

8

Protocolli anticollisione basati su alberi per tag RFID

Petar Popovski

Dipartimento di Sistemi Elettronici, Università di Aalborg

8.1 Introduzione

I sistemi di identificazione a radiofrequenza (RFID) stanno digitalizzando sempre più e in modo ubiquitario l'ambiente fisico attaccando minuscoli tag a oggetti e persone [1]. Si prevede che supportino un grande volume/diversità di applicazioni, che vanno dalla logistica, ai trasporti, ai servizi di vendita al dettaglio, al controllo degli accessi, nonché a molte applicazioni ancora da immaginare appartenenti al paradigma "Internet delle cose" [2].

I due componenti chiave di un sistema RFID sono tag e lettori. Un tag è un piccolo microchip dotato di antenna che viene fissato all'oggetto o alla persona. Un tag può essere passivo (che non utilizza una batteria), semi-attivo (che utilizza una batteria per il rilevamento/elaborazione, ma non per la comunicazione) o attivo (tag alimentato a batteria con un consumo energetico molto basso). I lettori, noti anche come interrogatori, sono ricetrasmittitori che possono comunicare con i tag, leggendo informazioni dai tag o scrivendo informazioni su di essi.

Sebbene una varietà di sistemi RFID sia stata utilizzata per un periodo considerevole [3], negli ultimi anni la tecnologia RFID ha mostrato un'enorme crescita [4]. Ci sono diverse ragioni per tale crescita. Tag e lettori stanno diventando meno costosi, quindi il loro numero e la loro ubiquità aumentano. Chiaramente, i tag poco costosi saranno presenti in numero molto maggiore rispetto ai lettori. In particolare, grazie alla maggiore portata di lettura, stanno acquisendo importanza i sistemi RFID con tag passivi che operano nella banda UHF (Ultra-High Frequency). Il potenziale di tali sistemi è aumentato notevolmente negli ultimi anni a causa del

maggior conoscenza e applicazioni delle reti wireless. Tuttavia, i tag passivi e senza batteria pongono sfide uniche nel campo delle reti wireless a causa delle modeste capacità di elaborazione/memorizzazione dei tag. I tag passivi utilizzano l'energia irradiata dal lettore per alimentare i loro circuiti e anche per ritrasmetterla al lettore utilizzando il backscattering [3]. Un tag può essere collegato a un sensore e fungere da interfaccia di comunicazione, creando così un sensore RFID. Si possono facilmente immaginare futuri sistemi comprendenti tag RFID e sensori che presentano una combinazione di tutti i tipi di tag; con tag passivi che superano di gran lunga i tag attivi e semi-passivi.

La modalità di comunicazione nei sistemi RFID può essere descritta come segue: il lettore invia una sonda a una serie di tag all'interno della sua portata radio. In base al contenuto della sonda, un tag decide se inviare la propria risposta retrodiffondendo l'alimentazione fornita dal lettore. Se più tag dovessero rispondere alla sonda, allora al lettore arriveranno più risposte contemporaneamente, dando così luogo al *collisione dei tag* problema. Se le risposte di due o più tag entrano in collisione, è molto probabile che il segnale risulti danneggiato e illeggibile dal lettore. Si noti che il problema della collisione dei tag si verifica sul lettore. Se un tag si trova nel raggio d'azione di più di un lettore, a *collisione del lettore* può verificarsi nel tag [5]. Ciò si verifica quando due o più lettori trasmettono simultaneamente, in modo che un tag che si trova nel raggio di entrambi i lettori non possa ricevere la sonda inviata da nessuno dei lettori (Figura 8.1).

La collisione dei tag viene risolta eseguendo un protocollo anticollisione (chiamato anche protocollo di arbitrato o protocollo di risoluzione delle collisioni). L'obiettivo di un protocollo di arbitrato è quello di stabilire uno schema di trasmissione per i tag, in modo che alla fine ciascun tag riesca a inviare al lettore una risposta positiva. In alternativa, invece di risolvere completamente la collisione, il lettore potrebbe aver bisogno solo di una risoluzione parziale. Ad esempio, il lettore può eseguire un processo decisionale sequenziale e raccogliere dati dai tag per effettuare un test statistico di determinate ipotesi e interrompere il processo di arbitrato non appena ha un insieme sufficiente di risposte. In generale, i requisiti del protocollo di arbitrato sono determinati dal compito di livello superiore che viene imposto alla rete dei lettori. La collisione dei tag può essere un problema particolarmente acuto nei sistemi RFID UHF, a causa del raggio di lettura più ampio e quindi della possibilità di avere un numero maggiore di tag nel raggio d'azione del lettore.

La collisione dei lettori diventa un problema significativo in ambienti con un'elevata densità di lettori. Quindi, il problema è allocare le risorse di comunicazione (tempo, frequenza) ai lettori in modo da ridurre al minimo le interferenze tra di loro. Ad esempio, in Colorwave [6], i lettori comunicano tra loro con l'interfaccia aerea che viene utilizzata anche per comunicare con i tag e si coordinano in modo indipendente per ridurre al minimo le collisioni dei lettori utilizzando un

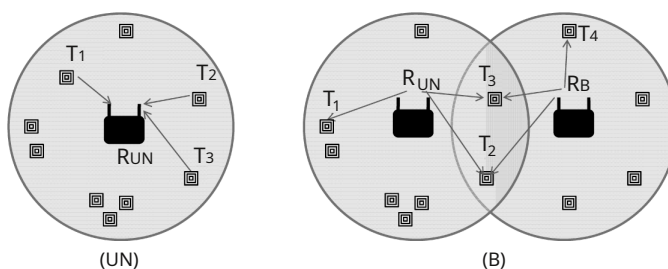


Figura 8.1 Illustrazione di (a) collisione di tag e (b) collisione di lettori. La collisione del tag avviene sul lettore R_{UN} . La collisione del lettore avviene sui tag T_2 , T_3 , ma non a T_1 , T_4 .

Algoritmo anticollisione a divisione di tempo distribuito. Uno studio completo dei problemi di collisione e delle procedure di risoluzione delle collisioni nei sistemi RFID è fornito in [7].

Questo capitolo introduce le idee e le prospettive principali di una classe di protocolli anticollisione denominata *algoritmi ad albero*. I protocolli ad albero rappresentano un paradigma importante nella progettazione di protocolli di arbitrato per l'accesso casuale. Questi protocolli sono stati introdotti per un singolo canale con accesso multiplo, in cui un singolo lettore raccoglie risposte da una popolazione di tag che si trovano nel raggio di quel lettore [8–10]. In sostanza, quando si incontra una collisione, i protocolli basati su alberi risolvono ricorsivamente la collisione finché tutti i tag inizialmente coinvolti nella collisione non hanno inviato con successo le loro risposte al lettore. I protocolli di arbitraggio basati su alberi rappresentano uno dei principali approcci per risolvere il problema della collisione dei tag nei sistemi RFID [7, 11]. Negli ultimi anni sono stati condotti numerosi studi che propongono nuove varianti degli algoritmi dell'albero [12–19]. L'obiettivo di questo capitolo non è fornire un resoconto dettagliato delle diverse proposte basate sugli alberi e/o il loro confronto quantitativo, ma piuttosto fornire informazioni sui meccanismi utilizzati negli approcci basati sugli alberi e sulle possibilità di ottimizzare ulteriormente questi algoritmi in vari scenari.

Questo capitolo è organizzato come segue. Sezione 8.2, dove i principi dei protocolli ad albero vengono discussi in un contesto generico, non strettamente limitato ai sistemi RFID. Questa sezione introduce innanzitutto diverse varianti di base del protocollo dell'albero, seguite da tecniche per migliorare le varianti di base e infine viene introdotto un quadro di arbitrato alternativo che consente la stima della popolazione dei tag. La sezione 8.3 descrive i protocolli ad albero utilizzati nelle specifiche per i sistemi RFID UHF. La sezione 8.4 presenta questioni pratiche che devono essere prese in considerazione, come gli errori di trasmissione e il trattamento dei tag in movimento/arrivati in ritardo. La Sezione 8.5 discute come i protocolli ad albero possano essere generalizzati a scenari con più lettori cooperativi. L'ultima sezione conclude il capitolo.

8.2 Principi dei protocolli anticollisione basati su alberi

Questa sezione introduce i principi e le idee di base utilizzati nei protocolli ad albero in un contesto che non è necessariamente limitato ai sistemi RFID. Ad esempio, non teniamo conto delle limitazioni computazionali/di elaborazione dei tag, della comunicazione soggetta a errori, ecc. Il modello di sistema utilizzato in questa sezione è idealizzato, ma sufficiente per introdurre i meccanismi pertinenti per comprendere i protocolli ad albero. La relazione con i sistemi reali e il funzionamento in base a presupposti non ideali è fornita nelle Sezioni 8.3 e 8.4.

8.2.1 Modello di sistema

In questa sezione descriviamo il contesto per presentare le idee di base dei protocolli anticollisione basati su alberi. A livello radio, assumiamo che se un tag si trova a una certa distanza D dal lettore, potrà sempre ricevere sonde da quel lettore senza errori. Viceversa, se un singolo tag nel raggio d'azione del lettore trasmette, allora il suo pacchetto viene ricevuto con successo dal lettore.

Usiamo il modello di collisione per canali ad accesso multiplo, il che significa che se due o più tag che si trovano nel raggio d'azione del lettore trasmettono simultaneamente, il lettore non riceve nessuno dei pacchetti trasmessi. Si noti che queste sono ipotesi piuttosto forti per i sistemi RFID passivi, poiché molti fattori possono causare errori di trasmissione, come sbiadimento, orientamento del tag, ostacoli tra il lettore e il tag, ecc. Tuttavia, questo modello idealizzato del livello MAC è sufficiente per trasmettere l'idea concetti pertinenti al sistema ad albero

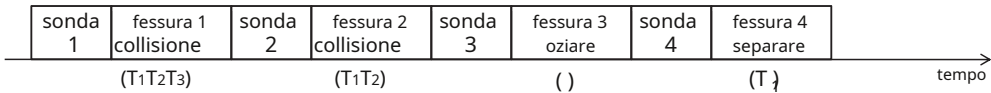


Figura 8.2Diagramma temporale che rappresenta il canale su cui vengono eseguiti l'arbitraggio e la comunicazione del lettore di tag. Le sonde vengono inviate dal lettore, mentre negli altri slot il lettore riceve. Ad esempio, nei tag dello slot 1 t_1 , t_2 , t_3 si scontrano quando si inviano risposte simultaneamente.

protocolli. Nella Sezione 8.4 allenteremo i presupposti e discuteremo come i vari tipi di errore influenzano i protocolli anticollisione.

Un tag invia una risposta solo se gli viene richiesto da una sonda inviata dal lettore. Ciò è illustrato nella Figura 8.2, dove i tag inviano risposte in risposta alla sonda. Ad esempio, dopo aver ricevuto la sonda 1, tag t_1 , t_2 , t_3 inviare le loro risposte. Abilitando le risposte da determinati sottoinsiemi di tag, il lettore arbitra le collisioni sul canale. Da notare che la durata del pacchetto inviato da ciascun tag è costante e pari ad uno slot, per cui se due tag trasmettono contemporaneamente i loro pacchetti risultano completamente sovrapposti. Il lettore quindi trasferisce il feedback ai tag nella sonda successiva. Ad esempio, nella Figura 8.2,sonda 2inviato dal lettore trasporta un feedback che informa i tag dell'esito (collisione) infessura 1.Quando K tag trasmettono nello stesso slot, l'interrogatore percepisce il canale in quello slot come:

- *Oziare* (IO) $Se K=0$ ovvero nessuna risposta ai tag.
- *Ricezione riuscita* (S) *Risoluzione dell'etichetta* $Se K=1$.
- *Collisione* (C) $Se K \geq 2$.

Per introdurre protocolli di arbitraggio basati su alberi, assumeremo che ciascun tag sia in grado di produrre bit casuali quando richiesto dal lettore. Questo presupposto verrà rivisto nella Sezione 8.4, dove discuteremo anche come i bit del vettore binario che descrive l'ID del tag possano essere utilizzati per l'arbitraggio.

In assenza di errori, l'efficienza dei protocolli arbitrali si misura sia in termini di tempo che di numero di messaggi. Se ci sono M tag, allora siamo interessati al tempo medio T_M che il protocollo utilizza per identificare tutti i tag. Permettere T essere una variabile casuale che rappresenta il tempo impiegato quando viene eseguita una particolare istanza del protocollo di arbitrato. Dalla Figura 8.2 si può vedere che T_M ha due componenti: (1) il tempo utilizzato per inviare le sonde; e (2) gli intervalli di tempo utilizzati per la trasmissione dai tag. Assumeremo che la durata di una sonda inviata dal lettore sia zero e ci concentreremo solo sugli intervalli di tempo consumati per il backscattering. Con la sonda a lunghezza zero, l'efficienza temporale di un protocollo arbitrale è definito come $\eta = \frac{1}{T_M} \rightarrow$ vedere [20]. Nota che un'altra misura quella è tradizionalmente utilizzato per valutare la prestazione dei protocolli ad albero è il numero medio di messaggi M inviato durante il processo arbitrale. Ciò sarebbe rilevante se si considerassero i tag attivi, poiché in quel caso ogni messaggio trasmesso consuma energia dal tag. D'altra parte, quando si ha a che fare con i tag passivi, M è piuttosto irrilevante, in quanto l'energia del tag è fornita dal lettore e l'energia media spesa dal lettore è linearmente proporzionale alla durata media del processo di arbitraggio.

8.2.2 Protocolli di base dell'albero

I protocolli basati su alberi (noti anche come *protocolli ad albero diviso* o *protocolli di camminata sugli alberi*) sono emersi come soluzione al problema dell'accesso multiplo su un mezzo condiviso. Diversi gruppi di ricerca hanno inventato praticamente lo stesso approccio piuttosto simultaneamente [8–10]. La motivazione per proporre questi algoritmi può essere spiegata attraverso un semplice esempio con un singolo lettore e due tag. Il lettore invia una sonda richiedendo risposte dai tag. Dato che il lettore non conosce l'ID dei tag, questa sonda sollecita una risposta da un tag non specificato e quindi entrambi i tag hanno il diritto di inviare una risposta. Dopo aver osservato la collisione, il lettore sa che ci sono almeno due tag che hanno inviato una risposta. Tuttavia, il numero e l'ID di questi tag sono sconosciuti al lettore, dando luogo al problema della simmetria, cioè tutti i tag coinvolti nella collisione appaiono uguali al lettore. Per rompere questa simmetria, il lettore utilizza la randomizzazione. Ciò significa che dopo la collisione, entrambi i tag lanciano una moneta equilibrata per generare un valore 0 o 1 e nella sonda successiva il lettore chiede solo al tag che ha ottenuto 0 di inviare una risposta. Questa procedura viene ripetuta finché i due tag non hanno ottenuto valori diversi per i bit casuali.

La Figura 8.3 mostra, attraverso un esempio, come questa idea di base possa essere generalizzata a più di due tag. Nell'esempio, il numero totale di tag N è uguale a $N=8$. Definiamo molteplicità di collisione (o molteplicità di conflitto) come il numero di tag che trasmettono quando si osserva una collisione. Il nodo etichettato S_i si riferisce al risultato nel i -esimo slot. Ad esempio, la collisione nel primo slot S_1 ha una molteplicità di otto, perché tutti i tag hanno inviato le loro risposte quando hanno risposto al sondaggio iniziale. Il livello di un nodo dell'albero è la lunghezza del percorso da quel nodo alla radice dell'albero. Ogni nodo dell'albero è associato in modo univoco ad una stringa chiamata *indirizzo*. L'indirizzo di un nodo dell'albero è determinato dai risultati del lancio dei tag appartenenti a quel nodo. Utilizzando la rappresentazione ad albero, possiamo dire che an

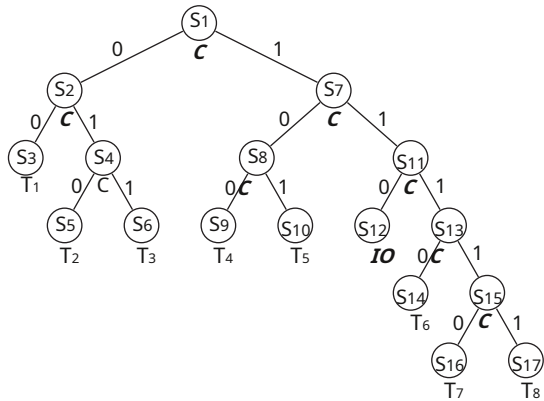


Figura 8.3 Un'istanza dell'algoritmo dell'albero binario per la molteplicità delle collisioni di $N=8$. Ciascun vertice rappresenta uno slot che può trovarsi in uno dei tre stati: Idle (I), Single (S) o Collision (C). Per lo stato del canale "S", T_i denota il tag risolto o singolato.

l'indirizzo (un nodo) è abilitato nello slot. Se i tag che appartengono a quel nodo possono trasmettere. Alla fessura S_1 la radice dell'albero (livello zero) è abilitata, tutti gli otto tag trasmettono e il lettore osserva una collisione. L'indirizzo del nodo radice è vuoto, indicato da ϵ . Ogni tag lancia una moneta equa e, in questo caso di protocollo ad albero, τ_1 , τ_2 , τ_3 ottengono valore 0. Allo slot S_2 , il discendente sinistro del nodo radice è abilitato e τ_1 , τ_2 , τ_3 sono abilitati a trasmettere. Da notare che il nodo è abilitato su slot S_2 è al livello uno dell'albero e ha indirizzo 0. I tag τ_4 – τ_8 che ottengono il valore della moneta 1 dopo lo slot S_1 appartengono all'altro nodo al livello uno, con l'indirizzo 1. Presente collisione nello slot S_2 , i tag τ_1 , τ_2 , τ_3 lanciano di nuovo monete giuste. Soltanto τ_1 ottiene 0, mentre τ_2 , τ_3 ottengono 1. Pertanto, quando il nodo con indirizzo 00 (al livello due dell'albero) è abilitato, solo τ_1 invia una risposta che viene ricevuta correttamente dal lettore. Avere trasmissione da un unico tag, a slot S_4 viene abilitato un altro nodo allo stesso livello, con indirizzo 01. In base a questa spiegazione il lettore può procedere con gli altri passi dell'esempio di Figura 8.3. La Figura 8.5 mostra il diagramma temporale dell'algoritmo dell'albero tratto dall'esempio della Figura 8.3. Si può vedere che il lettore risolve ricorsivamente le collisioni finché non ottiene un'unica risposta (riuscita) da ciascun tag.

La Figura 8.4 mostra una possibile implementazione della variante base del protocollo ad albero binario utilizzando uno pseudocodice. Il codice per l'algoritmo eseguito sul lettore rappresenta un attraversamento completo dell'albero. IMPOSTANDO $un = vuoto$, il lettore può ricominciare l'attraversamento dalla radice dell'albero. Lo pseudocodice sul tag presuppone che ciascuna sonda contenga l'indirizzo completo. Si noti che gli algoritmi originali dell'albero proposti da [10] vengono eseguiti in modo completamente distribuito, senza sonde centralizzate, utilizzando un'implementazione basata sullo stack.

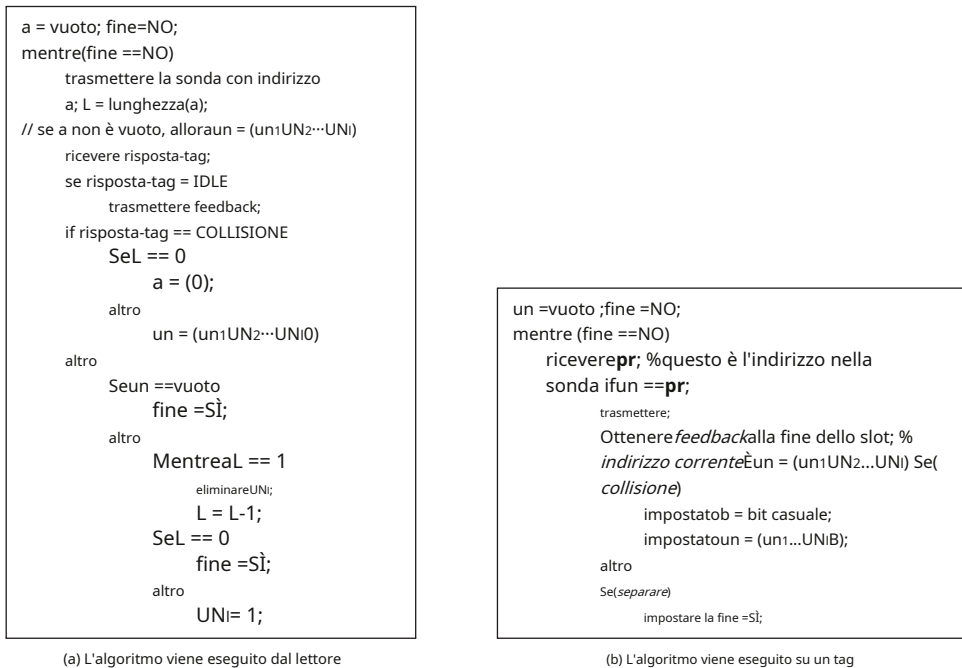


Figura 8.4 Lo pseudocodice degli algoritmi viene eseguito sul lettore e su un tag per la variante base del protocollo ad albero binario.

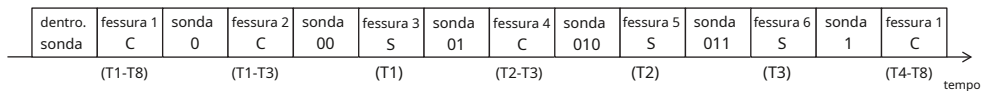


Figura 8.5Diagramma temporale per parte dell'esecuzione del protocollo descritto nella Figura 8.3. La notazione $(\tau_{io}-\tau_j)$ sotto uno slot indica l'insieme di tag $\tau_{io}, \tau_{io+1}, \tau_{io+2} \dots \tau_j$ che trasmette in quello slot.

Gli algoritmi ad albero appartengono alla classe dei protocolli di risoluzione delle collisioni, piuttosto che ai protocolli di prevenzione delle collisioni, dove un rappresentante di questi ultimi sono i protocolli che utilizzano il backoff casuale, come il protocollo MAC per IEEE 802.11 [21]. Quando non ci sono altri errori oltre alle collisioni, l'algoritmo dell'albero è garantito per raccogliere una risposta da ciascun tag che si trova nel raggio d'azione del lettore. Un vantaggio interessante è che l'algoritmo crea indirizzi ad hoc per i tag. Vale a dire, dopo la risoluzione della collisione, il lettore potrebbe aver bisogno di comunicare con un particolare tag. Invece di utilizzare l'indirizzo completo di quel tag, il lettore può assegnare un indirizzo breve al tag e questo indirizzo è rappresentato dall'indirizzo del nodo dell'albero in cui questo tag è stato risolto o singolarizzato con successo. Ad esempio, nella Figura 8.3 l'indirizzo breve ad hoc del tag τ_3 è 011.

8.2.3 Miglioramenti al protocollo dell'albero base

In questa sezione descriviamo diverse idee che migliorano e generalizzano il protocollo base dell'albero, descritto nella sezione precedente.

Innanzitutto, si noti che l'algoritmo dell'albero descritto costruisce un albero mediante ramificazione binaria come dettato dal bit flipping. La ramificazione può essere generalizzata a M piuttosto che due risultati nel processo di randomizzazione. Più grandi M diminuisce la probabilità di collisione, ma aumenta la probabilità di ottenere uno slot inattivo. Se lo slot inattivo ha la stessa durata di uno slot singolo/coordinato, allora è chiaro che aumenta M non aumenta l'efficienza. Tuttavia, uno slot inattivo può avere una durata inferiore rispetto a uno slot in cui stanno trasmettendo uno o più tag. Questo perché il lettore, in linea di principio, non ha bisogno di attendere l'intera durata dello slot per rilevare che nessun tag è stato trasmesso. Quindi la durata T_{io} di uno slot inattivo può essere inferiore alla durata dello slot con trasmissione del tag, indicato con T_s . Un tale modello per la risoluzione delle collisioni è chiamato modello CSMA (Carrier Sense Multiple Access). Se $T_{io} < T_s$, quindi uno slot inattivo è molto "economico" in termini di durata e il protocollo dell'albero dovrebbe essere progettato in modo da produrre slot inattivi con maggiore probabilità rispetto alla probabilità che si verifichino slot in collisione. Tuttavia in questo testo non tratteremo specificatamente il modello CSMA. Possibili ottimizzazioni degli algoritmi basati su alberi nell'impostazione CSMA sono discusse in [22] e [18].

Massey in [23] e Tsybakov e Mikhailov in [10] hanno proposto un modo semplice per migliorare la variante di base dell'algoritmo dell'albero binario. Hanno notato che ci sono alcuni nodi dell'albero che contengono sicuramente più di un tag e quindi provocheranno sicuramente una collisione se il nodo è abilitato. Questi nodi dovrebbero essere saltati durante l'attraversamento dell'albero. Ad esempio, nella Figura 8.3, dopo la collisione nello slot S_{11} e uno slot inattivo S_{12} , è chiaro che l'abilitazione dell'indirizzo 111 comporta una certa collisione. Quindi, dopo lo slot inattivo S_{12} , i terminali appartenenti al nodo 111 lanciano immediatamente una moneta nella fessura S_{13} il nodo abilitato è 1110. Questo algoritmo verrà chiamato algoritmo Modified Binary Tree (MBT).

Tuttavia, come discusso nella Sezione 8.4, questa piccola ottimizzazione ha un prezzo molto alto se gli errori di trasmissione si verificano con probabilità diversa da zero.

Inizialmente, i ricercatori hanno osservato che conoscere la molteplicità (numero) di tag coinvolti in una collisione può accelerarne la risoluzione. Capetanakis in [9] ha osservato che gli algoritmi degli alberi binari sono più efficienti per conflitti di piccola molteplicità e ha applicato questa osservazione per ideare un *algoritmo dell'albero dinamico* per scenari con arrivi di Poisson. L'idea è di dividere l'insieme iniziale di M tag in gruppi più piccoli e quindi applicare un protocollo ad albero all'interno di ciascun gruppo. La stima della molteplicità per accelerare i protocolli degli alberi è stata utilizzata in [24] e [20]. In entrambi i lavori, la proposta è un algoritmo ibrido composto da due fasi. La prima fase è dedicata alla stima della molteplicità. Dopo aver ottenuto un preventivo N , inizia la seconda fase. I terminali irrisolti vengono suddivisi casualmente in circa N gruppi e ciascun gruppo viene risolto utilizzando l'algoritmo dell'albero di base.

Si noti che sia [24] che [20] hanno una fase esplicita per la stima N . L'accuratezza di tale stima dipende dall'utilizzo di parametri specifici dell'algoritmo scelti in anticipo. Al contrario, in [18] abbiamo introdotto un nuovo framework per protocolli di arbitrato basati su alberi in cui una stima in esecuzione N si ottiene la popolazione iniziale, che diventa sempre più accurata man mano che la risoluzione della collisione progredisce. Prima di procedere nei dettagli del framework di [18], introduciamo l'algoritmo Clipped Binary Tree (CBT), che è l'ingrediente principale degli algoritmi ad albero con una stima corrente.

L'algoritmo CBT è stato introdotto indipendentemente da diversi autori nel contesto dell'accesso casuale per una popolazione infinita di terminali che genera richieste di trasmissione con arrivi di Poisson [25]. È identico all'algoritmo MBT tranne per il fatto che viene interrotto, ovvero l'albero viene tagliato ogni volta che due trasmissioni consecutive riuscite seguono un conflitto. CBT risolve parzialmente il conflitto batch, poiché non necessariamente tutti i nodi di un batch iniziale vengono risolti durante l'esecuzione di CBT. Ad esempio, facendo riferimento all'albero nella Figura 8.3, la prima istanza dell'algoritmo CBT inizia in slot S_1 e termina nello slot S_6 , risultando in tre tag risolti. Ora, l'algoritmo standard dell'albero binario continuerebbe ad abilitare il nodo etichettato con S_7 , cioè tutti i tag che hanno invertito 1 dopo la collisione iniziale. L'osservazione chiave di [18] può essere applicata a questo esempio come segue. Dopo la terminazione del primo algoritmo CBT, si nota che 3 tag hanno capovolto 0, per cui il valore atteso del numero di tag che hanno inizialmente capovolto 1 è 3. Ciò implica che l'abilitazione del nodo S_7 , che appartiene al livello 1 dell'albero, molto probabilmente provocherà una collisione e quindi la prossima sonda dovrebbe abilitare direttamente un nodo che ha un livello più alto nell'albero. Ad esempio, se il nodo con indirizzo 10, dal livello 2, viene abilitato dopo la terminazione dell'algoritmo CBT iniziale, allora uno slot viene salvato saltando il nodo S_7 al livello 1. In tal caso, la successiva istanza dell'algoritmo CBT inizierebbe abilitando il nodo con indirizzo 10 e terminerebbe quando il tag z invia la risposta con successo, cioè dopo che è stato abilitato l'indirizzo 101. Nella sezione successiva, descriviamo il framework che stima sistematicamente quali sottoinsiemi di tag dovrebbero essere abilitati per la trasmissione al termine di una singola istanza di CBT.

8.2.4 Quadro generale di arbitrato per protocolli basati su alberi

La parte fondamentale del quadro proposto in [18] è l'interpretazione delle procedure di randomizzazione utilizzate in un protocollo ad albero. Supponiamo che, alla trasmissione della sonda iniziale, ogni tag generi un numero reale casuale, uniformemente distribuito nell'intervallo