

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Specialistica in Informatica

**Progetto e valutazione di
un protocollo MAC per
reti wireless ad-hoc**

Tesi di Laurea in Sistemi e Reti Wireless

Relatore:
Chiar.mo Prof.
Luciano Bononi

Presentata da:
Angelo Trotta

Correlatore:
Marco Di Felice

**Sessione III
Anno Accademico 2009-2010**

Indice

Introduzione	ix
1 IEEE 802.11e	1
1.1 Introduzione	1
1.2 802.11 legacy	1
1.2.1 Distributed Coordination Function	2
1.2.2 Point Coordination Function	3
1.3 802.11e EDCF con supporto al QoS	6
1.3.1 Introduzione	6
1.3.2 EDCA (Enhanced Distributed Channel Access)	7
1.4 Conclusioni	13
2 PAB	17
2.1 Introduzione	17
2.2 Funzionamento	24
2.2.1 Header pacchetti	24
2.2.2 Definizioni	25
2.2.3 Scheduler	26
2.2.4 Trasmissione dei pacchetti	27
2.2.5 Accesso al canale con priorità dinamica	31
2.2.6 Superframe	35
2.2.7 Flussi e sottoflussi	39
2.3 Considerazioni	40

3	Analisi dei due protocolli	43
3.1	Prestazioni generali	44
3.1.1	Scenario	44
3.1.2	Goodput	45
3.1.3	MAC delay link-to-link	45
3.2	Proprietà QoS	48
3.2.1	Scenario	48
3.2.2	Priorità bassa	49
3.2.3	Priorità medio-bassa	49
3.2.4	Priorità medio-alta	52
3.2.5	Considerazioni	53
3.3	Problema della starvation	54
3.3.1	Scenario	54
3.3.2	PAB	55
3.3.3	802.11e	57
3.4	Fairness	58
3.4.1	Scenario	58
3.4.2	Risultati	59
	Conclusioni	63
A	Panoramica sui modelli	67
A.1	Modello PAB	67
A.2	Modello 802.11e	69
	Bibliografia	71

Elenco delle figure

1.1	Il post-backoff viene effettuato dopo ogni trasmissione riuscita, anche se non ci sono MSDU da trasmettere	3
1.2	In questo esempio la stazione 6 non riceve l'RTS trasmesso da 2, ma il CTS trasmesso da 1	4
1.3	Esempio di una operazione PCF. La stazione 1 è il PC che invia un CF-Poll alla stazione 2. La stazione 3 riceve il beacon e setta il NAV per l'intero CFP. La stazione 4 non vede la stazione 1 e quindi continua ad operare con il DCF	5
1.4	Scambio di dati con il PCF. Il beacon viene trasmesso in ritardo. .	6
1.5	Quattro AC, ogniuna con la sua coda, AIFS, CW e backoff timer .	8
1.6	Meccanismo di priorità basato sul AIFS	10
1.7	Cambio della grandezza del CW con $PF=2$ per ogni AC	12
1.8	Cambio della grandezza del CW. In questo vengono usati differenti PF per i diversi AC	12
1.9	Contention Free Bursting (CFB)	13
1.10	Problematiche del servizio QoS in 802.11e	14
2.1	Meccanismo di contesa tramite BB	19
2.2	Schema di contesa attraverso l'uso di black burst e AIFS differenti	20
2.3	Esempio di accesso del protocollo QMA	21
2.4	Esempio di accesso del protocollo DAQ-MAC	22

2.5	Comportamento del protocollo BTPS. (a) mostra il comportamento di una stazione di alto livello che ha un pacchetto da inviare; (b) mostra il comportamento di chi ascolta il segnale inviato su BT1; (c) mostra una stazione che ascolta soltanto il BT2	23
2.6	Header di un frame PAB	24
2.7	Flow chart che mostra l'accesso al canale di una stazione	28
2.8	Caso in cui $PrIFS_1 < PrIFS_2$	30
2.9	Caso in cui $PrIFS_1 = PrIFS_2$ ma $burst_len1 > burst_len2$	30
2.10	Struttura di un superframe	35
2.11	Esempio di sincronizzazione in cui si nota come S2 al primo invio sincronizza S3, mentre al secondo invio sincronizza S1	38
3.1	Confronto delle prestazioni (Goodput) dei due protocolli al variare del numero delle stazioni. Il numero di flussi presenti per tutte le priorità è uguale	46
3.2	Confronto delle prestazioni (Mac Delay link-to-link) dei due protocolli al variare del numero delle stazioni. Il numero di flussi presenti per tutte le priorità è uguale	47
3.3	Confronto delle prestazioni (Mac Delay link-to-link) dei due protocolli.	48
3.4	Confronto delle prestazioni dei due protocolli al variare del numero delle stazioni aventi flussi di priorità 3	50
3.5	Confronto delle prestazioni dei due protocolli al variare del numero delle stazioni aventi flussi di priorità 2	51
3.6	Confronto delle prestazioni dei due protocolli al variare del numero delle stazioni aventi flussi di priorità 1	52
3.7	Goodput PAB nel caso di valore $\alpha=0.85$	53
3.8	Confronto delle prestazioni del PAB al variare del numero delle stazioni di livello superiore al 3	55
3.9	Confronto delle prestazioni del PAB al variare del numero delle stazioni di livello superiore al 3	57
3.10	Confronto della caratteristica di fairness dei due protocolli	61

A.1 Modello del protocollo PAB 68

Elenco delle tabelle

1.1	Mappatura da User Priority (UP) a Access Category (AC)	7
1.2	Valori di default per i parametri dell'EDCA	9
1.3	Valori del CW per differenti tipi di livello fisico	10
2.1	Esempio evoluzione dei flussi	40
3.1	Parametri utilizzati per le simulazioni	44
3.2	Parametri utilizzati per lo studio della QoS	49
3.3	Parametri utilizzati per lo studio della starvation	54
3.4	Valori del goodput PAB della stazione a priorità 3	56
3.5	Valori del goodput PAB della stazione a priorità 3 in presenza di un numero crescente di stazioni di priorità 1 e con un valore di $\alpha=0.85$	56
3.6	Valori del goodput 802.11e della stazione a priorità 3	58
3.7	Parametri utilizzati per lo studio della fairness	59
A.1	Parametri utilizzati per le simulazioni del modello PAB	69
A.2	Parametri utilizzati per le simulazioni del modello IEEE 802.11e	70
A.3	Parametri caratterizzanti le diverse priorità	70

Introduzione

Nell'ultimo decennio il mondo si è mosso nella direzione in cui il vincolo di una comunicazione ristretta a quella tramite cavi viene sciolto dalla possibilità, attraverso l'uso di tecnologie senza fili, di essere connessi ovunque ci si trovi. Il 2002 è una data storica per le comunicazioni wireless: il numero dei cellulari mobili superavano per la prima volta il numero dei telefoni fissi. Intanto la gente cominciava a muoversi e viaggiare avendo la possibilità di connettersi attraverso le wireless wide-area network (WWAN), mentre negli uffici i lavoratori "mobili" cominciavano ad accedere alla rete tramite le wireless local area network (WLAN). Contemporaneamente privati e aziende hanno cominciato ad installare sempre più frequentemente access point in ogni dove, così da permettere una connettività illimitata.

Un altro passo avanti verso l'ubiquità di connessione è lo sviluppo delle cosiddette reti ad-hoc (MANET - Mobile Ad-hoc NETWORK). A differenza delle reti statiche in cui è presente un access point che fornisce la connettività ai nodi che si collegano ad esso, una MANET è un sistema autonomo di router mobili e dei loro host associati, connessi con collegamenti di tipo wireless che sono uniti formando un grafo di forma arbitraria. Tali router sono liberi di muoversi casualmente e di auto organizzarsi arbitrariamente, sebbene la topologia wireless vari rapidamente ed in modo imprevedibile. Grazie a queste possibilità non sono più necessarie infrastrutture costose per portare connettività; deve essere comunque presente un nodo gateway per fornire accesso ad Internet. Le applicazioni che più si adattano a reti di tipo MANET sono le comunicazioni in campo di battaglia, stato di emergenza durante i disastri

naturali, conferenze temporanee, nelle reti veicolari (VANET) e nei paesi in via di sviluppo dove una grande porzione della popolazione vive in aree con una infrastruttura molto limitata e che quindi beneficerebbero molto di una tecnologia MANET.

Conseguenzialmente è cominciato a crescere il numero di applicazioni con richieste quality of service (QoS), le quali si sono diffuse in moltissimi ambienti (es. reti delle industrie automatizzate, sistemi embedded, sistemi di conferenza, streaming di file multimediali). Con un servizio QoS si intende garantire maggiori prestazioni ad alcuni flussi di dato che hanno una maggiore priorità rispetto ai flussi aventi bassa priorità, dando delle garanzie per alcune caratteristiche di trasmissione (throughput, delay massimo, ecc...).

Nel garantire queste caratteristiche di comunicazione, giocano un ruolo importante i protocolli usati per gestire l'accesso al canale e l'invio dei dati tra i vari nodi facenti parte della rete. I collegamenti wireless possiedono caratteristiche che generalmente sono molto differenti dai collegamenti via cavo: sono soggetti ad attenuazioni di segnale dipendenti dal tempo e dallo spazio, fading, interferenze e rumore che danno come esito errori nelle trasmissioni e qualità del canale molto variabili nel tempo.

Il livello MAC dello stack protocollare ha un ruolo fondamentale nella gestione degli accessi al canale; questo livello si scontra con le problematiche di contesa del canale e possibili collisioni tra i vari nodi della rete che cercano di accedere contemporaneamente allo stesso canale.

I protocolli di livello MAC possono essere di due tipi: centralizzati o distribuiti. Gli schemi ad accesso centralizzato si basano su un meccanismo di accesso coordinato da una stazione (di solito l'access point) che gestisce le trasmissioni del canale; ad esempio le trasmissioni possono basarsi su un TDMA, FDMA o CDMA, dove le stazioni ricevono rispettivamente time slots, frequenze o codici; un altro esempio è quello utilizzato dal PCF in IEEE 802.11, dove le stazioni possono trasmettere solo se ricevono il polling dalla stazione centrale. Questi schemi purtroppo sono inappropriati per le reti di tipo MANET in quanto è difficile trovare o scegliere un amministratore

centrale che coordini le trasmissioni. In questo caso vengono utilizzati schemi di tipo distribuito, come per esempio il protocollo Alhoa, CSMA, MACA, MACAW, FAMA o il CSMA/CA utilizzato dall'IEEE 802.11 DCF.

L'Istitute of Electrical and Electronics Engineers (IEEE) ha sviluppato un standard per supportare le QoS chiamato 802.11e [4], facente parte della famiglia 802.11. Il meccanismo di accesso al canale dell'802.11e prevede l'assegnamento di una differente priorità ad ogni tipo di dato basandosi sulla sua richiesta QoS e assegnando un set di parametri differente ad ogni priorità, permettendo quindi un accesso più privilegiato ai dati con alta priorità rispetto a quelli di bassa priorità.

Per dare supporto al traffico multimediale in una rete totalmente distribuita come le reti ad-hoc, il protocollo MAC deve fornire garanzie di QoS. Generalmente i protocolli QoS raggiungono questo obiettivo di differenziazione dei servizi assegnando alle varie classi di traffico parametri differenti per l'accesso al canale trasmissivo. Un esempio è l'appena citato IEEE 802.11e che assegna parametri differenti alle sue 4 classi di priorità; però le prestazioni del delay trasmissivo e del throughput degradano in presenza di una rete molto carica. Per risolvere questi problemi molti protocolli sono nati a partire dall'802.11e modificandone delle parti. Un esempio è [5] che adatta i valori della finestra di contesa iniziali in modo tale da garantire le richieste QoS in reti one-hop. QPART [6] adatta la finestra di contesa in base alle condizioni del canale e differenzia il servizio alle varie tipologie di traffico fornendo un servizio QoS-aware. Anche AQMP in [7] adatta i parametri di modifica della finestra di contesa, il persistence factor, in base alla priorità del flusso e attraverso l'uso di un indice che stima il carico della rete.

Oltre a questi schemi sviluppati a partire dall'IEEE 802.11 sono stati proposti altri protocolli. Nel protocollo Hiperlan [8] i nodi hanno più fasi di contesa; in una di queste trasmettono una serie di burst la cui lunghezza dipende dal delay riscontrato fino a quel momento, dando quindi più priorità ai flussi che sono in attesa da più tempo. Lo schema del Priority Binary Countdown (PCB) [9] prevede l'uso da parte di ogni stazione l'uso di un competition

number (CN) come segnale per la contesa; questo numero è composto da un priority number, un random number e un ID number. Ogni bit del CN viene considerato come un bit-slot in cui la stazione trasmette un segnale se il bit è a 1 o ascolta il canale se è a 0; se mentre ascolta sente il canale occupato allora ha perso la contesa. Vince chi arriva alla fine del CN senza aver mai perso e quindi ha la possibilità di trasmettere il segnale. Questo protocollo ha bisogno di una sincronizzazione molto precisa che nelle reti ad-hoc è difficile da ottenere.

I problemi che affliggono questi protocolli e quelli che verranno descritti poi nei capitoli successivi del documento sono quelli per cui se un protocollo decide di essere più ferreo nei riguardi dei servizi QoS finisce nel cadere nel problema della starvation, mentre se adotta metodi di accesso randomizzato per fornire statisticamente l'accesso al canale anche alle stazioni più "deboli" non riesce a garantire servizi QoS. In tutto questo le politiche di fairness non sono ben definite e quindi molti di questi protocolli non sono capaci a supportare un accesso equo tra le stazioni che contendono il canale.

Per dare supporto al QoS viene proposto un nuovo protocollo chiamato PAB (*Priority Access with Burst*) che consiste in un accesso al canale preceduto da una serie di invii di burst, inviati alla stessa frequenza dei dati, che inibiscono la trasmissione di stazioni avente minore priorità. Questo protocollo è caratterizzato dalle sue politiche di fairness che offrono a tutte le stazioni la possibilità di trasmissione. Lo scopo del MAC PAB è quello di fornire servizi QoS senza incorrere nel problema della starvation, un problema che affligge la maggior parte dei protocolli presenti. Per far ciò innesca un meccanismo di accesso al canale di tipo round-robin per cui una stazione che non trasmette acquista priorità attraverso un meccanismo di aging che quindi gli dà la certezza di trasmissione.

In questo documento verranno analizzate le differenze tra i protocolli IEEE 802.11e e PAB confrontandone le prestazioni al fine di valutare le caratteristiche positive e negative di entrambi i protocolli.

Capitolo 1

IEEE 802.11e

1.1 Introduzione

Lo standard IEEE 802.11 per WLAN viene ampiamente usato in moltissimi ambienti. Le caratteristiche principali delle reti 802.11 sono la sua semplicità e la sua robustezza contro rotture improvvise di host in un approccio distribuito. Usando la banda ISM a 2.4 GHz, la versione 802.11b fornisce un data rate fino a 11 Mbit/s. La 802.11a, invece, può arrivare ad un data rate di 54 Mbit/s usando la tecnica di modulazione OFDM ad una frequenza di banda di 5 GHz. Oggi le WLAN 802.11 possono essere considerate come la versione wireless dell'Ethernet, la quale fornisce un supporto best-effort. Negli ultimi anni è andato crescendo l'interesse per il supporto QoS nelle reti wireless, così da poter dare la possibilità alle varie applicazioni multimediali di essere inserite sui vari dispositivi mobili.

1.2 802.11 legacy

In questa sezione verrà fatto un breve sommario sul livello MAC dell'802.11 e discusse le sue limitazioni al supporto QoS.

1.2.1 Distributed Coordination Function

La funzione di base del protocollo MAC 802.11 è la Distributed Coordination Function (DCF) basata sul Carrier Sense Multiple Access (CSMA). Le stazioni inviano il loro MAC Service Data Units (MSDUs) di lunghezza arbitraria (fino a 2304 byte) solo dopo aver rilevato che non ci sono altre trasmissioni in corso nel canale wireless. Può accadere però che due o più stazioni trasmettino contemporaneamente provocando quindi una collisione. L'802.11 definisce un meccanismo di Collision Avoidance (CA) per ridurre le probabilità di collisione. Prima di iniziare a trasmettere viene avviata una procedura di backoff: la stazione deve controllare che il canale sia libero per un intervallo di tempo casuale, con un minimo di tempo di attesa chiamato DCF Interframe Space (DIFS). Solo se il canale resta libero per questo periodo, la stazione ha il permesso di trasmettere. La durata di questo tempo è calcolato come un multiplo dello slot time. Ogni stazione ha una variabile chiamata *Contention Window* (CW), la quale è usata per determinare il numero di slot time che bisogna attendere prima di una trasmissione. Per ogni ricezione di un frame andata a buon fine, la stazione ricevente invia un messaggio di conferma (ACK) di avvenuta ricezione del frame. Il valore del CW viene incrementato ad ogni trasmissione fallita, cioè quando non viene ricevuto l'ACK di conferma. Dopo una trasmissione non riuscita si avvia un'altra procedura di backoff con un valore doppio del CW. Questo riduce la probabilità delle collisioni nel caso ci fossero altre stazioni che contendono per l'accesso al canale wireless. Se una stazione rileva che il canale diventa occupato durante il suo tempo di backoff, blocca il suo timer e lo riattiva soltanto dopo aver riascoltato il canale libero per almeno un DIFS, senza ricominciare la procedura di backoff. In questo modo una stazione che ha rimandato la sua trasmissione a causa di un valore alto di backoff rispetto ad altri, avrà una priorità più alta al suo prossimo tentativo. Dopo ogni trasmissione andata a buon fine viene fatta una nuova procedura di backoff anche se non ci sono MSDU da inviare. Questo si chiama 'post-backoff'. Esiste una situazione in cui una stazione non deve eseguire la procedura di

backoff, e questo succede proprio quando un MSDU arriva dopo che il post-backoff è terminato, e quindi una stazione ha la libertà di trasmettere il suo dato dopo aver rilevato il canale libero per il tempo minimo di DIFS. Per

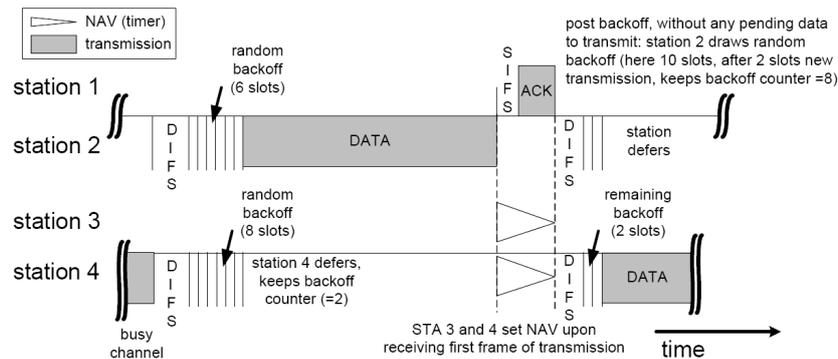


Figura 1.1: Il post-backoff viene effettuato dopo ogni trasmissione riuscita, anche se non ci sono MSDU da trasmettere

ridurre il problema delle Hidden Terminal, 802.11 definisce un meccanismo Request-to-Send/Clear-to-Send (RTS/CTS). Prima di trasmettere il dato, una stazione invia un RTS, seguito da una risposta, da parte del ricevente, del CTS. Questi due messaggi contengono informazioni sulla durata della trasmissione, compreso di frame e risposta ACK, e quindi informano gli altri nodi che sta per iniziare una trasmissione. Un nodo che vede il CTS sa di essere vicino al ricevitore, per cui non potrà trasmettere per il periodo di tempo richiesto, e quindi setta un timer chiamato Network Allocation Vector (NAV). Un nodo che vede invece un RTS ma non il CTS non è abbastanza vicino al ricevitore per interferire, per cui può trasmettere liberamente.

La figura 1.2 mostra un esempio del DCF.

1.2.2 Point Coordination Function

Per dare supporto alle applicazioni time-bounded, l'IEEE 802.11 definisce il *Point Coordination Function* (PCF) per permettere alle stazioni di

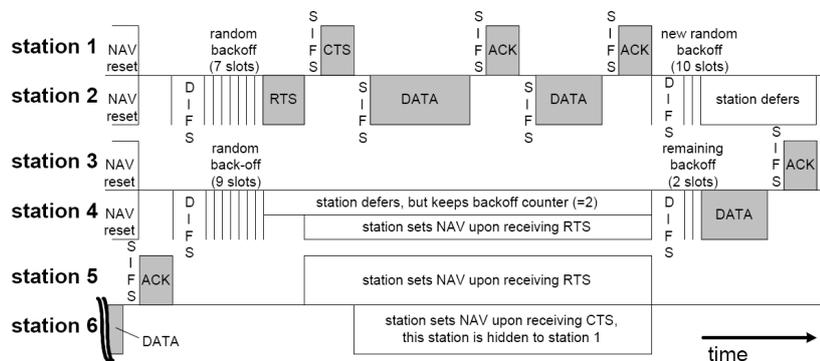


Figura 1.2: In questo esempio la stazione 6 non riceve l'RTS trasmesso da 2, ma il CTS trasmesso da 1

avere una sicurezza di accesso al canale wireless. Il coordinamento delle trasmissioni avviene tramite una stazione chiamata *Point Coordinator* (PC).

Contention Free Period e Superframe

Il PCF ha una priorità maggiore rispetto al DCF perché è possibile trasmettere dopo un periodo di tempo minore del DIFS; questo spazio temporale è chiamato PCF Interframe Space (PIFS). Il tempo viene suddiviso in periodi che si susseguono, chiamati superframe. Con il PCF si alternano nel tempo un *Contention Free Period* (CFP) e un *Contention Period* (CP), dove un CFP e il successivo CP formano un superframe. Per l'accesso al canale, durante il CFP viene usato il PCF, mentre nel CP il DCF. E' ovvio che la lunghezza del CP deve essere tale per consentire l'invio di almeno un MSDU con il DCF. Un superframe inizia con un frame speciale chiamato *beacon*, al fine di segnare l'inizio del CFP. Il PC genera il beacon ad intervalli regolari al fine di informare tutte le stazioni della durata del superframe; questo periodo di tempo è chiamato *Target Beacon Transition Time* (TBTT) ed è dichiarato ad ogni invio del beacon. Durante il CFP non c'è contesa per il canale ma viene effettuato un meccanismo di polling. Il PC invia messaggi CF-Poll con disciplina round robin, e solo un terminale che ha ricevuto un CF-Poll oppu-

re un pacchetto dati è autorizzato a trasmettere. La figura 1.3 mostra una tipica sequenza di scambio di frame durante un CFP. Il PC invia un CF-Poll ad una stazione, ma siccome aveva dei dati da trasmettergli, sfrutta l'occasione e li inserisce nel CF-Poll. Se il PC non riceve risposta dalla stazione a cui ha inviato il CF-Poll entro un PIFS, allora può proseguire con l'invio del CF-Poll alla prossima stazione, o terminare il CFP con un CF-End. Come si può vedere, durante un CFP non ci sono periodi di inattesa maggiori di PIFS.

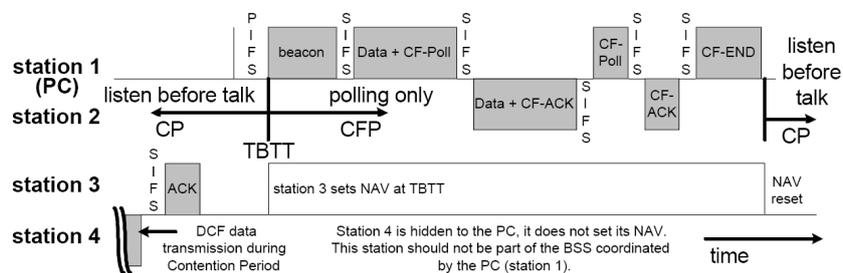


Figura 1.3: Esempio di una operazione PCF. La stazione 1 è il PC che invia un CF-Poll alla stazione 2. La stazione 3 riceve il beacon e setta il NAV per l'intero CFP. La stazione 4 non vede la stazione 1 e quindi continua ad operare con il DCF

Supporto al QoS con PCF

Ci sono molti problemi che non permettono a questo sistema di essere capace di soddisfare le richieste delle applicazioni QoS. I problemi principali sono il tempo di delay del beacon che risulta imprevedibile e la durata sconosciuta delle trasmissioni delle stazioni che hanno ricevuto il CF-Poll. La figura 1.4 mostra un'illustrazione del problema. Allo scadere del TBTT il PC schedula il beacon come prossimo frame da inviare e che potrà essere trasmesso solo dopo aver rilevato il canale libero per un tempo pari a PIFS. A questo punto dipende dallo stato del canale: se è libero allora il frame può essere inviato, altrimenti si ha un delay nell'invio del beacon. Da questo ne

quattro differenti *Access Categories* (AC), ogniuna per una differente categoria di priorità. L'accesso al canale è basato sulla priorità del traffico dati, cioè ogni frame avente una certa priorità, viene assegnato ad una categoria tra le AC e quindi si usa un diverso set di parametri, diverso per ogni AC, per la contesa del mezzo.

1.3.2 EDCA (Enhanced Distributed Channel Access)

L'EDCA offre un sistema distribuito e differenziato al canale usando differenti priorità per ogni tipo di traffico dati.

Access Categories (ACs)

EDCA definisce quattro Access Categories (ACs) per i differenti tipi di traffico ai quali verrà assegnato un set di parametri distinto per ogni categoria. I frame provenienti dai diversi tipi di traffico dati vengono mappati nelle quattro categorie AC a seconda della loro richiesta QoS. Le quattro AC vengono chiamate AC-BK, AC-BE, AC-VI e AC-VO, rispettivamente per Background, Best Effort, Video e Voice, dove AC-BK ha la più bassa priorità, mentre AC-VO la più alta. Ogni frame che arriva dai livelli superiori arrivano al MAC con una priorità chiamata *User Priority* (UP) e che ha valori tra 0 e 7.

Priorità	User Priority(UP)	Access Category (AC)	Tipo Dati
Lowest	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
Highest	7	AC-VO	Voice

Tabella 1.1: Mappatura da User Priority (UP) a Access Category (AC)

Enhanced Distributed Coordination Function (EDCF)

Ogni stazione ha quattro code di dati da trasmettere, una per ogni AC e quattro indipendenti EDCF, una per ogni coda, come illustrato in figura 1.5. EDCF è una versione migliorata del DCF e per la contesa del canale applica gli stessi principi del CSMA/CA e backoff, ma basati sui parametri caratterizzanti ogni AC.

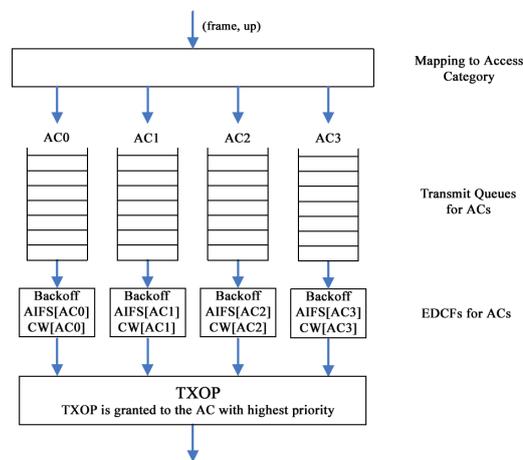


Figura 1.5: Quattro AC, ogniuna con la sua coda, AIFS, CW e backoff timer

La contesa per l'accesso al canale si basa sui seguenti parametri:

- AIFS - Il periodo di tempo che il canale deve essere ascoltato libero prima di una trasmissione o prima dell'inizio della procedura di backoff.
- CWmin, CWmax - Grandezza della CW usata per il backoff.
- PF - Fattore di moltiplicazione usato nel calcolo del backoff.
- TXOP Limit - La durata massima di una trasmissione dopo che il mezzo trasmissivo è stato preso.

I valori di questi parametri sono differenti per ogni AC. Fondamentalmente gli AC con una priorità alta hanno AIFS, CWmin, CWmax e PF minori e TXOP Limit maggiore rispetto agli AC con priorità bassa. I valori di questi parametri sono specifici per ogni AC e vengono chiamati AIFS[AC], CWmin[AC], CWmax[AC], PF[AC] e TXOP Limit[AC]. Questa è la differenza principale tra il DCF e l'EDCF: parametri differenti per ogni priorità.

AC	CWmin	CWmax	AIFSN	TXOP Limit	
				FHSS	DSSS
AC-BK	CWmin	CWmax	7	0	0
AC-BE	CWmin	CWmax	3	0	0
AC-VI	(CWmin+1)/2-1	CWmin	2	6.016ms	3.008ms
AC-VO	(CWmin+1)/4-1	(CWmin+1)/2-1	2	3.246ms	1.504ms

Tabella 1.2: Valori di default per i parametri dell'EDCA

AIFS (Arbitration Inter-Frame Space) E' l'intervallo di tempo minimo durante il quale il canale deve essere rilevato libero prima che una stazione abbia il permesso di inviare dati o di avviare la procedura di backoff. L'AIFS non è fissata come nel DCF, ma è un valore variabile che dipende dall'AC del frame da inviare. L'AIFS è calcolato nel seguente modo:

$$AIFS = AIFSN \cdot aSlotTime + aSIFSTime$$

dove aSlotTime è lo slot time, ASIFSTime è il periodo di tempo SIFS e AIFSN (Arbitration Inter-Frame Space Number) è usato per determinare la lunghezza dell'AIFS. AIFSN definisce il numero degli slot che devono essere aggiunti al SIFS. Il valore minimo del AIFSN è 2, cioè AIFS uguale al DIFS. La figura 1.6 mostra come la priorità viene data agli AC in base agli AIFS. Un valore basso di AIFSN per le alte priorità permette di attendere un tempo minore prima di trasmettere o avviare la procedura di backoff. In questo modo le priorità alte degli AC hanno uno sfruttamento maggiore della banda rispetto alle basse priorità. Un basso AIFS assicura anche un delay minore, e questo è importantissimo per le applicazioni time-bounded.

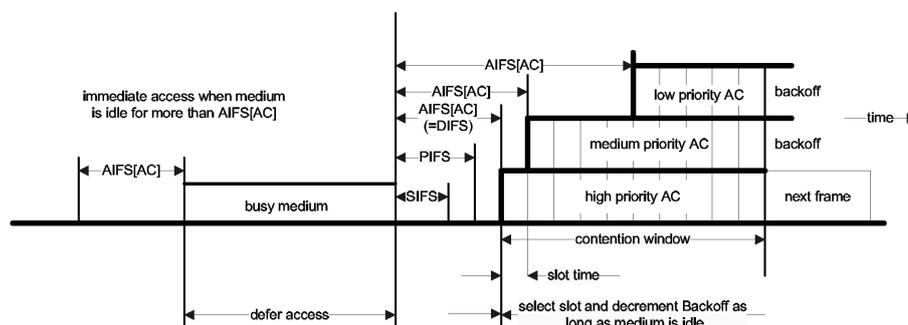


Figura 1.6: Meccanismo di priorità basato sul AIFS

CWmin e CWmax Anche il valore limite minimo e massimo del Contention Window (CW) non è fissato come nel DCF, ma è variabile a seconda della priorità. I valori di default per CWmin e CWmax sono quelli presenti nella tabella 1.2. I valori specifici CW dipendenti dal livello fisico sono quelli in tabella 1.3.

	FHSS	DSSS
CWmin	15	31
CWmax	1023	1023

Tabella 1.3: Valori del CW per differenti tipi di livello fisico

Un valore piccolo per il CW causa un valore di backoff minore e quindi il tempo di attesa da sommare all'AIFS è minore rispetto ad un valore che si avrebbe con un CW alto. Come si può notare dalla tabella 1.2, per i livelli bassi di AC, i valori sono gli stessi del DCF, mentre per i livelli alti sono la metà e un quarto, rispettivamente per AC-VI e AC-VO. Il significato di questi valori è quello di offrire ai livelli più alti di AC un delay minore per l'accesso al canale. L'aspetto negativo di avere un CW basso è quello che il numero di collisioni aumenta a causa dell'incremento della possibilità che più nodi inizino la comunicazione contemporaneamente. I valori dei CWmin e CWmax sono anche tali da evitare sovrapposizioni tra i livelli. Questo

significa che dopo un raddoppiamento del valore del CW a causa di una trasmissione fallita, il valore del CW resta comunque inferiore al CWmin del livello a priorità minore. Dopo una trasmissione fallita il CW viene raddoppiato fino ad arrivare al valore massimo, e questo fa sì che per i livelli ad alta priorità il valore del CWmax viene subito raggiunto, consentendogli di scegliere valori di backoff minori e quindi vincere la contesa per l'accesso al canale. In altre parole le performance dei livelli a bassa priorità degradano velocemente in presenza di un flusso continuo di dati provenienti da livelli superiore.

Persistence Factor come parametro per le AC In 802.11 legacy dopo una trasmissione fallita, si sceglie la nuova CW in base a quella vecchia moltiplicata per due. Il nuovo timer di backoff viene poi scelto in questa finestra più ampia per cercare di evitare le collisioni. Nell'802.11e invece di incrementare la CW per un fattore due, viene usato il *Persistence Factor* (PF[AC]) per il supporto alle priorità degli AC. Mentre nell'802.11, il CW viene raddoppiato ad ogni trasmissione fallita (equivalente ad un PF=2), nell'802.11e si usa un PF[AC] differente per ogni AC. La grandezza del CW è calcolato nel modo seguente

$$CW(i) = \min(CWmax[AC], (CWmin[AC] + 1) \cdot PF[AC]^{i-1} - 1),$$

dove i indica il livello del backoff.

Il PF[AC] non è incluso nello standard 802.11e ma è un importante fattore per il supporto alle applicazioni QoS come si può notare nelle figure 1.7 e 1.8. Nella 1.7 il PF è uguale per tutte le AC, mentre nella 1.8 si hanno tre diversi fattori PF. Come si può notare avere un fattore piccolo implica che la CW non si espande velocemente e quindi offre una possibilità di accesso al canale maggiore.

Le figure indicano che il PF[AC] aiuta al supporto per le QoS specialmente in scenari con molte collisioni. Per esempio, in caso di collisione, un CW ad alta priorità raggiunge il CWmax solo dopo molti tentativi di trasmissione falliti, al contrario delle basse priorità che lo raggiungono in pochi passi.

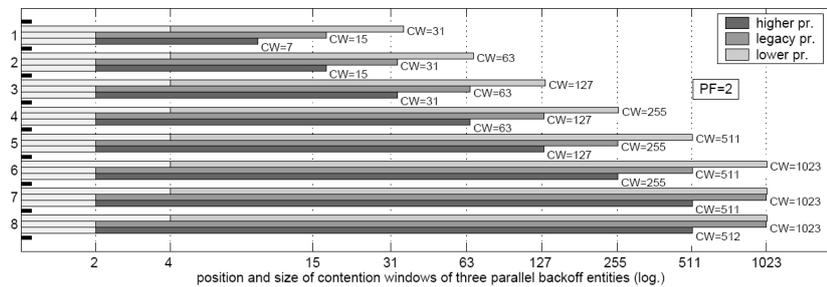


Figura 1.7: Cambio della grandezza del CW con $PF=2$ per ogni AC

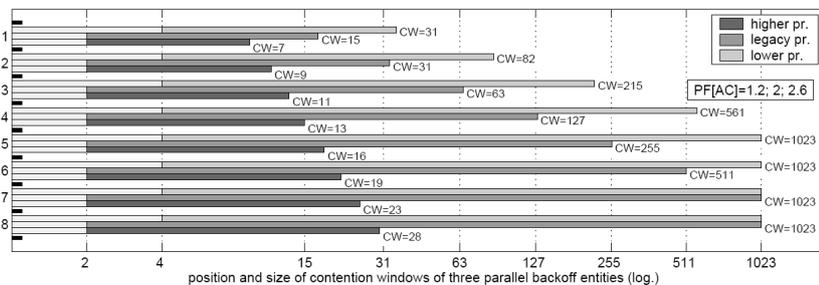


Figura 1.8: Cambio della grandezza del CW. In questo vengono usati differenti PF per i diversi AC

TXOP (Transmission Opportunity) TXOP è la durata di tempo durante il quale un AC può trasmettere dopo aver vinto la contesa per il mezzo trasmissivo. TXOP è caratterizzato da un limite massimo chiamato TXOP Limit. La durata della trasmissione copre l'intera sequenza di scambio frame, incluso i SIFS intermedi, gli ACK, l'RTS e il CTS. La tabella 1.2 mostra i valori di default del TXOP Limit per i diversi AC. Questo limite indica che un AC ha il permesso di trasmettere frame multipli durante il TXOP, l'importante è che non superi il limite. Questo periodo di invio continuo di frame durante il TXOP viene chiamato *Contention Free Bursting* (CFB). Le trasmissioni consecutive di frame sono separate da un SIFS, come mostra la

figura 1.9. E' importante sottolineare che la trasmissione di frame multipli è concessa all'AC e non alla stazione, quindi l'invio dei frame è consentito solo ai frame che appartengono all'AC vincente. Se viene usato il meccani-

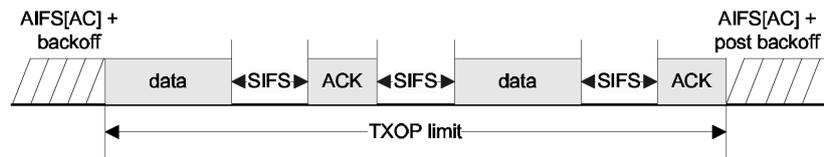


Figura 1.9: Contention Free Bursting (CFB)

smo RTS/CTS con il CFB, allora lo scambio dei messaggi RTS e CTS viene fatto una sola volta, e cioè prima del primo frame. Un TXOP Limit a zero indica che il CFB è disabilitato per quel AC, e quindi può essere inviato solo un frame dopo il RTS/CTS. Come si può vedere nella tabella 1.2 si hanno valori nulli per le basse priorità e quindi CFB disabilitato. Nel caso in cui il CFB è abilitato, il campo *Duration* presente nell'header del frame è settato alla durata del tempo rimanente dell'intero TXOP, così le stazioni riceventi settano il loro NAV per l'intera durata del TXOP e non solo di un frame.

1.4 Conclusioni

Un numero di studi ha valutato lo standard sia in modo analitico sia tramite simulazioni e hanno dimostrato l'utilità dell'802.11e.

Sebbene questo protocollo migliora l'accesso al canale introducendo delle politiche volte a dare servizio QoS, ci sono delle questioni che devono essere risolte prima di poter affermare che questo protocollo fornisca un completo supporto al QoS. Ci sono principalmente tre problematiche (figura 1.10): la gestione delle condizioni molto variabili della rete nel tempo, l'adattabilità del protocollo al variare delle richieste dell'applicazione sovrastante e la gestione delle risorse del link layer [10].

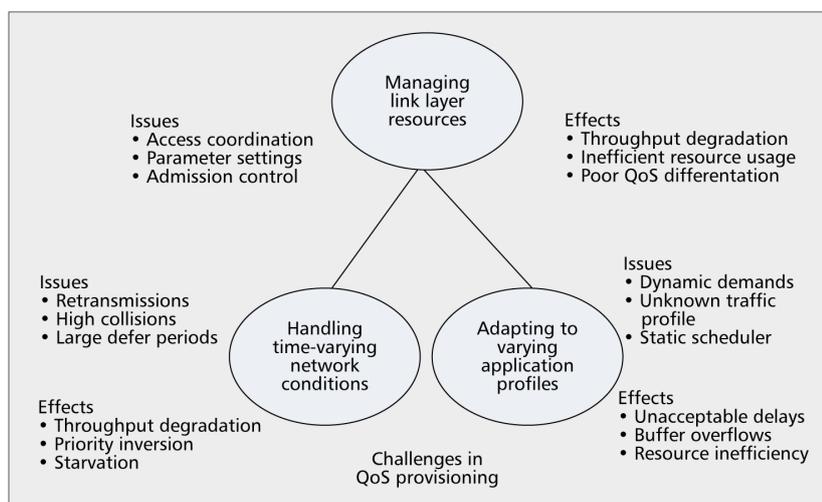


Figura 1.10: Problematiche del servizio QoS in 802.11e

Gestione delle condizioni variabili della rete Una delle limitazioni dell'802.11e è che non considera l'impatto della variazione delle condizioni della rete. Ci sono due fattori differenti che impattano sullo stato del canale: le condizioni del canale e il carico di rete. La condizione del canale è importante per la gestione del servizio QoS perché può potenzialmente far degradare il meccanismo di differenziazione dell'802.11e. Questo protocollo fornisce differenziazione quando tutte le stazioni sperimentano le stesse condizioni del canale; quando invece le condizioni sono differenti a causa della locazione fisica della stazione, interferenza e mobilità, il meccanismo fornito è insufficiente per garantire la differenziazione dei flussi trasmissivi. Nel peggiore dei casi le cattive condizioni del canale possono portare a situazioni chiamate *priority inversion*, dove le priorità alte soffrono maggiormente e raggiungono prestazioni più basse dei nodi a bassa priorità. La differenziazione del servizio è anche impattata negativamente dal numero delle stazioni che contendono per il canale; a causa di ritrasmissioni e rinvii nelle trasmissioni, i nodi con accesso prioritario non riescono più a raggiungere le loro richieste minime di accesso al canale.

Adattarsi alle modifiche dei profili delle applicazioni La seconda problematica da affrontare è quella della presenza di flussi aventi profili applicativi che variano nel tempo; questi profili generano poi le richieste di QoS. Se il profilo può essere stimato correttamente, allora l'accesso al canale può essere settato perfettamente per fornire le QoS richieste. Nello schema centralizzato HCCA, lo scheduler usa un meccanismo round-robin e alloca spazio di tempo alle stazioni in base alle informazioni inviate dalla stazione stessa. Le informazioni usate sono semplicemente la grandezza media del pacchetto e le richieste di throughput. L'HCF (Hybrid Coordination Function) può effettuare solamente polling di spazio a dimensioni fisse adatto ad un tipo di traffico di tipo CBR. Questo tipo di scheduling è inadatto però a flussi di tipo VBR in cui la dimensione del pacchetto e il throughput richiesto è variabile nel tempo.

Inoltre le informazioni riguardo le richieste real-time delle applicazioni non sono quasi mai disponibili all'inizio della trasmissione del flusso. Ad esempio l'uso di applicazioni che modificano la loro richiesta di bit rate in base alle condizioni del canale.

Gestione delle risorse del livello link Rimane il problema della gestione e coordinamento delle risorse di livello link e dei parametri QoS. Oltre alle problematiche sopra descritte, l'AP deve anche prendere in considerazione i parametri di tutto il sistema e le richieste di tutti i flussi. Ci dovrebbero essere tecniche di coordinamento tra la parte centralizzata e quella distribuita e uno schema di admission control che limiti il numero di flussi per garantire maggiormente la QoS. Il primo aspetto del gestore delle risorse è la coordinazione tra i due periodi di accesso per decidere le modalità di accesso. L'approccio usato è quello di inserire i flussi real-time solamente nel periodo HCCA, mentre il resto nel periodo EDCA. Con questo meccanismo però le stazioni real-time sperimentano valori alti di delay.

L'altro aspetto è il controllo degli accessi al canale tramite tecniche di admission control. Senza questo tipo di tecniche la rete non riuscirà mai a garantire

le caratteristiche QoS dei flussi. L'802.11e ha una politica di gestione degli accessi, ma solamente per il periodo HCCA.

In conclusione l'IEEE802.11e non da nessuna certezza di garantire QoS tra i flussi presenti nel sistema; questo a causa della sua natura probabilistica e dei problemi appena descritti; poi non garantisce l'assenza di starvation ne una politica fairness per l'accesso al canale.

Capitolo 2

PAB

Dopo una breve introduzione verrà spiegato il funzionamento del protocollo PAB.

2.1 Introduzione

Il protocollo *Priority Access with Burst* (PAB) è un protocollo di livello MAC che usa dei burst di segnale per accedere al canale. Questi burst vengono trasmessi prima di inviare il dato per impossessarsi del canale ed inibire temporaneamente le altre stazioni. La contesa del canale è *flow-based* e questo significa che l'accesso al canale viene conteso tra i flussi di dato e non tra le stazioni; conseguentemente anche i flussi presenti in una singola stazione contendono tra di loro.

Ogni flusso appartiene ad una classe di priorità (ne sono possibili quattro) e questa ne determinerà le proprietà per l'accesso al canale. I burst vengono trasmessi ad intervalli regolari prima dell'invio del dato vero e proprio; l'intervallo di tempo che intercorre tra un invio e l'altro (questo intervallo è chiamato *PrIFS* – Priority IFS ed è un multiplo dello slot size) e la lunghezza del burst trasmesso determinano la priorità temporanea o dinamica del flusso che si sta trasmettendo.

Ogni priorità viene suddivisa in sottopriorità; ad ogni flusso viene assegnata

una sottopriorità, che come vedremo sarà variabile e determinerà la lunghezza del PrIFS.

L'intento di questo protocollo è quello di cercare di creare un ordine nell'accesso al canale diminuendo la contesa diretta tra le stazioni e quindi evitando collisioni. Per questo sono state introdotte le priorità dinamiche che come effetto collaterale creano nel sistema un accesso al canale di tipo round-robin in cui l'invio dei dati viene effettuato a turno da tutte le stazioni. L'accesso comunque resta sempre non deterministico in quanto ci troviamo in un ambiente dinamico come le reti ad-hoc, in cui il numero di stazioni presenti nel sistema in un dato istante di tempo non è determinabile.

Un'altro problema che cerca di affrontare questo protocollo è l'accesso di tipo fairness tra i flussi. Viene a crearsi infatti una sorta di aging per cui una stazione che perde la contesa aumenta la sua priorità dinamica (responsabile dell'accesso al canale) aumentando la sua possibilità di trasmettere alla prossima occasione, mentre il flusso che ha trasmesso resetta la sua priorità dinamica in modo da lasciare spazio agli altri flussi. Questa caratteristica cerca di evitare anche il problema della starvation in cui una stazione non riesce mai ad inviare dati a causa della presenza costante di stazioni ad alta priorità.

Il principale obiettivo del protocollo resta comunque l'offerta di un servizio di *QoS* ai flussi per cui uno di priorità più alta deve trasmettere prima di uno a priorità più bassa. Come vedremo nel seguito del documento questo viene ottenuto giocando con le priorità dinamiche e grazie all'uso di un superframe per cui un flusso trasmetterà prima di un altro se ha maggiore priorità.

Questo protocollo si basa sullo standard IEEE 802.11, quindi ogni altro funzionamento non descritto in questo documento è da considerarsi uguale a quello dello standard (l'uso dell'RTS e CTS, la dimensione del SIFS, il metodo di backoff per evitare collisioni con l'uso della finestra di contesa variabile, ecc...).

PAB è ispirato a [11] dove viene proposto un meccanismo di black burst per il carrier sense multiple access with collision avoidance (CSMA/CD) del

MAC 802.11 per dare supporto al traffico real-time. Vengono classificati due tipi di stazioni: i *real-time node* che trasmettono dati di priorità alta ed i *data node* che trasmettono dati di bassa priorità. I real-time node hanno una più alta priorità rispetto ai data node nell'accesso al canale accedendo con un inter-frame space più piccolo. Inoltre il real-time node inviano degli impulsi di energia (black burst - BB) prima della trasmissione. La lunghezza del BB trasmesso dai nodi real time è una funzione crescente rispetto al delay che il nodo sta avendo dall'ultima trasmissione. I BB dei vari nodi real-time differiscono tra loro di almeno un black slot e quindi c'è un solo vincitore tra i nodi che vogliono accedere al canale. Infatti se c'è ancora il BB sul canale dopo la trasmissione del proprio BB, allora vuol dire che la contesa è persa. In figura 2.1 c'è un esempio della contesa tra due nodi. Questo protocollo si

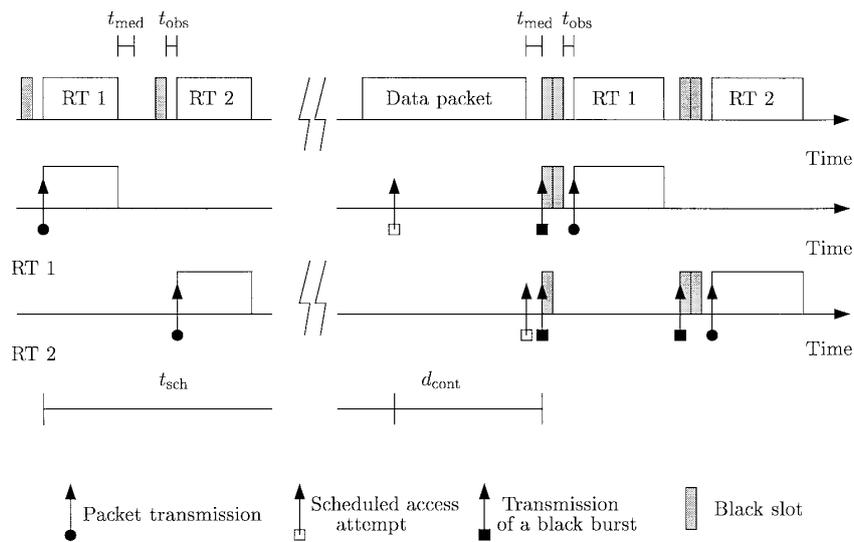


Figura 2.1: Meccanismo di contesa tramite BB

basa su delle ipotesi: assenza di hidden terminal, tutti i nodi real-time hanno lo stesso data rate nella creazione dei pacchetti real-time da trasmettere e questa trasmissione deve essere costante nel tempo. Quindi questo protocollo non prevede l'accesso a stazioni real-time aventi richiesta di banda differente

nel tempo come nel caso di un'applicazione VBR. In [12] viene modificato questo meccanismo per permettere l'invio di flussi real time provenienti da applicazioni VBR; viene proposto un protocollo chiamato EBB (Enhanced Black Burst) in cui si setta un limite alla lunghezza del burst, oltrepassato il quale la stazione real-time accede al canale in modalità DCF.

In [13] le classi di priorità sono differenziate dalla lunghezza del burst e quindi le stazioni di alta priorità aventi un BB più lungo riescono ad accedere prima delle altre. In questo caso la risoluzione della contesa tra stazioni avente la stessa priorità avviene mediante un algoritmo che assegna ad ogni stazione un ID in base al quale accedere.

[14] invece assegna un AIFS differente ad ogni priorità come 802.11e (fig 2.2). Dopo che il canale resta libero per un AIFS (diverso per ogni priorità), invece di attendere ancora per un tempo definito dal backoff scelto, la stazione invia un jamming signal (black burst) sul canale per tutta la durata del backoff. Al termine del burst se il canale è libero allora si suppone di aver vinto la contesa e si invia il dato. In questo caso le stazioni di alto livello vincono sicuramente contro le altre stazioni visto la differenza nella lunghezza dell'AIFS; questo però potrebbe portare a problemi di starvation.

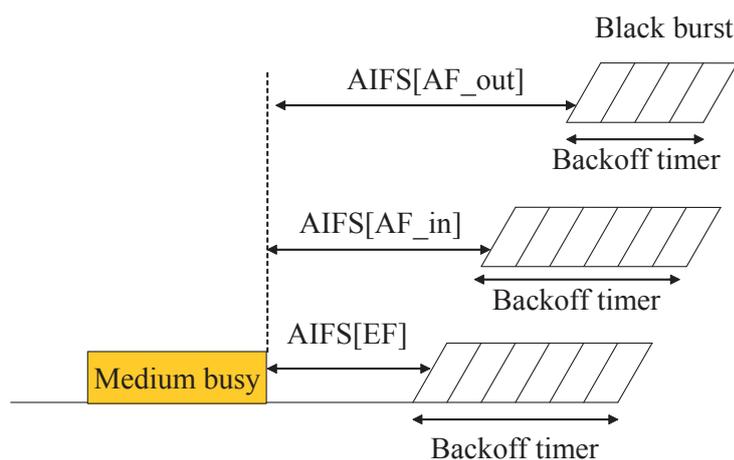


Figura 2.2: Schema di contesa attraverso l'uso di black burst e AIFS differenti

QMA [15] che implementa l'accesso al canale tramite forecast bursts (FB). L'accesso è diviso in due fasi: fase di contesa e fase trasmissiva. Un nodo deve sentire il canale libero per almeno T_{win} prima di accedere; se il canale resta libero allora la stazione ha il permesso di accedere alla fase di contesa. Questa fase consiste di $(n + m)$ mini slot lunghi T_{slot} ($T_{slot} < T_{win}$). I primi n slot sono assegnati ai nodi con traffico RT, mentre i restanti m sono assegnati al traffico NRT; la contesa avviene trasmettendo prima del dato vero e proprio una serie di FB dove $T_{FB} = T_{slot}$. Quindi i nodi con traffico NRT devono attendere prima che tutti gli n slot RT siano liberi prima di tentare l'accesso al canale. Se un nodo ascolta i primi B slot vuoti, allora può trasmettere $k * FB$ a partire dal B^{th} slot, altrimenti ferma la contesa. B è una variabile con una distribuzione geometrica limitata nell'intervallo $[0..n]$ per i flussi RT e $[n+1..m]$ per i flussi NRT. Dopo la trasmissione degli FB il nodo ascolta il canale e se resta libero per un tempo lungo T_{obs} ($T_{obs} = T_{slot}$) allora la contesa è vinta. Figura 2.3 mostra un esempio di accesso al canale.

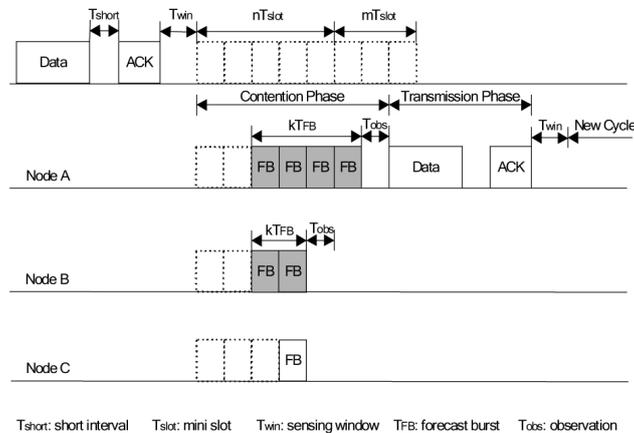


Figura 2.3: Esempio di accesso del protocollo QMA

Il protocollo DAQ-MAC [16] è simile al precedente; viene usato anche in questo caso il forecast burst FB. Il ciclo di accesso al canale consiste di tre fasi: la fase di priorizzazione, la fase di contesa e la fase trasmissiva. Se il canale è libero per un tempo lungo T_{win} allora la stazione accede alla fase di

priorizzazione; altrimenti addendono il prossimo ciclo. Lo scopo della fase di priorizzazione è quello di permettere il passaggio alla fase di contesa solo ai flussi RT, se presenti. Soltanto i nodi RT trasmettono un FB, mentre quelli BE ascoltano il canale; se c'è almeno una stazione RT che invia un FB, allora la contesa è persa e devono rimandare l'accesso al prossimo ciclo. La fase di contesa è composta da al massimo $n + m$ slot più un T_{obs} ed è formata da due sottofasi: la fase di ascolto e la fase di broadcast. Ci sono al massimo n slot nel periodo di ascolto ed ogni nodo resta in ascolto del canale per $J - 1$ slot. I nodi che ascoltano il loro ultimo slot libero hanno il permesso di passare alla fase di broadcast. J è una variabile che può assumere valori nell'intervallo $[1..n]$. La fase di broadcast è composta da massimo m slots e un T_{obs} ed ogni nodo trasmette $k * FB$, dove $0 < k < m$. Dopo la trasmissione degli FB la stazione attende T_{obs} e se il canale resta libero allora accede alla fase trasmissiva e trasmette il dato, altrimenti passa al prossimo ciclo. Figura 2.4 mostra un esempio di accesso al canale.

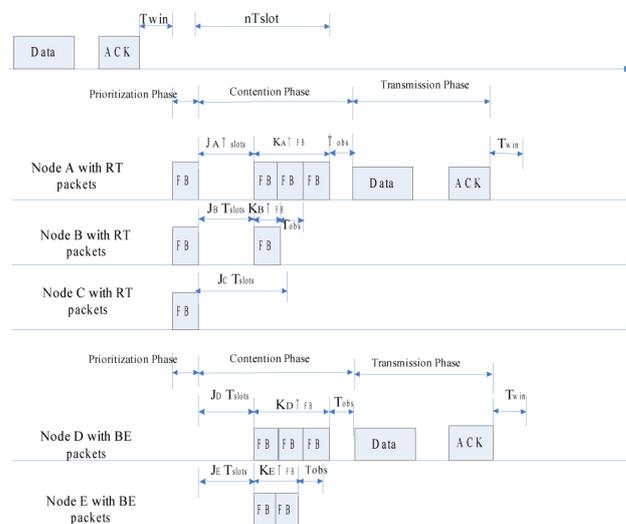


Figura 2.4: Esempio di accesso del protocollo DAQ-MAC

Altri protocolli sfruttano dei canali separati per trasmettere dei *busy tone signal* per risolvere la contesa del canale. Il protocollo BTPS in [17] utilizza

due busy tone: BT1 e BT2; le stazioni di alta priorità, prima della trasmissione del dato, inviano un impulso su BT1 ogni M slot (dove M è una costante del sistema); le stazioni di bassa priorità che ascoltano questo impulso inviano a loro volta un impulso su BT2. Le stazioni che trovano BT1 o BT2 occupato differiscono nella trasmissione (figura 2.5). Prima dell'invio del dato viene trasmesso un black burst per permettere alle stazioni che stavano inviando segnali sul BT2 di ritornare in ascolto sul canale dati. In questo modo le stazioni con bassa priorità vengono inibite dalla trasmissione dei busy tone signal.

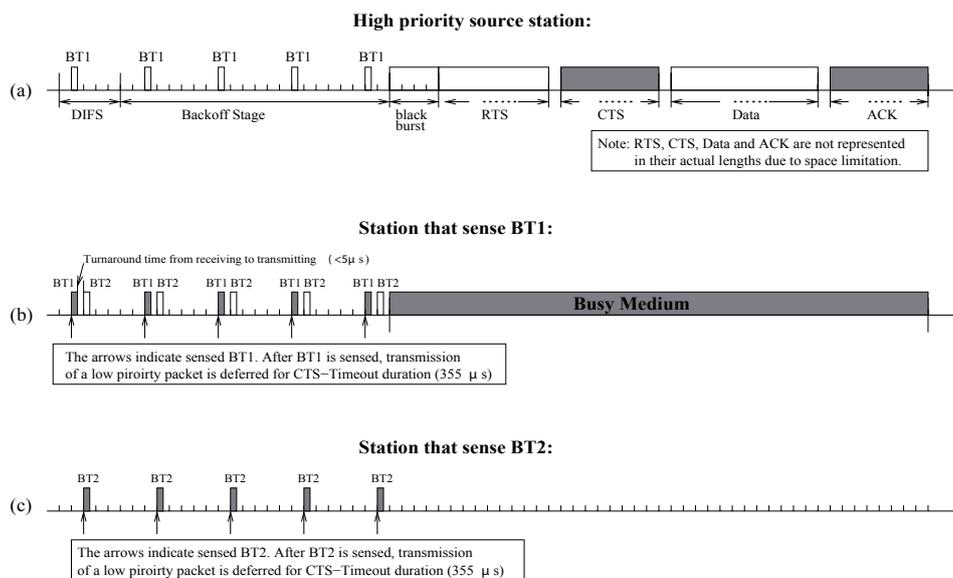


Figura 2.5: Comportamento del protocollo BTPS. (a) mostra il comportamento di una stazione di alto livello che ha un pacchetto da inviare; (b) mostra il comportamento di chi ascolta il segnale inviato su BT1; (c) mostra una stazione che ascolta soltanto il BT2

2.2 Funzionamento

2.2.1 Header pacchetti

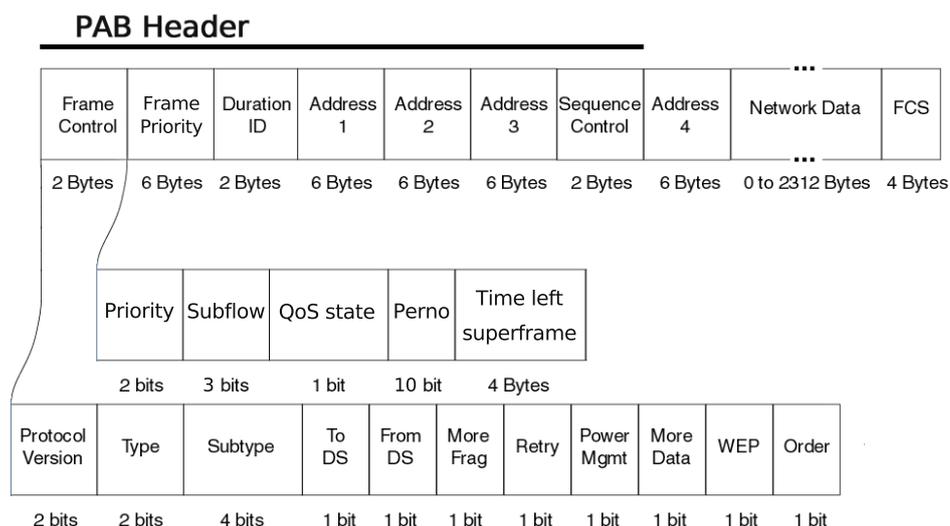


Figura 2.6: Header di un frame PAB

L'header dei pacchetti è lo stesso dell'802.11 con l'aggiunta dei seguenti campi:

Priority - un campo di 2 bit che indica la priorità del pacchetto (sono possibili 4 priorità);

Subflow - un campo di 3 bit che indica il subflow di appartenenza (chiamato anche subflow_id);

QoS state - un campo di 1 bit che indica in quale stato si trova la stazione trasmittente (QoS o CB);

Perno - un campo di 10 bit che indica il perno utilizzato dal flusso per trasmettere;

Time left superframe - un campo a 32 bit che indica i microsecondi che mancano dalla scadenza del timer della fine del superframe locale.

2.2.2 Definizioni

In seguito verranno descritti alcuni dei nomi di variabili e costanti che verranno poi usate e spiegate più in dettaglio in seguito.

Viene definito un nuovo interframe space: *PrIFS*. Questo valore è dinamico e dipende dalla *sub_priority* del flusso. *PrIFS* è calcolato nel seguente modo:

$$PrIFS = PrIFS_0 + (slotTime * sub_priority) \quad (2.1)$$

dove

$$PrIFS_0 = (2 * slotTime) + SIFS + \Delta PrIFS \quad (2.2)$$

$$\Delta PrIFS = slotTime - (SIFS \% slotTime) \quad (2.3)$$

in pratica il $\Delta PrIFS$ è calcolato in modo tale che $PrIFS_0$ sia un multiplo di *slotTime*.

- Ogni pacchetto appartiene ad un flusso dati. Un flusso presente nella stazione è caratterizzato, tra le altre cose, da una *sub_priority* e da un *burst_len* per l'accesso al canale.

Il valore del *burst_len* indica la lunghezza del burst; quest'ultimo viene calcolato nel seguente modo: $burst_len * slotTime$.

Un flusso è identificato dal suo *ID*, dal valore del *subflow_id* e dall'indirizzo MAC della stazione che lo sta trasmettendo; un flusso inoltre possiede una priorità che verrà chiamata in seguito *Pr* oppure *priorità originale*.

- *PR_MIN* e *PR_MAX* sono rispettivamente la priorità originale minima e quella massima che un flusso può avere. *PR_MIN* vale 3, mentre *PR_MAX* vale 0.

- *LV_PRIORITY* definisce il numero di sub_priority in cui ogni priorità originale viene suddivisa.
- Il valore del subflow_id può variare tra 0 e SUB_MAX_{Pr} dove Pr varia tra PR_MAX e PR_MIN.
- SUB_MAX_{Pr} dipende dalla priorità del flusso ed è uguale a $(PR_MIN - Pr) + 1$.
- *MAX_SUBPRIORITY* definisce il valore sub_priority più basso. Questo valore serve per lasciare una priorità massima ad un possibile access point che deve avere sempre la possibilità di accesso prima delle altre stazioni; può valere 0 se questa funzionalità non è attiva, altrimenti un valore maggiore o uguale a 1.
- *WEIGHT_PERNO_CALC* e *WEIGHT_PERNO_MEAN* sono delle costanti che indicano i pesi da usare nel calcolo del sub_priority_PERNO.
- *Superframe_END_CB* definisce il tempo rimanente alla fine del timer locale del superframe e quindi il termine del FRAME_CB.
- *Superframe_END_QOS* definisce il tempo rimanente alla fine del timer locale del FRAME_QOS.
- α è un numero compreso tra 0 e 1 che indica il rapporto tra la durata del FRAME_CB e tutto il superframe.

2.2.3 Scheduler

I pacchetti provenienti dai livelli superiori vengono mantenuti in delle code dallo scheduler in attesa di essere trasmessi. Lo scheduler mantiene SUB_MAX_{Pr} code di pacchetti per ogni flusso che viene gestito. Le code quindi sono definite dall'ID del flusso e dal subflow_id ed hanno una politica *FIFO*.

Quando deve essere preso un pacchetto per essere trasmesso, lo scheduler sceglie la coda del flusso con maggiore *priorità dinamica* e in caso di parità

tra più flussi sceglie il pacchetto che è arrivato prima. La priorità dinamica di un flusso dipende dai suoi valori di `sub_priority` e `burst_len`. Un flusso *f1* ha priorità dinamica maggiore rispetto a *f2*:

```

se sub_priority(f1) < sub_priority(f2)
oppure se (
    (sub_priority(f1) == sub_priority(f2)) &&
    (burst_len(f1) > burst_len(f2))
)

```

Lo scheduler ha anche il compito di mantenere tutte le informazioni riguardo ai flussi, tra cui la `sub_priority`, il `burst_len` e il `CWMin` (finestra di contesa minima, presente anche in 802.11) attuali del flusso. Quando gli viene richiesto un pacchetto dalla coda, restituisce oltre al pacchetto anche questi tre parametri che serviranno al MAC per accedere al canale.

Quando invece un pacchetto deve essere inserito in coda bisogna prima assegnargli un `subflow_id` (essendo questo valore non nativo del pacchetto); questo valore viene assegnato nel seguente modo: nel caso si tratti di un nuovo flusso allora viene assegnato l'identificativo 0, altrimenti $(last_sub + 1) \% SUB_MAX_{Pr}$, dove *last_sub* è l'identificativo assegnato all'ultimo pacchetto di quello stesso flusso inserito in coda. Quindi il pacchetto viene inserito nella coda identificata dal flusso a cui appartiene e dal `subflow_id` assegnatogli.

Se durante la gestione di un pacchetto arriva dai livelli superiori uno appartenente ad un flusso con priorità dinamica maggiore, allora viene rimesso in coda il pacchetto con priorità dinamica minore e viene preso in gestione il nuovo pacchetto.

2.2.4 Trasmissione dei pacchetti

Quando arriva un nuovo pacchetto dai livelli superiori, questo viene inserito in coda dallo scheduler. Dopodiché se la stazione non sta gestendo nessun altro pacchetto, allora viene preso dallo scheduler il prossimo pacchetto da inviare.

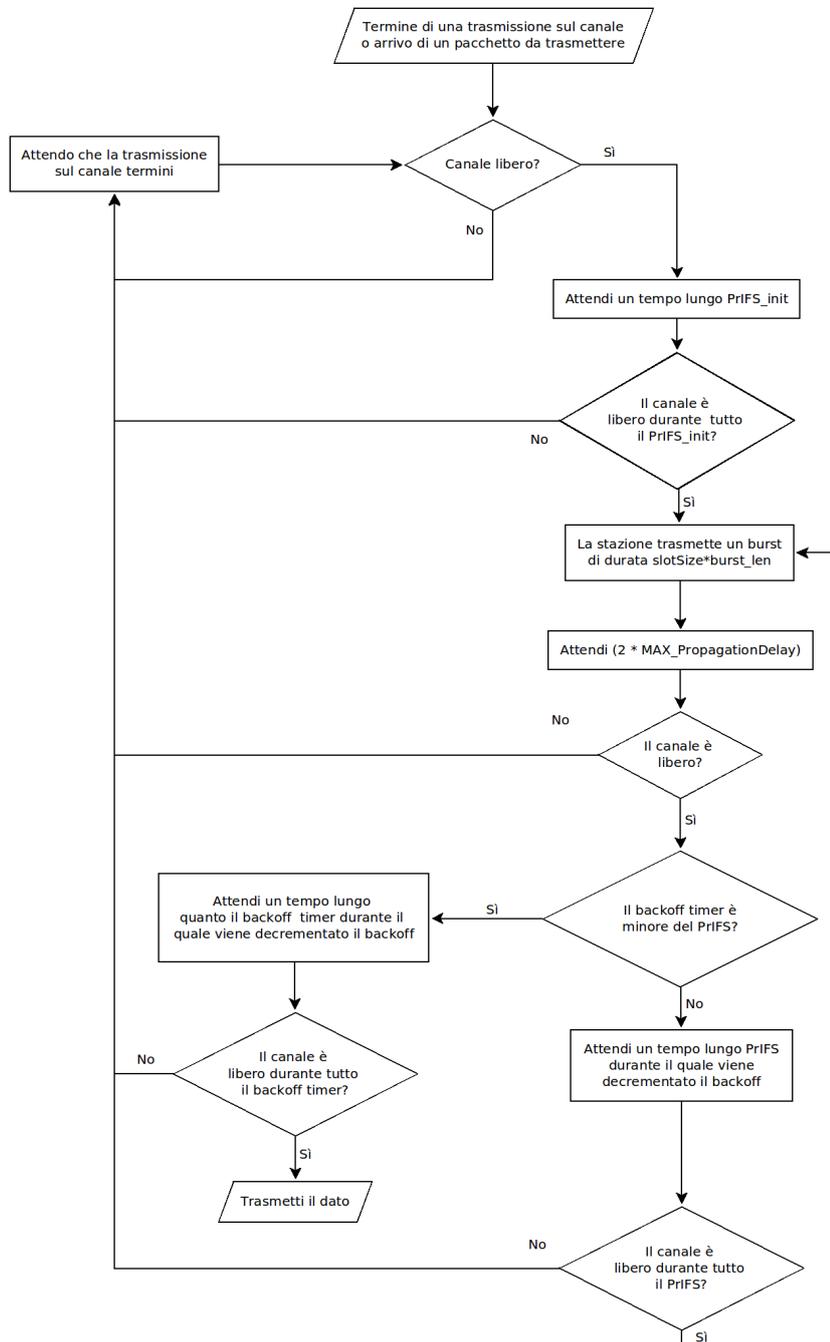


Figura 2.7: Flow chart che mostra l'accesso al canale di una stazione

In figura 2.7 viene mostrato un *flow chart* che descrive l'accesso al canale.

Insieme al prossimo pacchetto da inviare lo scheduler restituisce anche i parametri di *sub_priority*, il *burst_len* e *CWMin* del flusso a cui appartiene il pacchetto. Tramite il parametro di *sub_priority* viene calcolato il valore del PrIFS. Il valore $\text{PrIFS}_{\text{INIT}}$ indica il primo PrIFS e viene distinto dal normale PrIFS perché durante questo periodo di tempo non viene decrementato il backoff; il valore $\text{PrIFS}_{\text{INIT}}$ è identico al valore PrIFS.

L'algoritmo di accesso è il seguente:

1. se il canale è libero si avvia un timer di dimensioni $\text{PrIFS}_{\text{INIT}}$;
2. se durante l'attesa il canale resta libero, al termine del timer si trasmette un burst di dimensione $\text{burst_len} * \text{slotTime}$;
3. al termine della trasmissione si attende un tempo $2 * \text{MAX_PropagationDelay}$;
4. se il canale non è libero si attende la fine della trasmissione e si torna al punto 1;
5. se il timer di backoff è minore del PrIFS si avvia un timer di dimensioni del backoff timer e si passa al punto 6, altrimenti si avvia un timer di dimensioni PrIFS e si passa al punto 7;
6. durante l'attesa il valore del backoff viene decrementato e se il canale resta libero fino al termine del timer si trasmette il dato e la procedura termina, altrimenti si attende la fine della trasmissione e si torna al punto 1;
7. durante l'attesa il valore del backoff viene decrementato e se il canale resta libero fino al termine, la stazione invia un burst di dimensione $\text{burst_len} * \text{slotTime}$ e ritorna al punto 3, altrimenti si attende la fine della trasmissione e si torna al punto 1.

Nelle figure 2.8 e 2.9 vengono mostrati due esempi che mostrano come avviene l'accesso al canale dove due stazioni cercano di trasmettere.

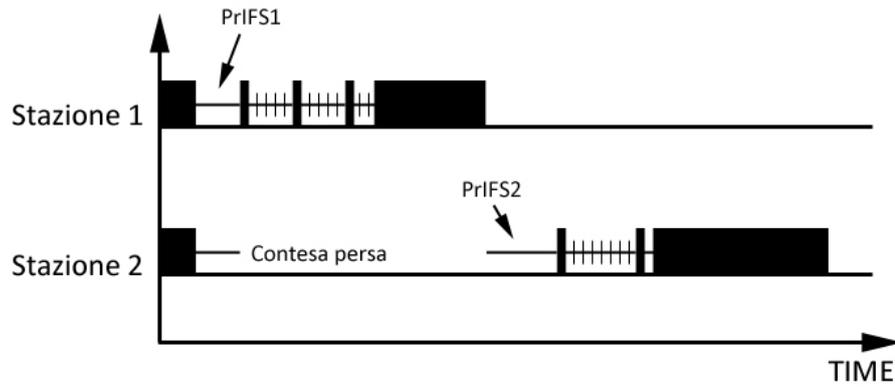


Figura 2.8: Caso in cui $PrIFS_1 < PrIFS_2$

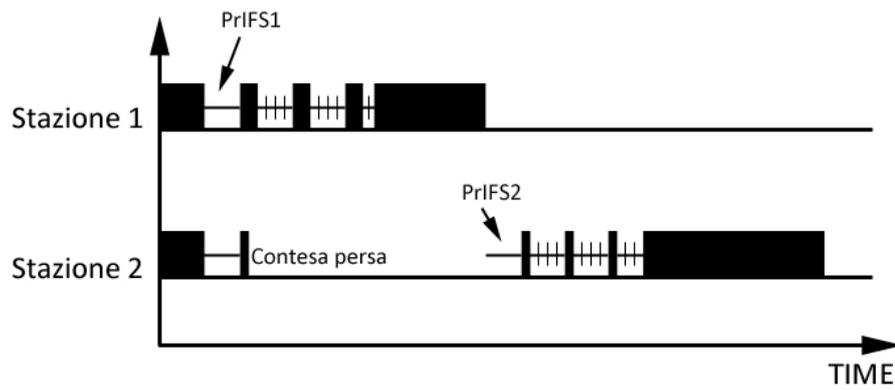


Figura 2.9: Caso in cui $PrIFS_1 = PrIFS_2$ ma $burst_len1 > burst_len2$

2.2.5 Accesso al canale con priorità dinamica

Viene definito un parametro denominato `sub_priorityPERNO` che indica la `sub_priority` che viene assegnata ad un flusso quando viene inviato correttamente un pacchetto.

Come già detto, ogni flusso è caratterizzato anche da una `sub_priority` e da un `burst_len` che ne definiscono la priorità dinamica per l'accesso al canale. Quando una stazione deve gestire un nuovo flusso allora questi parametri vengono definiti in base alla priorità del flusso e precisamente nel seguente modo:

```
sub_priority = (LV_PRIORITY * Pr) + (LV_PRIORITY / 2) + MAX_SUBPRIORITY
burst_len = 1
sub_priority(PERNO) = sub_priority
```

Prima di definire come vari la priorità dinamica, bisogna introdurre il concetto di superframe. Il tempo viene suddiviso in superframe, il quale viene a sua volta suddiviso in due frame: *FRAME_QOS* e *FRAME_CB*. La differenza principale tra i due frame è quella che durante il *FRAME_QOS* la `sub_priority` dei flussi non può superare il limite definito da $MAX_SUBPRIORITY + (LV_PRIORITY * Pr)$, mentre durante il *FRAME_CB* il limite è solamente *MAX_SUBPRIORITY*.

Perdita della contesa

Quando un flusso perde la contesa (cioè riceve un pacchetto MAC mentre cerca di accedere al canale) aumenta la sua priorità dinamica nel seguente modo:

```
se (frame == FRAME_CB){
  se (sub_priority > MAX_SUBPRIORITY)
    allora sub_priority = sub_priority - 1;
  altrimenti burst_len = burst_len + 1;
}altrimenti{
  se (sub_priority > (MAX_SUBPRIORITY + (LV_PRIORITY * Pr))){
    se (priorityRcv == Pr)
```

```

    allora sub_priority = sub_priority - 1;
altrimenti se (priorityRcv > Pr)
    allora sub_priority = (sub_priority / 2) + 0.5;
}
altrimenti
    burst_len = burst_len + 1;
}

```

Dove *priorityRcv* è la priorità del pacchetto ricevuto. Quindi nel caso di un FRAME_QOS se un flusso perde la contesa contro uno di priorità più bassa, allora dimezza il suo valore di *sub_priority* (quindi aumentandola più velocemente). Questo potrebbe accadere nel caso in cui il *sub_priority*_{PERNO} di un flusso diventi alto a causa della presenza di più flussi della stessa priorità che ne fanno aumentare il valore; se improvvisamente questi flussi smettessero di trasmettere, il flusso in questione si troverebbe con una priorità dinamica bassa (quindi un valore di *sub_priority* alto) che gli farebbe perdere la contesa anche contro i flussi con priorità originaria minore anche se ci si trovasse nel FRAME_QOS.

Questa funzione viene effettuata anche dagli altri flussi locali della stazione quando il pacchetto viene trasmesso con successo.

Collisione

Quando una stazione ipotizza che c'è stata una collisione (quando invia un pacchetto ma non riceve una risposta) allora aumenta la sua priorità con il seguente algoritmo (molto simile al precedente) con una probabilità del 50%.

```

se ((RANDOM % 100) < 50){
    se (frame == FRAME_CB){
        se (sub_priority > MAX_SUBPRIORITY)
            allora sub_priority = sub_priority - 1;
        altrimenti burst_len = burst_len + 1;
    }altrimenti{
        se (sub_priority > (MAX_SUBPRIORITY + (LV_PRIORITY * Pr)))

```

```

        allora sub_priority = sub_priority - 1;
    altrimenti
        burst_len = burst_len + 1;
    }
}

```

Comunque il protocollo continua a comportarsi come l'802.11 in caso di collisione, cioè duplicando la finestra di contesa dove poi verrà scelto il nuovo backoff.

Vincita della contesa

Quando un flusso vince la contesa e riesce ad inviare il suo pacchetto modifica i suoi parametri prendendo in considerazione la `sub_priority` e il `burst_len` con i quali è riuscito ad inviare (`sub_priorityLAST` e `burst_lenLAST`). Il metodo è il seguente:

```

mPerno = meanPerno(Pr, frame);
se (frame == FRAME_CB){
    se (burst_lenLAST > 1) allora
        sub_priorityPERNO = sub_priorityPERNO + (burst_lenLAST / 2);
    altrimenti se (sub_priorityLAST > MAX_SUBPRIORITY) allora
        sub_priorityPERNO = sub_priorityPERNO -
            ((sub_priorityLAST - MAX_SUBPRIORITY) / 4);
}altrimenti{
    se (burst_lenLAST > 1) allora
        sub_priorityPERNO = sub_priorityPERNO + (burst_lenLAST / 2);
    altrimenti se
        (sub_priorityLAST > (MAX_SUBPRIORITY + (LV_PRIORITY * Pr))) allora
        sub_priorityPERNO = sub_priorityPERNO - ( (sub_priorityLAST -
            (MAX_SUBPRIORITY + (LV_PRIORITY * Pr)) ) / 4);
}

se ( ((RANDOM % 100) < meanPernoProbability) &&
    (mPerno >= sub_priorityPERNO)
)
    sub_priority = mPerno;
altrimenti

```

```

sub_priority = ( (WEIGHT_PERNO_CALC * sub_priorityPERNO) +
  (WEIGHT_PERNO_MEAN * mPerno) )
/ (WEIGHT_PERNO_CALC + WEIGHT_PERNO_MEAN);

sub_priorityPERNO = sub_priority;
burst_len = 1;

```

Dove *meanPerno* calcola la media dei perni utilizzati dagli altri flussi in base al frame attuale e la priorità del flusso; questo perché in caso di FRAME_QOS i flussi con cui si contende, sono solo quelli con priorità minore o uguale; il calcolo viene effettuato prendendo in considerazione tutti i flussi ascoltati nell'arco di $N_SUPERFRAME * superframeTime$. *meanPernoProbability* indica la percentuale con cui si sceglie se settare il perno al valore calcolato con la funzione appena descritta; questo serve a far sì che le stazioni cerchino di settare il loro valore di perno allo stesso valore. Le due costanti WEIGHT_PERNO_CALC e WEIGHT_PERNO_MEAN indicano il peso da dare ai due valori calcolati per la scelta del perno. Come considerazione indicherei un peso minore a WEIGHT_PERNO_MEAN perché si basa su un indice solo indicativo che ha l'utilità di stabilizzare il perno.

La causa per cui viene fatta la divisione per 2 nel caso il `burst_lenLAST` sia maggiore di 1, mentre nell'altro caso viene fatta la divisione per 4, è che avere un burst lungo è dannoso visto che viene sprecato tempo ed energia; quindi si cerca di rimediare aumentando più velocemente il perno.

La procedura appena descritta serve a far adattare il flusso in modo tale che trasmetti con `burst_len` a 1 e con `sub_priority` a `MAX_SUBPRIORITY` nel caso del FRAME_CB o a $(MAX_SUBPRIORITY + (LV_PRIORITY * Pr))$ nel caso del FRAME_QOS.

Questo metodo di accesso al canale cerca di far evitare le collisioni introducendo queste proprietà del flusso che ne caratterizzano l'accesso; in questo modo è possibile diminuire la finestra minima dimezzandola dopo un numero di accessi consecutivi pari a `NUM_SUCCESI_CONSEC` senza la presenza di collisioni.

2.2.6 Superframe

Come è stato accennato prima, il tempo viene suddiviso in superframe di lunghezza *superframeTime*. Questi sono a loro volta suddivisi in due frame: *FRAME_QOS* e *FRAME_CB*, in cui il *FRAME_QOS* ha dimensioni $(\alpha * \text{superframeTime})$ mentre il *FRAME_CB* ha dimensioni $((1 - \alpha) * \text{superframeTime})$ (figura 2.10). Nel frame contention based tutti i flussi hanno la possibilità di far raggiungere la loro *sub_priority* al valore massimo (*MAX_SUBPRIORITY*), mentre nel frame QoS le *sub_priority* sono limitate dalla priorità del flusso e più precisamente al valore $(\text{MAX_SUBPRIORITY} + (\text{LV_PRIORITY} * \text{priority}))$.

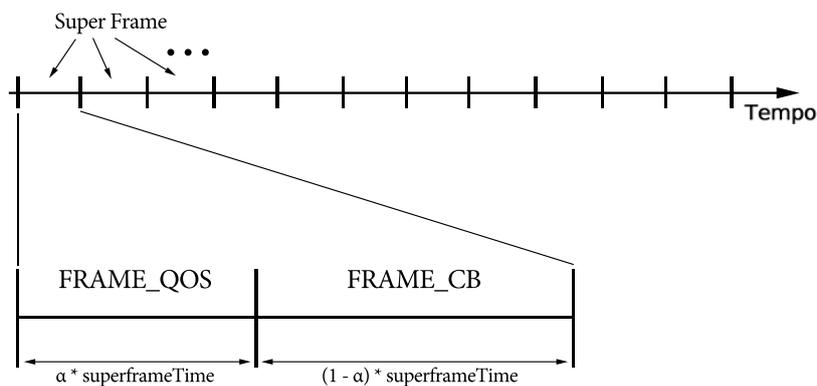


Figura 2.10: Struttura di un superframe

Il problema dell'utilizzo di un superframe è la sincronizzazione che ci deve essere tra le stazioni visto che ci troviamo in un ambiente ad-hoc dove non c'è una stazione centrale che decide quando comincia e finisce il superframe. Per effettuare la sincronizzazione sono stati inseriti due campi nell'header dei pacchetti MAC: *qos_state* e *time_left_qos*. Questi due parametri vengono calcolati ed inseriti nel pacchetto appena prima della trasmissione del pacchetto stesso sul canale. Questi parametri vengono calcolati prendendo in considerazione il fatto che il pacchetto deve essere ancora trasmesso e quindi questi due parametri sono calcolati nel seguente modo:

```

fine_trasmissione = now + txtTime(pacchetto);
se (frame == FRAME_CB){
  se (SuperframeEND_CB < fine_trasmissione){
    qos_state = FRAME_QOS;
    Time_left_superframe = SuperframeEND_CB +
                          superframeTime - fine_trasmissione;
  }
  altrimenti{
    qos_state = FRAME_CB;
    Time_left_superframe = SuperframeEND_CB - fine_trasmissione;
  }
}altrimenti{
  se (SuperframeEND_QOS < fine_trasmissione){
    qos_state = FRAME_CB;
    Time_left_superframe = SuperframeEND_CB - fine_trasmissione;
  }
  altrimenti{
    qos_state = FRAME_QOS;
    Time_left_superframes = SuperframeEND_CB - fine_trasmissione;
  }
}
sub_priority = sub_priorityPERNO;
burst_len = 1

```

Dove *txtTime* calcola il tempo necessario per trasmettere il pacchetto sul canale, *now* è l'ora locale della stazione mentre i valori di *SuperframeEND_CB* e *SuperframeEND_QOS* sono rispettivamente l'ora locale in cui il timer del superframe scadrà e l'ora locale in cui scadrà il FRAME_QOS.

Dal lato ricevente la stazione aggiorna il suo timer per la scadenza del superframe nel seguente modo:

```

se (frameRcv == frame){
  se (frame == FRAME_CB){
    se ( (SuperframeRcvEND_CB < SuperframeEND_CB) &&
        ((SuperframeEND_CB - SuperframeRcvEND_CB) < FRAME_THERESHOLD)
    ){
      changeTimerCB (SuperframeRcvEND_CB);
    }
  }
}

```

```

}
altrimenti{
  se ( (SuperframeRcvEND_CB < SuperframeEND_CB) &&
      ((SuperframeEND_CB - SuperframeRcvEND_CB) < FRAME_THERESHOLD)
  ){
    changeTimerCB (SuperframeRcvEND_CB);
    SuperframeRcvEND_QOS = SuperframeRcvEND_CB -
                          ((1 - alpha) * superframeTime);
    se ((SuperframeEND_QOS - now) > FRAME_THERESHOLD){
      changeTimerQOS (SuperframeRcvEND_QOS);
    }
  }
}
}
}
}

```

Dove $SuperframeRcvEND_CB$ è il parametro “Time left superframe” contenuto nell’header MAC del pacchetto ricevuto, sommato al valore now. $FRAME_THERESHOLD$ è due volte il valore di $MAX_PropagationDelay$; le due funzioni `changeTimerQOS` e `changeTimerCB` cambiano rispettivamente il timer del $FRAME_QOS$ e del $FRAME_CB$; α e il parametro α sopra descritto.

Inoltre appena viene creata una nuova stazione, questa al suo avvio, è in stato *FRAME_WAITING*. Questo stato viene modificato alla prima ricezione di un pacchetto dalla rete con i valori presenti nel pacchetto; oppure appena riceve un pacchetto dai livelli superiori avvia indipendentemente il timer locale.

Come descritto nei precedenti paragrafi, i flussi hanno dei parametri di `sub_priority`, `burst_len` e `sub_priority_PERNO` che si modificano in funzione del frame in cui si trovano. Durante il cambio di frame questi tre parametri avrebbero dei valori obsoleti per il frame in cui si trovano; per evitare questo problema, ad ogni cambio di frame questi parametri vengono congelati momentaneamente per poi essere ripristinati al successivo cambio di frame; quindi ogni flusso memorizza i parametri appartenenti al $FRAME_CB$ che vengono salvati al passaggio al $FRAME_QOS$ per venire poi ripristinati al

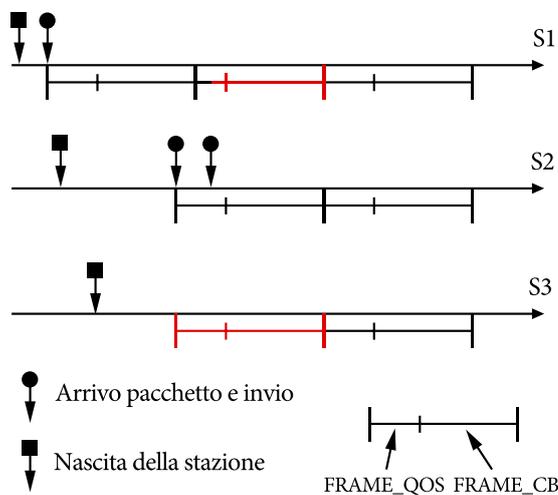


Figura 2.11: Esempio di sincronizzazione in cui si nota come S2 al primo invio sincronizza S3, mentre al secondo invio sincronizza S1

nuovo passaggio al FRAME_CB; vengono salvati nello stesso modo nel caso del FRAME_QOS.

Con questo nuovo accorgimento sorge il problema della trasmissione sul fronte del cambio del frame, quando cioè un flusso vince la contesa in un frame, ma finisce la trasmissione del pacchetto nel frame successivo. Dal lato ricevente non è possibile essere certi di questo problema, quindi il pacchetto ricevuto viene considerato appartenere al frame indicato nell'header del pacchetto.

Il lato trasmittente gestisce la trasmissione sul fronte come se fosse avvenuta una sconfitta; questo perché se modificasse i valori di `sub_priority`, `burst_len` e `sub_priority_PERNO` come se avesse vinto nel nuovo frame sbaglierebbe perché in effetti ha vinto la contesa nel frame appena passato; se invece modificasse i suoi valori congelati del frame precedente, modificando quindi i parametri con i quali ha vinto la contesa, ci sarebbe una discordanza con le altre stazioni, le quali hanno gestito questo successo trasmissivo come una sconfitta nell'accesso al canale nel nuovo frame. Questo procedimento sembrerebbe

discordante col principio del protocollo per cui un flusso aumenterebbe la propria priorità dinamica solo in caso di perdita della contesa; in questo caso particolare però un flusso incrementa la propria priorità dinamica anche nel caso di vittoria; si noti però che i parametri che vengono modificati non sono quelli con cui ha vinto la contesa, ma quelli che sono stati “scongelati” al passaggio nel nuovo frame.

Ad ogni cambio di frame, se una stazione è ancora in fase di contesa, allora reinserisce il pacchetto che stava gestendo in coda e rieffettua la deque dallo scheduler per prendere un nuovo pacchetto. Questa procedura viene effettuata perché al cambio del frame le caratteristiche dei flussi vengono cambiate e quindi bisogna essere sicuri che si stia gestendo il pacchetto con la maggiore priorità dinamica e con le caratteristiche di accesso giuste.

2.2.7 Flussi e sottoflussi

Per far aumentare la possibilità di accesso al canale da parte dei flussi con maggiore priorità anche durante il FRAME_CB è stato introdotto il concetto di sottoflusso. Ogni flusso viene suddiviso in più sottoflussi in base alla priorità dello stesso. Il calcolo del numero dei sottoflussi è il seguente: $n = (PR_MIN - Pr) + 1$. Essendo la contesa flow-based si ha un aumento dell'accesso al canale con la presenza di più sottoflussi.

Ad esempio se un flusso con priorità 0 viene suddiviso in 4 sottoflussi e uno di priorità 2 in 2, si avrà che il flusso con priorità 0 accederà al canale il doppio rispetto al flusso con priorità 2.

Il campo dell'header *subflow* consente l'uso di 8 sottoflussi per ogni flusso. Un altro possibile calcolo del numero dei sottoflussi potrebbe essere il seguente: $n = 2^{(PR_MIN - Pr)}$. Questo calcolo farebbe avvantaggiare maggiormente i flussi di priorità alta durante il FRAME_CB in quanto la disparità in numero di flussi sarebbe più elevata.

2.3 Considerazioni

Tramite il meccanismo delle priorità dinamiche l'accesso al canale diventa fairness tra i pacchetti con la stessa priorità nel caso ci si trovi nel FRAME.QOS, mentre tra tutti i flussi nel caso del FRAME.CB. Questo avviene perché quando un flusso perde la contesa, questo aumenta la propria priorità dinamica di accesso, mentre il flusso che ha vinto la contesa ritorna ad una priorità dinamica minore.

Oltre alla proprietà di fairness viene a crearsi un accesso al canale di tipo round-robin: infatti il calcolo dei valori dei perni, che definiscono la priorità dinamica iniziale dalla quale partire dopo ogni successo, fa sì che le stazioni scelgano all'incirca tutte lo stesso perno in modo consecutivo, così da creare una coda di stazioni aventi priorità dinamiche crescenti che permetteranno l'accesso sequenziale delle stazioni.

	T1	T2	T3	T4	T5
Flusso1	3	2	1	0	3
Flusso1	2	1	0	3	2
Flusso1	1	0	3	2	1
Flusso1	0	3	2	1	0

Tabella 2.1: Esempio evoluzione dei flussi

In tabella 2.1 c'è un esempio semplificato in cui ci sono 4 flussi che vogliono accedere al canale; i valori all'interno delle celle indicano il valore di $sub_priority$ che i flussi hanno per accedere al canale; ogni flusso si ipotizza abbia il suo valore di $sub_priority_{PERNO}$ uguale a 3. Quando una stazione con $sub_priority$ 0 trasmette (perché riesce ad acquisire il canale prima degli altri), resetta il suo valore a $sub_priority_{PERNO}$, mentre le altre diminuiscono il loro. Con questo meccanismo si viene a creare un accesso round-robin dove a turno si riesce a catturare il canale per trasmettere.

I parametri che caratterizzano principalmente il relativo funzionamento del protocollo PAB sono i seguenti:

- α è un parametro importantissimo che decide la grandezza dei due frame: FRAME_QOS e FRAME_CB.
- *WEIGHT_PERNO_CALC* e *WEIGHT_PERNO_MEAN* decidono quanto un flusso reagisce alle modifiche della rete: un valore più alto di *WEIGHT_PERNO_CALC* indica un flusso più reattivo e indipendente alla modifica del numero dei flussi presenti nel sistema; aumentando invece il valore di *WEIGHT_PERNO_MEAN*, d'altro canto, si cerca di far stabilizzare tutti i flussi allo stesso perno.
- *superframeTime* che determina la lunghezza di un super frame. Un valore troppo alto determinerebbe un ritardo troppo elevato per le stazioni di alto livello che resterebbero per troppo tempo fuori dal FRAME_QOS; mentre un valore troppo basso destabilizzerebbe troppo il sistema a causa del continuo cambio di frame.

Capitolo 3

Analisi dei due protocolli

Di seguito verranno analizzate le prestazioni dei due modelli sotto le medesime condizioni e verranno confrontate le prestazioni al fine di valutarne l'affidabilità che forniscono al supporto QoS.

Dapprima verranno analizzate le prestazioni generali come goodput e MAC delay dei due protocolli al variare del numero dei flussi presenti nel sistema con una parità di presenza di tutte le priorità. Poi verranno analizzate le proprietà QoS del servizio che viene offerto dai due protocolli mostrando la garanzia di trasmissione che viene offerta alle priorità alte rispetto a quelle più basse. Verrà analizzata anche la problematica della starvation che può sorgere in caso di prevalenza di flussi di priorità alta rispetto a quelli di priorità bassa. Infine sarà mostrata la caratteristica di fairness dei due protocolli in presenza di flussi della stessa priorità.

Le caratteristiche che verranno studiate sono il *goodput* ed il *delay link-to-link*. Il goodput viene inteso come quantità di bit dati al secondo che una stazione riesce a trasmettere. Il mac delay link-to-link è il tempo necessario al livello MAC per inviare il pacchetto una volta preso dalla coda.

Negli scenari che verranno proposti tutte le stazioni saranno full connected in cui tutti sentono tutto. Inoltre ogni stazione trasmetterà un solo flusso e i destinatari di questi flussi avranno soltanto il compito di rispondere. Quindi per ogni flusso presente nel sistema saranno presenti 2 stazioni: una

trasmittente e una ricevente.

I parametri utilizzati per le simulazioni del protocollo 802.11e sono quelli descritti in appendice A.2. Per quanto riguarda il protocollo PAB i valori sono quelli presentati in A.1; alcuni dei parametri però saranno mostrati di volta in volta ad ogni descrizione di scenario.

3.1 Prestazioni generali

3.1.1 Scenario

Lo scenario proposto per questo studio è composto da un numero variabile di flussi con un data rate di creazione di pacchetti tale da avere sempre a disposizione pacchetti da inviare, cioè la generazione dei pacchetti da parte delle applicazioni supera la possibilità di smaltimento da parte del MAC. Nel sistema sono presenti n flussi di livello 0, n di livello 1, n di livello 2 e n di livello 3, con n variabile. Queste simulazioni mostrano come si comportano i due protocolli in presenza di un numero elevato di stazioni con un'equità di presenze delle varie priorità.

I parametri principali del sistema sono quelli descritti nella tabella 3.1.

α	0.45
meanPernoProbability	5
WEIGHT_PERNO_CALC	65
WEIGHT_PERNO_MEAN	35
DataRate	1Mbps
superframeTime	1 sec
n° di classi di priorità presenti	4
n° flussi di priorità 0, 1, 2 e 3	variabile ma in modo omogeneo

Tabella 3.1: Parametri utilizzati per le simulazioni

3.1.2 Goodput

Come si vede dalle simulazioni il protocollo PAB riesce a sfruttare meglio il canale raggiungendo un goodput generale quasi quattro volte maggiore rispetto all'802.11e (fig. 3.1(a)). Questo risultato si ottiene grazie alla poca presenza di collisioni che con il nuovo meccanismo di accesso vengono drasticamente diminuite.

Spostandoci sui grafici specifici per le diverse priorità (fig. 3.1(b), 3.1(c), 3.1(d) e 3.1(e)) si nota come il protocollo PAB garantisce una resa migliore del sistema anche in presenza di numerosi flussi trasmissivi.

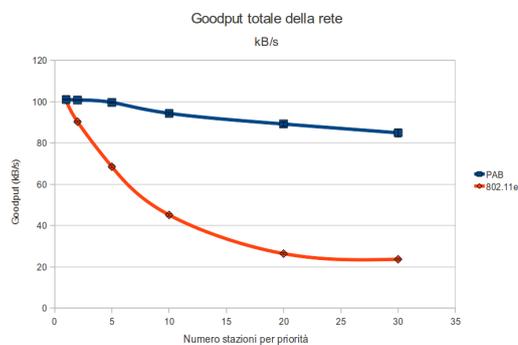
3.1.3 MAC delay link-to-link

I valori del Mac delay link-to-link sono visibili nelle figure 3.2(a), 3.2(b), 3.2(c) e 3.2(d).

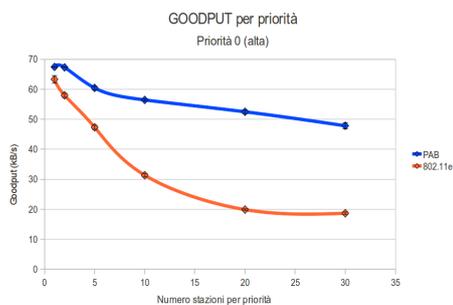
I grafici mostrano come l'802.11e è più performante per quanto riguarda il mac delay offerto. Questi valori sono da leggere però tenendo conto anche dei valori di goodput riscontrati nel paragrafo precedente; infatti i pacchetti in 802.11e attraversano velocemente la coda perchè a causa delle continue collisioni molti dei pacchetti vengono droppati, e quindi quelli più fortunati che riescono ad essere trasmessi ce la fanno con un mac delay basso.

Per quanto riguarda invece i valori del protocollo PAB, si nota la quasi linearità della crescita dei valori; in questo caso infatti non viene droppato nessun pacchetto a causa di collisioni successive e quindi l'attesa è dovuta semplicemente al numero più elevato di flussi presenti nel sistema; infatti ricordando che l'accesso al canale che si viene a creare è simile al round-robin, allora è normale che col crescere del numero di flussi presenti nel sistema aumenti linearmente il delay percepito dai vari flussi.

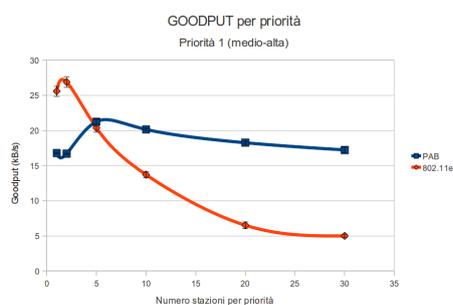
La differenza tra le varie priorità che si nota principalmente è quella tra la priorità 0 e le altre. Questa differenza è dovuta al fatto che durante il FRAME_QOS l'unica priorità a trasmettere è quella alta. In figura 3.3 viene visualizzato meglio la differenziazione del servizio che viene effettuato dai due



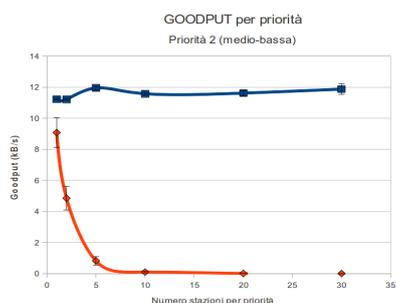
(a) Goodput totale



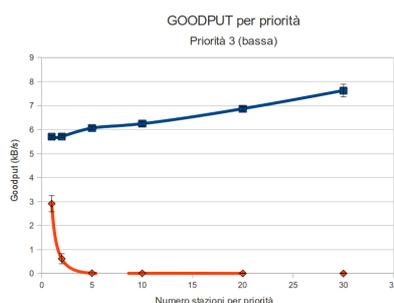
(b) Goodput totale riguardante la priorità 0 (alta)



(c) Goodput totale riguardante la priorità 1 (medio-alta)

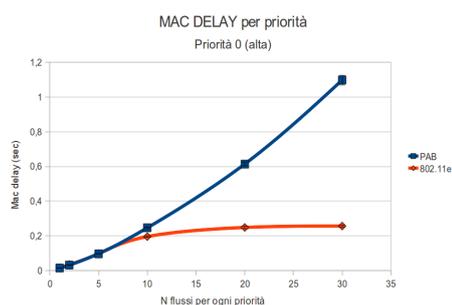


(d) Goodput totale riguardante la priorità 2 (medio-bassa)

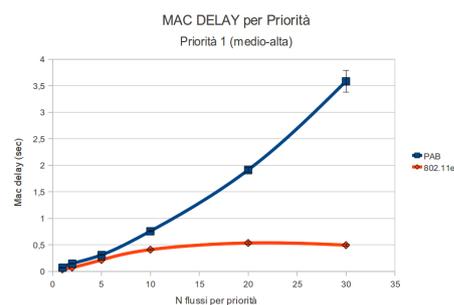


(e) Goodput totale riguardante la priorità 3 (bassa)

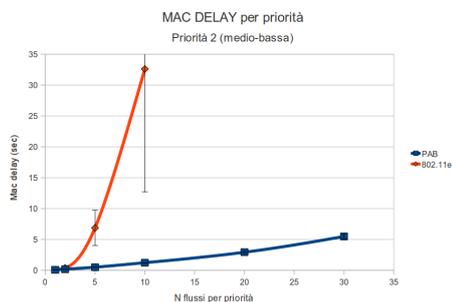
Figura 3.1: Confronto delle prestazioni (Goodput) dei due protocolli al variare del numero delle stazioni. Il numero di flussi presenti per tutte le priorità è uguale



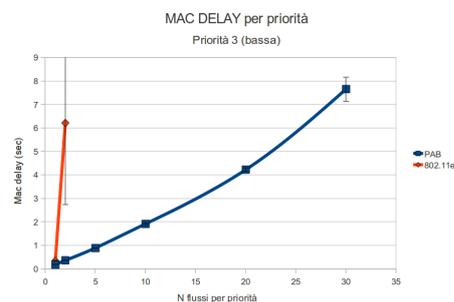
(a) Delay medio riguardante la priorità 0 (alta)



(b) Delay medio riguardante la priorità 1 (medio-alta)



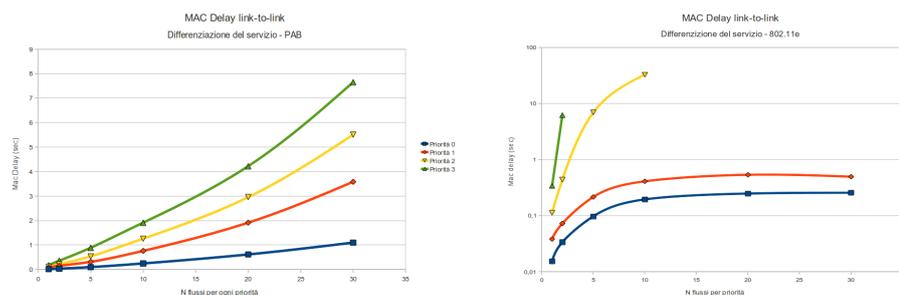
(c) Delay medio riguardante la priorità 2 (medio-bassa)



(d) Delay medio riguardante la priorità 3 (bassa)

Figura 3.2: Confronto delle prestazioni (Mac Delay link-to-link) dei due protocolli al variare del numero delle stazioni. Il numero di flussi presenti per tutte le priorità è uguale

protocolli. Si nota come in 802.11e i flussi di livello basso non sopravvivono all'elevata presenza di flussi di alto livello; inoltre per l'asse delle ordinate si è dovuta usare una scala logaritmica per far denotare le differenze dei valori.



(a) Differenziazione del servizio per le diverse priorità in PAB

(b) Differenziazione del servizio per le diverse priorità in 802.11e

Figura 3.3: Confronto delle prestazioni (Mac Delay link-to-link) dei due protocolli.

3.2 Proprietà QoS

3.2.1 Scenario

Lo scenario utilizzato per mostrare la qualità del servizio fornita dai due protocolli è quello in cui c'è una stazione di livello massimo (priorità 0) e un numero variabile di stazioni di un altro livello più basso. Quindi ci saranno 3 diversi scenari: uno con un numero variabile di stazioni di livello 1, un secondo con un numero variabile di stazioni di livello 2 e l'ultimo con un numero variabile di stazioni di livello 3. Verranno mostrati i grafici per goodput e mac delay link-to-link confrontando la priorità massima con tutte le altre priorità. Lo scopo di questa simulazione è testare la QoS del protocollo; infatti vengono inserite un numero crescente di stazioni che non dovrebbero intaccare, almeno in parte, le prestazioni del flusso di alta priorità.

In tabella 3.2 sono mostrati i parametri principali usati.

α	0.45
meanPernoProbability	5
WEIGHT_PERNO_CALC	65
WEIGHT_PERNO_MEAN	35
DataRate	1Mbps
superframeTime	1 sec
n° di classi di priorità presenti	2
n° flussi di priorità 0	1
n° flussi di priorità 1, 2 e 3	variabile

Tabella 3.2: Parametri utilizzati per lo studio della QoS

3.2.2 Priorità bassa

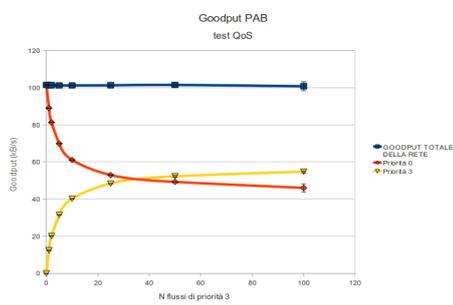
In figura 3.4 sono mostrati i risultati delle simulazioni dove il flusso di priorità alta viene testato con i flussi di priorità bassa. Come si vede dai grafici l'802.11e riesce a garantire una più ampia differenza tra i due tipi di flussi rispetto al PAB.

Per quanto riguarda il grafico 3.4(d) i valori del mac delay per l'802.11e sono così instabili perchè il flusso di priorità bassa riesce ad inviare pochissimi pacchetti rispetto al flusso di priorità alta.

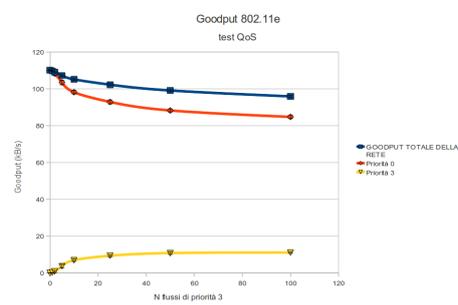
3.2.3 Priorità medio-bassa

In questo caso i due protocolli sono simili, come si vede dai grafici in figura 3.5.

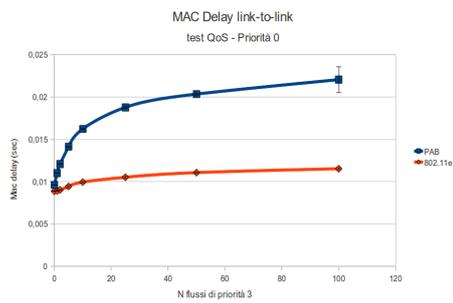
I valori del goodput del PAB sono migliori rispetto l'802.11e per la poca presenza di collisioni che permette un migliore sfruttamento del canale.



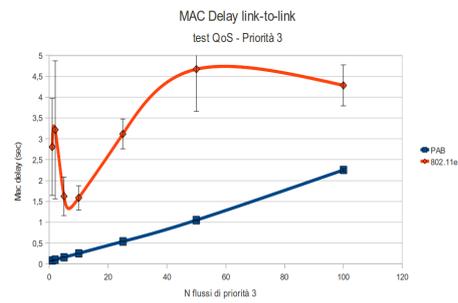
(a) Goodput PAB



(b) Goodput 802.11e

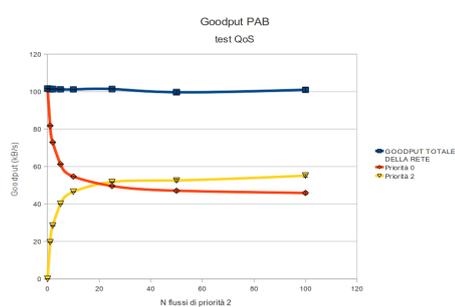


(c) MAC delay per la priorità 0

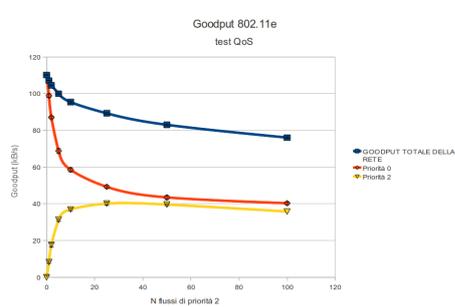


(d) MAC delay per la priorità 3

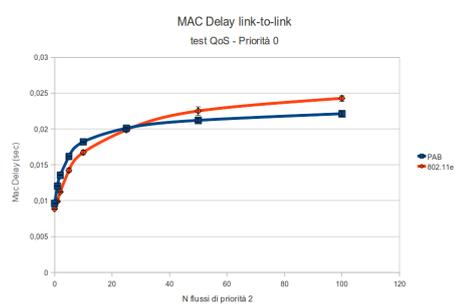
Figura 3.4: Confronto delle prestazioni dei due protocolli al variare del numero delle stazioni aventi flussi di priorità 3



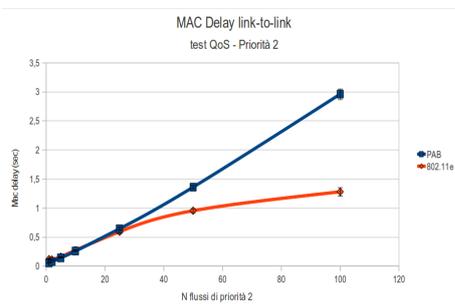
(a) Goodput PAB



(b) Goodput 802.11e



(c) MAC delay per la priorità 0



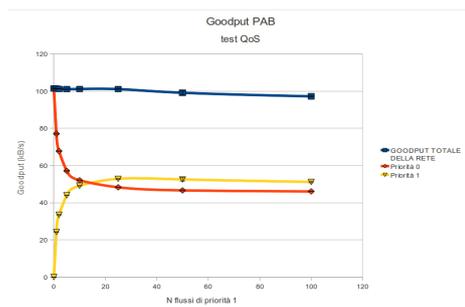
(d) MAC delay per la priorità 2

Figura 3.5: Confronto delle prestazioni dei due protocolli al variare del numero delle stazioni aventi flussi di priorità 2

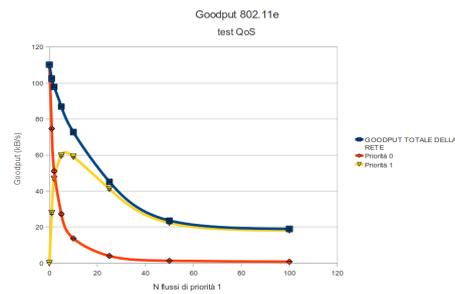
3.2.4 Priorità medio-alta

In questo caso invece il protocollo 802.11e non garantisce la QoS (figura 3.6(b)). Infatti con l'aumentare di flussi di priorità 1, il goodput del flusso di priorità alta scende fino ad arrivare quasi a zero. L'aumento del numero di flussi provoca anche una drastica discesa del goodput totale e questo a causa di un numero molto elevato di collisioni visto il basso valore di CW_{MAX} della priorità 1.

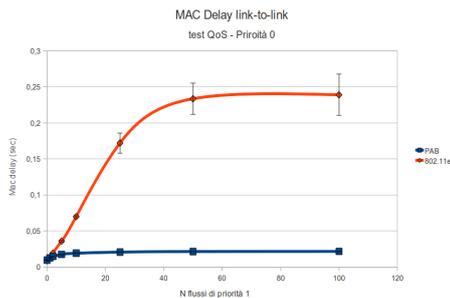
Il PAB invece si comporta bene perchè garantisce una soglia minima di trasmissione ai flussi di priorità alta.



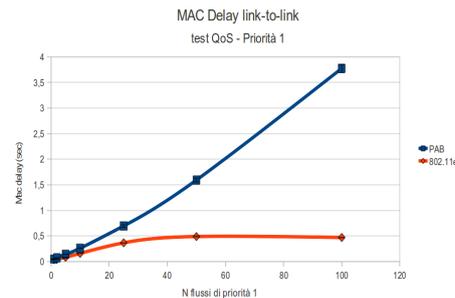
(a) Goodput PAB



(b) Goodput 802.11e



(c) MAC delay per la priorità 0



(d) MAC delay per la priorità 1

Figura 3.6: Confronto delle prestazioni dei due protocolli al variare del numero delle stazioni aventi flussi di priorità 1

3.2.5 Considerazioni

Come abbiamo visto il protocollo 802.11e non garantisce la QoS nel caso in cui pochi flussi di priorità alta si trovano a scontrarsi con tanti flussi di priorità poco più bassa (3.2.4). Questo accade perchè i valori di AIFS, CW_{MIN} e CW_{MAX} sono molto simili tra loro e quindi il rischio di collisioni è molto alto. Per quanto riguarda invece le priorità di livello più basso, i valori sono molto distanti e quindi garantiscono una più alta qualità del servizio, dando più possibilità trasmissiva ai flussi di livello alto.

Per quanto riguarda il protocollo PAB vediamo che per il valore del goodput il comportamento risulta molto simile tra i vari esempi riportati nei precedenti paragrafi (figure 3.6(a), 3.5(a) e 3.4(a)). Infatti il limite a cui tende la curva del valore del goodput della priorità 0 è dovuto al valore di α , il quale è responsabile della grandezza del FRAME_QOS; come vediamo infatti in figura 3.7 il goodput del flusso di priorità 0 aumenta con l'aumentare del valore di α . Infatti durante questo frame la presenza di flussi di alta priorità

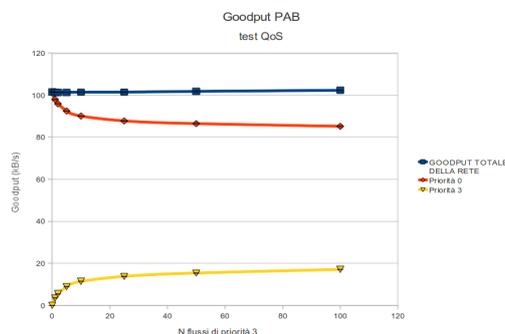


Figura 3.7: Goodput PAB nel caso di valore $\alpha=0.85$

inibisce completamente i flussi di più bassa priorità perché questi ultimi non hanno la possibilità di superare il confine definito dalla loro priorità (2.2.5). Per lo stesso discorso, anche il delay ha un limite a cui tende la curva del flusso di priorità 0; questo limite è raggiunto quando il flusso di alta priorità riesce a trasmettere solamente durante il FRAME_QOS. Le altre priorità

si comportano come nella sezione precedente e cioè con un mac delay che cresce linearmente in base al numero di flussi presente nel sistema a causa dell'accesso round-robin.

3.3 Problema della starvation

3.3.1 Scenario

Lo scenario utilizzato per queste simulazioni è quello in cui è presente una stazione di livello basso (priorità 3) e un numero variabile di stazioni di livello superiore, a partire dal livello 2 fino al livello 0; in totale quindi ci saranno 3 scenari dove nel sistema saranno presenti due soli tipi di flussi. Saranno mostrati i grafici del goodput per mostrare per i due protocolli se si soffre della problematica della starvation in cui una stazione di priorità bassa non riesce mai ad inviare dati.

In tabella 3.3 sono mostrati i parametri principali usati dal protocollo PAB.

α	0.45
meanPernoProbability	5
WEIGHT_PERNO_CALC	65
WEIGHT_PERNO_MEAN	35
DataRate	1Mbps
superframeTime	1 sec
n° di classi di priorità presenti	2
n° flussi di priorità 3	1
n° flussi di priorità 0, 1 e 2	variabile

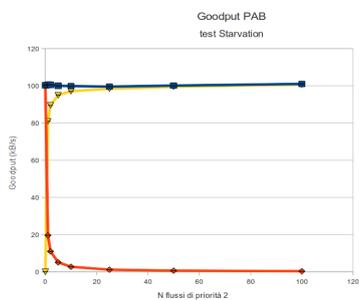
Tabella 3.3: Parametri utilizzati per lo studio della starvation

3.3.2 PAB

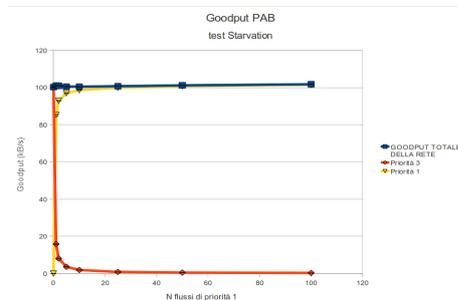
Nei grafici 3.8(a), 3.8(b) e 3.8(c) sono mostrati i valori di goodput al variare del numero di stazioni che inviano dati di priorità 2, 1 e 0 rispettivamente.

La prima osservazione da fare è che viene rispettata la proprietà di QoS per cui l'avvento delle stazioni con più alta priorità fa precipitare i valori del goodput della stazione che invia a priorità bassa.

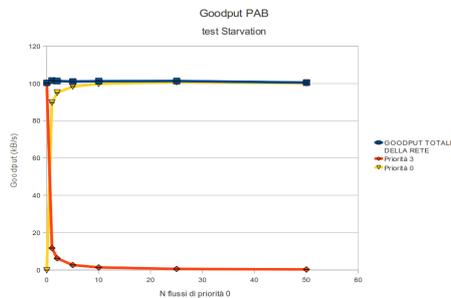
Per quanto riguarda la starvation si nota come in tutti e tre i casi, all'au-



(a) Goodput PAB con un numero variabile di stazioni di livello 2



(b) Goodput PAB con un numero variabile di stazioni di livello 1



(c) Goodput PAB con un numero variabile di stazioni di livello 0

Figura 3.8: Confronto delle prestazioni del PAB al variare del numero delle stazioni di livello superiore al 3

mentare del numero delle stazioni, il goodput della stazione a priorità bassa non decade mai a zero. In tabella 3.4 sono mostrati i valori del goodput presenti nei grafici ma che sono poco visibili.

N stazioni	Priorità 2	Priorità 1	Priorità 0
0	100,2747138	100,277969	100,2747138
1	19,6123893	15,8068684	11,7810743
2	11,0130338	8,0327278	6,3189322
5	5,1081836	3,5724804	2,728392
10	2,7316472	1,87265	1,4747594
25	1,1829232	0,8152734	0,6515694
50	0,6438805	0,4661652	0,3757748
100	0,3593556	0,2756644	0,198698

Tabella 3.4: Valori del goodput PAB della stazione a priorità 3

Quindi data la natura del protocollo, una stazione, pur essendo di priorità inferiore, durante il FRAME_CB ha la certezza di inviare dopo un certo numero di sconfitte nella contesa grazie alla priorità dinamica che crea una forma di aging che assicura l'invio del pacchetto.

L'assenza di starvation si denota anche in tabella 3.5 dove è stato settato il valore di α a 0.85 e quindi il tempo disponibile per l'invio di dati con minore priorità diventa più ristretto; questo tempo resta comunque sufficiente per permettere alle classi meno privilegiate di inviare dati senza incorrere nel problema della starvation.

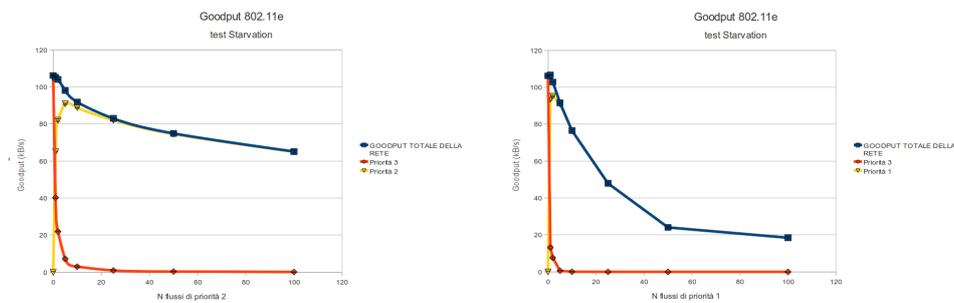
N stazioni di priorità 0	Priorità 3
0	98,9061053
1	3,4808228
2	1,8024266
5	0,8076063
10	0,47591
25	0,2690674
50	0,1961082
100	0,015918

Tabella 3.5: Valori del goodput PAB della stazione a priorità 3 in presenza di un numero crescente di stazioni di priorità 1 e con un valore di $\alpha=0.85$

3.3.3 802.11e

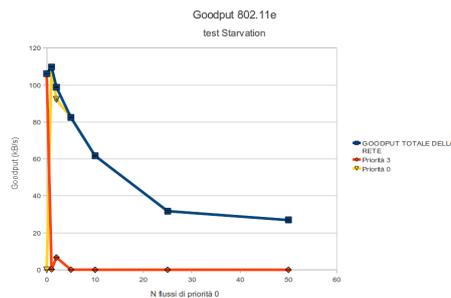
Per quanto riguarda invece il protocollo 802.11e i grafici sono mostrati nella figura 3.9(a), 3.9(b) e 3.9(c).

Anche in questo caso la proprietà di QoS viene rispettata in quanto i flussi di priorità alta vincono drasticamente contro il flusso di priorità bassa. La caduta del goodput generale, principalmente in presenza di priorità 0 e 1, è dovuta al numero molto elevato di collisioni che accadono a causa dei bassi valori di CW_{MIN} e CW_{MAX} che caratterizzano queste due priorità; questo fa anche notare come questo protocollo non sia in grado di supportare un numero elevato di stazioni di livello alto.



(a) Goodput PAB con un numero variabile di stazioni di livello 2

(b) Goodput PAB con un numero variabile di stazioni di livello 1



(c) Goodput PAB con un numero variabile di stazioni di livello 0

Figura 3.9: Confronto delle prestazioni del PAB al variare del numero delle stazioni di livello superiore al 3

In questo caso però sorge il problema della starvation (tabella 3.6). Come

si nota dai dati, un aumento delle stazioni di più alta priorità nega permanentemente l'accesso al canale alla stazione di priorità 3. Questo è visibile dai dati in corrispondenza delle colonne appartenenti ai flussi di priorità 1 e 0 in cui all'aumento del numero delle stazioni corrisponde un azzeramento del goodput della stazione a priorità bassa.

N stazioni	Priorità 2	Priorità 1	Priorità 0
0	106,1002607	106,1002607	106,1002607
1	40,2376303	13,1966146	0,2311197
2	21,9628907	7,6041667	6,673177
5	7,2070312	0,595703	0,0553384
10	3,0175782	0,0520832	0
25	0,9635418	0	0
50	0,3776041	0	0
100	0,1497394	0	0

Tabella 3.6: Valori del goodput 802.11e della stazione a priorità 3

3.4 Fairness

3.4.1 Scenario

In questa sezione verranno studiati i protocolli studiandone la fairness tramite l'indice di Jain [18]. Questo indice viene definito nel seguente modo:

$$f(x) = \frac{(\sum_{i=1}^n x_i)^2}{n \cdot \sum_{i=1}^n x_i^2} \quad x_i \geq 0 \quad (3.1)$$

dove n è il numero di stazioni che contengono per la risorsa e x_i è l'allocazione ricevuta dalla stazione i^{th} . Se tutte le stazioni ricevono la stessa quantità di risorsa, cioè tutti gli x_i sono uguali, allora l'indice varrà 1 e il sistema sarà al 100% equo. Con la crescita della disparità nell'acquisizione delle risorse questo valore tenderà a 0. Quindi se una risorsa di grandezza m fosse distribuita tra n contendenti e venissero scelti k di loro a cui distribuire

equamente m/k risorse, allora l'indice sarebbe k/n . Perciò se per esempio l'indice valesse 0.80 sarebbe come se l'80% dei contendenti avessero preso equamente la risorsa, mentre il restante 20% sarebbero rimasti senza.

Lo scenario proposto è quello in cui sono presenti un numero variabile di stazioni con la stessa priorità di accesso. Quindi verranno proposti 4 scenari, uno per ogni priorità, in cui la variabile è il numero delle stazioni.

In tabella 3.7 sono mostrati i parametri principali usati dal protocollo PAB.

α	0.45
meanPernoProbability	5
WEIGHT_PERNO_CALC	65
WEIGHT_PERNO_MEAN	35
DataRate	1Mbps
superframeTime	1 sec
n° di classi di priorità presenti	1
n° flussi di priorità 0, 1, 2 e 3	variabile

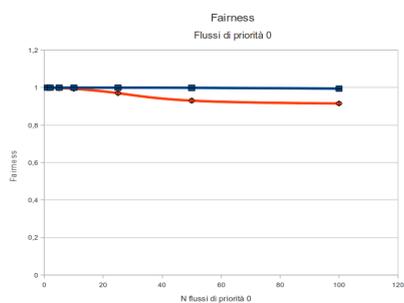
Tabella 3.7: Parametri utilizzati per lo studio della fairness

Dopodiché verrà utilizzato lo stesso scenario proposto in 3.1.1 per l'ultima simulazione.

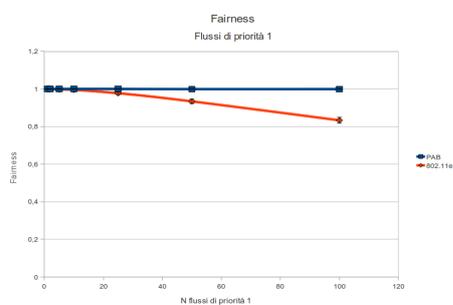
3.4.2 Risultati

Nei grafici di figura 3.10(a), 3.10(b), 3.10(c) e 3.10(d) viene mostrato il comportamento dei due protocolli con uno scenario in cui sono presenti nel sistema flussi di una sola priorità. Il protocollo PAB si comporta in maniera molto efficace e mantiene il valore dell'indice costantemente ad 1 nonostante l'aumento del numero delle stazioni presenti nel sistema. L'802.11e invece con l'aumentare del numero delle stazioni diminuisce la propria equità nel dividere la risorsa totale tra le varie stazioni contendenti al canale.

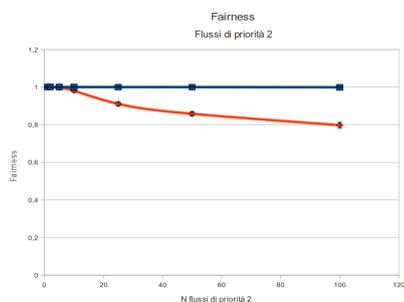
In figura 3.10(e) invece è mostrato il comportamento dei protocolli in uno scenario in cui sono presenti tutti i flussi che crescono in modo uniforme. Anche in questo caso il protocollo PAB mantiene una fairness costantemente al valore 1. L'802.11e al contrario perde in fairness all'aumentare del numero dei flussi; infatti si nota come le priorità basse soffrono molto della presenza dei flussi di alte classi e diminuiscono la loro fairness fino a scomparire quando non riescono ad inviare più nessun pacchetto.



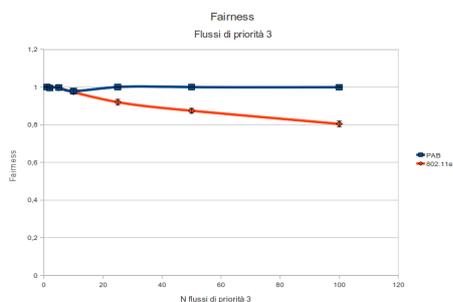
(a) Flussi di priorità 0



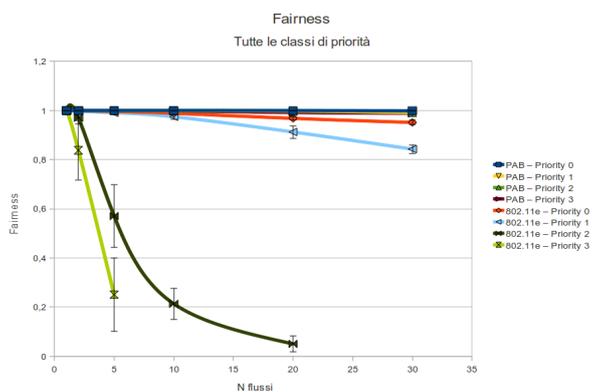
(b) Flussi di priorità 1



(c) Flussi di priorità 2



(d) Flussi di priorità 3



(e) Flussi di priorità 0, 1, 2 e 3

Figura 3.10: Confronto della caratteristica di fairness dei due protocolli

Conclusioni

In una rete di comunicazione di tipo wireless con un singolo canale condiviso, il protocollo di livello MAC gioca un ruolo importante nel sincronizzare l'accesso da parte di più stazioni. A causa della natura dell'accesso al canale di tipo casuale è impossibile evitare totalmente le collisioni tra pacchetti. L'approccio volto al solo schema del carrier sense come l'IEEE 802.11e costruito sulla base dell'IEEE 802.11 specificatamente per fornire servizi QoS, è sì molto potente, ma con dei problemi da non sottovalutare. Come visto dalle simulazioni infatti questo protocollo, nonostante sia di natura ad accesso casuale, soffre del problema della starvation per cui ci sono stazioni che non riusciranno mai ad inviare in certe condizioni della rete. Anche dal punto di vista del servizio QoS il protocollo degrada le sue prestazioni in presenza di un numero elevato di stazioni; in questo caso non riesce a garantire una differenziazione nelle prestazioni fornite tra varie classi di priorità presenti nella rete. Un altro problema è quello della fairness fornita ai flussi della stessa classe in cui le prestazioni non vengono percepite dalle stazioni nello stesso modo, ma ci sono alcune che riescono ad accedere più frequentemente rispetto ad altre aventi lo stesso livello.

È stato proposto un nuovo protocollo MAC chiamato PAB (*Priority Access with Burst*), il quale attraverso l'invio di impulsi sul canale e tramite l'uso delle priorità dinamiche, permette un uso migliore del canale con una presenza molto ridotta di collisioni a causa di una bassa contesa diretta tra le stazioni. Infatti grazie all'uso delle priorità dinamiche le stazioni accederanno alla rete ordinatamente in modalità round-robin. Questa caratteristica

rende questo protocollo equo dal punto di vista delle prestazioni fornite alle stazioni che cercano di accedere alla rete, proprio come viene mostrato dalle analisi che dimostrano come tutte le stazioni ricevono la stessa possibilità di trasmissione.

Un'altra caratteristica importante di questo protocollo è il supporto QoS che fornisce. Le prestazioni di una stazione di alta priorità non scenderanno mai sotto un determinato limite anche all'aumentare del numero dei flussi di priorità più basse presenti nel sistema; questo è possibile grazie alla presenza di un superframe che divide il tempo in due parti: una parte QoS in cui le priorità vengono rispettate e i flussi di alta priorità hanno la certezza di vincere la contesa contro i flussi a bassa priorità e una CB in cui si dà la possibilità a tutti di poter trasmettere.

Quest'ultimo meccanismo evita anche la presenza del problema della starvation dove una stazione non riesce mai ad inviare il suo dato; infatti nel frame contention based le stazioni hanno tutte la possibilità di inviare e, grazie alle priorità dinamiche che non hanno limiti, hanno la certezza di inviare prima o poi.

Una questione che si evince dalle simulazioni è il problema che sorge quando il numero delle stazioni che partecipano alla contesa diventa elevato; in questo caso infatti le prestazioni cominciano a degradare. Questa caratteristica non è un vero e proprio problema, ma una naturale conseguenza della presenza nel sistema di più flussi che contendono per una singola risorsa; il protocollo PAB essendo equo, cerca di effettuare una divisione equale tra le parti che contendono portando però anche ad una diminuzione delle prestazioni. Se prendiamo ad esempio il MAC delay, si nota dalle simulazioni effettuate un aumento del tempo di attesa lineare rispetto al numero dei flussi presenti. In questo caso se ci sono flussi delay-sensitive le proprietà QoS non verrebbero rispettate all'aumento del numero dei flussi.

Per risolvere questo problema c'è bisogno di un qualche tipo di sistema di *admission control* che limiti la possibilità di ingresso dei flussi nel sistema così da evitare l'ingresso di ulteriori flussi che determinerebbero una degradazione

troppo elevata delle prestazioni.

In futuro lo studio di questo protocollo potrebbe essere indirizzato ad un cross layering con il livello link superiore, così da poter essere studiato in ambito multi-hop inserendo quindi anche una politica di admission control che gestisca i vari flussi end-to-end che avvengono a più hop tra le varie stazioni. Inoltre il protocollo è già predisposto per l'inserimento di stazioni speciali (come ad esempio gli access point o i nodi fissi di una rete mesh) che grazie a delle priorità dinamiche non accessibili dai normali flussi trasmissivi, possono catturare il mezzo trasmissivo a prescindere dal suo stato ed avere una priorità maggiore per avere la possibilità di gestione del canale.

Appendice A

Panoramica sui modelli

In questa appendice verrà fatta una panoramica su come sono stati modellati i due protocolli sopra descritti attraverso l'uso del simulatore NS2 [1] [2].

A.1 Modello PAB

Il modello PAB (fig. A.1) è stato sviluppato a partire dal modello IEEE 802.11 presente nel pacchetto *ns-allinone-2.34* [1].

Sono stati aggiunti al codice pre-esistente due moduli: uno scheduler e un gestore dei flussi.

Lo scheduler si occupa di prendere i pacchetti provenienti dai livelli superiori ed inserirli nelle apposite code. Queste ultime sono gestite dal gestore dei flussi che crea delle code per ogni flusso presente; il numero delle code create per ogni flusso è definito, come già spiegato, dalla priorità originaria del flusso, per cui più è alta la priorità e più code, cioè sottoflussi, verranno create.

Inoltre il MAC informa lo scheduler di tutti gli avvenimenti che avvengono nella rete. Tramite queste informazioni (collisioni, successi o perdita di contesa) il gestore dei flussi è in grado di modificare le caratteristiche delle code presenti e quindi reagire di conseguenza. Ogni volta infatti che un flusso per-

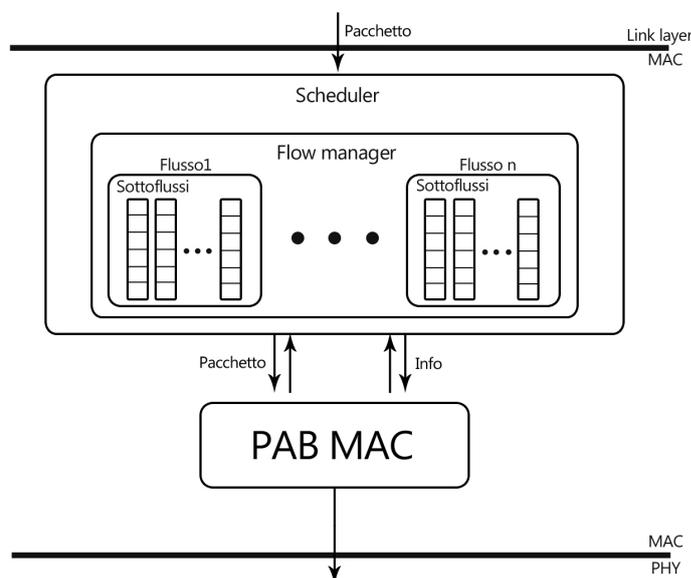


Figura A.1: Modello del protocollo PAB

de la contesa del canale, lo scheduler dopo aver aggiornato la code, informa il MAC dei nuovi valori di $PrIFS$, $burst_len$ e cw da utilizzare per accedere al canale.

Come è stato detto nella procedura di modifica di priorità dinamica, in caso di successo un flusso modifica la sua $sub_priority$ e $sub_priority_{PERNO}$ in base anche alla media dei perni ascoltati negli ultimi $N_SUPERFRAME * superframeTime$ secondi. Per fare questo quindi la stazione deve memorizzarsi il campo $used_perno$ di tutti gli header dei pacchetti che ascolta nella rete e tenerli aggiornati facendo in modo che vengano presi in considerazione soltanto quelli che hanno un timestamp compreso nel limite di tempo appena descritto.

I valori utilizzati per effettuare le simulazioni sono mostrati in tabella A.1.

CWMin	15
CWMax	255
SlotTime	20 μ s
BurstLength	20 μ s
SIFS	10 μ s
DataRate	1Mbps
Queue length	50
α	0,45
MAX_SUBPRIORITY	0
LV_PRIORITY	5
WEIGHT_PERNO_CALC	65
WEIGHT_PERNO_MEAN	35
N_SUPERFRAME	4
superframeTime	1 sec
NUM_SUCCESI_CONSEC	20
meanPernoProbability	5

Tabella A.1: Parametri utilizzati per le simulazioni del modello PAB

A.2 Modello 802.11e

Gli esperimenti effettuati sul protocollo IEEE 802.11e sono stati effettuati con il modello sviluppato e verificato da Wietholter e Hoene [3].

I valori utilizzati per effettuare le simulazioni sono presentati in tabella A.2, mentre i parametri che caratterizzano le diverse priorità i valori sono mostrati in tabella A.3.

CWMin	31
CWMax	1024
SlotTime	20 μs
SIFS	10 μs
DataRate	1Mbps
Queue length	50

Tabella A.2: Parametri utilizzati per le simulazioni del modello IEEE 802.11e

	0	1	2	3
PF	2	2	2	2
AIFS	50 μs	50 μs	70 μs	150 μs
CW_MIN	7	15	31	31
CW_MAX	15	31	1023	1023
TXOPLimit	3264 μs	6016 μs	0	0

Tabella A.3: Parametri caratterizzanti le diverse priorità

Bibliografia

- [1] *Network Simulator 2* (NS2), pp. [online] Available: <http://www.isi.edu/nsnam/ns/>
- [2] Teerawat Issariyakul, Ekram Hossain “*Introduction to Network Simulator NS2*”, Springer, 2008.
- [3] Sven Wiethölter, Christian Hoene, “*Design and Verification of an IEEE 802.11e EDCF Simulation Model in ns-2.26*”, Technical Report TKN-03-019, Telecommunication Networks Group, Technische Universität Berlin, November 2003.
- [4] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, L. Stibor “*IEEE 802.11e Wireless LAN for Quality of Service*”.
- [5] Yang Xiao, Yi Pan, “*Differentiation, QoS Guarantee, and Optimization for Real-Time Traffic over One-Hop Ad Hoc Networks*”, IEEE Transactions on Parallel and Distributed Systems, 2005, vol. 16, no. 6, pp.538-549
- [6] Yang Yaling, R. Kravets, “*Distributed QoS guarantees for realtime traffic in ad hoc networks*”, in Sensor and Ad Hoc Communications and Networks (IEEE SECON), Oct 2004, pp.118-127
- [7] Rong Geng, Zhe Li, Lei Song, “*AQMP: An Adaptive QoS MAC Protocol Based on IEEE802.11 in Ad Hoc Networks*” Wireless Communications, Networking and Mobile Computing, 2009, WiCom '09. 5th International Conference on , vol., no., pp.1-4, 24-26 Sept. 2009

-
- [8] G. Anastasi, L. Lenzini, E. Mingozzi, “*HIPERLAN/1 MAC protocol: stability and performance analysis*”, IEEE Journal on Selected Areas in Communications, 2000, vol. 18, no. 9, pp.1787 -1798
- [9] C. H. Yeh, T. You, “*A QoS MAC protocol for differentiated service in mobile ad hoc networks*”, in the 2003 International Conference on Parallel Processing (ICPP '03), 2003, pp.349-356
- [10] N. Ramos, D. Panigrahi, S. Dey, “*Quality of service provisioning in 802.11e networks: challenges, approaches, and future directions*” Network, IEEE, vol.19, no.4, pp. 14-20, July-Aug 2005
- [11] J. L. Sobrinho, A. S. Krishnakumar, “*Quality-of-service in ad hoc carrier sense multiple access wireless networks*”, Selected Areas in Communications, IEEE Journal on, Vol. 17, No. 8. (August 1999), pp. 1353-1368.
- [12] L. Jacob, Li Xiang, Zhou Luying, “*A MAC protocol with QoS guarantees for real-time traffics in wireless LANs*”, Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia, Proceedings of the 2003 Joint Conference of the Fourth International Conference on , vol.3, no., pp. 1962-1966 vol.3, 15-18 Dec. 2003
- [13] Jang-Ping Sheu, Chi-Hsun Liu, Shih-Lin Wu, Yu-Chee Tseng, “*A priority MAC protocol to support real-time traffic in ad hoc networks*”, Wireless Networks, Vol 10, pp. 61-69, 2004
- [14] Hai Jiang, Ping Wang, Weihua Zhuang, “*Enhanced QoS Provisioning in Distributed Wireless Access*”, Communications, 2006. ICC '06. IEEE International Conference on, Vol. 11, No., pp. 5111-5116, june 2006
- [15] Dapeng Wang, Kai Liu, “*A Novel QoS MAC Protocol for Multi-Hop Ad Hoc Networks*” Wireless Communications, Networking and Mobile

- Computing, 2007, WiCom 2007, International Conference on , vol., no., pp.1484-1487, 21-25 Sept. 2007
- [16] Mang Liao, Kai Liu, “*A distributed adaptive QoS MAC protocol for mobile ad hoc networks*” Communications and Networking in China (CHINACOM), 2010 5th International ICST Conference on , vol., no., pp.1-5, 25-27 Aug. 2010
- [17] Xue Yang, Nitin H. Vaidya “*Priority scheduling in wireless ad hoc networks*”, Wireless Networks, Vol 12 n.3, pp. 273-286, May 2006
- [18] R. K. Jain, D. M. W. Chiu, W. R. Hawe, “*A quantitative measure of fairness and discrimination for resource allocation in shared computer systems*”, DEC-TR-301, Digital Equipment Corporation, Tech. Rep., Sep. 1984.

