POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Informatica

Tesi di Laurea Magistrale

Implementazione ed Ottimizzazione di Tecniche di Ridondanza WiFi a Livello Applicativo

Relatori
Dr. Stefano Scanzio
Prof. Riccardo Sisto
Dr. Gianluca Cena

Candidato
Giuseppe Pedone

Luglio 2020

Indice

Elenco delle figure

Sommario

Lo scopo principale della presente tesi consiste nell'implementazione, e successiva sperimentazione in dispositivi reali, di tecniche di ridondanza finalizzate ad incrementare determinismo e affidabilità di reti WiFi, per consentirne il loro utilizzo come estensioni di rete wireless in applicazioni industriali. In particolare, è stato implementato uno livello software che si occupa della gestione della duplicazione dei dati da trasmettere su un mezzo trasmissivo wireless, al fine di ottenere miglioramenti prestazionali. Nelle tecniche di ridondanza, al fine di diminuire la latenza di trasmissione e aumentare l'affidabilità, lo stesso pacchetto di dati viene trasmesso su più canali, due nel caso di questa tesi. La tecnica Wi-Red rappresenta l'implementazione base di tecniche di ridondanza, in cui il pacchetto di dato viene sempre trasmesso in entrambi i canali. Allo scopo di ottimizzare il consumo di banda, evitando la trasmissione di dati duplicati quando non strettamente necessario, è stato introdotto ed analizzato sperimentalmente un meccanismo di Reactive Duplication Avoidance on Queue (RDA/Q). Il principio base di funzionamento di RDA/Q consiste nell'eliminazione, dalle code di trasmissione dei dispositivi WiFi coinvolti, di tutti quei pacchetti che sono stati consegnati con successo almeno su uno dei due canali, evitando in tal modo trasmissioni sicuramente inutili. Gli obiettivi pratici di questa tesi riguardano la verifica sperimentale dell'aumento del determinismo e riduzione della latenza, a parità di affidabilità, generato dall'utilizzo di RDA/Q rispetto alla tecnica base Wi-Red. Naturalmente, per verificare sperimentalmente i miglioramenti desumibili teoricamente, si è prestata molta attenzione agli aspetti implementativi, al fine di evitare che i ritardi introdotti dal software potessero inficiare i risultati sperimentali ottenuti.

Introduzione

L'esigenza di dare connettività ai dispositivi di rete anche in mobilità, ha portato alla nascita delle tecnologie cosiddette wireless, ovvero "senza fili". Mediante la trasmissione e la ricezione di onde elettromagnetiche, è diventato possibile far comunicare due dispositivi, anche in movimento, sfruttando come mezzo di propagazione non più dei cavi, bensì l'etere. Questa soluzione tecnologica ha permesso molte innovazioni in campi commerciali, pubblici, di ricerca e privati. La nascita della rete cellulare, e della Wireless Local Area Network (WLAN) hanno esteso la copertura Internet distribuita in tutto il mondo, nonché aumentato il numero di dispositivi mobili che università, aziende e famiglie utilizzano ormai tutti i giorni. Si pensi alle reti locali che le istituzioni e le aziende mettono a disposizione dei clienti e dei dipendenti per favorire un servizio gratuito di benessere comune, oppure agli oggetti intelligenti e connessi la cui diffusione sempre più ampia è dovuta proprio all'introduzione delle comunicazioni wireless. Nel campo dell'automazione industriale e della ricerca i sistemi cyber fisici e di automazione hanno adottato le tecniche wireless estendendo il loro raggio di applicazione e la loro flessibilità di utilizzo grazie alla possibilità di avere dispositivi in grado di muoversi e comunicare con altri senza fili. Tuttavia, come tutte le tecnologie, presenta vantaggi e svantaggi che vanno pesati a seconda delle applicazioni che la usano a dalle loro esigenze di qualità, efficienza e sicurezza.

In applicazioni di tipo general purpose si raggiungono accertabili livelli di qualità (Quality of Service) oltre che di efficienza. Lo stesso non si può dire per applicazioni in campo industriale dove gli standard di efficienza e sopratutto di qualità e sicurezza sono ben più elevati. Il segnale elettromagnetico ed i relativi frame trasmessi nell'etere, soffrono di problemi d'interferenza e di disturbo, al contrario di un segnale cablato isolato e protetto. Nel campo industriale molte applicazioni richiedono condizioni di determinismo, oppure viene richiesto un alto grado di affidabilità nella trasmissione dei dati (cioè una bassa percentuale di perdita dei pacchetti), altre possono richiedere un consumo energetico limitato oppure un protocollo con elevati standard di sicurezza. In ogni caso la recente comparsa di nuovi paradigmi come Industrial Internet of Things (IIoT) e Industry 4.0 stanno lanciando nuove sfide per migliorare l'infrastruttura di comunicazione wireless. Tipiche applicazioni critiche riguardano la sicurezza informatica, il cloud computing, l'analisi dei big

data, l'integrazione orizzontale/verticale, i sistemi cibernetici e sopratutto i loop di controllo. Ciascuna di esse necessita di tecnologie wireless differenti in grado di soddisfare le diverse specifiche.

La tecnologia Bluetooth ad esempio, è utilizzata per piccoli dispositivi di basso consumo energetico, a raggio d'azione dell'ordine delle decine di metri, a bassa capacità di trasferimento. Grazie alle ridotte esigenze hardware oltre a rivelarsi poco costosa, può essere installata anche su piccoli dispositivi come strumenti di misura, sensori, lettori digitali. Un'alternativa valida è costituita dalle cosiddette Low Rate Wireless Personal Area Network (LR-WPAN) che offrono collegamenti a bassa capacità ed a corto raggio (tipicamente inferiore a trenta metri). E' caratterizzata da un basso consumo energetico e richiede minimi costi di produzione, mantenendo la sua flessibilità (ha diverse modalità di funzionamento e permette diverse topologie di rete) e generalità (tutti i dispositivi la possono usare). La LR-WPAN è regolata dallo standard IEEE 802.15.4 che definisce il livello fisico (PHY) ed il livello Media Access Control (MAC) oltre che vari meccanismi di sincronizzazione, sicurezza e gestione dell'energia. Specifiche come ZigBee, 6loWPAN, WirelessHART, 6TiSCH e MiWi offrono delle implementazioni dei livelli mancanti della pila OSI (Open Systems Interconnection) appoggiandosi proprio su questo standard. In campo industriale, un'ulteriore soluzione è rappresentata dallo standard aperto IO-Link, che permette la comunicazione digitale tra sensori/attuatori ed il sistema di controllo (PLC o HMI). Questo protocollo è orientato a connessioni master-slave tra nodi che si scambiano continuamente informazioni riguardo allo stato del sistema. Il protocollo permette comunicazioni cicliche (dati di stato, di processo, di velocità) ed acicliche (dati del dispositivo, di diagnostica, di errore). La centralizzazione dei dati (il master immagazzina i dati utili provenienti dai vari slave) favorisce la diagnostica dei guasti, la manutenzione predittiva, ed il monitoraggio delle risorse. Lo standard consente inoltre di assegnare dinamicamente i parametri tramite segnalazione del master (basta intervenire sul master per la configurazione degli slave) e di sostituire in modalità di funzionamento un nodo guasto senza perdere la parametrizzazione diminuendo i rischi ed i costi di fermo impianto. Utilizzando comunicazioni digitali, è meno soggetto ad interferenze esterne garantendo cosi un'affidabilità maggiore. Questa è una soluzione orientata prettamente all'area industriale.

Per quanto riguarda reti wireless ad ampio raggio d'azione una soluzione potrebbe essere costituita da Long Range (LoRa), ovvero una tecnica di modulazione del segnale basata su diffusione dello spettro espanso derivata dal Chirp Spread Spectrum (CSS). Consiste nell'inviare un segnale con una banda molto ampia (quella del chirp copre molte frequenze) cosi da aumentare il rapporto segnale/rumore, ridurre l'effetto di interferenze, e far utilizzare la banda da più utenti contemporaneamente. La potenza utilizzata da LoRa è ridotta ed opera a frequenze sub-gigahertz (868 MHz Europa, 915 MHz Nord America) con raggio d'azione dell'ordine dei kilometri. Il livello fisico di LoRa è proprietario, dunque sappiamo che utilizza una modulazione simile a CSS e riduce gli errori tramite codifica Forward Error Correction

(FEC). Per quanto riguarda il livello MAC esiste un protocollo chiamato LoRaWAN che si occupa degli strati più alti della pila OSI, definendo le comunicazioni tra i dispositivi e l'architettura di rete. I nodi della rete trasmettono in maniera asincrona i dati, che vengono poi raccolti da più gateway ed inviati ad un server di rete centralizzato che scarta i duplicati e gestisce il collegamento applicativo. Questa struttura di rete aumenta notevolmente l'affidabilità, ma provoca problemi di prestazione specialmente con gli acknowledgement e con carichi elevati. Infatti viene utilizzata per applicazioni che richiedono la trasmissione di una bassissima mole di dati a lunghissima distanza (SmartCity, Impianti Petrolchimici, etc.).

Un'altra tecnologia wireless ad ampio raggio è costituita dalle reti WiFi (Wireless Fidelity) regolate dallo standard IEEE 802.11 che offre collegamenti ad elevata capacità, a raggio d'azione dell'ordine delle centinaia di metri ed anche più. Una rete di questo tipo da possibilità ai dispositivi di muoversi e comunicare tra loro entro un certo spazio e permette il trasferimento di grandi moli di dati in tempi ridotti. A discapito di un consumo energetico maggiore e di un hardware più complesso e costoso, questa soluzione si rivela ottima quanto ad uniformità della rete (in quanto la rete sarebbe realizzabile grazie all'uso di una sola tecnologia che si adatta a tutti i tipi di comunicazione), compatibilità con altre reti (come quelle IP che sono le più diffuse), scalabilità, capacità di trasferimento e raggio di copertura oltre ad eliminare completamente i costi di gestione e manutenzione del cablaggio. Tuttavia non tutte le applicazioni possono usare questa tecnologia poiché permette di soddisfare specifiche di soft real-time (cioè una certa percentuale di pacchetti deve arrivare entro una deadline fissata) ma non hard real-time (tutti i pacchetti arrivano entro una deadline fissata). Ad esempio, in uno scenario in cui il controllore (PLC) può dare i comandi in anticipo (rispetto al tempo di attuazione), viene inviato il processo da eseguire ai vari nodi una sola volta. Essi, collaborando tra loro, hanno bisogno soltanto di sincronizzazione temporale, che viene comandata dal controllore, verso i nodi agenti, attraverso una comunicazione continua. Si pensi ad un Robot, che tra due nastri trasportatori, compie azioni standard, in un certo istante di tempo (sposta un oggetto da un nastro all'altro). Qui l'azione è fissa (spostare il pezzo) e coordinata tra i due nastri ed il Robot. Per garantire la coordinazione, il controllore invia periodicamente segnali di sincronizzazione. Diverso invece è il caso in cui un sistema in loop di controllo effettua richieste e riceve risposte da un controllore, per effettuare determinate azioni in rapida successione e non note a priori. In questo scenario industriale le specifiche richiedono una rete deterministica in cui il segnale abbia un upperbound temporale per garantire che il nodo agisca in tempo utile, eseguendo l'azione richiesta.

Per quanto riguarda gli elementi architetturali della rete, come mostrato in figura 1, esistono gli Access-Point (AP) che sono dispositivi, collocati solitamente tra due reti differenti (ad esempio WiFi e Industrial Ethernet preesistente) e gli End-Node, ovvero nodi che usano WiFi per comunicare con gli AP come ad esempio gli Automated Guided Vehicles, Robot o sensori. Gli AP sono in grado di gestire gli

accessi degli EN alla rete esterna, con funzione di controllo, di ripetitore di segnale (per estendere la banda), di cambiamento di protocollo (ad esempio da frame 802.11 a frame Ethernet) e di gateway (interfaccia di collegamento con la rete esterna). In una rete WiFi industriale vengono posizionati più AP per garantire una copertura completa dell'area d'interesse (le aree d'azione sono sovrapposte cosi da evitare la mancanza di segnale in alcuni punti dello spazio). Tale disposizione non solo permette di coprire l'intero polo industriale sfruttando la rete preesistente, ma previene anche il problema del handover (procedura di rete di passaggio da un AP ad un altro, senza perdere la comunicazione) che tutti i dispositivi mobili, come gli AGV, hanno.

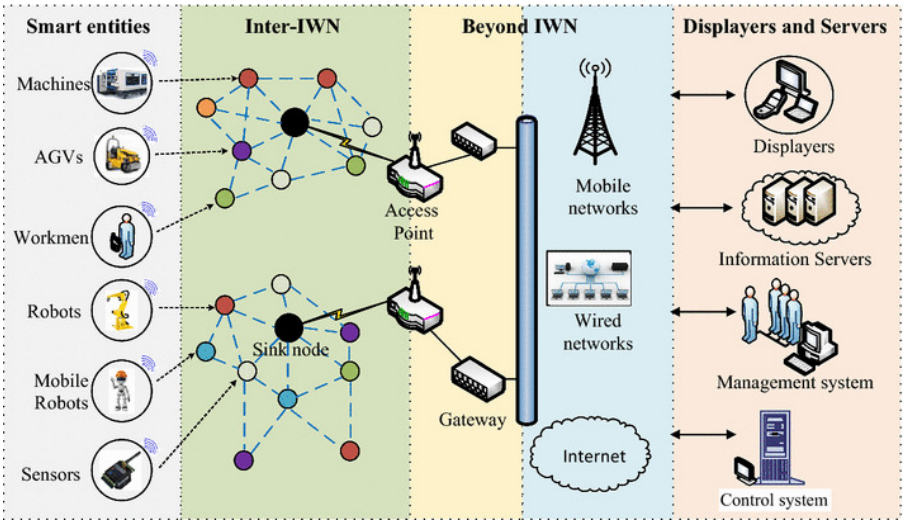

Figura 1. Questa figura mostra l'architettura e l'infrastruttura di una tipica rete industriale illustrandone le STA e le categorie di dispositivi utilizzati tipicamente nell'area industriale. Negli scenari industriali, generalmente le attività vengono eseguite ormai prevalentemente mediante la rete ed i dispositivi IIoT comandati da remoto. Dispositivi tipici sono Robot, Macchine comandate, AGV, Sensori, Attuatori. Fonte: [1]

Sebbene sono adottati in molteplici scenari industriali, le reti WiFi non sono ancora adeguate per supportare attività come il controllo dei movimenti, oppure la gestione dei processi industriali, che richiedono specifiche caratteristiche di affidabilità, sicurezza e determinismo del segnale. Perciò spesso viene installata utilizzando delle strategie d'Implementazione differenti dal WiFi tradizionale, che vanno ad agire sia a livello protocollare che architetturale, riuscendo a produrre delle reti utilizzabili anche in scenari estremi come quelli sopra citati.

Questa tesi si colloca nel contesto precedentemente descritto, ed in particolare nel riuscire a migliorare le caratteristiche in termini di latenza, determinismo ed affidabilità di un canale WiFi.

Nel successivo capitolo saranno introdotte le caratteristiche principali dello standard IEEE 802.11. Nei capitoli successivi, verranno specificate le evoluzioni più significative del protocollo IEEE 802.11. In particolare verranno discussi gli standard 802.11a/b/g/n/ac/ax/be. Ciascun'evoluzione ha delle peculiarità che lo contraddistinguono dai predecessori sopratutto in termini di prestazioni e affidabilità ma anche dal punto di vista architetturale. Nei capitoli 3 e 4 verranno trattate le architetture Seamless Redundancy ed SDMAC utilizzati all'interno di questa tesi. Verranno specificate le loro caratteristiche fondamentali relative ad applicazioni industriali soft real-time. Nel capitolo 5 verrà trattata l'architettura software implementata in questa tesi. In esso verranno specificati scopi, scelte implementative e verranno introdotti anche eventuali sviluppi futuri della medesima architettura. L'ultimo capitolo tratterà l'analisi dei dati sperimentali. Verranno commentati gli esperimenti e discussi i risultati sperimentali ottenuti dagli esperimenti effettuati su canali di rete reali.

Lo standard IEEE 802.11

Il protocollo IEEE 802.11 definisce una serie di standard di trasmissione per le reti WLAN con particolare attenzione ai livelli PHY e MAC della pila ISO-OSI. Vengono definite le interfacce tra AP ed STA oltre che quelle delle stazioni wireless, denominate Station (STA). Con il termine WiFi vengono etichettate tutte le apparecchiature appartenenti alla WiFi Alliance di cui fanno parte numerose aziende costruttrici di hardware (come Cisco, Netgear, Nokia, Intel, Broadcom, etc). Con WiFi nasce uno standard che rende interoperabili i dispositivi che lo usano, anche se appartengono a case costruttrici differenti. Il protocollo si è evoluto nel tempo, e ciascuna evoluzione viene indicata con una lettera come suffisso al nome base dello standard (802.11x). Ad esempio la famiglia di protocolli 802.11a/b/g/n sono stati sviluppati con particolare attenzione alla trasmissione delle informazioni, mentre l'802.11i si concentra sulla sicurezza. Le altre varianti offrono estensione e miglioramenti dei servizi base offerti.

Per quanto riguarda le frequenze di trasmissione utilizzate, i protocolli 802.11b e 802.11g usano la banda 2,4Ghz ISM (Industrial-Scientific-Medical banda denominata dall'ITU per indicare una gamma di frequenze ad uso non commerciale) mentre l'802.11a utilizza la banda 5,0Ghz ISM. I protocolli che operano a frequenze di 2,4Ghz suddividono la banda in 14 sottocanali da 20Mhz o 40Mhz (solo per il protocollo n) mentre quelli a 5,4Ghz la suddividono in 30 canali da un minimo di 20Mhz ad un massimo di 160Mhz. Fanno eccezione i protocolli IEEE 802.11 legacy ed IEEE 802.11b che hanno un ampiezza di canale di 22Mhz, e l'802.11n a 5,0Ghz che ha un ampiezza massima di 40Mhz. Anche si se hanno a disposizione molti canali di trasmissione, spesso ne vengono utilizzati un sottoinsieme poiché le bande dei canali sono parzialmente sovrapposte quindi due canali consecutivi non possono essere utilizzati contemporaneamente. Dalle specifiche in frequenza dei protocolli, vengono stabilite velocità di trasmissione e compatibilità con gli altri dispositivi.

Una differenza importante è che la banda attorno ai 2,4Ghz viene disturbata da apparecchiature come telefoni-cordless, forni a microonde, ripetitori audio/video ed altri apparecchi di uso domestico mentre quella attorno ai 5.4 Ghz soffre meno di problemi di interferenza. Poiché l'oramai datato protocollo IEEE 802.11a, che usa tale banda, non rispetta la normativa Europea ETSI EN 301 893 che prevede due meccanismi come Dynamic Frequency Selection (DFS) e Transmit Power Protocol

(TPC) e radar meteorologici, viene introdotto un nuovo protocollo di trasmissione a 5.4 Ghz, chiamato IEEE 802.11h, per risolvere i "difetti" della versione a.

Per il forte sovraffollamento delle frequenze 2,4 Ghz, è stato sviluppato un nuovo protocollo, chiamato IEEE 802.11n, che opera in Dual Band (ovvero sia sulle frequenze 2,4Ghz che 5,0Ghz) e garantisce una flessibilità e compatibilità con protocolli differenti.

Di base, il protocollo IEEE 802.11 nasce senza prevedere alcune funzioni di sicurezza come confidenzialità, integrità o accesso non autorizzato, quindi molte reti (specialmente di uso domestico) restavano libere da protezioni. Con l'incremento dei cyber-attacchi e la rottura di algoritmi di protezione, sono nati alcuni standard di sicurezza integrati nel protocollo WiFi. Il primo ad essere ideato è stato il protocollo Wired Equivalent Protocol (WEP) il quale definisce i protocolli di rete utilizzati per rendere sicure le trasmissioni tra dispositivi wireless ma viene tipicamente usato solamente per gli accessi non autorizzati, poiché offre poche proprietà di sicurezza. Dato che era un metodo di sicurezza opzionale per il funzionamento del protocollo, spesso non veniva nemmeno usato. Gli algoritmi di sicurezza utilizzati erano l'RC4 stream (molto veloce in contesti real-time come il wireless) e per l'integrità invece il CRC 32 (un crc calcolato ciclicamente, che necessita dunque di un vettore di inizializzazione spesso inviato in chiaro). Purtroppo questo protocollo era soggetto a molti cyber-attacchi data la lunghezza scarsa delle chiavi (24bit IV, 104bit KEY1, 40 bit KEY2) e la mancanza di un meccanismo efficiente di ricambio delle chiavi (osservando un numero sufficiente di messaggi si riesce facilmente a risalire alla chiave). Data la necessità di reti WiFi sicure sono stati ideati altri due protocolli di rete chiamati WiFi Protected Access (WPA, WPA2). Il primo implementa un diverso protocollo rispetto a WEP come il Temporal Key Integrity Protocol (TKIP) che genera chiavi diverse per ciascun pacchetto e gestisce l'integrità del pacchetto e l'autenticità del messaggio. Mentre WPA può essere installato su macchine con schede di rete che implementavano il WEP, il WPA2 necessità di schede di rete più evolute, con capacità crittografiche più elevate. Infatti esso usa algoritmi di sicurezza differenti rispetto all'RC4 ed al meccanismo di gestione chiavi TKIP introdotto da WPA, implementando CBC-MAC in CTR mode unitamente allo standard Advanced Encryption Standard (AES) che sono decisamente più robusti rispetto ai precedenti. I moderni standard di sicurezza prevedono l'uso di WPA2 con autenticazione Remote Authentication Dial-in User Service (RADIUS) protocollo di rete utilizzato per autenticazione, autorizzazione ed accounting degli utenti che utilizzano la connessione con metodo Extensible Authentication Protocol - Transport Layer Security (EAP-TLS), protocollo utilizzato per l'autenticazione, spesso incapsulato da RADIUS per effettuare trasferimenti dati di utenti verso un NAS-network authentication server gestito da un ISP-Internet Service Provider. Tuttavia è stato scoperto un possibile attacco legato al meccanismo WiFi Protected Setup (WPS) che mira a leggere il WPS pin, riuscendo così ad autenticarsi con il server ed a recuperare le chiavi di crittografia. Questo perché il WPS non

viene definito dallo standard IEEE 802.11 bensì dalle case costruttrici degli AP e dei router.

Fatta una panoramica sulle caratteristiche generali del protocollo IEEE 802.11, entriamo nel dettaglio del funzionamento del protocollo e delle caratteristiche hardware e software che un apparecchio deve avere per supportarlo.

Ciascun dispositivo, che necessita di effettuare una trasmissione WiFi, deve avere una scheda WiFi (scheda elettronica che fa da interfaccia logica e da supporto fisico al sistema di elaborazione che la usa) che si occupa, attraverso un antenna, di trasmettere i dati sull'etere. Questo Integrate Circuit (IC) implementa in hardware il livello PHY ed in hardware/firmware il livello MAC del protocollo IEEE 802.11. In particolare si distinguono in dispositivi FullMac/HardMac e SoftMac. I primi implementano il livello MAC sulla scheda tramite hardware (HardMac) o firmware (FullMac), mentre i secondi si servono delle capacità elaborative dei sistemi host che le ospitano limitandosi ad implementare quelle parti del protocollo con criticità temporali, come mostrato in figura 2.

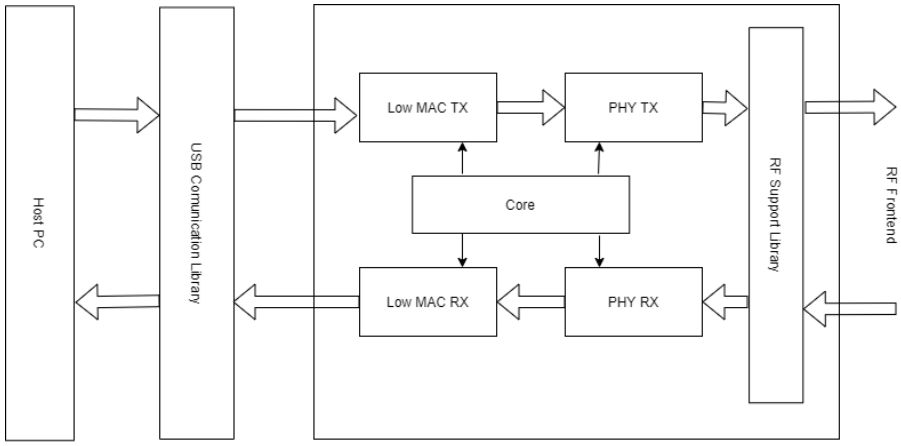

Figura 2. Architettura SoftMac di un WNIC generico. L'IC è composto da un core centrale che esegue le istruzioni, due moduli I/O MAC che eseguono le operazioni MAC di basso livello, due moduli I/O PHY, un analizzatore PHY che esegue alcuni controlli sulla sequenza ricevuta, un modulo per gestire il segnale radio (RF) ed un collegamento I/O Interrupt verso l'host.

Le Wireless Network Interface Controller (WNIC) possono operare generalmente in due modalità: Infrastructure ed AdHoc. La prima richiede l'utilizzo di un AP poiché in questa modalità tutti i nodi si connettono ad un unico nodo centralizzato che fa da mediatore per tutte le connessioni. La seconda invece è un'architettura distribuita ed opera senza l'utilizzo di un nodo centrale, inoltre connette tutti i nodi direttamente (i nodi nel raggio di azione dell'antenna). Generalmente la WNIC è connessa tramite un bus (tipicamente bus PCI-Peripheral Component Interconnect

o PCI-express) al sistema di elaborazione, anche se esistono modalità di connessione differenti come USB o PC Card (Parallel Communication Peripheral Interface).

Così come tutte le NIC può interfacciarsi alla Central Porcessing Unit (CPU) in polling, dove la CPU interroga l'interfaccia per sapere se è libera per trasmettere oppure in interrupt dove invece è l'interfaccia ad avvisare la CPU della sua disponibilità a trasmettere.

Per il trasferimento dati può operare in programmed Input/Output, cioè la CPU si occupa del trasferimento dei pacchetti dal NIC alla memoria e viceversa per via programmatica (codice eseguito sulla CPU), oppure in Direct Memory Access (DMA) cioè il dispositivo ha accesso diretto alla memoria del dispositivo host. Nel secondo caso, tipico nelle moderne schede WiFi, il NIC deve essere dotato di logica aggiuntiva per gestire il trasferimento, ma può fare a meno delle code visto che non deve attendere la CPU per il prelevamento dei pacchetti e ciò riduce la latenza.

Le NIC più moderne implementano anche il MultiQueue (code multiple) cioè utilizzano diverse code di trasmissione/ricezione così da poter smistare i pacchetti in arrivo in base al risultato di una funzione di hash (tipicamente un calcolo che richiede meno risorse computazionali) applicata al singolo pacchetto. Inoltre ogni coda di ricezione può essere mappata su un canale interrupt dedicato e processato da una CPU diversa aumentando così le prestazioni sfruttandone il parallelismo. Questa tecnica di distribuzione degli interrupt può essere implementata in hardware e si parla di Receive-Side Scaling (RSS) oppure in software ed in tal caso si parla di Receive Packet Steering (RPS) o Receive Flow Steering (RFS).

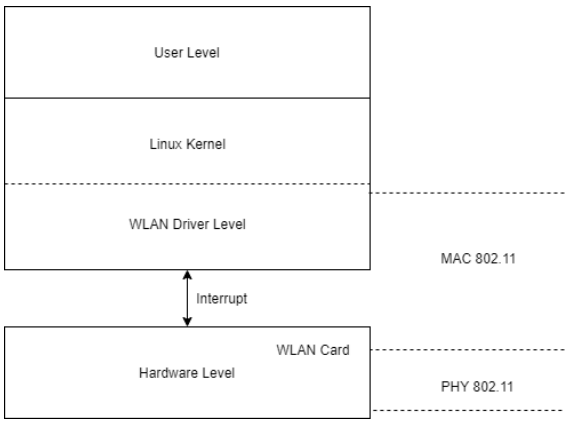

Figura 3. Architettura software di un generico dispositivo basato su sistema operativo Linux e collocazione dei livelli protocollari di IEEE 802.11 rispetto ad esso, nonché modalità di I/O utilizzata. Come si vede l'utente può, attraverso una system call, notificare al kernel la trasmissione di un pacchetto. Il sistema operativo si preoccupa di implementare il driver WLAN sottostante che, attraverso i canali di Interrupt, comunica alla scheda eventuali frame da trasmettere.

Un ulteriore miglioramento può essere ottenuto assegnando i pacchetti alle code affidate a quelle CPU che in quel momento eseguono l'applicazione a cui sono destinati. Questa tecnica migliora la località degli indirizzi, aumenta le prestazioni globali del sistema, riduce la latenza, richiede meno context-switches (cambio del flusso di esecuzione tra due processi) nonché migliora l'utilizzo del hardware, nel senso che viene utilizzata molto la CPU-cache. La stessa tecnica può essere utilizzata per i pacchetti in trasmissione, aumentando ulteriormente le prestazioni ed in tal caso si parla di Trasmit Packet Steering (TPS).

Esistono anche schede di rete che permettono l'implementazione dello stack TCP/IP (implementazioni del livello trasporto e collegamento dello stack ISO-OSI) direttamente a bordo tramite un TCP offload-engine di supporto, ma viene utilizzato per reti con throughput elevato, dove l'overhead di esecuzione (dello stack TCP/IP) è significativo. Oltre ad un apposito hardware, sono necessari degli strati software che permettono la comunicazione tra applicazioni e scheda di rete. Questi strati vengono implementati su vari livelli, come mostrato in figura 3, a partire dallo spazio utente fino ad arrivare al driver della scheda hardware. L'implementazione hardware di TCP/IP è raramente utilizzata in pratica.

Lo spazio utente comprende applicazioni di gestione, attraverso le quali vengono configurati e controllati i dispositivi di rete, riuscendo a definire i parametri dei livelli PHY e MAC sottostanti, inclusa la sicurezza. Applicazioni tipo sono: il Network Manager che consente, attraverso la Graphical User Interface (GUI), di gestire e configurare tutte le opzioni della parte di rete, i comandi di sistema `iwconfig`, `wpa_supplicant` ed `hostapd` che, mediante system call (definite nel modulo `nl80211` del sistema operativo) interagiscono con il modulo `cfg80211` (modulo kernel di basso livello per la configurazione ed il controllo della scheda). Altre funzioni vengono effettuate attraverso `wext` (wireless-extensions) che racchiude una serie di primitive di controllo a livello utente. Il Network Manager usa internamente i servizi forniti da `wpa_supplicant` per supportare le funzionalità wireless. `Hostapd` invece sfrutta le system call di `nl80211` unitamente a primitive `radiotap`, per implementare gli AP.

Gli strati di software sottostanti, mostrati in figura 4 e che sono stati sviluppati sopratutto nello spazio kernel, permettono un esecuzione di codice più performante ed adatta a tecnologie come il WiFi ed il protocollo IEEE 802.11. Il primo strato software è implementato dal modulo `cfg80211` che s'interpone tra lo spazio utente ed il driver di protocollo `mac80211`. Questo modulo si occupa della verifica di funzionalità e della traduzione dei protocolli dei dispositivi di rete. Fornisce funzioni che riguardano la registrazione dei dispositivi hardware, la gestione delle STAs, delle chiavi e delle interfacce virtuali, la scansione della rete, e il controllo di regolarità del dispositivo. I parametri di registrazione dei dispositivi includono la banda, i canali, il throughput e le modalità d'interfacciamento. Il controllo di regolarità invece assicura che, durante la registrazione dei dispositivi, vengono abilitati solamente i canali e le frequenze permessi dallo strato in cui si trovano. La gestione delle

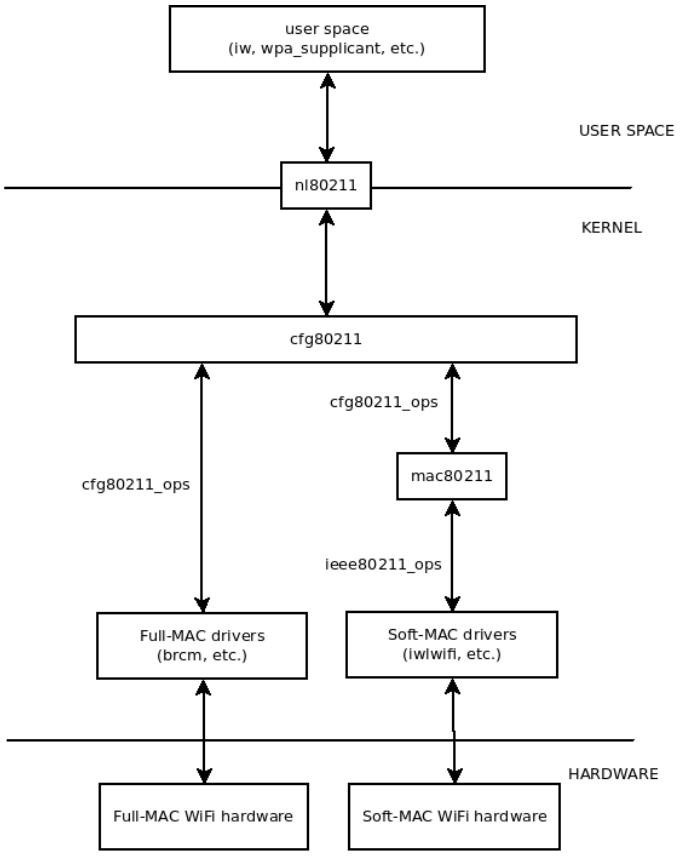

Figura 4. Architettura software divisa in moduli. Ciascun modulo esegue una specifica funzione e comunica con gli altri tramite un'apposita struttura dati, indicata sulle frecce di I/O. Inoltre ciascun modulo è implementato in uno specifico livello architetturale (Hardware, Kernel, Driver, User). Come si nota, l'architettura software, per i dispositivi FullMAC, è più semplice e meno dispendiosa poiché la maggior parte delle operazioni vengono fatte a bordo sulla scheda hardware. Fonte: [8]

STA, che fa parte delle capacità degli AP, offre funzionalità di aggiunta, rimozione, modifica e scaricamento dei dati delle stesse. La gestione delle interfacce virtuali consente di creare, rimuovere, cambiare tipo e monitorare i flag che le riguardano. Inoltre tiene traccia dell'interfaccia wireless di rete. La funzionalità di scansione fornisce all'utente la possibilità di inizializzare la scansione delle reti ed avviare attività di reportistica su di esse.

Al di sotto di questo modulo kernel si trova un ulteriore strato detto driver di protocollo, chiamato `mac80211`, che interagisce direttamente con il driver hardware e si occupa dell'implementazione del livello MAC (SoftMac). Questo supporta i protocolli IEEE 802.11a/b/d/g/n/s, differenti tipi di interfacce (STA, AP, monitor

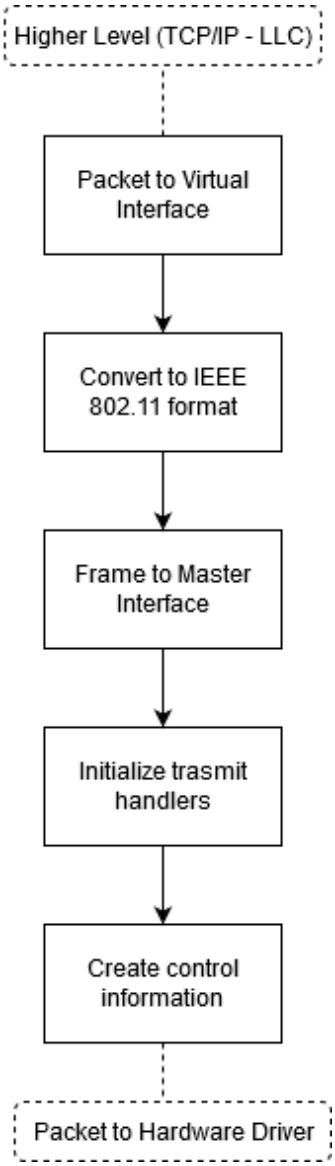

Figura 5. Processo di trasmissione di un generico dispositivo WiFi, dal livello TCP/IP fino al livello Hardware Driver, e le varie fasi intermedie che processano i dati per tradurre i pacchetti TCP/IP in frame IEEE 802.11 da inviare al NIC tramite Hardware Driver.

e mesh) e QoS e gestisce le seguenti funzioni di protocollo: trasmissione e ricezione. Nella trasmissione, come mostrato in figura 5, i livelli più alti (TCP/IP ed LLC-Logical Link Control) trasferiscono la struttura del pacchetto a livello MAC, sfruttando funzioni pubbliche del kernel. Esso viene poi convertito in formato IEEE 802.11, inizializzando i buffer e le intestazioni. Gli handler (gestori), selezionano la chiave crittografica e la velocità di trasmissione, inseriscono il numero di sequenza

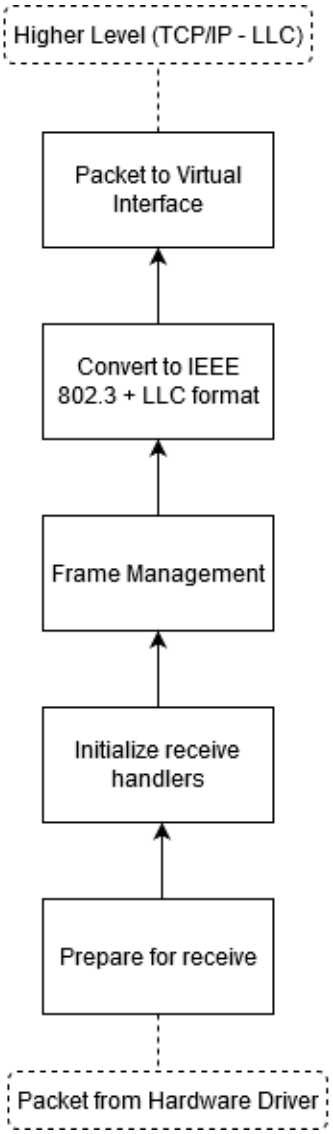

Figura 6. Processo di ricezione di un generico dispositivo WiFi, dal livello Hardware Driver fino al livello TCP/IP, e le varie fasi intermedie che processano i dati per tradurre i frame in pacchetti TCP/IP da passare al sistema operativo.

del pacchetto (che dipende dal hardware), selezionano l'algoritmo di criptazione e di frammentazione, calcolano il tempo di trasmissione e generano le informazioni di controllo. Successivamente il pacchetto viene inviato al livello driver hardware.

Nella ricezione invece, come mostrato in figura 6, il driver hardware invia i pacchetti catturati al driver di protocollo `mac80211` insieme alle informazioni di stato

del hardware. Quest'ultimo effettua un check sul tipo di pacchetto, ne riceve lo stato, e prepara gli handler di ricezione del livello superiore. Essi verificano l'allineamento dei pacchetti per il processamento, la decriptazione, e la deframmentazione dello stesso. Il processamento riguarda solamente i pacchetti di aggregazione, controllo, e di gestione (in seguito verranno discussi in dettaglio tutti i tipi di pacchetti IEEE 802.11). Alla fine viene effettuata una conversione del pacchetto dal protocollo IEEE 802.11 al protocollo 802.3 + LLC, che viene inviato ai livelli superiori (TCP/IP e LLC).

All'ultimo livello software troviamo il driver hardware, che si occupa dell'interfacciamento tra driver di protocollo e sistema hardware (NIC). Per comprendere meglio la sua funzionalità, lo analizziamo nelle fasi di trasmissione e ricezione.

Quando un pacchetto, proveniente dai livelli superiori, viene ricevuto dal driver hardware vengono inizializzati i buffer di memoria e mappati sulle code hardware della scheda. Inoltre vengono assegnati i flag in base al tipo di pacchetto trasmesso, ai parametri del livello PHY ed alle informazioni di controllo. Infine vengono trasferiti i pacchetti alla scheda attraverso il driver di interconnessione con il DMA. Eventuali interrupt di trasmissione vengono scatenati nel caso di ritrasmissioni o controllo di informazioni.

Quando invece un pacchetto viene catturato dal NIC, viene scatenato un interrupt di ricezione che richiama il driver hardware. Interagendo mediante opportuni blocchi con la scheda, i pacchetti vengono trasferiti al livello superiore, con le relative informazioni di stato.

Per entrare nel dettaglio degli aspetti protocollari dello standard IEEE 802.11, analizziamo i vari tipi di pacchetti introdotti e le interazioni tra dispositivi che li sfruttano. Nella terminologia IEEE 802.11 i pacchetti vengono chiamati frame (trame). Esistono tre categorie di frame:

- Data frame usati per la trasmissione di dati.
- Control frame usati per controllare l'accesso al mezzo fisico.
- Management frame trasmessi come i Data frames, usati per la gestione.

Ciascun tipo di frame è suddiviso in due sottotipi, a seconda della loro funzionalità. Tutti i frames IEEE 802.11 di basso livello sono composti dalle seguenti parti: Preambolo, PLCP Header, MAC data, CRC.

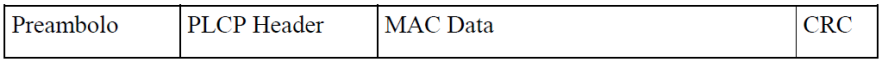

Figura 7. Struttura di un frame PHY generico. Fonte: [2]

Il Preambolo include un campo di sincronismo, ovvero una sequenza di 80 bit usata dalla scheda di ricezione per selezionare l'antenna appropriata e per ottenere la correzione dell'offset di frequenza e la sincronizzazione sul pacchetto ricevuto. Inoltre contiene un delimitatore di inizio frame chiamato SDF che consiste in una sequenza binaria fissa di 16 bit che viene utilizzata per definire la sincronizzazione dei data rate per singoli frame.

Il PLCP Header contiene informazioni logiche utilizzate dallo livello PHY per la codifica del frame. Esso è composto da:

- PLCP_PDU Lenght Word che indica la quantità di bytes contenuti in un pacchetto, utile al PHY per capire la lunghezza e quindi la fine del pacchetto.
- PLCP Signaling Field che contiene le informazioni sulla velocità di trasmissione codificate in intervalli discreti.
- Header Error Check Field, campo a 16 bit contenente il CRC del solo Header.

Il campo MAC data contiene tutti i dati provenienti dal livello superiore (livello MAC appunto), ed il suo header è articolato nelle seguenti sezioni: Frame Control, Duration ID, Address Fields, Sequence Control, Frame Body, CRC.

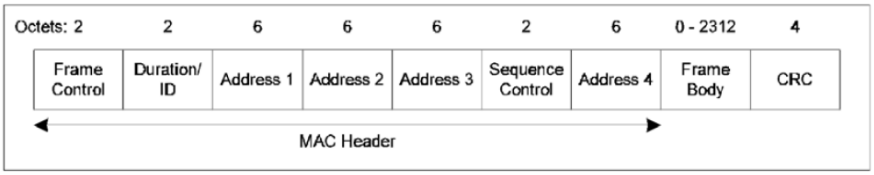

Figura 8. Struttura del header MAC IEEE 802.11. Fonte: [2]

Il campo Frame Control contiene le seguenti informazioni:

- Versione del Protocollo che consiste in 2 bit che sono utilizzati dalle successive versioni dello standard IEEE 802.11. Fissato a 0 per la versione base.
- Tipo e Sottotipo identificati da 6 bit che indicano il tipo ed il sottotipo del frame utilizzato.
- ToDS rappresentato da un bit impostato al valore 1 quando il frame è indirizzato all'AP per l'inoltro al Distribution System oppure a 0 in tutti gli altri frames.
- FromDS bit impostato ad 1 quando il frame è ricevuto dal Distribution System, a 0 negli altri casi.

- More Fragments bit ad 1 quando ci sono più pacchetti appartenenti allo stesso frame.
- Retry bit che indica un pacchetto ritrasmesso. È usato dal ricevitore per riconoscere la trasmissione di frame duplicati che indicano eventuali frame di dato o ACK persi.
- Power Management indica lo stato di Power Management raggiunto dopo la trasmissione corrente. È usato dalle stazioni per monitorare lo stato energetico della sorgente.
- WEP bit indica l'utilizzo del protocollo WEP per il pacchetto corrente.
- Order bit indica che è stata usata la QoS-class Strictly-Ordered.

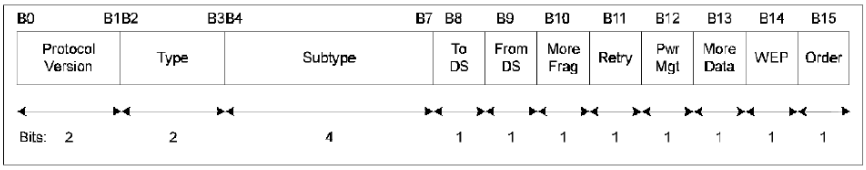

Figura 9. Struttura del campo Frame Control, il primo campo del header MAC IEEE 802.11. Fonte: [2]

Il Duration ID è un campo che ha due significati dipendenti in dipendenza dal tipo di frame. Nei pacchetti di Power-Save Poll (messaggio di management dell'energia) indica l'ID della stazione.

Un frame può contenere fino a 4 indirizzi in funzione del valore dei bit FromDS e ToDS. Address-1 è sempre l'indirizzo del ricevente, tipicamente la stazione a cui si vuole sperire il pacchetto. Valori ad 1 del ToDS corrispondono agli indirizzi degli AP, valori a 0 invece corrispondono ad indirizzi delle STAs di destinazione. Address-2 è sempre l'indirizzo sorgente. Se FromDS vale 1 allora rappresenta l'indirizzo dell'AP, se è 0 allora è l'indirizzo della sorgente. Address-3 è in molti l'indirizzo mancante. Nei frames con FormDS a 1, Address-3 rappresenta l'indirizzo originale della sorgente, mentre se il ToDS è a 1 allora l'Address-3 indica l'indirizzo del destinatario. Address-4 è usato in casi particolari dove è implementato un Wireless Distribution System ed il pacchetto passa da un AP all'altro. In questo caso il ToDS ed il FromDS sono impostati ad 1 in modo che non vengono impostati indirizzo sorgente e destinazione.

Il Sequence Control permette di enumerare i pacchetti appartenenti ad uno stesso frame e di riconoscere pacchetti duplicati. Consiste di 2 campi, Fragment Number e Sequence Number, i quali rispettivamente rappresentano il frame principale ed il numero del pacchetto di quel frame specifico.

Il Frame Body rappresenta il payload del livello superiore (tipicamente IP).

Il Cyclic Redundancy Check (CRC) è una sequenza di 32 bit contenente un codice di rilevazione dell'errore calcolato sul frame MAC trasmesso.

Prima di descrivere la struttura di ciascun tipo di pacchetto, saranno analizzate le procedure protocollari da seguire per implementare il livello PHY e MAC dello standard IEEE 802.11. Lo standard definisce un singolo livello MAC che interagisce con uno dei seguenti tre livelli PHY sottostanti che operano a velocità di trasmissione differenti (dipende dal protocollo):

- Frequency Hopping Spread Spectrum (FHSS) nella banda ISM 2,4Ghz.
- Direct Sequence Spread Spectrum (DSSS) nella banda ISM 2,4Ghz.
- Trasmissione ad Infrarossi.

Oltre a fornire le funzionalità di base della comunicazione, il livello MAC IEEE 802.11 fornisce funzioni, tipicamente implementate ai livelli superiori, come la gestione della frammentazione del payload (messaggio del livello superiore), la ritrasmissione dei frame e la gestione degli ACK (per migliorare l'affidabilità). Per quanto riguarda le modalità di accesso al mezzo ne esistono 2 denominate Distributed Coordination Function (DCF) e Distributed Coordination Function (PCF).

La prima, DCF, quella più comunemente utilizzata, è basata sul meccanismo Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) che serve per accedere in maniera coordinata ad un canale di comunicazione. Di base il protocollo CSMA funziona nel modo seguente: la stazione che vuole trasmettere ascolta il canale per un certo lasso di tempo. Se il canale è occupato (qualcun'altro sta trasmettendo) la trasmissione viene rimandata ad un istante successivo, mentre nel caso in cui il canale sia libero, la trasmissione viene eseguita. Questa famiglia di protocolli risulta efficiente nel caso in cui il mezzo di trasmissione non venga sfruttato eccessivamente. Può succedere che due stazioni trasmettano simultaneamente un frame, pur avendo controllato il canale. In questo caso si verifica una collisione sul mezzo di trasmissione. Per evitare successive collisioni nelle ritrasmissioni, questo caso deve essere gestito dal livello MAC in modo da non appesantire i livelli superiori dello stack protocollare. Nel caso del Carrier Sense Multiple Access with Collision Detection (CSMA/CD) questo evento viene gestito direttamente dal protocollo. Il trasmettitore ascolta il canale anche nella fase di trasmissione, in modo da accorgersi di eventuali trasmissioni simultanee. Se ne vengono rivelate altre, scatta la procedura di posticipo dell'algoritmo di Backoff Esponenziale Casuale (un ritardo casuale introdotto dalla stazione stessa per evitare statisticamente una collisione successiva). Il ritardo viene aumentato esponenzialmente nel caso di ulteriore collisione in modo da rendere la collisione successiva molto meno probabile. Purtroppo questa variante del protocollo CSMA non può essere adottata nel caso di comunicazioni wireless per due principali motivi:

- La capacità ricetrasmittenti della scheda wireless dovrebbero consentire di ascoltare il canale anche in trasmissione, e questo richiederebbe l'implementazione di un meccanismo full-duplex che renderebbe decisamente più costoso l'apparato.
- Nelle reti wireless le stazioni non possono ascoltarsi a vicenda e questo impedirebbe la corretta implementazione del Collision Detection (recepire un canale libero non vuol dire che l'area di ricezione sia libera).

Per questo motivo il protocollo IEEE 802.11 usa CSMA/CA unitamente ad un meccanismo di supporto chiamato Positive Acknowledge. Questo approccio modifica il comportamento classico del CSMA/CA nel modo seguente: la stazione che vuole trasmettere ascolta il canale. Se è occupato, rimanda la trasmissione. Se invece il canale è libero per una certa quantità di tempo detta Distributed Inter Frame Space (DIFS) allora la trasmissione avviene. La stazione che riceve, calcola il CRC ed invia un Acknowledgement. La ricezione della conferma denota che non sono avvenute collisioni e che la stazione ricevente ha ricevuto il pacchetto per intero. In mancanza di conferma, il trasmettitore ritrasmette il pacchetto fino ad un certo numero di volte, oltre le quali il pacchetto viene eliminato dalla coda di trasmissione.

Per ridurre ulteriormente la probabilità di una collisione a causa dell'impossibilità di ciascuna stazione di percepire le altre, lo standard IEEE 802.11 definisce un altro meccanismo denominato Virtual Carrier Sense (VCS): una stazione che vuole trasmettere, invia un pacchetto di controllo denominato Request To Send (RTS) che definisce l'indirizzo sorgente, destinatario e la durata della prossima trasmissione. La destinazione risponde (se il mezzo è libero) con un pacchetto di replica denominato Clear To Send (CTS), che definisce le stesse informazioni di durata del pacchetto RTS, depurata dalla lunghezza del pacchetto CTS. Tutte le stazioni riceventi il pacchetto RTS e/o CTS, comunicano al Network Allocation Vector (NAV) il contenuto del pacchetto, così da evitare trasmissioni in quell'arco di tempo indicato dalla durata nel pacchetto RTS/CTS. Queste informazioni vengono sfruttate dal Physical Carrier Sense (PCS) quando ascoltano il canale fisico. Il VCS riduce la probabilità di collisione su un'area di ricezione più ampia, anche non rilevabile dalla stazione trasmettitrice, e consente al ricevitore di riservare il mezzo fino alla fine della trasmissione corrente. Il pacchetto RTS inoltre protegge il trasmettitore da eventuali collisioni durante l'ACK (protegge durante tutta la fase di autenticazione) da parte di stazioni che sono invisibili al ricevente (colui che invia l'ACK). Oltretutto grazie alle ridotte dimensioni dei pacchetti RTS e CTS, è stato ridotto l'overhead dovuto alla collisione, poiché questi vengono processati più velocemente (vero solamente nel caso in cui il pacchetto è decisamente più grande di RTS/CTS, infatti il protocollo permette ai pacchetti dello stesso ordine di grandezza, di essere trasmessi senza RTS/CTS e questo viene gestito dalle stazioni grazie ad un parametro detto RTS Threshold).

Esistono diverse motivazioni che portano all'utilizzo di pacchetti di dimensioni inferiori in un ambiente wireless:

- Il Bit Error Rate (BER) per una connessione wireless è molto rilevante e la probabilità di trasmissione del pacchetto è proporzionale alla sua lunghezza.
- Per i pacchetti non trasmessi o trasmessi con errori, si ha un overhead minore per quelli più piccoli.
- In un sistema FH non è garantita la continuità del mezzo trasmissivo dovuta ai salti di frequenza continui. Riducendo la dimensione del pacchetto diminuisce la probabilità che la trasmissione sia posticipata dopo il periodo di pausa (dwell time).

Molto importante è il ruolo svolto dagli spazi Inter Frame Space (IFS) che regolano le tempistiche di trasmissione, sincronizzano i nodi, rendono le collisioni meno probabili e definiscono delle priorità di trasmissione dei frame tra stazioni. Lo standard definisce 4 tipi di IFS:

- SIFS - Short Inter Frame Space corrisponde alla differenza di tempo tra l'invio del primo simbolo del frame di risposta e la ricezione dell'ultimo simbolo del frame di richiesta. Esso è relazionato alla latenza del ricevitore e corrisponde alla somma dei tempi di computazione del header PLCP e MAC. Dipende sempre dal livello fisico, e viene impostato ad un valore fisso in base allo standard.
- PIFS - Point Coordination IFS, è usato dall'Access Point per quanto riguarda l'accesso al canale per l'invio dei beacon frame (un particolare ed importante frame di controllo). Il suo valore è pari al SIFS + Time Slot.
- DIFS - Distribuited IFS corrisponde al tempo di attesa di una stazione che vuole trasmettere un frame dati. Il suo valore è pari al SIFS + Time Slot.
- EIFS - Extended IFS corrisponde al tempo di attesa di una stazione per evitare una collisione.

I tempi di IFS servono ad evitare la collisione tra classi di frame, in ogni caso all'interno della classe può sempre avvenire. In questi casi viene usato un algoritmo per rendere meno probabile future collisioni, già precedentemente accennato, e chiamato backoff esponenziale. Questo algoritmo prevede la scelta, da parte di ogni stazione, di un numero casuale intero tra 0 ed n e la successiva attesa di tale numero di time slot, ascoltando il canale per recepire eventuali occupazioni del mezzo. Il time slot è definito in maniera tale che una stazione può sempre accorgersi di una trasmissione nel time slot precedente. Ciò rende meno probabile le collisioni.

Il numero n (il massimo numero di time slot di attesa) viene incrementato esponenzialmente ad ogni collisione successiva ed azzerato nel caso di trasmissione con successo. Il protocollo IEEE 802.11 definisce l'uso del backoff nei seguenti casi:

- La stazione ascolta la prima volta il canale per un dato pacchetto.
- Viene effettuata una ritrasmissione.
- Viene effettuata una trasmissione con successo (che non sia la prima).

Quando una stazione vuole accedere ad una rete WiFi, deve associarsi con l'AP e dunque viene eseguito la scansione delle reti vicine in modo passivo oppure attivo mediante i seguenti metodi:

- Passive Scanning in cui la stazione aspetta il Beacon frame periodico inviato dall'AP. Catturando quel frame, la stazione riesce ad associarsi con l'AP automaticamente. Richiede un tempo minimo di attesa ma nessuna elaborazione aggiuntiva.
- Active Scanning in cui la stazione tenta di localizzare un AP mediante l'invio esplicito di un Probe Request Frame ed attendendo un Probe Response Frame da quest'ultimo.

La scelta del metodo dipende dal consumo energetico e dalle prestazioni che si vogliono raggiungere.

Una volta ottenuta l'associazione con l'AP, si procede con la fase di autenticazione che prevede lo scambio di messaggi di sicurezza (tipicamente messaggi crittografati) contenenti eventuali password oppure challenge svolte dalla STA e verificate dal server. In aggiunta alla scelta dell'AP, la fase di associazione serve a definire le caratteristiche della stazione che richiede l'associazione ed a trasmettere quelle informazioni. Terminate queste due fasi, l'AP è abilitato a ricevere e trasmettere pacchetti dati da e verso quella STA.

Un altro aspetto da tenere in considerazione per le reti WiFi è il concetto di roaming. Con questo processo si consente lo spostamento di una stazione da un AP all'altro (da una cella all'altra), senza perdita di pacchetti. La gestione del meccanismo di roaming assomiglia a quello di handover nella telefonia mobile, ma in una LAN la transizione è di più facile gestione (esistono entità chiamate pacchetti, e tra un pacchetto e il successivo può intercorrere del silenzio, mentre nella telefonia la voce non può subire interruzioni) ma eventuali perdite di pacchetti riducono notevolmente le prestazioni poiché le ritrasmissioni vengono effettuate tipicamente dai livelli superiori dello stack protocollare (nella voce, una perdita momentanea del segnale non influenza la qualità della conversazione). L'802.11 non definisce la modalità d'implementazione del roaming ma solamente il funzionamento di base. Mediante lo scanning (passivo o attivo) la stazione in movimento rileva altre stazioni

vicine e decide a quale connettersi successivamente in base al livello di segnale ricevuto. Attraverso la ri-associazione (processo definito dallo standard) i due AP interessati configurano lo scambio utente. Le comunicazioni vengono mediate dal Distribution System senza occupare il canale principale.

Come detto più volte, le stazioni hanno bisogno di mantenere il sincronismo di canale (per il frequency hopping, per il power management e per il corretto uso del canale condiviso). Questo viene ottenuto mediante l'aggiornamento del clock dei singoli sistemi attraverso l'invio periodico dei Beacon frame. Questo contiene le informazioni sul clock dell'AP nel momento della trasmissione (viene catturato il clock di sistema un istante prima della trasmissione in modo da avere un valore il più affidabile possibile). Ricevuto questo frame, le stazioni aggiornano il loro clock per mantenerlo in linea con quello dell'AP.

Essendo nate per gestire dispositivi mobili, le reti WiFi devono gestire ed ottimizzare anche le risorse energetiche che spesso sono limitate. Per questo motivo lo standard IEEE 802.11 definisce delle modalità di funzionamento del dispositivo detto Power Saving, che permettono di far sospendere le STA in determinati istanti temporali. Per gestire questo tipo di modalità, l'AP deve tenere traccia dei dispositivi in questo stato e deve mantenere in coda tutti i pacchetti indirizzati a quelle stazioni che lui riconosce essere in Power Saving in modo da trasmetterle solo nel caso in cui queste ritornino in uno stato consistente. L'implementazione di questo meccanismo sfrutta i Beacon frame e precisamente l'AP che detiene i pacchetti relativi alla stazione in idle, ed inserisce nel Beacon frame tale informazione. Il dispositivo interessato, esce dallo stato di Power Saving per ricevere il Beacon frame e si accorge che ci sono dei pacchetti per lui. Ecco che invia una richiesta di polling all'AP per farsi inviare i pacchetti attesi. Per quanto riguarda i pacchetti broadcast e multicast, essi vengono inviati dall'AP solo quando tutte le stazioni di destinazione sono attive, quindi tipicamente hanno dei tempi di coda più lunghi.

Analizzando da vicino alcuni tipi di frame di management possiamo capire meglio la struttura dell'header MAC di quest'ultimi.

Ad esempio il formato del frame RTS prevede un Receiver Address (RA) pari a quello della STA destinataria del prossimo pacchetto da trasmettere (dati o di gestione). Il TA (Trasmission Address) è quello del trasmettitore del pacchetto RTS corrente. Mentre il campo Duration ID viene impostato calcolandolo come il periodo di tempo, in microsecondi, della prossima trasmissione di tipo dati o management, più il tempo di un frame CTS più tre intervalli SIFS.

I frame di tipo CTS invece hanno come RA il campo TA del frame RTS precedente, come Duration ID lo stesso valore del frame RTS inviato prima meno il tempo di invio del CTS meno ancora un SIFS.

Per quanto riguarda il frame ACK, esso ha come RA l'Address-2 del precedente frame dati. Il campo Duration ID dipende dal valore di More Fragment nel campo Frame Control del frame dati precedente. Se era basso, allora il campo Duration ID viene impostato a 0, mentre se era alto, allora viene calcolato partendo dallo

stesso campo del precedente frame dati meno il periodo di tempo, in microsecondi, richiesto dalla trasmissione del frame ACK e del SIFS.

Oltre alla DCF, esiste anche ed è opzionale, una Point Coordination Function (PCF) che viene tipicamente utilizzata per applicazioni time-bounded come voce, video, oppure in contesti industriali. La PCF fa uso di IFS più piccoli dei DIFS, chiamati PIFS, che forniscono una priorità più elevata. Grazie a questo tipo di accesso al canale condiviso, l'AP gestisce le richieste di polling da parte delle STA, in modo da liberarsi prima dei pacchetti in coda da più tempo. Ovviamente questa funzione di accesso potrebbe causare problemi di priorità per l'accesso continuo al canale, infatti vengono utilizzate politiche di fairness sull'AP in modo da rispettare anche l'accesso distribuito (DCF).

Dopo aver descritto le caratteristiche generali del livello MAC IEEE 802.11 passiamo al livello fisico. Esso è composto da due sottolivelli chiamati PLCP e PMD. Il primo si occupa della convergenza del livello fisico e traduce frame fisici in pacchetti comprensibili a livello MAC. Il secondo invece dipende dal mezzo fisico e gestisce l'invio e la ricezione dei singoli bit mediante segnali radio o infrarossi. La relazione tra questi due livelli è di dipendenza, nel senso che il primo sta sopra al secondo. Il livello PMD riceve uno stream di dati dal livello PLCP e li converte in sequenza di bit da poter trasmettere ad un'altra stazione. Il modulo PLCP invece fa da interfaccia fisica al livello MAC sovrastante. Di base questi due sottolivelli sono stati progettati in modo tale che la dipendenza dal mezzo resti al livello PMD e quella del MAC resti a livello PLCP. Il sottolivello PMD permette l'utilizzo di tre meccanismo di gestione del mezzo fisico quali FHSS, DSSS ed IR. Per ciascuna modalità d'uso del mezzo, bisogna fornire un sottolivello PLCP corrispondente in maniera da tenere separati il livello MAC da quello PHY. Ad esempio PLCP FHSS riceve i frame dal livello MAC (chiamati anche MPDU) che vengono incapsulati come Payload dentro pacchetti PLCP e spediti poi a livello PMD. Il frame PLCP, chiamato PPDU, è composto da un preambolo ed un header PLCP oltre che dal frame MAC (MPDU) oppure dal frame di servizio PLCP (PSDU).

Il preambolo PLCP è composto dal SYNC e dallo Start Frame Delimiter. Il primo è costituito da un pattern del tipo 0101... di 80 bit e serve al demodulatore (circuito fisico) per sincronizzarsi sul frame e per aiutare il livello fisico a riconoscerne l'inizio. Il secondo (SFD) è formato da 16 bit statici (0000 1100 1011 1101) e fissa un riferimento di tempo del frame. Il primo bit del PPDU segue immediatamente l'header PLCP. Questo contiene il PLW (PSDU Length Word) costituito da 12 bit che indicano il numero di ottetti contenuti nel PSDU e precisamente serve ad identificare la fine del frame PPDU (l'ultimo bit del frame). Oltre al PLW, viene definito un ulteriore campo chiamato PLCP Signaling Field (PSF) composto da 4 bit, che definisce il bitrate di invio del PSDU. Il primo di questi bit è riservato, mentre gli altri 3 (da 000 ad 111) indicano in maniera discreta dei bitrate secondo una certa ampiezza d'intervallo. Per finire lo Header Error Check (HEC) è composto da 16 bit di rilevazione dell'errore sui soli campi PLW e PSF. Gli errori

sull'intero frame vengono identificati dal livello MAC.

Prima del processamento PLCP, il frame MAC passa per uno scrambler (frame sincrono) a 127 bit che riorganizza i bit così da evitare sequenze lunghe di 1 e 0 che fanno perdere il sincronismo. Ovviamente è necessario uno de-scrambler lato ricevitore che riordina i bit nel modo giusto.

Il livello fisico supporta differenti velocità di trasmissione (tipicamente 1 e 2 Mbps nelle versioni meno recenti del protocollo). Ciascuna porzione del frame viene inviata con una certa velocità. Il preambolo e l'header PLCP sono trasmessi ad 1 Mbps in maniera che tutte le STA possono sincronizzarsi e leggere l'header in tempo. Se il data rate indicato nel PSF non è adatto alla comunicazione (cioè la STA ricevente non supporta il bitrate) allora viene scartato il pacchetto oppure viene sollevata un'eccezione (unsupported data rate).

Entrando nel merito della modulazione del segnale fisico, una delle prime modulazioni utilizzate è il Gaussian Frequency Shift Key a due livelli (2GFSK) per le trasmissioni ad una data velocità. Sfruttando lo spostamento minimo (di un Fd) della frequenza portante rispetto a quella centrale del canale in uso. Ovviamente questo sfasamento deve essere determinato al ricevitore e viene elaborato tenendo conto degli ultimi 8 bit del campo SYNC del preambolo. Uno 0 è codificato in uno spostamento negativo (minimo 110 Khz) rispetto alla frequenza centrale, mentre un 1 è codificato in uno spostamento positivo rispetto alla medesima.

Per fare in modo che la velocità di trasmissione sia modificabile, bisogna adottare una modulazione differente rispetto a quella 2GFSK, ovvero quella a 4 livelli detta anche 4GFSK. Questo tipo di modulazione codifica i simboli con 4 spostamenti in frequenza della portante rispetto alla frequenza centrale di canale. In questo modo è possibile codificare 2 bit anziché 1 bit. Il funzionamento del sistema viene modificato in questo modo: il PMD riceve i dati a 2 Mbps anziché a 1 Mbps. Essi vengono convertiti in simboli, che rappresentano 2 bit, e trasmessi sul canale ad 1 Mbps. Mediante questo stratagemma, la velocità di canale rimane uguale ma il throughput raddoppia.

In questo capitolo sono stati forniti alcuni dettagli sul protocollo IEEE 802.11 ed sui vari meccanismi usati fin da quando lo standard è nato. Tuttavia negli anni è stato studiato a fondo ed ha subito cambiamenti che riguardano sia il livello PHY che il livello MAC (negli ambiti delle prestazioni, della qualità del servizio, della sicurezza, del consumo). Molte funzionalità e sopratutto molti meccanismi di comunicazione sono stati introdotti nelle versioni successive per supportare diversi tipi di applicazioni. Nel capitolo successivo quest'evoluzione verrà analizzata nel dettaglio.

Evoluzioni dello standard IEEE 802.11

802.11a

Lo standard 802.11a è stato introdotto nel 1999 sfruttando di base le caratteristiche del suo predecessore, ossia l'802.11 legacy, ma apportando delle modifiche sopratutto legate al livello PHY [3]. Le novità introdotte sono state quelle di operare in un range di frequenze diverso, utilizzando la banda ISM 5 Ghz e non più la banda ISM 2,4 Ghz, nonché di utilizzare una nuova modulazione che si basa sul Orthogonal Frequency Division Multiplexing (OFDM) (è stato il primo standard ad utilizzare OFDM orientato ai pacchetti). Cosi facendo sono stati raggiunti notevoli miglioramenti sia per quanto riguarda il Raw Data Rate (RDR) che il throughput effettivo (54 Mbps per RDR e 20Mbps per il throughput). La velocità di trasmissione è regolabile in base alle esigenze del protocollo e alle caratteristiche del mezzo trasmissivo (oltre ai 54 Mbps sono utilizzabili 48 Mbps, 36 Mbps, 24 Mbps, 18 Mbps, 12 Mbps, 9 Mbps e 6 Mbps). In molti paesi è stato concesso di operare nelle bande di frequenza tra i 5,470 Ghz e 5,725 Ghz, e questo ha portato all'aumento della capacità e del numero di canali wireless utilizzabili nella banda ISM 5 Ghz (802.11a ha a disposizione 12 canali non sovrapposti, 8 per le comunicazioni indoor e 4 per le comunicazioni punto
-punto). Questo standard non è interoperabile con l'802.11b poiché lavorano in bande di frequenza differenti. Tuttavia gli AP moderni, per sopperire a questo problema, sono in grado di operare in multibanda (sia ISM 2,4 Ghz che ISM 5 Ghz). Il vantaggi principale di utilizzare la banda dei 5 Ghz è l'aumento significativo di affidabilità delle connessioni (dato che ci sono più canali utilizzabili dai dispositivi ed il numero di dispositivo che li usano sono inferiori, il numero di conflitti sulla rete diminuisce). Naturalmente ci sono anche degli svantaggi da tenere in considerazione come la diminuzione della copertura di rete (la diminuzione dell'intensità di segnale è direttamente proporzionale al quadrato della sua frequenza). Ecco perché per cercare di contrastare questo effetto di perdita del segnale è stata introdotta la modulazione OFDM che permette la costruzione di antenne più piccole e con guadagno più elevato.

L'adozione della tecnica OFDM ha portato notevoli vantaggi anche in ambito di affidabilità e robustezza della connessione. Questo tipo di modulazione utilizza 52 portanti a frequenze ravvicinate, ciascuna modulata con un basso data rate. Di queste, 48 sono utilizzate per i dati e 4 sono utilizzate come portantipilota. Ciascuna portante è separata dalle sue adiacenti da una distanza di 0,3125Mhz (corrispondente alla divisione tra 20MHz e 64) arrivando ad occupare cosi 16,6Mhz (OFDM utilizza in maniera ottimale lo spettro di frequenze). La parte restante (3,4Mhz) viene utilizzata come distanza di guardia intercanale. Le interferenze tra onde con frequenze simili vengono evitate grazie al fatto che i segnali uscenti vengono ortogonalizzati (non possono interferire). Il fatto che i dati siano condivisi tra tutti i segnali uscenti, rende la connessione robusta alla dissolvenza selettiva (solo di alcune frequenze). La modulazione di ciascuna portante viene effettuata seguendo tecniche di codifica dipendenti dal data rate che si vuole raggiungere (BPSK, QPSK, 16QAM e 64 QAM). Ad esempio per data rate di 6Mbps viene utilizzata la modulazione Binary Phase Shift Keying (BPSK) con rate di codifica 0,5 milioni di simboli per secondo. Per data rate crescenti vengono utilizzate le stesse modulazioni con rate di codifica superiori, oppure viene cambiata la modulazione (da BPSK si passa a QPSK che raddoppia il data rate, oppure a 16QAM e 64QAM con data rate ancora più alti). I segnali vengono elaborati per via digitale (codifica digitale) e poi convertiti in forma d'onda analogica o viceversa (nel caso della ricezione). Questo implica l'utilizzo di convertitori analogicodigitale in ricezione e di convertitori digitaleanalogico in trasmissione. Le portanti vengono rappresentate da numeri complessi (detti fasori) nel dominio della frequenza (le elaborazioni vengono fatte nel dominio della frequenza) e vengono convertiti in segnali temporali da trasmettere eseguendo una IFFT (Inverse Fast Fourier Trasform). Per riottenere i coefficienti iniziali, le portanti rilevate dalla scheda vengono convertite nel dominio della frequenza mediante FFT e processati per via digitale.

L'802.11a è stato poco adottato nelle reti private dato l'elevato costo di produzione della parte di elaborazione analogica e Digital Signal Processing (DSP), ma in reti aziendali, dove sono richieste prestazioni ed affidabilità più elevate, ha riscontrato un notevole successo.

802.11b

La prima variante del protocollo IEEE 802.11 legacy è stata introdotta nel 1997 sotto il nome di 802.11b. Lo standard definisce le specifiche di livello PHY e MAC per l'implementazione di reti WLAN a 2,4 Ghz. Le differenze principali rispetto al suo predecessore sono:

- La possibilità di variare il data rate (in base alla qualità del canale) in un range più ampio (da 1 Mbps fino ad 11 Mbps) mantenendo la compatibilità con l'802.11 classico.

- la scelta automatica della banda di trasmissione meno occupata.
- la scelta automatica dell'AP di riferimento in base alla potenza del segnale ricevuto e del traffico di rete.
- la possibilità di creare un certo numero di aree parzialmente sovrapposte così da consentire il roaming e renderlo trasparente.
- la possibilità di variare il data rate in base alla qualità del canale (ARS-Adaptive Rate Selection).

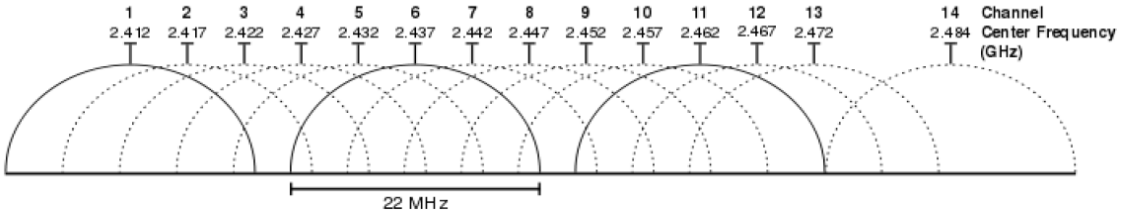

Figura 10. Canali WiFi disponibili nella banda a 2,4 Ghz. Si nota come sono sovrapposte e quali set di canali sono utilizzabili senza overlapping. Fonte: [4]

Lo standard ha riscosso un notevole successo tra le industrie leader nel settore telecomunicazionistico (Nokia, 3Com, Apple, Cisco System, Intersil, Compaq, IBM) tanto che nel 1999 è stata fondata la Wireless Ethernet Compatibility Alliance (WEGA) per certificare l'interoperabilità e la compatibilità tra i prodotti che implementano 802.11b.

Le frequenze operative utilizzate appartengono alla banda 2,4Ghz, come mostrato in figura 10. Per evitare l'overlapping dei canali, vengono scelti dei subset di canali adatti a trasmettere le informazioni necessarie, per evitare perdite significative di informazione, dovute ad eventuali interferenze mutue di canali poco disgiunti.

Per quanto riguarda il livello PHY, è stata introdotta una nuova tecnica di modulazione (rispetto ai classici FHSS ed DSSS dell'802.11) chiamata Complementary Code Keying Direct Sequence Spread Spectrum (CCKDSSS). Questo tipo di modulazione estende le caratteristiche di base delle trasmissioni DSSS del protocollo IEEE 802.11, aggiungendo una tecnica di multiplazione chiamata CCK. Essa si basa su Code Division Multiple Access (CDMA), ovvero una tecnica di multiplazione di canale che utilizza un codice di canalizzazione diverso per ciascun flusso di bit appartenente ad una destinazione diversa. Ciascun bit del flusso viene moltiplicato per un codice che ha un tempo di bit ridotto (tipicamente 1/N dove N è il numero di flussi da codificare in parallelo). Essendo ciascun codice ortogonale all'altro, si

riesce ad evitare interferenze tra i flussi di bit originali. La divisione di codice permette dunque l'utilizzo di tutta la banda, senza limiti di tempo ed inoltre aggiunge un livello di sicurezza della connessione poiché solamente chi conosce il codice di canalizzazione corretto può recuperare la sequenza di bit originale.

Entrando in dettaglio del funzionamento di CCK i bit di informazione vengono raggruppati da un convertitore serie/parallelo in gruppi da N. Ad ogni blocco viene associata una parola di codice composta da K simboli appartenenti ad un alfabeto Lario, ciascuno dei quali viene trasmesso in maniera sequenziale, utilizzando un'opportuna modulazione Laria senza memoria. Supponendo di raggruppare i bit in gruppi da 8 (N=8), di utilizzare parole di codice formate da 8 chip (K=8), e di codificare ciascun chip con un alfabeto a 4 valori (L=4), abbiamo a disposizione $L\hat{K}$ = 65.536 possibili parole di codice (sequenze di chip diverse), ma dovendo codificare solamente blocchi da 8 bit, scegliamo le 256 parole ortogonali tra loro in maniera da ridurre gli effetti di interferenza. Come tecnica di modulazione l'802.11b si basa su QPSK con blocchi da 8 bit e parole di codice composte da 8 chip.

Il sistema di accesso al mezzo del protocollo 802.11b si basa sul già noto CSMA/CA, dunque è stato mantenuto lo stesso protocollo MAC rispetto ad IEEE 802.11. Ricordandone il funzionamento: nel momento della trasmissione un nodo ascolta il canale per un certo tempo ed eventualmente trasmette i dati aspettando un acknowledgement di ritorno. Se l'ACK non arriva entro un certo tempo, viene rilevata la collisione ed il nodo attende un numero random di istanti di tempo per riaccedere al canale e ritrasmettere i dati in caso di canale libero.

Come si nota, il protocollo CSMA/CA aggiunge un overhead temporale notevole dovuto a politiche di condivisione che richiedono scambio aggiuntivo di messaggi ed attese da parte di ciascun nodo. Un protocollo di accesso al mezzo debole, porta ad una mancanza di sfruttamento completo della banda e quindi della velocità di trasmissione raggiunta. Oltre al CSMA/CA (di livello datalink) l'overhead di altri protocolli superiori, portano ad un decremento del data rate effettivo. Infatti le prestazioni di applicazioni su rete realtime sono nettamente inferiori. Anche in condizioni di segnale ottimo e bassa interferenza, il massimo data rate raggiungibile su connessioni TCP è 5,9 Mbps. Questo è dovuto all'uso sia del CSMA/CA dove il sistema deve attendere la liberazione del canale prima della trasmissione effettiva che all'overhead di trasmissioni di frame di dati e del controllo di congestione del TCP stesso. Con l'uso di UDP, che ha un header ridotto e non è dotato di meccanismi di controllo del canale come il TCP, si raggiungono prestazioni attorno a 7,1 Mbps.

Da quando è stato introdotto questo protocollo, le case costruttrici di sistemi di comunicazione WiFi hanno apportato delle modifiche legate alle velocità di trasmissione supportate, maggiori rispetto all'802.11b classico. Visto che gli standard successivi supportano velocità maggiori, sono state apportate delle modifiche proprietarie allo standard 802.11b introducendo le velocità 22 Mbps, 33 Mbps, 44 Mbps. Questo protocollo proprietario viene chiamato 802.11b+, tuttavia l'IEEE

non lo ha mai riconosciuto ufficialmente. Infatti sono stati sviluppate versioni ufficiali successive all'802.11b che introducono tecniche di modulazione e di accesso al canale migliori ed offrono funzionalità aggiuntive molto utili alle comunicazioni WiFi.

802.11g

Lo standard IEEE 802.11g è stato introdotto nel 2003 per raggruppare in un unico protocollo le funzionalità dell'802.11a e l'802.11b. Il primo di questi standard (802.11a) utilizza la modulazione OFDM che incrementa notevolmente le prestazioni, portando la velocità di trasmissione a 54 Mbps nella banda ISM 2,4Ghz. Il secondo introduceva delle velocità aggiuntive a quelle di 1Mbps e 2Mbps, ovvero 5.5Mbps e 11Mbps lasciando inalterati la larghezza di banda per canale, il numero di canali ed il protocollo di accesso al mezzo condiviso. Il nuovo standard ha esteso le funzionalità di 802.11a anche nella banda di 802.11b, così da restare compatibile con questo ed aumentare il throughput nella banda ISM 2,4 Ghz. E' noto che all'aumentare della frequenza di trasmissione, aumentano anche gli effetti interferenti come il multipath fading e gli effetti di assorbimento sugli ostacoli. Proprio per questo è preferibile 802.11g rispetto a 802.11a, poiché a parità di velocità di trasmissione gli effetti interferenti sono minori. Tuttavia, a causa del sovraffollamento delle frequenze interessate, l'ITU ha imposto l'utilizzo della modulazione Spread Spectrum (SS) per le trasmissioni in banda ISM. Con questa modulazione si trasmette il segnale su una banda molto più estesa del necessario, in modo che sembri rumore ai dispositivi non interessati. Le implementazioni più utilizzate sono l'FHSS e l'DSSS.

L'FHSS divide la banda in 70 canali dalla larghezza di 1Mhz. Durante la trasmissione di una trama la frequenza della portante viene modificata, passando da un canale all'altro (hopping) in base ad uno schema di salto conosciuto sia dal mittente che dal ricevente. Questo schema è unico per tutte le coppie mittente-ricevente che utilizzano lo stesso insieme di canali. Questo meccanismo riduce la probabilità che due trasmettitori utilizzino la stessa sottobanda contemporaneamente. Tuttavia, questo continuo hopping dei canali, causa un aumento del flusso dati ed introduce un ritardo che rende questa tecnica di modulazione piuttosto lenta.

L'DSSS suddivide la banda in 14 canali da 22 Mhz parzialmente sovrapposti. Quindi solamente un sottoinsieme di essi può essere utilizzato contemporaneamente. In questa modulazione ogni bit viene trasmesso tramite una sequenza fissa di valori chiamati 'chip'. Maggiore è la durata del chip, maggiore è la probabilità di recupero del dato. Ad esempio nel DSSS utilizzato da 802.11b ogni bit viene codificato da una sequenza ridondante di 11 bit detta sequenza di Barker, così da favorire il riassemblaggio del flusso di dati originale. Il DSSS non esegue nessun salto in frequenza e quindi viene usato un solo canale di trasmissione. Questa modulazione

viene utilizzata per segnali deboli, perché il segnale trasmesso occuperà una larghezza di banda superiore, infatti riduce notevolmente l'interferenza tra simboli, ma soffre di problemi d'interferenza tra canali adiacenti.

La Ortogonal Frequency Division Multiplexing (OFDM) permette di partizionare il flusso dei dati e di trasmetterli contemporaneamente su frequenze portanti differenti, ortogonali tra loro. Le modulazioni per ciascuna porzione di canale sono indipendenti, così da poter essere diverse. Questo tipo di tecnica incrementa la la velocità di trasmissione a 54 Mbps.

I cambiamenti apportati dallo standard IEEE 802.11g interessano lo strato PHY e MAC. Precisamente sono:

- La differenziazione in 4 strati del livello PHY.
- Supporto del preambolo corto.
- Meccanismi di protezione riguardo l'interoperabilità dei vari standard.

Sono stati introdotti (grazie alla specifica ERP-Extended Rate Physicals) 4 schemi di modulazione coesistenti (che codificano i dati in 4 modi diversi) che riguardano lo strato PHY:

- ERP-DSSS/CCK modulo fisico che implementa lo stesso strato fisico di 802.11b. Usa il DSSS e la sequenza CCK per codificare i bit.
- ERP-OFDM modulo fisico che implementa lo strato fisico di 802.11a. Usa la modulazione OFDM.
- ERP-DSSS/PBCC modulo fisico che implementa la modulazione DSSS mediante codifica Packet Binary Convolution Coding. Questo consente velocità aggiuntive (22 Mbps e 33 Mbps)
- DSSS-OFDM modulo fisico che implementa sia DSSS (utilizzato per la trasmissione del header) che OFDM (utilizzato per la trasmissione del PDU).

Nella versione IEEE 802.11b è stato rilevato che l'overhead più significativo era rappresentato dal header PLCP e dal preambolo della trama. Per queste ragioni è stato introdotto il preambolo corto, come mostrato in figura 11. Mentre il preambolo lungo, come mostrato in figura 12, è stato mantenuto per questioni di retrocompatibilità. Anche in IEEE 802.11g viene gestito questo tipo di preambolo e per ragioni di compatibilità i dispositivi IEEE 802.11g implementano 4 formati diversi di composizione della trama PLCP. Ognuno di essi viene utilizzato dai 4 strati fisici discussi sopra:

- Preambolo lungo - Questo corrisponde precisamente al frame PLCP con preambolo IEEE 802.11b (campo SYNC 128bit etc). Il preambolo e l'header vengono trasmessi ad 1 Mbps mentre il PDU viene trasmesso alle velocità 2/5,5/11 Mbps.

- Preambolo corto - Questo tipo di preambolo riduce il campo SYNC a 56 bit mantenendo inalterato il campo SDF. Le velocità di trasmissione sono diverse per ciascun campo. SYNC ed SDF vengono trasmessi ad 1 Mbps mentre il l'header PLCP è trasmesso a 2 Mbps mediante DQPSK. Il PDU può essere trasmesso a diverse velocità (2/5,5/11 Mbps).

- Preambolo ERP-OFDM - In questo tipo di preambolo si è mantenuto lo schema di 802.11a. Viene usato solamente quando tutte le STA sono in grado di comunicare mediante OFDM.

- Preambolo DSSS-OFDM - Questo tipo di preambolo dipende dal tipo scelto (preambolo corto o lungo).

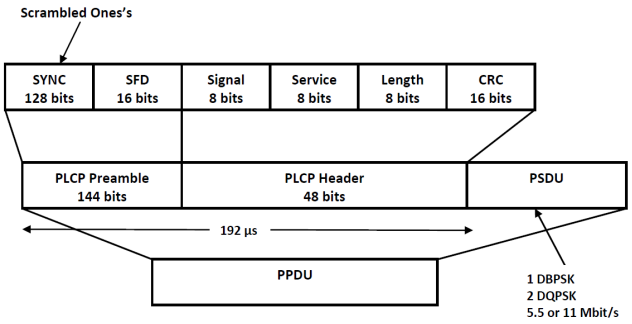

Figura 11. Struttura di un pacchetto PPDU con preambolo PHY lungo in cui il campo SYNC è lungo 128bit. Inoltre, poiché il preambolo va inviato ad un rate più basso, l'overhead temporale è maggiore. Fonte: [5]

Avendo a disposizione più di una modalità di funzionamento e più di una possibile velocità di trasmissione, lo scenario creato dal protocollo IEEE 802.11g è molto variegato. Possono esistere diversi tipi di dispositivi (in base alla modalità scelta). A causa di questo, sono stati sviluppati meccanismi di mantenimento della retrocompatibilità come il CTS-TO-Self e sono state adottate tecniche che usano i precedenti tipi di messaggi esistenti già nelle versioni passate dello standard. Mantenendo la retocompatibilità con formati di messaggi precedenti, in aggiunta ad essi sono stati introdotti da IEEE 802.11g dei nuovi tipi di messaggi, caratterizzati da diversi tipi di modulazioni per ciascun campo del frame da trasmettere o ricevere. Essi, come rappresentati nelle figure 13, 14, 15, sono messaggi contenenti vari tipi

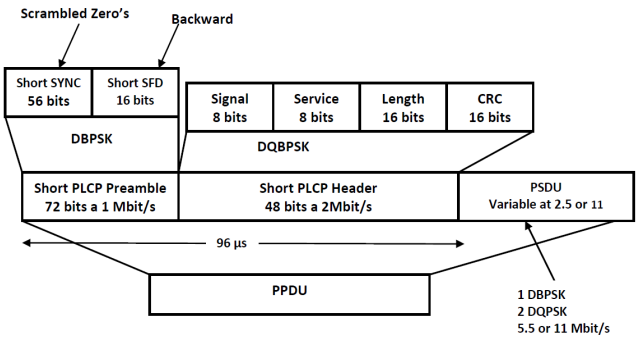

Figura 12. Struttura di un pacchetto PPDU con preambolo PHY corto in cui il campo SYNC è lungo 56 bit. Fonte: [5]

di preamboli (lungo o corto), e ciascun campo è rappresentato da campioni di segnali modulati con varie tecniche di modulazione, la cui più importante ed efficace è OFDM.

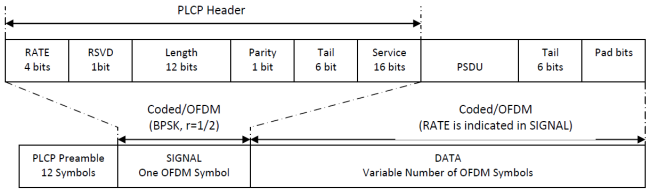

Figura 13. Struttura di una PPDU implementata dal modulo ERP-OFDM introdotto in IEEE 802.11g. Come si vede l'header PLCP è rappresentato da un simbolo OFDM modulato tramite BPSK, mentre il preambolo da 12 simboli OFDM. La parte di payload restante viene rappresentata da un numero di simboli variabili in base alla lunghezza del pacchetto. Fonte: [5]

Ad esempio, l'uso dello strato fisico DSSS-OFDM consente alle STA non-OFDM di rilevare l'occupazione del canale. L'uso dei messaggi CTS/RTS in DSSS fa si che anche una STA non-OFDM riesca a capire se il canale è occupato oppure no. Un meccanismo completamente nuovo come il CTS-to-Self è stato introdotto in IEEE 802.11g. Questa tecnica evita la collisione per incompatibilità DSSS/OFDM (cioè non risolve il problema del terminale nascosto, ma riduce l'overhead rispetto a RTS/CTS perché viene trasmesso 1 solo pacchetto anzicchè 2). La stazione che trasmette avvisa mediante CTS le altre STA della volontà di trasmettere (questo però non viene ascoltato dai terminali nascosti). A differenza del meccanismo CTS/RTS, anziché inviare il frame RTS prima della trasmissione, viene inviato un CTS che viene ricevuto solamente dai terminali nel raggio di trasmissione. La stazione interessata non risponderà con il CTS (non è previsto dal protocollo) mentre

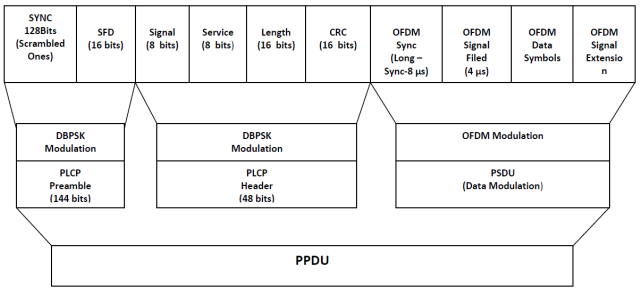

Figura 14. Struttura di una PPDU con preambolo lungo, implementata dal modulo opzionale ERP-DSSS-OFDM introdotto in IEEE 802.11g. Come si può vedere il preambolo viene inviato mediante modulazione DBPSK, così come l'header PLCP, mentre per i dati è utilizzata la modulazione OFDM. Fonte: [5]

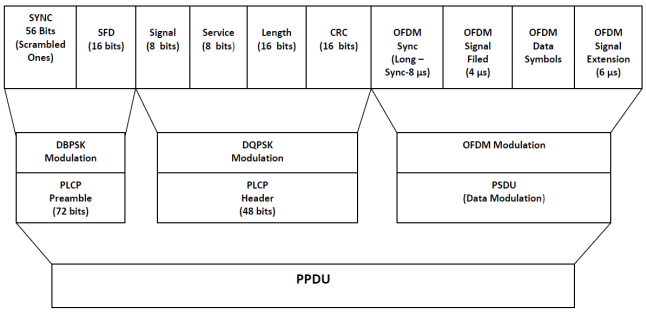

Figura 15. Struttura di una PPDU con preambolo corto, implementata dal modulo opzionale ERP-DSSS-OFDM introdotto da IEEE 802.11g. Come si può vedere il preambolo viene inviato mediante modulazione DBPSK, mentre l'header PLCP con modulazione DQPSK ed i dati con la modulazione OFDM. Fonte: [5]

quelle non indirizzate vengono avvisate dell'occupazione del canale. Tuttavia, i terminali nascosti che non ricevono l'unico CTS inviato, possono collidere sul canale. Dunque questa tecnica si utilizza solamente quando non vi sono possibili terminali nascosti da gestire.

802.11n

Lo standard IEEE 802.11n è stato approvato nel 2009 ed ha introdotto notevoli miglioramenti sia a livello PHY che MAC. Un'innovazione sostanziale è il meccanismo dual band (il protocollo può operare sia nella banda ISM 2,4Ghz che 5Ghz). Questo ha reso il protocollo retrocompatibile con la famiglia di protocolli IEEE 802.11a/b/g. Infatti supporta le modulazioni DSSS, OFDM e CCK a livello PHY,

offre data rate più elevati (600 Mbps), una maggiore portata, affidabilità e raggio di copertura. I miglioramenti sono stati raggiunti grazie all'introduzione delle seguenti caratteristiche aggiuntive:

- Modulazione a livello PHY chiamata Multiple Input Multiple Output (MIMO).
- Larghezza di canale flessibile, con l'aggiunta dell'ampiezza 40Mhz.
- Frame Aggregation a livello MAC.
- Block Acknowledgement (BA) sempre a livello MAC.

La tecnica MIMO, come mostrato in figura 16, consiste nell'uso simultaneo di antenne multiple per la trasmissione e ricezione. Questa strategia permette la suddivisione del flusso dati in 'spatial streams' (flussi spaziali) indipendenti dagli altri, che vengono trasmessi e ricevuti da antenne diverse, processati e riassemblati per consentire la lettura completa del frame. Questa tecnica sfrutta a proprio vantaggio il multipath fading. In contesti indoor la comunicazione radio non rispetta la propagazione LOS (Line Of Sight - percorso diretto dal trasmettitore al ricevitore) a causa degli ostacoli trovati dall'onda elettromagnetica sul percorso che cambiano la traiettoria del segnale. Quando il segnale impatta su un oggetto, subisce una riflessione (una componente del segnale viene riflessa - dipende dalla frequenza) che obbliga la componente riflessa a seguire un percorso alternativo a quello della componente principale, come mostrato in figura 17. In definitiva, il destinatario riceve diversi segnali (provenienti dallo stesso trasmettitore) con fasi e ritardi differenti che portano alla distorsione ed attenuazione del segnale originale.

I parametri utilizzati dal collegamento MIMO sono N (numero di antenne di ricezione) ed M (numero di antenne di trasmissione) e sono configurabili da N=1 ed M=1 (1x1) fino ad N=4 ed M=4 (4x4) in NxM modi differenti.

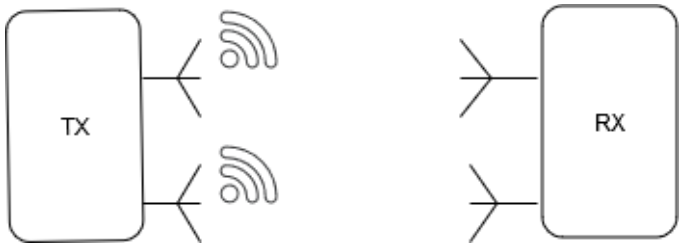

Figura 16. Architettura PHY MIMO 2x2 di un ricevitore e di un trasmettitore IEEE 802.11n.

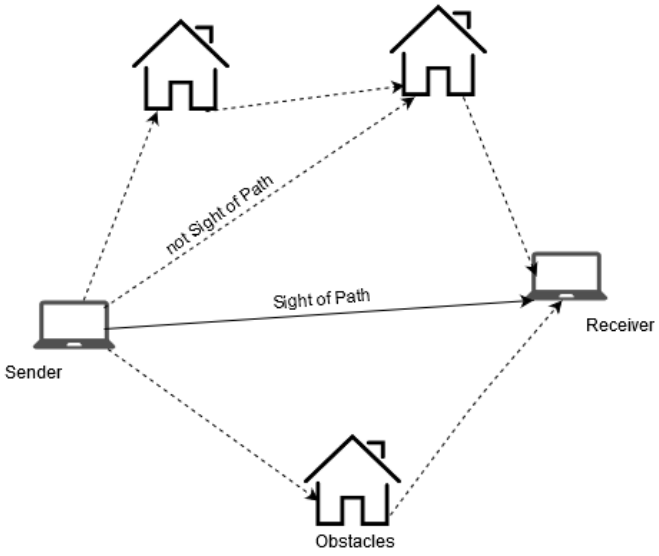

Figura 17. Scambio di messaggi tra un trasmettitore ed un ricevitore, alterato da interferenti generate dal multipath fading. Come si può osservare, il percorso è alterato ed inoltre i segnali interferenti che si generano non sono unici, ma vi è una potenziale generazione di copie del segnale ad ogni impatto con un oggetto.

Le modalità di utilizzo del MIMO non è specificata poiché dipendono dall'applicazione e sopratutto dal canale fisico (vengono tipicamente utilizzate per incrementare il throughout). Tuttavia la tecnologia MIMO, come mostrato in figura 18, permette un utilizzo flessibile delle antenne multiple. Infatti, lo strato fisico MIMO supporta diverse elaborazioni del segnale, le quali possono coesistere:

- Space Time Block Coding (STBC)
- Maximum Ratio Combining (MRC)
- Spatial Division Multiplexing (SDM)
- Trasmit Beamforming (TxBF)

Sia STBC che MRC riguardano la trasmissione e la ricezione di uno stesso flusso dati attraverso diverse antenne ed agiscono per migliorare l'affidabilità della comunicazione, il rapporto segnale/rumore (SNR) e per minimizzare gli effetti del fading.

STBC viene utilizzato quando il numero di antenne di trasmissione è maggiore di quelle di ricezione. Vengono inviati più copie dello stesso flusso in tempi successivi, su diverse antenne (ridondanza spazio-temporale). Il ricevitore riesce a ricostruire il flusso originale anche in presenza di interferenze e distorsioni, grazie

al riordinamento dei dati ricevuti in sequenza e l'eventuale correzione (grazie ai CRC).

MRC viene applicato a comunicazioni in cui il numero di antenne di ricezione supera il numero di antenne di trasmissione. Questa funzione del ricevitore permette di ricostruire una copia, più fedele possibile, del segnale originario a partire dalle combinazioni dei segnali ricevuti.

SDM viene utilizzato dai dispositivi MIMO per incrementare il data rate e si riferisce alla ricezione ed invio di flussi diversi su diverse antenne. In questo contesto, le antenne riceventi sentono un campo risultante dato dalla sovrapposizione di tutti i segnali trasmessi (con le relative copie riflesse) con cammini differenti. Se i dati, appartenenti a flussi diversi, giungono al ricevitore da cammini indipendenti (cammini sufficientemente distinti nello spazio), possono essere ricostruiti tenendo conto di tutti i segnali ricevuti su tutte le antenne. Questa tecnica è utilizzabile quando il numero di flussi inviati parallelamente è minore del numero di antenne riceventi. Lo standard IEEE 802.11n impone l'implementazione delle modalità MIMO 2x2 (doppio bit rate) e 4x4 (quadruplo bit rate).

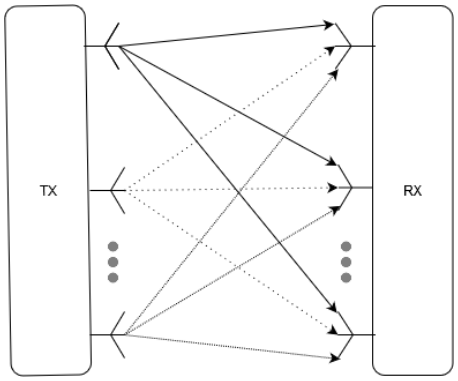

Figura 18. Architettura PHY MIMO NXN. Come si vede il segnale ricevuto da una singola antenna dipende dalla somma del segnale diretto e dei segnali riflessi che arrivano come disturbo. Tuttavia utilizzando le antenne in questo modo è possibile dividere il flusso iniziale in N flussi ed inviarli contemporaneamente sulla stessa banda, aumentando cosi la velocità effettiva.

TxBF è l'ultima novità introdotta, a livello PHY, dal protocollo IEEE 802.11n. Essa permette di ottimizzare il consumo dell'energia, direzionando a piacimento nello spazio la potenza irradiata, in modo da focalizzare maggiore energia nel punto di ricezione dell'antenna ricevente. Il Beamforming sfrutta la riflessione ed il multipath per migliorare la potenza del segnale ricevuto ed il data rate.

La tecnologia MIMO abbinata alla modulazione OFDM (multi-portante) permette di separare le operazioni di equalizzazione del canale e di decodifica dei dati, grazie all'eliminazione della distorsione introdotta dai cammini multipli (OFDM) ed alla possibilità di decodificare separatamente il segnale MIMO su portanti diverse.

Un'altra innovazione apportata dallo standard consiste nel cosiddetto Channel Bonding che permette l'utilizzo di canali ampi 40Mhz, oltre che di canali da 20/22Mhz, come nelle precedenti versioni. Questa caratteristica viene implementata fondendo due canali adiacenti (ciascuno da 20Mhz) che vengono utilizzati contemporaneamente come fosse uno, come mostrato in figura 19. Questo meccanismo consente di raddoppiare il bit rate poiché raddoppia la quantità di informazione per unità di tempo. L'efficienza massima viene raggiunta sulla banda ISM 5Ghz a causa del numero maggiore di canali disponibili, mentre nella banda ISM 2,4Ghz (tipicamente più sfruttata) la probabilità di interferenza è maggiore. In particolare, per effettuare una trasmissione occorre che il mezzo trasmissivo sia rilevato IDLE per entrambi i canali da 20Mhz. Questo, in caso di traffico, porta in genere ad un aumento delle latenze. Il Channel Bonding è opzionale ma in pratica viene implementato da tutti i dispositivi 802.11n.

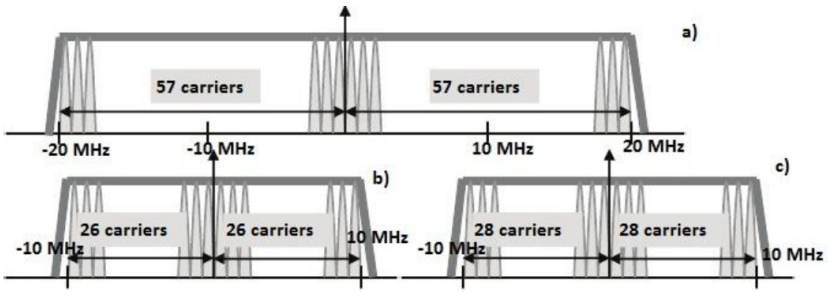

Figura 19. Divisione in sottoportanti della banda utilizzata da OFDM. Come si nota, il numero di sottoportanti del canale a 40Mhz sono meno rispetto a quelle da 20Mhz e questo perché le sottoportanti devono essere distanziate l'una dall'altra per non creare interferenza. Inoltre a) 114 subcarriers (108 utilizzabili) per un canale largo 40Mhz (IEEE 802.11n); b) 52 subcarriers (48 utilizzabili) per un canale largo 20Mhz (legacy 802.11a/g); c) 56 subcarriers (52 utilizzabili) per un canale largo 20Mhz (IEEE 802.11n). Fonte: [5]

Un ulteriore incremento della velocità di trasmissione può essere ottenuto grazie all'aumento del numero di portanti OFDM per le trasmissioni. Grazie ad 802.11n è possibile usufruire di un insieme maggiore di data rate rispetto ad 802.11a/g.

Tutti i miglioramenti discussi fin'ora riguardano unicamente il livello PHY. Per quanto riguarda il livello MAC, le modifiche apportate sono essenzialmente due: Packet Aggregation e Block Acknoledgement (BA).

Ciascun frame inviato ha un overhead dovuto essenzialmente al Preambolo PHY, all'header PLCP, all'header MAC ed ai campi di controllo ACK e FCS. Per limitarne gli effetti, incrementando l'efficienza protocollare, IEEE 802.11n implementa il Frame Aggregation all'interno del processo di invio, come mostrato in figura 20. I frame che devono essere trasmessi sono raggruppati in un unico frame.

In questo modo il frame complessivo contiene solamente un Preambolo PHY ed un header PLCP (ciò riduce l'overhead) ed ha un payload più corposo (contiene tutti i payload dei frame originali). Inoltre vengono ridotti nel complesso i tempi di contesa del canale e quelli di backoff nonché il numero di collisioni. Esistono due meccanismi di Frame Aggregation:

- Aggregation MAC Service Data Unit (A-MSDU) con dimensione massima di 8 Kbyte.
- Aggregation MAC Protocol Data Unit (A-MPDU) con dimensione massima di 64 Kbyte.

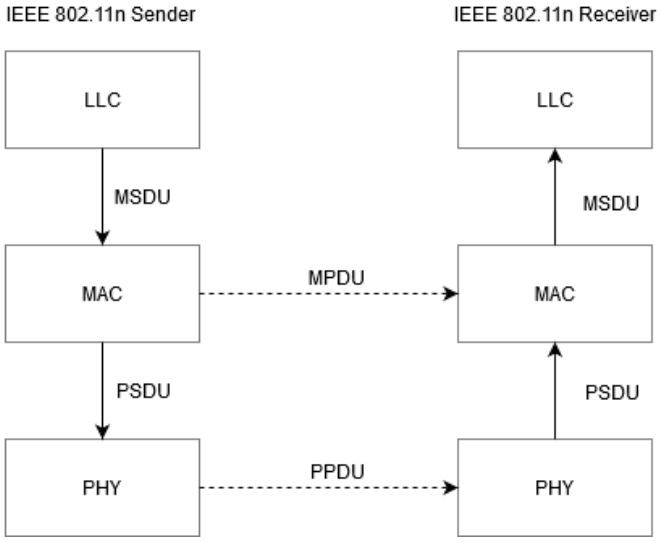

Figura 20. Scambi dei servizi, offerti fra i vari livelli dello stack protocollare. PPDU per le unità scambiate tra livelli PHY, MPDU per le unità scambiate tra livelli MAC, mentre per quanto riguarda le comunicazioni intra-device abbiamo MSDU per le unità scambiate tra il livello LLC e MAC, invece PSDU per le unità scambiate tra livello MAC e livello PHY.

Nel A-MSDU i pacchetti provenienti dai livelli superiori vengono mappati in un unico frame che viene visto come unico MPDU dal livello PHY (l'header MAC e l'FCS sono unici per tutti i pacchetti raggruppati). Ciascun frame elementare, come illustrato in figura 21, viene congiunto ad un sub-header che contiene l'indirizzo sorgente, destinazione e la lunghezza di ciascun sub-frame. Ogni sub-frame deve avere la stessa classe di servizio (altrimenti vengono persi i vantaggi di avere diverse frame-priority, poiché aspettare ulteriori frame da aggregare aumenta il ritardo di trasmissione) e sopratutto devono avere lo stesso indirizzo RA anche se possono avere indirizzi DA differenti (sarà il ricevitore a smistare i frame a destinazioni

diverse). Ogni sub-frame non contiene nessun checksum/crc, per cui eventuali ritrasmissioni selettive (dei soli sub-frame persi) non possono essere gestite. A-MSDU è obbligatorio per tutti i ricevitori 802.11n.

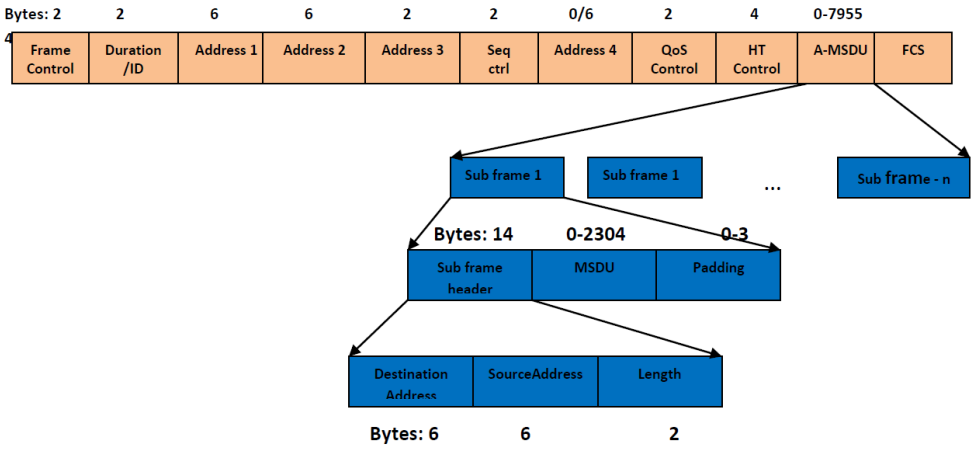

Figura 21. Metodo di aggregazione a livello LLC-MAC dove i singoli pacchetti provenienti dai livelli superiori vengono aggregati in un unico frame MAC MPDU. Come si vede ciascuna MSDU è contenuta in una MPDU. Tuttavia l'overhead dei dati è dovuto all'header aggiuntivo per ciascun MSDU che viene inserito per mantenere i pacchetti MSDU separati e riconoscibili (Sub Frame Header). Invece il lato positivo è che l'header MAC è unico. Fonte: [5]

Mentre A-MSDU opera a livello MAC, A-MPDU effettua l'aggregazione a livello PHY. Quest'ultimo tipo di aggregazione consiste nell'imbustamento di tante MPDU dentro un unico frame PHY (unico PPDU), come rappresentato in figura 22. Nel momento dell'aggregazione, ciascuna MPDU viene preceduta da un Delimiter che contiene i seguenti campi: Reserved, MPDU-Length, CRC ed Unique Pattern. In questa modalità d'aggregazione, l'AP può unire i frame aventi indirizzi SA diversi ma con lo stesso indirizzo RA ed inviarli come frame aggregato, mediante l'uso del BA. Inoltre a differenza dell'A-MSDU, i sub-frame possono essere ritrasmessi singolarmente (ognuno di essi ha un CRC dedicato).

Tra i vantaggi principali di A-MSDU c'è la sua efficacia sotto condizioni di canale ideale (canale non rumoroso) mentre l'A-MPDU si presta meglio ad ambienti rumorosi ma risulta meno veloce (il numero di frame aggregati è maggiore, di conseguenza l'attesa della trasmissione provoca un ritardo maggiore) nonostante il vantaggio prodotto dall'aggregazione. Quindi spesso si cerca un compromesso tra velocità e ritardo.

L'ultima innovazione introdotta dallo standard IEEE 802.11n è il BA. Esso riduce il numero di ACK che un ricevitore manda ad un trasmettitore una volta ricevuto il pacchetto. Ogni BA conferma la ricezione di più di un frame, cosi da

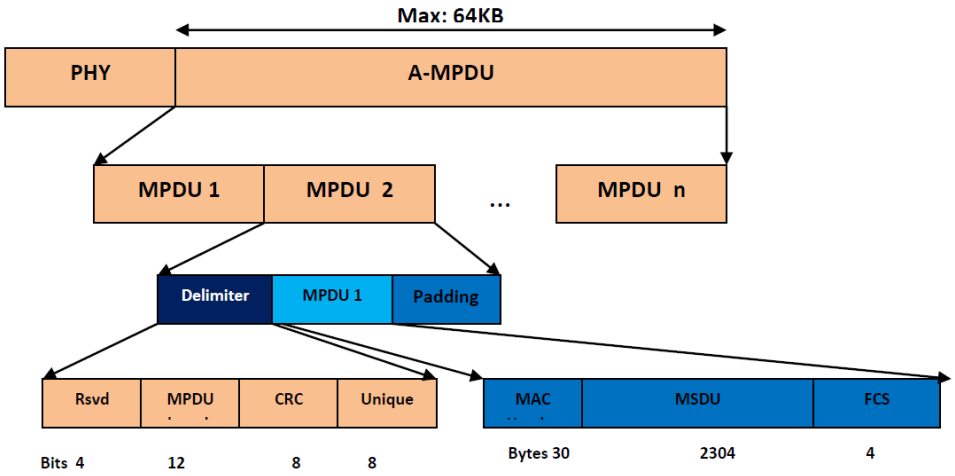

Figura 22. Metodo di aggregazione a livello MAC-PHY dove i singoli pacchetti provenienti dai livelli MAC (MPDU) vengono aggregati in un unico frame PHY PPDU. Come si vede ciascuna MPDU è contenuta in una PPDU. L'overhead dei dati è dovuto all'header aggiuntivo chiamato delimiter su ciascun MPDU e viene inserito per mantenere i pacchetti MPDU separati e riconoscibili (Delimiter). Invece il lato positivo è che l'header PHY è unico. Inoltre ciascuna MPDU è protetta da un proprio CRC separato da tutti gli altri. Fonte: [5]

ridurre l'overhead e l'occupazione del canale. Nel caso di A-MSDU non ci sono cambiamenti poiché i sub-frame vengono imbustati in un unico MPDU, mentre nell'A-MPDU ciascun sub-frame rappresenta un MPDU differente quindi necessita di una conferma per ciascun sub-frame. In questo caso, usando il BA, vengono aggregati tutti gli ACK riferiti a quel frame (contenente i sub-frame) e vengono trasmessi sulla rete come fosse un unico ACK. Questo meccanismo permette di gestire anche ritrasmissioni selettive (come detto sopra) dei frame A-MPDU.

In conclusione possiamo dire che IEEE 802.11n, grazie alla tecnologia MIMO ed alle modifiche apportate a livello MAC, porta dei notevoli benefici in termini di velocità di trasmissione, affidabilità, copertura e throughput.

802.11ac

L'evoluzione successiva di WiFi è rappresentata dalla versione 802.11ac dello standard (noto anche come VHTVeryHigh Throughput) che estende le funzionalità di 802.11n migliorandolo in termini di potenza, efficienza e retrocompatibilità nella banda ISM 5Ghz. Le esigenze di mercato hanno spinto verso dei cambiamenti rispetto a 802.11n. Ad esempio un aumento della velocità di trasmissione (fino a 1Gbps), della densità di bit per pacchetto, del bit rate per STA, dell'affidabilità del

collegamento AP-STA, dell'efficienza spettrale e della capacità dell'AP di trasmettere contemporaneamente dati a più STA. Per ottenere questi risultati sono state apportate rispettivamente le seguenti modifiche:

- Canali più larghi (velocità di trasmissione maggiore).
- Aumento della densità di codifica (densità di bit per pacchetto maggiore).
- Aumento del numero di flussi spaziali (bit rate per STA maggiore).
- Beamforming (affidabilità del canale AP-STA maggiore).
- Multi-User MIMO (maggiore numero di STA servite contemporaneamente).

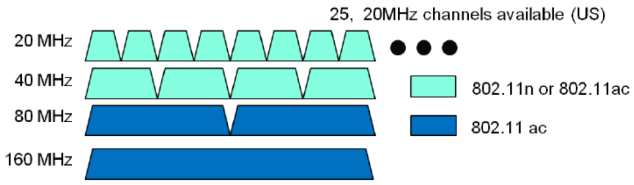

Figura 23. Suddivisione della banda in canali, più o meno ampi, mettendo in evidenzia le differenze con IEEE 802.11n (che non supporta i canali ad ampiezza 80Mhz + 80Mhz o 160Mhz). I canali più piccoli hanno diverse giunzioni in frequenza, dove non è possibile trasmettere, invece i canali più ampi soffrono meno di questo problema. Fonte: [5]

Le modifiche si riferiscono a due livelli: Radio (modulazioni, larghezze di canale, banda di frequenze, flussi spaziali multipli) e MIMO sul lato AP (MU-MIMO, Beamforming).

Come già evidenziato, la problematica principale della banda ISM 2,4Ghz è legata al suo congestionamento dovuto alla presenza di molti dispositivi che operano in tale banda. Ecco perché IEEE 802.11ac opera esclusivamente nell'intorno dei 5Ghz, cosi da evitare le interferenze ed aumentare la velocità di trasmissione. Rispetto alla versione IEEE 802.11n, offre la possibilità di avere canali più ampi (fino a 160Mhz) che portano a velocità più elevate. Le larghezze di canale obbligatorie sono 20Mhz, 40Mhz, 80Mhz mentre quella opzionale è solamente 160Mhz. I canali e le varie aggregazioni ottenibili sono mostrate in figura 23. Le ultime due larghezze di canale sono ottenute mediante Channel Bonding (fondendo rispettivamente due canali da 40Mhz adiacenti ma non sovrapposti, o da 80Mhz contigui o non contigui). Grazie a queste modifiche vengono raggiunti livelli di throughput maggiori, ma come conseguenza diminuiscono i canali non sovrapposti disponibili nella banda ISM 5Ghz.

In ambito modulazione, le codifiche obbligatorie (che garantiscono la retrocompatibilità con IEEE 802.11a ed IEEE 802.11n) sono OFDM, BPSK, QPSK, 16-QAM e 64-QAM mentre quella opzionale è data dalla modulazione 256-QAM (Quadrature Amplitude Modulation). Rispetto a 64-QAM (IEEE 802.11n) la modulazione 256-QAM permette un aumento del data rate del 33%, come mostrato in figura 24. Usando canali da 40Mhz a 4 flussi spaziali si raggiungono throughput vicini a 600 Mbps con una modulazione 64-QAM mentre con modulazione 256-QAM otteniamo livelli di throughput di 800 Mbps e questo perché le due modulazioni codificano un numero diverso di bit per simbolo. 64-QAM codifica 6 bit per simbolo mentre in 256-QAM i bit per simbolo sono 8, come mostrata dalla costellazione dei simboli in figura 25. Tuttavia uno svantaggio della modulazione 256-QAM è la richiesta di un rapporto segnale rumore maggiore poiché la differenza tra i simboli è minore rispetto alla codifica 64-QAM, dunque è più sensibili al rumore.

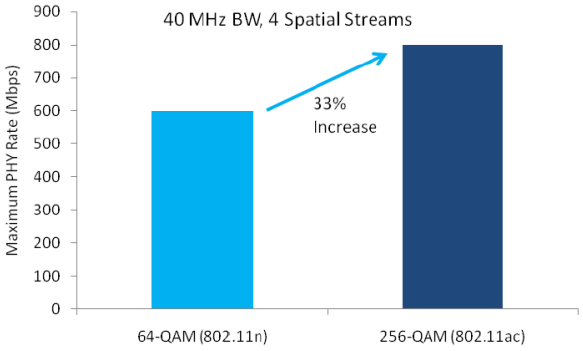

Figura 24. Differenza in termini di throughput tra la modulazione vecchia 64-QAM (IEEE 802.11n) e la modulazione evoluta 256-QAM (IEEE 802.11ac). Come si vede le prestazioni sono nettamente superiori. Fonte: [5]

Riguardo gli stream spaziali IEEE 802.11ac permette 8 flussi a differenza dei 4 permessi in IEEE 802.11n. L'implementazione di più flussi spaziali è opzionale tuttavia è molto utile se abbinata alla tecnica MU-MIMO che verrà illustrata nel seguito. Infatti il MU-MIMO consente di raddoppiare il traffico di rete grazie agli 8 stream disponibili (4 per stazione di destinazione).

MU-MIMO è una funzione aggiunta dal protocollo IEEE 802.11ac che permette ad un AP di migliorare la trasmissione verso le STA. Questa tecnica, come mostrato in figura 26, permettere di trasmettere 4 flussi indipendenti a 4 STA raggiungendo un throughput dell'ordine di 1Gbps. Per servire 4 diverse STA in IEEE 802.11n bisognava multiplexare i flussi servendone uno per volta riducendo il throughput di un fattore 4. Il funzionamento di MU-MIMO si basa sul concetto di MIMO IEEE 802.11 che trasmette 4 flussi ad una singola stazione. La differenza sta nell'utilizzo delle antenne poiché i flussi vengono rediretti su antenne diverse in base all'STA

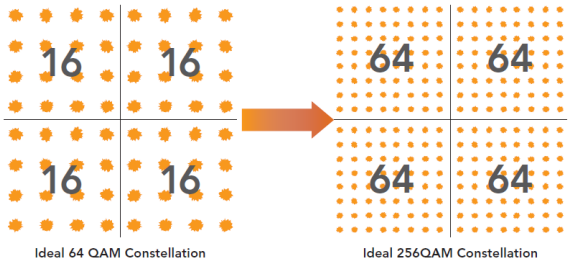

Figura 25. Costellazioni dei simboli delle modulazioni 64-QAM e 256-QAM. Fonte: [5]

presa in considerazione. Ad esempio in SU-MIMO l'AP trasmette 4 flussi spaziali e la STA ricevente deve avere 4 antenne su cui ricevere separatamente ciascun flusso, mentre in MU-MIMO, l'AP trasmette i 4 flussi spaziali (diretti ciascuno ad una STA diversa) contemporaneamente su 4 antenne differenti ed all'utente basta un'antenna per riceverne i dati. La tecnica MU-MIMO richiede un miglioramento del meccanismo di Beamforming (già introdotto in IEEE 802.11n) aumentando cosi la robustezza alle interferenze dei singoli canali. Grazie al miglioramento del beamforming, i flussi sul canale vengono gestiti dal trasmettitore (cioè viene controllata la fase del segnale) in modo da evitare interferenze distruttive dovute al multipath-fading.

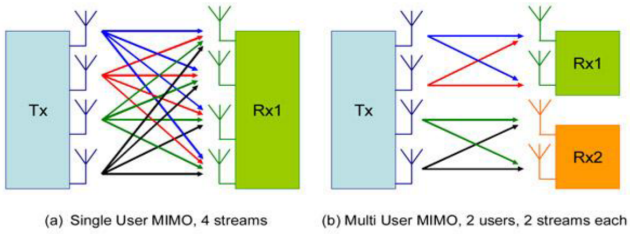

Figura 26. Utilizzo delle antenne multiple in MU/SU MIMO in IEEE 802.11ac a differenza di SU-MIMO in IEEE 802.11n. Nella prima modalità SU-MIMO i flussi spaziali sono diretti tutti verso unq STA, mentre nel caso MU-MIMO i flussi spaziali sono orientati verso due STA differenti per parallelizzare le comunicazioni. Fonte: [5]

L'IEEE 802.11ac definisce un proprio meccanismo di Beamforming in cui l'AP trasmette un pacchetto particolare denominato Sounding che contiene l'indirizzo dell'AP e quello dei destinatari. Questo passo viene eseguito per permettere una

stima del canale da parte dei singoli ricevitori, tenendo in conto dei possibili problemi nel percorso sorgente-destinazione. A canale stimato, ogni destinatario inoltra la stima fatta all'AP che riesce a sfruttare le informazioni per STA ricevute in modo da migliorare il flusso. Ogni destinatario misura il collegamento tra l'AP e se stesso in modo tale che l'AP riesca ad orientare il segnale in maniera ottimale evitando percorsi inefficienti, aumentando l'affidabilità, massimizzando la capacità e la copertura del link nonché riducendo l'interferenza inter-user (tra flussi appartenenti a STA differenti).

802.11ax

Il protocollo IEEE 802.11ax è stato ideato con lo scopo di migliorare il throughput in scenari ad alta densità di nodi. Inoltre sono state apportate delle modifiche a livello PHY, a livello MAC, nell'uso della tecnologia MU-MIMO e nella gestione degli stati energetici che lo rendono decisamente più efficiente rispetto al suo predecessore IEEE 802.11ac. Il continuo incremento delle velocità di trasmissione dei vari standard è stato reso possibile grazie a tecniche di modulazione e codifica sempre più avanzate, larghezze di canale maggiori, e l'adozione del MIMO. Sfortunatamente, dalle analisi del protocollo IEEE 802.11ac, si è visto che per aumentare la velocità di trasmissione non basta avere canali di trasmissione più larghi o un maggior numero di stream spaziali. Inoltre, il data rate nominale non costituisce a sé un buon indice di qualità della rete WiFi poiché esso dipende anche da eventuali interferenze ed attenuazioni su frequenze selettive, da inefficienze di accesso al mezzo nonché dallo scenario di rete. Il solo rate nominale non basta per soddisfare le richieste di applicazioni e servizi. Negli scenari di connessione ad elevata densità di rete (numero di terminali connessi per unità di superficie) la copertura è garantita tipicamente da AP multipli che lavorano su frequenze operative simili e su canali parzialmente sovrapposti. In ambienti di questo tipo, il throughput non è una metrica di performance corretta, piuttosto ci si concentra nell'aumentare la densità di throughput come ad esempio il throughput per area che è definito come il rapporto tra il throughput dell'intera rete e l'area di rete stessa. Il parametro più importante in queste applicazioni è l'interferenza di massa, su cui IEEE 802.11ax si concentra. Anziché vietare le trasmissioni che causano collisioni tra terminali nascosti, esso cerca di migliorare il riutilizzo spaziale per evitare l'esposizione della STA nascosta. In applicazioni reali, raramente le STA operano in modalità satura poiché l'overhead temporale per l'invio di una trama impatta in maniera significativa sul throughput (a parità di tempo di accesso al canale, la grandezza del payload è proporzionale al throughput) e quindi sulla user-experience. Un altro fattore influente è la diminuzione di asimmetria nei canali. Con l'avvento di applicazioni full-duplex (sfruttano sia canali di Download che di Upload) il traffico di rete sui canali UpLoad (UL) è in aumento. Lo standard IEEE 802.11ac si

preoccupava di migliorare le prestazioni sui canali DownLoad (DL) mediante uso di MU-MIMO (MU-MIMO richiede un'elevata sincronizzazione per gli UL) mentre lo standard IEEE 802.11ax cerca di migliorare anche le prestazioni sui canali UL. Per quanto riguarda le innovazioni introdotte in IEEE 802.11ax è stato introdotto un nuovo livello PHY con più modulazioni e codifiche disponibili. Inoltre, una maggiore efficienza spettarle unitamente al nuovo livello PHY, hanno migliorato notevolmente le prestazioni in termini di throughput. La caratteristica chiave di IEEE 802.11ax è l'adozione di una tecnica di accesso al canale chiamata Ortogonal Frequency Division Multiple Access (OFDMA) un approccio ampiamente utilizzato nelle reti cellulari, ma completamente nuovo in reti WiFi. La logica è che i canali molto ampi (80Mhz, 80 + 80Mhz e 160Mhz) introdotti in IEEE 802.11ac soffrono di interferenze selettive in frequenza e questo altera in modo significativo la qualità del servizio. Con OFDMA, le sottoportanti adiacenti (detti toni) sono raggruppate in un'unità di risorse chiamata Resource Unit (RU) e un mittente può scegliere la migliore RU per ciascun destinatario. Questa tecnica di accesso al canale migliora il Signal to Interference plus Noise Ration (SINR), lo schema di codifica (MCS) ed il throughput. Per superare il problema dell'abbassamento di throughput in modalità saturated, la STA che deve tramettere solo pochi dati può allocare RU più piccole. Le RU sono gestite dagli AP. Solitamente un AP può decidere di usare trasmissioni DL-DCF, MU-MIMO DL, MU-OFDMA, oppure ancora RU UL. Nello standard IEEE 802.11ax, OFDMA è frame-based (basato su frame) e ciò significa che un frame MU contiene dati da/per utenti diversi e vari toni sono assegnati agli utenti per l'intera durata del frame. Per una trasmissione DL MU, il preambolo PHY specifica la durata ed il mapping delle frequenze tra STAs. Al contrario, in una trasmissione UL MU, tali informazioni sono contenute nel frame precedente (frame trigger, controllo, dati). In figura 28 vi è un esempio di DL MU ed UL MU ACKs.

L'introduzione di OFDMA porta anzitutto alla modifica dei parametri dello schema di modulazione OFDM (per renderlo più adatto a supportare OFDMA). In secondo luogo alla modifica del formato dei pacchetti poiché il preambolo PHY dovrà contenere le informazioni di mapping delle frequenze OFDMA. Inoltre IEEE 802.11ax tenta di spostare alcune informazioni riguardanti il livello MAC al livello PHY seguendo il principio secondo il quale nelle trame completamente corrotte, il preambolo è decodificabile in ogni caso. In terzo luogo OFDMA comporta numerose modifiche MAC relative alla gestione delle trasmissioni MU per mantenere la compatibilità tra dispositivi di diverse generazioni. Molti altri sforzi sono stati compiuti per migliorare il rendimento e per ridurre il consumo di energia, come mostrato in figura 27, in reti fitte (con tanta sovrapposizione tra aree di connessione). Tra questi abbiamo:

- La colorazione BSS ereditata (ed estesa) da IEEE 802.11ac ed IEEE 802.11ah,

che consente di distinguere tra frame intra/inter BSS, basandosi solamente sul preambolo in modo da evitare di analizzare i payload possibilmente inutilizzabili a seguito di collisioni.

- Network Allocator Vector (NAV) modificato rispetto alla versione IEEE 802.11 legacy.
- Virtualizzazione più efficiente.
- Target Wake Time redisegnato.
- Microsleep per consentire ad una STA di andare in stato "dormiente" per un certo tempo noto.
- Risparmio Energetico opportuno.

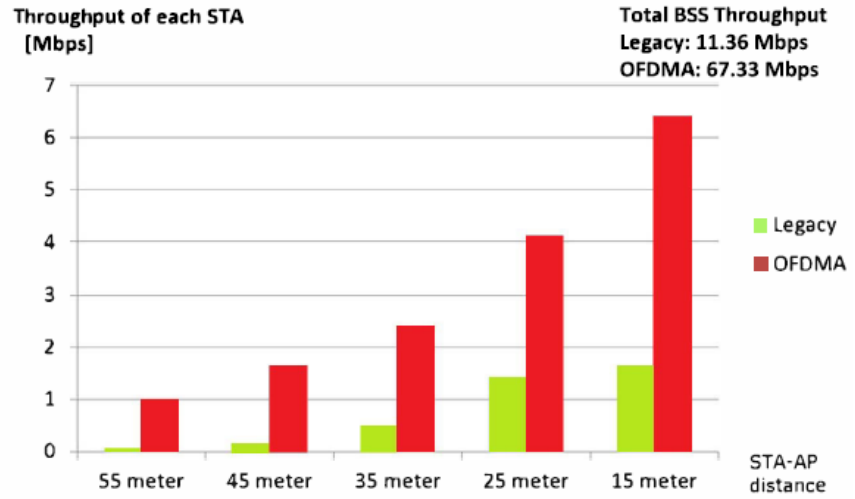

Figura 27. Incremento delle prestazioni nel caso di trasmissioni su reti parzialmente sovrapposte, mediante OFDMA rispetto alla versione 802.11ac. Fonte: [6]

Inoltre, il riutilizzo spaziale in una distribuzione densa è stato migliorato modificando la soglia di sensibilità (in termini di spazio) e la potenza di trasmissione (in termini di emissioni dell'antenna) tuttavia ancora oggi questo argomento è uno dei più dibattuti all'interno del team di sviluppo.

Un'altra tecnica riutilizzata in IEEE 802.11ax è quella delle prenotazioni periodiche di canale durante le quali possono trasmettere solo le STA che hanno richiesto il canale. Questo tipo di accesso può essere utilizzato sia per proteggere certi collegamenti che per stabilire divisioni di tempo tra BSS diverse (coordinando gli accessi).

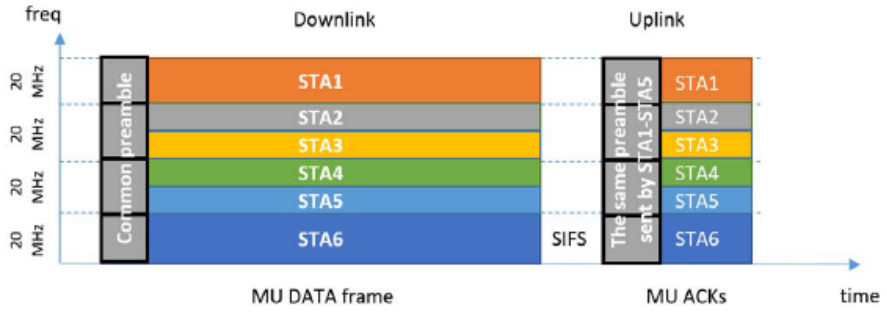

Figura 28. Trasmissioni DL MU e UL MU mediante OFDMA. Le due trasmissioni sono separate da un SIFS. Durante le trasmissioni, una STA viene mappata su un certo canale di frequenza (divisione di frequenza). Come si nota il preambolo sia nel caso di DL che UL è comune per tutte le STA coinvolte. Fonte: [6]

Il livello PHY 802.11ax eredita diversi aspetti dal suo predecessore 802.11ac. Analogamente a quest'ultimo, è basato sul OFDM e supporta operazioni in 20Mhz, 40Mhz, 80Mhz, 80Mhz + 80Mhz e 160Mhz. Per aumentare il numero di sotto-portanti adiacenti (questo favorisce l'OFDMA che riesce a trasportare frame a più utenti) è stata quadruplicata la durata dei simboli utilizzati per il payload PHY (12,8 μs) cosi da evitare effetti di jittering inter-user specialmente in scenari di reti outdoor. Inoltre, simboli più lunghi consentono di ridurre l'ovehead temporale sui Guard Intervals (GI). Basandosi sulle condizioni del canale, un dispositivo IEEE 802.11ax può selezionare un GI in modo consentire la riduzione del sovraccarico fino al 6% (a differenza di IEEE 802.11ac che consente una diminuzione del sovraccarico solamente fino al 25%).

L'emendamento IEEE 802.11ax introduce opzionalmente anche una nuova tecnica di modulazione chiamata 1024-QAM che può essere sfruttata in scenari indoor con condizioni di canale molto buone (cioè un SINR alto). Combinando insieme diversi codici di correzione dell'errore (controllo di parità convoluzionale o a bassa densità) che hanno rate di 1/2, 2/3, 3/4 e 5/6, con diverse modulazioni come BPSK, 16-QAM, 64-QAM, 256-QAM e 1024-QAM, si ha a disposizione un'ampia scelta di velocità di trasmissione fino ad un massimo di 9.6 Gbps. Un'altra modulazione che è stata introdotta è quella Dual Carrier Modulation (DCM) in cui si utilizzano due portanti separate in frequenza per trasportare lo stesso segnale informativo. I pro di questa modulazione sono l'aumento della robustezza e del guadagno SINR, mentre i contro sono legati al fatto che invio due volte lo stesso segnale informativo diminuendo cosi il throughput effettivo.

Sono stati introdotti 4 tipi di frame PHY (indicati come PPDU, unità dati protocollo PHY, a seguito della modifica): per la trasmissione Single-User (SU), per la gamma estesa di trasmissione SU e per le trasmissioni DL MU ed UL MU.

La PPDU a gamma estesa è stata progettata per una consegna affidabile e può solo essere trasmesso in un canale da 20Mhz, con uno dei tre MCS a più basso rate di codifica e senza MIMO.

Il frame PHY, come mostrato in figura 29, ha una struttura a telaio, nel senso che il frame è unico ma contiene i vari campi specializzati per ciascun tipo di trasmissione. Per tutti i tipi di frame il preambolo è duplicato in ogni sotto-canale da 20Mhz all'interno della banda di trasmissione, come mostra la figura 30, e si compone di due parti: la parte legacy e quella HE. Il primo è incluso per retro-compatibilità con le versioni precedenti, mentre HE è stato introdotto per segnali IEEE 802.11ax ed è decodificabile solamente da dispositivi che implementano IEEE 802.11ax.

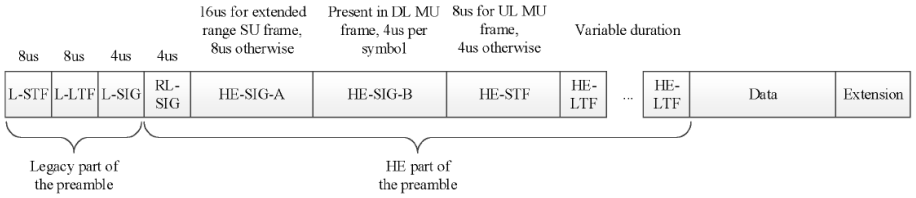

Figura 29. Struttura a telaio di un frame PHY 802.11ax e nuovo preambolo PHY. Esso è composto da una parte legacy e da una parte specifica per la gestione dei pacchetti IEEE 802.11ax chiamata HE. Quest'ultima è suddivisa a sua volta in base ai tipi di PPDU IEEE 802.11ax e si differenzia in HE-SIG-A, HE-SIG-B, HE-STF. Fonte: [6]

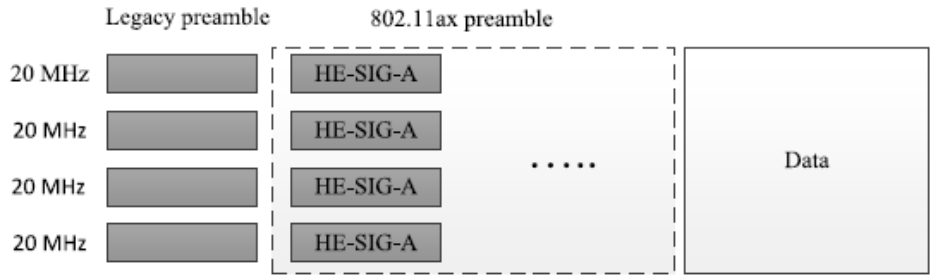

Figura 30. Invio delle parti legacy ed HE-SIG-A del preambolo specificando che sono duplicate su tutti i canali da 20Mhz in modo che tutte le STA riescono a leggerlo. Fonte: [6]

La parte legacy contiene un campo di sincronizzazione tra trasmettitore e ricevitore ed un campo che descrive i parametri del resto del frame PHY e consente il calcolo della durata complessiva del frame (L-SIG).

La parte HE del preambolo inizia con una ripetizione del campo L-SIG, seguito dal campo HE-SIG-A obbligatorio, dal campo HE-SIG-B opzionale e dai campi di configurazione (HE-STF e HE-LTF) necessari per la tecnica MIMO. Considerando i campi HE-SIG-A e HE-SIG-B in dettaglio, il primo fornisce informazioni su MCS, larghezza di banda, il numero di flussi spaziali (NSTS) e alcuni altri parametri necessari per la decodifica del resto del frame. Lo sviluppo del protocollo sta portando a spostare alcuni segnali MAC sul livello PHY e precisamente al preambolo. Siccome il rate di trasmissione è molto basso per quella parte di frame questo comporta notevoli overhead temporali. Inoltre questo implica un adattamento del segnale MAC ad un segnale rigido e statico. Tuttavia l'inclusione delle informazioni relative al MAC nel preambolo è vantaggiosa poiché il preambolo viene trasmesso con un più robusto MCS e può essere decodificato prima che il payload PHY venga ricevuto completamente.

Inoltre HE-SIG-A contiene anche informazioni sull'area BSS, sulla trasmission opportunity (TXOP) e sui parametri di riutilizzo spaziali (SRP) che vengono utilizzati per segnalare la potenza di trasmissione ed il livello di interferenza del segnale.

Poiché le reti IEEE 802.11ax sono progettate per adattarsi ad ambienti sia indoor che outdoor, le trasmissioni sono inclini all'effetto Doppler, causato principalmente dai riflessi di oggetti in rapido movimento. Per migliorare la resistenza all'elevata mobilità, l'emendamento propone di inserire periodicamente nel pacchetto payload midambles PHY, ovvero copie del campo HE-LTF. Grazie a midambles, il canale può essere stimato non solo durante il preambolo del pacchetto, ma anche durante la lettura di tutto il pacchetto. Questo meccanismo viene utilizzato nelle connessioni ad alta velocità di trasmissione, ovvero quando le condizioni del canale variano rapidamente. In caso di canali da 40Mhz, il campo HE-SIG-A è duplicato su ciascun sottocanale a 20Mhz. Nella variante a trasmissione estesa SU, viene ripetuto il contenuto di HE-SIG-A dopo un'ulteriore procedura bit interleaving (pattern di separazione dei vari campi).

In caso di trasmissione UL SU e DL SU, nonché in quelle UL MU, tutte le informazioni necessarie possono essere inserite nel campo HE-SIG-A che consiste in due simboli OFDM. Mentre nel caso di trasmissioni DL MU le informazioni per i vari utenti possono differire quindi devono essere specificate per ciascuno di essi separatamente. In questi casi un ulteriore campo HE-SIG-B di lunghezza variabile è incluso nel preambolo PHY. In particolare, il campo contiene due blocchi: uno con informazioni comuni ed uno con informazioni dedicate. Il blocco comune descrive l'allocazione delle risorse OFDMA, mentre il blocco per utente è costituito da diversi sottocampi che definiscono per ogni unità di risorse il suo MCS, il numero di flussi spaziali ed altri parametri relativi all'utente specifico.

Come accennato in precedenza, i campi HE-STF e HE-LTF sono usati per il MIMO. Lo scopo principale del primo campo è quello di migliorare la stima ed

il controllo del guadagno in trasmissione di un segnale MIMO, mentre il secondo campo viene usato per stimare il canale MIMO tra l'insieme di output della costellazione e gli input ricevuti dal canale.

Analogamente alla PPDU legacy, il campo dati contiene il sottocampo SIGNAL necessario allo scrambler per inizializzare la codifica/decodifica ed il campo MAC contenenti i dati di livello datalink codificati.

Il campo dati è trasmesso con simboli OFDM 4 volte più lunghi e questo comporta un tempo di elaborazione maggiore al lato ricevitore, mentre il tempo disponibile per quest'ultimo per fare tali calcoli prima di spedire l'ACK è limitato dal SIFS. Questo può portare problemi per i dispositivi WiFi a basso costo, che non fanno in tempo a generare il suddetto ACK, quindi il pacchetto precedentemente non confermato viene perso e si verifica una collisione. Una semplice soluzione consiste nell'aumentare il tempo SIFS ma non è stata implementata per questioni di retrocompatibilità e perché avrebbe ridotto l'efficienza di utilizzo del canale. Un'ulteriore soluzione viene offerta grazie alla possibilità di estendere la lunghezza del frame aggiungendo un'estensione al termine del campo MAC. Per ridurre al minimo le spese aggiuntive di questo campo, la sua durata è flessibile e dipende dallo specifico frame e dalle dimensioni del payload. In particolare, quando una STA dichiara le sue capacità, indica l'estensione massima del frame con un dato MCS e numero di flussi spaziali. Si noti che questo valore può essere ridotto se la dimensione del payload codificato non è divisibile per la dimensione del simbolo OFDM ed in tal caso l'ultimo simbolo conterrebbe padding. Infatti, il ricevitore avrebbe bisogno di meno tempo per decodificare i bit ottenuti da un simbolo OFDM così piccolo. L'ultimo simbolo OFDM viene diviso in in 4 segmenti di eguali dimensioni. Pertanto, l'estensione può essere ridotta ad intervalli discreti di 4 μs.

Nel corso degli ultimi due decenni, il processo di standardizzazione IEEE 802.11 si è concentrato sull'introduzione di nuove funzionalità, ma non ha mai specificato la modalità di realizzazione. Tuttavia, le prestazioni di una rete dipendono anche dal modo in cui le funzionalità introdotte vengono utilizzate. Dopo aver esteso il set di possibili velocità di trasmissione dei dati, lo standard aggiunge anche nuovi gradi di libertà (come DCM e IG più brevi) che influenzano la velocità di trasmissione e l'affidabilità della comunicazione.

Un numero elevato di opzioni complica la selezione del miglior set di parametri da utilizzare in trasmissione. Per implementare un meccanismo di selezione opportuno ci si affida ad algoritmi come Minstrel che provano vari MCS ed avendo ottenuto le statistiche da essi, selezionano i migliori per la trasmissione. Ovviamente, più aumenta la complessità del protocollo, maggiore è il numero di parametri da impostare in una rete durante le connessioni ed in proporzione maggiori sono il tempo di raccolta delle statistiche e quello di decisione per applicare un determinato set di parametri in un dato momento. In particolare, questo fattore è accentuato dal fatto che in IEEE 802.11ax le caratteristiche del canale cambiano per ogni sottobanda di 20Mhz dato che ciascuna di esse potrebbe avere un modello di interferenza

diverso. Inoltre, nelle trasmissioni UL MU, l'AP raccoglie i report sull'intensità del segnale da tutti gli utenti per allocare correttamente le risorse (RU, MCS etc.) con conseguente aumento dell'overhead temporale.

Nonostante la scelta dei parametri migliori non fa parte dello standard IEEE 802.11ax, dato che molti problemi sono sorti per gli sviluppatori e per i fornitori di tali dispositivi, esso si preoccuperà di chiarire questi aspetti approfondendo gli studi ed i dettagli legati a queste aree del protocollo. Un altro problema è che il preambolo PHY IEEE 802.11ax è più lungo di quello precedente, quindi dovrebbe essere usato solo in trasmissioni a lungo raggio che beneficiano maggiormente delle funzionalità IEEE 802.11ax, perché se fosse utilizzato per tutti i tipi di connessione l'overhead PHY sarebbe degradante. Inoltre, poiché i frame IEEE 802.11ax non possono essere decodificati da dispositivi legacy, il Virtual Carrier Sense (VCS) non funziona correttamente e può ridurre le prestazioni in scenari che presentano STA nascoste.

Dunque, molte funzionalità introdotte in passato andrebbero riviste e rianalizzate per consentire la compatibilità e la correttezza d'uso nei nuovi scenari emergenti. Lo standard IEEE 802.11ax sviluppa un nuovo protocollo di accesso al mezzo che permette sia l'accesso deterministico che quello casuale. Questo protocollo si basa su OFDMA. In questa tecnica di accesso multiplo le risorse del canale sono allocate nel tempo e nella frequenza, ma al fine di semplificare la gestione delle risorse, il funzionamento del dispositivo e per mantenere la retrocompatibilità, la trasmissione OFDMA è frame-based. Ciascun frame può trasportare informazioni da/per più STA. Vengono assegnati vari toni a STA diverse ma la durata di tutte le trasmissioni per le varie RU è la stessa. Una RU può contenere 26, 52, 106, 242, 484, 996 o 1992 toni mentre le bande da 20Mhz, 40Mhz, 80Mhz e 160Mhz corrispondono ad RU rispettivamente di 242 toni, 484 toni, 996 toni e 2 RU da 996 toni. A causa di vari problemi con i codici binari convoluzionali, sono vietate assegnazioni multiple di RU a diverse STA.

Sebbene MU-MIMO e OFDMA possano essere usati insieme, entrambi UL e DL MU-MIMO devono essere eseguiti solo in RU da 106 toni. Grazie a MU-MIMO, è possibile assegnare un RU fino ad otto utenti. È anche possibile allocare fino a quattro flussi spaziali per utente, se il numero totale di flussi spaziali non supera otto. Grazie a MU-MIMO è possibile assegnare fino ad otto utenti una stessa RU. È anche possibile allocare fino a quattro flussi spaziali per utente, se il numero totale di flussi spaziali non supera otto.

Considerando nel dettaglio l'organizzazione delle trasmissioni mediante OFDMA, in DL OFDMA il campo HE-SIG-B del preambolo comune contiene una mappa di allocazione delle RU seguita da campi di contenuto per utente che indicano le RU assegnate a ciascuna STA ed i parametri di trasmissione che devono essere utilizzati dalla STA (NSTS, MCS, codifica, ecc.). Si noti che una RU può rappresentare un SU o un MU-MIMO. In quest'ultimo caso viene indicata anche la configurazione spaziale delle antenne della STA ricevente. Organizzare la trasmissione UL MU è

molto più impegnativo.

Le trasmissioni WiFi UL MU invece devono avere una sincronizzazione nel tempo molto precisa. Dal momento che è difficile mantenere una sincronizzazione temporale rigorosa a causa della deriva del clock, un AP coordina la trasmissione di UL MU come segue: l'AP trasmette un trigger frame in cui vengono specificati i parametri comuni della prossima trasmissione UL MU (durata, GI che deve essere la stessa per tutti gli STA che partecipano alla trasmissione UL MU), alloca RU per gli STA e definisce i parametri di trasmissione per ciascuno STA (MCS, codifica, ecc.). Per ottenere la sincronizzazione, la trasmissione MU viene eseguita immediatamente, cioè un SIFS dopo il frame Trigger. Poiché potrebbe occupare più di un SIFS per preparare una trasmissione UL, l'AP può allungare tramite padding il frame trigger.

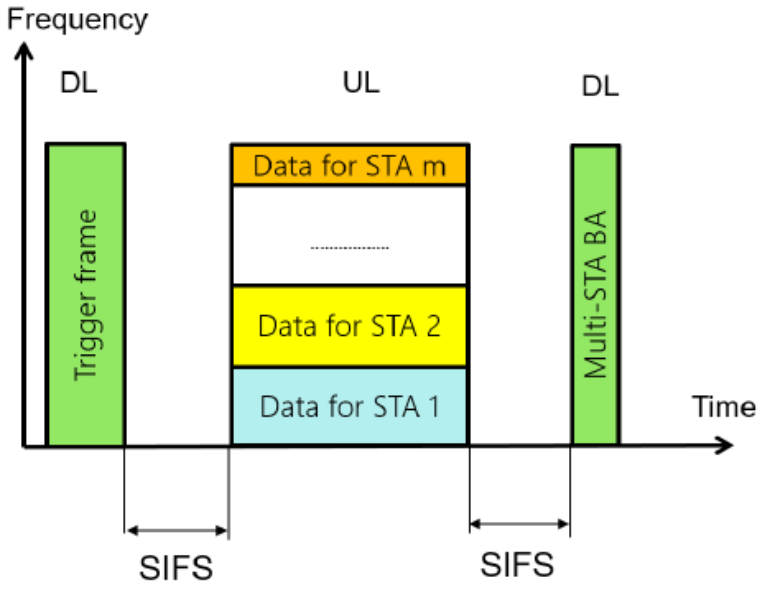

Figura 31. Sequenza delle operazioni e dei frame per eseguire un iterazione UL MU. Fonte: [6]

Per le trasmissioni UL MU OFDMA, l'AP deve ricevere segnali provenienti da diverse STA all'incirca della stessa potenza. Per questo, IEEE 802.11ax definisce un meccanismo di pre-correzione della potenza di trasmissione, in base al quale l'AP indica nel trigger frame la potenza di trasmissione corrente e la potenza del segnale target che l'AP dovrà ricevere dalle STA coinvolte nella trasmissione UL MU OFDMA. Quindi, conoscendo la potenza di trasmissione degli AP e la potenza del segnale di trigger frame ricevuto, la singola STA può stimare la perdita di potenza dovuta al percorso verso l'AP e può calcolare un potenza di trasmissione più appropriata da usare per la successiva trasmissione UL. Si noti che poiché l'AP

(e non un STA) seleziona l'MCS per le trasmissioni UL, ogni STA include anche informazioni sul suo headroom di potenza UL, cioè la differenza tra la massima potenza di trasmissione e la potenza di trasmissione per il MCS assegnato. Per essere efficiente, l'AP deve assegnare le RU solo a STA che hanno dati da trasmettere. Per questo motivo le STA riferiscono all'AP la quantità di dati bufferizzati che hanno. Tale rapporto può essere richiesto dall'AP oppure inviato dalle STA per propria iniziativa.

Un altro punto critico sta nel fatto che l'AP non conosce l'occupazione del canale rispetto a ciascuna STA. Per ogni STA, l'AP specifica nel trigger frame se dovrà eseguire il rilevamento del NAV prima di una trasmissione OFDMA o meno. Se è necessario il rilevamento del vettore, la STA dovrà eseguire il rilevamento del vettore virtuale e della portante fisica almeno nei canali da 20Mhz che contengono sottoportanti assegnati ad essa. Se il rilevamento della portante indica che il mezzo è occupato, ovvero viene rilevata un'alta energia, la STA annulla la trasmissione UL. La trasmissione UL è vietata anche se alcune sottoportanti sono inattive. Tuttavia in alcuni casi la STA può trascurare il rilevamento del vettore virtuale se il valore di durata è stato impostato da un frame originato da un STA vicina o da un STA che sta per trasmettere un ACK/BlockAck la cui durata è limitata da un valore concordato. In ogni caso, la STA annulla sempre la trasmissione UL se la sua durata supera la durata della trasmissione UL MU indicata nel trigger frame precedente.

Lo standard IEEE 802.11ax consente inoltre di eseguire una trasmissione UL MU subito dopo una trasmissione DL MU, che può essere utile, ad esempio, per l'invio simultaneo di frame di ACK. Per quello, il frame trigger delle trasmissioni DL MU deve contenere anche le allocazioni delle RU del UL MU successiva.

Seguendo le idee descritte, IEEE 802.11ax implementa anche il collegamento in cascata trasmissioni MU, il che significa che all'interno di un TXOP, le trasmissioni DL MU e UL MU possono alternarsi. Nota che all'interno delle trasmissioni MU in cascata l'AP può scambiarsi frame (UL/DL) con diversi set di STA.

Le trasmissioni MU in WiFi devono essere allineate nel dominio del tempo. Pertanto, se una STA ha frame di dimensione ridotta da trasmettere, il frame verrà modificato usando il padding o cercando di aggregarlo con un altro frame piccolo in maniera da rispettare il tempo di frame deciso dalla trasmissione MU. Per migliorare ulteriormente l'efficienza, le STA possono aggregare frame di diversi categorie di accesso (AC). Un approccio simile è utilizzato in IEEE 802.11ac DL MU MIMO.

Poiché l'aggregazione di diversi frammenti è complicata, è stato trovato un compromesso, avendo definito diversi livelli opzionali di frammentazione HE:

- Il primo livello consente solo l'invio un frammento senza alcuna aggregazione.
- Il secondo livello consente ad una STA di aggregare non più di un frame per MSDU in una AMPDU.

- Il terzo livello consente l'aggregazione di due o più frammenti per MSDU in una AMPDU.

L'OFDMA consente di mitigare l'effetto delle interferenze su frequenza selettive assegnando le migliori sottoportanti per gli STA. Inoltre riduce il sovraccarico causato da backoff, spazi di interframe, preamboli e intestazioni PHY, che portano informazioni comuni per tutte le STA in caso di trasmissione DL.

Tuttavia, il sovraccarico maggiore si ha ancora per i frame di controllo che essendo brevi hanno un grosso overhead temporale. Pertanto, oltre al trigger frame di base per i frame di dati e di gestione, IEEE 802.11ax ha sviluppato trigger frame speciali che hanno la funzione di avviare in parallelo l'handshake RTS/CTS, richiedere BlockAck da un gruppo di STA e raccogliere rapporti beamforming o rapporti sullo stato del buffer (BSR).

Per proteggere una trasmissione DL MU da nodi nascosti è stato introdotto l'handshake MU-RTS/CTS, il cui funzionamento è mostrato in figura 32. Grazie al UL MU, i frame CTS possono essere inviati contemporaneamente. La principale peculiarità del MU-RTS/CTS è che un frame CTS viene trasmesso sull'intero canale da 20Mhz, 40Mhz, 80Mhz o da 160Mhz e viene duplicato su ciascun sottocanale secondario a 20Mhz utilizzando il formato di frame CTS legacy. Il canale che deve essere utilizzato da una particolare STA per trasmettere il CTS è determinato dal contenuto del MU-RTS che specifica tutte le sottoportanti da utilizzare nel caso di trasmissione verso quella specifica STA che aspetta la risposta CTS.

L'uso congiunto di aggregazione e frammentazione è esplicitamente vietato. Il protocollo consente a diversi ricevitori di trasmettere i CTS contemporaneamente. Tuttavia, questi frame CTS sono assolutamente uguali dal punto di vista PHY, quindi non collidono. Un tale approccio ha un'importante limitazione. Avendo ricevuto più CTS uguali sullo stesso canale, l'AP non ha informazioni su quali riceventi hanno inviato il CTS. Tale limitazione costringe l'AP a non usare le trasmissioni parallele sulle sottoportanti dello stesso canale da 20Mhz.

L'emendamento 802.11ax propone un modo aggiuntivo per riconoscere le trasmissioni UL MU inviando nuovi Multi-Frame STA Block ACK (BA). Analogamente all'esistente Frame BA Multi-TID utilizzato per riconoscere un set di frame di varie AC, viene utilizzato un frame BA Multi-STA che agisce come ACK/BA diretto a diverse STA. Per accorciare la trasmissione dei frame BA Multi-STA, essi vengono inviati con il solo preambolo IEEE 802.11 legacy.

Un altro nuovo frame definito in IEEE 802.11ax è il frame MU di richiesta degli ACK (BAR). Viene utilizzato per richiedere gli ACK da più STA invece di inviare singoli frame BAR.

Un'ulteriore variante del trigger frame viene utilizzata per la raccolta BSRS. In ogni BSR, ogni STA informa l'AP in merito al quantità di traffico bufferizzato in una coda in base ad una AC richiesta (AC_BE, AC_BK, AC_VI o AC_VO) o di un sottoinsieme di AC.

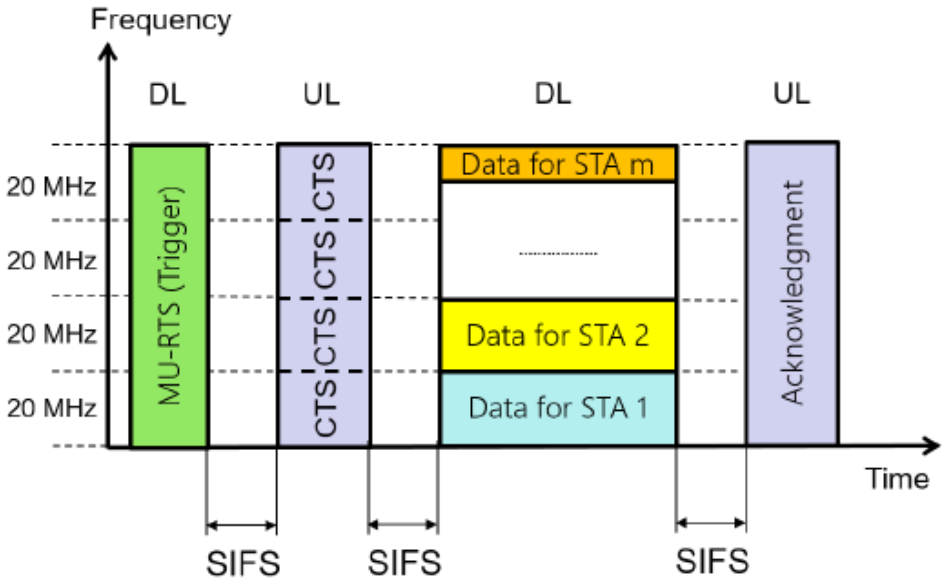

Figura 32. Sequenza delle operazioni e dei frame per eseguire un'iterazione MU RTS/CTS multi-STA. Fonte: [6]

Infine, vengono anche definiti dei frame speciali utilizzati per raccogliere informazioni di beamforming o richiedere informazioni sullo stato del canale. Oltre all'accesso programmato UL MU sopra descritto, è stato progettato un meccanismo opzionale che consente di eseguire trasmissioni UL OFDMA ad accesso casuale. Tale funzionalità è particolarmente importante quando l'AP non sa quali STA associate dispongono di dati da trasmettere o quando una STA non associata desidera connettersi ad esso. DCF / EDCA non sono efficiente per trasmissioni brevi, in particolare a causa del grande overhead dovuto alle intestazioni ed agli spazi di interframe PHY.

L'accesso casuale ideato è simile al meccanismo di slotted-Aloha multicanale. Ogni trigger frame può allocare determinate RU per l'accesso casuale. In particolare, alcune RU contigue vengono allocate in diversi gruppi per la trasmissione casuale, e questo mapping viene definito nei trigger frame sotto forma di record. Queste RU possono essere utilizzate da STA associate oppure da STA non associate che desiderano connettersi. Le RU di un gruppo hanno le stesse dimensioni e hanno gli stessi parametri di trasmissione. L'AP inoltre può indicare che nella prossima serie di trasmissioni MU a cascata, nessun gruppo di RU è allocato per le trasmissioni ad accesso casuale, fino alla fine del frame TXOP.

Per decidere se trasmettere e su quale RU farlo, le STA utilizzano la cosiddetta procedura di back-off OFDMA (OBO). Ogni STA sceglie un valore casuale tra 0

e OCW, dove OCW è la finestra di contesa OFDMA. Se il valore OBO scelto è inferiore al numero di RU assegnate per l'accesso casuale (Random Resource Unit-RRU) da un frame trigger, la STA seleziona casualmente una RRU e trasmette un frame su di essa. Altrimenti, la STA riduce OBO del numero di RRU e attende il successivo trigger frame contenente RRU. Se il tentativo di trasmissione fallisce, la STA raddoppia il suo OCW fino a quando non raggiunge OCWMAX e seleziona un valore OBO dal intervallo 0 - OCW iniziale. Se il tentativo di trasmissione ha esito positivo, la STA reimposta OCW sul valore minimo OCWMIN. Tutti e due OCWMAX e OCWMIN sono specificati dall'AP nei beacon frame e nei frame di risposta (probe response frame).

Poiché l'accesso casuale è meno efficiente dell'accesso pianificato, vale la pena usarlo solo per trasmissioni di pacchetti brevi e per BSR. In quest'ultimo caso, un STA con dati pronti da trasmettere può generare un BSR e inviarlo con accesso casuale per chiedere all'AP le risorse di canale. È chiaro che un tale schema risulta essere più efficiente del puro accesso casuale UL OFDMA. Tuttavia l'argomento è tutt'oggi poco definito, quindi sarà soggetto a cambiamenti futuri.

Nelle reti IEEE 802.11ax, OFDMA opera sopra CSMA/CA. Significa che per trasmettere un frame trigger, l'AP deve contendere per il canale con altre STA. Ipotizzando una rete con un AP e diverse STA con traffico UL, poiché il numero di STA connesse è molto alto, l'AP vince raramente la contesa se utilizza gli stessi parametri e meccanismi di accesso al canale. Tuttavia, quando l'AP vince la contesa, invia un frame trigger per allocare le risorse delle STA associate. Tuttavia OFDMA è molto più efficiente di EDCA. Infatti, per raggiungere maggiore produttività, le STA dovrebbero usare il meno possibile il meccanismo EDCA e preferire le trasmissioni OFDMA. L'AP deve quasi sempre vincere la contesa per garantire un accesso distribuito con poche collisioni. Fortunatamente, l'AP può cambiare i parametri EDCA per tutte le STA associate trasmettendole mediante beacon frame. Così, impostando valori elevati per CWmin e CWmax, l'AP può evitare le trasmissioni EDCA che le varie STA tentano di effettuare.

Quando dispositivi IEEE 802.11ax ed IEEE 802.11 legacy si trovano nella stessa rete, visto che i parametri EDCA non possono essere impostati singolarmente, impostando valori elevati di CWmin e CWmax per entrambi i tipi di dispositivi, e vengono penalizzate maggiormente le STA legacy. Questo può portare al degrado delle prestazioni dei dispositivi legacy.

Un altro problema è legato a un comportamento scorretto dell'AP che alloca meno RU per un client di un fornitore concorrente. Per evitare tali problemi, è stato introdotto un secondo set di parametri EDCA che viene utilizzato solo dai dispositivi 802.11ax a cui sono state concesse RU durante un intervallo di tempo precedente. Cosi facendo l'ECDA per i dispositivi legacy può dissociarsi dai parametri OCWMIN e OCWMAX ed utilizzare il canale con efficienza, contendendo il canale con l'AP.

Inoltre è stato migliorato il meccanismo di handshake RTS/CTS che aiuta a mitigare le collisioni da nodi nascosti e riduce la durata della collisione. Storicamente l'uso del meccanismo RTS/CTS era determinato dalla lunghezza dei dati trasmessi. Se la lunghezza del frame superava la soglia RTS, allora la trasmissione dei dati era preceduta da uno scambio di messaggi RTS/CTS. Tuttavia è stato proposto un meccanismo alternativo che ha due differenze principali. Innanzitutto la grandezza, da paragonare tra RTS e messaggio dati, è la durata della trasmissione anziché la dimensione del frame (RTS/CTS time-based handshake). È più naturale concentrarsi sulla durata della trasmissione piuttosto che sulla lunghezza del pacchetto, perché usando un MCS alto, anche frame grandi possono occupare il canale per poco tempo, ed in questi casi l'handshake RTS/CTS con un basso MCS sarebbe controproducente. In secondo luogo, il valore della soglia RTS/CTS in termini di durata, è sotto il controllo dell'AP che avendo una visione migliore della situazione della rete può impostare il valore di soglia migliore per le STA associate. In tal modo l'AP può abbassare la soglia in caso di interferenza da nodi nascosti o aumentarla per ridurre le spese generali di trasmissione ed ottimizzare l'utilizzo delle risorse di rete.

Avendo introdotto OFDMA, gli sviluppatori hanno reso il WiFi simile ad LTE. La forte somiglianza è dovuta alla modalità di allocazione delle risorse, che in WiFi è molto più difficile che in LTE per vari motivi. In primo luogo, le reti LTE tradizionali operano in bande di licenza. Ciò significa che un operatore può controllare le interferenze tra due celle vicine e regolare i canali di comunicazione per raggiungere prestazioni ottimali. Al contrario, le reti WiFi funzionano in bande esenti da licenza in cui nessuno può garantire il livello di interferenza in futuro. Questo complica il calcolo della stima di qualità del canale che viene eseguita quindi da algoritmi sofisticati finalizzati a ridurre le interferenze. In secondo luogo, nelle reti LTE il canale è diviso in blocchi di risorse di uguale dimensione. Per il canale DL, la stazione di accesso può selezionare un sottoinsieme arbitrario di blocchi per trasmettere alcuni dati per un utente. Per l'uplink, il sottoinsieme di blocchi scelti deve essere contiguo.

Nel WiFi invece le restrizioni sulle possibili RU da utilizzare sono più stringenti, il che complica lo sviluppo di scheduler ottimali, ovvero logiche che assegnano le RU a ciascun STA al fine di massimizzare una o più funzioni di utilità. In terzo luogo, per le trasmissioni UL, il WiFi consente l'aumento della densità spettrale di potenza se la STA trasmette su RU strette (composte da pochi toni). In particolare, la STA può trasmettere con la stessa potenza su qualunque RU. Si noti che poiché gli STA si trovano in luoghi diversi, non violano i vincoli di emissione di energia anzi, maggiore è la densità spettrale di potenza, maggiore è l'MCS utilizzabile. Ciascun frame trigger deve allocare RU soltanto per le STA che hanno dati nel canale UL. Tuttavia la soluzione è tutt'altro che banale ed infatti presenta notevoli ostacoli ed è ancora in via di sviluppo.

Il primo problema è che lo standard vieta di usare MCS alti per trasmettere

su RU da 26 toni. Quindi, suddividendo il canale in RU troppo strette, potremmo ottenere un rendimento inferiore. Il secondo è l'impossibilità di suddividere alcuni canali in un determinato numero di RU. Ad esempio, nel caso di tre STA con traffico UL, l'AP può dividere un canale da 40Mhz in due RU (242 toni + 242 toni) o in quattro RU (242 toni + 2x 106 toni + 26 toni), ma non in tre RU. Ciò significa che viene sprecata un'UR a 26 toni. Fortunatamente, RU cosi piccole possono essere utilizzate per trasmissioni ad accesso casuale nei rapporti sullo stato dei buffer. Alcuni studi dimostrano che l'allocazione ottimale delle RU dipende dalle posizioni dei dispositivi. I risultati mostrano che il throughput medio dipende maggiormente da come il canale è diviso in RU. Così, anche nel caso di una sola funzione di utilità (ad es. geometrica, media dei throughput che fornisce proporzionalmente equa allocazione delle risorse), lo schema di allocazione delle RU è molto sofisticato.

Una parte delle RU viene utilizzata dall'AP. Ovviamente il numero di RU assegnate all'AP determinano la latenza e la capacità della rete e devono essere selezionate basandosi su alcune stime di traffico. Nel caso di invio di pacchetti su trasmissioni UL, la STA può utilizzare il meccanismo EDCA per trasmettere sia questi pacchetti che i BSR. Tuttavia tali trasmissioni sono meno efficienti rispetto a quelle OFDMA, inoltre, a causa dei parametri EDCA trasmessi dall'AP (che come dicevano sono valori alti di CWMIN e CWMAX), il tempo di accesso al canale con EDCA è molto maggiore rispetto ad OFDMA. Un altro problema riguarda il fatto che una rete WiFi è costituita da dispositivi costruiti da vari produttori. Nel WiFi legacy, tutte le STA della rete dovrebbero usare gli stessi parametri di accesso al canale dell'AP corrispondente. Pertanto, tutti i dispositivi hanno la stessa opportunità di trasmettere (TXOP). Visto che, in una rete IEEE 802.11ax le risorse del canale sono assegnate dall'AP, esso potrebbe adottare dei meccanismi di allocazione favorevoli ai dispositivi prodotti dallo stesso venditore dell'AP. I metodi di rilevamento di tali AP (che si comportano male) dovrebbe essere oggetto di ulteriori studi ed analisi.

Un altro problema aperto è come selezionare una durata appropriata di un frame MU (viene impostata dall'AP). Ciò può influire sull'efficienza di utilizzo del canale, nonché sulla correttezza ed il QoS. Infatti, un AP deve trovare un compromesso nell'utilizzo di frame lunghi (efficienti in caso di grosse trasmissioni di dati) e brevi (efficienti per l'accesso casuale e per BSR).

Poiché lo scenario ad alta densità di rete è quello principale per IEEE 802.11ax, ci sono molti dibattiti su come migliorare le prestazioni in caso di reti fitte. Da una parte si vogliono ridurre le interferenze tra le reti, ma, d'altra parte, si vuole consentire il riutilizzo spaziale, cioè trasmissioni simultanee in reti sovrapposte per aumentare la portata. Un'attività considerevole è legata al rilevamento del vettore (NAV), soglie di sensibilità dinamiche e trasmissioni con controllo di potenza dinamico. Per fare ciò sono state introdotte nuove funzionalità.

Per determinare quale BSS abbia generato un certo frame senza decodificarlo interamente, IEEE 802.11ax utilizza l'ID non univoco del BSS, chiamato colore

BSS, che viene trasmesso nel preambolo del frame. Inizialmente, il campo di colore BSS era di 3 bit ed era apparso in IEEE 802.11ah per ridurre il consumo energetico, perché il ricevitore può smettere di decodificare un frame proveniente da un BSS "alieno" (BSS non di sua appartenenza). Poiché il colore BSS è selezionato casualmente dall'AP, i colori di due BSS vicini possono coincidere o scontrarsi. Per ridurre la probabilità di collisione dei colori BSS, in IEEE 802.11ax è stato deciso di aumentare la lunghezza del campo a 6 bit. Se si verifica la collisione, le STA associate possono notificarla all'AP associato che può avviare una procedura per cambiare il colore della BSS in conflitto. Per fare ciò, avvisa il futuro colore BSS e il momento in cui il colore verrà modificato tramite un elemento informativo speciale del beacon frame. Quindi tutte le STA, anche quelli in stato di sleep, si accorgono del cambiamento e registrano le nuove informazioni legate al loro BSS. Dall'identificazione del campo colore BSS vengono determinate le regole di accesso al canale e di risparmio energetico. Sebbene siano stati standardizzati metodi di accesso al canale centralizzato, come PCF o HCCA, che consentono all'AP di eseguire il polling delle STA, essi non sono utilizzati nei dispositivi out-of-the-shelf a causa della loro complessità implementativa e di alcune anomalie nel funzionamento in condizioni di distribuzioni dense. Per disabilitare il colore BSS basta porre a 0 il campo BSS color.

Altri cambiamenti sono stati apportati a livello NAV. L'accesso al canale WiFi segue il protocollo CSMA/CA, la STA esegue il rilevamento della portante prima di trasmettere. Il canale è rilevato occupato nei seguenti casi:

- Se Durante il rilevamento del NAV, una STA rileva il preambolo di un frame e considera il canale occupato per la durata del frame segnalata nel campo durata del preambolo.
- Durante il rilevamento del NAV, una STA rileva un segnale sconosciuto a più di 20 dBm al di sopra del livello minimo di sensibilità.
- Il Virtual Carrier Sense (VCS) segnala un canale occupato.

Il rilevamento del gestore virtuale in WiFi, chiamato NAV, è organizzato come segue. Nell'intestazione MAC, una STA indica il valore del NAV in base a quanto tempo il seguente scambio di frame occuperà il canale. Dopo aver decodificato correttamente il frame, le altre STA impostano il proprio NAV e considerano il canale occupato per il tempo indicato. Se una STA riceve un frame con un valore del NAV maggiore del suo, aumenta il suo NAV, tuttavia se invece il valore indicato dal NAV è più piccolo, il suo NAV non viene aggiornato bensì rimane immutato. Quando una STA riceve un frame CF-End, annulla il suo NAV.

Nel WiFi legacy le STA non tengono conto del valore del NAV di un frame. Questo può portare a comportamenti di rete scorretti. Supponiamo che una STA trasmette un frame ed imposta di conseguenza il valore del NAV su tutte le STA

che si trovano nella stessa BSS. Successivamente, un'altra STA di quella stessa BSS riceve un frame CF-End proveniente da un BSS sovrapposto (OBSS). Secondo le regole esistenti, la STA ripristinerà il NAV non considerando più il canale occupato, potrebbe iniziare la propria trasmissione provocando così una collisione. Queste situazioni critiche sono più significative in ambienti con alta densità di distribuzione. Tuttavia questo ragionamento non è più vero per reti IEEE 802.11ax poiché è stato impedito il ripristino del NAV da CF-End proveniente da un OBSS. Le STA IEEE 802.11ax supportano due NAV, uno per il proprio BSS e l'altro per tutti gli OBSS e modificheranno i NAV separatamente.

Inoltre IEEE 802.11ax cerca di migliorare i tempi di occupazione del canale introducendo la possibilità di instaurare una comunicazione ad hoc tra STA. Tuttavia, tali operazioni in una rete IEEE 802.11ax possono aumentare le interferenze e causare un significativo peggioramento delle prestazioni. Per risolvere questo problema, l'emendamento 802.11ax definisce il meccanismo Quiet Time Period (QTP). La STA può richiedere all'AP un QTP, cioè una serie di intervalli di tempo periodici di uguale durata utilizzati per operazioni di collegamento diretto o ad hoc. Il QTP è descritto dall'offset del primo intervallo allocato, la durata ed il periodo degli intervalli ed il totale numero di intervalli richiesti. Se l'AP soddisfa la richiesta, diffonde informazioni sul QTP riservato e proibisce alle altre STA di accedere al canale durante gli intervalli riservati. Questo meccanismo è stato proposto piuttosto di recente, quindi la sua descrizione è poco definita e presenta problemi che dovrebbero essere affrontati in futuro. In particolare, per diffondere informazioni sul QTP, IEEE 802.11ax definisce la seguente modalità di funzionamento: all'inizio di un intervallo di tempo riservato, l'AP trasmette informazioni sulla sua durata e sul tipo di operazione che è consentita durante l'intervallo. Tale comportamento presenta diversi inconvenienti. Innanzitutto il messaggio viene trasmesso solo una volta, quindi può essere perso. Secondo, il tipo di operazione non identifica l'insieme di STA che possono accedere al canale durante l'intervallo. Infine, non esiste alcun meccanismo esplicito per mettere a tacere le STA legacy che ignorano i frame IEEE 802.11ax.

Lo standard si concentra anche nel migliorare il riutilizzo spaziale in ambienti densi dove è importantissimo usare bene lo spazio. Una possibile soluzione potrebbe essere l'ottimizzazione dei meccanismi di rilevamento del NAV, ad esempio utilizzando il controllo dinamico della sensibilità (DSC). L'idea di DSC si basa sulla regolazione dinamica della soglia di rilevamento del segnale chiamata DSC. In base alla soglia, la STA considera il mezzo occupato o meno. Ovviamente, per impedire che una trasmissione all'interno un BSS venga bloccata da una trasmissione OBSS, la soglia DSC dovrebbe essere aumentata. Tuttavia, per consentire la comunicazione tra tutti i dispositivi all'interno di un BSS, la soglia DSC deve essere abbastanza piccola da non perdere informazioni. Per stimare l'attenuazione (fattore fondamentale per il riconoscimento del segnale, poiché altera la sua potenza) gli

autori propongono quanto segue. Ogni STA riceve il valore dell'indicatore di potenza del segnale (AvgRSSI) attraverso i beacon frame in arrivo dall'AP e impostano la soglia DSC su AvgRSSI - MRG . Tuttavia, l'attenuazione può aumentare così tanto che la potenza del segnale radio dell'AP resta inferiore alla soglia e la STA inizierà ad ignorare i beacon frame. Per prevenire tale comportamento, si propone di fare decrementare AvgRSSI di RSSIDEC (valore costante) se si perdono più frame di fila cosi da diminuire automaticamente il valore della soglia DSC. I test finora eseguiti variano MRG ed il valore di RSSIDEC per valutare l'efficienza dello schema proposto in termini di rendimento aggregato, equità, numero di nodi nascosti e Packet Error Rate (PER) ed i risultati mostrano un netto miglioramento. Il guadagno di rendimento ed equità è raggiunto a costo di a maggior numero di nodi nascosti e, di conseguenza, un maggiore PER. Da un lato, è naturale pensare che DSC possa diminuire l'equità, perché vicino agli AP le STA impostano un valore alto della soglia DSC ed hanno maggiore possibilità di trasmettere un pacchetto. Tuttavia il DSC riduce il numero di nodi che ottengono un vantaggio in termini di equità.

Avendo ridotto il numero di STA esposte, il DSC aumenta il numero di STA nascoste. Per risolvere questo problema sono stati proposti vari metodi. Uno di questi prevede di usare il meccanismo RTS/CTS accoppiato al DSC. Questo approccio si è dimostrato efficace. Un'altra soluzione è di utilizzare il DSC combinato a l'inter-BSS Time Division Multiple Access (TDMA) che effettua trasmissioni di OBSS ortogonali nel dominio del tempo per attenuare l'interferenza inter-BSS. Sebbene DSC e TDMA sono approcci opposti, la loro combinazione dà le prestazioni migliori in assoluto, aumentando contemporaneamente il rendimento medio e quello peggiore. Sfortunatamente l'implementazione di inter-BSS TDMA è troppo complicata e richiede una sincronizzazione stretta tra OBSS. Per bilanciare tra riutilizzo spaziale e prevenzione delle collisioni, è stato deciso di vincolare le variazioni della soglia di sensibilità per i frame OBSS (denominato come soglia di rilevamento del preambolo OBSS, OBSS_PD) e la potenza di trasmissione (TX) secondo una regola semplice: maggiore è l'OBSS_PD, minore è il TX. Per impostazione predefinita, un STA trasmette segnali di potenza TX_PWR e considera il mezzo inattivo se la potenza del segnale è inferiore a OBSS_PD = -82 dB. Nel caso in cui una STA riceve un segnale da un OBSS STA di x dB più forte rispetto al valore OBSS_PD -82 dB significa che l'attenuazione tra la STA e l'OBSS STA è più debole del necessario, quindi prima della prossima trasmissione deve aumentare il suo OBSS_PD di x dB e di conseguenza diminuire la potenza di trasmissione dello stesso valore per non subire l'interferenza dovuta alle trasmissioni di quella particolare OBSS-STA. Le STA possono modificare dinamicamente i propri parametri OBSS_PD e TXPWR. Durante il backoff ad esempio, un STA imposta il suo OBSS_PD a un certo valore. Ogni volta che rileva l'inizio di un pacchetto sospende il suo backoff e se rileva un pacchetto appartenente ad una OBSS, può riprendere il backoff anche prima della

fine del frame. Quando la STA ottiene accesso al canale, può iniziare la trasmissione con una potenza minore di OBSS_PD. Tale livello di potenza viene utilizzato fino alla fine della TXOP.

Inoltre l'AP può specificare i colori degli OBSS per i quali viene applicata la regola descritta sopra. Per ottenere il massimo beneficio dal riutilizzo spaziale, la regola dovrebbe essere applicata per tali BSS, il cui livello di segnale è molto più basso di quello delle STA associate all'AP. Ovviamente è necessario un algoritmo di decisione che classifica le BSS circostanti in base a delle scelte. Un'altra opzione che consente il riutilizzo spaziale è relativa al frame trigger. In particolare l'AP può consentire la sovrapposizione tra una trasmissione non appartenente al BSS ed una trasmissione UL nel proprio BSS. Se il segnale ricevuto da tale trasmissione non appartenente al BSS non supera un livello significativo di interferenza (livello del segnale ricevuto sopra la soglia OBSS_PD). Tale livello dipende dall'interferenza attuale del canale vicino all'AP e dal MCS utilizzato. Per consentire un trasmissione sovrapposta, nel frame trigger l'AP specifica la potenza di riutilizzo spaziale S come somma della sua potenza di trasmissione più il livello accettabile di interferenza meno un certo margine. Avendo ricevuto il frame trigger ad una certa potenza R, un OBSS STA può avviare una trasmissione con potenza S - R dopo la fine del frame trigger, se tale trasmissione non supera la fine della trasmissione programmata UL. Naturalmente, per accedere al canale, l'OBSS STA deve usare il backoff, ignorando l'imminente trasmissione UL.

In IEEE 802.11ax le STA possono selezionare in modo adattivo la larghezza di banda per la trasmissione di un frame particolare. Lo standard definisce una gerarchia di canali distinguendoli in primari e secondari, ed il channel bonding si può effettuare solamente tra canali primari e canali secondari. L'assemblamento dei canali primari e secondari nonché la banda risultante, vengono mostrati in figura 33. Dopo aver ottenuto l'accesso al canale primario a 20Mhz secondo le regole EDCA, una STA può espandere la larghezza di banda concatenando passo dopo passo i rispettivi canali secondari se sono liberi. In altre parole, se il canale secondario da 20Mhz è libero, la STA può trasmettere con una larghezza di banda di 40Mhz. Se i canali secondari da 20Mhz e da 40Mhz sono liberi, è possibile utilizzare una larghezza di banda di 80Mhz e così via fino ad una banda massima di 160Mhz. Inoltre, anche se il canale secondario da 40Mhz è libero, se il canale secondario da 20Mhz è occupato, la STA può trasmettere solo sul canale primario da 20Mhz. Questa limitazione è particolarmente cruciale per reti dense, in cui il canale secondario da 20Mhz di un BSS può essere il canale primario a 20Mhz di un altro. Per migliorare l'efficienza del channel bonding in un ambiente denso, IEEE 802.11ax introduce una nuova funzionalità opzionale chiamata foratura del preambolo. Per una trasmissione MU OFDMA su un canale maggiore o uguale a 80Mhz, i canali occupati da 20Mhz possono essere rimossi. Significa che per quel frame il preambolo non viene trasmesso e le RU non sono allocate in questi particolari sottocanali. In una distribuzione densa, tale funzionalità consente di

canalizzare le risorse in modo molto più flessibile.

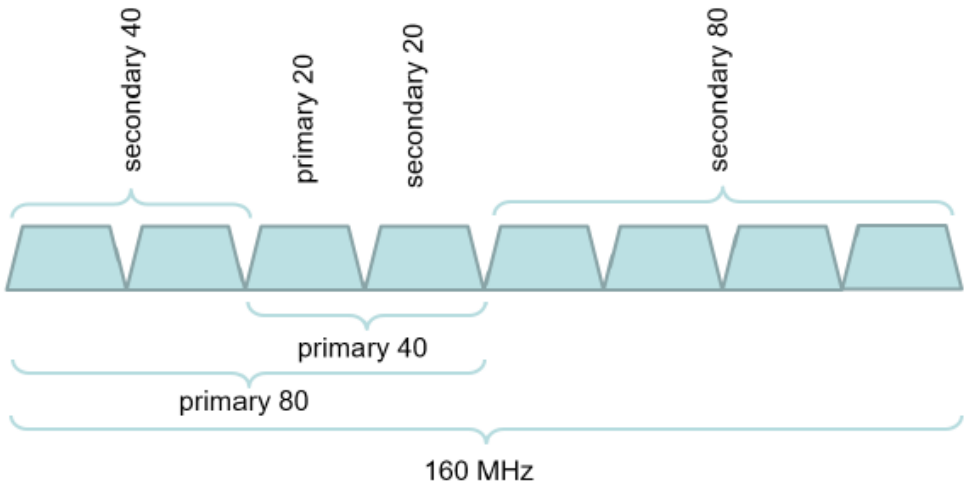

Figura 33. Suddivisione dei canali in primari e secondari, e aggregazioni possibili in modo da formare canali più ampi. Fonte: [6]

Una delle funzionalità più diffuse nei moderni AP è il supporto per AP virtuali multipli (VAP). Ciò significa che a un singolo dispositivo fisico può creare più BSS indipendenti, raggiungendo fino a 32 VAP in alcune apparecchiature. Questo può essere utile quando, ad esempio, si desidera separare rete WiFi client da una rete WiFi interna senza installazione di un AP aggiuntivo. Una delle carenze di VAP è che molte informazioni di servizio per tutti i VAP possono essere le stesso, ma sono trasmesse separatamente da ciascuno di essi. Per ridurre il sovraccarico, l'emendamento 802.11ax ha introdotto il supporto al BSSID multiplo, che consente l'invio degli stessi frame a tutte le BSS simultaneamente. Tutti i BSS nel BSSID multiplo utilizzano lo stesso colore BSS e i frame delle BSS appartenenti al BSSID sono considerati intra-BSS.

Nelle reti dense, il bilanciamento del carico è un problema importante, poiché ogni STA ha diversi AP candidati a cui associarsi. Sebbene il problema abbia suscitato notevole interesse tra i ricercatori, è fuori dal campo di applicazione dello standard.

Nelle reti IEEE 802.11, la gestione dell'alimentazione si basa sull'alternanza tra due stati: sveglio e dormiente. In stato sveglio, una STA può trasmettere e ricevere frame, mentre in stato di sospensione, la ricezione radio è spenta. Una STA attiva è sempre sveglia, mentre un PS STA si alterna tra questi due stati. Poiché l'AP non conosce lo stato corrente di una PS STA, memorizza su un buffer tutti i frame (tranne alcuni in tempo reale) destinati a quella STA. Per notificare ai PS STA i pacchetti bufferizzati, l'AP include una mappa di indicazione del traffico (TIM)

nei beacon frame. Un PS STA può dormire a lungo, tuttavia di tanto in tanto quando si sveglia per ricevere un frame con un elemento TIM oppure può svegliarsi prima, se ha un frame da trasmettere. In questo caso, prima di iniziare l'accesso al canale, la STA deve attendere la ricezione di un frame. Se il beacon frame indica che non ci sono pacchetti bufferizzati destinati alla STA, questa ritorna nello stato dormiente. Altrimenti, la STA invia un frame PS-Poll. Come risposta al messaggio PS-Poll, l'AP invia i frame bufferizzati. Sebbene il concetto descritto sia piuttosto semplice, è stato progettato per un carico piuttosto basso e su misura per utilizzare l'accesso casuale. Nei tipici scenari IEEE 802.11ax con reti fitte, con l'alto carico di traffico e un gran numero di smartphone a consumo limitato e laptop, i meccanismi di risparmio energetico legacy sono inefficienti. Innanzitutto, potrebbero bloccarsi, vale a dire che i PS STA potrebbero rimanere svegli a lungo, quando il traffico viene consegnato ad altre STA. In secondo luogo, l'AP non può fornire i frame senza essere sottoposto a polling. In terzo luogo, i sondaggi PS consentono solo la trasmissioni SU meno efficienti di quelle MU ed inoltre, l'overhead causato dal messaggio PS-Poll è relativamente grande. L'attuale standard contiene anche diversi metodi che consentono la programmazione in anticipo dei periodi di servizio, cioè quando le PS STA possono trasmettere il messaggio PS-Poll e recuperare i pacchetti bufferizzati dal AP. Questi metodi sono profondamente connessi con la funzionalità HCCA che tipicamente non è implementata. A parte questo, questi metodi non sono adatti per trasmissioni OFDMA. L'idea chiave dei miglioramenti introdotti da IEEE 802.11ax è che devono essere sveglie solo le STA trasmittenti/riceventi, mentre tutti le altre STA possono andare a riposo. Questo può essere fatto nel modo seguente: in IEEE 802.11ax le STA possono rimanere nella cosiddetta modalità microsleep, cioè possono spegnere la loro interfaccia radio durante alcune trasmissioni, quando non possono essere coinvolti nel processo di scambio di frame. In secondo luogo, IEEE 802.11ax adatta il Target Wakeup Time (TWT), un meccanismo leggero progettato in IEEE 802.11ah per programmare periodi di servizio, senza utilizzare la funzionalità HCCA.

Il meccanismo di microsleep è stato introdotto in IEEE 802.11ac. In quella versione, l'intestazione PHY contiene l'AID parziale che indica il trasmettitore e i/il ricevitori/e di un frame. Quindi le altre STA possono passare allo stato di sospensione per la durata del frame. Lo standard IEEE 802.11ax estende questa idea consentendo a una STA di spegnersi durante trasmissioni UL o TXOP di un'altra STA nella stessa BSS. Per questo il campo HE-SIG-A contiene informazioni come il colore BSS, la durata TXOP rimanente, la direzione di trasmissione (UL o DL), ecc. In particolare, se un frame con lo stesso colore con una struttura UL MU o DL MU, non è destinato alla STA, essa può essere sicura che nessun frame verrà trasmesso ad essa fino alla fine di TXOP, e può andare nello stato di spegnimento.

Al fine di ridurre al minimo la contesa tra STA e ridurre il consumo energetico, IEEE 802.11ax ha adattato il meccanismo TWT introdotto in IEEE 802.11ah. TWT consente ad una STA, chiamata richiedente TWT, di negoziare con un'altra

STA o AP chiamato rispondente TWT. Quando la STA richiedente TWT si sveglia per qualche tempo (chiamato TWT Service Period o TWT SP) e scambia i frame con la STA rispondente TWT. Grazie a questo meccanismo, la STA richiedente TWT può andare in idle sempre, eccetto durante gli intervalli TWT SP. In particolare, avendo stabilito il TWT SP con l'AP, la STA non deve necessariamente svegliarsi anche per i beacon, che possono ridurre significativamente il consumo di energia. Inoltre, la scelta di uno stesso TWT SP non proibisce ad altre STA di accedere al canale. Pertanto, TWT non fornisce un accesso al canale privo di contese e le STA trasmettono frame in TWT SP usando il canale con politiche di accesso legacy (EDCA, DCF). Per proteggere la trasmissione da collisioni è possibile utilizzare il VCS. Tuttavia IEEE 802.11ax ha reimplementato ed esteso il concetto di TWT. Nelle reti IEEE 802.11ax, gli TWT SP possono essere concordate singolarmente o trasmesse in broadcast. Gli TWT SP concordati individualmente sono negoziati tra una coppia di dispositivi. Durante i negoziati, si trasmettono l'un l'altro un elemento informativo speciale che contiene i parametri TWT e può essere interpretato come richiesta, suggerimento, domanda, alternanza, accettazione, dettatura o rifiuto. Sia l'AP che le STA possono modificare il TWT trasmettendo un frame TWT Teardown.

I parametri TWT più importanti sono l'inizio del primo TWT SP ed il Wake Interval, ovvero l'intervallo tra due TWT SP consecutivi. Questi due parametri determinano l'intera serie di TWT SP. Oltre ai parametri appena citati, le STA possono negoziarne altri :

- Minimum Wake Duration che indica il valore minimo di TWT SP, dopo il quale la STA richiedente il TWT può tornare allo stato di sospensione anche se non ha ricevuto un frame. Se necessario, un determinato SP può essere troncato anche al di sotto del suo valore, ad esempio trasmettendo un frame con il flag EOSP (periodo di fine servizio) impostato.
- Quali tipi di frame devono essere trasmessi all'interno di SP TWT.
- Se la trasmissione deve essere eseguita sul canale primario a 20Mhz.
- Se il periodi di servizio TWT SP deve essere protetto mediante NAV, ad esempio (MU-) RTS/CTS o CTS-to-self.
- Se lo STA rispondente TWT può essere nel periodo esterno a TWT SP.
- Se deve essere la STA richiedente TWT ad eseguire il polling del rispondente TWT all'inizio di ogni TWT SP che gli indica che è attivo, oppure deve essere la STA rispondente TWT ad inviare i frame al TWT.
- Se gli TWT SP sono abilitati dal trigger. Trigger-enabled TWT SP sono favorevoli al funzionamento UL MU e sono possibili solo se la STA rispondente

TWT è l'AP. All'interno di tali TWT SP, l'AP deve inviare almeno uno frame trigger che alloca le risorse per la richiesta TWT. I Trigger-enabled TWT SP sono molto utili per le STA a risparmio energetico. Innanzitutto, portano tanti vantaggi alle trasmissioni UL OFDMA. Inoltre, secondo lo standard, un STA che si sveglia non può avviare immediatamente la trasmissione senza attendere almeno un frame con il quale può impostare il proprio NAV o far scadere qualche timeout. Quindi i frame trigger consentono alla STA di ridurre i tempi di attesa.

La STA richiedente TWT non deve trasmettere nessun frame alla STA rispondente TWT all'interno dei Trigger-enabled TWT SP ed al di fuori di qualsiasi TWT SP negoziata. Questo per prevenire collisioni con trasmissioni nascoste in corso.

Gli TWT SP in broadcast sono simili a quelli concordati, salvo piccole discrepanze. In particolare, non sono negoziati ma sono programmate dall'AP distribuendo informazioni in beacon frame. Le STA che hanno ricevuto queste informazioni ma non hanno ancora stabilito i singoli SP TWT con l'AP, dovrebbero trasmettere solo all'interno del SP indicata dal beacon frame. Detto questo, può essere fatto un esempio di risparmio energetico con UL OFDMA Random Access. In questi casi le STA possono spegnersi sempre, tranne che per riceve certi frame (beacon compresi) e per partecipare ai TWT SP. In particolare, l'AP può programmare una serie di frame trigger (contenenti anche le informazioni sulle varie TWT SP). Per avvisare le STA del primo intervallo TWT SP (e quindi del primo frame trigger), l'AP utilizza Broadcast TWT. Per informare le STA dei seguenti frame trigger, l'AP alza un flag speciale in ogni frame tranne nell'ultimo. Questo flag indica che un altro frame trigger Random Access segue quello corrente (trasmissione UL della STA e DL Ack inviata dall'AP). Poiché un frame trigger contiene anche la durata della seguente trasmissione, le STA che non partecipano alla sessione, possono dormire fino al prossimo frame trigger. Il risveglio per i beacon frame può essere evitato nel modo seguente. Con la segnalazione TWT, la STA può negoziare con l'AP l'intervallo durante il quale la STA rimane dormiente senza ricevere beacon.

Il meccanismo mostrato in figura 34, chiamato Opportunistic Power Save (OPS), consente ad un AP di dividere un intervallo beacon in diversi sottointervalli - broadcast TWT SP - e di fornire, all'inizio di ogni sottointervallo, informazioni su quali STA saranno servite in questo sottointervallo. Sulla base di queste informazioni, le STA AP possono opportunisticamente andare a dormire fino al prossimo TWT SP broadcast trasmesso. Questo meccanismo si basa sull'uso congiunto di TWT ed un elemento TIM legacy. TIM viene utilizzato nella gestione dell'alimentazione legacy per indicare l'insieme di STA per i quali l'AP ha dati bufferizzati. In OPS, TIM viene trasmesso dall'AP in broadcast, insieme al TWT SP all'inizio del sottointervallo TWT SP. In questo caso, TIM indica l'insieme di STA che dovrebbe essere attive (l'AP le ha svegliate) durante l'attuale TWT SP. Se una STA non è indicata in TIM, può dormire durante la corrente TWT SP. L'idea di OPS è vicina

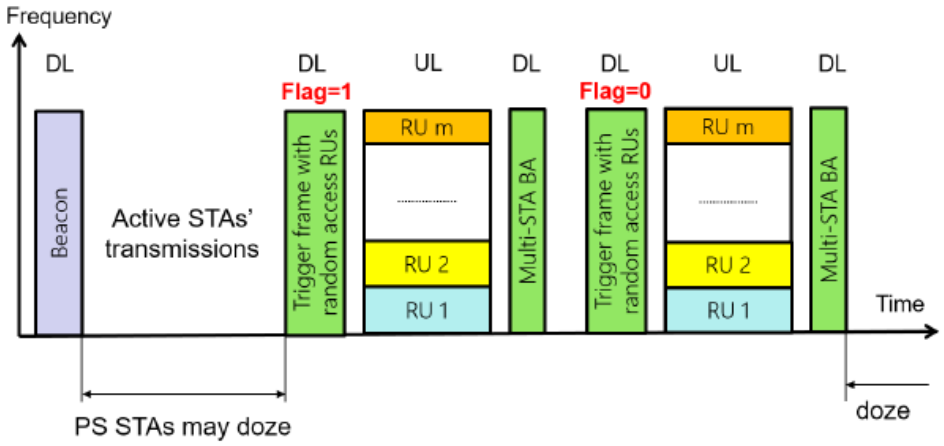

Figura 34. Sequenza di frame che vengono scambiati tra l'AP e le STA per implementare le funzionalità di salvaguardia di energia. Fonte: [6]

alla segmentazione TIM utilizzata in IEEE 802.11ah. Tuttavia, a differenza della segmentazione TIM, OPS riduce la granularità temporale.

I meccanismi di gestione dell'alimentazione permettono un efficiente utilizzo della batteria, aumentando cosi la durata di utilizzo. Allo stesso tempo il loro utilizzo porta ad una serie di domande. Dal momento che Microsleep e Opportunistic Power Save consentono alle STA di spegnere il loro sistema radio, dovrebbero essere usati deliberatamente anche nella ricezione di traffico sensibile al QoS. Quando la STA è attiva, l'AP può modificare istantaneamente il suo programma di decisione nel momento stesso in cui sulla coda arriva un pacchetto destinato a questa STA. Al contrario, sia Microsleep che OPS rendono impossibile un comportamento real-time. Per quanto riguarda il TWT, il problema più importante è come garantire accesso rapido al canale e senza contesa ad un STA durante il negoziato TWT SP in un ambiente denso. All'inizio della negoziazione TWT SP, il canale potrebbe essere occupato con trasmissioni provenienti da OBSS vicine. Pertanto, nonostante gli ovvi vantaggi di TWT, la sua reale efficienza è ancora oggetto di studio.

802.11be

I cambiamenti apportati dal protocollo IEEE 802.11be riguardano aspetti importanti come la coesistenza con altre tecnologie wireless (il numero di dispositivi IIoT wireless è in crescita) e l'aumento del throughput rispetto al suo predecessore IEEE 802.11ax, guadagnandosi il nome di Extremely High Throughput (EHT).

I requisiti dei servizi di dati wireless sono in continua evoluzione in molti scenari come case, imprese e hotspot. La tecnologie, che si adatta alle esigenze applicative,

deve crescere di conseguenza. Il traffico video sarà il tipo di traffico dominante negli anni a venire, quindi il throughput richiesto per questo tipo di traffico continuerà a crescere di decine di Gbps con l'emergere di tecnologie come il 4K o 8k. Contemporaneamente, le nuove applicazioni sono sempre più esigenti di alto rendimento e bassa latenza di trasmissione. Tra queste applicazioni, di tipo industriale, vi sono l'ufficio remoto ed il cloud computing che richiedono latenze inferiori a 5 ms. Anche l'affidabilità è una delle maggiori preoccupazioni nell'emergente industria digitale, dove garantire che il 99,9% dei pacchetti di dati sono consegnati correttamente è il requisito minimo richiesto. L'obiettivo principale è quello di raggiungere tali livelli di affidabilità non con il cablato ma con connessioni wireless che ottengono prestazioni accettabili in termini di soft real-time. Il nuovo protocollo IEEE 802.11be è in fase di sviluppo, nel maggio 2018 sono stati formati il gruppo Topic Interest Group (TIG) e di studio (SG) dal Working Group IEEE. Gli argomenti d'interesse sono molteplici come:

- La larghezza di banda 320Mhz ed un uso più efficiente degli spettri non contigui.
- Aggregazione ed operazioni multi-canale/multi-banda.
- L'utilizzo di 16 flussi spaziali ad I/O migliorando la tecnologia MIMO.
- Coordinamento del punto di accesso multiplo (in presenza di più AP nel raggio di copertura).
- Adattamento avanzato del collegamento e miglioramento del protocollo di ritrasmissione.

Analizziamo i punti appena citati, l'uso dello spettro dell'aria può essere realizzato in maniera più efficiente con l'introduzione della banda a 6Ghz che porta ad un aumento della velocità di trasmissione. La suddivisione della banda, nonché l'utilizzo delle sotto bande è mostrato in figura 35.

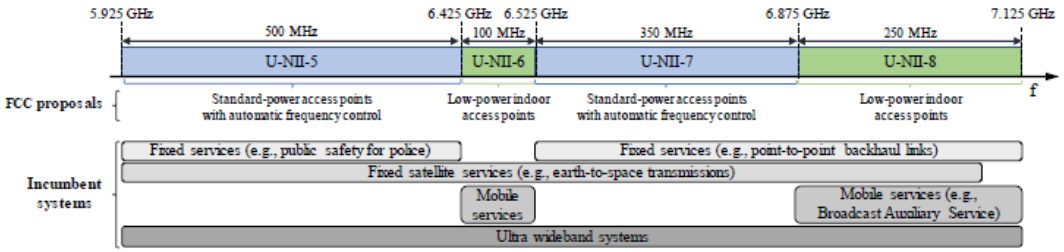

Figura 35. Banda e le sottobande discusse in IEEE 802.11be. Fonte: [7]

A questo proposito sono discussi vari approcci efficienti per operare fino a 1,2Ghz di larghezza di banda senza licenza tra 5.925Ghz e 7.125Ghz, raddoppiando così la banda rispetto alla versione a 5Ghz. L'adozione delle larghezze di canale 160Mhz e 320Mhz per gli AP nella banda 6Ghz obbligatoria e le caratteristiche opzionali, sembrano non essere modifiche difficilmente integrabili nel predecessore IEEE 802.11ax che aveva già l'obbligatorietà della banda di trasmissione 160Mhz per gli AP. Inoltre, una dimensione minima del canale di 40Mhz o 80Mhz nella banda 6Ghz sembrano più appropriati rispetto a quella da 20MHz utilizzata nella banda 2,4Ghz e 5Ghz e garantiscono un throughput più elevato. Mentre i vantaggi dell'utilizzo della banda 6Ghz per migliorare il throughput di picco e di sistema sono ovvi, l'utilizzo di una nuova banda aperta permette nuovi approcci di rete. Sono in corso discussioni sul fatto che un AP EHT dovrebbe sempre programmare trasmissioni UL a 6Ghz, riducendo così il tempo speso per la contesa del canale. Inoltre è probabile che le STA implementino lo standard IEEE 802.11ax anche nella banda 6Ghz. Per questo motivo, servono schemi di coordinamento dei canali tra i dispositivi IEEE 802.11ax ed EHT che consentono ad un AP EHT di selezionare il canale di comunicazione degli AP IEEE 802.11ax.

Con l'emergere di STA dual-radio e triband AP, in grado di operare contemporaneamente a 2.4, 5 e 6Ghz, uno dei principali obiettivi di EHT è quello di fare un uso più efficiente di queste bande a disposizione. Le tecniche discusse finora sono:

- Aggregazione di dati multibanda. L'aggregazione dello spettro a 6Ghz e 5Ghz per la trasmissione o la ricezione dei dati incrementa il thoughput. In effetti, l'aggregazione richiede che i dispositivi WiFi sincronizzino l'inizio dell'opportunità di trasmissione (TXOP) in diversi bande. Quindi questo approccio non è utilizzato in scenari con alta densità (dove la contesa del canale è problematica), bensì in reti sparse.

- Trasmissione e ricezione simultanee in diversi bande. Questa funzione, comunemente nota anche come multibanda full duplex, ha il potenziale di ridurre la latenza della comunicazione e migliorare la produttività abilitando trasmissioni UL/DL asincrone e simultanee in bande separate. Se questa funzione dovrà essere inclusa in EHT, il TG probabilmente imposterà dei IG tra i canali DL ed UL per impedire mutue interferenze.

- Trasmissioni e ricezioni simultanee nella stessa banda. Parallelamente a EHT SG, IEEE 802.11 WG ha anche approvato la formazione di una TIG nel gennaio 2018 per esaminare la fattibilità tecnica del full duplex nelle bande WiFi. Il TIG ha terminato la sua attività in Dicembre 2018, concludendo che può essere realizzato con piccole modifiche e può offrire vari vantaggi come un aumento del throughput, una latenza ridotta, il rilevamento delle collisioni e la mitigazione dei nodi nascosti all'interno di rete densa.

- Separazione dei dati e del piano di controllo. Dispositivi EHT con le capacità multibanda full duplex hanno un'opportunità di separazione dei dati senza precedenti. Ad esempio, il feedback sullo stato del buffer di una STA viene attualmente trasmesso utilizzando lo stesso canale dedicato alla trasmissione e ricezione dei dati, introducendo pertanto ritardi e spese aggiuntive che si traducono in una scarsa pianificazione delle decisioni e perdite di throughput. Questi problemi potrebbero essere mitigati dedicando una banda alla trasmissione/ricezione dei dati ed una banda complementare affidabile per trasmissioni/ricezioni frequenti di informazioni di controllo.

Nel corso degli anni la tecnologia si è evoluta aumentando il numero di antenne e migliorando le capacità di multiplexing spaziale degli AP WiFi per soddisfare le richieste di traffico generate dall'aumento del numero di dispositivi con connettività wireless. Attualmente, la maggior parte degli AP di fascia alta sul mercato, basati su IEEE 802.11ac, sono dotati di 4 antenne e possono multiplexare fino a quattro flussi spaziali in una data risorsa tempo/frequenza, mentre molte STA hanno più di un'antenna. Lo standard IEEE 802.11ax ha aggiornato queste funzionalità, consentendo agli AP multi-antenna di multiplexare spazialmente fino a otto dispositivi in UL e DL, tramite MU MIMO. Coerentemente con questa tendenza e grazie ai requisiti ETH, molti stakeholder WiFi necessitano di migliorare ulteriormente le capacità di multiplexing spaziale degli AP fino a trasmettere sedici flussi spaziali contemporaneamente. Questo aggiornamento potrebbe raddoppiare l'efficienza spettrale rispetto a IEEE 802.11ax, sfruttando appieno la velocità fornita da soluzioni Fiber To The Home (FTTH) e la ricca dispersione negli ambienti interni dove i sistemi WiFi funzionano in genere. Tali guadagni di multiplexing spaziale tuttavia potrebbero essere ostacolati dal sovraccarico del processo di suono del canale (CSI) fondamentale per rilevare lo stato del canale. Raddoppiare il numero di flussi spaziali riutilizzando la stessa procedura esplicita di acquisizione CSI attualmente specificata in IEEE 802.11ax potrebbe a problemi di scalabilità. Per questa ragione, EHT sta considerando di introdurre una procedura CSI implicita che si basa sulla reciprocità di canale (teorema dell'elettromagnetismo che lega l'area efficacie dell'antenna con le grandezze dell'onda trasmittente/ricevente come la lunghezza d'onda ed il guadagno dell'antenna stessa). Tuttavia questo obbligherebbe gli AP ad implementare una procedura di calibrazione per recuperare eventuali disallineamenti hardware che romperebbero la reciprocità di canale.

Una collaborazione tra AP EHT vicini consentirebbe un utilizzo più efficiente del tempo, della frequenza e delle risorse spaziali disponibili. Di seguito discutiamo tre delle principali alternative, mostrate in figura 36, considerate all'interno di EHT, seguendo un ordine crescente di complessità di coordinamento:

- OFDMA coordinato in cui gli AP collaborativi sincronizzano la trasmissione di dati ed usano risorse di tempo/frequenza ortogonali tra loro. Questa

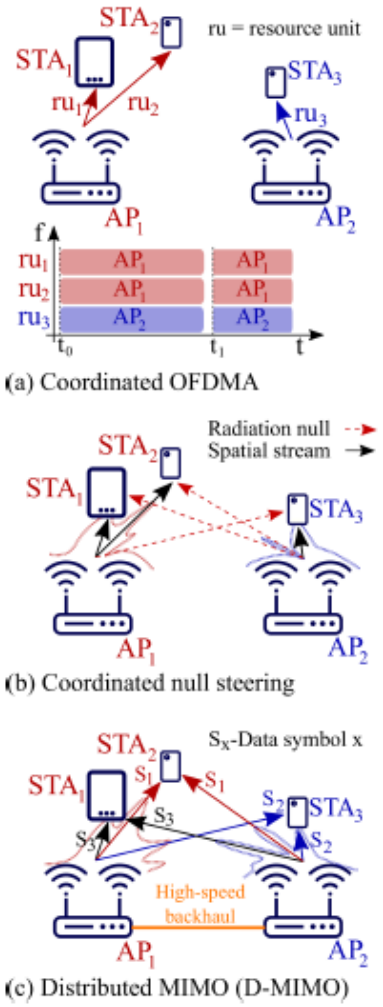

Figura 36. Alternative proposte in IEEE 802.11be per l'ottimizzazione nell'utilizzo delle grandezze fisiche in gioco, come il tempo, la frequenza e dello spazio. Fonte: [7]

assegnazione coordinata delle risorse diminuisce la probabilità di collisione rispetto al caso in cui gli AP implementano l'accesso al canale basato su procedure di contesa indipendenti. OFDMA coordinato è particolarmente utile per ridurre al minimo la latenza dei dati di pacchetti brevi da trasmettere, poiché consente una condivisione efficiente ed una piena occupazione della banda, a differenza della contesa indipendente che non riesce a sfruttare appieno le potenzialità delle risorse.

- Null Steereng coordinato. AP multi-antenna in genere utilizzano le proprie

capacità per il multiplexing spaziale delle STA con le stesse risorse tempo/frequenza e per fornire un guadagno utile di potenza del segnale attraverso il beamforming. In alternativa, gli AP possono anche sfruttare le loro antenne per rendere nulla la radiazione spaziale da e verso dispositivi non associati nelle vicinanze. Questo approccio è mirato ad aumentare ulteriormente il riutilizzo spaziale abilitando la trasmissione simultanea dei dati - usando il stesse risorse tempo/frequenza - dei dispositivi all'interno di stessa BSS. Rispetto ad OFDMA coordinato, la direzione nulla richiede generalmente un'ulteriore grado di cooperazione tra BSS sovrapposte per organizzare la pianificazione delle decisioni e facilitare l'acquisizione del CSI dei dispositivi non associati, essenziale per il posizionamento efficace delle radiazioni nulle.

- MIMO Distribuito (D-MIMO). La soluzione più elaborata in termini di complessità di coordinamento considerata da EHT è D-MIMO, dove gli AP eseguono una trasmissione e/o ricezione congiunta dei dati da più STA riutilizzando le stesse risorse tempo/frequenza. Rispetto ai sistemi con AP indipendenti, la stretta collaborazione tra AP attraverso D-MIMO può fornire una copertura estesa, grazie ai guadagni di beamforming aggiuntivi ed al miglioramento del multiplexing spaziale. D-MIMO è il meccanismo che rende possibile l'uso di 16 flussi spaziali. Raggiungere questi guadagni richiede una collaborazione inter-AP per elaborare congiuntamente i dati ed il CSI di tutte le STA coinvolte nei dati in trasmissione/ricezione, aumentando così la necessità di un'elevata capacità di collegamenti cablati a bassa latenza (ad esempio fibra) o wireless (ad esempio rete di backhauling a onde millimetriche). È importante sottolineare che l'implementazione di D-MIMO in EHT richiederebbe la progettazione di un nuovo meccanismo di accesso multiplo con prevenzione delle collisioni (CSMA/CA), conformi alle normative, per ottimizzare l'accesso al canale e garantire una giusta convivenza con AP indipendenti appartenenti alla stessa area di copertura. Questo aspetto, insieme alle caratteristiche di tempo e frequenza ed ai vincoli relativi alla sincronizzazione di fase imposti dal livello PHY ha portato EHT a proporre implementazioni D-MIMO, costituito da un AP master che deve essere visibile a tutti gli AP collaboranti e che supervisiona le operazioni del cluster. Mentre teoricamente comprometterebbe le prestazioni, la presenza di un AP master potrebbe semplificare notevolmente i requisiti di coordinamento, ad esempio abilitando un efficiente sincronizzazione nell'aria attraverso la trasmissione di un frame di trigger di controllo per i dispositivi collaboranti.

Gli attuali sistemi WiFi si affidano alla ritrasmissione della MPDU quando queste non sono decodificate correttamente al ricevitore oppure quando un ACK non viene ricevuto dal trasmettitore. In questa richiesta di ripetizione automatica (ARQ), il ricevitore scarta l'MPDU errata prima di ricevere la sua versione ritrasmessa, così facendo non c'è la possibilità di un controllo tra le due. L'iterazione

tra AP ed una data STA che partecipa alla trasmissione corrente, viene mostrata in figura 37. Con i miglioramenti in affidabilità e latenza, sempre più aziende sostengono che EHT dovrebbe evolversi verso capacità ibride di ARQ (HARQ). I dispositivi compatibili con HARQ non ignorano le precedenti MPDU errate, invece combinano i loro (MPDU corretta ed MPDUs errate) soft-bit per migliorare la probabilità di decodifica corretta. Il meccanismo HARQ - già implementato nei sistemi cellulari e precedentemente discusso durante la standardizzazione di WiFi 5 e 6 - può fornire al rapporto SINR guadagni di circa 4dB rispetto ad ARQ in condizioni di ideali di canale additivo con rumore gaussiano bianco (AWGN). Mentre i vantaggi di HARQ sono chiari in condizioni di AWGN, considerando che le trasmissioni WiFi spesso soffrono di interferenze improvvisa dovute alle collisione, le sue prestazioni non sono ancora state analizzate. Inoltre, dal punto di vista computazionale, l'operazione di combinazione di HARQ richiede capacità computazionali e requisiti di memoria extra per elaborare le ritrasmissioni e memorizzare quelle passate.

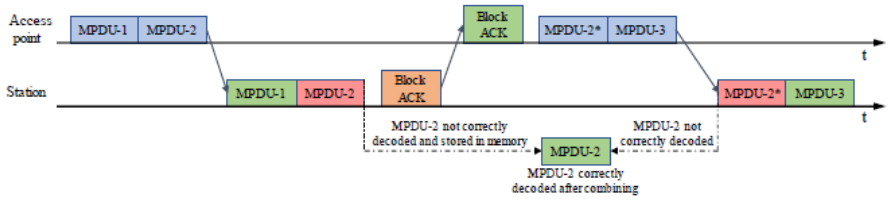

Figura 37. Sequenza di frame che vengono scambiati tra l'AP e le STA per implementare la funzionalità di Multi Block Acknowledgment. Fonte: [7]

In conclusione, IEEE 802.11be si sta sviluppando in maniera totalmente diversa rispetto ai suoi predecessori. Visto che la banda 6Ghz ospita già varie tecnologie come New Radio Unlicensed (NRU), l'idea generale è di trovare un modo per far coesistere tali tecnologie sulla stessa banda. Le soluzioni proposte sono molteplici come ascoltare il canale prima di trasmettere, utilizzare un preambolo unico per tutte le tecnologie in quella banda, oppure cercare politiche di coordinamento globale basate sulla geolocalizzazione. Tuttavia, lo sviluppo di questa tecnologia richiederà l'intervento congiunto di IEEE 802.11be, sperando che la loro esperienza in banda 5Ghz possa permettere loro di risolvere questo tipo di problemi.

Seamless Redundancy

Uno dei meccanismi per incrementare le prestazioni del WiFi si basa sull'uso della ridondanza a livello Data Link Layer (DLL). Questa tecnica usa schede di rete multiple per incrementare le prestazioni in termini di determinismo ed affidabilità. Anzitutto si basa sul concetto secondo il quale i disturbi non si manifestano tipicamente su tutti i canali contemporaneamente, infatti questo tipo di trasmissione soffre meno di interferenza selettiva (cioè di quell'interferenza che si verifica solo in alcune bande di frequenza). Una tecnica molto utilizzata di ridondanza è WiFi Redundancy (Wi-Red), che tuttavia consuma N volte la banda di un singolo canale. Ipotizzando una ridondanza doppia (ottenuta utilizzando due canali), il consumo di spettro è raddoppiato. L'adozione di meccanismi che operano in fase di esecuzione, volti a impedire la trasmissione di frame superflui, consente di ridurre il traffico generato. Nella soluzione presentata in [11] si è pensato ad una ridondanza doppia del canale, dove le ritrasmissioni superflue sono limitate completamente via software. Questo rende il meccanismo quasi indipendente dal hardware. Questa tecnica si è dimostrata utile in applicazioni di controllo distribuito in ambienti industriali che in genere richiedono scambi di dati affidabili tra processi e dispositivi, ed entro un determinato tempo (deadline) in modo da garantire un comportamento prevedibile. Le tecnologie di comunicazione wireless esistenti erano principalmente concepite per scenari di consumo e ufficio, e non riescono a soddisfare le specifiche di affidabilità e determinismo richieste in molte applicazioni industriali. Per questo motivo negli ultimi anni sono state proposte numerose soluzioni per affrontare in modo specifico i contesti industriali. Meccanismi deterministici di controllo dell'accesso al mezzo (MAC) mirano a ridurre le interferenze coordinando le trasmissioni tra i nodi all'interno della rete. Tra questi vi sono Time Slotted Channel Hopping (TSCH) per IEEE 802.15.4 e Hybrid Coordination Function (HCF), in particolare HCCA che consiste in una funzione di accesso al canale per IEEE 802.11. Altre soluzioni si basano su Time Division Multiple Access (TDMA) dove ogni stazione ha un time slot in cui trasmettere. Tuttavia, tali tipologie di soluzioni non sono efficaci contro i disturbi a larga banda causati ad esempio dal rumore elettromagnetico, inoltre i vantaggi vengono persi quando si tratta di trasmissioni non coordinate di nodi non appartenenti alla stessa rete. Per contrastare efficacemente questi fenomeni l'utilizzo di canali differenti mediante la ridondanza può essere d'aiuto. Anche nel

caso di overlay di protocollo non deterministici (che si basa su CSMA/CA), come ad esempio i ritardi di accesso introdotti da DCF o HCCA, la ridondanza di canale porta notevoli benefici. Una delle soluzioni che usa la ridondanza è Parallel Redundancy Protocol (PRP) su WiFi chiamata PRP over WiFi (PoW) dove la ridondanza end-to-end è stata posta in cima allo stack WiFi per implementare reti più sicure. Rispetto a PoW, la diversità dei canali in Wi-Red viene trattata direttamente dal livello link. Ciò consente ottimizzazioni specifiche, come l'uso di meccanismi di prevenzione della duplicazione, Duplication Avoidance (DA), che operano in fase di esecuzione e mirano a ridurre il consumo della larghezza di banda. L'approccio di DA presuppone che trasmissioni consecutive sullo stesso canale non siano dipendenti, il che non può essere verificato visto che l'accodamento avviene nei buffer di trasmissione. Questa ipotesi è ragionevolmente vera per un sistema industriale di controllo distribuito ben dimensionato, poiché i vincoli di temporizzazione possono essere difficilmente rispettati in presenza di ritardi di accodamento. Tuttavia, a causa dell'imprevedibilità dello spettro wireless, il buffering dei pacchetti non può essere del tutto prevenuto. Pertanto è necessario utilizzare altri approcci per analizzare il comportamento dei meccanismi DA nel caso in cui l'interferenza non sia trascurabile. Di seguito viene descritta un'implementazione di un prototipo, come mostrato in figura 38, che fa utilizzo di Wi-Red, basata su apparecchiature commerciali, che esegue il meccanismo DA direttamente nel software.

I nodi in Wi-Red sono definiti STA ridondanti (RSTA). Un RSTA è composta da diversi STA secondarie (due, nel caso di ridondanza doppia), le cui operazioni possono essere assunte, in teoria, completamente indipendenti. Ogni sub-STA include sia un blocco MAC che uno radio, e ricorda l'architettura di un STA convenzionale. Inoltre vi è un blocco chiamato entità ridondante di collegamento (LRE) che si occupa del coordinamento delle sub-STA locali e delle operazioni di gestione della ridondanza. Una BSS ridondante (RBSS) è solo un insieme di RSTA vicine che possono comunicare su due o più canali distinti. Le operazioni eseguite in Wi-Red sono abbastanza semplici. Ogni volta che viene fatta una richiesta di trasmissione di pacchetto nel RSTA di origine, una copia di quel pacchetto viene messa in coda separatamente per la trasmissione su ogni sotto-STA del livello LRE. Allo stesso modo, non appena una di tali copie è stata ricevuta correttamente su qualsiasi sub-STA del destinatario, il pacchetto viene inviato ai livelli superiori dal modulo LRE. Quest'ultimo, al momento della ricezione, deve anche scartare tutte le copie del pacchetto già ricevute e prendersi cura del riordino dei pacchetti quando necessario. L'operazione sopra descritta implica che:

- Un pacchetto è completamente perso solo quando tutte le sue copie non riescono a raggiungere correttamente il destinatario,
- La latenza di trasmissione per un recapito riuscito coincide con la sua copia più veloce.

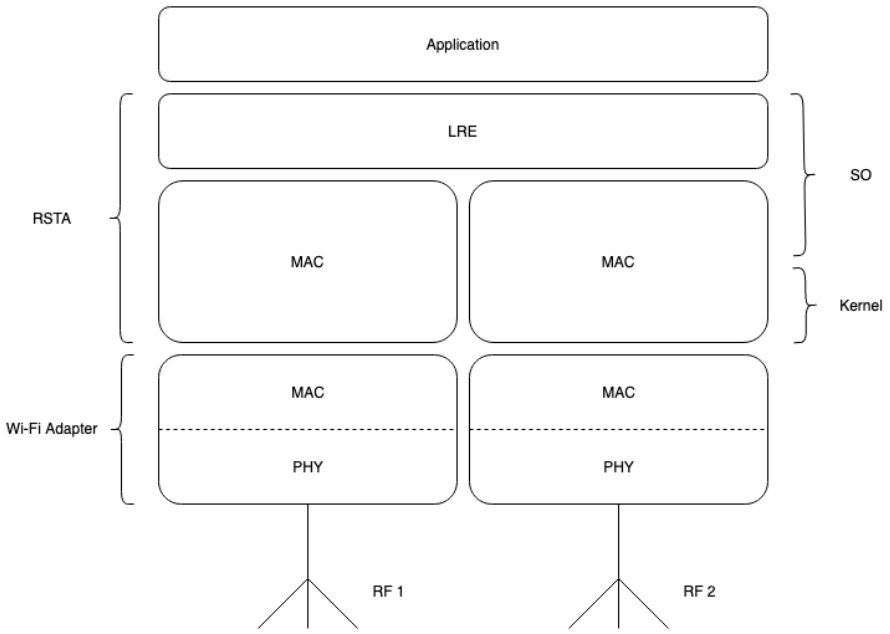

Figura 38. Architettura Seamless Redundancy. Come si vede dalla figura, l'architettura SoftMac scelta si compone essenzialmente di un livello RSTA ed un livello WA gestito dai driver hardware. Le duplicazioni di pacchetti vengono gestite anche da un livello superiore integrato nel modulo RSTA, chiamato LRE. Esso si occupa della cancellazione dei pacchetti duplicati e della segnalazione degli ACK ricevuti da una data sub-STA macchina, alle singole sub-STA che ancora non hanno completamente trasmesso il pacchetto confermato sulla macchina.

A sua volta, questo significa che i collegamenti ridondanti sono più veloci e più affidabili dei singoli canali fisici. Il principale miglioramento che Wi-Red introduce rispetto PoW è la capacità di sfruttare le sinergie a livello MAC tra schemi di ritrasmissione automatici e ridondanza (come intesa in PRP), al fine di migliorare la qualità della comunicazione.

L'affidabilità dello spettro wireless viene abitualmente trattata dal livello MAC utilizzando un frame di conferma (ACK) del pacchetto precedentemente spedito che si basano su tecniche Automatic Repeat reQuest (ARQ). A seguito della trasmissione di un frame di dati, la sub-STA di origine imposta un ACKtimeout definito dallo standard WiFi. Se un frame ACK (proveniente dal destinatario) viene ricevuto dalla sub-STA prima della scadenza del timeout, il tentativo di trasmissione ha esito positivo. Al contrario, dopo un adeguato backoff, la sub-STA di origine invia nuovamente il frame. Per prevenire la starvation (impossibilità non quantificabile in termini di tempo, da parte di un processo, di accedere alle risorse hardware e/o software del sistema), viene definito un limite massimo di tentativi di ritrasmissione. Se tale limite viene raggiunto, il processo di trasmissione viene terminato ed il pacchetto scartato. I tentativi di trasmissione devono essere effettuati in tempi

precisi e molto brevi. Durate tipiche di attesa tra i frame dati e ACK (durata denominata SIFS) sono di 10 μs o 16 μs, a seconda del banda di frequenza (2,4 Ghz o 5 GHz, rispettivamente). Quindi, questa parte del protocollo deve essere necessariamente eseguita in hardware (o nel firmware) tramite adattatori WiFi (WA). IEEE 802.11 prevede anche trasmissioni di frame non confermate, per le quali non viene eseguita nessuna ritrasmissione. Tuttavia, la loro affidabilità è molto inferiore alle trasmissioni confermate, e quindi sono raramente impiegate per scambi di dati con requisiti temporali. Ciò è ancora più vero per le applicazioni industriali, in cui le perdite dei pacchetti incidono molto sulla qualità del controllo e devono essere mantenute al minimo possibile. In pratica, vengono utilizzate solo trasmissioni non confermate per comunicazioni multicast, come quelle per la trasmissione di segnali di controllo (ad esempio il frame di beacon trasmesso dall'AP alle STA da lui coordinate).

Per prevenire trasmissioni inutili di pacchetti non necessari in contesti di ridondanza multipla bisogna trovare un modo efficace per migliorare le prestazioni dei collegamenti ridondanti sia in termini di consumo di spettro che di qualità del canale. Nel caso specifico di trasmissioni di dati confermate, le prestazioni di Wi-Red possono essere migliorate in modo significativo sfruttando DA, che rappresenta un approccio olistico per gestire congiuntamente le ritrasmissioni e la ridondanza. Di seguito, descriveremo il reactive DA (RDA), meccanismo di DA che evita l'invio dei pacchetti già confermati. L'idea alla base di RDA è di interrompere la trasmissione di tutte le copie di un determinato pacchetto sul RSTA di origine non appena viene ricevuto un ACK relativo a quel pacchetto su una qualsiasi sub-STA. Questo evento è indicato come ACK incrociato (XACK) ed è concettualmente trattato dal livello LRE. Le azioni svolte dopo un evento XACK hanno lo scopo di prevenire inutili tentativi di trasmissione sul canale del pacchetto correlato. In termini di qualità della comunicazione e del consumo della larghezza di banda, le prestazioni RDA non sono mai peggiori del puro PoW. Oltre ad RDA, esistono meccanismi proattivi di prevenzione della duplicazione (PDA) che consistono principalmente in soluzioni euristiche volte a migliorare ulteriormente le prestazioni. Una varietà di tecniche PDA esistono, che in genere richiedono l'impostazione di un compromesso tra consumo di spettro e qualità comunicazione.

Da un punto di vista pratico, il funzionamento e le prestazioni di RDA possono variare a seconda della sua implementazione. Tuttavia, poiché RDA è interamente gestito sul RSTA di origine, è garantita la completa compatibilità, indipendentemente dall'implementazione Wi-Red del destinatario. Per semplicità, nel seguito faremo solo riferimento alla ridondanza doppia, ma il ragionamento può essere facilmente esteso al caso di ridondanza multipla. All'occorrenza di un evento XACK, la copia più lenta del pacchetto può essere o in procinto di essere inviato dal MAC correlato o in attesa nella coda di trasmissione. Nel primo caso la trasmissione della copia è terminata prima da RDA rispetto a PoW (dove le trasmissioni su singoli canali avvengono esattamente allo stesso modo come nel WiFi convenzionale).

Invece, nel secondo caso la trasmissione di tale copia può essere impedita completamente, rimuovendola dalla coda. La forma di implementazione RDA più semplice è RDA a livello di coda (RDA/Q). Quando un XACK viene generato da una data sub-STA per un determinato pacchetto, la coda di trasmissione dell'altra sub-STA viene esaminata, cercando copie dello stesso pacchetto non ancora trasmesse dal MAC correlato. Se viene trovata, viene rimossa dalla coda, il che significa che non avverranno tentativi di trasmissione della copia (risparmio della banda). Tuttavia, la rimozione può avvenire solo per i pacchetti in coda e l'accodamento dei pacchetti si verifica raramente quando il traffico di rete è basso. Questo approccio può essere implementato sulla maggior parte degli adattatori commerciali, ad esempio apportando modifiche ai driver del dispositivo. La coda di trasmissione è in genere organizzata come un buffer ad anello (ring buffer) memorizzato nella memoria principale e può essere manipolato nel software. Oltre alla rimozione dei pacchetti dalle code di trasmissione, è anche possibile imporre la risoluzione anticipata dei pacchetti già ricevuti da altri adattatori, mentre tale pacchetto è in fase di invio da parte del livello MAC correlato. La trasmissione di pacchetti può essere interrotta in modo sicuro sul confine tra i tentativi di trasmissione (ovvero dopo che ACKtimeout è scaduto e prima dell'inizio del prossimo tentativo). Al contrario, interrompere una trasmissione in atto è inutile e problematico. È possibile ideare una versione avanzata di RDA, indicata come RDA a livello MAC (RDA/M), che combina entrambi i concetti descritti sopra, ottenendo così le massime prestazioni. Lo svantaggio principale di questo approccio è che pochi adattatori commerciali consentono di interrompere ordinatamente le trasmissioni in corso, perché questa capacità non è esplicitamente prevista dallo standard IEEE 802.11.

Nell'implementazione RDA/Q, LRE e le code dei messaggi possono essere completamente implementate in software, mediante un pezzo di codice appositamente scritto che viene eseguito nello spazio utente. Una possibile implementazione nel caso di ridondanza doppia (ovvero basata su due canali) è la seguente. La variabile c, che rappresenta il canale, può assumere solo due valori (0 ed 1). Una coda di trasmissione può essere definita separatamente nello spazio utente per ciascun canale c, al fine di memorizzare i messaggi in sospeso, in attesa e da inviare. Ciò è necessario perché i buffer circolari, congiuntamente gestiti dal controller di rete e dal driver di dispositivo, non possono essere gestiti direttamente nello spazio utente. In particolare, Q[0] e Q[1] rappresentano le code associate ai due Wireless Adapter (WA) nel PC. Invece, W[0] e W[1] sono due flag nello spazio utente che tengono traccia dello stato dell'adattatore sottostante (inattivo o occupato):

- W[c] = NULL significa che il WA sintonizzato su c non è attualmente in trasmissione, ovvero il suo buffer circolare è vuoto e non ha frame da trasmettere.
- mentre W[c] = m significa che il messaggio m è stato trasferito nel ring buffer e l'adattatore si occupa della sua trasmissione in aria secondo le regole del protocollo WiFi.

Per come è stata descritta la precedente implementazione, un solo messaggio m (quello in capo alla coda dello spazio utente) può essere trasferito al ring buffer sottostante (quello che risiede sul WA), in un dato momento e questo vale per qualsiasi WA. Significa che in un dato istante, solo un messaggio può essere trasferito dallo spazio utente a qualunque WA. Le code Q[0] e Q[1] sono gestite in modo analogo ai ring buffer, ma nello spazio utente. Seguono una politica First In First Out (FIFO), ma consentono anche la rimozione degli elementi intermedi. L'interazione tra l'LRE e tali code avviene per mezzo di due funzioni principali: `enqueue (m, c)` e `dequeue (m, c)`. La prima inserisce il messaggio m_i nella coda di trasmissione Q[c], immediatamente dopo m_{i-1} (ovvero l'ultimo elemento inserito). Essa restituisce una condizione di errore se la coda è piena. La seconda funzione, invece, cerca il messaggio m nella coda Q[c] e, in caso di successo, rimuove l'elemento. La coda viene quindi deframmentata in modo che non ci siano buchi a causa del messaggio rimosso. In figura 39, viene mostrata una possibile implementazione C delle funzioni per gestire il layer Seamless Redundancy.

Inoltre, in un'ipotetica implementazione è prevista una terza funzione, chiamata `peek (c)`, che restituisce il più vecchio elemento in coda in Q[c], senza rimuoverlo dalla coda. L'attuale schema implementativo gestisce l'interazione tra lo spazio utente (che emette richieste di trasmissione sul collegamento ridondante) e l'LRE per mezzo della funzione `send (m)`. Questa funzione, inserisce il messaggio m in entrambe le code Q[0] e Q[1], quindi invoca la funzione `scheduler ()`, che ha il compito di trasferire i messaggi nelle code dei WA. La funzione `scheduler ()`, funziona come segue: per ogni coda Q[c], se l'adattatore correlato non è attualmente coinvolto in alcuna trasmissione (ovvero W[c] = NULL), `peek (c)` viene chiamata per ottenere il messaggio più vecchio nella coda (o, a seconda dell'implementazione effettiva, un riferimento a quel messaggio). Un'informazione che consente di identificare in modo inequivocabile tale messaggio viene quindi archiviata. La funzione `send_WA ()`, che può essere facilmente mappata sulla relativa funzione `sendto ()` di socket raw POSIX, che è quella che si occupa di inviare effettivamente il messaggio al WA. Sia la corretta ricezione di un frame ACK che la scadenza dell'ACKTimeout genera l'evento di fine della trasmissione nel driver di dispositivo, ed il risultato viene consegnato al LRE attraverso SDMAC [14]. SDMAC definisce alcune interfaccie di livello utente e di livello kernel, per esportare alcune funzionalità del protocollo MAC 802.11 a livello utente, oltre che per ampliarne alcune e di crearne di nuove, in base alle informazioni fornite dal driver di dispositivo. In risposta all'evento di fine ricezione, nel caso di PoW, la LRE invoca la funzione `ack ()` sul relativo canale RSTA del destinatario, mentre nel caso di RDA/Q invoca la funzione `ack_RDA ()`.

Uno degli obiettivi di progettazione era mantenere le differenze tra il codice per i due schemi di ridondanza che stiamo confrontando (PoW vs. RDA/Q) il più piccolo possibile. La funzione `ack ()` è estremamente semplice. Qualunque sia il risultato della trasmissione su c (esito positivo o negativo) rimuove il messaggio m

```
send(m_i) {
  enqueue(m_i, A);
  enqueue(m_i, B);
  scheduler();
}

ack(m_i, c, acked) {
  dequeue(m_i, c);
  W[c] = NULL;
  scheduler();
}

ack_RDA(m_i, c, acked) {
  dequeue(m_i, c);
  W[c] = NULL;
  if ( acked & W[!c]!=m_i ) {
    dequeue(m_i, !c);
  }
  scheduler();
}

scheduler() {
  for (c in A, B) {
    if ( W[c] = NULL ) {
      W[c] = peek(c);
      send_WA(W[c], c);
    }
  }
}
```

Figura 39. Implementazione di codice che gestisce le trasmissioni e ricezioni sul canale ridondato, come la funzione `send (mi)` e `ack (mi, c, acked)` che trasmettono e ricevono rispettivamente frame dati ed ack. Inoltre è prevista una funzione ack speciale per gestire le conferme duplicate in RDA/Q, chiamata `ack_RDA (mi, c, acked)` che si occupa anche di gestire la cancellazione in modalità RDA/Q. Infine la funzione `scheduler ()`, che esegue il trasferimento dei messaggi, dalla coda applicativa al ringbuffer. Fonte: [9]

dalla coda, imposta lo stato dell'adattatore correlato ad inattivo (W[c] = NULL) e invoca la funzione `scheduler ()` per riporre il prossimo messaggio in sospeso nella Q[c] (se vi sono) all'interno del ring buffer dell'adattatore. La funzione `ack_RDA ()` è più complessa, dal momento che è stata ideata per gestire l'evento XACK. In particolare, se la trasmissione del messaggio m sul canale c è andata a buon fine (come specificato dal set di ack a true) e lo stesso messaggio non è già stato inserito nell'altro adattatore (indicato come !c), come determinato dal suo stato (ad esempio W[!c] != m), viene rimosso dalla coda correlata (meccanismo di RDA/Q). Tutte le funzioni che interagiscono con la LRE (ad es. `send ()`, `ack ()` e `ack_RDA`

()) devono essere eseguite ovviamente in mutua esclusione, tramite POSIX mutex.

Per quanto riguarda gli indici di qualità della soluzione Wi-Red mediante RDA/Q, la latenza è la quantità misurata più importante per applicazioni in tempo reale sia in termini di soft real-time che hard real-time, a cui la capacità delle applicazioni di inviare i loro messaggi tempestivamente, all'interno scadenze predefinite, è direttamente correlata. La capacità di una tecnica di trasmissione di limitare le latenze è essenziale per prevenire una crescita eccessiva delle code. In effetti, più basso è il numero di messaggi in attesa, in una determinata coda, più breve è la latenza che subirà qualsiasi nuovo messaggio inserito successivamente.

Un'altra misura importante è il consumo di larghezza di banda. Essendo lo spettro wireless una risorsa condivisa, lo spreco di larghezza di banda si ripercuote sulle prestazioni degli altri nodi che utilizzano la stessa frequenza per comunicare.

Software defined MAC

La possibilità di accedere al livello MAC IEEE 802.11 tramite primitive di basso livello ed in particolare di gestire il singolo tentativo di trasmissione attraverso software eseguito in spazio utente, è un prerequisito fondamentale per molti scenari applicativi basati su WiFi e caratterizzati da vincoli temporali. Le tecniche di ridondanza, la pianificazione del traffico (scheduling) e le tecniche Time Division Multiple Access (TDMA) sono solo alcuni esempi significativi per cui sia importante quest'accesso. In tal senso è stata utilizzata una nuova architettura software, chiamata Software Defined Mac (SDMAC) [14], che si basa su PC Linux convenzionali dotati di adattatori WiFi di uso comune e che permette alle applicazioni un controllo diretto dei frame in trasmissione. La sua attuazione e valutazione sperimentale su un banco di prova reale ha mostrato che l'integrazione di SDMAC nello stack protocollare di Linux è una soluzione valida, in termini di latenze introdotta dai componenti software e hardware ed è adatta ad una vasta gamma di applicazioni soft real-time.

Sebbene il funzionamento del controllo di accesso al mezzo Medium Access Control (MAC) è definito accuratamente nelle specifiche IEEE 802.11, sono fornite poche indicazioni su come deve essere implementato nei dispositivi del mondo reale. Attualmente molti WA si basano su un'architettura SoftMAC, e questo significa che l'entità di gestione dei sottolivelli MAC (MLME) non è implementata nell'hardware (o nel firmware) bensì viene eseguita in software dalla CPU del dispositivo host. Il kernel Linux, ad esempio, fornisce un set completo di driver di dispositivo per la gestione dei dispositivi SoftMAC. Al contrario, gli adattatori FullMAC si prendono cura da soli dell'intero stack protocollare IEEE 802.11 e sollevano l'host dalle relative attività di gestione. Inutile dire che sono più complessi e costosi di quelli SoftMAC. Vale la pena notare che, in SoftMAC, solo le operazioni non critiche in termini di tempo sono delegate al software, mentre tutte le azioni che richiedono tempistiche precise (ad esempio rispondere ad un frame di dati con il frame ACK correlato all'interno di SIFS, gestire il conto alla rovescia del backoff, e così via) devono essere eseguite dall'adattatore (parte nel firmware e parte nell'hardware, a seconda dell'implementazione specifica).

Nel passato sono state analisi di ricerca scientifica molte soluzioni finalizzate

a migliorare il comportamento di base del WiFi per mezzo di strati software sovrapposti al di sopra del livello Data Link. In genere, questi overlay implementano una sorta di meccanismo di accesso distribuito, basato ad esempio su TDMA, che fornisce determinismo comunicativo superiore. Tuttavia, questi non possono alterare in nessun modo le strategie di accesso di base previste dal livello MAC IEEE 802.11, ovvero la funzione di coordinamento distribuito (DCF) e suoi derivati (PCF ed HCCA). Al limite, riconfigurando i parametri operativi dell'adattatore, è possibile, ad esempio, disabilitare il backoff casuale (impostando la finestra di contesa al valore 0), o ottimizzare l'accesso a priorità delle stazioni (selezionando una durata personalizzata per l'Arbitration Interframe Space (AIFS).

SDMAC propone un'architettura software completamente nuova. Dal punto di vista pratico, l'obiettivo degli sviluppatori è rendere le applicazioni in grado di accedere direttamente alle operazioni di livello inferiore eseguite dall'adattatore WiFi. In particolare, SDMAC considera un'operazione di base come eseguire una trasmissione di un frame mediante un solo tentativo (disabilitando le ritrasmissioni automatiche) ed ottenere una conferma tempestiva quando il relativo frame ACK è stato ricevuto (one-shot). Inoltre, lo standard prevede trasmissioni non confermate, che prevedono l'invio di un frame senza l'attesa della conferma tuttavia, anche se necessario per gestire i servizi multicast, non sono affidabili come quelle confermate, e perciò non utilizzate in applicazioni tempo-critiche. Sfruttando queste primitive specifiche a livello MAC possono essere progettate ed attuate strategie di accesso al mezzo efficaci, che superano la maggior parte delle limitazioni delle proposte esistenti (che si basano su servizi di livello Data-Link). Lo scopo principale di SDMAC è consentire alle applicazioni di acquisire il pieno controllo in fase di esecuzione di ciò che viene inviato sul supporto wireless, in maniera da abilitare strategie più sofisticate per coordinare gli scambi di frame e consentire il rispetto dei vincoli di temporizzazione dei sistemi di controllo distribuiti in tempo reale. Fondamentalmente le applicazioni possono essere implementate sia nello spazio kernel che in quello utente. Sebbene la prima opzione (l'implementazione nello spazio kernel) probabilmente fornisce maggiori prestazioni e determinismo, ci concentreremo sulla seconda implementazione (quella nello spazio utente). Il motivo, sia del software implementato in questa tesi che di SDMAC, è che si sta cercando un framework flessibile in grado di facilitare l'implementazione e il testing per una vasta gamma di soluzioni diverse. La programmazione nello spazio utente è di gran lunga più semplice e veloce e consente di ottenere molte più informazioni diagnostiche. Chiaramente, applicare gli stessi concetti a livello kernel non è precluso ma questa opzione richiede modifiche sostanziali ai driver degli adattatori wireless e del kernel di Linux. I framework che verranno descritti di seguito si concentrano esplicitamente su adattatori WiFi commerciali standard. Inoltre, sono necessarie solo lievi modifiche ai driver di dispositivo, che possono essere quindi aggiornati rapidamente e facilmente. È abbastanza chiaro che la fattibilità dell'approccio proposto dipende fortemente dalla capacità della piattaforma reale (incluso l'hardware ed il sistema

operativo) di eseguire le operazioni in modo tempestivo. Latenze aggiuntive e jitter (in questo caso si intende una variazione statistica nel ritardo di ricezione dei pacchetti trasmessi tra una sorgente ed una destinazione, solitamente causata dai tempi di coda variabili di un pacchetto durante il suo percorso sorgente destinazione) infatti potrebbero far si che la stessa implementazione su piattaforme differenti dia risultati diversi. Queste variabili potrebbero dunque annullare la maggior parte dei benefici previsti o, al limite, causare una violazione dei vincoli in modo tale da compromettere il comportamento corretto. Per questo motivo gli sviluppatori di SDMAC hanno eseguito un approfondito esperimento preliminare per valutare quanto la gestione delle trasmissioni dei frame nello spazio utente incida sul determinismo. L'analisi sperimentale che è stata effettuata ed i risultati che sono stati ottenuti sono molto importanti per i progettisti e costituiscono la chiave per consentire lo sviluppo e il collaudo di un'intera famiglia di tecniche volte a migliorare l'affidabilità e il determinismo su reti WiFi. Seamless Redundancy, scheduling centralizzato del traffico basato su deadline e schemi TDMA sono solo alcuni dei contesti applicativi che possono beneficiare dall'adozione di SDMAC.

Il paradigma SDMAC può essere visto come un Application Programming Interface (API) che consente alle applicazioni di gestire con precisione la trasmissione (e la ricezione) dei frame, oltre a configurare (anche in fase di esecuzione) alcuni parametri rilevanti di IEEE 802.11, che di solito non sono accessibili attraverso l'interfaccia socket. Per quanto riguarda i vincoli temporali, SDMAC può essere classificato come un sistema soft real-time visto che possono essere fissati solamente dei limiti probabilistici sul numero di pacchetti inviati entro una certa scadenza. In effetti, anche se esistono sistemi real-time come RTAI o Xenomai, i driver di dispositivo real-time per gli adattatori WiFi sono raramente disponibili e limitati a poche implementazioni di prototipi, come RTnet. Nei sistemi hard real-time sono previsti limiti rigorosi in termini di latenza, al punto che le scadenze non possono non essere rispettate (hard real-time significa che tutti i pacchetti devono arrivare entro una certa scadenza). Nell'esempio di implementazione parziale di SDMAC presentata in [14], ogni singolo tentativo di trasmissione di un frame è effettuato sotto controllo diretto dell'applicazione. Per fare ciò, le ritrasmissioni automatiche a livello MAC sono state disabilitate.

Per consentire alle applicazioni di gestire in modo personalizzato le trasmissioni, il driver si occupa di comunicare l'esito di ogni singolo tentativo di trasmissione: *success*, alla ricezione del frame ACK correlato, o *failure*, in caso di errori di trasmissione (ad es. dopo che è trascorso un intervallo ACK Timeout). La propagazione degli eventi di ACK e ACKT dal driver fino all'applicazione richiede alcune lievi modifiche al driver stesso, come già anticipato. Come lavoro futuro, in vista di una vera e propria implementazione SDMAC (in grado di gestire configurazioni più complesse), è stato pianificato dai proponenti di SDMAC di allargare la semantica delle API correlate. Per esempio, gli sviluppatori dovrebbero essere autorizzati a impostare, in base al pacchetto, il numero massimo di tentativi di trasmissione e

le velocità di trasferimento a cui sono trasmessi. Ciò consente di definire schemi di ritrasmissione in cui i tentativi sono gestiti parte in software (dalla CPU) e parte in hardware / firmware (dal livello MAC e della scheda di rete).

Tali schemi misti possono offrire il miglior compromesso tra flessibilità e prestazioni. Alcuni parametri importanti che, dovrebbero essere resi direttamente disponibili agli sviluppatori di applicazioni tramite le API già citate, sono quelli relativi alla qualità del servizio (QoS), ovvero i valori TXOP, AIFSN, CWmin e CWmax per ognuna delle 4 categorie di accesso previste da IEEE 802.11e. In questo modo, le caratteristiche di ciascun flusso di dati in tempo reale possono essere regolate con precisione e sarebbe possibile disabilitare il backoff casuale per alcuni di loro in modo da aumentare ulteriormente il determinismo. Inoltre, l'API SDMAC dovrebbe trasmettere all'applicazione alcune informazioni utili che sono disponibili solo all'interno driver. Ciò include le statistiche relative alla trasmissione e ricezione dei pacchetti, indicatori di qualità del canale, ma anche altri dati specifici dell'applicazione. Ad esempio, i timestamp precisi, acquisiti alla ricezione dei pacchetti all'inizio dell'Interrupt Service Routine (ISR) possono essere successivamente sfruttati dall'applicazione per regolare la base dei tempi del nodo secondo un protocollo di sincronizzazione del clock. Ad esempio, il protocollo di sincornizzazione Reference Broadcast Infrastructure Synchronization Protocol (RBIS) [15], concepito specificamente per le reti WiFi, è basato solo su timestamp rilevati sui frame ricevuti.

SDMAC è dotato di funzioni di interfaccia sia a livello utente che a livello kernel. In ogni caso, l'aggiunta di un nuovo tipo di informazione a quelle già scambiate tra il driver e l'applicazione (o viceversa), richiede la modifica del driver stesso. Questo porta a ulteriori sforzi richiesti agli sviluppatori per l'integrazione delle nuove funzionalità nel driver, in caso di aggiornamenti alle nuove versioni dello stesso o nel caso di porting su diversi dispositivi. Chiaramente l'architettura SDMAC è rilevante se e solo se la sua reale implementazione si rivelasse deterministica e abbastanza veloce. Per questo motivo, gli sviluppatori si sono concentrati sulla caratterizzazione sperimentale di quella che è, probabilmente, la primitiva SDMAC più interessante (e critica) ovvero la trasmissione di frame one-shot (a singolo frame).

La capacità di SDMAC di gestire i tentativi di trasmissione a singolo frame è essenziale in molti contesti applicativi. Ad esempio, l'efficacia dei meccanismi di DA, concepiti per ridurre la larghezza di banda sprecata nelle soluzioni WiFi che sfruttano la ridondanza (Wi-Red), è legata direttamente alla capacità di interrompere prontamente la trasmissione di un pacchetto a seguito della ricezione di un ACK, corrispondente ad un frame, sull'altro adattatore. Inoltre, approcci proattivi di DA che sfruttano euristiche basate, ad esempio, su statistiche raccolte sui nodi di trasmissione o sulle caratteristiche del supporto wireless (ad es. perdite di pacchetti, traffico di rete, potenza del segnale, schemi di modulazione), possono trarre vantaggio da SDMAC, perché possono distribuire dinamicamente il traffico sui due canali.

L'accesso distribuito al canale è un altro esempio di meccanismi che potrebbe beneficiare di SDMAC. Quando si parla di time-driven traffic scheduling ed in particolare di TDMA, la capacità di un nodo di inviare frame a istanti precisi e con basso jitter è un prerequisito rigoroso per gestire correttamente i time-slot (intervalli di tempo dedicati ad un dato nodo per accedere in modo esclusivo al mezzo condiviso) e perfezionarne la durata. Ciascuno di questi slot, infatti, corrisponde all'intervallo di tempo all'interno del quale è consentito l'accesso esclusivo a un nodo specifico al mezzo condiviso, prevenendo in tal modo le collisioni in anticipo. TDMA non si basa su superframe inviati periodicamente dal coordinatore della rete, ma si basa su una base dei tempi a cui tutte le stazioni si adeguano (ottenuta mediante protocolli di sincronizzazione), inoltre la qualità della sincronizzazione può influire sulla scelta della durata dei time-slot. Un altro punto rilevante è dato dallo scheduling di traffico deadline-driven (fine del processamento del traffico entro un certo istante di tempo). Avere un feedback tempestivo su quando le trasmissioni nell'aria terminano consente di attuare una gamma più ampia di strategie di scheduling.

Inoltre, tecniche euristiche basate su informazioni statistiche fornite dal driver, possono essere sfruttate in modo proattivo anche su singoli canali, per aumentare ulteriormente il determinismo. Come un esempio, la capacità dell'algoritmo di scheduling Earliest Deadline First (EDF) di ottenere livelli di prestazione accettabili (ovvero, per aumentare l'utilizzo della rete il più possibile) è correlato al tempo impiegato per attivare la trasmissione del frame successivo dopo aver rilevato il completamento del precedente.

Una delle implementazioni che cercano di unire i vantaggi del traffic scheduling e di TDMA, seguendo un approccio distribuito e consentendo allo stesso tempo la comunicazione multihop e la trasmissione di flussi aperiodici, è SchedWiFi. Funziona adattando il traffico ad alta priorità in finestre temporali predefinite, cosi da garantire ai pacchetti a priorità elevata un canale più affidabile e deterministico. SchedWiFi fa uso di un modulo specifico, chiamato Time-Aware Shaper, per impedire la trasmissione di traffico indesiderato durante il tempo riservato alle finestre. Questo modulo, in una versione parzialmente semplificata che può mostrare un leggero aumento degli sprechi di larghezza di banda (a causa del jitter più elevato sui limiti delle finestre), potrebbe essere implementato nello spazio utente (sotto vincoli soft real-time) sfruttando SDMAC.

Come precedentemente riportato, l'obiettivo della fase preliminare di presentazione di SDMAC alla comunità scientifica non era di definire minuziosamente l'API SDMAC (ovvero i servizi che fornisce l'interfaccia software), ma piuttosto di dimostrare che l'esecuzione di questo software su apparecchiature commerciali soddisfa le esigenze di molte applicazioni pratiche. In questa misura, è stata sviluppata una versione prototipo di SDMAC ed è stata effettuata una valutazione sperimentale. A livello di spazio utente, l'applicazione interagisce con l'adattatore WiFi per mezzo di due funzioni, ovvero `send(data)` e `wait_ack()`. L'architettura SDMAC, presentata in figura 40, si compone di due percorsi: uno di invio dei dati applicativi

ed uno di ricezione degli ACK per ciascun pacchetto trasmesso correttamente. La funzione di invio dei dati può essere direttamente mappata su socket raw POSIX, e in particolare sulla funzione `sendto()`. Il percorso di invio, che va dall'esecuzione della funzione `send(data)` fino al punto in cui il pacchetto si trova effettivamente in coda di trasmissione dell'adattatore WiFi (il ring buffer), non richiede alcuna modifica al driver di dispositivo e corrisponde esattamente allo stack protocollare tipico di un sistema Linux. Non è stato eseguito alcun tipo di ottimizzazione sul percorso di invio, visto che il contributo più significativo in termini di latenza in questa fase è correlato all'hardware della scheda di rete e al percorso di ricezione dell'ACK.

La ritrasmissione automatica dei frame a livello MAC, normalmente gestito dalla scheda di rete, deve essere disabilitato. Farlo di solito non è complesso, ma dipende dal tipo di driver e dal tipo di scheda di rete. Alcuni driver consentono di impostare il numero massimo di tentativi direttamente a livello di spazio utente. In questo caso, l'impostazione di quel parametro a 0, ha l'effetto di limitare il numero di tentativi di trasmissione per pacchetto esattamente ad un frame. In altri casi, questo parametro è incorporato nel driver. In tal caso, il suo codice sorgente deve essere disponibile, e deve essere localizzato il punto più appropriato in cui il parametro di ritrasmissione può essere impostato. Gli esperimenti fatti nell'articolo di descrizione di SDMAC si basano sul driver Linux ath9k per Atheros Adapter IEEE 802.11, che è di gran lunga la soluzione più popolare nella comunità Linux e nell'ecosistema di ricerca, è open source e non richiede alcun firmware proprietario. Dovrebbe essere possibile implementare SDMAC anche su altri tipi di dispositivi ed in particolare su quei dispositivi i cui driver sono basati su architettura SoftMAC e che non fanno affidamento su firmware proprietario.

L'algoritmo di adattamento della velocità e codifica di trasmissione (*rate adaptation*) per ath9k è Minstrel, che ha il compito di determinare i migliori rate per ciascuna trasmissione e ritrasmissione. In particolare, si avvale di una struttura di memoria (descrittore TX), che è memorizzata nel ring buffer insieme al pacchetto da trasmettere. Il descrittore TX specifica anche il numero di tentativi di trasmissione consentiti e la velocità con cui vengono effettuati. Questa struttura può gestire al massimo 4 diverse serie di trasmissioni, eseguite in sequenza dall'adattatore. In particolare, i campi tx_tries0, tx_tries1, tx_tries2 e tx_tries3 specificano il numero di tentativi di trasmissione eseguiti, per ciascuna tipo di serie, alla velocità di trasmissione associata. Quando la scheda di rete è pronta per l'invio un nuovo pacchetto sul mezzo di trasmissione, preleva dal ring buffer sia il payload che il descrittore TX. A causa di quanto detto sopra, le ritrasmissioni possono essere facilmente disabilitate impostando tx_tries0 a 1 e tx_tries1/2/3 a 0. Inoltre vale la pena notare che, impostando i valori dei campi, l'algoritmo Minstrel di adattamento della frequenza non funziona più. Come conseguenza, l'adattamento del tasso in SDMAC deve essere gestito direttamente dall'applicazione (o, in alternativa, con il codice SDMAC). Per assicurare il controllo completo sulla

trasmissione, l'API SDMAC deve fornire un modo per impostare la velocità desiderata per ogni serie, consentendo quindi agli algoritmi custom di adattamento della velocità di funzionare a livello applicazione. La disabilitazione dell'algoritmo Minstrel sul driver non rappresenta un problema per valutare l'esperimento eseguito mediante SDMAC perché, come spiegato nella sezione prima, per ottenere misure affidabili abbiamo deciso di disabilitare l'adattamento della velocità e utilizzare una velocità di trasmissione fissa.

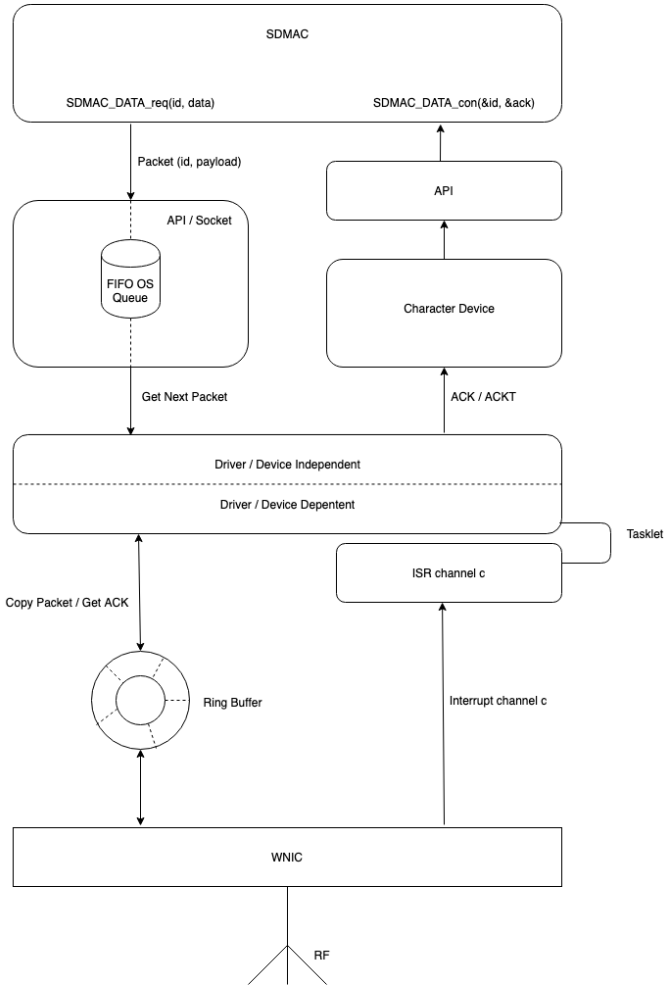

Figura 40. Architettura SDMAC. Come si può vedere, la catena di trasmissione e ricezione gestita dall'architettura, prevede la collaborazione di varie entità (come i driver, l'SO, ed il livello applicativo) che si passano dati tramite strutture SDMAC appositamente create in accoppiata con le APIs SDMAC.

Il percorso inverso sulla stazione di origine ha a che fare con le conferme relative alla corretta consegna dei pacchetti al nodo destinatario. Alla ricezione del frame ACK o quando scade l'ACK Timeout, l'adattatore wireless genera un interrupt

alla CPU. Il relativo ISR viene eseguito il prima possibile dal sistema operativo, in modo da far fronte ai compiti più urgenti (quello di prelevare l'ACK/ACKT). In genere, ciò introduce un piccolo ritardo casuale dell'ordine delle decine di microsecondi. L'ulteriore elaborazione è effettuata per mezzo di un tasklet, che è il meccanismo usato abitualmente dal sistema operativo per rinviare l'esecuzione del codice. Il tasklet rappresenta il confine tra la cosiddetta metà superiore (ovvero ISR eseguita in un contesto di interrupt disabilitati) e metà inferiore di un driver di dispositivo (eseguita in un contesto kernel). Ovviamente, visto che la parte ISR spesso disabilita alcuni o tutti i canali di interrupt e che di conseguenza aumenterebbe la latenza del servizio di altri dispositivi connessi alla CPU, si tende a minimizzare la parte di ISR e a lasciare ai livelli superiori (kernel) il lavoro più oneroso, in modo da aumentare la condivisione della CPU da parte degli altri dispositivi esterni. Il kernel ha il compito di determinare quando pianificare l'esecuzione del codice associato al tasklet. Il codice del driver può essere considerato diviso in due parti. La prima, che dipende dal dispositivo, fa accesso a registri specifici dell'adattatore wireless per acquisire tutte le informazioni necessarie al livello MAC. Invece la seconda è indipendente dal dispositivo, il che significa che lo stesso codice viene utilizzato per gestire diversi tipi di hardware. La maggior parte dei driver di dispositivo per adattatori wireless sono strutturati in questo modo e ath9k non fa eccezione. Gli sviluppatori di SDMAC hanno deciso di posizionare il codice che rileva l'esito dell'ultima trasmissione (ACK/ACKT) nella parte indipendente del dispositivo, e in particolare, nell'implementazione di SDMAC, nella funzione in `ieee80211_tx_status` del driver ath9k. Il motivo è che hanno voluto ridurre al minimo le modifiche al codice del driver, in vista del suo porting su dispositivi diversi da ath9k e della gestione delle versioni di driver più recenti. A discapito di un lieve aumento della latenza, è stata privilegiata l'utilizzabilità di SDMAC su architetture reali. L'evento di ricezione viene trasferito dallo spazio del kernel alla funzione `wait_ack ()` nello spazio utente mediante un device a caratteri. Nell'attuale implementazione è stato usato un semaforo kernel nello spazio del kernel e viene richiamata la versione bloccante della system call `read ()` sul dispositivo a caratteri nello spazio utente. La gestione di questa interazione (tra spazio kernel e spazio user) con l'attesa passiva evita l'attesa (da parte del kernel che deve servire il livello user) dovuta al polling. Anche se, per sistemi in tempo reale, esistono metodi per la sincronizzazione e il trasferimento di dati tra kernel e spazi utente che mostrano un maggiore determinismo, l'uso di un dispositivo intermedio, come il device a caratteri, è una soluzione tecnologica molto comune per applicazioni soft real-time e per lo più una scelta standard in sistemi Linux, specialmente quando driver di dispositivo in tempo reale non sono disponibili. In alternativa, invece di avere un dispositivo dedicato per SDMAC, è possibile utilizzare la system call `ioctl()` per scambiare dati direttamente con il driver del dispositivo. Nell'implementazione di SDMAC usata in questa tesi è stato preferito l'uso di un dispositivo a caratteri perché l'uso di `ioctl()`, che fornisce prestazioni simili, implica ulteriori

modifiche al software del driver di dispositivo, quindi peggiora la portabilità. Secondo la scelta fatta, l'interazione tra il driver di dispositivo e SDMAC deve essere gestito attraverso poche e ben definite funzioni. Inoltre, nel nuovo driver di dispositivo WiFi c'è una tendenza a sostituire la funzione `ioctl()` con l'interfaccia netlink, che purtroppo aumenta la latenza e riduce il determinismo, se confrontato con operazioni di lettura / scrittura su dispositivi a caratteri o `ioctl()`.

Implementazione Software

Poiché è stato dimostrato che l'introduzione di Seamless Redundancy, nei dispositivi WiFi, porta notevoli benefici in termini di affidabilità e determinismo dal punto di vista del soft real-time, la soluzione proposta in questa tesi si appoggia su queste tecniche per raggiungere prestazioni migliori. In aggiunta è stata utilizzata la libreria che implementa Software Defined MAC (SDMAC) che, come già detto nel precedente capitolo, permette di configurare e leggere una moltitudine di parametri della schede di rete (ad esempio ottenere il valore di SNR, imporre il numero di ritrasmissioni hardware etc.) senza dispendio significativo di latenza temporale. Inoltre, con l'introduzione delle AC, è possibile impostare livelli di QoS del traffico separatamente secondo le 4 classi (Best Effort, Background, Voice, Video) così da date una priorità differente ai pacchetti e, conseguentemente, di diminuire in media i tempi di coda dei pacchetti appartenenti a certe classi e di rendere più costante il data-rate di ognuna di esse (in media, pacchetti di classi uguali e di simile dimensione, hanno pressoché lo stesso rate di accodamento nel ring buffer). Per sfruttare il concetto di Seamless Redundancy, riuscendo a mantenere il controllo dei pacchetti inviati (trasmessi e confermati) e di quelli da ritrasmettere, si è scelto di implementare un software di livello applicazione che, sfruttando le APIs SDMAC, trasmetta mediante il protocollo User Datagram Protocol (UDP) i pacchetti provenienti dal livello applicazione. Si è scelto di utilizzare UDP poiché, rispetto al TCP, la trasmissione di un datagram è più veloce e questo è dovuto al fatto che non è previsto nessun riordino dei pacchetti, non è confermato ed inoltre non ha il controllo di congestione. Infine non prevede ritrasmissioni al livello trasporto dunque è più adatto, rispetto a TCP, perché non presenta comportamenti indeterministici in tal senso. Inoltre, grazie alla compatibilità con IP, aumenta il livello di scalabilità rispetto a quei sistemi che si appoggiano su protocolli proprietari. Il software in questione consentirà ad una data applicazione di accedere ad un insieme di funzionalità di rete che gli permetterebbero di:

- Sfruttare il sistema di Seamless Redundancy senza preoccuparsi della sua gestione.
- Aumentare le sue prestazioni, riguardanti la parte di rete, in termini di determinismo e affidabilità, potendo così rispettare i vincoli imposti dal protocollo soft real-time implementato.

- Sfruttare le diverse classi di servizio in modo da aggiungere una priorità al traffico in uscita.

- Migliorare le prestazioni di quei task basati su quei protocolli soft real-time che prima non potevano essere implementati o che avevano basse prestazioni, sfruttando anche il concetto di priorità del traffico.

In linea generale, come mostrato in figura 41, l'architettura software presentata in questo documento permetterebbe a tutte le applicazioni soft real-time di interagire, mediante APIs, con un layer applicativo comune che si preoccupa di garantire i requisiti di determinismo e affidabilità necessari. Ovviamente, avere a che fare con due schede di rete anziché una, significa gestire due flussi dati in uscita ed in entrata, quindi comporta un dispendio notevole di risorse computazionali (sopratutto I/O), dell'etere, ed una gestione dei pacchetti dati duplicati a livello applicativo. Tuttavia con l'implementazione di tecniche DA (come RDA o approcci proattivi) si può, ad esempio, evitare di trasmettere pacchetti che sono già stati trasmessi correttamente su uno dei due canali (evitando cosi di trasmettere due volte lo stesso pacchetto nei casi in cui ciò non sia utile). In questa soluzione si è scelto di implementare un meccanismo RDA che intervenga sulle code (RDA / Q). Probabilmente seguire un approccio proattivo avrebbe portato ad ulteriori benefici, tuttavia si è scelto di implementare RDA / Q perché i risultati che esso produce sono certi, mentre i metodi proattivi sono spesso basati su euristiche.

Prima di entrare nei dettagli dell'implementazione e delle caratteristiche del software in questione, sarà fatta una panoramica generale sullo scenario di rete, mostrato in maniera generale in figura 42, e sul sistema che lo ospiterà.

Per quanto riguarda lo scenario in cui agirà il software, sono previsti 3 dispositivi di rete:

- Un sistema Client-RSTA che invia pacchetti di dati su due canali di rete.

- Due AP, sintonizzati rispettivamente su un canale diverso, che ricevono i dati provenienti dal RSTA di origine ed inviano l'ACK di conferma.

La scelta del modello proposto è correlata al fatto che generalmente, nelle reti WiFi (e nei sistemi Wireless), si posizionano gli AP in maniera tale da creare delle aree sovrapposte, nelle quali due o più AP possano ricevere in media lo stesso traffico. Questa scelta progettuale favorisce la gestione del handover che, in condizioni di mobilità, è fondamentale per evitare perdite (il momento migliore per eseguirlo è quando una STA è vista da più AP, e con una scelta basata sulla direzione di spostamento della stessa STA, si può individuare verso quale AP essa si stia dirigendo in modo da effettuare il cambio di associazione da un AP quell'altro). Nel nostro caso è possibile sfruttare il principio di sovrapposizione delle aree BSS per ridondare anche il ricevitore. Visto che, generalmente gli AP fisici sono collegati da

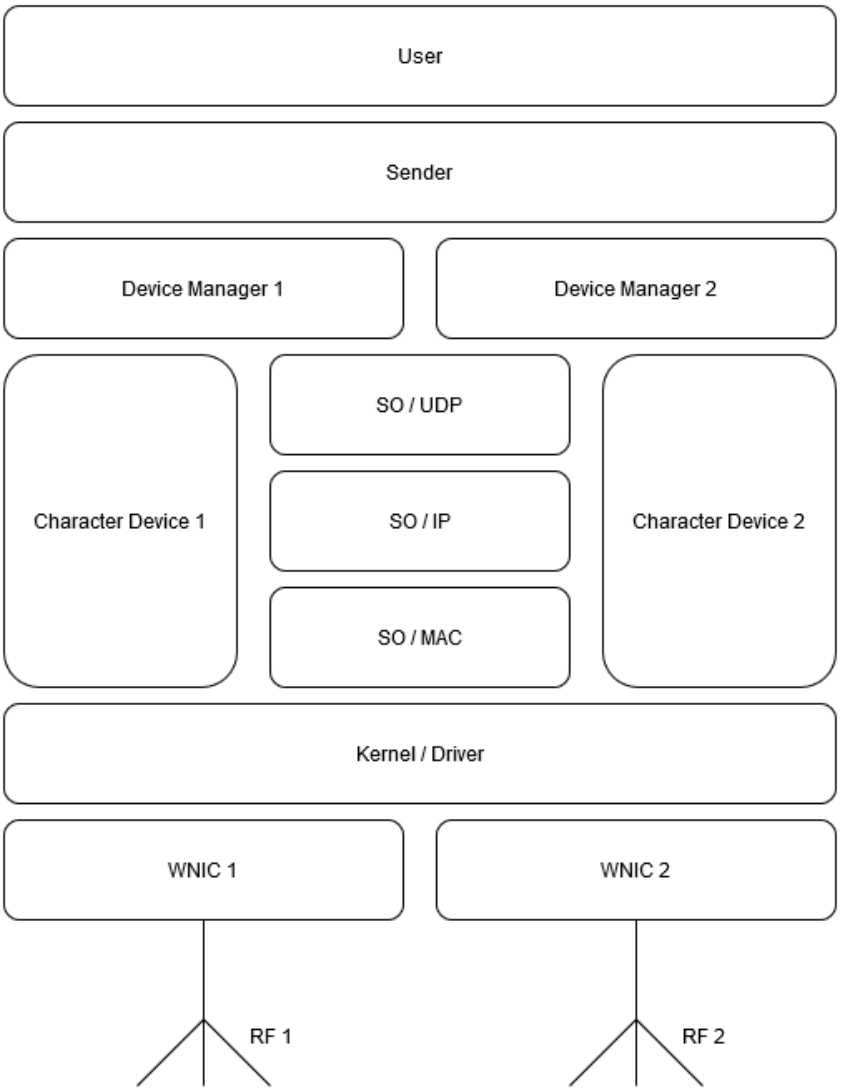

Figura 41. Architettura software completa. Al livello superiore, le applicazioni utente interagiscono con il layer applicativo sottostante composto dal Sender e dai singoli Device Mananger. Il livello sottostante a quello applicativo è composto dai moduli dell'SO (UDP, IP, MAC) e da un modulo speciale che rappresenta il device a caratteri (uno per ciascun canale WiFi). Il livello kernel, composto dal driver di dispositivo, interagisce con le schede di rete, WNIC1 e WNIC2, mediante interrupt per gestire la trasmissione/ricezione dei frame.

un'infrastruttura, che tipicamente si appoggia su Ethernet, e dato che un controllore (che trasmette e riceve rispettivamente comandi e feedback dagli end system) è generalmente collegato alla stessa infrastruttura, una perdita di un pacchetto

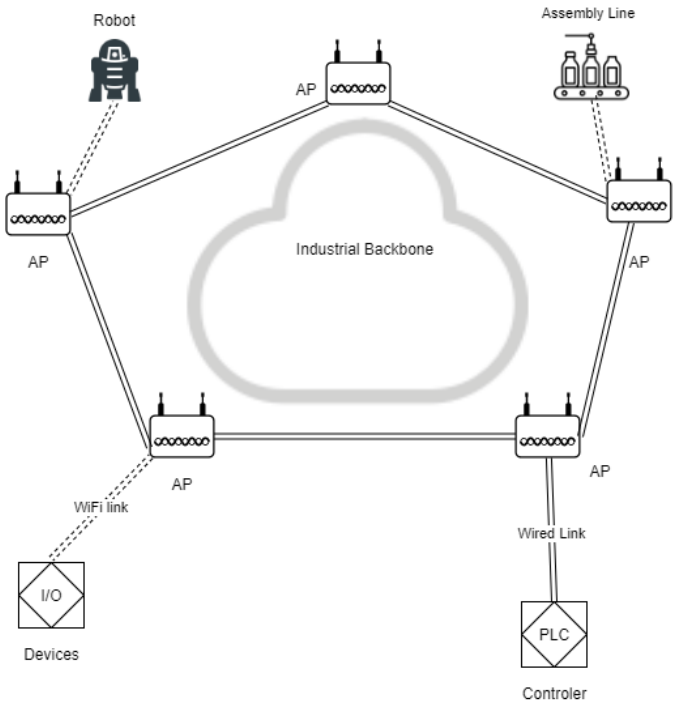

Figura 42. Tipica architettura di una rete industriale. Come si può vedere, l'infrastruttura principale è composta da una rete, a topologia variabile, di nodi AP. I collegamenti degli end-system vengono invece realizzati in WiFi.

da parte di un AP potrebbe non significare necessariamente che il ricevitore non lo abbia ricevuto, poiché quel pacchetto potrebbe essere stato ricevuto anche da altri AP che poi lo trasmetteranno al destinatario finale. Naturalmente è necessario un meccanismo di gestione delle copie da parte degli AP. Tuttavia, visto che probabilmente anche un controllore può utilizzare, con degli adattamenti, l'architettura software proposta, probabilmente si potrebbe evitare di implementare tale meccanismo e lasciare all'end system di controllo la gestione dei pacchetti duplicati ricevuti. Oltretutto, dato che solitamente un controllore è dotato di hardware più prestante, non sarebbe significativo l'impatto di una soluzione di questo tipo. Comunque tale descrizione non è argomento di questa tesi. Tornando all'analisi dello scenario di rete, gli AP sono configurati con indirizzi IP che fanno parte dello spazio di indirizzamento di BSS differenti, mentre la RSTA, dotata di due interfacce wireless lan (wlan0 e wlan1), è configurata con indirizzi IP appartenenti allo stesso spazio di indirizzamento delle BSS di appartenenza di ciascun AP. Questo per fare in modo che, attraverso la gestione delle sovrapposizioni delle BSS insita nel protocollo IEEE 802.11, ciascuna AP riceva solamente i frame provenienti dalla BSS di appartenenza. Il sistema Client RSTA trasmette una serie di datagram effettuando trasmissioni su ciascuna interfaccia wlan, ognuna delle quali è connessa

ad un determinato AP, e per ogni pacchetto aspetta un eventuale ACK (oppure ACKT, ACK Timeout, un evento generato dalla scheda per segnalare la mancata ricezione dell'ACK) inviato dall'AP. Si ritiene importante sottolineare che mentre la trasmissione dei dati dal RSTA di origine è realizzata in UDP mappato su IP mediante la chiamata alla system call *sendto()*, la ricezione della conferma è realizzata mediante la chiamata della system call `read ()` effettuata su un device a caratteri, evitando cosi il modulo IP ed UDP dell'OS. Naturalmente tale scelta di protocollo minimizza notevolmente l'overhead degli header in termini di dimensione dei frame. Questo poiché, data l'architettura software scelta, i frame dati necessitano dei protocolli UDP ed IP mentre, per quanto riguarda gli ACK, questi due protocolli non si rivelano necessari al fine del funzionamento globale del sistema. Infatti gli ACK che vengono ricevuti dal livello applicativo non sono altro che la propagazione, dell'evento ACK/ACKT generato dalla scheda, dal livello driver fino al livello applicativo stesso. Le conferme (ACK/ACKT) ricevute dal driver di dispositivo da parte del WA sono inoltre propagate direttamente al livello applicativo attraverso l'uso del device a caratteri (che fa da interfaccia) e di una struttura dati (`rate_data_t` di SDMAC) in comune. Nel caso in cui la conferma, per un dato pacchetto, non sia stata ricevuta entro l'ACKTIMEOUT impostato dalla scheda di rete, quest'ultima genera un evento ACKT il quale viene mappato dal driver sulla struttura dati SDMAC. In questa architettura si è scelto di utilizzare il device a caratteri anziché, ad esempio, il comando ioctl() (per parlare direttamente con il driver), per non influire sulla portabilità, visto che l'utilizzo di ioctl() richiederebbe modifiche al driver. La figura 44, mostra il percorso di trasmissione e di ricezione, descrivendo ciascuna fase dalla scheda di rete sino al livello applicativo. La dimensione del frame ACK ricevute dal driver e l'elaborazione eseguita su tale frame sono ridotte, e non necessarie nel nostro caso, rispetto al pacchetto dati che contiene anche gli header UDP e IP e, di conseguenza, anche la latenza di trasmissione e ricezione del primo è mediamente inferiore rispetto al secondo.

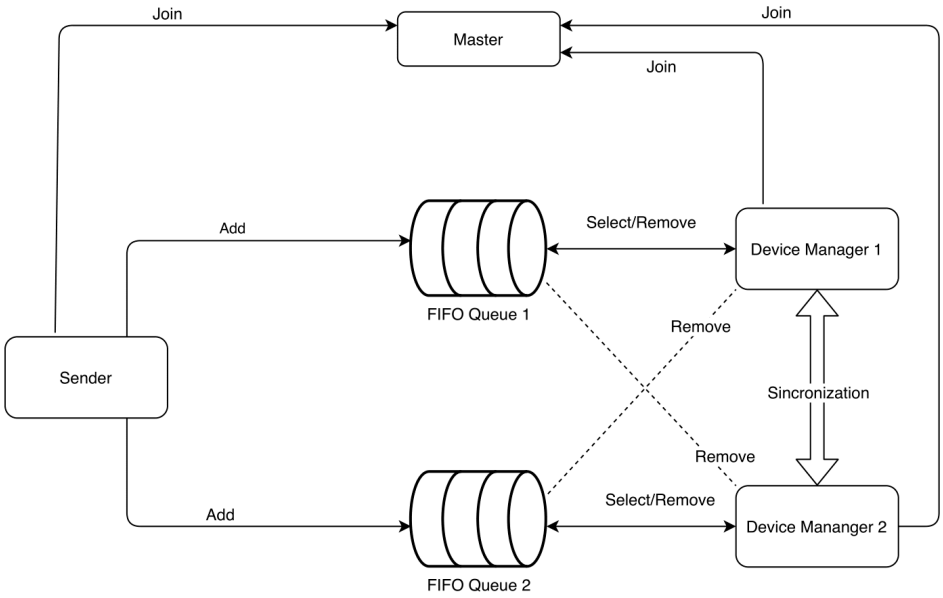

Figura 43. Modello ad entità dei thread implementati a livello applicativo dello stack software di questa architettura special purpose. Come si può vedere, il modello implementato prevede 3 entità principali quali il Master, il Sender ed i Device Manager. Sono raffigurate anche le code e sono rappresentate le operazioni principali che coinvolgono le entità ed i dati.

Entrando più nel dettaglio dell'implementazione software e del codice, saranno analizzati i processi di esecuzione ed il loro funzionamento. Ipotizziamo che la RSTA voglia inviare N pacchetti di dati ad AP1 ed AP2 mediante due schede di rete WNIC_1 e WNIC_2. I canali di comunicazione sono 2: uno è RSTA-AP1 e l'altro è RSTA-AP2. Per fare in modo di ridurre i disturbi e le mutue interferenze sono stati scelti dei canali WiFi non correlati tra loro, cosi che ciascun AP comunichi separatamente con la RSTA su canali differenti. Nella fase iniziale vengono configurate le schede di rete del RSTA a livello IP (indirizzi IP) ed i parametri di livello applicativo necessari per il funzionamento (a partire dal numero massimo di ritrasmissioni software, numero di pacchetti da inviare, dal parametro sulla modalità operativa e dal parametro sulla modalità di rete). Inoltre vengono inizializzate le strutture dati necessarie, almeno per la prima fase, destinate poi ai vari moduli software. Il modello di concorrenza, mostrato in figura 43, è stato definito in base allo scenario d'utilizzo, nel senso che lo rispecchia, ed ogni entità facente parte del modello svolge un ruolo ben determinato. In particolare sono stati definiti:

Figura 44. Architettura del sistema special purpose preso in considerazione. Naturalmente essa non rappresenta esattamente l'architettura completa, nel senso che vi è rappresentato un solo canale WiFi. Infatti è significativa del funzionamento e della composizione interna dello stack software e del processamento dei pacchetti e dei frame, sia in trasmissione che in ricezione.

- Un'entità, denominata Sender, che rappresenta l'applicativo che effettua trasmissioni e ricezioni di pacchetti applicativi.

- Un'entità, denominata Device Manager, che si occupa di associare le richieste di trasmissione del Sender con il livello OS sottostante.

In questo caso, visto che ci riferiamo a ridondanza duplex, nell'implementazione vengono utilizzate due entità Device Manager. Ad ogni entità si è deciso di associare un thread i quali, rispetto ai processi, condividono l'heap, dunque lo scambio dei dati è meno dispendioso (per i processi bisogna utilizzare strutture di comunicazione come le pipe, i socket o le code di messaggi che si appoggiano su file). Anche se devono essere previste delle strutture di condivisione della memoria che garantiscono l'accesso condiviso, le operazioni di creazione ed eliminazione sono meno pesanti

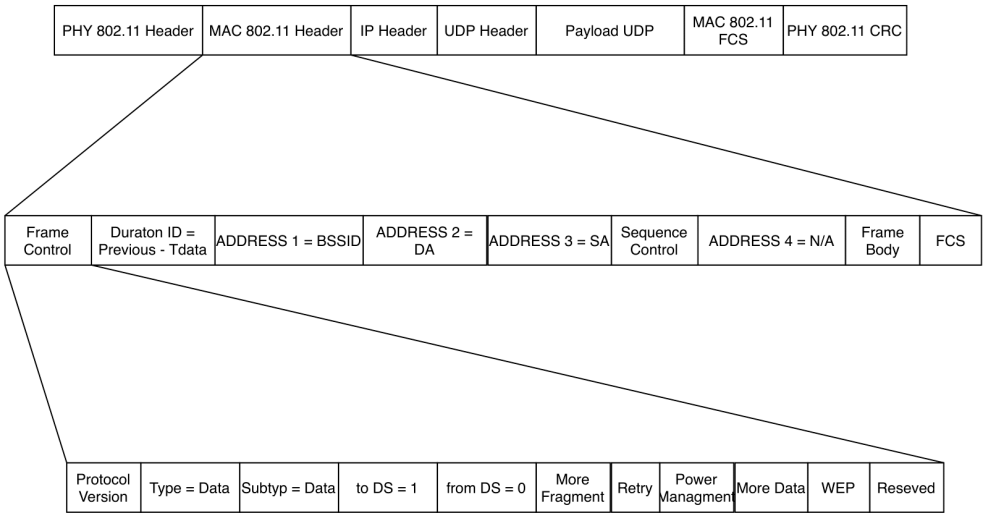

Figura 45. Campi del frame dati, in un modello multilivello in cui viene ispezionato il frame dati nei dettagli, definendone anche alcuni campi significativi.

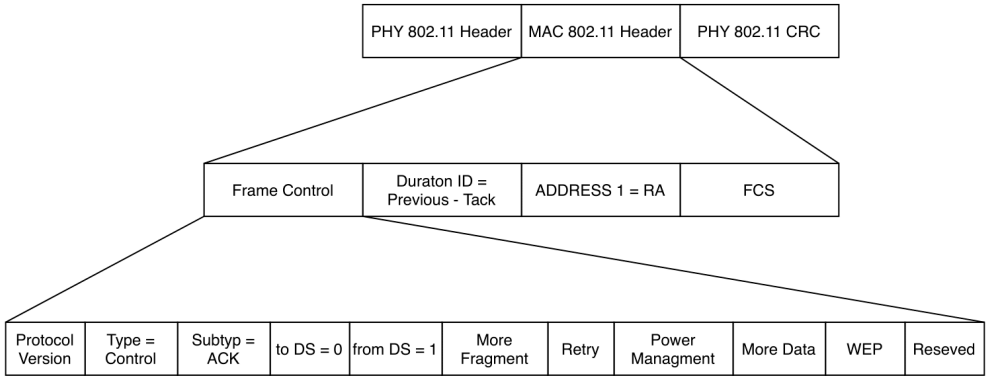

Figura 46. Campi del frame ACK, in un modello multilivello in cui viene ispezionato il frame ACK nei dettagli, definendone anche alcuni campi significativi.

poiché i contesti sono più leggeri (necessitano di meno informazioni) e di conseguenza anche il context-switch è più leggero, inoltre le strutture di sincronizzazione (mutex, semafori), appoggiandosi sulla memoria condivisa, hanno latenze d'accesso minori (le strutture di sincronizzazione dei processi si possono invece appoggiare su file o pipe). Dunque si è deciso di utilizzare i thread poiché l'algoritmo prevede una

forte condivisone delle risorse e necessita di bassa latenza temporale. Ritornando all'analisi del software, ogni thread, che implementa il protocollo applicativo, viene associato una scheda di rete ed una struttura dati. Inoltre per ciascun thread viene definita:

- Una coda contenente gli N pacchetti da inviare.
- Un numero di ritrasmissioni software massime R da effettuare in caso di ACKT.
- Una modalità operativa M, che si riferisce a politiche di condivisione delle code tra processi.

Il protocollo applicativo, come abbiamo già detto, prevede la trasmissione di un certo payload da parte di ogni thread Device Manager a cui segue l'attesa, dello stesso, di una risposta ACK/ACKT. Della gestione dell'affidabilità e cioè della ricezione dei messaggi ACK o ACKT se ne occupa il livello Device Manager. Nel caso di ricezione, da parte del Device Manager, di un ACKT il processo ritrasmette il pacchetto precedente. Le ritrasmissioni vengono interrotte nel caso in cui il pacchetto sia stato trasmesso correttamente (cioè è stato ricevuto un ACK) oppure il pacchetto è già stato ritrasmesso R volte, dove R è un parametro costante che prende il nome di retry limit. Il codice è simmetrico, quindi viene eseguito lo stesso algoritmo su entrambi i Device Manager. L'algoritmo termina quando entrambe le code sono vuote. Le modalità operative M, a cui è stato accennato sopra, cambiano il flusso di esecuzione del codice in base alla specifica modalità scelta. Quando uno dei due Device Manager riceve l'ACK di un pacchetto o raggiunge il massimo numero di ritrasmissioni R consentite per quel pacchetto, il comportamento del codice dipende dalla modalità utilizzata. In particolare, a seguito di una corretta ricezione del pacchetto, in:

- *Strict RDA / Q* il pacchetto viene eliminato da entrambe le code.
- *Flexible RDA / Q* il pacchetto viene eliminato dalla propria coda, ma dall'altra solamente nel caso in cui non sia in trasmissione sull'altro canale.
- *NO RDA / Q* il pacchetto viene eliminato solamente dalla propria coda.

Ovviamente Strict RDA / Q implica una collaborazione più stretta tra i Device Manager, ed una struttura dati più complessa ed adeguata ai vincoli di concorrenza richiesti. In compenso offre prestazioni maggiori in termini di affidabilità e latenza poiché impedisce ad un pacchetto, già confermato su uno dei due canali, di essere trasmesso sull'altro canale. La seconda modalità, Flexible RDA / Q, richiede una struttura dati più complessa rispetto a quella di Strict RDA / Q, nel senso che

richiede una variabile di stato che ciascun Device Manager deve leggere rispettivamente per conoscere lo stato di trasmissione dell'altro canale. Questa struttura più complessa è richiesta al momento dell'eliminazione del pacchetto dalle code (arrivo del frame ACK per quel pacchetto), escluso il caso in qui, per quel pacchetto, siano falliti tutti i tentativi di ritrasmissione possibili (nel suddetto esempio, in ogni modalità operativa, l'eliminazione viene eseguita solamente sulla propria coda). Questo è dovuto al fatto che in Flexible RDA / Q è stata implementata una logica leggermente più elaborata, nel senso che l'eliminazione, da parte di un certo Device Manager, del pacchetto dalla coda dell'altro dipende dallo stato di trasmissione di quest'ultimo. Quindi se, al momento della ricezione dell'ACK (ACKT nel caso in cui si sia raggiunto il limite R di ritrasmissioni) da parte di un certo Device Manager, lo stato di trasmissione dell'altro è uguale al suo, allora l'operazione di eliminazione del pacchetto dalla coda viene eseguita solo sulla propria. L'ultima modalità operativa che è stata implementata è NO RDA / Q (nota anche come Wi-Red) in cui i Device Manager sono indipendenti l'uno dall'altro nel senso che ciascuno di essi si occupa della propria coda separatamente e questo diminuisce la complessità delle operazioni e non necessita di particolari forme di sincronizzazione. Tuttavia vi è una sostanziale differenza tra tutte le modalità fin'ora discusse. Infatti, a parità di affidabilità, il determinismo peggiora progressivamente tra Flexible RDA / Q e NO RDA / Q, poiché nella prima il pacchetto non può essere eliminato contemporaneamente dalle due code se entrambi i processi lo stanno trasmettendo, mentre nella seconda il pacchetto in questione potrebbe essere eliminato dai due processi separatamente nello stesso momento. Flexible RDA / Q si comporta come Strict RDA / Q quando vi è un disallineamento temporale delle trasmissioni tra i due processi (cioè nei due canali sono trasmessi pacchetti afferenti a dati diversi), e come NO RDA / Q se vi è un allineamento temporale stretto tra gli stessi.

Avendo descritto il protocollo applicativo e le modalità operative dell'algoritmo, ora verranno riportati i dettagli implementativi che riguardano il codice e le strutture dati utilizzate, partendo dalla visione strutturale dell'architettura software e descrivendo successivamente ciascuna componente nel dettaglio. L'architettura, come già accennato, è stata realizzata mediante tecniche di programmazione multi-thread, sfruttando la libreria C Posix, secondo il paradigma produttore/consumatore in cui il produttore è il thread che genera i dati applicativi e li inserisce in ciascuna coda (Sender), mentre i consumatori sono rappresentati dai thread che prelevano i dati dalla coda e li trasmettono, mediante il protocollo UDP, al ricevitore (AP). Come già detto, ad ogni thread Device Manager è stata associata una WNIC dedicata. Prima della creazione dei thread operativi (due Device Manager e Sender) il thread principale (Main Thread associato all'intero processo) alloca alcune strutture dati dinamicamente ed esegue l'inizializzazione di tutte le strutture dati necessarie ai thread operativi ed ai vari componenti software che sono stati implementati, inoltre configura a livello IP/UDP le interfacce di rete del sistema. In aggiunta sono state prese scelte implementative adatte per sistemi soft

real-time, ad esempio per garantire maggiori prestazioni a livello di architettura. Infatti, in questa fase iniziale, il Main Thread esegue il lock in memoria ram di tutte le pagine logiche di memoria necessarie al funzionamento del software. Questo viene realizzato mediante l'utilizzo della system call *mlockall ()* che, mediante la configurazione di alcuni flag (*MCL_FUTURE* ed *MCL_CURRENT*), comunica al kernel di bloccare in memoria ram tutte le pagine presenti e quelle richieste dal software a run-time (attraverso la paginazione on-demand), cosi da non subire calo di prestazioni durante l'esecuzione, in termini di latenza temporale, causato dal processo di context-switch delle pagine del kernel (Swap In di una pagina dal disco rigido su memoria volatile; Swap Out di una pagina dalla memoria volatile su disco rigido). Quest'ultima azione potrebbe potenzialmente causare criticità a tutte le applicazioni real-time. Tornando al flusso di esecuzione del software in esame, il Main Thread, creati i thread operativi, attende la conclusione di quest'ultimi. Dopo l'attesa, effettua un dump (scrittura su disco) dei dati catturati durante l'esecuzione dei thread operativi, dealloca le strutture dati allocate dinamicamente, le risorse di rete utilizzate e termina. Tra i parametri di input del software, usati durante l'esperimento, vi sono:

- La modalità di rete da utilizzare (rete reale, rete simulata).
- Il numero di pacchetti da inviare.
- La probabilità di errore del canale in caso di modalità di rete simulata.
- Il numero di ritrasmissioni massime per ogni pacchetto.
- La modalità operativa da utilizzare.

Per quanto riguarda la configurazione delle interfacce wlan0 e wlan1, essa viene effettuata a partire dagli indirizzi IP (che appartengono rispettivamente alle reti BSS dei rispettivi AP) ed il numero di porta UDP. Viene effettuata la chiamata alla system call *socket ()* che inizializza un socket file a partire da alcuni flag come *AF_INET* che configura la tipologia di indirizzo ad IPv4, *SOCK_DRAGM* e la tipologia di protocollo di livello trasporto ad UDP.

Successivamente vengono inizializzati i buffer applicativi, sovra-allocati staticamente nel Main Thread, che conterranno il payload del pacchetto che, durante le fasi di trasmissione dei thread Device Manager, verrà inviato.

Tutti le variabili relative alla parte di rete sono state inglobate in una struttura dati, che rappresenta l'interfaccia di rete, chiamata *interface_t* che contiene:

- Il file descriptor associato al socket file.
- Un puntatore alla struttura *sockaddr_in* che contiene l'indirizzo IP dell'interfaccia.

- Il buffer applicativo.
- La dimensione, in termini di byte, dei dati contenuti all'interno del buffer applicativo.

Conclusa la fase di configurazione e di inizializzazione delle risorse di rete, vengono allocate dinamicamente ed inizializzate, nel Main Thread, le due strutture legate alla cattura dei dati sperimentali (si utilizza il metodo della sovra-allocazione dinamica, in base al numero di pacchetti da trasmettere ed al numero massimo di ritrasmissioni di ciascun pacchetto). Le strutture in questione, chiamate *log_s* e *log_ch*, si riferiscono rispettivamente al canale ridondato ed ai singoli canali. La prima contiene:

- L'id del pacchetto (id).
- L'id del canale (ch) che ha trasmesso correttamente (ha ricevuto l'ACK) per primo.
- Il numero di trasmissioni effettuate per quel pacchetto, da parte del canale (ch) che lo ha trasmesso correttamente per primo (ntx).
- Il timestamp di inserimento in coda di quel pacchetto, da parte del thread Sender (tsc_queue).
- Il timestamp di ricezione dell'ACK per quel pacchetto, da parte del primo canale (ch) che lo ha ricevuto (tsc_sent).

Inoltre, è importante sottolineare che, nel caso in cui un dato pacchetto non sia stato trasmesso correttamente da entrambi i canali, ciascuna struttura *log_s* perde di significato nel senso che conterrà solamente i dati relativi all'immissione in coda, ovvero l'id del pacchetto ed il timestamp di inserimento in coda (*tsc_queue*).

Mentre *log_s*, come già detto, si riferisce ai dati del canale ridondato, la struttura *log_ch* riguarda i dati dei singoli canali. Inoltre si riferisce ad un singolo tentativo di trasmissione (per ogni tentativo di trasmissione riuscito/fallito di un dato pacchetto). In particolare essa contiene:

- L'id del pacchetto da trasmettere (id).
- Il numero di trasmissioni del pacchetto effettuate fino a quel momento (ntx).
- Il timestamp del tentativo di trasmissione del pacchetto (tsc_send).
- Il timestamp di ricezione dell'ACK o del ACKT per quel pacchetto (tsc_ack).
- Un flag binario che indica la cancellazione del relativo pacchetto dalla propria coda (REM1)

- Un flag binario che indica la cancellazione del relativo pacchetto dalla coda concorrente (REM2).

I timestamp dei pacchetti vengono catturati sfruttando un registro della CPU che contiene il numero di colpi di clock dall'accensione del core, comunemente noto come Time Stamping Counter (TSC). Questo rende la misurazione più accurata, in quanto essendo la lettura di un registro non richiede la chiamata di system calls. La funzione di lettura utilizzata dal software è stata implementata mediante codice Assembly. Tutti i campi vengono impostati, dal relativo thread Device Manager o Sender, in base al contesto di cattura dei dati, nel senso che i vari dati che vengono registrati sono stati contestualizzati rispetto al momento della cattura cosi da individuare l'entità che meglio può rappresentare il dato e in modo da diminuire l'incertezza di misurazione.

Inizializzate le strutture di cattura dei dati, viene inizializzata una struttura globale (allocata staticamente), che viene utilizzata per rendere disponibili a certi componenti software del sistema certi dati. Questa struttura è utilizzata sia dal thread Sender che dai thread Device Manager nella fase di startup degli stessi. La struttura, chiamata *config_t* è composta dai seguenti campi:

- La probabilità di errore del canale, utilizzato nel caso di modalità simulata (pe).
- Il numero massimo di ritrasmissioni per ciascun pacchetto (rtx).
- Il numero di pacchetti da inviare (npkts).
- Il flag che indica la modalità simulata (simulation).
- Una stringa costante da inserire nel payload applicativo di ciascun pacchetto (constid).
- La modalità operativa da utilizzare (win_mode).
- Lo stato di trasmissione di ciascun thread Device Manager. Questo è stato realizzato con un vettore di strutture, gestito tramite indici di accesso predeterminati.

Per quanto riguarda i campi dello stato di trasmissione dei thread Device Manager, questi vengono utilizzati solo quando un dato Device Manager deve conoscere lo stato di trasmissione dell'altro per agire sulla sua coda ed eliminare eventualmente un pacchetto confermato sul suo canale. Infatti questi stati di trasmissione vengono scritti e letti solo in modalità operativa Flexible RDA / Q. Non è necessaria una protezione dei dati a livello di concorrenza poiché, dato uno stato di trasmissione appartenente ad un certo Device Manager, esso viene solamente scritto dallo stesso

thread mentre viene letto solamente dall'altro thread Device Manager, ovvero vi è dipendenza write-read, che non soffre il pericolo di race conditions o deadlock.

Inizializzata la struttura *config_t*, il Main Thread procede all'allocazione (dinamica e statica) delle risorse necessarie ai thread operativi ed inizializza le strutture dati necessarie per il passaggio dei parametri operativi. Le risorse che, in questo contesto, vengono allocate dinamicamente dal Main Thread sono le code di messaggi e le strutture di attesa per la protezione delle sezioni critiche. La struttura dati, che realizza le code, è composta dai seguenti campi:

- Un puntatore al primo elemento della lista (head).
- Un puntatore all'ultimo elemento della lista (tail).
- Un contatore degli elementi presenti (size).
- Un flag di notifica per indicare che non saranno più inseriti elementi in coda (last).
- Un mutex per garantire l'accesso atomico al singolo dato (mutex).
- Un semaforo per garantire le politiche di immissione dei pacchetti in coda da parte del produttore in base allo stato della coda stessa (full).
- Un semaforo per garantire politiche di cancellazione dei pacchetti dalla coda da parte del consumatore in base allo stato della coda stessa (empty).

Questa struttura è stata implementata considerando la politica FIFO. La scelta di questa politica è dovuta a ragioni di compatibilità con il protocollo FIFO di prelievo dei pacchetti che usa la scheda di rete wireless per effettuare, in DMA, il prelievo dei messaggi dal ring buffer. Tuttavia sarebbe possibile, in futuro, estendere questo comportamento considerando la possibilità di introdurre un meccanismo a priorità mediate l'utilizzo di strutture dati apposite come gli Heap, tabelle di Hash o comunque attraverso qualunque struttura che abbia complessità di ricerca O(1) e che sia adatta a sostenere vincoli di priorità come questi. Questo permetterebbe di garantire anche certi vincoli di QoS applicativo. Dal punto di vista della memoria si è scelto di implementare una struttura dati linkata non doppiamente, in modo da ridurre le latenze di ricerca ed eliminazione di un dato pacchetto. Infatti l'obiettivo è di cercare di aumentare le prestazioni, in termini di latenza, del meccanismo RDA / Q. Tuttavia un ulteriore sviluppo futuro, che porterebbe alla diminuzione della latenza di ricerca del pacchetto da eliminare, potrebbe essere rappresentato dall'introduzione di una coda più robusta realizzata da una struttura dati doppiamente linkata che viene sfruttata da una funzione di ricerca greedy. Ovviamente questo provocherebbe un aumento, anche minimo (copia di puntatori), sia della latenza di eliminazione dalla coda di un pacchetto individuato che l'aumento della dimensionalità della struttura stessa in termini di byte (aggiunta di un puntatore in più, per

pacchetto). Un'altra idea di miglioramento della latenza di ricerca potrebbe essere quella di utilizzare funzioni di ricerca ricorsive che, con una struttura dati opportuna come un BST, riescono a portare la complessità di ricerca ad O(log(n)), senza causare un aumento significativo della dimensionalità dei dati, rispetto alla struttura doppiamente linkata. Tuttavia, questa scelta potrebbe aumentare la latenza temporale dell'operazione di eliminazione del pacchetto dalla struttura. Molte di queste scelte dipendono sopratutto dalle politiche di prelevamento dei pacchetti, ma entrare nel dettaglio di tali meccanismi esula dagli scopi di questo documento.

Per quanto riguarda l'implementazione dei singoli nodi/elementi della coda, che rappresentano i singoli pacchetti, si è scelto di procedere con cautela, poiché essa peggiora l'overhead in termini sia di latenza temporale che di dimensione, portando rispettivamente a problemi di determinismo e di scalabilità delle operazioni. La struttura del nodo che rappresenta il singolo pacchetto in coda, chiamata *node*, è cosi composta:

- I campi del payload del messaggio applicativo.
- Un puntatore al nodo successivo (next).

Strutturando il nodo in questo modo si ottengono dei notevoli vantaggi. Anzitutto non è presente l'informazione costante per tutti i pacchetti dunque l'overhead, in termini di byte, è ridotto ai minimi termini sia dal punto di vista del singolo nodo (contiene solamente le informazioni variabili per pacchetto del payload applicativo) che dal punto di vista della struttura di memorizzazione dei messaggi (i vincoli strutturali della coda sono stati rispettati mediante un puntatore di pochi byte che linka i dati tra loro). Infatti, l'informazione costante per ogni pacchetto, viene aggiunta al payload applicativo on-the-fly, nel senso che viene effettuata appena prima di inviare il pacchetto applicativo. Inoltre il vantaggio dato da una struttura a coda di messaggi di questo tipo, in termini di overhead dimensionale, diventa più significativo con l'aumento del payload del pacchetto da trasmettere. Si fa notare come un aumento del numero di pacchetti provoca anche un aumento della latenza di ricerca che porta ad un problema di scalabilità, perciò è importante valutare un compromesso in base alle esigenze applicative. Nel caso di questa architettura software, il determinismo è un'esigenza fondamentale a cui è correlata la latenza operativa in maniera direttamente proporzionale.

Tornando al codice ed al flusso di esecuzione del Main Thread, il passo successivo riguarda il passaggio dei parametri ai thread operativi Sender e Device Manager. In tal senso, si è scelto di creare una struttura contenente i dati necessari ai vari thread, in modo da minimizzare sia la dimensionalità che la dimensione in termini di byte della struttura stessa. Si è pensato di strutturare il passaggio dei parametri in maniera gerarchica, considerando una struttura di passaggio dei parametri a livello globale, una struttura di passaggio dei parametri per entità (Sender e Device Manager), ed un meccanismo di indicizzazione delle risorse locali (utilizzate

solamente da un certo thread operativo in base all'algoritmo). Questo porta ad una riduzione sia dell'overhead in termini di byte delle strutture (riutilizzo della stessa informazione da parte di più entità) che della complessità delle strutture dati in quanto tali (modularizzazione delle strutture dati) poiché strutture globali implicano un accesso in concorrenza da gestire (ammesso che si voglia mantenere la stessa struttura in memoria condivisa, per tutte le entità) che porterebbe ad un aumento della latenza media di accesso al dato. Infatti tutti parametri in comune a tutti i thread operativi, necessari durante l'algoritmo, sono stati passati tramite la struttura globale (*config_t*), di cui tutti i thread ne hanno visibilità. Mentre i parametri che riguardano i thread Device Manager vengono passati attraverso un'apposita struttura dati che ingloba sia parametri comuni, che parametri relativi al singolo thread Device Manager. Per indirizzare il singolo thread Device Manager verso le proprie risorse, come già accennato, si è pensato di introdurre un meccanismo di indicizzazione che è stato implementato attraverso indici numerici di indirizzamento delle risorse. Questa scelta di sviluppo è stata fatta nell'ottica futura di un software facilmente comprensibile e modificabile. Sarebbe possibile modificare questo meccanismo tenendo in considerazione lo sviluppo di una struttura dati dedicata ai singoli thread operativi e cercando di tenere basso l'overhead dovuto a ridondanza nei parametri. Le strutture dati che sono state utilizzate per il passaggio dei parametri e l'indicizzazione delle risorse sono principalmente due. La prima, chiamata *data_s*, che si riferisce al thread di tipo Sender è cosi composta:

- Il puntatore alle code di messaggi (queues).
- Il numero di code da considerare (n_queue).

Invece la struttura dati per i thread Device Manager è stata composta nel seguente modo:

- Un puntatore alle code (queues).
- Un indice di indirizzamento della coda (this_queue).
- Un puntatore ad un mutex di sincronizzazione del flusso di esecuzione (mutex).
- Un puntatore all'interfaccia da utilizzare (interface).
- Un identificatore di canale (ch).

I campi che ne fanno parte sono stati scelti, oltre che per motivi di tempi di sviluppo, anche per generalizzare il concetto di ridondanza, eventualmente estendibile ad un modello che prevede la ridondanza multipla e non solo duplex, in modo da facilitare l'implementazione di versioni future dell'architettura software.

Allocate le risorse ed inizializzate le strutture dati necessarie ai thread operativi, il Main Thread avvia, attraverso la funzione Posix *pthread_create ()* della libreria *pthread.h*, i 3 thread operativi con le rispettive strutture dati e successivamente resta in attesa della terminazione di questi mediante la funzione Posix *phtread_join*. Dopo la terminazione dei thread operativi, il Main Thread effettua:

- Un dump, su memoria persistente, dei dati sperimentali raccolti durante l'esecuzione dei thread operativi.
- La deallocazione delle risorse come quelle di rete e quelle di memoria dinamica.

Per quanto riguarda il flusso di esecuzione dei thread operativi, come già accennato, l'entità Sender immette i pacchetti nelle code operative dei Device Manager, quindi svolge un ruolo di produttore. Mentre i Device Manager, l'entità che preleva i pacchetti dalle code operative, quindi svolgono entrambi il ruolo di consumatori. Nel seguito sarà analizzato il flusso di esecuzione congiunto (di Sender e dei Device Manager in sincronia) da quando vengono lanciati i thread operativi nel Main Thread fino a quando non terminano il loro compito.

Procedendo in ordine di tempo, il primo thread ad essere eseguito è il thread Sender. Nella prima parte di codice il Sender copia i dati necessari, passati all'interno della struttura dati data_s sotto forma di parametri e quelli globali della struttura *config_t*, all'interno di variabili e strutture locali, affinché gli accessi (W/R) siano più efficienti in termini di latenza d'accesso. Successivamente esegue l'inserimento in coda dei pacchetti, definendone per ciascuno l'id (un intero incrementale) ed il numero di ritrasmissioni correnti (inizializzato a 0 dal Sender ed incrementato ad ogni ritrasmissione del pacchetto dal Device Manager). Durante l'inserimento dei pacchetti, vengono raccolti dati riguardo al timestamp di inserimento su ciascuno di essi (cosi da poter stimare il tempo di coda). Al termine dell'inserimento dei pacchetti, il Sender termina. Ovviamente il flusso di esecuzione del Sender dipende dal flusso di esecuzione dei thread Device Manager che operano, in linea teorica, in parallelo. Infatti la direttiva di inserimento dei pacchetti in coda, chiamata *add_node ()*, prevede internamente uno meccanismo di attesa che sincronizzi le due tipologie di entità Sender (Produttore) e Device Manager (Consumatore) durante l'esecuzione, cosi da non permettere al Sender di non produrre più pacchetti rispetto alla capacità residua e viceversa, permette a Device Manager di non consumare più pacchetti rispetto all'occupazione della coda.

I Device Manager, che vengono istanziati dal Main Thread successivamente rispetto al Sender, eseguono le medesimi operazioni e si comportano in maniera speculare rispetto alle code del canale concorrente. Così come il Sender, nella prima parte di codice, il Device Manager copia i parametri necessari, passatigli attraverso la struttura dati *data_dm* al momento della creazione dal Main Thread oppure recuperati dalla struttura di accesso globale globale *config_t*, all'interno di variabili e strutture locali, affinché gli accessi (W/R) siano più efficienti in termini di latenza

d'accesso. Questo è dovuto al fatto che accessi ripetuti alla stessa variabile aumentano la frequenza media di accesso alla locazione di memoria e quindi la probabilità che tale variabile sia memorizzata in memoria cache. Per cui, in generale, è molto importante considerare anche i tempi di accesso alle variabili poiché, in contesti di accesso ripetuto questi potrebbero aumentare notevolmente la latenza temporale media complessiva e, specialmente in contesti real-time dove il tempo è prezioso, far peggiorare notevolmente le prestazioni. Tornando al flusso di esecuzione del singolo Device Manager, dopo l'inizializzazione delle variabili in base ai parametri ricevuti, in modalità reale, viene aperto un canale di comunicazione con il driver, attraverso l'apertura di un file di comunicazione intermedio con il device a caratteri. Questo viene gestito tramite la chiamata alla system call *open ()*. Inoltre, attraverso la direttiva *ioctl ()*, viene resettato il device a caratteri stesso, in quanto potrebbe contenere dati dell'esperimento precedente. In modalità simulata questo non viene fatto, poiché la ricezione del messaggio di ACK di conferma dei pacchetti viene simulata da un generatore di numeri casuali, dunque il device a caratteri non è necessario in questo caso. Inoltre non vi è bisogno del suo utilizzo nemmeno per quanto riguarda la simulazione dell'ACK poiché avviene a livello utente. Dopo la fase di configurazione delle risorse (in termini di thread e di strutture dati), vi è una fase operativa. Questa viene realizzata attraverso un'iterazione ripetuta fino a quando il thread Device Manager non termina. Entrando nei dettagli del codice, possiamo elencare i vari passi eseguiti dai thread Device Manager. Questi ultimi, in ordine di tempo d'esecuzione, sono:

1. Il controllo della condizione di terminazione.
2. La selezione di un pacchetto dalla propria coda di pacchetti.
3. La trasmissione del pacchetto, sul canale associato al thread.
4. La ricezione dell'ACK/ACKT.
5. Il controllo della condizione di eliminazione dalle code del pacchetto confermato o meno (in base alla modalità operativa).

La condizione di terminazione del particolare Device Manager è verificata quando non vi sono più pacchetti in coda e termina la fase di immissione dei pacchetti sulla coda specificata da parte del Sender (controllo del flag *last* sulla coda, che viene settato dal Sender quando non ha più pacchetti da immettere). La scelta di una condizione di questo tipo è stata fatta in funzione del comportamento ideale dell'algoritmo e del modello software utilizzato, quindi non ha una vera e propria valenza a livello di prestazioni, ma si tratta di politiche di sviluppo algoritmiche. La condizione di eliminazione dalle code di un dato pacchetto, da parte di un certo Device Manager, dipende sia dal numero di ritrasmissioni del pacchetto effettuate fino a quel momento (rispetto al numero di ritrasmissioni massime possibili per

ciascuno di essi), sia dal tipo di conferma ricevuta, da parte del Device Manager, per quel pacchetto (ACK/ACKT), che dalla modalità operativa scelta che cambia il comportamento dell'algoritmo rispetto alla politica di eliminazione dalle code. La struttura di ricezione dell'ACK/ACKT, come accennato prima, considera due canali di comunicazione per ciascun Device Manager, nel senso che uno di questi è un canale di rete reale mentre l'altro è un canale di rete simulato (è stato implementato a partire dalla rete di Loopback ed una probabilità d'errore statica). Infatti, il flusso di esecuzione del software in modalità reale è differente dal flusso di esecuzione del software in modalità simulata. La ricezione della conferma in modalità reale, utilizza la system call *read ()* per ricevere l'ACK/ACKT da parte del device a caratteri. La fase di trasmissione di un certo pacchetto da parte di uno dei due canali, è stata implementata mediante l'utilizzo della system call *sendto ()* utilizzata per trasmettere pacchetti UDP. Uno degli sviluppi futuri potrebbe essere rappresentato dalla sostituzione della direttiva *sendto ()* con un'altra direttiva, come sendmsg(), che permette di comporre un payload più complesso. Tuttavia, visto che questo prevede lo sviluppo del protocollo applicativo, non è argomento di questa tesi. E' importante precisare che l'implementazione di questa sezione di codice è stata pensata in funzione del massimo grado di parallellismo possibile, nel senso che è stato fatto in modo di rendere le sezioni critiche il più piccole possibili. Ovviamente, la realizzazione del codice è stata resa possibile dalle architetture Seamless Redundancy ed SDMAC precedentemente create, che hanno fornito due canali di comunicazione dedicati con le rispettive schede di rete, cosi da permettere il parallellismo reale delle risorse. Parlando del tipo di pacchetto che viene trasmesso in UDP, possiamo dire che è formato, a livello di payload, da 3 campi che sono:

- L'id su 8 byte (id).
- Una stringa costante su 8 byte (constid).
- L'id del canale che lo sta inviando, su 8 byte (ch).

Questa struttura è necessaria poiché, negli esperimenti in modalità di rete reale, si vuole distinguere la ricezione di un certo pacchetto da parte di un certo canale per effettuare delle analisi sui dati ricevuti dagli AP, e poiché l'indice interno al pacchetto (id) serve per gestire la rimozione dall'altra coda nel caso di ACK. Il campo costante constid è stato scelto per poter distinguere, on-the-fly, il tipo di pacchetto che si vuole ricevere. Questo è stato fatto solo per rendere più facile la creazione del prototipo sperimentale, e cioè per permettere una facile selezione dei pacchetti relativi all'applicazione di ridondanza, rispetto ad altri pacchetti presenti nell'etere. In questo caso, constid ed id vengono inseriti nel pacchetto applicativo per fare in modo che il kernel, alla ricezione dell'ACK corrispondente a quel pacchetto, possa inserire nella struttura SDMAC *rate_data_t* i dati applicativi (campi del payload)

che sono necessari per l'identificazione del pacchetto confermato, sia dal punto di vista del tipo (constid) sia dal punto di vista dell'identificazione univoca. Inoltre il kernel in questione, alla cattura di un ACK/ACKT proveniente dal livello MAC della scheda, inserisce nella struttura *rate_data_t* da inoltrare al livello applicativo, oltre ai dati applicativi anche alcuni dati relativi al livello DLL come l'SNR *ack_signal*, il timestamp di ricezione a livello DLL dell'ACK/ACKT (*tsc_ack* ed altri dati legati al pacchetto inviato, come il suo id, il canale utilizzato, il timestamp della trasmissione effettuata a livello DLL, un flag che indichi ACK/ACKT ed il rate di trasmissione utilizzato per quel pacchetto.

Eseguendo un'analisi di alto livello all'algoritmo, sopratutto riferita al suo comportamento a run-time, possiamo dire che i Device Manager si occupano di inviare i pacchetti e di ricevere un ACK/ACKT, in modo da sfruttare le informazioni ottenute per ridurre il consumo di banda complessiva e cercando di evitare l'eventuale trasmissione, sull'altro canale, di un pacchetto confermato sul suo. Inoltre, come detto in precedenza, il comportamento dell'algoritmo può essere predeterminato in base al parametro di input che riguarda la modalità di funzionamento voluta (Strict RDA / Q, Flexible RDA / Q, NO RDA / Q). In ogni caso, il comportamento di un Device Manager rispetto alla coda concorrente, alla ricezione di un ACK/ACKT, può essere di 3 tipi:

- Se viene ricevuto un ACKT ed il pacchetto non eccede il numero massimo di ritrasmissioni, si riparte con la selezione del prossimo pacchetto da trasmettere (che sarà lo stesso, se non è stato cancellato dal Device Manager concorrente).

- Se viene ricevuto un ACKT ed il pacchetto ha raggiunto il numero massimo di ritrasmissioni, viene eliminato solamente dalla propria coda (questo in tutte le modalità operative).

- Invece, se viene ricevuto un ACK, il comportamento dipende dalla modalità. In Strict RDA / Q viene cancellato il pacchetto da entrambe le code, in Flexible RDA / Q viene cancellato dalla propria coda e dall'altra solamente se il pacchetto in questione non è in trasmissione, mentre in NO RDA / Q viene cancellato solamente dalla propria coda, indipendentemente dallo stato del Device Manager concorrente.

Tuttavia, indipendentemente dalla modalità operativa e dal tipo di conferma ricevuta (ACK/ACKT), vengono raccolti i dati relativi alla ricezione ed in particolare, il timestamp applicativo di ricezione della conferma riferito al pacchetto. Anche in fase di cancellazione di un pacchetto dalle code, l'operazione viene registrata e vengono raccolti dati sia sul tipo di conferma (ACK/ACKT) eventualmente ricevuta sul canale ridondato che sull'entità che cancella un pacchetto da una delle due code (da quale coda il pacchetto è stato cancellato, il canale che effettua la

cancellazione). Il Device Manager termina quando la propria coda è vuota e non ci sono più pacchetti da inserire in essa da parte del Sender.

L'algoritmo che viene eseguito sui thread Device Manager, prevede la protezione di codice e precisamente contiene due sezioni critiche eseguite da ciascun thread in mutua esclusione. Il mutex di protezione del codice è stato allocato dinamicamente ed inizializzato dal Main Thread ed è condiviso tra i Device Manager in modo che vedano lo stesso mutex (non è possibile sfruttare le proprietà di sincronizzazione se vengono viste le copie dei mutex). In particolare:

- La prima sezione critica inizia nel momento della selezione di un nuovo pacchetto da inviare. Prima della selezione viene chiamata la system call *pthread_mutex_lock ()* che riceve il mutex e lo blocca. Il thread viene rilasciato, tramite la system call *pthread_mutex_unlock*, nel caso in cui la funzione *select ()* non torni un valore valido (*NULL*), oppure dopo aver selezionato il pacchetto, catturato il timestamp di selezione, aggiornato i tentativi di trasmissione per quel pacchetto e riempito il buffer applicativo con il payload da trasmettere tramite *interface_prepare_pkt*.

- La seconda sezione critica inizia dopo la ricezione dell'ACK/ACKT. Dopo aver catturato l'ACK/ACKT, viene chiamata nuovamente la system call *pthread_mutex_lock ()* sul mutex per bloccarlo. Viene catturato il timestamp di arrivo dell'ACK/ACKT ed in base al valore ricevuto (ACK/ACKT), alla modalità operativa (Strict RDA / Q, Flexible RDA / Q , NO RDA / Q) ed al numero di ritrasmissioni effettuate per quel pacchetto fino a quel momento, verrà eliminato il pacchetto da una/entrambe/nessuna delle code. Successivamente viene chiamata la system call *pthread_mutex_unlock* per sbloccare il mutex (consentendo all'altro thread di eseguire la stessa sezione critica) ed iterare nuovamente, selezionando nuovamente un pacchetto dalla propria coda.

Inoltre va ribadito il fatto che per proteggere le due sezioni critiche è stato utilizzato lo stesso mutex poiché in realtà esse devono essere eseguite in mutua esclusione (ogni Device Manager non può eseguire una delle due sezioni critiche, se prima il Device Manager concorrente non esce da entrambe le sezioni critiche ed abilita il mutex). Dunque è possibile vedere queste due sezioni critiche come un'unica sezione critica, protetta dallo stesso mutex, a differenza del fatto che le due sotto-sezioni sono distribuite nel codice in due punti diversi. La scelta di prevedere un solo mutex, è stata pensata in ottica della dimensione e della latenza temporale del codice e del consumo delle risorse (se avessi avuto due mutex avrei acceduto a due locazioni di memoria condivisa differenti e ciò potrebbe portare ad un aumento di complessità del codice e delle strutture dati, nonché ad un aumento della latenza media di attesa dell'abilitazione, poiché l'aumento della memoria condivisa, di solito, porta ad una diminuzione del grado di parallelismo oltre che un aumento

delle risorse di sincronizzazione). Tornando alle sezioni critiche e parlando dei tipi di protezione, possiamo scendere nel dettaglio ed analizzare lo pseudo codice sottostante. Queste due porzioni di codice fanno si che ciascun Device Manager non selezioni un pacchetto già confermato dal Device Manager concorrente (infatti è arrivato l'ACK per quel pacchetto) e che sarà eliminato subito dopo.

DM-SECTION-1($my\_queue, mutex, interface, channel$)

```
1  LOCK(mutex)
2  if SELECT(my_queue, &id, &try) == 0
3        UNLOCK(mutex)
4        CONTINUE
5  PREPARE_PACKET(id, channel, const, interface)
6  my_curr = id
7  timestamp = READTSC
8  UNLOCK(mutex)
```

DM-SECTION-2($my\_queue, other\_queue, mutex, id, try, ack, maxtry, mode, other\_curr$)

```
 1  LOCK(mutex)
 2  if ack == 1
 3        if mode == 0
 4              REMOVE(my_queue, id)
 5              REMOVE(other_queue, id)
 6        elseif mode == 1
 7              REMOVE(my_queue, id)
 8              if other_curr ≠ id
 9                    REMOVE(other_queue, id)
10        elseif mode == 2
11              REMOVE(my_queue, id)
12  elseif try == maxtry
13        REMOVE(my_queue, id)
14  UNLOCK(mutex)
```

La sezione di codice rimanente non necessita di protezione poiché non prevede la condivisione di risorse e/o la comunicazione tra thread. Infatti è composta solamente dalla parte di trasmissione e ricezione dei pacchetti e dei rispettivi ACK/ACKT e, visto che il canale di rete è duplicato (ci sono 2 schede di rete, anche se il driver ed i moduli del OS sono unici) il grado di parallelismo aumenta. Senza dubbio l'operazione di trasmissione dei dati e la successiva attesa del ACK/ACKT è la più costosa in termini di tempo quindi la struttura di sincronizzazione peggiora le prestazioni solamente per la parte meno costosa (l'elaborazione delle code e la selezione del pacchetto dalla coda), tuttavia le prestazioni, analizzate nel prossimo capitolo sull'analisi dei dati, sono riguardevoli.

Parliamo infine della modalità simulativa. Anzitutto tra i parametri passati da riga di comando, come già accennato, vi è un flag che indica se il programma andrà in esecuzione in simulazione o meno. Per valori nulli del parametro, il software sarà eseguito in modalità reale, mentre per valori non nulli del parametro il software sarà eseguito in simulazione. Questa consiste nel inviare i pacchetti tramite due interfacce di Loopback, mentre l'ACK/ACKT viene generato dal livello utente tramite un generatore di numeri casuali e non più ricevuto dal device a caratteri come nella modalità reale. Inoltre la generazione degli ACK/ACKT è controllata da un altro parametro di input del software (pe) che rappresenta la probabilità di errore del canale simulato (è un numero in virgola mobile tra 0 ed 1). Le interfacce vengono inizializzate con indirizzi di Loopback (127.X.X.X) e non privati. Naturalmente in modalità simulata le attese e gli invii dei pacchetti sono ridotte rispetto alla modalità reale, quindi il codice prevede delle *sleep ()* che simulano l'attesa reale (anche se le latenze di accesso al canale sarebbero, in media, fisse mentre l'attesa reale è una variabile casuale e quindi può variare) in modo da modellare e simulare il canale fisico. Nella modalità simulativa, si possono anche utilizzare le diverse modalità operative, poiché non dipendono dalla configurazione di rete bensì dalla gestione dei pacchetti sulle code. Nel prossimo capitolo, l'analisi dei dati sperimentali, vedremo come si comporta l'algoritmo in risposta ad una serie di esperimenti. Verrà analizzata più nel dettaglio la natura del RSTA (Architettura reale utilizzata) e degli AP. Inoltre, grazie ai grafici ed alle tabelle di raccolta dei dati sperimentali, sarà possibile valutare con cura le prestazioni sui singoli canali e sul canale ridondato.

Analisi dei risultati

In questo capitolo saranno analizzati i risultati sperimentali ottenuti dall'elaborazione dei dati raccolti dal software specificatamente realizzato per la comparazione delle tecniche descritte nei precedenti capitoli. Gli esperimenti sono stati eseguiti in tutte le modalità implementate (Strict, Flexible, e No RDA/Q) su rete reale. La modalità simulativa è stata invece utilizzata per il debug del software realizzato. Inoltre, per una corretta comparazione delle modalità operative di esecuzione del software, si è cercato di minimizzare le variazioni delle condizioni sperimentali al contorno, soprattutto per quanto riguarda la probabilità media di errore dei canali WiFi utilizzati per la comunicazione end-to-end tra RSTA ed APs che, nel tempo, può variare notevolmente. Questo rende per definizione la probabilità di errore non stazionaria (probabilità non costante nel tempo). Infatti, le condizioni del canale cambiano di continuo, dunque è fondamentale eseguire gli esperimenti consecutivamente nel tempo, proprio per ridurre questo fenomeno non deterministico. Le elaborazioni post-esecuzione sono state effettuate da un analizzatore software creato ad-hoc ed implementato in Python il quale, mediante funzioni statistiche, rielabora i dati grezzi raccolti dal software eseguito per ottenere i dati sperimentali.

Gli indici prestazionali di qualità sono stati selezionati in funzione delle caratteristiche peculiari che le applicazioni soft real-time generalmente richiedono. Essi riguardano indici di determinismo ed affidabilità che, come ampiamente discusso in precedenza, costituiscono due proprietà fondamentali per rispettare le specifiche di comunicazione industriali di tipo soft real-time. Il determinismo rappresenta una proprietà strettamente necessaria, anche se ideale nelle reti best-effort, per raggiungere prestazioni accettabili in termini delle tempistiche real-time richieste. Inoltre l'affidabilità rende possibile l'utilizzo della tecnologia WiFi ad una vasta gamma di applicazioni che la richiedono. Quindi, determinismo ed affidabilità sono essenziali per servizi real-time in ambito industriale. Dunque, proprio per questo, sono state valutate queste due proprietà attraverso indici di qualità che vedremo nel dettaglio in seguito. Infatti, attraverso essi, è possibile valutare in maniera abbastanza precisa le caratteristiche qualitative e quantitative in termini di deadline non rispettate (operazione che è effettuata in termini probabilistici nel caso di sistemi soft real-time). Naturalmente, le valutazioni in ambito probabilistico, effettuate sotto certe condizioni sperimentali e simulative nell'ambito delle reti industriali,

costituiscono una indice importante sull'applicabilità della tecnologia proposta al campo industriale.

Il meccanismo di Reactive Duplication Avoidance on Queue (RDA/Q), sviluppato in questa tesi sperimentale, rappresenta un modo efficace di contrastare il consumo di banda dovuto alla trasmissione di pacchetti già confermati su uno dei due canali. L'architettura Seamless Redundancy infatti, già ampiamente discussa in precedenti capitoli, necessita di un apposito meccanismo di DA. Questo si occupa della gestione delle conferme multicanale e della cancellazione dalle code dei pacchetti che sono già stati confermati su uno dei canali consistenti la ridondanza. In particolare, il meccanismo RDA/Q implementato, agisce a livello applicativo e si occupa di cancellare dalle code dei thread Device Manager di tali pacchetti già confermati su uno dei due canali. Così RDA/Q costituisce un meccanismo in grado di ridurre drasticamente il consumo di banda. Poiché si tratta di una funzionalità di tipo reattiva, esso viene innescato automaticamente dai thread operativi a run-time. Grazie a questa caratteristica, RDA/Q, è più che idoneo a supportare applicazioni soft real-time su reti industriali di tipo WiFi, in cui le interferenze e le condizioni del canale che spesso variano, influenzano le prestazioni in termini di affidabilità e determinismo. Valuteremo in seguito, nel dettaglio, le prestazioni delle modalità operative RDA/Q, attraverso considerazioni su dati sperimentali concreti.

Per quanto riguarda il contesto di rete, gli esperimenti sono stati effettuati tutti su rete reale-sperimentale. In particolare, come mostrato in figura 47, si tratta di una rete mista (realizzata per simulare una rete di tipo industriale) in cui l'infrastruttura di rete principale, è stata realizzata in IP over Switched-Ethernet (IPoSE), mentre l'infrastruttura secondaria è stata realizzata mediante collegamenti wireless attraverso il protocollo WiFi IEEE 802.11. Nella configurazione sperimentale i dispositivi wireless sono stati configurati, sebbene supportino lo standard IEEE 802.11n, per aderire allo standard IEEE 802.11g, disabilitando le seguenti caratteristiche protocollari:

- il meccanismo di Frame Aggregation.
- la funzionalità di Block Acknowledgement.

Il meccanismo di ricezione per quanto riguarda la rete wireless, come descritto nel capitolo precedente, è stato realizzato mediante l'interposizione tra il modulo MAC dell'SO ed il layer applicativo, di un device a caratteri che si occupa della propagazione dell'ACK/ACKT tra essi.

L'analisi dei risultati sperimentali, dunque, partirà da una breve introduzione sugli indici statistici che saranno presi in considerazione, e poi il discorso si concentrerà sulle singole tecniche e sull'analisi dettagliata dei risultati ottenuti. Le

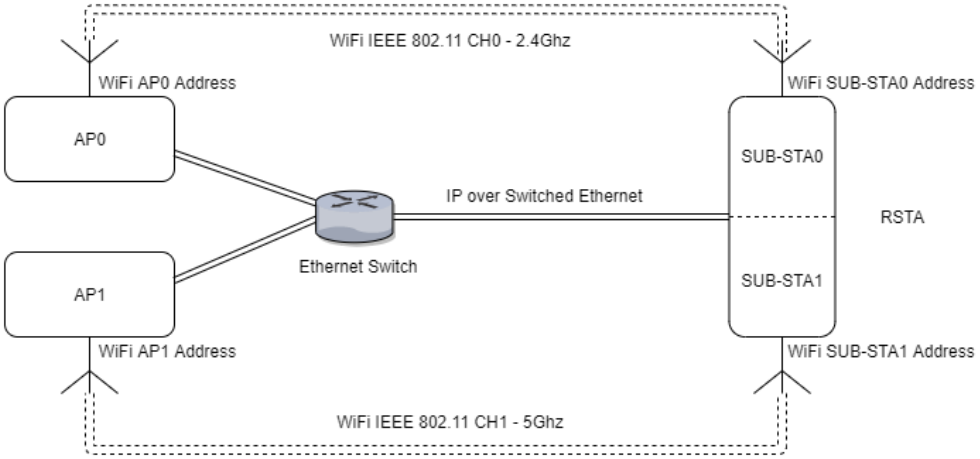

Figura 47. Architettura di rete reale utilizzata per simulare la rete industriale. Essa non rispetta l'architettura di rete industriale, ma dal punto di vista del funzionamento e delle prestazioni è idonea ad essere usata per effettuare esperimenti reali. Funzionalmente, come si può vedere, è composta da sistemi reali, che si utilizzano in ambito industriale per implementare l'infrastruttura di una rete.

considerazioni su affidabilità e determinismo si baseranno sui dati sperimentali stessi. A livello sperimentale saranno usati PC commerciali basati su architetture open source Linux.

I risultati degli esperimenti svolti su codesta architettura hardware/software, sono organizzati nel modo seguente:

- Due tabelle rappresentati i dati sperimentali raccolti, in modalità reale di rete, in cui i tre esperimenti svolti in tutte le modalità operative sono stati realizzati uno di seguito all'altro. Questo sequenzialità è stata applicata al modello di sperimentazione per ridurre il più possibile le variazioni dello stato del canale tra un esperimento ed il successivo.

- Un grafico rappresentante la funzione Cumulative Density Function (CDF). Come da definizione, essa mostra essenzialmente la funzione cumulativa di densità di probabilità statistica della variabile casuale ideata per valutare la latenza.

I risultati sono stati tutti comparati con la modalità NO RDA/Q (altresì nota come Wi-Red) che nel contesto di questa campagna sperimentale rappresenta la base di riferimento. NO RDA/Q, come si evincerà dai risultati sperimentali, fornisce ottimi miglioramenti prestazionali se paragonata ai singoli canali. La modalità Flexible RDA/Q è invece un implementazione più semplice ma meno efficace in diversi contesti applicativi se paragonata alla tecnica Strict RDA/Q. Per quanto

riguarda i dati sperimentali essi sono riportati in Tabella 1 e Tabella 2. I risultati riportati in queste tabelle sono stati ottenuti spedendo 200000 pacchetti, con un periodo di generazione pari a 10 ms. Gli esperimenti presi in considerazione nel seguito sono stati effettuati in modalità di rete reale e senza interferenti, in modo da valutare le prestazioni senza variazioni significative del canale.

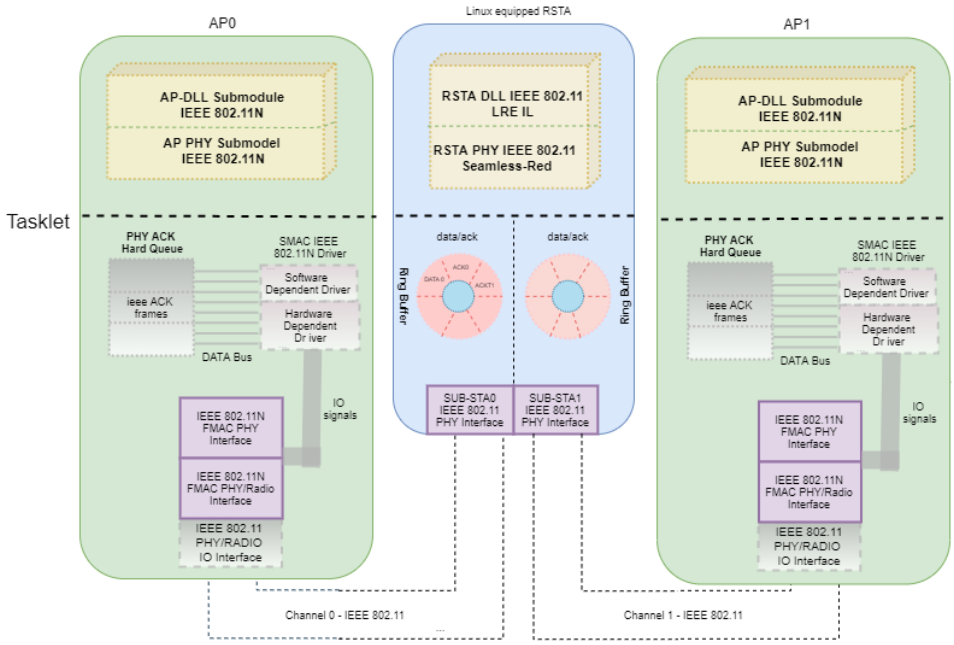

Figura 48. Architettura HW/SW della rete del prototipo sperimentale. Essa rappresenta anche l'architettura dei singoli sistemi che ne fanno parte, come la RSTA e gli AP. Inoltre, oltre a rappresentare nel dettaglio collegamenti tra moduli ibridi HW/SW definiti, mostra anche le modalità di scambio a livello di pila ISO-OSI dei sistemi che ne fanno parte.

In particolare, in Tabella 1 sono riportati i seguenti dati statistici:

- min, che rappresenta un Lower Bound (Limite inferiore) di latenza.
- max, che rappresenta un Upper Bound (Limite superiore) di latenza.
- μ, che rappresenta la latenza media.
- σ, che rappresenta la deviazione standard della latenza.
- *lost*, che rappresenta la quantità di pacchetti persi, rispetto al totale trasmessi.

I risultati sperimentali mostrano chiaramente come la quasi totalità degli indici prestazionali relativi a Strict RDA/Q siano migliori rispetto a quelli delle altre

Mode	***Channel***	*Min*	μ	σ	*Max*	*Perc lost*
		ms	ms	ms	ms	
Strict RDA/Q	0	0.48	4.76	4.74	50.79	0.04
	1	6.12	8.87	6.12	40.67	0.002
	0+1	0.17	0.6	0.54	40.6	0
Flexible RDA/Q	0	0.44	5.16	8.51	1016.3	0.64
	1	0.17	7.35	7.47	212.95	0.804
	0+1	0.17	1.2	4.13	874.07	0.0002
No RDA/Q	0	0.41	44.52	193.6	2251.33	3.383
	1	0.17	19.07	53.58	1339.39	1.008
	0+1	0.17	2.94	22.03	2004.92	0.0004

Tabella 1. Risultati sperimentali (tipici di sistemi soft real-time)

Mode	*δ<1ms*	*δ<2.5ms*	*δ<5ms*	*99 Perc*	*99.9 Perc*
				ms	ms
Strict RDA/Q	88.35	98.71	99.9	2.74	5.09
Flexible RDA/Q	60.33	90.9	98.54	5.83	20.31
No RDA/Q	38.83	76.06	92.49	216	173.6

Tabella 2. Risultati sperimentali

due tecniche. Il miglioramento notevole degli indici statistici relativi alla latenza, specialmente quelli maggiormente legati al determinismo come deviazione standard e massimo, è dovuto principalmente al numero minore di elementi presenti nelle code dei due canali su cui sono state implementate le tecniche di ridondanza. In particolare, nel caso di Strict RDA/Q, appena una conferma di ricezione (ACK) arriva su uno dei due canali soggetti a ridondanza, un evento di XACK permette la quasi istantanea rimozione del frame dall'altro canale, facendo in modo di avere, per questa tecnologia, code di trasmissione più vuote. In No RDA/Q, invece, non vi è nessuna rimozione degli elementi in coda, invece in Flexible RDA/Q l'eliminazione non è ottimale, in quanto occorre aspettare che tutte le ripetizioni di un singolo pacchetto siano eseguite, prima di procedere all'eventuale eliminazione. I risultati ottenuti sono molto migliorativi. Si fa notare come la latenza massima si sia ridotta da circa 2 secondi a 40 ms. Relativamente alla percentuale di perdite, il valore ottenuto è molto basso ed adatto a molte applicazioni di tipo industriale.

Per quanto riguarda Tabella 2, gli indici statistici in essa contenuti sono i seguenti:

- $\delta < \beta$ *ms*, dove δ rappresenta la percentuale di pacchetti confermati sul canale ridondato (0+1), con latenza non superiore a β millisecondi.

- Percentile K *ms*, indica la percentuale K di pacchetti confermati sul canale ridondato con latenza inferiore al valore del percentile stesso.

Anche nel caso di Tabella 2 si può notare come le prestazioni di Strict RDA/Q siano nettamente superiori a quelle delle altre due tecniche.

Tali conclusioni sono confermate analizzando i dati riportati nella CDF in figura 48.

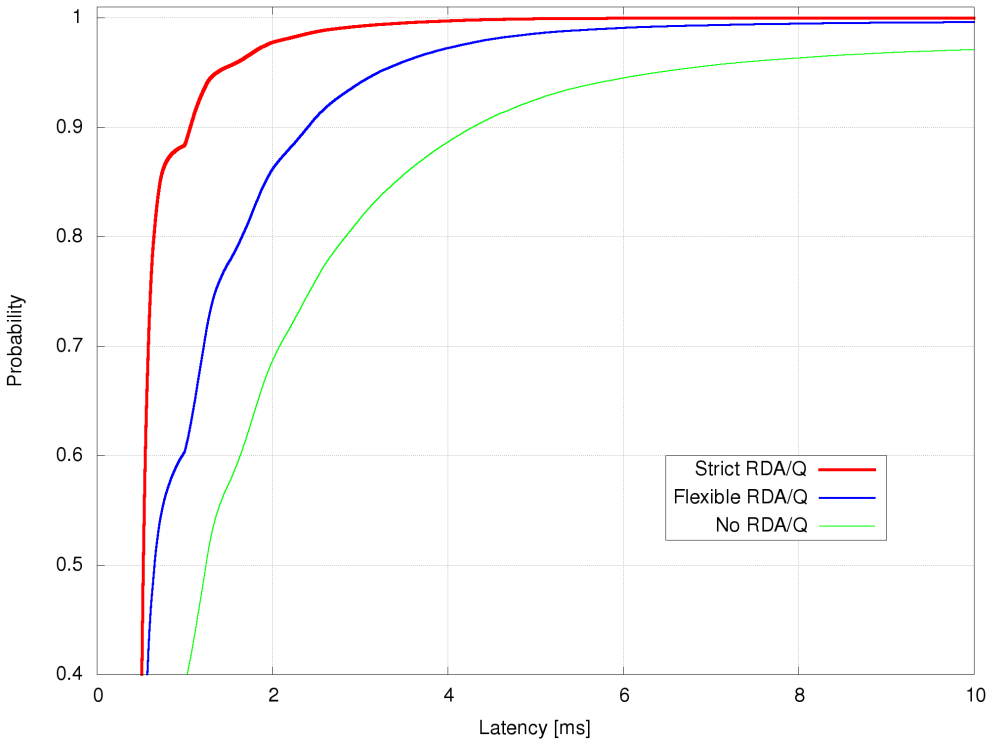

Figura 49. Funzione cumulativa di densità di probabilità del canale ridondato, dove la variabile aleatoria che rappresenta la latenza di una certa quantità di pacchetti, è rappresentata sull'asse delle ascisse. Mentre, l'asse delle ordinate rappresenta la probabilità di avere una latenza per percentuale di pacchetti inferiore ad una certa soglia di incertezza assolutistica.

In definitiva, si è potuto validare sperimentalmente l'efficacia di Strict RDA/Q, e si sono potuti osservare:

- notevoli miglioramenti in termini di affidabilità dovuti all'applicazione di tecniche di ridondanza

- sostanziali miglioramenti in termini di latenza rispetto alle altre tecniche prese in considerazione, così come si può evincere dalla CDF di figura 48

- un minor spreco di banda dovuto al fatto che Strict RDA/Q blocca la trasmissione di duplicati quando un frame è arrivato sull'altro canale
- una minor occupazione delle code di trasmissione interne ai nodi.

Si fa notare come l'applicazione di tecniche di ridondanza di tipo RDA/Q, cioè sia Strict che Flexible, si sempre migliorativa se non sono considerati i tempi di software dovuti alla gestione delle trasmissioni/ritrasmissioni e delle code a livello applicativo. Tali tempi sono comunque sufficientemente piccoli in architetture veloci come quelle attuali, e praticamente annullabili in implementazioni hardware appositamente realizzate per gestire queste tecniche di ridondanza. Infatti, quello che fanno queste tipologie di tecniche è la rimozione dalle code di trasmissione di quei pacchetti la cui trasmissione non ha più nessuna utilità in quanto sono già arrivati su un altro canale. La minor occupazione media delle code ha come effetto diretto una minor latenza di trasmissione, e la mancata trasmissione di tali frame comporta un minor spreco del canale wireless. I risultati qui riportati evidenziano in modo chiaro l'efficacia di queste tecniche in dispositivi reali.

Conclusioni

L'implementazione su sistemi reali di tecniche di ridondanza e la successiva verifica sperimentale ha permesso di validare in un ambiente con disturbi ed interferenti reali l'utilizzo di tecniche di ridondanza di tipo RDA/Q.

Sono state confrontate due implementazioni di tipo RDA/Q (Strict e Flexible) e comparate con una tecnica classica di ridondanza WiFi denominata Wi-Red. L'analisi sperimentale, sebbene effettuata su un numero di condizioni sperimentali ridotto, ha permesso senza ombra di dubbio di evidenziare gli effetti positivi in termini di latenza e di affidabilità delle tecniche basate su RDA/Q, ed in particolare la tecnica Strict RDA/Q ha evidenziato risultati migliori sia rispetto a Wi-Red che rispetto a Flexible RDA/Q. Le tecniche di tipo RDA/Q permettono infatti l'eliminazione dalla coda di trasmissione associata ad un'interfaccia di rete di alcuni dei dati che sono arrivati al nodo destinazione attraverso l'altra interfaccia. Questo si traduce in code mediamente più vuote, e conseguentemente in una maggior velocità di servizio per i successivi frame che saranno in esse accodati. In aggiunta, le tecniche RDA/Q (sia Strict che Flexible), oltre a consentire un miglioramento delle prestazioni, hanno permesso di migliorare notevolmente la banda risparmiata, andando a non trasmettere parte dei dati duplicati che sarebbero stati trasmessi da tecniche di ridondanza classiche come Wi-Red.

L'utilizzo di tecniche di tipo RDA/Q permette perciò di incrementare i contesti applicativi in ambito industriale in cui le reti wireless possono essere utilizzate. Infatti, oltre all'utilizzo di WiFi in applicazioni di tipo "office", caratterizzate dall'assenza di requisiti temporali, con tecniche di tipo RDA/Q si riesce ad ottenere il risultato che, nelle condizioni sperimentali analizzate in questa tesi, il 99.9% dei pacchetti è trasmesso con latenza inferiore a 5 ms. Questo dato, molto significativo e decisamente migliore rispetto al WiFi tradizionale, è adatto per molte applicazioni di tipo soft real-time industriale.

Gli sviluppi futuri di questa tesi includono una più estesa campagna sperimentale finalizzata alla valutazione di RDA/Q per un maggior numero di condizioni operative in termini di disturbo e di interferenza.

Bibliografia

[1] L. Xiaomin, L. Di, W. Jiafu, V. V. Athanasios, L. Chin-Feng and W. Shiyong, *"A review of industrial wireless networks in the context of Industry 4.0"*, Wireless Networks, vol. 23, pp. 23–41, 2017. URL: https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5b7e16b15&appId=PPGMS

[2] F. Sartori, *"Progetto di un codificatore e decodificatore digitale per lo standard IEEE 802.11 realizzato mediante FPGA"*, 2002. URL: http://fsartori.altervista.org/capitolo_1.pdf

[3] Wikipedia, *"IEEE 802.11a-1999"*, Ottobre 2019. URL: https://en.wikipedia.org/wiki/IEEE_802.11a-1999

[4] Wikipedia, *"IEEE 802.11b-1999"*, Dicembre 2019. URL: https://en.wikipedia.org/wiki/IEEE_802.11b-1999

[5] C. Naso, *"Evoluzione dello standard IEEE 802.11 (WiFi): verso IEEE 802.11ac"*, Dicembre 2012. URL: http://tesi.cab.unipd.it/42079/1/NasoCaterina578399.pdf

[6] E. Khorov, A. Kiryanov, A. Lyakhov and G. Bianchi, *"A Tutorial on IEEE 802.11ax High Efficiency WLANs"* in IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 197-216, 2019. URL/DOI: https://doi.org/10.1109/COMST.2018.2871099

[7] D. Lopez-Perez, A. Garcia-Rodriguez, L. Galati-Giordano, M. Kasslin and K. Doppler, *"IEEE 802.11be — Extremely High Throughput: The Next Generation of Wi-Fi Technology Beyond 802.11ax"*, in IEEE Communications Magazine, vol. 57, no. 9, pp. 113-119, 2019. URL/DOI: https://doi.org/10.1109/MCOM.001.1900338

[8] AirCrack-ng, *"Linux, Open Source drivers"*, 2020. URL: https://www.aircrack-ng.org/doku.php?id=install_drivers

[9] G. Cena, S. Scanzio and A. Valenzano, *"A Prototype Implementation of Wi-Fi Seamless Redundancy with Reactive Duplication Avoidance"* in IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA 2018), Turin, 2018, pp. 179-186. URL/DOI: https://doi.org/10.1109/ETFA.2018.8502636

[10] M. Vipin and S. Srikanth, *"Analysis of open source drivers for IEEE 802.11 WLANs"* in 2010 International Conference on Wireless Communication and Sensor Computing, pp. 1-5, 2010. URL/DOI: https://doi.org/10.1109/ICWCSC.2010.5415877

[11] G. Cena, S. Scanzio and A. Valenzano, *"Seamless Link-Level Redundancy to Improve Reliability of Industrial Wi-Fi Networks"* in IEEE Transactions on Industrial Informatics, vol. 12, no. 2, pp. 608-620, 2016. URL/DOI: https://doi.org/10.1109/TII.2016.2522768

[12] G. Cena, S. Scanzio and A. Valenzano, *"Experimental Evaluation of Seamless Redundancy Applied to Industrial Wi-Fi Networks"* in IEEE Transactions on Industrial Informatics, vol. 13, no. 2, pp. 856-865, 2017. URL/DOI: https://doi.org/10.1109/TII.2016.2641469

[13] G. Cena, S. Scanzio and A. Valenzano, *"Improving Effectiveness of Seamless Redundancy in Real Industrial Wi-Fi Networks"* in IEEE Transactions on Industrial Informatics, vol. 14, no. 5, pp. 2095-2107, 2018. URL/DOI: https://doi.org/10.1109/TII.2017.2759788

[14] G. Cena, S. Scanzio and A. Valenzano, *"SDMAC: A Software-Defined MAC for Wi-Fi to Ease Implementation of Soft Real-Time Applications"* in IEEE Transactions on Industrial Informatics, vol. 15, no. 6, pp. 3143-3154, 2019. URL/DOI: https://doi.org/10.1109/TII.2018.2873205

[15] G. Cena, S. Scanzio, A. Valenzano and C. Zunino, *"Implementation and Evaluation of the Reference Broadcast Infrastructure Synchronization Protocol"* in IEEE Transactions on Industrial Informatics, vol. 11, no. 3, pp. 801-811, 2015. URL/DOI: https://doi.org/10.1109/TII.2015.2396003