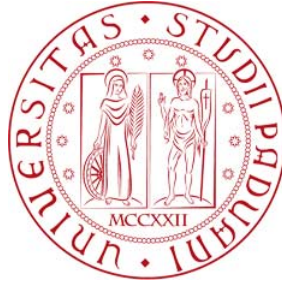


# PROTOCOLLI FRAMED SLOTTED ALOHA

RELATORE: Ch.mo Prof. Andrea Zanella

LAUREANDO: Enrico Carraro

A.A. 2012-2013



UNIVERSITÀ DEGLI STUDI DI PADOVA  
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE  
CORSO DI LAUREA IN INGEGNERIA DELL'INFORMAZIONE

---

# PROTOCOLLI FRAMED SLOTTED ALOHA

RELATORE: Ch.mo Prof. Andrea Zanella

LAUREANDO: *Enrico Carraro*

Padova, 26 settembre 2013



# Indice

<b>Sommario</b>	<b>1</b>
<b>1 Introduzione</b>	<b>2</b>
<b>2 Il protocollo Aloha</b>	<b>4</b>
2.1 Aloha puro . . . . .	4
2.2 Slotted Aloha . . . . .	7
2.3 Framed Slotted Aloha . . . . .	8
<b>3 Evoluzione di Framed Slotted Aloha: i protocolli CRDSA e IRSA</b>	<b>10</b>
3.1 Presentazione . . . . .	10
3.2 Descrizione del sistema . . . . .	11
3.3 Rappresentazione attraverso grafo bipartito . . . . .	12
3.4 Prestazioni asintotiche . . . . .	15
3.5 Simulazioni e risultati numerici . . . . .	16
<b>4 Frameless Aloha</b>	<b>19</b>
4.1 Presentazione dei frameless code: codici LT . . . . .	19
4.2 Frameless Aloha: modello del sistema . . . . .	23
4.3 Analisi e prestazioni . . . . .	24
<b>5 Conclusioni</b>	<b>28</b>
<b>Bibliografia</b>	<b>30</b>

## Sommario

Il protocollo Aloha è un noto protocollo di accesso al mezzo a contesa indicato per canali di tipo broadcast. In questo lavoro si studiano alcune innovazioni del protocollo Aloha recentemente proposte che riescono a incrementarne le prestazioni in termini di throughput. Il protocollo Contention resolution diversity slotted ALOHA (CRDSA) sfrutta la trasmissione di pacchetti ripetuti su slot diversi e tecniche di Successive Interference Cancellation (SIC) per aumentare la probabilità di recupero dei pacchetti collisi. Irregular repetition slotted ALOHA (IRSA) è un'evoluzione del protocollo CRDSA che propone un efficace schema di trasmissione a rate variabile dei pacchetti ritrasmessi secondo determinate distribuzioni di probabilità, in grado di migliorare significativamente il throughput. Infine il protocollo Frameless Aloha, in analogia coi codici "rateless", propone un'evoluzione dello schema Framed Aloha in cui il frame non ha dimensione fissa.

# Capitolo 1

## Introduzione

In un sistema in cui il canale è condiviso tra tutti gli utenti un problema da risolvere è quello di garantire che trasmissioni di utenti diversi interferiscano il meno possibile tra loro così da permettere a tutti gli utenti di poter comunicare affidabilmente col rispettivo interlocutore. A tal fine i protocolli di accesso al mezzo descrivono le regole che ogni utente deve seguire per trasmettere. I metodi di accesso al mezzo si possono dividere in tre classi principali [1, pp. 599-600]:

- Protocolli deterministici: un controllore programma l'accesso al mezzo di ogni utente in modo che non ci sia più di un utente ad accedere contemporaneamente alle stesse risorse del sistema, siano esse slot temporali (time-division multiple access, TDMA), bande di frequenza (frequency-division multiple access, FDMA) o codici (code-division multiple access, CDMA). Essendo l'ordine stabilito dal controllore, non c'è contesa del mezzo tra gli utenti.
- Protocolli demand-based: un controllore stabilisce l'utente che accederà al mezzo nel prossimo slot temporale e la grandezza dello slot temporale di servizio, in modo dinamico in base a una prestabilita strategia, che tenga conto per esempio del tempo trascorso da un utente in attesa di trasmissione o della priorità dei suoi pacchetti da trasmettere.
- Protocolli ad accesso casuale: diversamente dalle classi precedenti, non c'è un controllore a gestire le risorse del canale. Ogni utente infatti trasmette appena ha pacchetti da trasmettere senza alcuna coordinazione. Ciò permette di realizzare sistemi distribuiti, in grado di operare anche in reti la cui topologia o il numero di utenti possono non essere noti a tutti gli utenti.

---

Il protocollo Aloha, che verrà trattato in seguito, appartiene alla classe dei protocolli ad accesso casuale. I principali vantaggi di questi protocolli derivano come detto dall'assenza di un controllore: la complessità del sistema è ridotta e ogni utente decide quando trasmettere. Demandare ai singoli utenti la decisione di quando trasmettere è particolarmente vantaggioso in caso di sistemi a basso traffico di trasmissione. In tali condizioni un protocollo deterministico assegnerebbe pari risorse del canale sia agli utenti che devono trasmettere sia agli utenti che non devono trasmettere, comportando uno spreco di risorse da parte di questi ultimi.

Tutto ciò avviene al costo di dover gestire le inevitabili collisioni di pacchetti inviati da utenti non coordinati. Dopo la presentazione nel prossimo capitolo del protocollo Aloha nella sua versione pura, “slotted” e “framed”, verranno introdotti nel Capitolo 3 i protocolli Contention resolution diversity slotted ALOHA (CRDSA) e Irregular repetition slotted ALOHA (IRSA), che si prefiggono di gestire le collisioni tramite tecniche di Interference Cancellation. Nel Capitolo 4 verrà infine presentato il protocollo Frameless Aloha, che applica una strategia di accesso al mezzo derivata dalla filosofia alla base dei codici “rateless”.

# Capitolo 2

## Il protocollo Aloha

### 2.1 Aloha puro

Il nome del protocollo Aloha deriva dal noto saluto hawaiano: il protocollo infatti fu originariamente sviluppato presso la University of Hawaii, all'interno di un progetto che prevedeva lo sviluppo di una rete satellitare che collegasse i dipartimenti dislocati sulle varie isole dell'arcipelago all'unità di calcolo centrale situata nel campus di Manoa Valley vicino Honolulu [3]. Il funzionamento di questo protocollo è molto semplice:

- ogni utente trasmette i pacchetti appena vengono generati, e in caso di avvenuta ricezione, riceve un feedback;
- se un utente non riceve nessun feedback, ritrasmette il pacchetto dopo un certo tempo  $\tau_b$  chiamato tempo di back-off, modellato con una opportuna variabile aleatoria. L'utente in questo caso viene detto backlogged.

Se due o più utenti provano a trasmettere i rispettivi pacchetti nello stesso istante, tali pacchetti vanno in collisione e si assume che tutti i pacchetti collisi vadano perduti. In Figura 2.1 è mostrato l'andamento temporale del sistema di trasmissioni e collisioni in un sistema Aloha.

Per valutare le prestazioni studiamo il throughput, adottando prima alcune ipotesi semplificative [1, pp. 624-625]:

- il numero di utenti del sistema è assunto infinito;



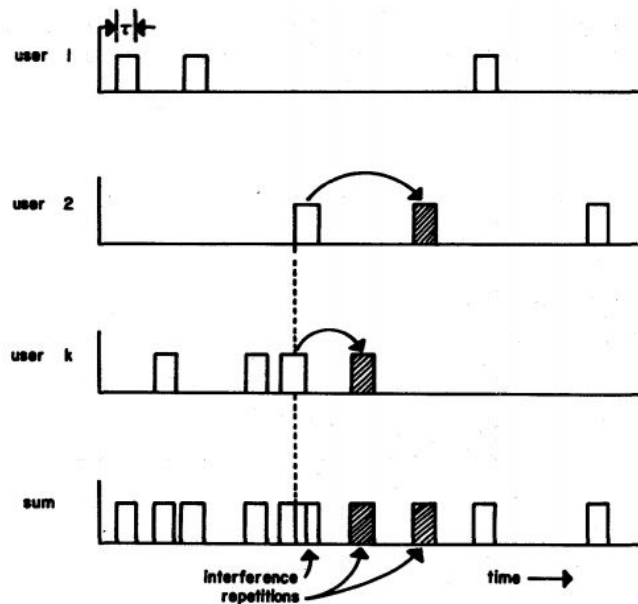


Figura 2.1: Collisioni in sistema Aloha, da [3].

- il tasso di arrivi dei pacchetti è modellato con un processo di Poisson di parametro  $\lambda$ . Quando un pacchetto viene generato, viene accodato nella coda di un utente che non ha altri pacchetti da trasmettere, in modo da non avere ritardo di accodamento; inoltre i pacchetti hanno lunghezza costante e vengono trasmessi in un tempo costante  $T$ ;
- il tempo di backoff  $\tau_b$  è molto maggiore di  $T$  (tempo di pacchetto) ed è modellato secondo una variabile aleatoria con funzione di densità di probabilità esponenziale:

$$p_{\tau_b}(a) = \beta e^{-\beta a}, \quad a, \beta > 0 \quad (2.1)$$

- gli utenti trasmettono in modo indipendente, contendendosi il canale; quando la trasmissione di più pacchetti si sovrappone, si assume che tutti i pacchetti vadano persi e che l'unica causa di pacchetti corrotti sia la collisione, mentre il canale è assunto senza rumore;
- il processo totale degli arrivi dei pacchetti, dato dalla sovrapposizione degli arrivi dei nuovi pacchetti generati e degli arrivi dei pacchetti ritrasmessi, è assunto poissoniano.

## 2. IL PROTOCOLLO ALOHA

---

Il tasso totale  $\lambda_{tot}$  degli arrivi dei pacchetti è dato da  $\lambda$ , tasso dei pacchetti generati trasmessi con distribuzione poissoniana, a cui si somma il tasso dei pacchetti ritrasmessi dovuti agli utenti backlogged, che è funzione del numero di utenti  $n$  in backlog,  $\lambda_b(n)$ :

$$\lambda_{tot}(n) = \lambda + \lambda_b(n) \quad (2.2)$$

Ora si osserva che per trasmettere correttamente un pacchetto, nessun altro pacchetto deve essere trasmesso  $T$  secondi prima o  $T$  secondi dopo l'inizio della trasmissione del pacchetto in oggetto in modo da evitare collisioni. L'intervallo di tempo  $2T$  è detto intervallo di vulnerabilità. Quindi la probabilità di successo corrisponde alla probabilità di non avere nessun ulteriore arrivo durante un intervallo di  $2T$  secondi e vale:

$$P_{succ} = e^{-\lambda_{tot}(n)2T} \quad (2.3)$$

Il Traffico offerto normalizzato, essendo funzione di  $\lambda_{tot}(n)$ , dipenderà anch'esso da  $n$ :

$$G(n) = \lambda_{tot}(n)T \quad (2.4)$$

e indica il periodo di tempo di occupazione del canale. Solo però una percentuale di questo tempo è impiegata a trasmettere pacchetti ricevuti correttamente, e sarà:

$$S_{ALOHA} = G(n) \cdot P_{succ} = G(n) e^{-2G(n)} \quad (2.5)$$

ed è il throughput normalizzato di un sistema Aloha.

Per trovare il punto di picco di throughput  $S_{max}$ , è possibile trovare un punto critico della funzione  $S(G(n))$  e verificare che trattasi di un punto di massimo:

$$\frac{d}{dG(n)} S_{ALOHA} = e^{-2G(n)} (1 - 2G(n)) = 0 \quad \Rightarrow \quad G(n) = \frac{1}{2} \quad (2.6)$$

a cui corrisponde un throughput di circa 0.18. Risulta evidente come Aloha non sia particolarmente prestante per reti ad alto traffico a causa delle collisioni tra pacchetti.

## 2.2 Slotted Aloha

È possibile incrementare le prestazioni del protocollo Aloha puro introducendo nel sistema un parziale coordinamento tra gli utenti, al fine di migliorare le performance quando c'è un numero più ampio di utenti a contendersi il mezzo. Slotted Aloha [4] adotta due ulteriori strategie rispetto all'Aloha puro:

- il tempo viene suddiviso in slot temporali, ciascuno della durata di trasmissione di un pacchetto  $T$ ;
- gli utenti sono sincronizzati in modo tale da trasmettere unicamente a inizio di uno slot, e qualora un pacchetto fosse disponibile per la trasmissione nel mezzo di uno slot temporale, esso dovrebbe aspettare l'inizio dello slot successivo.

Questi accorgimenti fanno sì che si riduca l'intervallo di vulnerabilità di un pacchetto: sia  $\tau_0$  l'istante di inizio di uno slot temporale in cui viene trasmesso un pacchetto; la trasmissione andrà a buon fine se nell'intervallo  $(\tau_0 - T, \tau_0)$  nessun altro utente accoda un pacchetto in quello stesso slot. Il feedback di avvenuta trasmissione viene mandato a termine dello slot. Rispetto all'Aloha, l'intervallo di vulnerabilità è quindi dimezzato. In Figura 2.2 è illustrato uno schema delle trasmissioni e delle collisioni di Slotted Aloha.

La trattazione delle prestazioni ricalca quella vista per l'Aloha puro: il processo delle trasmissioni è ancora poissoniano di parametro  $\lambda_{tot}(n)$ , e la probabilità di successo è

$$P_{succ} = e^{-\lambda_{tot}(n)T} = e^{-G(n)} \quad (2.7)$$

avendo tenuto conto del nuovo intervallo di vulnerabilità che determina un traffico offerto pari a  $G(n) = \lambda_{tot}(n) * T$ . Studiando il punto critico di questa funzione si ottiene:

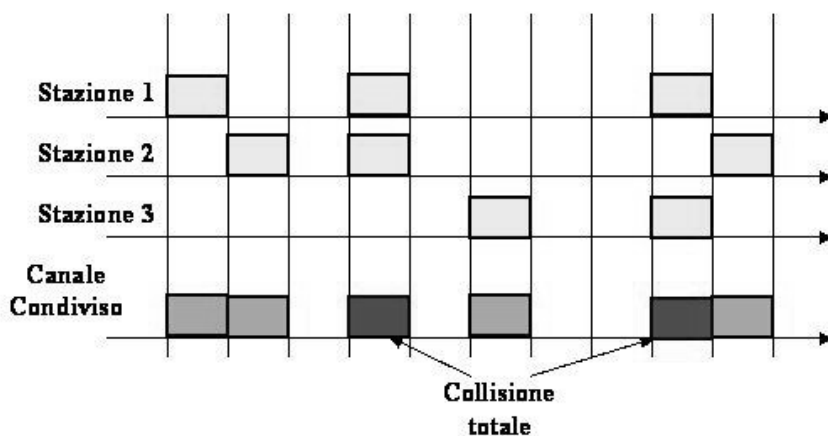


Figura 2.2: Collisioni in Slotted Aloha

$$\frac{d S_{ALOHA}}{d G(n)} = e^{-G(n)} (1 - G(n)) = 0 \Rightarrow G(n) = 1 \quad (2.8)$$

Il throughput in questo caso è massimo quando  $G(n)=1$ , e vale  $S_{MAX} = \frac{1}{e} \approx 0.37$ , cioè quasi il doppio del massimo throughput dell'Aloha puro. In Figura 2.3 sono riportati gli andamenti dei throughput normalizzati di Aloha e Slotted Aloha.

## 2.3 Framed Slotted Aloha

Framed Slotted Aloha è una variante del protocollo Slotted Aloha in cui gli slot sono organizzati in frame. All'inizio di ogni frame, ogni utente sceglie uno slot del frame in cui trasmettere il pacchetto, in modo casuale e indipendente dagli altri utenti. In caso di positiva ricezione, l'utente riceve il feedback solo al termine del frame. In caso di mancato feedback, il pacchetto viene ritrasmesso nel frame successivo. I protocolli che verranno presentati nei prossimi capitoli adottano questo approccio.

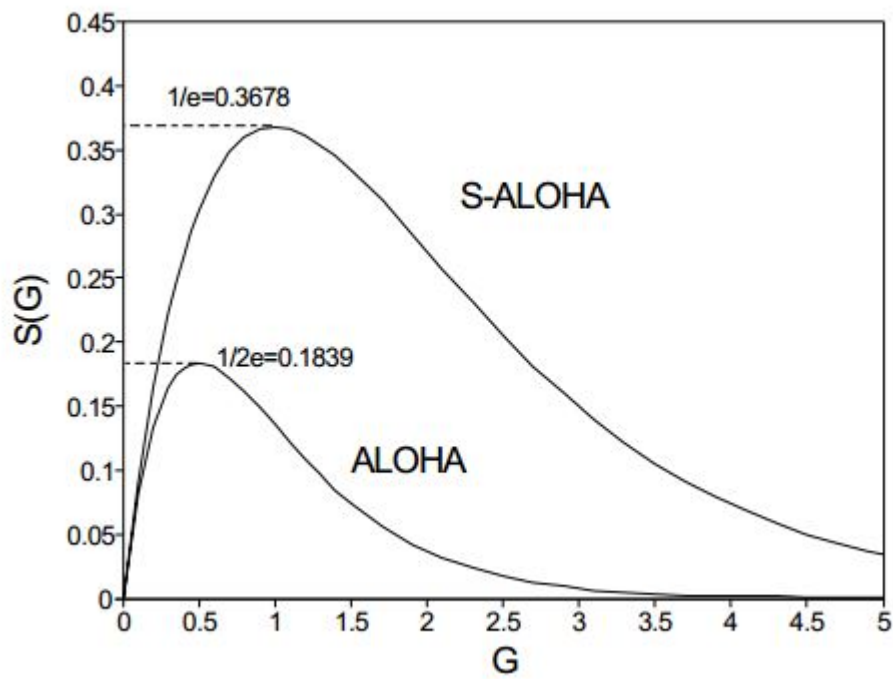


Figura 2.3: Confronto throughput Aloha VS Slotted Aloha [1]

## Capitolo 3

# Evoluzione di Framed Slotted Aloha: i protocolli CRDSA e IRSA

### 3.1 Presentazione

Nei protocolli casuali finora presentati si è sempre assunto che una collisione comporti necessariamente la perdita dei pacchetti collisi. Contention Resolution Diversity Slotted Aloha (CRDSA) [9] e la sua estensione Irregular Repetition Slotted ALOHA (IRSA) [5] sono due implementazioni del protocollo Framed Slotted Aloha in cui le collisioni non sono considerate necessariamente distruttive, potendo essere risolte grazie alla tecnica di successive interference cancellation (SIC). CRDSA sfrutta la tecnica di ripetizione di pacchetti (burst repetition): ogni pacchetto viene trasmesso in due slot diversi all'interno dello stesso frame. Ciascuna delle due istanze del pacchetto possiede un puntatore allo slot in cui la rispettiva copia è stata inviata. Ogni volta che un pacchetto viene identificato e ricevuto correttamente, estraendo il suo puntatore si risale allo slot in cui è contenuta la sua replica, e il potenziale contributo di interferenza causato dalla replica viene eliminato. Questa procedura viene reiterata fino a permettere il riconoscimento, potenzialmente, di tutti i pacchetti inviati in un frame. Il recupero dei pacchetti collisi si traduce in un notevole miglioramento del throughput, inteso come probabilità di corretta trasmissione del pacchetto per slot, in confronto al protocollo Slotted Aloha. IRSA è una evoluzione di CRDSA sviluppata a partire da un

analisi di CRDSA basata su i grafi bipartiti, che vedremo in seguito. La struttura a grafo bipartito suggerisce di generalizzare il funzionamento delle ripetizioni, concedendo a ogni pacchetto di essere ripetuto con una frequenza di ripetizione variabile. Le prestazioni in termini di throughput dipenderanno dalla distribuzione di probabilità su cui verrà modellata la frequenza di ripetizione dei pacchetti.

### 3.2 Descrizione del sistema

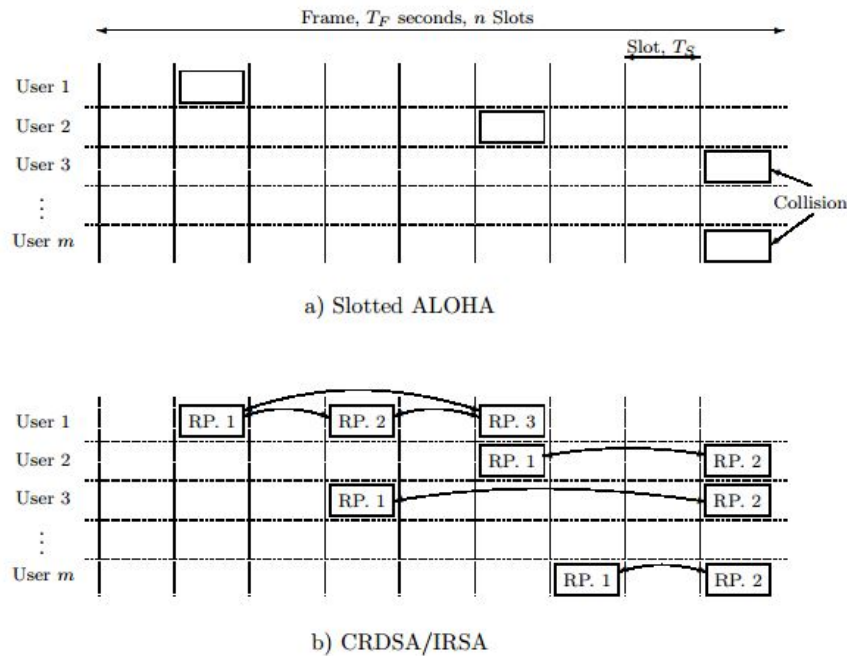


Figura 3.1: Confronto tra a) frame in Slotted aloha e b) frame in CRDSA/IRSA

Si consideri un sistema con frame di durata  $T_F$ , ciascuno composto da  $n$  slot di durata  $T_N = T_F/n$ . Assumiamo che in ogni frame un numero finito  $m$  di utenti provi a trasmettere un solo pacchetto, sia che si tratti di un pacchetto nuovo o di un pacchetto colliso ritrasmesso. Il traffico offerto normalizzato, ovvero il numero medio di pacchetti trasmessi per slot è pari a  $G = m/n$ , mentre il throughput normalizzato  $T$  è definito come la probabilità di corretta trasmissione di un pacchetto per slot. Ogni volta che un pacchetto è decodificato correttamente, il suo puntatore viene usato per trovare lo slot in cui la sua copia è stata trasmessa. Se per esempio l'altra copia ha colliso con un altro pacchetto, è possibile sottrarre dal segnale ricevuto nello slot corrispondente il contributo della copia che ha cau-

sato l'interferenza. Lo slot appena liberato potrebbe portare alla decodifica di un nuovo pacchetto. Questo processonoto come Interference Cancellation procede iterativamente cercando di risolvere quante più collisioni possibili.

L'approccio seguito dal protocollo IRSA è una generazione di quanto scritto sopra: ogni pacchetto è ritrasmesso  $l$  volte all'interno dello stesso frame, col rate di ritrasmissione  $l$  che varia da pacchetto a pacchetto in base a una data distribuzione di probabilità. In questa ottica si può vedere il protocollo CRDSA come un caso speciale di IRSA con  $l$  fissato a 2 per ogni pacchetto.

### 3.3 Rappresentazione attraverso grafo bipartito

Viene introdotta ora una rappresentazione per mezzo di un grafo bipartito del processo di Interference Cancellation. Continuiamo a considerare frame divisi in  $n$  slot, in cui tentano di trasmettere  $m$  utenti. Lo stato del frame può essere descritto iterazione dopo iterazione da un grafo bipartito,  $G = (B, S, E)$ , dove  $B$  è un insieme di  $m$  nodi (ciascuno corrispondente a un pacchetto trasmesso) chiamati *burstnode*, un insieme  $S$  di nodi detti *sumnode* (ciascuno per ogni slot del frame), e un insieme  $E$  di archi. Un arco connette un elemento  $b_i \in B$  a un elemento  $s_j \in S$  se e solo se una copia dell' $i$ -esimo pacchetto è trasmessa nello slot  $j$ -esimo. Quindi un pacchetto con  $l$  repliche è rappresentato da un *burst node* con  $l$  archi uscenti, mentre uno slot in cui collidono  $l$  pacchetti è rappresentato da un *sum node* in cui incidono  $l$  archi. Il numero di archi incidenti in un vertice si definisce *grado* di quel vertice. I grafi i cui burst node hanno un grado costante vengono definiti grafi regolari, altrimenti vengono definiti grafi irregolari. Da queste definizioni segue che lo schema CRDSA è descritto da un grafo regolare, mentre IRSA da grafo irregolare. La copia di un pacchetto può essere rivelata con due modalità:

- la copia è decodificata correttamente nello slot in cui si trova, in quanto unica trasmissione in quello slot oppure grazie alla cancellazione di tutti gli altri pacchetti interferenti;
- un'altra copia dello stesso pacchetto è stata decodificata in un altro slot.

Segue un esempio del processo di Interference Cancellation su un frame di  $n = 4$  slot, con  $m = 4$  utenti pronti a trasmettere il cui avanzamento iterativo è



descritto in Figura 3.2, dove i cerchi rappresentano i burst node, mentre i quadrati rappresentano i sum node. Etichettiamogli archi incidenti in un burst node già rilevato con “1”, e con “0” gli altri. Il processo iterativo inizia decodificando il pacchetto presente nel secondo sum node, che non è un pacchetto colliso, essendo il grado del secondo burst node pari a 1. Ora tutte le repliche del pacchetto appena codificato possono venire rimosse dagli slot in cui erano state trasmesse. Gli archi rivelati, etichettati con “1”, vengono quindi rimossi dal grafo. Nella seconda iterazione cerchiamo sum node di grado 1: questi nodi rappresentano infatti gli slot in cui possono essere decodificati pacchetti “ripuliti” da interferenze di altri pacchetti. Il primo sum node ha grado 1, dal quale ricaviamo il primo pacchetto, e rimuoviamo la sua copia nel terzo slot. Con la terza iterazione si ricava il terzo pacchetto, che ha una copia nel terzo sum node di grado 1. Rimuovendo il contributo del pacchetto appena decodificato dal quarto slot, si rende il quarto sum node di grado 1, permettendo così la decodifica dell’ultimo pacchetto durante la quarta iterazione.

La distribuzione del grado di un burst node (burst node degree distribution) è definita da  $\{\Lambda_l\}$ , dove  $\Lambda_l$  descrive la probabilità che un burst node possenga  $l$  connessioni. La rappresentazione polinomiale della burst node degree distribution è data da:

$$\Lambda(x) = \sum_l \Lambda_l x^l \quad (3.1)$$

Analogamente, la distribuzione del grado di un sum node (sum node degree distribution) è rappresentata da  $\{\Psi_l\}$ , dove  $\Psi_l$  è la probabilità che un sum node abbia  $l$  archi incidenti. La rappresentazione polinomiale della sum node degree distribution è data da:

$$\Psi(x) = \sum_l \Psi_l x^l \quad (3.2)$$

Si osserva che, in accordo con le definizioni adottate nella descrizione del grafo,  $\Lambda_l$  è anche la probabilità che un pacchetto sia ripetuto  $l$  volte, invece  $\Psi_l$  rappresenta la probabilità che  $l$  copie di pacchetti siano trasmessi nello stesso slot. La distribuzione  $\Lambda(x)$  è il parametro sotto il controllo del progettista del sistema,

### 3. EVOLUZIONE DI FRAMED SLOTTED ALOHA: I PROTOCOLLI CRDSA E IRSA

---

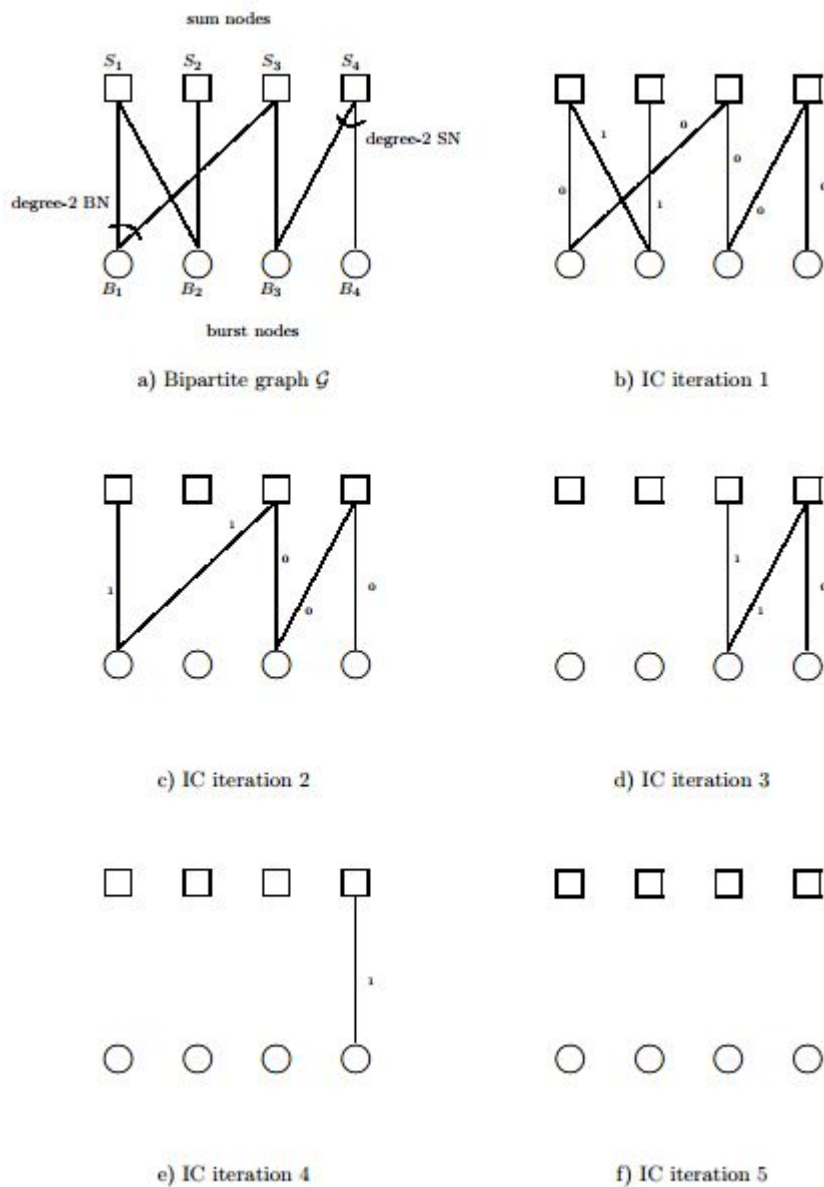


Figura 3.2: Avanzamento del processo di Interference Cancellation: ogni Figura rappresenta un passo iterativo [5].

mentre  $\Psi(x)$  si dimostra essere un parametro dipendente dal traffico  $G$  del sistema e dalla media del rate di ripetizione dei pacchetti, che vale  $\sum_l l\Lambda_l = \Lambda'(1)$ , mentre il numero medio di collisioni per slot è  $\sum_l l\Psi_l = \Psi'(1)$ . La distribuzione del grado può essere definita anche dalla prospettiva degli archi. Si definisce  $\lambda_l$  come la probabilità che un arco sia connesso a un burst node di grado  $l$ . Allo stesso modo  $\rho_l$  definisce la probabilità che un arco incida un sum node di grado  $l$ . Questi termini caratterizzeranno il limite asintotico del throughput presentato in seguito.

### 3.4 Prestazioni asintotiche

Attraverso un'analisi delle statistiche sopra introdotte, si dimostra [5] che per valori del traffico  $G$  inferiori ad una certa soglia  $G^*$ , il meccanismo di Interference Cancellation riesce a recuperare i pacchetti con probabilità vicina a 1. Per valori superiori alla soglia  $G^*$  invece la probabilità di recuperare pacchetti diminuisce tendendo a 0. La soglia  $G^*$  viene stimata attraverso un suo limite superiore, che vale:

$$G^* \leq \frac{1}{\lambda_2 \Lambda'(1)} \quad (3.3)$$

L'obiettivo del progettista del sistema è quindi quello di trovare una distribuzione  $\Lambda(x)$  tale da rendere il limite superiore della soglia  $G^*$  più alto possibile. In Figura 3.3 sono riportati alcune distribuzioni e il throughput asintotico da loro raggiunto. Per un sistema CRDSA, una distribuzione  $\Lambda(x) = x^2$  (ogni pacchetto è ripetuto 2 volte) si traduce in una soglia  $G^* = 0.5$ . Per quanto riguarda i sistemi IRSA, per prima cosa si fissa il grado massimo dei burst node: esso è fissato a 8 per limitare il numero di puntatori all'interno dello header del pacchetto alle altre repliche nel frame. Per ottenere una distribuzione ottimizzata si fa ricorso a tecniche di evoluzione differenziale [5]. Con la distribuzione  $\Lambda(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$  si ottiene una soglia  $G^* = 0.938$ . In accordo con la definizione di  $G^*$ , tutto il traffico offerto  $G$  inferiore a  $G^* = 0.938$  si trasforma in throughput utile. Per un sistema CRDSA invece ciò è valido solo per  $G \leq 0.5$ . In tabella sono riportate alcune distribuzioni del grado dei burst node, ottenute sempre per evoluzione differenziale. Si osserva che concedendo un numero maggiore di ripetizioni, si ottengono

### 3. EVOLUZIONE DI FRAMED SLOTTED ALOHA: I PROTOCOLLI CRDSA E IRSA

---

soglie vicine a 1: per esempio fissato il rate di ritrasmissioni a 16, è possibile ottenere  $G^*=0.965$ . Limitando il rate di ritrasmissioni, la soglia cala: con un rate di ritrasmissione di 4,  $G^*$  vale 0.868. Per sottolineare la bontà dell'intuizione di adottare rate di ritrasmissione variabili, confrontiamo l'ultimo risultato con una distribuzione del rate di ritrasmissione costante 4 ( $\Lambda(x) = x^4$ ). Si ottiene una soglia  $G^*=0.772$  che confrontata con la soglia data dalla distribuzione irregolare (0.868) risulta inferiore di circa il 12%. La percentuale aumenta fino al 28% se si confrontano due distribuzioni con rate massimo pari a 5, una con rate costante ( $G^* = 0.701$ ) e l'altra con rate variabile ( $G^* = 0.898$ ).

Distribution, $\Lambda(x)$	$G^*$
$0.5102x^2 + 0.4898x^4$	0.868
$0.5631x^2 + 0.0436x^3 + 0.3933x^5$	0.898
$0.5465x^2 + 0.1623x^3 + 0.2912x^6$	0.915
$0.5x^2 + 0.28x^3 + 0.22x^8$	0.938
$0.4977x^2 + 0.2207x^3 + 0.0381x^4 + 0.0756x^5 + 0.0398x^6 + 0.0009x^7 + 0.0088x^8 + 0.0068x^9 + 0.0030x^{11} + 0.0429x^{14} + 0.0081x^{15} + 0.0576x^{16}$	0.965

Figura 3.3: Le varie distribuzioni comportano diverse soglie, da[5].

## 3.5 Simulazioni e risultati numerici

I grafici che seguono sono il risultato di due tipi di simulazioni. Il primo tipo di simulazioni tiene conto unicamente del MAC layer, e si assume che un pacchetto venga recuperato se e solo se non c'è alcuna collisione con altri pacchetti nel corrispondente slot. Nel secondo tipo di simulazioni viene anche simulato l'impatto del physical layer. I risultati delle simulazioni sono riportati nei grafici delle figure 3.4 e 3.5, dove si osserva che la simulazione del MAC layer si avvicina molto ai risultati ottenuti con una simulazione più complessa che tenga conto anche del mezzo fisico. Per la simulazione del protocollo IRSA è stata adottata la distribuzione  $\Lambda(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$  già vista in precedenza. La prima serie di simulazioni è ottenuta comparando vari sistemi a parità di numero di slot per

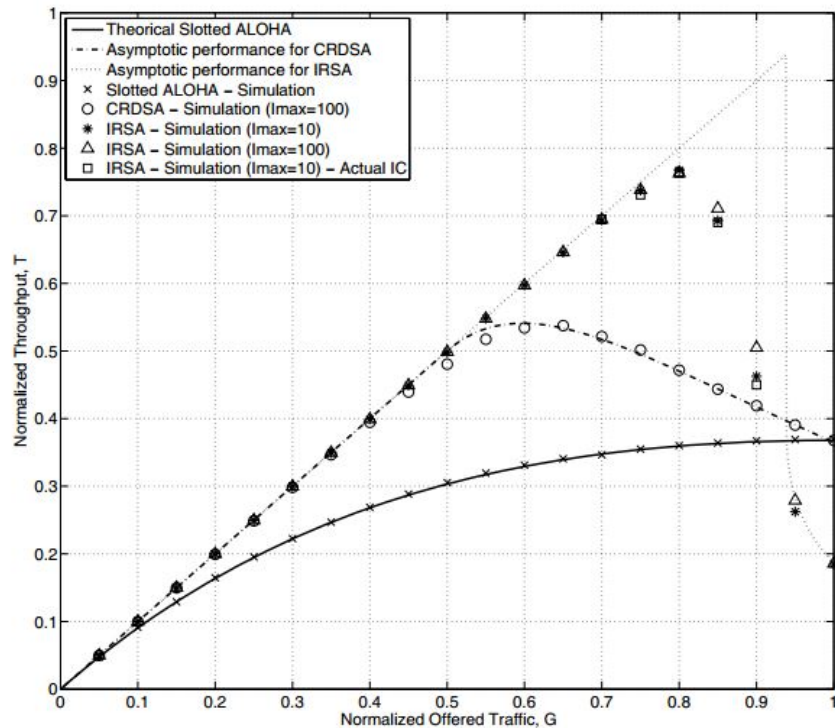


Figura 3.4: throughput asintotico e simulato di SA, CRDSA e IRSA,  $n=200$ . I risultati ottenuti simulando anche il canale fisico sono segnati in '□', e sono riferiti ad un SNR di 2dB [5].

frame, con  $n = 200$  slot. In Figura 3.4 è graficato l'andamento del throughput  $T$  del protocollo Slotted Aloha e l'andamento asintotico di CRDSA e IRSA, per valori di  $G$  compresi in  $[0, 1]$ , variando il numero massimo di iterazioni del processo di Interference Cancellation  $I_{MAX}$ . Guardando i risultati della simulazione, si osserva che per  $I_{MAX} = 100$  iterazioni, IRSA raggiunge un throughput vicino a 0.8, mentre CRDSA non oltrepassa un throughput di 0.55. Per  $I_{MAX} = 10$  IRSA mostra un calo di prestazioni in termini di throughput. Si osserva che la relazione  $T/G$  è lineare sotto  $T = 0.7$ , cioè fino a questo valore la maggior parte del traffico diventa throughput. Per valori di  $G \rightarrow 1$  invece le prestazioni subiscono un brusco peggioramento. Fintanto che  $G$  è minore di  $G^*$ , il vantaggio prestazionale che hanno CRDSA e IRSA nei confronti di Slotted Aloha è evidente, ma per valori di traffico vicini a 1 Slotted Aloha supera in prestazioni i sistemi basati sul recupero di pacchetti (sempre tenendo conto che per  $G > 1$  Slotted Aloha diventa instabile). Ciò è causato dal funzionamento del sistema di Interference Cancellation: per valori di traffico sotto la soglia  $G^*$  il processo iterativo riesce

### 3. EVOLUZIONE DI FRAMED SLOTTED ALOHA: I PROTOCOLLI CRDSA E IRSA

---

a recuperare bene i pacchetti, ma quando il traffico supera la soglia il processo di Interference Cancellation “si ingolfa” e il numero di repliche dei pacchetti per frame supera di molto il numero di pacchetti dello schema Slotted Aloha, comportando un alta percentuale di pacchetti persi. Si osserva inoltre che con  $n = 200$  slot, IRSA raggiunge un picco di throughput minore di 0.8, quando nell’analisi di Interference Cancellation con grafo bipartito era stato predetto un picco maggiore ( $T=0.938$ ). D’altronde è anche ragionevole pensare che le prestazioni asintotiche siano raggiunte per frame di grandi dimensioni. In Figura 3.5 è mostrato l’andamento di IRSA per frame di grandezze differenti. Maggiori sono le dimensioni del frame, maggiore è il throughput raggiunto. Per quanto riguarda CRDSA invece sembra esserci minore dipendenza delle prestazioni dalla dimensione del frame, e la varianza delle curve per valori differenti di  $n$  è trascurabile.

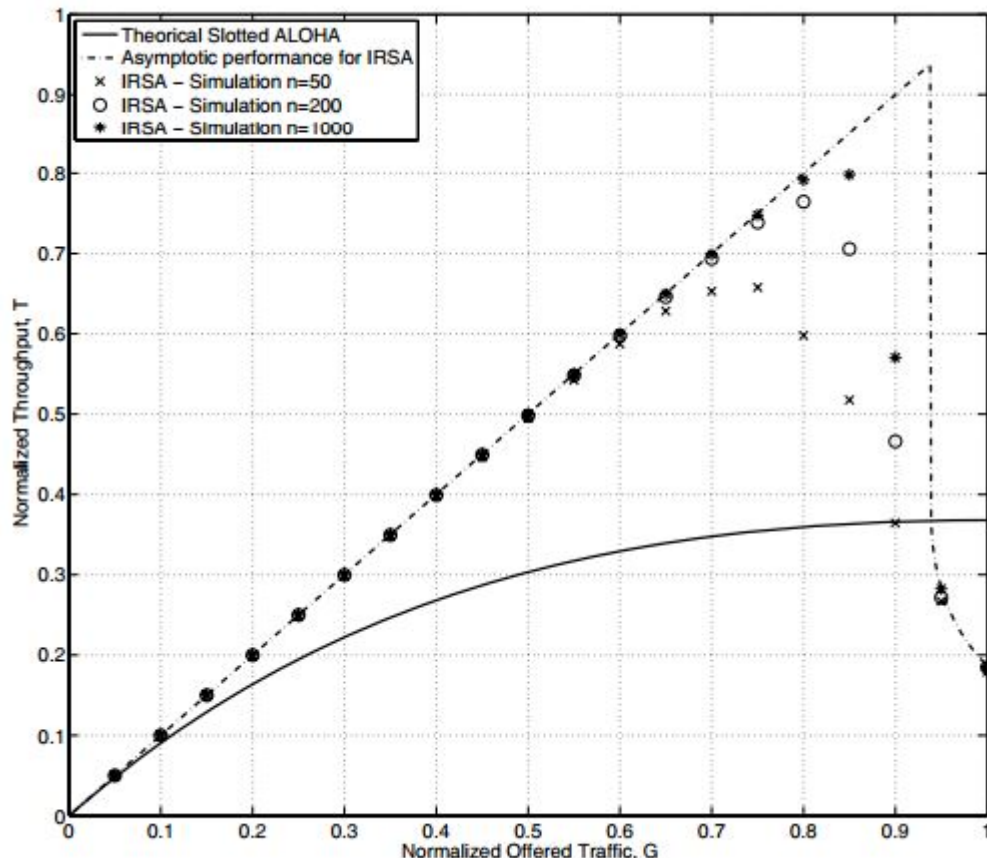


Figura 3.5: Trought asintotico e simulato per SA e IRSA al variare di  $n$ .  $I_{MAX} = 20$  [5].

# Capitolo 4

## Frameless Aloha

Frameless Aloha [6] costituisce un ulteriore sviluppo all'interno della famiglia dei protocolli Aloha. L'idea alla base è quella di implementare in un protocollo di accesso al mezzo la filosofia dei codici “*rateless*”: essi sono codici in cui l'algoritmo di codifica non produce parole di codice di lunghezza fissata a priori, potendo idealmente produrre un numero infinito di *codeword* per uno stesso messaggio. Analogamente, in Frameless Aloha la durata del periodo di contesa del mezzo tra gli utenti (cioè la lunghezza del frame) non è costante, ma dura finché il ricevitore decide che una frazione sufficientemente alta di trasmissioni degli utenti è stata correttamente decifrata. Nel prossimo paragrafo verranno introdotti i codici frameless, e successivamente il protocollo Frameless Aloha.

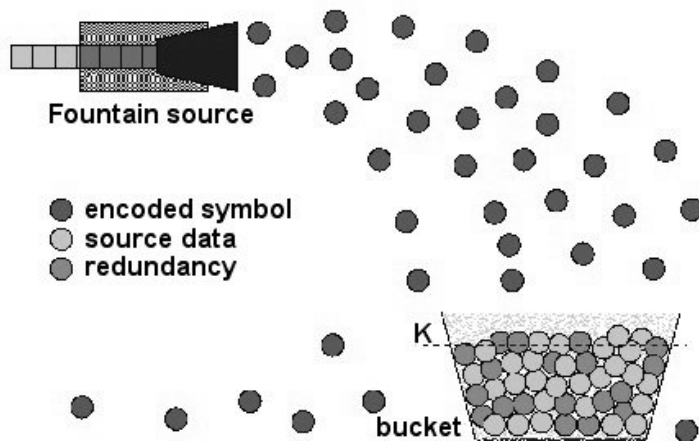
### 4.1 Presentazione dei frameless code: codici LT

*Erasure channel* (canale a cancellazione) è un modello di canale in cui i pacchetti inviati possono o essere ricevuti correttamente senza errori, oppure non ricevuti. I codici “*erasure*” lavorano in canali modellati in questo modo e codificano un messaggio di  $k$  pacchetti in un altro lungo  $n$  pacchetti. La decodifica avviene con successo se al ricevitore giunge un certo numero  $k^*$  pacchetti tra gli  $n$  inviati, con  $k < k^* < n$ . I codici rateless sono una classe di codici “*erasure*” caratterizzata dal fatto che  $n$  non è costante, ma varia in relazione ad una certa distribuzione di probabilità. Essi sono chiamati anche codici “a fontana” (*fountaincode*): il codificatore è paragonato ad una fontana che produce una infinita cascata di

#### 4. FRAMELESS ALOHA

---

gocce d'acqua (ossia di pacchetti codificati); il decodificatore invece è un secchio che raccoglie le gocce fino a quando è riempito.



I codici LT (Luby Transform code) [10] sono un tipo di codici rateless. La codifica prevede questi passi [8]:

- il messaggio è diviso in  $k$  pacchetti  $s_1, s_2, \dots, s_k$ ;
- tramite una funzione di distribuzione, viene scelto casualmente il grado  $d_n$  del nuovo pacchetto codificato  $t_n$ ;
- $d_n$  distinti pacchetti vengono scelti casualmente e uniformemente tra  $s_1, s_2, \dots, s_k$ . Viene creato un pacchetto codificato eseguendo l'operazione XOR tra tutti i  $d_n$  pacchetti scelti:  $t_n = s_1 \oplus s_2 \oplus \dots \oplus s_{d_n}$ ;
- $t_n$  viene inviato, e si procede alla creazione di un nuovo pacchetto  $t_{n+1}$  fino a quando il ricevitore segnala di aver codificato correttamente tutto il messaggio.

La decodifica è così descritta, tenendo conto che l'operazione di confronto menzionata è lo XOR tra due pacchetti:

- si creano due buffer, uno contenente i singoli pacchetti decodificati, l'altro contenente i pacchetti codificati di grado  $> 1$ ;
- se viene ricevuto un pacchetto codificato di grado  $d_n > 1$ , viene confrontato con tutti i pacchetti già decodificati per cercare di ridurre il grado, e se il grado ridotto è  $> 1$  viene conservato nel buffer assieme agli altri pacchetti di grado non unitario;



- ogni volta che si riceve un pacchetto codificato di grado 1 (o perché appena ricevuto o perché il grado di un pacchetto si è ridotto a 1 in seguito di una operazione XOR), esso viene prima accodato insieme agli altri pacchetti decodificati di grado 1 e quindi viene confrontato coi pacchetti di grado  $> 1$  presenti nel buffer.

Quando tutti i  $k$  pacchetti del messaggio originale sono stati decodificati, il ricevitore segnala di aver ricevuto correttamente il messaggio.

L'operazione di codifica definisce un grafo bipartito che connette i pacchetti codificati ai pacchetti del messaggio sorgente. L'esempio illustrato in Figura 4.1 in cui i pacchetti per semplicità hanno lunghezza di 1 bit descrive l'operazione di decodifica.

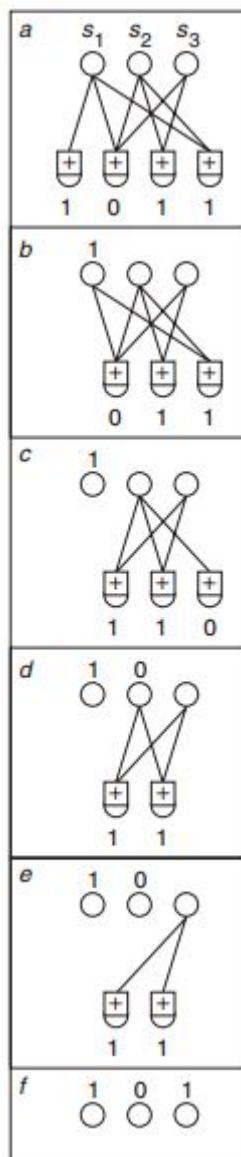


Figura 4.1: I tre pacchetti di sorgente rappresentati da cerchi (source node) sono codificati in 4 pacchetti, rappresentati dai blocchi in basso (check node), e valgono  $t_1, t_2, t_3, t_4 = 1011$  all'inizio dell'algoritmo. Alla prima iterazione il solo check node connesso ad un unico source node è il primo. Il bit  $s_1$  viene impostato di conseguenza a 1, il check node viene scartato, quindi il valore di  $s_1$  (1) è sommato al valore del check node cui è connesso (0).

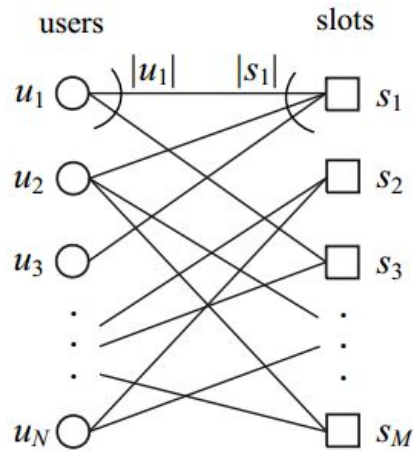


Figura 4.2: In Figura è descritto un frame per mezzo della rappresentazione con grafo bipartito. Agli utenti corrispondono i nodi di sinistra definiti *user node*, agli slot i nodi di destra definiti *slot node*, mentre gli archi indicano le ritrasmissioni.

## 4.2 Frameless Aloha: modello del sistema

Si assume che ci siano  $N$  utenti a contendersi il canale. L'inizio e la fine del periodo di contesa sono stabiliti attraverso un segnale inviato dal ricevitore. Il periodo di contesa è suddiviso in slot di uguale durata, e gli utenti sono sincronizzati con gli slot. Inoltre il canale è assunto non rumoroso. Il segnale ricevuto nello slot  $j$ -esimo è  $Y_j = \sum_{i \in A_j} X_j$ , con  $A_j$  che è l'insieme degli utenti che stanno trasmettendo nello slot  $j$ . Analogamente a IRSA, in un periodo di contesa, ogni utente può trasmettere un unico pacchetto, ripetuto su più slot, secondo una opportuna distribuzione statistica. Il numero di slot in un periodo di contesa è  $M$ : esso non è conosciuto a priori e viene stabilito solo quando il periodo di contesa è terminato, ovvero dopo la segnalazione del ricevitore. In ciascun pacchetto trasmesso è contenuto un riferimento a tutte le copie di quel pacchetto inviate durante il periodo di contesa. Il ricevitore può distinguere quali sono gli slot liberi, quelli contenenti un solo pacchetto e quelli collisi, ma in caso di collisione non può determinare il numero di pacchetti in uno slot, detto anche *grado* di collisione dello slot. Le collisioni vengono risolte attraverso il processo di Interference Cancellation. Sia  $R$  il numero di pacchetti decifrati, quando la frazione di pacchetti risolti  $R/N$  (con  $N$  noto dal ricevitore), raggiunge la soglia prefissata, il ricevitore invia il segnale di terminazione del periodo di contesa.

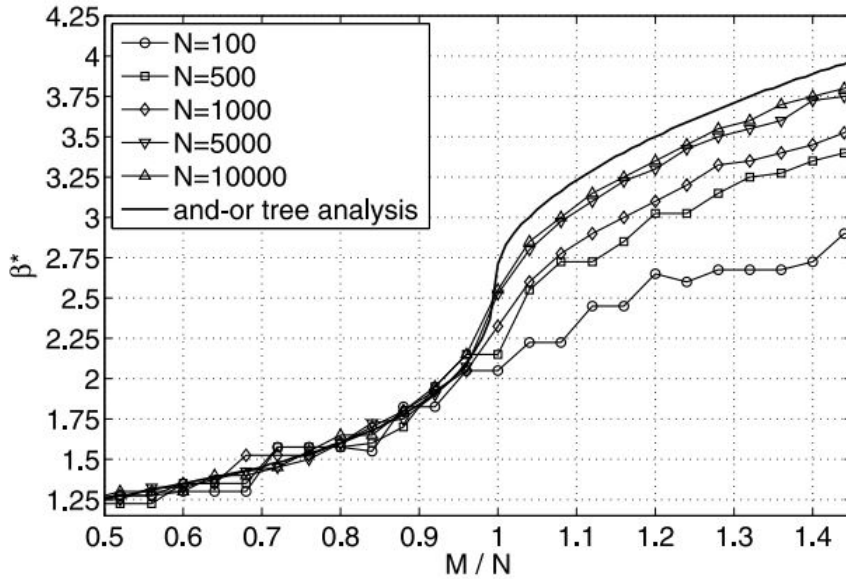


Figura 4.3: media del grado ottimo  $\beta^*$  che massimizza il throughput, in funzione di  $M/N$  [6].

### 4.3 Analisi e prestazioni

Per decodificare tutti i pacchetti, il numero totale di slot deve essere non minore del numero di utenti che trasmettono nello stesso periodo di contesa, cioè  $M = (1 + \epsilon)N$ ,  $\epsilon > 0$ . I parametri di interesse per l'analisi delle prestazioni di Frameless Aloha sono la probabilità di risoluzione di un pacchetto  $P_R$  e il throughput normalizzato  $T$ , definito come la probabilità di risoluzione di un pacchetto per slot. Ne segue che:

$$T = \frac{P_R}{M/N} = \frac{P_R}{(1 + \epsilon)} \quad (4.1)$$

$P_R$  e  $T$  sono funzioni sia della media del grado degli slot node, detta  $G$ , sia del numero  $M$  di slot che formano il periodo di contesa. Al fine di massimizzare  $T$ , occorre trovare i valori di  $G$  e  $M$  che massimizzano  $P_R$ , definiti rispettivamente  $\beta^*$  e  $M^*$ . Il loro valore viene ricavato grazie ad analisi basate sugli alberi "and-or" [6].

In seguito sono riportati risultati ottenuti sia attraverso l'analisi basata sia sugli alberi and-or sia attraverso simulazioni. Per semplicità si è assunto che la media del grado degli slot  $G$  fosse costante. Inoltre il sistema simulato teneva conto unicamente del Mac layer, senza considerare gli effetti di un canale rumoroso.

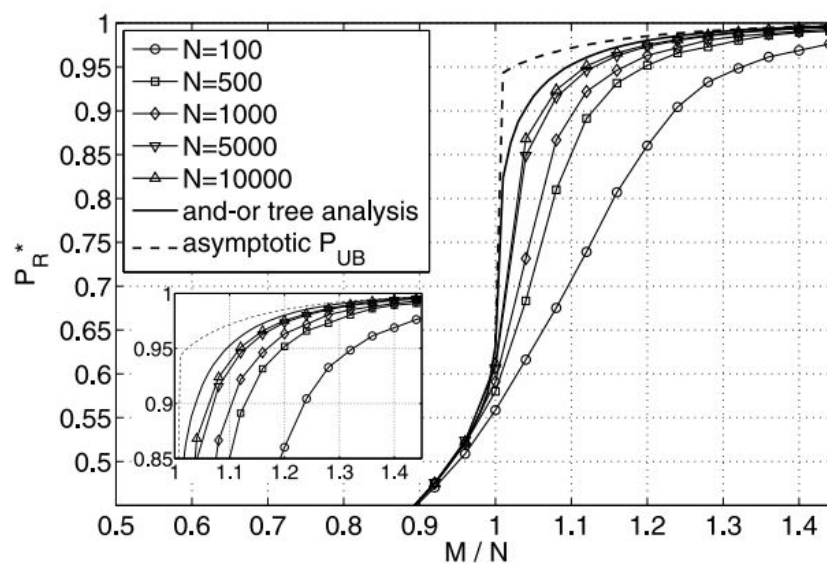
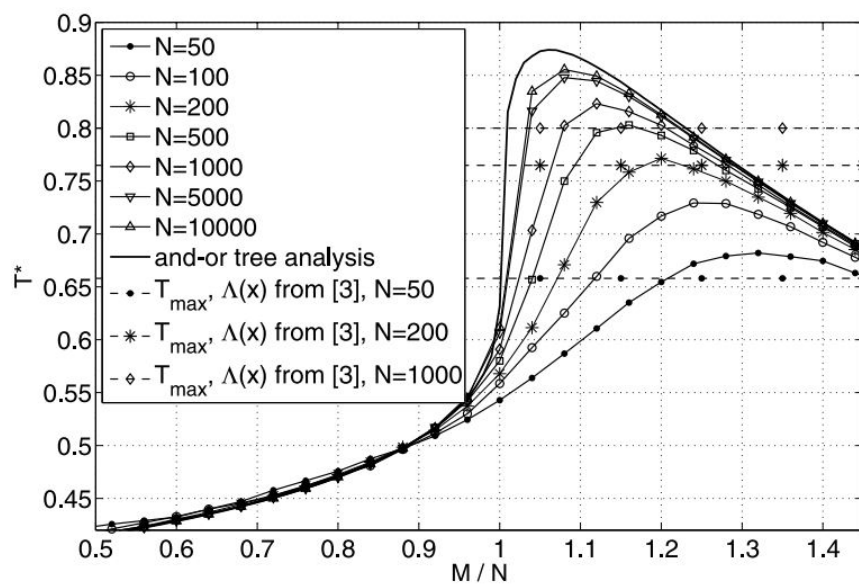


Figura 4.4: Massima probabilità di risoluzione dei pacchetti  $P_R$ , in funzione di  $M/N$  [6].

In Figura 4.3 è mostrato l'andamento di  $\beta^*$  che massimizza  $P_R$ . Il valore ottimale della media del grado degli slot cresce costantemente al crescere di  $M/N$ . Figura 4.4 presenta la massima  $P_R$  in funzione di  $M/N$ . Maggiore è il numero di utenti  $N$ , tanto più ci si avvicina al limite dedotto dall'analisi basata sugli alberi and-or. In Figura 4.5 infine è graficato il massimo throughput  $T^*$  corrispondente alla  $\beta^*$  di Figura 4.3, ovvero il throughput ottenuto per  $\beta^*$  in funzione di  $M/N$ .  $T^*$  segue l'andamento di  $\beta^*$  e raggiunge il picco per valori di  $M/N$  maggiori di 1. In seguito, poichè  $\beta^*$  satura e  $M/N$  aumenta,  $T$ , essendo dipendente dal loro rapporto, cala. Il massimo throughput ottenuto è circa 0.87, raggiunto per grandi valori di  $N$ . In Figura sono anche presenti i valori di picco del throughput di un sistema IRSA per vari valori di  $N$ .

Figura 4.5: throughput massimizzato  $T^*$  in funzione di  $M/N$  [6].

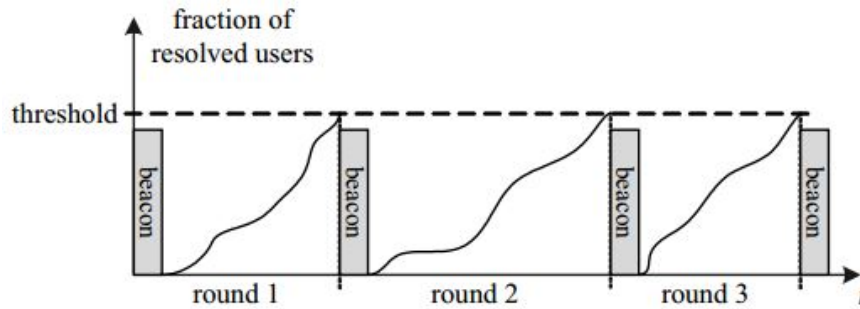


Figura 4.6: Rappresentazione dell'andamento dei pacchetti risolti all'interno dei periodi di contesa. Raggiunta la soglia, viene inviato il segnale di fine periodo [6]

Si sottolinea che un passaggio chiave per ottenere così alti valori di throughput è la scelta di terminare il periodo di contesa dopo aver superato una certa soglia di pacchetti decodificati, e non dopo aver decodificato tutti i pacchetti trasmessi in quel periodo di contesa. Per quanto riguarda il contesto dei codici rateless, la scelta naturale è quella di terminare il flusso di pacchetti codificati trasmessi dalla sorgente quando è avvenuta la decodifica di tutti i pacchetti. Tuttavia implementare un criterio analogo in Frameless Aloha comporterebbe un uso inefficiente degli slot. Infatti, attraverso l'analisi del grafo bipartito, si osserva che la probabilità che un user node abbia grado 0 (cioè che non abbia trasmesso il pacchetto in nessuno slot) vale [7]:

$$P[|u| = 0] = e^{-\frac{M}{N}G} \quad (4.2)$$

Essa quindi decresce esponenzialmente con  $M$ , il numero di slot. Ciò implica che la probabilità che un utente non trasmetta il pacchetto, e quindi la probabilità che quel pacchetto non abbia la possibilità di venire decodificato, decresce all'aumentare di  $M$ .

# Capitolo 5

## Conclusioni

A partire dalla loro introduzione negli anni '70, i protocolli di accesso al mezzo della famiglia Aloha si sono sempre più evoluti. L'aspetto determinante da analizzare al fine di ottenere buone prestazioni in questo tipo di protocolli è la gestione delle collisioni dei pacchetti inviati da utenti diversi, inevitabili in sistemi privi di controllore e fortemente distribuiti.

il protocollo Aloha originale in caso di collisione prevede la ritrasmissione dei pacchetti collisi dopo un certo tempo casuale di backoff. Slotted Aloha introduce una sincronizzazione tra gli utenti suddividendo in slot la linea temporale, così che un pacchetto possa essere trasmesso unicamente all'inizio di uno slot: ciò garantisce un throughput doppio rispetto ad Aloha puro. Framed Slotted Aloha è una variante di Slotted aloha in cui gli slot sono organizzati in frame.

Contention Resolution Diversity Slotted Aloha (CRDSA) introduce la ritrasmissione dello stesso pacchetto all'interno del frame. Il processo iterativo di Interference Cancellation recupera poi i pacchetti collisi all'interno del frame.

Irregular Repetition Slotted Aloha (IRSA) è un'evoluzione di CRDSA in cui i pacchetti vengono ritrasmessi sugli slot di un frame secondo una data distribuzione di probabilità e non più con una frequenza di ritrasmissioni costante.

Infine Frameless Aloha è un protocollo di accesso al mezzo che condivide la filosofia dei codici "rateless": la lunghezza del frame non è stabilita a priori, ma nuovi slot sono aggiunti al frame finché il ricevitore decide che è stato ricevuto correttamente un sufficiente numero di pacchetti. IRSA e Frameless Aloha raggiungono asintoticamente valori di throughput vicini a 1, offrendo prestazioni incredibilmente migliori rispetto allo schema Aloha originale. L'analisi e le si-



mulazioni mostrano che il protocollo IRSA raggiunge le migliori prestazioni per valori di traffico normalizzato minori di 1, mentre Frameless Aloha raggiunge il picco di throughput in condizioni di alto traffico.

# Bibliografia

- [1] N. Benvenuto, M. Zorzi, *Principles of Communications Networks and Systems*, Wiley, 2011.
- [2] M. Luby, *LT codes*, Proc. 43rd Annual IEEE Symp. Foundations Computer Science, Vancouver, Canada, Nov. 2002.
- [3] N. Abramson, *The ALOHA System - Another Alternative for Computer Communications*, Proc. 1970 Fall Joint Computer Conference. AFIPS Press.
- [4] L. G. Roberts, *ALOHA packet systems with and without slots and capture*, ARPANET System Note 8 (NIC11290), June 1972.
- [5] Gianluigi Liva, *Graph-Based Analysis and Optimization of Contention Resolution Diversity Slotted ALOHA*, IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 59, NO. 2, FEBRUARY 2011.
- [6] C. Stefanovic, P. Popovski, and D. Vukobratovic, “*Frameless ALOHA protocol for Wireless Networks*”, IEEE Comm. Letters, vol. 16, pp. 2087–2090, Dec. 2012.
- [7] P. Popovski, C. Stefanovic, J. H. Sørensen, *Random access protocols based on rateless codes*, Aalborg University, Department of Electronic Systems.
- [8] D.J.C. MacKay, *Fountain codes*, E Proc.-Commun., Vol. 152, No. 6, December 2005.
- [9] E. Casini, R. De Gaudenzi, O. del Rio Herrero, *Contention resolution diversity slotted ALOHA (CRDSA): an enhanced randomaccess scheme for satellite access packet networks*, IEEE Trans. Wireless Commun., vol. 6, no. 4, pp. 1408-1419, Apr. 2007.