

Capitolo 2

Power Saving nelle reti WLAN

In questo capitolo verranno introdotti alcuni algoritmi di Power Saving utilizzati in reti W-LAN: descriveremo il meccanismo standard di Power Saving, l'Automatic Power Save Delivery (APSD) introdotto dal DRAFT 6.0 [20] e il power saving affiancato agli scheduler FBDS e PI-FBDS, rispettivamente, PS FBDS (FBDS con Power Saving) e PS PI-FBDS (PI-FBDS con Power Saving) proposti in questo lavoro.

2.1 *Power Save Standard*

Prima di descrivere il funzionamento del *Power Save Standard*, è opportuno distinguere quelli che sono gli stati in cui una STA può trovarsi.

Posiamo distinguere due diversi stati:

- *Awake State*
- *Doze State*

Nel primo caso la STA è attiva, quindi può comunicare con le altre stazioni, nel secondo invece essa è spenta, per cui non può né ricevere né inviare frame; il passaggio da uno stato all'altro dipende dall' algoritmo di Power Management utilizzato.

Definiti questi stati, lo Standard 802.11 [2] stabilisce che una STA può operare in due distinte modalità:

- *Active Mode* (AM), in cui è sempre in *Awake State*;
- *Power Save Mode* (PS Mode), in cui passa da uno stato all' altro in funzione di specifici eventi.

Il meccanismo di base di power saving in uso nelle WLAN è di tipo strutturato, esso sostanzialmente è un meccanismo di memorizzazione e successiva consegna dei frame da parte dell' AP alle stazioni che operano in Power Save Mode (*PS STA*): è quindi utilizzato solo per il traffico in *downlink* (cioè diretto dall' AP alle *PS STA*). Una stazione che intende utilizzare questo algoritmo deve comunicarlo all' AP settando a "1" il bit *Power Management* [cfr. par. 1.5.1] nel un frame immediatamente precedente all'entrarla in *PS Mode*; una volta completata la sequenza di scambio (*DATA* e relativo *ACK*), la stazione passerà in *Doze State*. L' AP dovrà memorizzare tutto il traffico diretto verso quella stazione per poi consegnarlo quando essa ritornerà in *Awake State*.

Le *STA* operanti in modalità *PS* per le quali l' AP ha del traffico pendente sono identificate attraverso l' elemento *TIM* (*Traffic Indication Map*), presente in ogni beacon, corrispondente ad un opportuno bit per ogni nodo associato alla *BSS*: se la stazione opera in *PS Mode* e il suo bit di *TIM* è settato a 1, vorrà dire che l' AP avrà traffico pendente per essa. E' evidente che ogni *PS STA* dovrà accendersi poco prima dell' arrivo del beacon (ovvero poco prima di ogni *TBTT*) per poter ricevere tale frame, leggere il *TIM* e stabilire se c'è traffico per essa oppure no.

Successivamente, se la *PS STA* opera in modalità *PCF*, una volta appurato che l' AP ha traffico memorizzato per essa, rimarrà in *Awake State* durante il *CFP* finché non arriverà il *CFPoll*, dopo di che riceverà un frame dati e si spegnerà, dopo aver inviato l' *ACK*, per riaccendersi al prossimo beacon. Qualora la stazione non dovesse ricevere il *CFPoll* oppure dovesse ricevere un pacchetto con il campo *More Data* settato a 1 (il che significherebbe che l' AP avrebbe ancora dati pendenti per essa), potrà decidere se rimanere in *Awake State* anche durante il *CP* (e in tal caso seguirà le regole di accesso al canale previste dal *DCF*), o se passare in *Doze State* e riaccendersi al prossimo beacon: questa procedura si ripeterà finché l' AP indicherà la presenza di traffico pendente per la *PS STA*.

Diversamente durante il *CP*, l' AP non potrà inoltrare il traffico memorizzato per una *PS STA* finché non sarà sicuro che questa si trovi in *Awake State*, di conseguenza la stazione,

una volta vinta la contesa, comunicherà all' AP di essersi accesa mediante un frame *PS Poll* e rimarrà accesa finché non riceverà risposta al polling o finché l' AP indicherà che non c'è più traffico destinato ad essa. Qualora la PS STA dovesse ricevere un frame con il bit *More Data* a 1, potrà decidere se rimanere accesa e partecipare ad una nuova contesa per continuare a ricevere dati (l' AP può inviare alla PS STA un unico MSDU a seguito di un *PS Poll*) o tornare in *Doze State*.

Un discorso a parte merita il traffico di tipo *Broadcast*: viene utilizzato l' elemento *DTIM* (*Distributed TIM*) per indicare la presenza di traffico broadcast memorizzato nell' AP; i *DTIM* vengono inviati tramite particolari beacon ad intervalli regolari denominati *DTIMPeriod* (generalmente multipli del *TBTT*); subito dopo il beacon in cui è presente il *DTIM*, l' AP trasmetterà tutto il traffico *Broadcast* seguito da normale traffico *Unicast*.

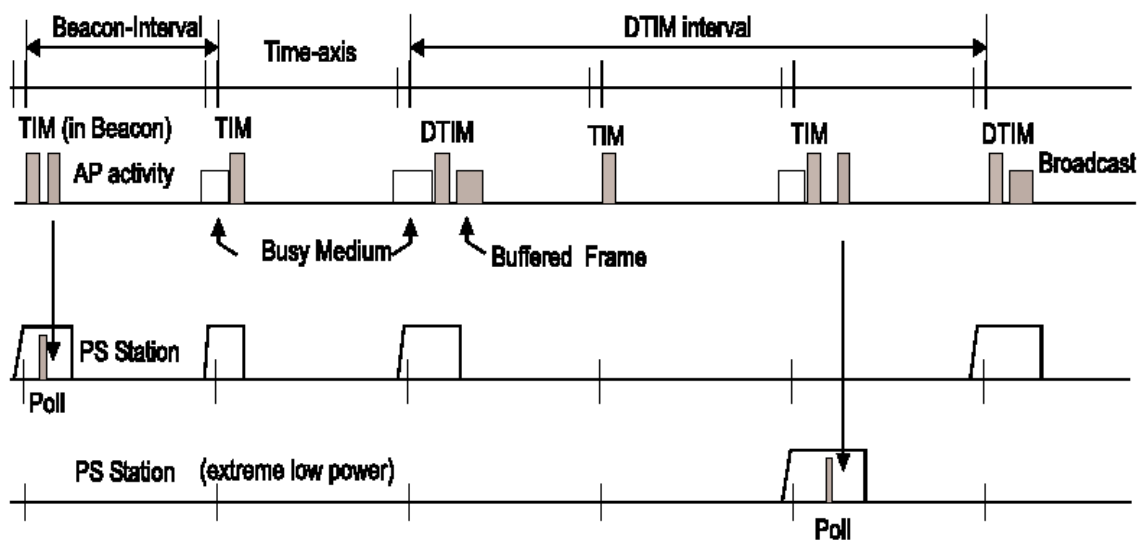


Figura 2.1: Alternanza tra TIM e DTIM

2.2 *Automatic Power Save Delivery*

L' *Automatic Power Save Delivery (APSD)* viene introdotto dal DRAFT 6.0 dell'IEEE 802.11e [20] nell'ambito delle QoS WLAN ed è sostanzialmente una evoluzione dello Standard Power Save.

L' APSD prevede la consegna di QoS frame in downlink mediante due differenti meccanismi legati al tipo di SP (*Service Period*) utilizzato [12]:

- *Unscheduled Service Period*, che ha inizio quando una QSTA comincia a trasmettere e termina quando il QAP le avrà inviato tutti i dati destinati ad essa;
- *Scheduled Service Period*, che è simile all' *Unscheduled Service Period* con la differenza che ha inizio in istanti predefiniti la cui distanza si definisce *Service Interval* ovvero il periodo di ripetizione di uno *Scheduled Service Period*.

Un *Service Period* è un un intervallo continuo di tempo all' interno del quale uno o più downlink frame e/o uno o più polled TXOP sono garantiti alla QSTA.

Un *APSD SP* è un periodo nel quale la QSTA è sicuramente in *Awake State*, per cui può ricevere pacchetti dal QAP.

Nel caso di *Scheduled SP*, la trasmissione di frame (sia in uplink che in downlink) avverrà in istanti ben precisi negoziati con il QAP: la QSTA si accenderà rispettando la tempistica concordata con il QAP per ricevere il traffico in downlink.

Nel caso di *Unscheduled SP* invece sarà la stazione a inviare un "trigger" (può essere o un *QoS Data* o un *QoS Null*) al QAP, che trasmetterà tutto il traffico memorizzato destinato ad essa.

Un nodo che intende utilizzare APSD deve comunicarlo al QAP nel momento in cui entra a far parte della QBSS settando il bit opportuno nel TSPEC, deve comunicare anche il tipo di SP che intende utilizzare e, nel caso di *Scheduled SP*, quando si accenderà. Se il QAP non dovesse essere in grado di soddisfare quest' ultima richiesta, risponderà alla QSTA con uno *Schedule Element* (figura 2.2), uno speciale *management frame*, in cui indicherà in

Octets: 1	1	2	4	4	2	2
Element ID (15)	Length (14)	Schedule Info	Service Start Time	Service Interval	Maximum Service Duration	Specification Interval

Figura 2.2: Formato dello Schedule Element

quale istante essa potrà accendersi; da questo momento il QAP memorizzerà tutto il traffico diretto alla stazione per consegnarglielo quando essa sarà accesa (all' inizio dell' APSD SP): infatti la QSTA passerà automaticamente in *Awake State* all'inizio di ogni *Scheduled SP*, istante specificato nel campo *Service Start Time* dello *Schedule Element*. Il QAP cercherà di inoltrare tutti i dati diretti al nodo in modo *Aggregate* (cioè i frame appartenenti a tutte le TC) e comunicherà la fine del SP impostando nell' ultimo downlink frame il campo *End of Service Period* (EOSP), corrispondente al quarto bit del QoS Control Field; a questo punto la QSTA ritornerà in *Doze State*; se una QSTA dovesse utilizzare APSD per una TC e non per le altre, essa passerà in *Awake State* solo nel SP relativo a quella TC. Dopo aver ricevuto il traffico in downlink, la stazione può inviare il traffico in uplink. La ricezione di pacchetti multicast/broadcast segue le stesse regole già viste per lo Standard Power Save.

Rispetto allo Standard Power Save, l' APSD introduce diversi miglioramenti sia dal punto di vista del consumo energetico che dal punto di vista dell' efficienza del protocollo nella gestione del traffico: infatti, la PS STA non rimane più accesa in attesa del poll (il che, poteva anche significare restare in *Awake State* per tutto il CFP), ma si accende solo durante il suo SP, con un conseguente maggiore risparmio energetico; inoltre il traffico generato dai nodi sarà minore, dato che non c'è più bisogno di ricorrere ai PS Poll per richiedere i frame memorizzati dal QAP.

2.3 *PS FBDS e PS PI-FBDS (FBDS e PI-FBDS con Power Saving)*

Gli algoritmi *PS FBDS (FBDS con Power Saving)* [21] e *PS PI-FBDS (PI-FBDS con Power Saving)*, ideati presso il DEE del Politecnico di Bari per essere implementati in reti di tipo 802.11e, sono nati con l' intento di applicare un meccanismo di power saving agli algoritmi di scheduling FBDS e PI-FBDS, in modo da far conciliare nel miglior modo possibile due esigenze contrastanti: il supporto della QoS e il risparmio energetico. Per garantire la QoS è necessario, infatti, che vengano rispettate precise tempistiche nella consegna dei frame, ciò implica che le QSTA debbano essere in grado di riceverli/inviarli in istanti opportuni: è palese come questa esigenza mal si sposi con le tecniche di risparmio energetico fin qui analizzate.

In questo caso viene eliminata la fase di pre-polling introdotta per evitare eventuali *deadlock* delle QSTA a coda nulla [cfr. par. 1.6.4.2]. L' eliminazione del pre-polling ha determinato la necessità di una nuova soluzione al problema del *deadlock*, che a sua volta ha comportato l' introduzione di una modifica nel funzionamento di FBDS: essa consiste nell' assegnare alle PS STA le cui code risultano vuote un TXOP di durata sufficiente a inviare un MSDU di dimensione massima al minimo data rate. Inoltre manca l' invio degli Schedule Element che comunicano ad ogni stazione quando riceverà il poll. Più semplicemente ogni QSTA si accenderà per ricevere il beacon (gli istanti di invio dei beacon sono fissi e noti a tutte le stazioni) e rimarrà in Awake State finché non avrà ricevuto il *poll* e utilizzato il TXOP (o SP) concesso per drenare la propria coda, poi si spegnerà. Se la QSTA non ha dati da trasmettere, invia il solito frame QoS Null (per indicare il proprio livello di coda ed evitare spreco di risorse) e ritorna in Doze State; se la QSTA ha dati da trasmettere, li invia ed attende l' arrivo dell' ACK relativo all' ultimo frame trasmesso prima di passare nuovamente in Doze State; nel caso di traffico Broadcast, ovviamente, la QSTA non attende nessun ACK prima di tornare in Doze State.

Tutte queste considerazioni teoriche esposte fin qui, saranno verificate attraverso una serie di simulazioni, per verificare la funzionalità dei vari algoritmi; essi verranno implementati su un simulatore (di cui si parlerà nel prossimo capitolo) che ci permetta, in maniera virtuale, di ricostruire il comportamento di una rete 802.11e, in modo da poter considerare tutte quelle situazioni che difficilmente sarebbero prevedibili con un approccio solo teorico al problema: bisogna sempre tener conto che un simulatore fornisce un'approssimazione il più vicino possibile alla realtà che risulta tanto più affidabile quanto più il modello del software realizzato risulta preciso; i risultati sono riportati nel capitolo 4.