Capitolo 1

Standard IEEE 802.11

1.1 Il progetto IEEE 802

Lo standard è la chiave di volta per garantire un'ampia diffusione di una tecnologia. Infatti, grazie allo standard, diversi costruttori possono realizzare lo stesso prodotto e venderlo in concorrenza; gli utilizzatori possono scegliere tra i diversi costruttori perché sanno che le interfacce di apparecchi analoghi realizzati da costruttori diversi sono le stesse.

Gli standard delle LAN sono stati definiti da comitati della IEEE (Institute for Electrical and Electronics Engineers), sotto il nome di progetto IEEE 802[1]. Il modello di riferimento a strati di questo progetto, riportato nella figura 1.1, definisce 3 strati:

- *fisico*, il cui compito è l'interfacciamento della stazione con il mezzo trasmissivo con annessi compiti di codifica/decodifica dei bit trasmessi/ricevuti;
- *Medium Access Control,* MAC, che governa le procedure di accesso al mezzo trasmissivo che nelle reti LAN è condiviso da tutte le stazioni connesse alla rete;
- *Logical Link Control, LLC*, il cui compito è la gestione di collegamenti logici di livello 2.

Il progetto IEEE 802 definisce diverse architetture di reti LAN e queste si differenziano per le modalità di implementazione del livello fisico e del livello MAC essendo il livello LLC comune a tutte le reti. Tali standard sono indicati come 802.X e coprono tutti gli aspetti generali di sistema e di implementazione delle diverse architetture di rete locale; essi sono:

[1] Gli standard della serie IEEE 802 sono anche pubblicati come standard ISO (Inrenational Organization for Standardization) e IEC (International Electrotechnical Commission) sotto la serie ISO/IEC 8802 con analoga numerazione.

- 802.1, *higher layer LAN*;
- 802.2, *Logical Link Control*, LLC;
- 802.3, *Ethernet*;
- 802.4, *Token Bus*;
- 802.5, *Token Ring;*
- 802.6, *metropolitan area network;*
- 802.11, *wireless LAN.*

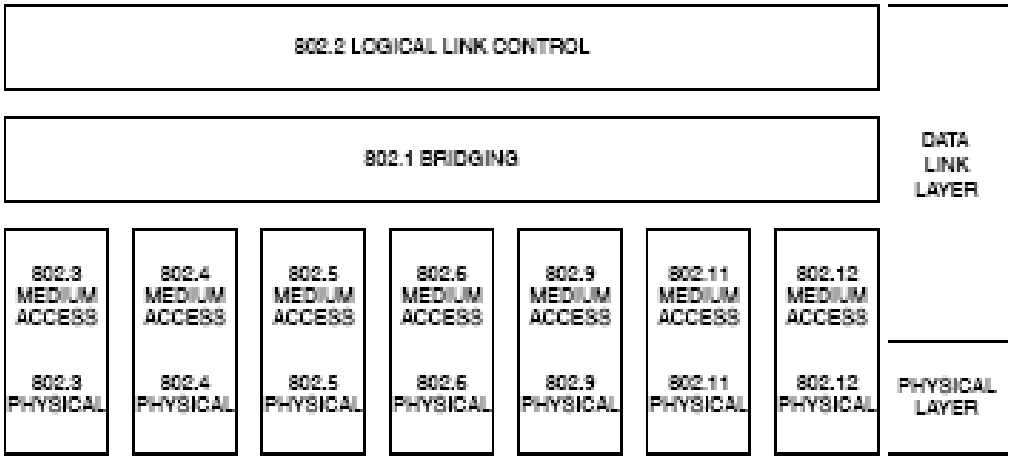


Figura 1.1: Modello di riferimento a strati del progetto IEEE 802

1.2 Introduzione alle WLAN

Uno degli sviluppi più promettenti del computer networking è la grande diffusione di terminali wireless. Questi reti non cablate comprendono già i PC portatili all'interno di una WLAN (*Wireless Local Area Network*), e le PDA (*Personal Digital Assistant*), che si connettono a Internet usando le infrastrutture esistenti ed emergenti di telefonia wireless. Nel giro di pochi anni è probabile che vedremo una più ampia varietà di dispositivi wireless che accedono ad Internet. Questi dispositivi possono comprendere videocamere, automobili, giocattoli, sistemi di sicurezza, elettrodomestici e impianti di distribuzione di energia elettrica, acqua, gas ecc. Infatti, in futuro i dispositivi wireless potranno davvero essere presenti ovunque: sui muri, nelle nostre automobili e sui nostri corpi.

Più in particolare per WLAN, si intende una rete locale di telecomunicazioni fra apparecchiature elettroniche di diversa natura, libere di muoversi senza i cavi di collegamento entro un'area dell'ordine delle centinaia di metri quadri, e che comunicano fra loro grazie alle tecnologie radio. Solitamente, oltre che comunicare fra loro, i nodi della rete wireless sono in grado di comunicare anche con l'esterno, e quindi con altre WLAN o con Internet. Si capisce quindi come le WLAN si possano considerare la naturale estensione senza fili delle comunissime reti LAN. Rispetto a quest'ultime, le WLAN presentano indiscutibili vantaggi legati alla mobilità, flessibilità e facilità d'installazione. Si pensi a più utenti domestici che stanno adottando l'economia tecnologica delle WLAN nelle loro case, per permettere a più membri della comunità domestica di accedere simultaneamente a Internet mentre si spostano nella casa. Per contro la particolare natura del mezzo di comunicazione radio pone dei problemi di attenuazione, legati alla presenza di ostacoli o al verificarsi di riflessioni indesiderate non prevedibili in fase di progetto che modificano il cammino delle onde elettromagnetiche, noto come *fenomeno dei cammini multipli*; di interferenza con segnali provenienti da altre sorgenti elettromagnetiche; di sicurezza delle comunicazioni (in particolare di autenticità della sorgente e di segretezza dei

dati trasmessi), anche se questi problemi di sicurezza sono ovviati con l'uso delle diverse tecniche di crittografia ed autenticazione; di durata finita delle batterie dei dispositivi, anche se per la soluzione di questo problema la ricerca sta facendo passi da gigante, grazie all'implementazione di algoritmi per il risparmio energetico, argomento di interesse di questa tesi. Inoltre il raggio di copertura di una rete locale senza fili è inferiore rispetto ad una LAN cablata: nel primo caso può raggiungere distanze dell' ordine del centinaio di metri, nel secondo invece può arrivare fino a 2.5 km di distanza [1].

Esistono in commercio e in progetto svariati standard che mirano al mercato delle WLAN, solitamente non compatibili fra loro. I principali sono[1]:

- *HiperLAN* (*High Performance Radio Local Area Network*), definito nel 1996 dall'organizzazione ETSI-BRAN (*European Telecommunications Standards Institute Broadband Radio Access Network*), e disponibile nelle due versioni HiperLAN/1 [1] e HiperLAN/2 [7][8]. Entrambe operano nella banda radio dei 5GHz, garantendo un supporto per la QoS per dati, voce, video e immagini, e raggiungendo velocità rispettivamente di 24Mpbs e 54Mbps.
- *SWAP* (*Shared Wireless Access Protocol*) [14], definito nel 1998 dall' HRFWG (*HomeRF Working Group*), operante nella banda radio dei 2.4GHz. Utilizzando la modulazione FHSS, prevede la trasmissione di dati e voce a 1Mbps e a 2Mbps. Sta nascendo una nuova versione che riuscirà a trasmettere a 10Mbps ancora con modulazione FHSS, ma con un' ampiezza di banda maggiore.
- *Bluetooth* [13], definito nel 1998 dal Bluetooth SIG (Special Interest Group), opera nella banda di frequenza dei 2.4GHz, con velocità massima di 1Mbps, usando la modulazione FHSS. A rigore le reti Bluetooth non sono classificate wireless LAN, ma come *Personal Area Network* (*PAN*) e possono essere viste come un sottoinsieme di una WLAN. Usando potenze più basse rispetto agli altri standard, il Bluetooth copre distanze minori (dell' ordine dei metri) ed è adatto a comunicazioni via etere fra dispositivi molto vicini (tipicamente fra computer e periferiche)[3].

- *ZigBee* [13], un protocollo di recentissima definizione molto simile al *Bluetooth*, ma che lavora a velocità ancora più basse (20~250 kbps) ed è particolarmente indicato per la sensoristica.
- *IEEE 802.11* [2], definito nel 1997 dall' IEEE (*Institute of Electrical and Electronics Engineers*), operante sia nella banda dei 2.4GHz che dei 5GHz, e raggiungendo rispettivamente velocità di 11Mbps (802.11b [5]) e 54Mbps (802.11a [6] e 802.11g [9]).

1.3 Standard IEEE 802.11

Lo standard IEEE 802.11 [2][5][6][9] si inserisce nel contesto di una ben più ampia famiglia di standard definiti dall'organizzazione IEEE.
Lo scopo di questo standard è lo sviluppo di un medium access control (MAC) un physical layer (PHY) specifici per la connettività wireless di fissi, portatili e stazioni mobili all'interno di un' area locale.
In particolare, questo standard

- Descrive le funzioni e i servizi richiesti da un' apparecchiatura conforme IEEE 802.11 che opera all'interno di queste reti.
- Definisce le procedure di MAC per supportare la consegna asincrona di MSDU (MAC Service Data Unit).
- Definisce varie tecniche di signalling a livello PHY e funzioni di interfaccia controllate dall' 802.11 MAC.
- Descrive i requisiti e procedure per offrire riserbo di informazioni di utente che sono trasferite sul mezzo senza fili (WM) e l'autenticazione di apparecchiature conformi all' IEEE 802.11.
- Gestione della potenza e dello spettro di frequenze, ecc.

In particolare lo standard 802.11 si riferisce alle WLAN ed è in continua evoluzione. La versione originale prevede l'uso della banda di frequenze dei 2.4GHz, la cosiddetta ISM (*Industrial, Scientific and Medical band*), che è disponibile a libero uso dei privati, quindi non richiede la concessione di licenza da parte degli enti governativi. Essa considera il livello MAC relativamente a tre distinti livelli fisici, che utilizzano tre distinte tecniche di modulazione del segnale: IR (*InfraRed*), FHSS (*Frequency Hopping Spread Spectrum*), DSSS (*Direct Sequence Spread Spectrum*). Con queste modulazioni si riesce a trasmettere a 1Mbps e a 2Mbps. Per far fronte ad una crescente domanda di velocità nelle comunicazioni, sono state introdotte, a partire dal 1999, nuove estensioni dello standard di base 802.11, che ottimizzano il livello PHY lasciando invariato il livello MAC:

- 802.11a[6], che opera nella banda di frequenza dei 5GHz e, usando la modulazione OFDM (*Orthogonal Frequency Division Multiplexing*), raggiunge velocità fino a 54 Mbps.
- 802.11b[5], che lavora nella banda dei 2.4GHz e si avvale della modulazione HR-DSSS (*High Rate – DSSS*), fornendo un bit rate fino a 11Mbps.
- 802.11g[9], l' estensione più recente (giugno 2003), che usa anch' essa la banda a 2.4GHz e la modulazione OFDM, ma riesce a raggiungere velocità di trasmissione fino a 54Mbps.

Standard	**Data Rate [Mbps]**	**Banda [GHz]**	**Modulazione**
802.11	1, 2	2.4	IR, FHSS, DSSS
802.11a	6, 9, 12, 18, 24, 36, 48, 54	5	OFDM
802.11b	1, 2, 5.5, 11	2.4	HR-DSSS
802.11g	6, 12, 24, 36, 48, 54	2.4	OF

Tabella 1.I: Estensioni a livello PHY dello standard IEEE 802.11

Esistono inoltre gruppi di lavoro che hanno introdotto o stanno studiando l'introduzione di altre estensioni allo standard, per perfezionarne i vari aspetti (vedi tabella 1.II). Tra i più importanti vi è l'802.11e, che si propone di migliorare il livello MAC al fine di garantire un supporto alla QoS. Tale gruppo di lavoro non è ancora approdato ad una versione definitiva,e quindi, la ricerca svolta e descritta in questa tesi si riferisce al più recente stato di avanzamento approvato dall'IEEE 802.11e, documento Draft 8.0 [12].

Working Group	**Descrizione**
802.11e	Supporto alla QoS
802.11f	Standardizzazione del protocollo di roaming
802.11h	Miglioramento dei livelli MAC e PHY delle reti 802.11a
802.11i	Incremento della sicurezza
802.11k	Definizione di un Radio Resource Measurement che comunichi con i livelli superiori.

Tabella 1.II: Gruppi di lavoro per le estensioni dello standard IEEE 802.11

1.4 Funzionamento di una rete 802.11

Una rete 802.11 è organizzata come un'insieme di una o piu celle elementari chiamate BSS (*Basic Service Set*) eventualmente comunicanti fra loro [1][2]. Un gruppo di stazioni (STA) situate nella stessa area geografica (BSA, Basic Service Area) e che sono in grado di

comunicare facilmente fra loro in maniera diretta formano una BSS. Una BSS può essere ti tipo *strutturato* o *Ad Hoc:* nel primo caso, all'interno della BSS, vi è un nodo particolare (l'*Access Point,* AP) il cui compito è quello di coordinare, sincronizzare ed identificare la BSS stessa; nel secondo caso non vi è nessun AP e le stazioni comunicano tra loro direttamente, se sono in visibilità radio, o passando attraverso nodi intermedi che hanno il compito di trasmettere il segnale fino a raggiungere il destinatario. Consideriamo reti di tipo strutturato. Ogni stazione della rete, in fase di start-up, si associa ad un ben preciso AP, solitamente il piu vicino disponibile. In funzione della configurazione, può essere necessario l'uso di un protocollo di autenticazione delle stazioni
che chiedono di associarsi e di algoritmi di crittografia per cifrare tutte le successive comunicazioni.

All'interno della BSS solitamente le comunicazioni fra due distinte stazioni avvengono sempre e solo tramite l'AP, che quindi deve essere in visibilità radio con tutte quante le stazioni della BSS. In questa modalità quindi ogni stazione comunica direttamente in upload o in download solo con l'AP. In realtà, nel Draft 8 [12], è stato introdotto un meccanismo di comunicazione diretta fra STA all'interno di una *Infrastrucured BSS* che prende il nome di *Direct Link Pootocol* (DLP) e che utilizza l'AP solo per il setup del collegamento fra le due STA. Più celle BSS indipendenti fra loro possono essere usate per formare una rete di telecomunicazioni wireless a più vasta copertura geografica (*ESS, Extended Service Set*) posizionando in luoghi opportuni gli AP delle diverse celle in modo da ottenere, in ogni punto dell'area d'interesse, una visibilità radio con almeno uno degli AP appartenenti all'ESS. Ovviamente, ogni stazione sarà sempre associata con un solo AP (tipicamente quello a migliore visibilià radio) e comunicherà direttamente solo con esso; saranno poi gli AP della ESS che dovranno comunicare fra loro e quindi permettere flussi di dati che vanno al di fuori della singola BSS, tra nodi wireless appartenenti a BSS distinte o addirittura, se predisposto, con nodi di reti esterne alla ESS stessa, ad esempio Internet. A tal fine, dunque, una delle funzionalità primarie degli AP, oltre quelle di individuare,

coordinare e gestire la propria BSS, è quella di *bridge* verso l'esterno della BSS, per mezzo del DS (*Distribution Service*), che è una rete, wired o wireless, che collega i diversi AP fra

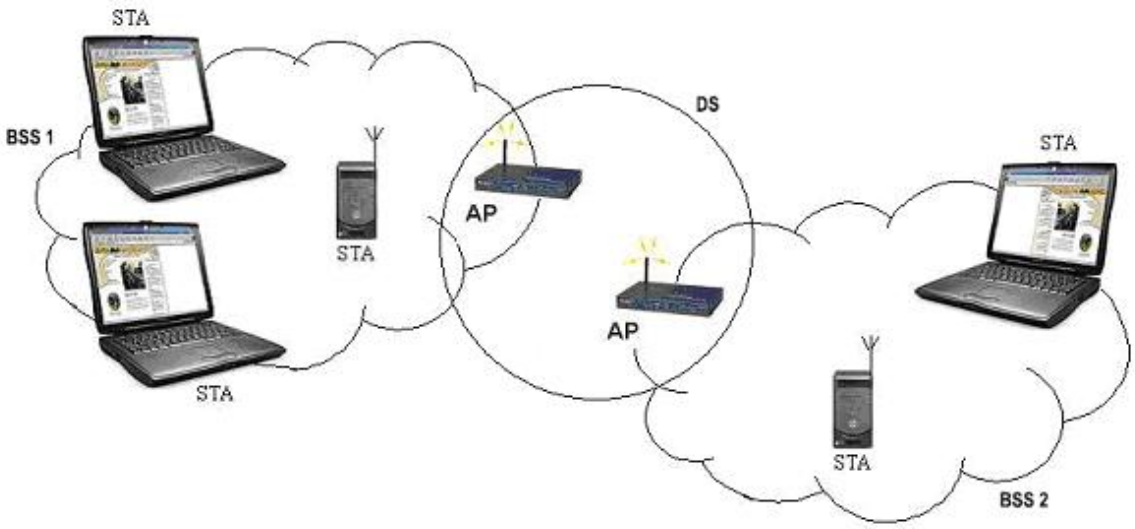


Figura 1.2: Esempio di una ESS formata da due BSS

loro ed eventualmente con l'esterno. Ciascuna stazione wireless è libera di muoversi in tutta l'area di copertura dell'ESS e, durante gli spostamenti da una BSS all'altra, è necessario che usi dei protocolli di roaming che ne permettano la dissociazione dalla vecchia BSS e l'associazione con la nuova BSS: in questo modo essa si troverà sempre nella migliore condizione possibile di visibilità radio messa a disposizione dalla rete. Lo standard 802.11 non specifica come il roaming debba essere realizzato, ma definisce una tecnica di base: la stazione in movimento rileva gli AP disponibili per la connessione e, in relazione al livello del segnale ricevuto, decide a quale associarsi. Attraverso un meccanismo di re-associazione, definito dallo standard, la stazione può interrompere l'associazione al vecchio AP per attivare un' associazione al nuovo AP. Il processo di re-associazione si instaura mediante uno scambio di informazioni tra i due AP in questione, che avviene per mezzo del DS senza appesantire la comunicazione radio.

Mentre per reti di tipo non strutturato o *Ad Hoc* le comunicazioni fra due distinte stazioni possono avvenire direttamente se esse sono in visibilità radio, oppure passando tramite

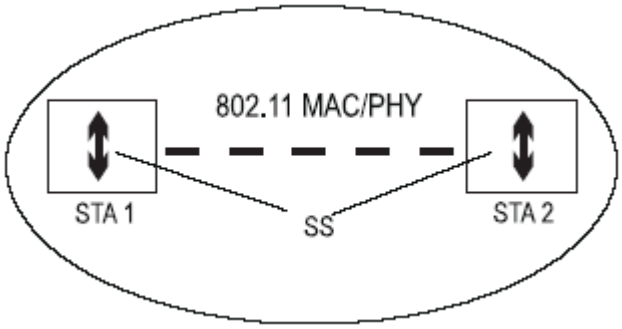


Figura 1.3: Struttura logica di una IBSS

stazioni intermedie che hanno il compito di inoltrare le trasmissioni fino a raggiungere il destinatario senza l'ausilio dell'AP. È evidente che questo tipo di rete necessiti di complicati protocolli di routing che gestiscano l'instradamento dei dati. Le BSS che funzionino in tale modalità non sono di solito collegate con altre BSS o con altre reti esterne in generale, e quindi vengono anche chiamate IBSS (*Independent Basic Service Set*), ne consegue che una IBSS è una rete autoconfigurante "chiusa", senza possibilità di ampliamento e funzionale ad esigenze specifiche: di qui il nome *ad hoc network* attribuito a questo tipo di rete.

1.5 Il livello MAC

Qualunque sia la topologia considerata, le regole con cui ogni nodo accede al canale wireless e scambia i dati col nodo destinatario sono regolamentate dallo standard [2][12] e definiscono il livello MAC (*Medium Access Control*). Dato che in ogni BSS, fissata la banda di frequenza di lavoro e il canale di utilizzo, tutti i nodi (compreso l' AP)

comunicheranno fra loro condividendo la stessa risorsa radio, affinchè le trasmissioni vadano a buon e fine, è necessario che ciascuna di esse avvenga in tempi diversi senza alcuna sovrapposizione temporale: quindi, se è già in corso una trasmissione, nessun altro deve trasmettere. In caso di contemporaneità di due o più trasmissioni, nessuna di esse risulterà distinguibile, per cui occorrerà effettuare la ritrasmissione degli stessi dati: si dice che è avvenuta una *collisione*. Tale evento è facilmente rilevabile da tutti i nodi in ascolto sul canale, i quali percepiranno la presenza dei segnali ma non li riusciranno a decifrare: questa tecnica è nota come CS (*Carrier Sensing, Rilevazione della Portante*). In realtà, la particolare natura del mezzo wireless (elevata attenuazione del segnale), rende impossibile il rilevamento delle collisioni ai nodi che le hanno originate, a causa del fatto che la potenza trasmessa da questi nodi rende trascurabile (e quindi non percepibile) la potenza ricevuta dalle altre trasmissioni contemporanee: è questa una delle differenze fondamentali che distingue la comunicazione wireless da quella wired. Una stazione che trasmette dei dati è quindi in grado di rilevare una collisione solo in modo indiretto e solo a trasmissione finita: non ricevendo il pacchetto di riscontro (*ACK*, acknowledgment) che confermi la buona riuscita della trasmissione, essa comprende che il pacchetto trasmesso non è stato ricevuto. Il riscontro è generalmente presente solo nel caso di trasmissione andata a buon fine ed è inviato dal nodo ricevente. Con l' intento di superare questo problema, nelle comunicazioni wireless si usano politiche di trasmissione che tentino soprattutto di evitare le collisioni e che vengono indicate con la sigla CA (*Collision Avoidance*), mentre nelle comunicazioni wired si tende ad usare meccanismi di rivelazione delle collisioni, indicati con la sigla CD (*Collision Detection*). Abbiamo così il protocollo CSMA/CA (*Carrier Sense Multiple Access/Collision Avoidance*) nelle WLAN e il protocollo CSMA/CD (*Carrier Sense Multiple Access/Collision Detection*) nelle LAN cablate. Quando una stazione desidera trasmettere, il protocollo CSMA s'incarica di testare il canale: se è occupato, rimanda la trasmissione ad un momento successivo, se invece è libero, inizia a trasmettere. Questo tipo

di protocolli sono molto efficienti quando il traffico sulla rete non è molto intenso, in quanto i tempi di attesa sono brevi e le stazioni possono trasmettere con il minimo ritardo.
Altri motivi di assenza del riscontro possono essere interferenze da parte di altri segnali radio ed errori dovuti al mezzo trasmissivo, che rendano impossibile la ricezione corretta dei dati. Tali disturbi vengono misurati e caratterizzati dai parametri PER (*Packet Error Rate*) e BER (*Bit Error Rate*) strettamente connessi fra loro. Un' altra sostanziale differenza con le trasmissioni wired è data proprio dal BER, che nelle trasmissioni radio è di diversi ordini di grandezza più elevato che nelle trasmissioni su cavo.
Il livello MAC definito dalla versione base dello standard IEEE 802.11 prevede due distinti modi di funzionamento e di accesso al canale, indicati come DCF e PCF.
Il DCF (*Distributed Coordination Function*) è un protocollo a contesa che si avvale di una funzione di coordinamento di tipo distribuito. E' la politica di accesso fondamentale ed obbligatoriamente presente, che consente a qualsiasi nodo che abbia dati da trasmettere, di partecipare ad una contesa per vincere il diritto di accesso al canale; non esiste nessuna unità centralizzata che stabilisca quale stazione ha diritto a inoltrare il suo traffico. Ovviamente si desidera che solo una stazione riesca ad impossessarsi del mezzo trasmissivo; nel caso poco probabile (ma pur sempre possibile) in cui più stazioni vincano contemporaneamente la contesa, avrà origine una collisione.
Il PCF (*Point Coordination Function*) è un protocollo di accesso con funzione di coordinamento di tipo centralizzato. Si tratta di una modalità opzionale pensata per cercare di garantire la QoS, ma che presenta ancora forti limitazioni che verranno superate dall'802.11e. Il PCF prevede l' esistenza di un nodo particolare chiamato PC (*Point Coordinator*), solitamente coincidente con l' AP, cui è affidato il compito di assegnare di volta in volta il diritto di trasmissione. Il PC effettua l' interrogazione ciclica di tutte le stazioni (*Polling*) e solo la stazione che riceve il *poll* può inviare i dati. Poiché la stazione che riceve il poll in un dato istante è una sola, non possono verificarsi collisioni.

I due metodi di accesso al canale possono essere presenti entrambi, non contemporaneamente, ma alternandosi temporalmente. Può essere presente solo il DCF, ma non è assolutamente possibile avere solo il PCF (figura 1.4). Il periodo temporale in cui viene utilizzato il DCF viene chiamato CP (*Contention Period*), mentre quello in cui viene utilizzato il PCF viene chiamato CFP (*Contention Free Period*).

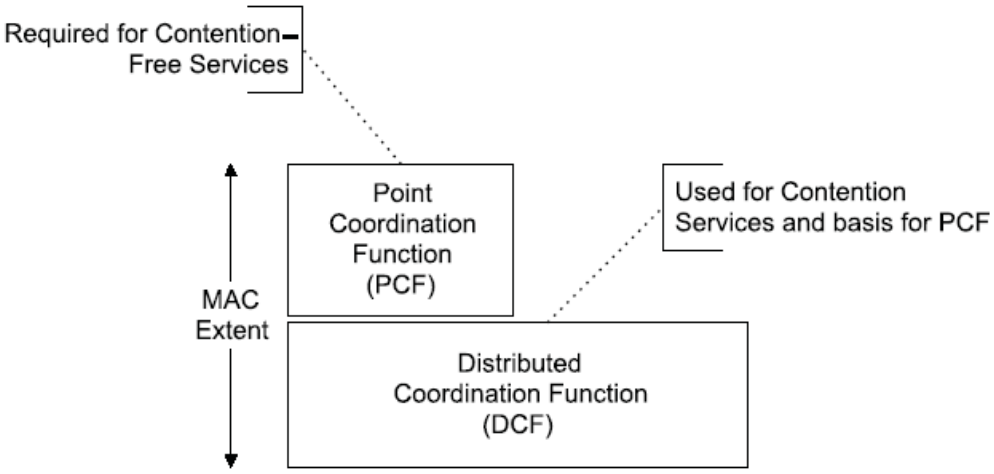


Figura 1.4: PCF e DCF

1.5.1 Organizzazione dei frame 802.11

L'unità di trasmissione al livello MAC per un nodo, sia esso una semplice stazione o l'AP, è il *frame* o anche MPDU (*Mac Protocol Data Unit*); un frame è un insieme di ottetti con una struttura ben definita e nota a priori sia al nodo trasmittente che a quello ricevente. All'interno del frame vi sono vari campi che contengono informazioni diverse.

Il tipico frame (di dati) è costruito unendo i contributi provenienti da tutti i livelli dello stack di protocolli ISO-OSI, in particolare per ogni livello il frame elaborato, proveniente dal livello superiore, cui si è aggiunto l' header, costituisce un PDU (*Protocol Data Unit*),

che a sua volta può essere visto come l'unione di un PCI (*Protocol Control Information*), ovvero l'header e un SDU (*Service Data Unit*), ovvero il payload: il livello applicativo genera i propri dati da trasmettere, questi vengono organizzati in segmenti dai protocolli di trasporto, solitamente il TCP (*Transfer Control Protocol*) o l'UDP (*User Datagram Protocol*), e in pacchetti dai protocolli di instradamento, l'IP (*Internet Protocol*), che ci aggiunge in testa il proprio header; questo blocco di dati costituisce il *frame body* del pacchetto a livello LLC che a sua volta diverrà il frame body del pacchetto a livello MAC e verrà indicato anche come MSDU (*MAC Service Data Unit*); proprio il livello MAC vi aggiunge ulteriori dati in testa (*MAC Header*) ed in coda (il campo *FCS*) fino a formare il MPDU, che è il *payload* di un pacchetto a livello e fisico e quindi viene anche indicato come PSDU (*Physical Service Data Unit*); Si possono distinguere tre tipi fondamentali di frame supportati dallo standard 802.11 [2]:

- *Data Frame,* usati per il trasporto dei dati sia durante il CP, sia durante il CFP; in realtà appartengono a questa categoria anche quei frame che non contengono dati, ma rendono possibile lo scambio di dati tra due nodi, come il *CFPoll* nel CFP.
- *Control Frame*, usati per il controllo dell' accesso al canale, come i frame *RTS* e *CTS* (di cui si parlerà più avanti) durante il CFP e i frame *ACK*, per la chiusura anticipata del CFP (frame *CFEnd*), per funzioni avanzate come il risparmio energetico (frame *PS Poll*), sono generati interamente al livello MAC e quindi il relativo eventuale MSDU non proviene dai livelli superiori.
- *Management Frame*, usati per lo scambio di informazioni di gestione, sono generati interamente a livello MAC, per cui i nodi che li ricevono, non passeranno tali frame ai livelli superiori; tipici esempi sono i frame Beacon, inviati dall' AP per la sincronizzazione e la temporizzazione, i frame per l' associazione e la dissociazione da una BSS, quelli per l' autenticazione e la deautenticazione.

La struttura generale di un frame 802.11 a livello MAC [2], riportata in figura 1.5, è la seguente:

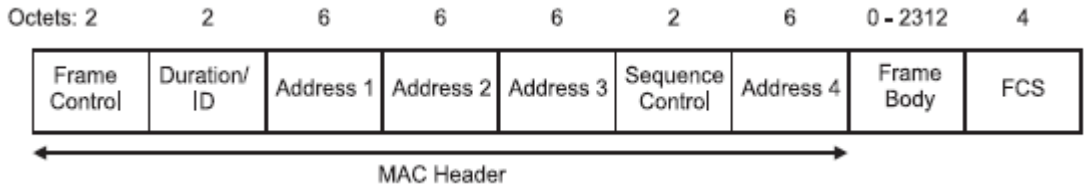
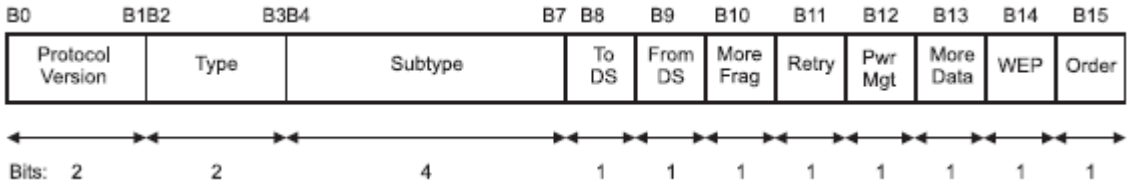


Figura 1.5: Struttura di un MAC frame 802.11

- un' intestazione (*MAC Header*), contenente informazioni di controllo, durata, indirizzamento e sequenza;
- un corpo (*Frame Body*), detto anche MSDU (*MAC Service Data Unit*), contenente informazioni specifiche del tipo di frame e corrispondente al PDU proveniente dal livello superiore;
- un campo di controllo dell'errore (FCS, *Frame Check Sequence*), che contiene un valore a 32 bit calcolato dal livello MAC del nodo mittente in funzione di tutti i precedenti ottetti presenti nel frame utilizzando un polinomio noto come CRC-32 (*Cyclic Redundancy Code*); il livello MAC del destinatario riesegue il calcolo, servendosi dello stesso polinomio generatore e controlla se si sono verificati errori in trasmissione.

Il primo campo del MAC Header, il Frame Control (illustrato in figura 1.6), è ulteriormente suddiviso nei seguenti sottocampi:



1.6: Struttura del Frame Control

- *Protocol Version*, che indica la versione del protocollo in uso (per 802.11 e versioni compatibili i due bit assumono sempre valore 00).
- *Type*, che serve a distinguere il tipo di frame inviato (00 Management, 01 Control, 10 Data, 11 Riservato).
- *Subtype*, che permette di distinguere il particolare sottotipo di frame;
- *To DS e From DS*, posti a 1 rispettivamente se il frame è indirizzato al DS o proviene dal DS.
- *More Frag*, posto a 1 se ci sono altri frammenti dello stesso MSDU da trasmettere;
- *Retry*, posto a 1 se il frame è una ritrasmissione;
- *Power Management*, posto a 1 se la stazione usa la modalità power save; è sempre 0 nei frame trasmessi dall' AP.
- *More data*, posto a 1 se ci sono degli MSDU nel buffer dell' AP in attesa di essere trasmessi alla STA operante in power save mode.
- *WEP* che, posto a 1, indica che i dati sono stati cifrati attraverso l' algoritmo WEP (*Wired Equivalent Privacy*).
- *Order*, settato ad 1 per indicare al nodo ricevente di processare i dati secondo l' ordine di arrivo.

Gli altri campi del MAC Header sono:

- *Duration/ID*, in cui è specificata la durata residua della trasmissione (variabile a seconda del tipo di frame), informazione utilizzata dalle stazioni riceventi per aggiornare il NAV; solo nella situazione power save poll tale campo riporta l'identificativo del mittente.
- *Address 1,2,3 4,* campi di indirizzamento la cui interpretazione è strettamente legata al valore dei campi To DS e From DS; ci sono quattro possibili eventualità:
    1) Se To DS = 0 e From DS = 0, significa che il DS non è coinvolto nella comunicazione, che quindi avviene tra due nodi appartenenti alla stessa BSS; allora nell' Address 1 compare l' indirizzo della

stazione destinataria, nell'Address 2 quello della effettiva sorgente dei dati, l' Address 3 può essere mancante o indicare l' indirizzo dell' AP, l' Address 4 non è utilizzato.

2) Se To DS = 0 e From DS = 1 significa che il frame non proviene direttamente dal nodo sorgente, ma dall' AP; in tal caso l' Address 1 contiene l' indiririzzo del destinatario finale, l' Address 2 l' indirizzo dell' AP di provenienza del frame (BSSID, *Basic Service Set Identification*) e l' Address 3 quello della vera sorgente dei dati, mentre l' Address 4 non è utilizzato.
3) Se To DS = 1 e From DS = 0, si ha la situazione complementare alla b), quindi l' Address 1 memorizza l' indirizzo dell' AP della BSS di destinazione (BSSID), l' Address 2 quello dell' effettiva sorgente del frame e l' Address 3 l'identificativo della stazione destinataria finale, mentre l' Address 4 non è utilizzato.
4) Se To DS = 1 e From DS = 1, significa che il frame viene trasmesso da un AP ad un altro attraverso un WDS (*Wireless Distribution System*); in questo caso l'Address 1 contiene l' indirizzo dell' AP ricevente, l' Address 2 dell' AP trasmittente, l' Address 3 della stazione destinataria, l' Address 4 della sorgente.

- *Sequence Control*, che indica l' ordine dei pacchetti e le duplicazioni che si rendono necessarie per la presenza di errori; può interessare l' intero MSDU o suoi frammenti, per cui tale campo è suddiviso in due sottocampi: il primo, di 4 bit, specifica il numero identificativo del frammento, che si incrementa di un' unità ad ogni trasmissione; il secondo, di 12 bit, specifica il numero identificativo dell'intero MSDU (anch' esso si incrementa di un' unità ad ogni trasmissione).

A questa struttura vi sono da aggiungere, in testa, le informazioni provenienti dal livello PHY [2][5][6][9] (fig.1.7) (chiamato anche PLCP, *Physical Layer Convergence Protocol*);

esse sono tipicamente costituite da un preambolo di sincronizzazione e da un'intestazione che, ovviamente, dipendono dal particolare livello fisico e dalla specifica modulazione usata; il livello PHY inoltre, traduce in segnale elettrico il pacchetto così ottenuto e lo trasmette sul canale fisico tramite una opportuna modulazione. In caso di pacchetto ricevuto il percorso da seguire è invertito.

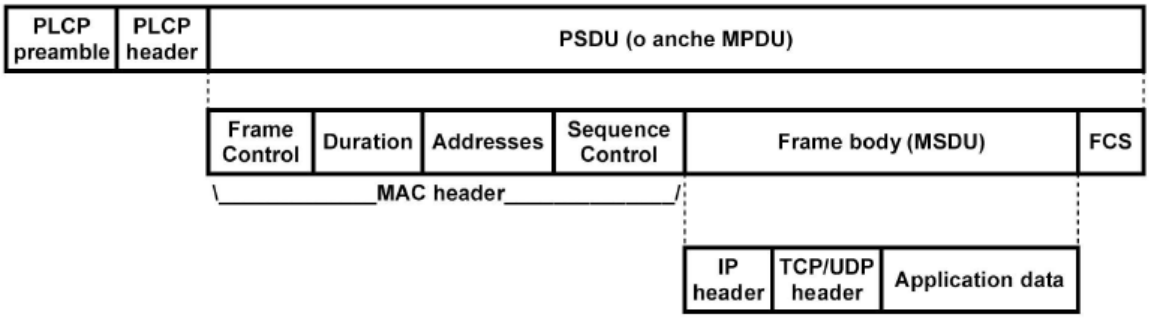


Figura 1.7: Struttura completa di un frame 802.11

Prima di inviare un qualsiasi frame, un nodo deve in generale verificare che il canale sia libero (assenza di portante) per un certo intervallo di tempo, detto IFS (*Interframe Space*), che varia in funzione della situazione specifica. Infatti, nello standard di base 802.11, esistono quattro distinti IFS:

1) *SIFS* (*Short IFS*), che è il più breve intervallo temporale che può sussistere tra due frame successivi e indica il tempo necessario al nodo per convertire il suo modo di funzionamento passando dalla fase di trasmissione a quella di ricezione, quindi è sostanzialmente il tempo necessario a decodificare il pacchetto entrante; il SIFS è usato come intervallo di separazione tra un frame e il relativo ACK, oppure tra due MPDU all' interno di un burst di dati; la sua durata dipende dal livello fisico utilizzato.
2) *PIFS* (*PCF IFS*), usato esclusivamente nella modalità PCF, è il tempo che l' AP (o il PC) deve attendere prima di guadagnare l' accesso al canale per consentire alle stazioni di trasmettere in modalità PCF; il PIFS è pari ad un SIFS più uno slot time

(*aslotTime*), l' unità temporale anch' essa dipendente dal particolare livello fisico (vedi Tabella 1.III), ed è minore del DIFS (definito subito dopo), affinché l' AP abbia la precedenza rispetto ad ogni altra stazione nell' accedere al mezzo trasmissivo all' inizio di un CFP:

$$PIFS = SIFS + aSlotTime > SIFS$$

3) *DIFS* (*DCF IFS*), usato esclusivamente nella modalità DCF, rappresenta il tempo (con il canale libero) che una stazione deve attendere prima di iniziare la contesa; il suo valore è dato da un PIFS più aSlotTime:

$$DIFS = PIFS + aSlotTime > PIFS > SIFS$$

4) *EIFS* (*Extended IFS*), che è l' interframe più lungo usato solo in modalità DCF quando una stazione, ricevendo un pacchetto, rileva degli errori di trasmissione tramite il controllo del CRC; la sua durata dipende dal livello fisico secondo la seguente espressione:

$$EIFS = DIFS + 8 \cdot (\textit{Durata ACK}) + (\textit{Durata Preambolo e Intestazione PLCP}) > DIFS$$

È molto importante notare le relazioni di disuguaglianza che vi sono fra questi intervalli: ciò, infatti, permette il rispetto di determinate priorità di accesso al canale determinate dalle singole situazioni in cui ci si trova.

1.5.2 DCF (Distribuited Coordination Function)

L'insieme di regole o protocollo di accesso al canale definito dallo standard e indicato col nome di DCF [2] costituisce il metodo fondamentale di funzionamento di una rete IEEE

802.11 e, quindi, l'unico obbligatoriamente presente in tutte le implementazioni. Questo è un protocollo *distribuito*, nel senso che l'ordine di accesso al canale viene determinato da un algoritmo che lavora simultaneamente su tutte le stazioni che partecipano alla contesa. Il tipo di approccio usato dal DCF è indicato come CSMA/CA (*Carrier Sensing Multiple Access / Collision Avoidance*). Il DCF è un protocollo finalizzato a minimizzare per quanto

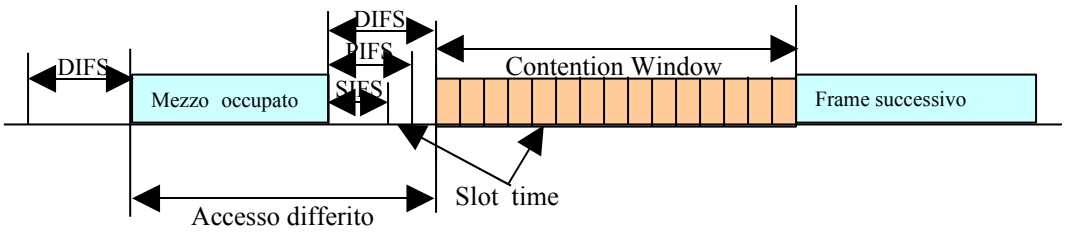


Figura 1.8: Accesso al canale mediante DCF

possibile le probabilità di collisioni: ciò si ottinene cercando di rendere casuale e differenziato l'istante di accesso al canale da parte di ciascuna stazione che ha dati da trasmettere. La generica procedura di trasmissione di un frame di dati è schematizzata in figura 1.8. Ciascuna stazione che vuole trasmettere un frame di dati ascolta il canale ed aspetta che finiscano le eventuali trasmissioni correnti; una volta che il canale diviene libero, viene atteso un DIFS, e poi ha inizio la contesa; al termine di questa, solo la stazione vincitrice trasmette il proprio frame di dati; terminata la sequenza di trasmissione per un frame dati il canale si libera e la procedura si ripete. Se invece il canale, nel momento in cui la stazione si accinge per la prima volta alla trasmissione di un frame, dovesse risultare subito libero, non c'è bisogno del ricorso al meccanismo di contesa: la trasmissione inizia semplicemente dopo aver constatato la condizione di canale libero per un tempo almeno pari ad un DIFS. Più in dettaglio, come mostrato in figura 1.9, la stazione che vince la contesa trasmette il proprio frame di dati; il nodo ricevente attende un SIFS dopo la fine della ricezione del frame di dati e genera un frame di ACK ad indicare che la trasmissione è

andata a buon fine ; in particolare il frame di ACK è generato solo se non sono avvenute collisioni e se il controllo degli errori a livello fisico, grazie all'uso dei CRC, non indichi la presenza di errori nei bit ricevuti: il verificarsi di queste due condizioni implica che il frame è stato ricevuto correttamente.

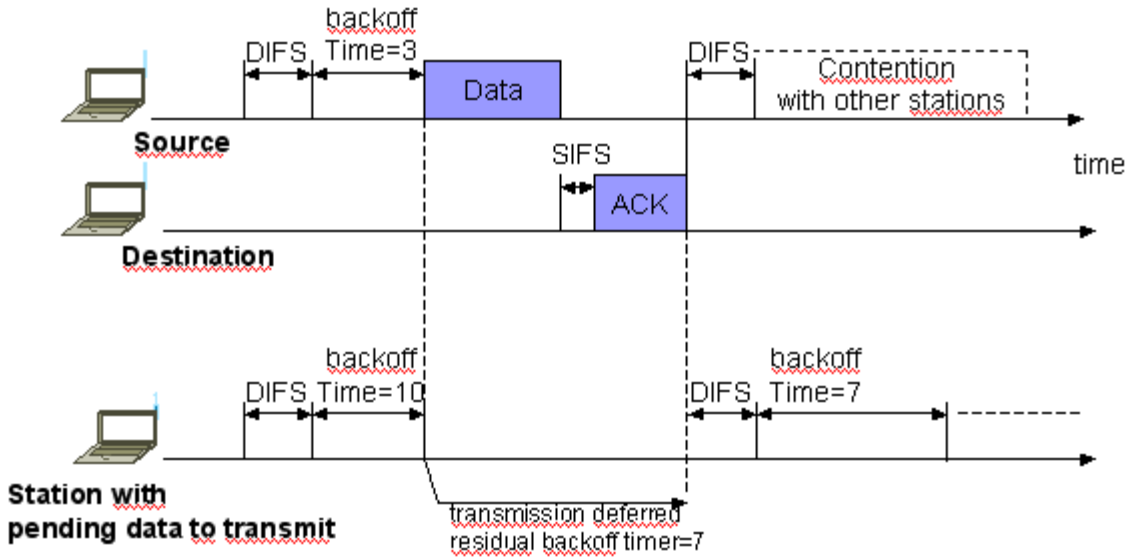


Figura 1.9: Sequenza di scambio per la trasmissione di un frame dati in DCF

Potrebbe comunque accadere che due o più stazioni, a seguito della contesa, inizino le trasmissioni nello stesso istante; come già detto questa situazione non è rilevabile dalle stazioni trasmittenti che, quindi, continueranno a trasmettere il frame di dati fino alla fine; sarà l'assenza dell'ACK dopo un SIFS ad indicare alla stazione trasmittente l'errata trasmissione del frame di dati e, in tal caso, dopo un DIFS a partire dall'istante in cui il canale è ritornato libero avrà inizio una nuova contesa per la ritrasmissione degli stessi frame. La contesa fra le diverse stazioni avviene secondo un meccanismo probabilistico denominato algoritmo di backoff casuale esponenziale (*Exponential Random Backoff*

Algorithm): ciascuna stazione estrae un valore intero casuale secondo una distribuzione uniforme nell'intervallo [0, CW], dove CW (*Contention Window*) è un parametro definito dallo standard che varia dinamicamente all'interno di ciascuna stazione nell'intervallo [CWmin, CWmax]. Il valore estratto indica il numero di *aSlotTime* che la stazione dovrà ulteriormente attendere, dopo il DIFS, prima di iniziare a trasmettere il proprio frame di dati. Ogni nodo possiede un proprio timer interno, chiamato *Backoff Timer* che conserva il numero corrente di *slot* ancora da attendere; passato un tempo pari ad un *aSlotTime*, se il canale dovesse essere ancora libero e non sono iniziate altre trasmissioni, il timer viene decrementato; solo quando il timer raggiunge il valore 0 la stazione avrà vinto la contesa e quindi inizierà la trasmissione. Se invece durante il conteggio alla rovescia in corrispondenza della fine di uno *slot* temporale il canale dovesse risultare occupato, il valore del timer non verrà ulteriormente decrementato, ma sarà congelato e conservato, per poi essere utilizzato come valore iniziale nella successiva contesa. In questo modo si crea una sorta di precedenza per le stazioni che da più tempo sono in attesa di trasmettere. Una volta vinta la contesa e trasmesso il frame, viene estratto un nuovo valore di Backoff da utilizzare nella successiva contesa; questo sarà sempre distribuito uniformemente nell'intervallo a [0, CW], ma il CW sarà variato a seconda che la trasmissione appena conclusa sia andata a buon fine o meno; in particolare, in caso di presenza dell' ACK, il nuovo CW verrà settato al valore CWmin; in caso di assenza (solitamente causata da una collisione) il CW verrà modificato secondo la formula:

$$CW = 2^{2+i} - 1 \qquad 1 \leq i \leq 6$$

dove i rappresenta il numero di tentativi di trasmissione di uno stesso frame. L'applicazione di tale formula si traduce in un progressivo raddoppio di CW, non superando però mai il valore CWmax. In figura 1.10 vi è un esempio di come il CW possa

variare a seguito di successive assenze del frame di riscontro (ACK). I due parametri CWmin e CWmax (come anche *aSlotTime*) sono definiti in funzione del livello fisico utilizzato (vedi tabella 1.III).

Per quanto riguarda i metodi con cui una stazione distingue lo stato di canale occupato da quello di canale libero da trasmissioni, vi è principalmente un *Carrier Sensing* di tipo fisico (in cui fisicamente viene misurata la potenza del segnale ricevuto) a cui viene affiancato un *Virtual Carrier Sensing*: nell'intestazione MAC di ogni tipo di frame trasmesso vi è un campo *Duration Field* in cui viene specificata in unità di microsecondi, a partire dalla fine

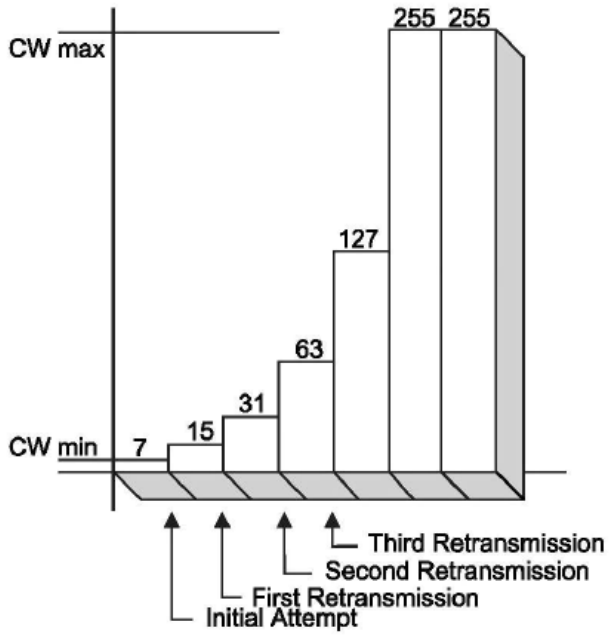


Figura 1.10: Valori possibili del parametro CW

della trasmissione del frame stesso, la durata necessaria a completare la corrente sequenza di scambio frame. In tal modo, ad esempio, nel *Duration Field* di un frame di dati sarà specificato il tempo occorrente per la trasmissione di un ACK sommato ad un SIFS (che intercorre fra il frame dati e l'ACK stesso). Tutte le stazioni in ascolto che ricevono il

frame, aggiornano un proprio registro interno indicato come NAV (*Network Allocation Vector*), in funzione del *Duration Field* ricevuto.

Il virtual carrier sensing è utile soprattutto nelle situazioni in cui vi sono stazioni non in grado di ascoltare le trasmissioni di tutte le altre; questo è il problema delle *hidden stations*: potrebbero esserci nella stessa BSS due stazioni non in visibilità radio fra loro (ma che comunque vedono l'AP); se a una delle due inizia la trasmissione di un frame di dati diretto all'AP, l'altra non riuscirà in alcun modo ad accorgersi dello stato occupato del canale e potrebbe quindi iniziare una propria trasmissione dando dunque origine ad una collisione e, quindi, alla perdita di entrambi i frame. La soluzione si è ottenuta con l'uso di due nuovi frame e con l'ausilio del virtual carrier sensing; la stazione che vince la contesa, prima di trasmettere il frame di dati, invia un frame RTS (*Request To Send*); dopo un SIFS la stazione destinataria trasmetterà un frame CTS (*Clear To Send*); solo dopo un SIFS dalla ricezione del CTS inizierà effettivamente la trasmissione del frame di dati seguito dall'eventuale frame di riscontro (fig.1.11). Tutte le stazioni che sono in visibilità radio con almeno uno fra nodo sorgente e destinatario (e sono queste le stazioni che potrebbero originare una collisione) saranno sicuramente in grado di aggiornare i propri NAV leggendo i duration field presenti nei frame di RTS o CTS e quindi la successiva trasmissione del frame di dati sarà protetta dal meccanismo del virtual carrier sensing (nel frame RTS il *Duration Field* copre la durata dell'intera trasmissione fino all'ACK del destinatario). Una eventuale collisione potrebbe ancora avvenire, ma tale evenienza è ristretta al solo intervallo in cui viene inviato il frame RTS e al SIFS immediatamente successivo. La modalità appena descritta che fa uso dei frame RTS/CTS ha lo svantaggio, soprattutto se i data frame sono di dimensioni molto piccole, di diminuire l'efficienza di utilizzo del canale. È per questo che viene solitamente utilizzata solo quando il frame di dati da trasmettere è sufficientemente grande, in particolare quando la sua dimensione in byte supera il valore *dot11RTSThreshold*, che è un parametro variabile e gestito indipendentemente da ciascuna stazione. Tale parametro regola anche il numero massimo

di ritrasmissioni per uno stesso data frame nel caso di mancato riscontro: infatti, a seconda se il data frame sia minore o maggiore di *dot11RTSThreshold,* il numero massimo di ritrasmissioni permesse prima di scartare il data frame è pari rispettivamente a *ShortRetryLimit* o a *LongRetryLimit* che di default valgono 4 e 7. Per pacchetti di dati eccessivamente grandi o anche semplicemente quando gli errori nel canale non tollerano data frame molto grandi, lo standard prevede l'uso (l'implementazione è facoltativa) di un meccanismo di frammentazione. In pratica al posto della trasmissione di un unico data frame grande vengono trasmessi molti frammenti piu piccoli e ciascuno di essi deve essere corrisposto da un frame di ACK; la ricomposizione dei frammenti e ricostruzione del frame originario viene fatta al livello MAC della stazione ricevente. All'interno del *fragment burst* tutti i frame sono separati da un SIFS.

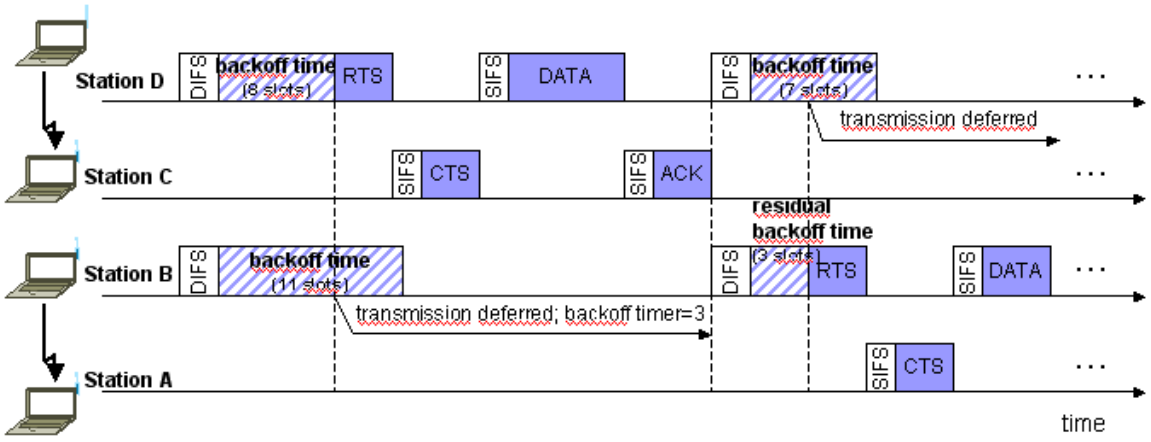


Figura 1.11: Trasmissione dati con l' uso dei frame RTS e CTS

1.5.3 PCF (Point Coordination Function)

Il PCF è un protocollo di accesso al mezzo senza contesa[2], implementato ed utilizzato facoltativamente e, comunque, sempre in appoggio al DCF, con lo scopo di fornire un supporto alla QoS. Nel caso si voglia usare il PCF, esso deve alternarsi nel tempo con il DCF, avremo per cui una fase chiamata CFP (*Contention Free Period*), in cui viene usato il protocollo PCF e una fase chiamata CP (*Contention Period*), in cui si usa invece il DCF, come possiamo vedere in figura 1.12. Nel PCF è l' unità di coordinamento PC (*Point Coordinator*), solitamente coincidente con l' Access Point, a designare quale stazione deve trasmettere; all'interno di un CFP, una STA può scegliere di rispondere ad un contention-free poll (il CF-Poll) inviato dal PC, una STA capace di rispondere ad un CF-Poll è segnata come CF-Pollable e può richiedere di essere interrogata da un PC attivo all'interno di una BSS, inoltre le CF-Pollable STAs e il PC non usano RTS/CTS prima della trasmissione dei frame; quando interrogata dal PC, una STA, può emettere solamente un MPDU che può essere a qualsiasi destinazione (non solo al PC), e può confermare l'avvenuta ricezione di un frame dal PC tramite la tecnica del "*Piggybacking*".

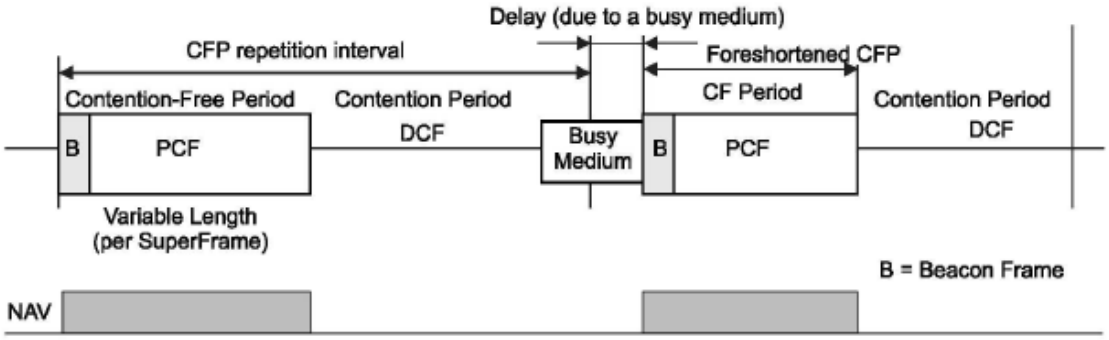


Figura 1.12: Alternanza tra CFP e CP

1.5.3.1 Struttura e temporizzazione di un CFP

L'AP invia periodicamente dei particolari frame chiamati *Beacon*, il cui scopo è quello di identificare e sincronizzare la BSS; l' intervallo di ripetizione dei beacon è indicato con l' acronimo TBTT (*Target Beacon Transition Time*). Subito dopo il beacon, inizia una fase chiamata CFP (*Contention Free Period*), e successivamente, una fase chiamata CP (*Contention Period*); la somma del CFP e del CP costituisce un *Superframe* che in generale è un multiplo intero del TBTT; il periodo di ripetizione del CFP è definito dal parametro *CFPRepetitionInterval* e coincide con il Superframe. Ogni CFP è collocato subito dopo la trasmissione del beacon che dà inizio al Superframe; il duration field presente in questo frame consente alle stazioni di aggiornare il proprio NAV, in modo da proteggere tutta la fase CFP tramite il virtual carrier sensing: infatti tutte le stazioni che non utilizzano il PCF vedranno il canale occupato e non potranno trasmettere se non mediante DCF; la sola stazione che avrà il diritto di trasmettere e che controllerà tutta la fase CFP è il PC, tutte le altre potranno solo rispondere al PC. La durata del CFP viene inizialmente settata al valore massimo, indicato dal parametro *CFPMaxDuration*, ma il CFP può essere terminato prematuramente dal PC tramite l'invio di un frame *CF-End*. I beacon sono inviati ad intervalli regolari, in modo che ogni stazione sappia precisamente quando arriverà il prossimo ma, se nell'istante previsto per la trasmissione di un beacon il canale risulta occupato, esso subirà un leggero ritardo e verrà inviato non appena il mezzo torna ad essere libero. Il PC, infatti, per impegnare il canale, dovrà aspettare che esso si mantenga libero solo per un PIFS: ciò gli dà maggiore priorità di accesso delle altre stazioni, che invece hanno un tempo di attesa almeno pari ad un DIFS.

Bisogna aggiungere che un beacon contiene, oltre al campo CFPMaxDuration, anche il campo *CFPDurRemaining*, che esprime la durata residua prima che il CFP termini. Inoltre va precisato che i beacon vengono trasmessi non solo durante il CFP, ma anche durante il CP (sebbene in quest' ultimo caso il traffico è smaltito solo per contesa), perché essi hanno il compito di mantenere la sincronizzazione dei timer locali delle stazioni: tali beacon saranno contraddistinti da CFPDurRemaining = 0.

Ogni CFP (e quindi ogni Superframe) ha inizio con un particolare beacon contenente il campo DTIM (*Delivery Traffic Indication Message*), in cui viene specificata la durata del Superframe come un multiplo intero dell' intervallo di ripetizione dei DTIM (che a sua volta è un multiplo di TBTT).

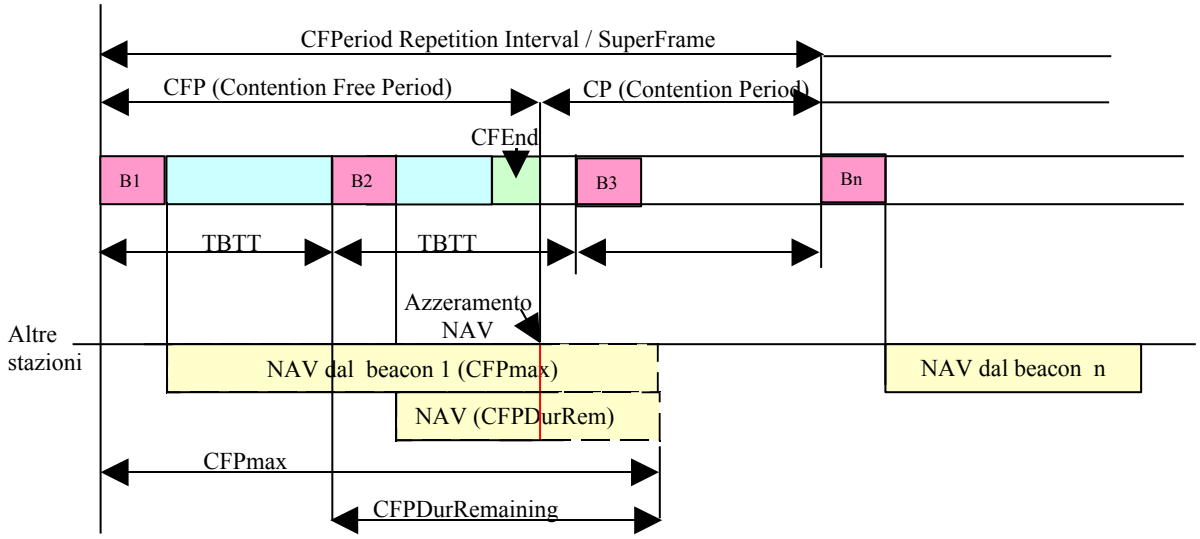


Figura 1.13: Schema temporale di un Superframe

In figura 1.13 è illustrato il diagramma temporale di un Superframe della durata di un periodo di DTIM, a sua volta pari a tre TBTT.

Come già detto, all' interno del CFP sarà sempre e solo il PC ad iniziare le trasmissioni e ad interrogare le stazioni, è quindi evidente, come il PCF sia un protocollo più efficiente in termini di uso del canale del DCF dato che, nella maggior parte dei casi, tutti i frame sono distanziati semplicemente da un SIFS. Dopo un tempo di un SIFS dall' invio del primo beacon (quello che dà inizio al Superframe e attiva il CFP), il PC interrogherà la prima stazione inviandole un frame *CFPoll* per chiederle se ha dati da trasmettere; a distanza di un SIFS la stazione risponderà con un frame dati (unito all'ACK di conferma della corretta ricezione del poll), che sarà seguito dall' eventuale ACK da parte del PC ancora dopo un

SIFS. Terminata la trasmissione dell' ACK, il PC attenderà un SIFS e passerà ad interrogare la prossima stazione. Nel caso la stazione interrogata non abbia dati da trasmettere, essa risponderà con un frame CFNull; nel caso lo stesso PC abbia dati pendenti per la stazione, le invierà un frame composto da *Data+CFPoll*, al quale la stazione risponderà con un ACK dopo aver atteso un SIFS. Nel PCF sono spesso accorpati frame di tipo diverso al fine di migliorare l' efficienza del protocollo:

- *Data+CF-Poll, Data+CF-Ack+CF-Poll, CF-Poll,* and *CF-Ack+CF-Poll*, che possono essere inviati solo dal PC.
- *Data, Data+CF-Ack, Null Function, and CF-Ack*, che possono essere inviati da un PC o da qualunque CF-Pollable STA.

Tutti questi frame, siano essi inviati dal PC o dalle stazioni interrogate, come già detto, saranno sempre intervallati da un SIFS, che è il più piccolo tempo di interframe previsto dallo standard. Nel caso in cui non riceva alcuna risposta dalla stazione interrogata (a causa di errori del canale o per avaria della stazione) il PC attenderà un PIFS prima di riprendere il controllo del canale rivolgendosi alla prossima stazione oppure, se ha terminato il ciclo, inviando il CFEnd che determinerà la chiusura anticipata del CFP e consentirà a tutte le stazioni di azzerare il NAV e dare inizio al CP. Si deduce che durante un CFP non può verificarsi un tempo di attesa maggiore di un PIFS.
Le stazioni da interrogare e la corrispondente priorità vengono desunte da uno specifico elenco, noto come *Polling List*, nel quale compaiono tutte le stazioni che si sono associate alla BSS e che hanno richiesto esplicitamente l' inserimento nella lista. Qualsiasi cambiamento di stato, inteso come ingresso o uscita dalla lista, comporta una riassociazione alla rete. L' ordine di precedenza nel polling è stabilito dal parametro AID (*Association ID*), un numero identificativo assegnato dall' AP ad ogni stazione in fase di associazione

alla BSS, con valore crescente in base all' ordine di arrivo: le stazioni inserite nella Polling List verranno interrogate in ordine crescente di AID durante il CFP.

1.5.4 Livello PHY 802.11

Il livello PHY [2][6][5][9] quello più basso previsto dal modello ISO-OSI; esso si interfaccia
direttamente con l'hardware ed è collocato logicamente al di sotto del livello
MAC; in particolare al livello PHY sono affidati tre compiti:

1) Rilevamento della portante fisica (*Physical Carrier Sense*), che consente di determinare se il canale è libero o occupato.
2) Trasmissione sul mezzo fisico, mediante opportuna modulazione del segnale, dei singoli bit che costituiscono il frame proveniente dal livello MAC.
3) Ricezione, mediante appropriata demodulazione del segnale, dei singoli bit che compongono il frame in arrivo.

Come illustrato in figura 1.14, l' architettura logica del livello PHY è strutturata in questo modo:

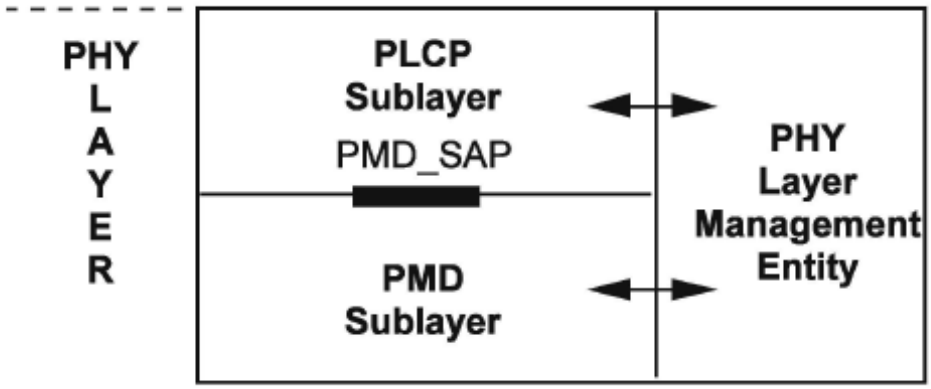


Figura 1.14: Architettura logica del livello PHY

- *Physical Layer Management Entity* (*PLME*), che costituisce l' interfaccia col livello MAC sovrastante e gestisce i due seguenti sottolivelli.
- *PLCP (Physical Layer Convergence Procedure) Sublayer* che, in trasmissione, provvede alla conversione di un frame a livello MAC in un frame a livello PHY (il viceversa in ricezione), aggiungendo (eliminando) i campi specifici di questo livello; più precisamente vengono inseriti in testa al PSDU un preambolo (*PLPC Preamble*) e un' intestazione (*PLCP Header*) che conducono alla formazione di un PPDU (*PLCP Protocol Data Unit*).
- *PMD (Physical Medium Dependent) Sublayer*, che gestisce l' hardware specifico a disposizione, provvedendo alla particolare modulazione e demodulazione per la trasmissione e ricezione dei segnali.

Diversi PHYs sono definiti come parte dello standard IEEE 802.11. ogni PHY può consistere di due funzioni di protocollo come:

a) Una funzione di convergenza di livello fisico, che adatta le capacità del physical medium dependent (PMD) system al servizio di PHY. Questa funzione è supportata

dal Physical layer convergence procedure (PLCP) che definisce un metodo di mappatura del MAC sublayer protocol data units (MPDUs) dell' IEEE 802.11 in un formato di framing appropriato per ricevere dati e informazioni di management tra due o più STAs che usano l'associato PMD system.

b) Un PMD system, la cui funzione definisce le caratteristiche di un metodo di trasmissione e ricezione dati tramite un wireless medium (WM) tra due STAs.

Ogni sottolivello di PMD può richiedere la definizione di un PLCP unico.

Il livello PMD ha il compito finale di trasformare i singoli bit in segnale fisico da trasmettere sotto forma di onde radio (nel caso delle modulazioni FHSS, DSSS, OFDM, HR-DSSS) o di luce infrarossa (nel caso di modulazione IR).

Si faccia riferimento alla tabella 1.I per le prestazioni fornite dai vari schemi di modulazione utilizzati. Il tipo di modulazione adottata influenza notevolmente le prestazioni della rete sia in termini di bit rate, sia in termini di immunità al rumore e alle interferenze. E' per questo motivo che nelle successive versioni dello standard IEEE 802.11 sono state introdotte tecniche di modulazione/demodulazione sempre più evolute che, in pratica, hanno sostituito la DSSS (*Direct Sequence Spread Spectrum*), la FHSS (*Frequency Hopping Spread Spectrum*) e la IR (*InfraRed*), usate nella prima stesura dello standard, che consentono velocità di 1Mbps e 2Mbps. Ci riferiamo alla OFDM (*Orthogonal Frequency Division Multiplexing*) e la HR-DSSS (*High Rate Direct Sequence Spread Spectrum*), versione potenziata della classica DSSS, introdotte la prima dagli standard 802.11a e 802.11g e la seconda dall' 802.11b, con le quali si raggiungono velocità rispettivamente di 54 e 11 Mbps (vedi tabella 1.I).

Le modulazioni a spettro esteso (DSSS, FHSS e HR-DSSS) [2] sfruttano la caratteristica di distribuire la potenza del segnale su un esteso intervallo di frequenze. Tale procedura, se da un lato richiede una maggiore ampiezza di banda utile, dall' altro consente livelli di potenza di trasmissione più bassi, un migliore rapporto segnale-rumore (*SNR*) e un' elevata

immunità alle interferenze e intercettazioni. Il risultato è una minore probabilità d' errore (*BER*, *Bit Error Rate*) e quindi una più fedele ricostruzione del segnale in ricezione.

Parametro	802.11a	802.11b	802.11g
aSlotTime	9µs	20µs	short=9µs long=20µs
aSIFSTime	16µs	10µs	10µs
aAirPropagationTime	< 1µs	< 1µs	< 1µs
aCWmin	15	31	31
aCWmax	1023	1023	-
aMACProcessingDelay	< 2µs	Non Applicabile	< 2µs
aMPDUMaxLenght	4095	4095	4095

Tabella 1.III: Parametri caratteristici dei livelli fisici 802.11

1.6 Supporto alla QoS in 802.11

Lo standard IEEE 802.11, per come è stato presentato, risulta particolarmente adatto per fornire servizi *best effort*, ovvero, senza garanzie di throughput e delay. Generano questo tipo di servizi, ad esempio, applicazioni quali HTTP (*Hyper Text Transfer Protocol*), FTP (*File Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*). Ci si rende subito conto come lo schema di funzionamento di IEEE 802.11 sia del tutto inadeguato nel caso di traffici *real-time*, per i quali, è necessario che siano rispettati stringenti vincoli, sui tempi di consegna dei pacchetti. Esempi di tali traffici sono la trasmissione di voce fra due utenti tramite VoIP (*Voice Over IP*), le videoconferenze (ad esempio con lo standard H.263), lo streaming di filmati (ad esempio con la codifica MPEG). Alcuni di essi implicano uno scambio di dati interattivo nelle due direzioni (*two-way*) come le telefonate e le videoconferenze, altri invece sono diretti in un solo verso (*one-way*) come i filmati. In questo tipo di traffici, affinchè vengano rispettati i vincoli di QoS (*Qualità del Servizio*) è

necessario avere bassi ritardi di trasmissione (*delay*), piccole percentuali di perdita di pacchetti (*packet loss*), bassa varianza nei ritardi (*jitter*), sufficiente *banda* a disposizione.

Applicazione	Tipo	Banda	Target Delay
VoIP	two-way	4-64 kbps	$\simeq 150ms$
Video Conferenza	two-way	16-384 kbps	$\simeq 150ms$
Streaming Video	one-way	56 kbps-2 Mbps	$\simeq 30s$

Tabella 1.IV: Caratteristiche di alcuni flussi real-time

I protocolli che hanno come scopo quello di garantire la QoS, devono tenere in conto i diversi vincoli dei traffici *real-time* in funzione dell'applicazione specifica; di fondamentale importanza è il vincolo temporale: ogni flusso infatti, sopporterà al più un certo ritardo nella trasmissione dei pacchetti, affinché esso sia correttamente utilizzabile, indicato come *target delay*. Sebbene il PCF sia stato inizialmente concepito per cercare di garantire alle stazioni che ne fanno uso di avere un accesso esclusivo e controllato al canale, esso ha alcune limitazioni che gli impediscono di essere un buon candidato come protocollo che garantisca la QoS in una wireless lan 802.11; il suo più grosso limite è costituito dal fatto che esso dà la possibilità alla stazione che riceve il *poll* di trasmettere al più un frame di dati e , quindi, non tiene assolutamente conto della reale necessità di banda della stazione stessa, un altro limite è costituito dal fatto che l'istante in cui verranno trasmessi i beacon può variare proprio a causa dell'occupazione del canale: ad ogni TBTT il PC considera il beacon come il prossimo frame da trasmettere, ma ciò può avvenire solo se il mezzo è risultato libero per almeno un PIFS; in funzione dello stato del canale (se è libero oppure occupato), si può registrare un ritardo nel beacon frame, la durata di tale ritardo si ripercuote sul numero di MSDU da trasmettere, incidendo notevolmente sulla qualità della comunicazione[12].

In tale contesto si inserisce il lavoro svolto dal *Woking Group IEEE 802.11e* il cui compito è quello di cercare un modo che permetta il rispetto dei vincoli della QoS nelle WLAN.

Dobbiamo tenere presente una cosa molto importante: una WLAN rappresenta solo l'ultimo (o il primo) *hop* nel percorso che un pacchetto compie per andare dal mittente al destinatario, si capisce quindi come sia necessario avere al suo interno dei ritardi ben inferiori rispetto a quelli indicati nella tabella 1.IV. Considerando solo l'ultimo (o il primo) hop all'interno di una rete wireless, il ritardo totale accusato da un pacchetto sarà dato dalla somma di quattro contributi:

1. il ritardo di elaborazione (piccolissimo ma comunque non nullo);
2. il ritardo di accodamento, in quanto i pacchetti vengono conservati in una coda in attesa di esser trasmessi, usualmente serviti secondo una politica FIFO (*First In First Out*);
3. il ritardo di trasmissione, dipendente direttamente dalla velocità di trasmissione;
4. il ritardo di propagazione fisica del segnale trasmesso (anch'esso piccolissimo).

Compito di un protocollo che miri a garantire la QoS, è quello di assicurare che la somma di questi ritardi sia sempre inferiore al target delay.

1.6.1 Gruppo di lavoro IEEE 802.11e

Come già visto, per la gestione di traffici multimediali e real-time la versione ufficiale dello standard non fornisce una soluzione soddisfacente. È per questo che il working group IEEE 802.11e sta studiando la possibilità di estendere la versione base dello standard ed integrarla sostanzialmente con una serie di concetti innovativi come ad esempio l'introduzione di una nuova funzione di coordinazione: HCF (*Hybrid Coordination Function*) che introduce due nuovi protocolli di accesso al canale:

- contention-base access, EDCA (*Enhanced Distributed Channel Access*);
- contention-free access, HCCA (*Hybrid Coordination Function Controlled Channel Access*);

che per questioni di compatibilità con l' hardware già esistente non sostituiranno ma affiancheranno il DCF e il PCF, come indicato dallo schema logico in figura 1.15.

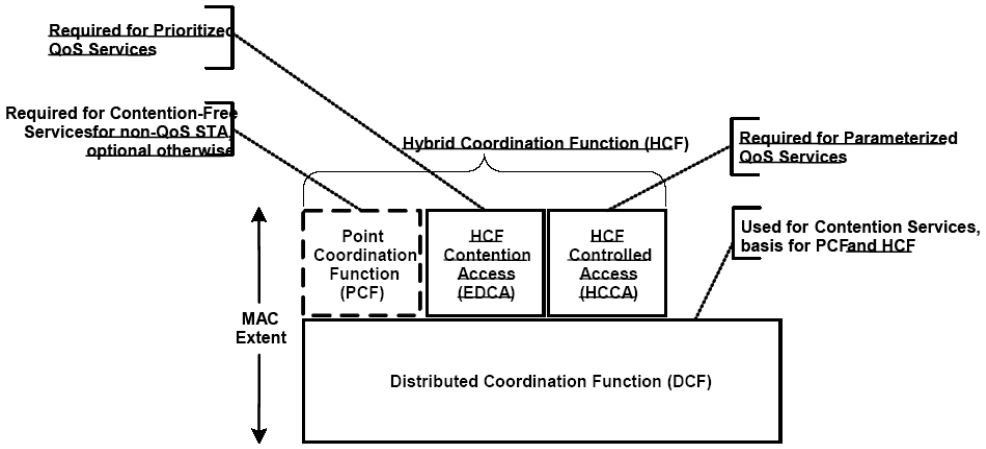


Figura 1.15: Architettura logica del livello MAC 802.11e

Lo studio è ancora in corso, per cui non si parla ancora di standard IEEE 802.11e: sono disponibili diversi *draft* che evolvono nel tempo e la cui versione definitiva porterà alla definizione di uno standard ufficiale. In questo lavoro di tesi si fa riferimento al Draft 8.0, pubblicato nel febbraio 2004 [12].

L' EDCA estende sostanzialmente il modo di funzionamento del DCF, esso rimane ancora un metodo di accesso al canale a contesa, quindi di tipo probabilistico, ma introduce il concetto di differenziazione dei vari tipi di traffico, offrendo maggiori opportunità di trasmissione ai flussi a più alta priorità. Per far sì che i flussi non vengano trattati tutti nello stesso modo, è stato introdotto il concetto di categoria di accesso al canale AC (*Access Category*) e l' implementazione di 4 code separate (una per ogni AC) (Tab.1.V):

- *Background (AC_BK)*
- *Best Effort (AC_BE)*
- *Video (AC_VI)*
- *Voice (AC_VO)*

AC	Access Category
AC_BE	Best Effort
AC_BK	Background
AC_VI	Video
AC_VO	Voice

Tabella 1.V: AC all'interno di una QSTA

in ciascun nodo di trasmissione, come visualizzato in figura 1.16: sono presenti 4 code, ognuna delle quali conterrà solo pacchetti appartenenti ad una determinata classe di traffico. Viene inoltre ripreso il concetto delle 8 TC (*Traffic Category*), già introdotto dallo standard IEEE 802.1D, portando così a 8 il

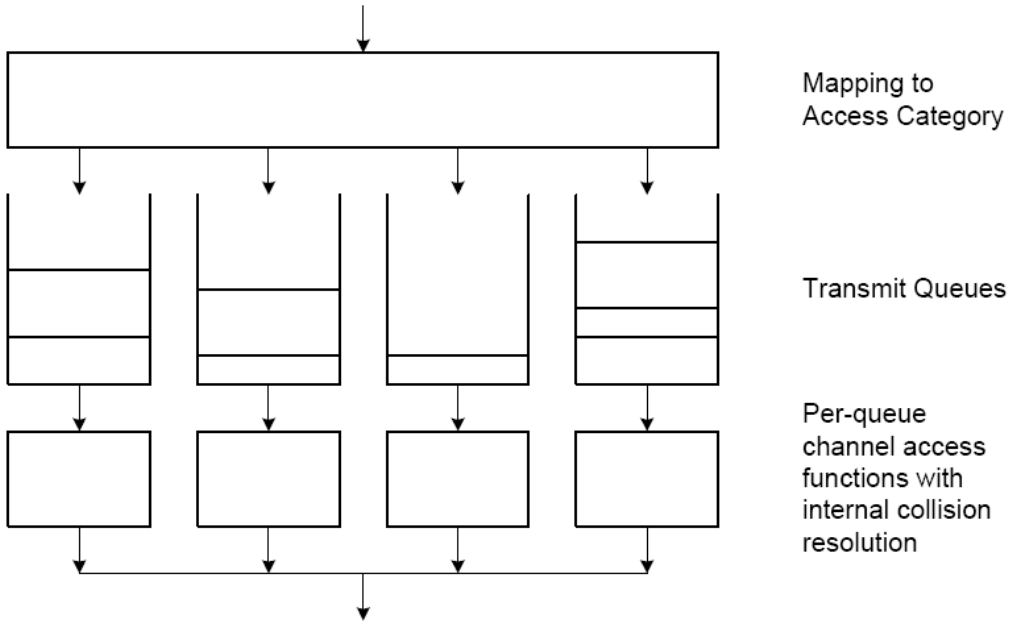


Figura 1.16: Implementazione di code distinte per ogni AC

numero di code implementabili in ciascuna QSTA. Fra AC e TC vi è una precisa mappatura come mostrato nella tabella 1.VI [12].

Priorità	UP (*User Priority*)	AC (*Access Category*)	Tipologia di flusso
bassa	1	AC_BK	Background
↓	2	AC_BK	Background
	0	AC_BE	Best Effort
	3	AC_BE	Best Effort
	4	AC_VI	Video
	5	AC_VI	Video
	6	AC_VO	Voice
alta	7	AC_VO	Voice

Tabella 1.VI: Mappatura tra AC e TC

Le stazioni che supportano la QoS verranno chiamate QSTA (*QoS Stations*) o anche ESTA (*Enhanced Stations*), per distinguerle dai nodi che non supportano la QoS (chiamati semplicemente STA), che possono comunque continuare ad essere presenti e a funzionare nella stessa QBSS (*QoS BSS*). Al posto dell' AP, in una QBSS vi è il QAP (*QoS Access Point*) che, oltre alle funzioni tipiche dell' AP, è dotato di nuove funzioni introdotte per garantire la QoS.

L'HCCA è un metodo di accesso che dispone di una funzione di coordinamento di tipo centralizzato, così come il PCF, ma con significative migliorie. Scompare il meccanismo della semplice polling list, che lascia spazio ad algoritmi capaci di rendere più intelligente il processo delle interrogazioni delle stazioni. Un altro importante concetto introdotto dal draft è quello del TXOP (*Transmission Opportunity*), inteso come un intervallo di tempo all'interno del quale una stazione ha la possibilità di trasmettere più di un frame. A seconda del metodo di accesso utilizzato si parla di *EDCA TXOP* o di *HCCA TXOP*, detto anche *polled TXOP* perché assegnato dall'HC mediante l' invio di un frame QoS CF Poll.

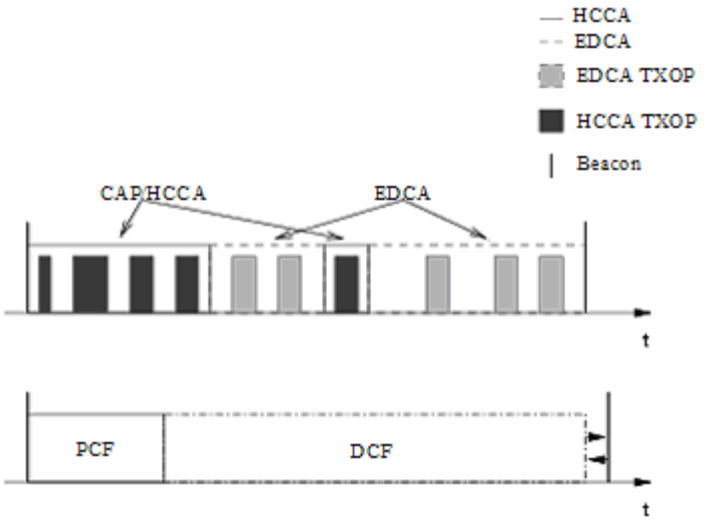


Figura 1.17: Alternanza tra CAP e CP in un Superframe

Vengono anche introdotte due nuove politiche per la gestione degli ACK:

- *No Acknowledgement*, in cui la ricezione di un frame semplicemente non viene confermata;
- *Group Acknowledgement*, in cui più frame vengono confermati con un unico ACK.

Per tener traccia della classe di appartenenza di un frame, l' 802.11e ha dovuto modificare la struttura generica del MAC frame, schematizzata in figura 1.18 [12]:

Octets: 2	2	6	6	6	2	6	2	~~0-2312~~n	4
Frame Control	Duration / ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	QoS Control	Frame Body	FCS
MAC Header									

Figura 1.18: Struttura di un MAC frame 802.11e

è stato aggiunto il campo *QoS Control*, in cui vengono specificati alcuni importanti parametri per la gestione della QoS. Si tratta di un campo a 16 bit che identifica la TC alla quale il frame appartiene, e fornisce ulteriori informazioni che variano a seconda del

particolare tipo di frame. Esso è suddiviso in cinque sottocampi, di cui i tre che più interessano in questo lavoro di tesi sono indicati, insieme alla loro funzione, in tabella 1.VII [12]:

Bit 0-3	**Bit 5-6**	**Bit 8-15**	**Tipo di frame**
TID	Politica degli ACK	TXOP in unità di 32 microsecondi	*Nei frame QoS Data, incluso i QoS Poll, inviati dall'HC*
TID	Politica degli ACK	Queue Size in unità di 256 ottetti	*Nei frame QoS Data inviati dalle QSTA e diretti all'HC*

Tabella 1.VII: Principali sottocampi del QoS Control Field

- Il sottocampo TID (*Traffic Identifier*) specifica il livello di priorità (TC) del flusso cui appartiene un MSDU; ogni TC mette in relazione funzionale la particolare coda con una delle quattro AC previste.
- Il sottocampo a due bit specifica la politica degli Ack da usare per il trasporto del corrente MPDU.
- Il sottocampo *Queue Size* fornisce un metodo per permettere all' HC di conoscere il livello di coda della particolare TC a cui il campo TID fa riferimento, per ciascun nodo di trasmissione; questo feedback permette all' HC nel funzionamento in modalità HCF di effettuare interrogazioni intelligenti, basate su una certa conoscenza della situazione globale del traffico ancora da trasmettere nella QBSS; da notare che il Queue Size fornisce una versione granulare (in unità da 256 byte) e limitata (essendo un campo di 8 bit può fornire valori fra 0 e 255) del livello di coda. In particolare il valore 0 indica coda vuota, il valore 255 indica uno stato imprecisato, il valore 254 indica un livello di coda superiore ai 64768 ottetti, tutti i

restanti valori intermedi sono usati per indicare livelli di coda compresi fra 1 e 64768 ottetti. Nello stesso sottocampo viene invece indicata la durata di un polled TXOP, se il frame è un QoS Poll inviato dall' HC ad una QSTA; la durata viene espressa in unità di 32 microsecondi, per cui può assumere valori compresi nell'intervallo [0, 8160] microsecondi.

1.6.2 EDCA (Enhanced Distributed Channel Access)

L'EDCA [12] (anche indicata come HCF *Contention-base Access*) è basato su un accesso al mezzo di tipo contention-based. Come già detto, la novità fondamentale introdotta dall'EDCA è proprio la possibilità di separare il traffico in uscita da ciascun nodo in diverse categorie di traffico, in modo da poter trattare differentemente e con maggior riguardo i flussi con più esigenze, in particolare, sono state definite quattro AC, che suddividono tutto il traffico in quattro classi con diverse priorità. L' EDCA può idealmente essere visto come una sorta di DCF, in cui la probabilità di vincere la contesa per il possesso del mezzo non è uguale per tutti i flussi, ma è più alta per i flussi con AC di priorità maggiore. A tal fine il DIFS viene adesso sostituito dagli AIFS[i] (*Arbitration IFS della classe i-esima*) che dipende dall' AC secondo la seguente relazione:

$$AIFS[AC] = SIFS + AIFSN[AC] \cdot aSlotTime$$

dove AIFSN[AC] è un numero intero dipendente dal livello di priorità dell' AC cui si riferisce e può essere maggiore o uguale a 2 per le QSTA, mentre per il QAP vale 1, per garantirgli la precedenza nell' accesso al canale. Viene così garantito ai frame appartenenti ad AC a priorità più alta un minore tempo di attesa prima di iniziare la procedura di backoff.

Anche l'intervallo [CWmin,CWmax] in cui varia il parametro CW dipende dalla classe in considerazione e quindi si parla di CW[AC]. I valori di CWmin[AC] e CWmax[AC] si ricavano dal sottocampo ECWmin/ECWmax (relativo ad ogni AC), contenuto nell' elemento *EDCA Parameter Set* figura 1.19 presente nei frame Beacon, secondo le seguenti definizioni [12]:

$$CWmin = 2^{ECWmin} - 1 \qquad CWmax = 2^{ECWmax} - 1$$

Il set di parametri di ciascuna AC viene riportato nel rispettivo campo *Parameters Record*, il cui formato è mostrato in figura 1.20 [12].

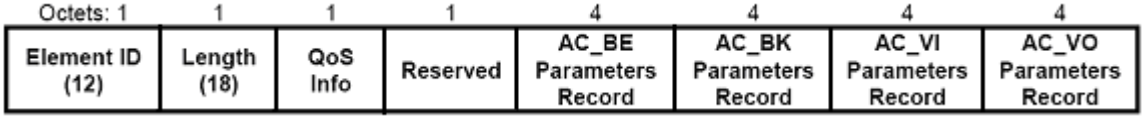


Figura 1.19: Formato dell'elemento EDCA Parameter Set

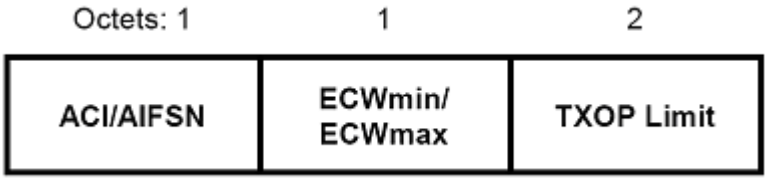


Figura 1.20: Formato del campo Parameter Record

Nella tabella 1.VIII sono indicati i valori di default proposti dal draft per i principali parametri [12], dove *aCWmin* e *aCWmax* come anche *TXOP Limit* (che esprime la durata massima del TXOP di cui può usufruire un determinato flusso), sono definiti in funzione del livello PHY utilizzato. Un valore nullo del parametro TXOP Limit indica che può essere trasmesso un solo MSDU con il relativo overhead (compresi anche gli eventuali frame RTS e CTS).

AC	CWmin	CWmax	AIFSN	TXOP Limit		
				DS-CCK[8]	Extended Rate /OFDM[9]	Other PHYs
AC_BK	aCWmin	aCWmax	7	0	0	0
AC_BE	aCWmin	aCWmax	3	0	0	0
AC_VI	(aCWmin+1)/2-1	aCWmin	2	6.016ms	3.008ms	0
AC_VO	(aCWmin+1)/4-1	(aCWmin+1)/2-1	2	3.264ms	1.504ms	0

Tabella 1.VIII: Principali parametri per ogni AC che regolano l' accesso al canale in EDCA

L'accesso al canale per ciascun frame avviene nel modo seguente: prima viene determinata l'AC di appartenenza del frame, poi si attende che il canale sia libero, si attende un ulteriore tempo AIFSD[AC] piu un *aSlotTime* (in sostituzione del DIFS nel DCF), e poi inizia la contesa dopo aver estratto un valore di backoff casuale uniforme nell'intervallo [0, CW[AC]], allo stesso modo che nel DCF. Nel caso di assenza di riscontro dopo la trasmissione, prima di estrarre il nuovo valore del backoff timer per effettuare la ritrasmissione, viene calcolato la nuova CW[AC] secondo la seguente formula:

$$CW[AC] = (CW[AC] + 1) \cdot 2 - 1$$

In generale i valori sono modellati in modo che la probabilità di vincita della contesa e quindi di trasmissione sia di volta in volta più elevata per frame appartenenti ad AC di priorità più alte, questo è ottenuto facendo in modo che frame a più elevata priorità attendano in media un tempo inferiore prima di vincere una contesa e quindi di trasmettere.
All' interno di ogni QSTA vengono implementate, come già detto, code di attesa differenti per ciascun flusso di dati che si contenderanno l' accesso al canale come se fossero stazioni separate (*stazioni virtuali*). Ovviamente i flussi a priorità più alta avranno una sorta di

precedenza; tuttavia potrebbe verificarsi il caso in cui due frame appartenenti ad AC distinte all' interno della stessa QSTA vincano la contesa nel medesimo istante. Questa situazione è indicata con il nome di *collisione virtuale*, perché si tratta di una collisione che viene risolta internamente alla stessa QSTA semplicemente trasmettendo il frame a priorità maggiore: infatti il livello MAC è in grado di rilevare l' evento prima della trasmissione fisica sul canale.

Quando una stazione guadagna il possesso del mezzo, il diritto di trasmissione perdura per un certo tempo indicato come *EDCA TXOP*, che è un parametro della QBSS annunciato periodicamente nei beacon; in tale tempo la stazione può trasmettere tanti frame quanti ne rientrano, a patto che siano appartenenti alla stessa AC e che la trasmissione dell' ultimo frame (compreso l' overhead) non sconfini al di fuori del TXOP limit, il cui valore, viene fornito periodicamente dai beacon (nel DCF invece era consentito trasmettere un unico frame). La QSTA setta il campo Duration Field del primo frame dati inviato con la durata prevista per il TXOP, in modo che le altre stazioni in ascolto possano aggiornare il proprio NAV. Per i frame dati successivi al primo, il campo Duration Field specificherà la durata residua del TXOP. Uno dei problemi risolti con l' introduzione dei TXOP è quello del ritardo nell' invio dei beacon: poiché è noto a tutte le QSTA l' istante di arrivo del prossimo beacon (indicato come TBTT), il nodo che guadagna il TXOP controlla che esso non si sovrapponga a tale istante e, in caso affermativo, blocca preventivamente la trasmissione.

1.6.3 HCCA (Hybrid Coordination Function Controlled Channel Access)

Nell'HCCA [12] (anche indicata come HCF *Contention-free Access*), l' accesso al mezzo è controllato dall' HC (*Hybrid Coordinator*), un' entità centrale che, partendo dalla conoscenza dei livelli di coda presenti nelle diverse QSTA della QBSS e delle esigenze di

ciascun traffico, decide di volta in volta chi ha diritto a trasmettere, in maniera simile al PCF. La necessità dell' HCCA è dettata dal fatto che l'EDCA, sebbene distingua fra le diverse esigenze dei traffici tramite l'introduzione delle AC, rimane pur sempre un metodo di accesso probabilistico che non dà garanzia sul rispetto del target delay. Ogni *SuperFrame* inizia con un beacon che, per questioni di compatibilità, può essere seguito da un CFP, per un accesso al canale di tipo PCF, seguito dal CP in cui si usa l' (E)DCA [2][12]. Dopo che il canale è rimasto libero per un PIFS, l'HC può dare inizio ad una fase chiamata CAP (*Controlled Access Phase*) (in generale è un multiplo del TBTT); all'interno di essa l' HC assegnerà dei *polled TXOP* alle stazioni mediante l' invio di particolari frame chiamati *QoS CF-Poll* [12], in modo che solo la stazione che avrà ricevuto il QoS CF-Poll avrà il diritto di trasmettere. Il numero e la posizione dei CAP all'interno di uno stesso Superframe viene deciso dall'HC a seconda delle esigenze delle stazioni ad esso associate; tuttavia l' estensione di un CAP non può superare uno specifico valore indicato dal parametro *dot11CAPLimit* [12].

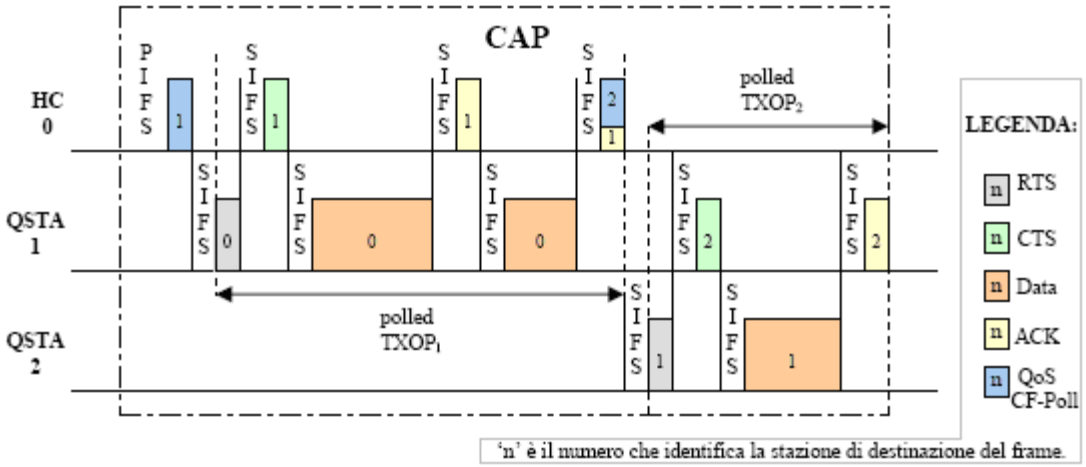


Figura 1.21: Esempio di trasmissione di frame dati all'interno di un CAP

L' assegnamento dei polled TXOP all' interno di un CAP avviene in modo tale che

vengano rispettati i requisiti di ritardo, banda e jitter richiesti dalla particolare stazione. Inoltre, l' HC potrebbe assegnare più polled TXOP consecutivi ad una stessa stazione, al fine di garantire alle varie AC la possibilità di inoltrare il proprio traffico: l' intervallo temporale dato dalla somma di questa serie di TXOP, più l' eventuale tempo impiegato dall' HC per inviare dati in downlink, prende il nome di SP (*Service Period*). Tale SP è dunque un lasso di tempo continuo, che si ripete ogni SI (*Service Interval*).

In seguito ad un QoS CFPoll, attraverso il Duration Field settato a TXOP limit più un SIFS, i NAV di tutte le stazioni vengono impostati in modo da proteggere (col virtual carrier sensing) il TXOP per tutta la sua durata. All' interno di un CAP tutti i frame saranno separati da un SIFS e, come nell' EDCA TXOP, anche nel polled TXOP ogni pacchetto di dati dovrà essere confermato da un ACK. Durante il CAP le QSTA e l' HC comunicano tra loro tramite lo scambio di frame *QoS CFPoll*, *QoS data*, *QoS CFAck* e loro combinazioni *QoS Data + CFAck*, *QoS Data + CFPoll*, *QoS Data + CFAck + CFPoll*, *QoS CFAck + CFPoll*. Qualora una QSTA non riceva un riscontro dall'HC in seguito all' invio di un frame dati, essa può tentare la ritrasmissione subito dopo un PIFS [12], sempre rispettando la durata del TXOP prevista, pertanto in un CAP il massimo tempo di interframe è dato da un PIFS, proprio come in un CFP.

Potrebbe capitare che una stazione non abbia nulla da trasmettere o che i frame che ha da trasmettere non "entrino" nel TXOP assegnatole, in questo caso la stazione risponderà al *QoS CF-Poll* con un frame *QoS-Null*; lo scopo di questo frame è duplice: il primo è quello di liberare il canale resettando i NAV di tutte le stazioni, il secondo è quello di aggiornare i livelli di coda memorizzati nell'HC, il quale si servirà di questa informazione per stabilire l' entità dei TXOP da attribuire alle QSTA. L'informazine sui liveli di coda presenti nella stazione è memorizzata in opportuni bit del *QoS Field* viene usata dall'HC per stabilire l'entità dei TXOP da assegnare ad una stazione. Quando una stazione entrerà a far parte di una QBSS, essa comunicherà all'HC i parame tri necessari affinchè venga garantita la QoS

per un particolare flusso di dati; ciò avviene attraverso l'uso di un particolare tipo di frame chiamato *TSPEC* (*Traffic SPECification*).

La TSPEC di una QSTA è suddivisa in diversi record (uno per flusso), ciascuno dei quali è costituito da una serie di campi; i più importanti sono [12]:

- *Nominal MSDU Size, Maximum MSDU Size*, che indicano rispettivamente la grandezza nominale (media) e massima espressa in ottetti dell' MSDU generato dal flusso cui si riferisce il record;
- *Minimum Service Interval, Maximum Service Interval*, che specificano rispettivamente il minimo e massimo intervallo di tempo (espresso in microsecondi) richiesto dal particolare flusso fra due successivi TXOP;
- *Minimum Data Rate, Mean Data Rate, Peak Data Rate, Maximum Burst Size*, che indicano rispettivamente i data rate minimo, medio e massimo (espressi in bps) che caratterizzano la specifica sorgente di MSDU (non viene quindi considerato l'overhead dovuto ai livelli MAC e PHY), mentre l' ultimo parametro esprime la grandezza massima in ottetti di un burst di MSDU;
- *Minimum PHY Rate*, che specifica la minima velocità fisica (espressa in bps) richiesta dal flusso considerato per il trasporto degli MSDU;
- *Delay Bound*, che è sicuramente il parametro più importante per la gestione della QoS, in quanto specifica il massimo ritardo (espresso in microsecondi) tollerabile nel trasporto di un MSDU.

Non è necessario che una QSTA, in fase di negoziazione, comunichi all' HC tutti i valori dei parametri di TSPEC, anche perché essi dipendono dal livello applicativo e non è detto che siano noti preventivamente.

E' previsto anche che l' HC possa rifiutare nuovi flussi di dati che hanno parametri di TSPEC troppo stringenti per essere rispettati nelle condizioni attuali della rete, questo meccanismo prende il nome di *CAC* (*Call Admission Control*) e serve proprio ad assicurare il rispetto dei vincoli di QoS ai flussi già ammessi.

Anche in HCCA viene risolto il problema del ritardo dei beacon: sarà l'HC stesso ad assegnare TXOP tali che non si possano sovrapporre ai beacon.

1.6.4 Algoritmi per l'allocazione della banda in HCCA

Lo standard IEEE 802.11e non specifica come schedulare i TXOP affinché sia possibile garantire il rispetto dei vincoli di QoS alle varie stazioni; il draft 8.0 si limita semplicemente a standardizzare e descrivere l' uso di questi meccanismi, lasciando però libera autonomia di implementazione hardware per quanto riguarda l' algoritmo da usare per la schedulazione dei polled TXOP. . In sostanza il draft propone solo un algoritmo molto semplice per la schedulazione dei TXOP, chiamato *Simple Scheduler* [12][15]: partendo dai valori specificati in ogni TSPEC, il calcolo viene fatto una sola volta e, quindi, non tiene conto dell'evoluzione dei traffici all'interno della QBSS. È evidente, quindi, come ciò possa portare ben presto ad un inutile spreco di banda.
Per ovviare a questo inconveniente, è stato proposto due nuovi algoritmi chiamati *Feedback Based Dynamic Scheduler FBDS* e *Proportional Integral (PI)-FBDS* [16][17][18] per l'assegnamento dei TXOP che si basa sulla teoria dei controlli automatici e sfrutta l'informazione sui livelli di coda presenti nelle stazioni per avere una gestione più intelligente nell'allocazione della banda disponibile.

1.6.4.1 Simple Sheduler

Come già detto il draft 8.0 si limita a suggerire questo semplice algoritmo, dove, partendo dai valori dei parametri specificati nelle TSPEC, il Simple Scheduler calcola in modo

statico la dimensione dei TXOP da assegnare a ciascun traffico che richiede la QoS; inoltre, viene calcolato anche il tempo che deve intercorrere fra due CAP adiacenti, definito come *SI* (*Scheduled Service Interval*).

I calcoli vengono eseguiti solo una volta e i risultati sono utilizzati staticamente per l' assegnazione dei TXOP in tutti i Superframe successivi. Fissato il numero di flussi QoS e i relativi parametri di TSPEC, noto anche la durata del Superframe (*Tsf*), viene calcolato dapprima SI come il più grande sottomultiplo del periodo di ripetizione dei beacon (TBTT) che sia contemporaneamente inferiore al minimo fra tutti i *Maximum Service Interval* (o in alternativa fra tutti i *Delay Bound*) dei flussi ammessi: in tal modo il servizio di un flusso si ripresenterà con una frequenza maggiore di quella minima richiesta dalle specifiche del flusso stesso. Poi, per l'*i*-esimo flusso, viene calcolato il relativo $TXOP_i$ in due passi:

$$N_i = \left\lceil \frac{S_i \cdot \rho_i}{L_i} \right\rceil \qquad (1)$$

dove ρ_i ed L_i sono rispettivamente il *Mean Data Rate* ed il *Nominal MSDU Size* del flusso *i*-esimo, ed il valore calcolato N_i rappresenta l'approssimazione intera per eccesso del numero di MSDU generati mediamente dalla sorgente *i*-esima durante l' intervallo *SI*; dunque il polled TXOP da assegnare sarà pari a:

$$TXOP_i = \max\left(\frac{N_i \cdot L_i}{R_i} + O, \frac{M}{R_i} + O \right) \qquad (2)$$

dove R_i è il minimo rate fisico di trasmissione ed *O* è l' overhead, cioè il tempo aggiuntivo necessario alla trasmissione completa degli N_i pacchetti (quindi include i SIFS di interframe, il tempo di trasmissione degli ACK ed il tempo di trasmissione dei campi aggiunti dai livelli MAC e PHY); *M* è invece la dimensione massima per un MSDU prevista dallo standard (2304 byte in assenza di crittografia).

Una volta effettuati questi calcoli, il Simple Scheduler si limita a creare il CAP ogni *SI* e a fare il polling all' interno di ciascun CAP assegnando ai vari flussi dei polled TXOP di dimensione fissa. Il principale difetto di questo algoritmo sta nel fatto che, allocando staticamente la banda disponibile, non tiene conto dell' evoluzione dei traffici all' interno della QBSS. Esso infatti considera i flussi come se fossero originati da sorgenti statiche e tempo invarianti, quindi effettua i calcoli necessari a dimensionamento dei TXOP sufficienti a svuotare le code di dati generati da queste ipotetiche sorgenti di tipo CBR (*Constant Bit Rate*). La realtà è molto diversa perché i traffici real-time sono estremamente variabili, per cui se in media è assicurato il rispetto del target delay, questo non è detto che avvenga sempre ed in maniera deterministica.

1.6.4.2 FBDS e PI-FBDS

Una soluzione più sofisticata, che tenga conto della natura VBR (*Variable Bit Rate*) dei traffici real-time, è sicuramente quella di sfruttare l' informazione sui livelli di coda presenti nel campo queue size in modo da soddisfare le reali necessità di banda dei flussi ammessi, allocando dinamicamente i TXOP, il che è quanto si propone di fare l' algoritmo descritto in questo lavoro di tesi, che è il risultato dell' attività di ricerca all' interno del Dipartimento di Elettrotecnica ed Elettronica del Politecnico di Bari.
Come già detto, questo algoritmo è basato sulla teoria dei controlli automatici per sistemi tempo discreti, per cui esso deve basarsi su un preciso modello di QBSS: il modello di QBSS utilizzato da FBDS è composto da un QAP (HC) e un certo numero di QSTA ad esso associate, per ognuna delle quali sono previste fino a quattro code in cui confluiscono gli MSDU in attesa di essere trasmessi, suddivisi in base all' AC di appartenenza. Viene

ipotizzato che il periodo di ripetizione dei CAP (T_{CA}) sia costante. All'interno di ogni CAP, l' HC concede alle QSTA, mediante l' invio di frame QoS CFPoll, TXOP di dimensione ritenuta sufficiente a svuotare le proprie code, basandosi sui livelli di coda precedentemente aggiornati; nel campo QoS Control l' HC specifica, oltre alla QSTA, anche la TC destinataria del TXOP.

Si assume che, all' inizio di ogni CAP, l' HC abbia a disposizione i livelli di coda aggiornati all' inizio del CAP precedente. In questo modo viene considerato il caso peggiore in cui una stazione, non avendo trasmesso alcun frame nell' ultimo CAP, non ha permesso all' HC di aggiornare i relativi livelli di coda.

Definiamo adesso le variabili che descrivono il nostro modello:

- $q_i(k) \geq 0$ il livello della coda i-esima all' inizio del k-esimo CAP, con $i = 1 \ldots M$, dove M rappresenta il numero totale di flussi QoS;
- $d_i(k) = d_i^S(k) - d_i^{CP}(k)$ la differenza tra $d_i^S(k) \geq 0$, che è il rate medio di generazione (e quindi di accodamento) del flusso i-esimo durante il k-esimo CAP, e $d_i^{CP}(k) \geq 0$, che è la quantità di dati drenati dalla coda i-esima durante il k-esimo CP in modalità EDCA, rispetto al periodo T_{CA};
- $TXOP_i(k)$ il polled TXOP assegnato nel CAP k-esimo per il drenaggio dell'i-esima coda;
- $u_i(k) \leq 0$ la banda assegnata nel k-esimo CAP per il drenaggio dell' i-esima coda.

La dinamica della generica coda è descritta dal seguente modello:

$$q_i(k+1) = q_i(k) + d_i(k) \cdot T_{CA} + u_i(k) \cdot T_{CA} \qquad i = 1 \ldots M \qquad (3)$$

Questa equazione indica che il livello di coda nel (k+1)-esimo CAP è dato da quello nel k-esimo CAP, incrementato dei dati generati dalla sorgente (e quindi accumulati in coda) durante il k-esimo T_{CA}, e decrementato dei dati smaltiti per mezzo della banda (o TXOP) assegnata nel k-esimo CAP e in modalità EDCA tra il k-esimo e il (k+1)-esimo CAP.

La variabile $d_i(k)$ ha un comportamento imprevedibile, quindi dal punto di vista della teoria dei controlli automatici è assimilabile ad un disturbo. Come tale, esso può essere modellato come la somma di infinite funzioni a gradino tempo discrete di diversa ampiezza $d_{0j} \in \Re$, traslate opportunamente nel tempo di una quantità t_j, secondo la seguente formula [16]:

$$d_i(k) = \sum_{j=0}^{+\infty} d_{0j} \cdot 1(k - t_j) \tag{4}$$

dove 1(k) è la funzione gradino unitario.

L' obiettivo che si pone FBDS è quello di controllare il ritardo di accodamento τ_i di ciascun flusso, cercando di non andare oltre il ritardo obiettivo τ_i^T (*target delay*), che rappresenta il vincolo da rispettare affinché venga soddisfatta la QoS per il flusso *i*-esimo. Poiché tali ritardi sono direttamente correlati ai livelli di coda, si studia un sistema che controlli il generico livello di coda *i*-esimo q_i e riesca a renderlo inferiore al livello di coda obiettivo q_i^T. Pertanto si pone $q_i(k)$ la variabile di stato che vogliamo controllare, $u_i(k)$ la variabile d' attuazione e $d_i(k)$ il disturbo da reiettare. Per la linearità dell' equazione 3 si può studiare il sistema considerando un disturbo pari a $d_i(k) = d_0 \cdot 1(k)$.

Adesso vedremo due algoritmi di tipo *feedback-based* per l'assegnamento dinamico della banda in HCCA.

Gli algoritmi proposti, che sono stati chiamati *Feedback Based Dynamic Scheduler FBDS* e *Proportional Integral (PI)-FBDS* [19], sono stati ideati usando la teoria dei controlli automatici retroazionati tempo-discreti.

Il sistema di controllo alla base dell' implementazione dello scheduler dinamico è riportato in figura 1.22 [17]:

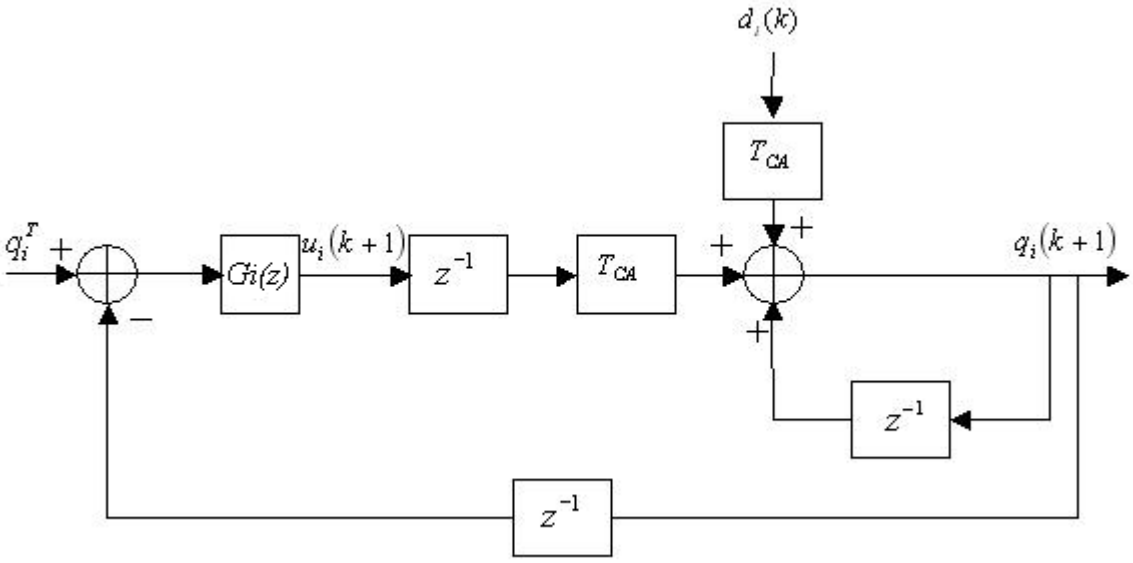


Figura 1.22: Schema di controllo dell'algoritmo proposto

dove il valore di riferimento q_i^T è fissato a zero, in modo da ottenere idealmente code vuote. Osservando la funzione di trasferimento del controllore Gi(z), focalizzeremo l'attenzione su due tipi di controllore: un controllore proporzionale (P), ottenuto ponendo $G_i(z)=Kp$ e un controllore proporzionale-integrale (PI), ottenuto con $G_i(z) = Kp\left(1 + \frac{z}{z+1}\cdot\frac{1}{T_I}\right)$; i rispettivi algoritmi di allocazione della banda saranno chiamati FBDS e PI-FBDS.

Considerando lo schema con $G_i(z)=Kp$, è semplice calcolare la trasformata-Z di $q_i(k)$ e $u_i(k)$:

$$Q_I(z) = \frac{zT_{CA}}{z^2 - z + KpT_{CA}}\cdot D_i(z); \qquad U_i(z) = -\frac{KpT_{CA}}{z^2 - z + KpT_{CA}}\cdot D_i(z) \tag{5}$$

con $D_i(z)=Z[d_i(k)]$.

Dall' equazione, i poli del sistema sono:

$$z_p = \frac{1 \pm \sqrt{1 - 4Kp \cdot T_{CA}}}{2}$$

l' asintotica stabilità è assicurata se e solo se $|z_p| < 1$, cioè se la costante di guadagno k_i soddisfa la condizione:

$$0 < k_i < \frac{1}{T_{CA}}$$

Per verificare il comportamento dello stato stabile del sistema, applichiamo il *Teorema del valore finale* all' Eq.(5). Considerando che la trasformata-Z della funzione a gradino $d_i(k) = d_0 \cdot 1(k)$ è $D_i(z) = d_0 \cdot \frac{z}{z-1}$, otteniamo i seguenti risultati:

$$u_i(+\infty) = \lim_{k \to \infty} u_i(k) = \lim_{z \to 1}(z-1)U_i(z) = -d_0 \; ; \qquad q_i(+\infty) = \frac{d_0}{Kp} \tag{6}$$

che implica che il ritardo di accodamento è:

$$\tau_i(+\infty) = \left|\frac{q_i(+\infty)}{u_i(+\infty)}\right| = \frac{1}{Kp} \tag{7}$$

questo ci fa notare che $q(+\infty)>0$, sebbene $q_i^T = 0$, che significa che il controllore proporzionale non riesce a reiettare completamente l'effetto del disturbo a gradino $d_0 \cdot 1(k)$. Inoltre dall' Eq.(7) osserviamo che deve essere rispettata la seguente disuguaglianza per ottenere un ritardo inferiore a τ_i^T:

$$Kp \geq \frac{1}{\tau_i^T}$$

considerando le condizioni (6) e (8), si deduce che il parametro T_{CA} deve soddisfare la seguente disequazione:

$$T_{CA} \leq \min_{i=1,\ldots,M} \left\{ \tau_i^T \right\} \tag{9}$$

Da questi risultati possiamo concludere che, quando usiamo un controllore proporzionale il guadagno Kp può variare all'interno di un range $\left[\frac{1}{\tau_i^T}, \frac{1}{T_{CA}} \right[$. Fisseremo $Kp = \frac{1}{\tau_i^T}$, al minimo valore ammissibile. Questa scelta permette, per un riscontro del livello di coda q_i(k), il minimo assegnamento di banda che soddisfi i bounded delays, quindi un più cauto uso della banda del mezzo WLAN.

Similarmente al caso del controllore proporzionale, considerando la fig.1.22 dove $G_i(z) = Kp\left(1 + \frac{z}{z+1} \cdot \frac{1}{T_I}\right)$ e $q_i^T = 0$, dopo un semplice calcolo algebrico possiamo ottenere la seguente trasformata-Z di $q_i(k)$ e $u_i(k)$:

$$Q_i(z) = \frac{T_{CA}T_I z(z-1)}{T_I z^3 - 2T_I z^2 + (T_I + T_{CA}KpT_I + T_{CA}Kp)z - T_{CA}KpT_I} \cdot D_i(z) \tag{10}$$

$$U_i(z) = \frac{T_{CA}KpT_I - T_{CA}KpT_I z - T_{CA}Kpz}{T_I z^3 - 2T_I z^2 + (T_I + T_{CA}KpT_I + T_{CA}Kp)z - T_{CA}KpT_I} \cdot D_i(z) \tag{11}$$

Applicando il criterio di Jury, il sistema in fig.1.22 è asintoticamente stabile se soddisfa le seguenti disequazioni:

$$Kp < \frac{1}{T_{CA}}; \qquad T_I > \frac{1}{(1 - T_{CA}Kp)}.$$

Anche in questo caso, per controllare il comportamento dello stato stabile del sistema, applichiamo il *Teorema del valore finale* all' Eq.(10), ottenendo così $q_i(+\infty) = 0$, il che implica che il ritardo di accodamento dello stato stabile è nullo.

Questo risultato è dovuto al comportamento integrale del controllore, che è in grado di reiettare il disturbo a gradino. In questo modo, i parametri del regolatore PI, sono soggetti solo alle restrizioni di stabilità. Di conseguenza abbiamo molti gradi di libertà nella scelta di Kp e T_I, rispetto al caso del controllore proporzionale.

Quando usiamo il controllore PI, potrebbe accadere che il depletion rate dovuto a $|u_i(k+1)|$ sia più grande di $\frac{q_i(k)}{T_{CA}}$, che è l'ammontare della banda richiesto per svuotare completamente l' i-esima coda durante il (k+1)-esimo CAP; questo assegnamento ovviamente significherebbe uno spreco di risorse. Per superare questo inconveniente, useremo il seguente espediente:

$$u_i(k+1) \leftarrow \max_{i=1,\ldots,M}\left\{u_i(k+1), -\frac{q_i(k)}{T_{CA}}\right\} \qquad (13)$$

dove, apprendendo dalle TSPEC che $u_i \leq 0$, il termine $\frac{-q_i(k)}{T_{CA}}$ è il rate necessario a svuotare interamente la coda. L' obiettivo dell' Eq.(13) è correggere l'allocazione della banda, che verrebbe fatta dal regolatore PI, non permettendo un assegnamento maggiore di quello necessario a svuotare completamente l' i-esima coda.

All' inizio del CAP, ogni T_{CA} secondi, l' HC calcola i TXOP che bisogna assegnare ai flussi per garantire il rispetto dei vincoli di QoS. Se C_i è il data rate al quale viene drenata la coda

i-esima, lo scheduler usa la seguente formula per passare dalla banda assegnata nel *k*-esimo CAP $u_i(k)$ al valore di $TXOP_i(k)$:

$$TXOP_i(k) = \frac{|u_i(k) \cdot T_{CA}|}{C_i} + O \tag{14}$$

in cui *O* rappresenta l' overhead dovuto essenzialmente ai tempi di trasmissione dei campi aggiunti dai livelli MAC e PHY e degli ACK e ai tempi di interframe.

Con l' aumentare del numero di flussi QoS, le limitate risorse del canale devono venire condivise da un numero crescente di traffici; compito dell'HC è di verificare che la somma dei polled TXOP allocati per ciascun traffico sia inferiore o al più pari al limite previsto per il CAP, specificato nel parametro *dot11CAPLimit*. Potrebbe comunque capitare che la somma dei TXOP allocati dallo Scheduler in un determinato CAP sia superiore al limite consentito. Quando si verifica questa situazione, nota come *saturazione del canale*, bisogna ridimensionare i TXOP calcolati in modo da ritornare al di sotto del limite di saturazione; la condizione di saturazione è data dalla seguente disequazione:

$$\sum_{i=1}^{M} TXOPO_i(k) > dot11CAPLimit$$

dove $TXOPO_i$ comprende anche gli overhead, ossia il PIFS, il tempo di trasmissione di un QoS CFPoll e il SIFS che intercorre tra due TXOP consecutivi.

Nel caso sia verificata la condizione di saturazione, lo schedulatore applicherà le seguenti formule per il ridimensionamento dei TXOP:

$$\Delta = \sum_{i=1}^{M} TXOPO_i(k) - dot11CAPLimit$$

$$\Delta_i = \frac{TXOPO_i(k) \cdot C_i}{\sum_{i=1}^{M} \left(TXOPO_i(k) \cdot C_i\right)} \cdot \Delta$$

$$TXOPO_i(k) = TXOPO_i(k) - \Delta_i$$

in cui Δ indica il tempo totale in eccesso rispetto al valore limite, quindi rappresenta la quantità di risorse da ridistribuire; Δ viene suddiviso nei tempi Δ_i da sottrarre a ciascun flusso; la formula di suddivisione è tale da sottrarre tempo al flusso i-esimo in maniera proporzionale alle risorse inizialmente allocate a tale flusso e alla velocità fisica della stazione destinataria del TXOP, al fine di non penalizzare eccessivamente i traffici a basso rate di trasmissione.

In generale, per il generico flusso i-esimo la dimensione di un MSDU sarà variabile in maniera non prevedibile dall' HC, che però conosce dalle TSPEC il valore medio *Nominal MSDU Size* e massimo *Maximum MSDU Size*. Potrebbe capitare il caso in cui in coda ci sia un solo pacchetto, proprio della dimensione massima; l' algoritmo potrebbe allocare invece una banda sufficiente alla trasmissione di un pacchetto di dimensione nominale (proprio perché si usa tale parametro per dimensionare il TXOP); ovviamente la QSTA non potrà trasmettere il data frame, ma semplicemente un *QoS Null* ; questa situazione porta all'accumulo di un ritardo pari ad un T_{CA} per il pacchetto in questione (trascurando l'eventualità che venga trasmesso in EDCA durante il CP), ritardo che potrebbe ulteriormente aumentare di CAP in CAP se la situazione non viene sbloccata dall' arrivo in coda di nuovi pacchetti che portino lo Scheduler ad allocare maggior banda. Il feedback sul livello di coda continuerà comunque ad essere fornito dal campo QoS Control Field dei frame QoS Null. Per ovviare a questo problema si potrebbe pensare di dimensionare il TXOP facendo riferimento solo a frame di dimensione massima, ma ciò porterebbe ad un inutile spreco di risorse, in quanto verrebbero assegnati TXOP di durata maggiore di quella che effettivamente servirebbe allo smaltimento delle code. Allora la soluzione adottata consiste nel fare un controllo dopo il calcolo di ogni $TXOP_i(k)$ e nell' imporre che esso sia almeno sufficiente per la trasmissione di un pacchetto di massima dimensione.

Come abbiamo visto quindi, lo scopo di questo scheduler è quello di assegnare TXOP di durata tale da poter drenare tutti i pacchetti rimasti in coda; così facendo ovviamente, l'ultimo pacchetto trasmesso comunicherà all'HC una queue size nulla. A questo punto, nel successivo CAP lo Schedulatore non vedrà la necessità di allocare banda per tale flusso, anche se nel frattempo potrebbero essere arrivati nuovi pacchetti in coda. Si crea così una situazione di stallo o *deadlock* in cui i pacchetti si accumulano in coda e non vengono smaltiti per la mancanza di feedback sul livello di coda all' HC. In realtà la situazione è destinata a sbloccarsi prima o poi quando la stazione riuscirà ad inviare mediante EDCA un pacchetto, che naturalmente trasporterà anche il campo Queue Size con il livello aggiornato, in modo che al prossimo CAP la coda possa ricevere un TXOP adeguato. Ovviamente l' istante in cui tale pacchetto sarà trasmesso è imprecisato: dipenderà infatti dal momento in cui la nostra stazione riuscirà a vincere la contesa, il che è assolutamente probabilistico, inoltre nel frattempo i pacchetti possono aver accumulato ritardi superiori a quelli ammessi. Il problema del deadlock è stato risolto attraverso l' uso di un meccanismo di *pre-polling*, che consiste in una fase, collocata immediatamente prima del CAP, in cui l' HC invia dei *QoS Poll* a tutti quei flussi con livello di coda nullo, al fine di poter effettuare l' eventuale aggiornamento. Il TXOP assegnato in questa fase consentirà l'invio di un solo MSDU di lunghezza nominale; la stazione interrogata dovrà comunque rispondere o con un *QoS Null* o con un eventuale frame di dati, garantendo quindi all' HC in entrambi i casi l' aggiornamento del livello di coda: in tal modo, quando lo Scheduler andrà, subito dopo il pre-polling, ad effettuare il calcolo dei TXOP da allocare nella fase di *polling* vera e propria, disporrà sicuramente di informazioni il più possibile aggiornate.