebook, ha preso il posto di Angular come framework più utilizzato.

Vue infine è il terzo dei principali framework che ha provato a prendere piede proponendo una versione intermedia tra il fortemente opinionato Angular e il flessibile React.

Attorno a questi, ciascuno con la propria semantica, organizzazione logica dei folder, CLI dedicato, ad un developer frontend viene solitamente richiesto di conoscere HTML, CSS e vanilla Javascript su cui si basano poi i vari framework.

Oltre a Javascript, se si vuole scrivere degli unit test facilmente mantenibili, bisogna conoscere TypeScript (specialmente se si utilizza Angular, che rende il suo utilizzo obbligatorio) e dei framework che facilitino i test (Enzyme, Karma + Jasmine, ...).

I continui cambiamenti nei framework utilizzati, la varietà degli strumenti, che spesso realizzano in modo diverso tutti la stessa cosa, rendono molto ampia la curva di apprendimento e il tempo necessario ad un neo-developer per essere pienamente operativo.

Proprio per quanto riguarda lo scambio di informazioni col mondo esterno è necessario munire questi dispositivi di interfacce per il collegamento con l'esterno. L'utilizzo di queste interfacce influisce in modo negativo sull'autonomia del dispositivo consumando molta energia, ciò implica che bisogna porre dei limiti sul numero e sull'utilizzo delle interfacce per non costringere l'installazione di batterie troppo ingombranti e più pesanti [2].

Un esempio molto pratico e concreto si può fare su un dispositivo entrato oramai nel modo di vita di tutti noi, il cellulare. Lo sviluppo di questo dispositivo nato inizialmente solo per effettuare telefonate è vertiginoso, ora permette di inviare messaggi di testo, di effettuare fotografie, di personalizzare le suonerie, di scambiare e-mail, di navigare in internet, di giocare e tant'altro; tutto ciò però consuma energia: fotocamera nel caso delle fotografie o display grandi e spaziosi per i giochi; ciò non tanto a discapito dell'ingombro grazie alla miniaturizzazione della componentistica elettronica quanto all'eccessivo consumo energetico colpevole di un'autonomia delle batterie minore. Proprio per questi ultimi due elementi è invece richiesto l'opposto, cioè un ingombro sempre minore e un'autonomia maggiore; è importante trovare un compromesso, in quanto per aumentare l'autonomia sarebbe sufficiente installare batterie di dimensioni superiori che però implica un aumento di peso del dispositivo non trascurabile.

Essendo dispositivi destinati ad un utilizzo mobile non sono accettabili i tradizionali collegamenti tramite cavo, le nuove tecnologie permettono comunicazioni senza cavo come infrarosso o onde radio; nello specifico Irda per la prima tipologia, Bluetooth e Wireless LAN per la seconda. A fronte di queste nuove tecnologie sono emerse nuove problematiche quali limitata copertura del servizio, tasso d'errore nella comunicazione maggiore che comporta la ritrasmissione delle informazioni perse, velocità di connessione minore e consumi di energia maggiori.

Proprio in questo quadro si colloca tale lavoro, il quale consiste in un nuovo modo di utilizzare una tecnologia esistente e consolidata quale la tecnologia wireless LAN, nella sua versione più diffusa e conosciuta, lo standard 802.11b; tale nuovo modo consente di minimizzare il consumo di energia per aumentare la durata delle batteria di cui ogni dispositivo portatile è fornito.

Generalmente ogni stazione wireless comunica con un server che può essere collegato ad internet, o disporre egli stesso delle informazioni richieste dal dispositivo portatile. Il nostro intento è la riorganizzazione (*reshaping*) del traffico, costituito dalle informazioni dirette verso il dispositivo portatile direttamente sul server, questo ci permette un risparmio energetico delle stazioni wireless senza modificare l'architettura attualmente esistente.

Sostenere una comunicazione costa molto in termini di energia consumata,

per tale ragione è necessario evitare comunicazioni inutili come l'ascolto di traffico destinato ad altri o l'attesa continua di informazioni (*pacchetti*) che per qualche ragione non vi sono attualmente. A seguito degli elementi sopra elencati ci troviamo di fronte ad un problema di *Dynamic Power Management*, cioè di variare dinamicamente il consumo di energia della stazione wireless a seconda che debba o meno scambiare informazioni. Principalmente i metodi di power management cercano di mantenere la scheda wireless il più possibile in uno stato di basso consumo, perciò è importante il modo in cui le viene passato il traffico. Se un'applicazione invia pacchetti sporadicamente, tiene attiva per molto tempo la scheda wireless inutilmente; sarà allora necessario riorganizzare il traffico ad un livello sottostante, raggruppandolo ed inviandolo solo quando è veramente corposo, da rendere conveniente il risveglio della scheda.

A fronte delle considerazioni soprastanti, abbiamo deciso di spostare il problema dalle stazioni wireless al server principalmente per due motivi: innanzi tutto il server è collegato all'alimentazione di rete per cui non ha i problemi di autonomia delle batterie, e poi perché ha la visione del traffico globale diretto verso tutte le stazioni wireless a lui collegate. Proprio tramite la visione globale del traffico è possibile un reshaping in modo che ogni stazione riceva solamente le informazioni a lei destinate o dirette a tutti (*broadcast*), cosa che allo stato attuale non è; infatti ogni stazione è sensibile anche alle informazioni dirette ad altri provocando un inutile consumo di energia. Il reshaping consiste nell'invio di blocchi di informazioni dirette verso una sola stazione, mettendola in stand-by quando non ve ne sono; attualmente esiste una sorta di modo di lavoro in stand-by analizzata nel dettaglio in seguito, ma al risveglio una stazione può ricevere comunque le informazioni altrui; il nostro intento è disaccoppiare il più possibile la visione delle informazioni altrui sino a farle scomparire.

1.2 Organizzazione

Prima di analizzare nel dettaglio come viene effettuato il reshaping è necessario vedere le caratteristiche principali del protocollo 802.11; proprio a tal fine il capitolo 2 è stato concepito per introdurre le problematiche relative allo scambio di informazioni in una rete senza fili e alla loro risoluzione. Inoltre sempre nel capitolo 2 viene mostrato lo stato attuale del protocollo da noi utilizzato, le implementazioni riguardanti il power management e la monitoraggio del consumo di potenza.

Nei successivi capitoli ... come si vede in Tabella 1.1 ...

1	2	3
1	2	3
1	2	3
1	2	3

Tabella 1.1: Numeri riepilogativi

Capitolo 2

Wireless LAN

2.1 Stato Attuale delle Wireless Network

Oggigiorno non esistono più dispositivi isolati che non comunicano con altri, tutti i PC prevedono la connessione ad internet, o in generale comunicano con altri dispositivi elettronici grazie alle reti di comunicazione. Inizialmente queste reti erano basate sui cavi (*wired network*), poi si sono rivelate più pratiche le reti senza filo (*wireless network*); da ciò si è dovuto affrontare il problema di permettere l'utilizzo delle tecnologie attualmente utilizzate nelle reti wired anche in reti wireless, che per loro natura sono intrinsecamente meno sicure a causa del mezzo trasmissivo. Le informazioni non vengono più convertite sotto forma di impulso elettronico e spedite su un filo metallico, ma essendo l'etere il mezzo di propagazione sono convertite in onde radio; queste ultime hanno una portata limitata non sempre ben determinabile, sono soggette a disturbi che degradano le informazioni inviate fino, a volte, a farle perdere; ed infine cosa a noi maggiormente sgradita l'energia consumata e direttamente connessa alla potenza trasmessa, ciò significa che aumentare la portata di trasmissione e la qualità del segnale comporta un consumo maggiore dell'energia consumata.

2.2 Dynamic Power Management

Essendo la parte a radio frequenza colpevole del maggior consumo di energia, in particolare gli amplificatori utilizzati immediatamente prima dell'invio del segnale o dopo una ricezione, le stazioni 802.11 possono allungare la vita delle batterie spegnendo i transceiver radio e mettendo la scheda wireless in sleep periodicamente.

Nella figura sottostante (Figura 2.1), si può osservare una finestra di 20 secondi del consumo di energia di una scheda wireless senza il power management abilitato:

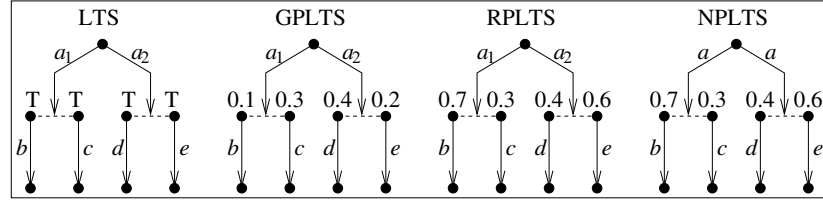


Figura 2.1: Consumo di energia senza power management

Attivando il power management si può notare un sensibile risparmio energetico, a fronte di un diverso consumo energetico dovuto ad un diverso modo di operare:

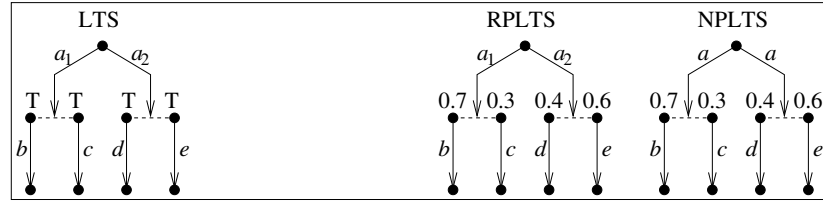


Figura 2.2: Consumo di energia con power management

In questa modalità la stazione resta per la maggior parte del tempo nello stato di sleep, risvegliandosi periodicamente (nella figura ogni 100 ms) per controllare se nel frattempo c'è stato traffico per essa. Durante il periodo di sleep, sarà demandato all'Access Point il compito di bufferizzare i frame destinati verso la scheda momentaneamente in sleep; al risveglio i frame bufferizzati saranno annunciati da un *Beacon frame*, la ricezione da parte della stazione wireless dei frame bufferizzati inizierà subito dopo con la loro richiesta tramite il *PS-Poll frame* (Figura 2.3).

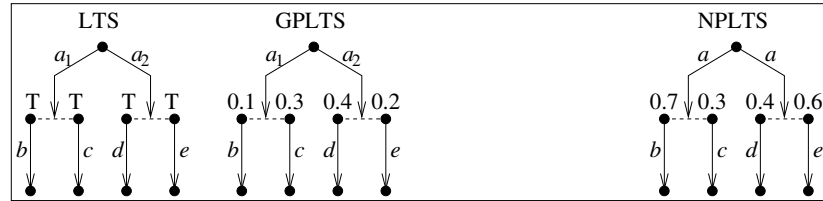


Figura 2.3: Visione beacon e Ps-Poll

Quando viene attivato il power management l'Access Point e la stazione wireless devono sincronizzarsi per risvegliarsi ad intervalli prestabiliti, che nel caso della figura soprastante è 100 ms; all'inizio di ogni risveglio l'Access Point invierà il beacon contenente la *traffic indication map - TIM* che dice alla stazione se vi sono pacchetti bufferizzati destinati ad essa; in tal cosa la stazione

risponderà con un PS-Pool contenente il suo Association ID. Grazie all'Association ID l'Access Point sarà in grado di selezionare i frame destinati alla stazione e provvederà ad inviarli.

Appendice A

Prova

A.1 Pippo

A seguito della crescente diffusione ...

A.2 Ciu Ciu

Prima di analizzare nel dettaglio ...

Bibliografia

- [1] V. J. Mathews e G. L. Sicuranza, *Polynomial Signal Processing*, New York: Wiley, 2000, pp.100-120.
- [2] A. Carini, E. Mumolo e G. L. Sicuranza, "V-Vector Algebra and Volterra Filters", in *Advances in Imaging & Electron Physics*, Vol. 124, P. W. Hawkes, Ed., San Diego: Academic Press, 2003, pp. 1-61.
- [3] A. Carini, "Efficient NLMS and RLS Algorithms for Perfect and Imperfect Periodic Sequences", *IEEE Transactions on Signal Processing*, vol. 58, No. 4, pp. 2048-2059, Aprile 2010.
- [4] A. Carini, V. J. Mathews e G. L. Sicuranza, "Efficient NLMS and RLS Algorithms for a Class of Nonlinear Filters Using Periodic Input Sequences", *Proceedings of 36th International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011*, Prague, Czech Republic, Maggio 22-27, 2011, pp. 4280-4283.
- [5] A. Carini, "Filter for the Equalization and/or Linearization of Nonlinear Systems", Brevetto Europeo EP 0939487A2, 25 Febbraio 1999.
- [6] *IEEE Criteria for Class IE Electric Systems*, IEEE Standard 308, 1969.
- [7] J. Jones, *Networks* (2nd ed.) [Online] (10 Maggio 1991). [Ultimo accesso: 11 Maggio 2001]. Disponibile: <http://www.atm.com>
- [8] *Prova Finale* [Online] (21 Febbraio 2013). [Ultimo Accesso: 22 Maggio 2013]. Disponibile: http://www.sti.uniurb.it/info_appl_l31/piano_studi/prov_fina.html
- [9] R. J. Vidmar, *On the Use of Atmospheric Plasmas as Electromagnetic Reflectors* [Online]. (1994). [Ultimo accesso: 11 Maggio 2001] Disponibile FTP: [atmnext.usc.edu](ftp://atmnext.usc.edu) Directory: [pub/etext/1994](ftp://pub/etext/1994) File: [atmosplasma.txt](ftp://atmosplasma.txt)

Ringraziamenti

Vorrei ringraziare il professor Lattanzi per avermi aiutato a scrivere questa tesi, i miei genitori per avermi permesso di procedere facendo di testa mia su un percorso a loro estraneo, e la mia ragazza Ivana per aver creduto in me e nelle mie capacità anche quando non l'ho fatto io.