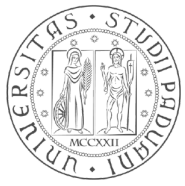

Università degli Studi di Padova

Facoltà di Ingegneria

Corso di Laurea in ingegneria dell'informazione

Valutazione del ritardo di accodamento in schemi di trasmissione ARQ

Laureando

Giovanni Pilon

Relatore

Leonardo Badia

Anno Accademico 2012/2013

Indice

Sommario

Questa tesi propone uno studio dei tempi di ritardo delle tecniche di controllo d'errore ARQ che vengono usate nella codifica di canale, vengono presentati i vari approcci che sono usati per questo tipo di analisi e si introducono le catene di Markov che permettono di condurre uno studio più accurato dei parametri d'interesse. In particolare, utilizzando Matlab, si sfruttano le catene di Markov per analizzare i tempi di accodamento di un ideal SR-ARQ e si mostra come è possibile estendere l'analisi ad altri tipi di ARQ.

Capitolo 1

Introduzione

1.1 La codifica di canale

I sistemi di comunicazione odierni si basano sulla commutazione a pacchetto. Il contenuto digitale viene diviso in elementi atomici di lunghezza relativamente piccola chiamati pacchetti che vengono trasmessi attraverso il canale utilizzando tecniche di modulazione digitale. Al livello fisico il canale di trasmissione introduce errori a causa dell'attenuazione del segnale combinata con la presenza di rumore o a causa di trasmissioni contemporenee [18] ed è quindi possibile che i pacchetti vengano interpretati in modo sbagliato dal ricevitore.

La codifica di canale è l'insieme di tecniche che vengono utilizzate per risolvere questo problema, si basa sull'aggiunta di ridondanza ai pacchetti di informazione: il pacchetto originale b viene trasformato in un pacchetto più grande c dove sono stati inseriti dei simboli aggiuntivi che consentono di aumentare la robustezza contro gli errori [10]. In questo modo il ricevitore conosce a priori l'insieme C dei possibili messaggi d'informazione e quando viene ricevuto un pacchetto d che non appartiene a C può cercare di correggerlo o può richiederne la ritrasmissione; chiameremo ε la probabilità che si verifichi un errore e che quindi

il pacchetto venga identificato come errato dal ricevitore.

Al livello data-link le possibili scelte che un ricevitore può compiere quando ha a che fare con un pacchetto non valido sono:

- FEC: Forward error correction, che si basa sul correggere i pacchetti errati scegliendo il pacchetto più verosimile tramite varie tecniche.
- ARQ: Automatic repeat request, che si basa sulla richiesta di ritrasmissione dei pacchetti errati, a differenza della FEC richiede un canale di ritorno dove vengono inviate le richieste di ritrasmissione.
- Tecniche ibride, note come HARQ, che sono una combinazione di FEC e ARQ. In genere effettuano la correzione degli errori più probabili (cioè più frequenti) e chiedono la ritrasmissione dei restanti.

La FEC ha il vantaggio di aumentare il tempo di trasmissione in modo deterministico in quanto ogni pacchetto una volta ricevuto viene identificato in modo univoco e viene passato ai layer successivi, se la decodifica è sbagliata il pacchetto viene identificato in modo sbagliato ma il ricevitore passa comunque il pacchetto ai layer successivi. Al contrario l'ARQ consente di passare sempre pacchetti corretti ai layer successivi al costo di avere un tempo di ritardo variabile dovuto al tempo perso per le ritrasmissioni, il pacchetto viene ritrasmesso fino a quando non viene ricevuto correttamente.
Per questi motivi la FEC viene impiegata in trasmissioni di tipo broadcast o applicazioni real-time mentre l'ARQ è usato per le trasmissioni di dati in cui i vincoli sui tempi di ritardo non sono stringenti (ad esempio nel transmission control protocol,

TCP) e diventa più efficiente recuperare i pacchetti persi tramite ritrasmissione anziché usare la FEC.

I protocolli di trasmissione effettuano la correzione su più livelli (data-link layer, transport layer, application), tuttavia i tempi di ritardo e le probabilità d'errore dei layer successivi sono strettamente legati alle prestazioni dell'ARQ, per questo motivo lo studio dei tempi di consegna del ARQ è di importanza cruciale per capire la correlazione fra le prestazioni dei layer successivi e quelle del data-link layer [4].

In questa tesi è stato analizzato un ideal SR-ARQ, è stata calcolata la distribuzione statistica del numero di pacchetti in coda e del tempo di accodamento che un pacchetto trova arrivando al trasmettitore. Per calcolare queste distribuzioni l'ideal SR-ARQ è stato modellato come una catena di Markov a tempo discreto ed è stato quindi implementato in Matlab. Sono stati fatti confronti grafici fra le distribuzioni al variare delle ipotesi sul tipo di arrivi (tasso di arrivo Bernoulliano o tasso di arrivo Poissoniano).
I risultati ottenuti mostrano che la distribuzione del tempo di accodamento che un pacchetto deve affrontare segue l'andamento della distribuzione del numero dei pacchetti in coda. Il confronto fra i due diversi tassi di arrivo mostra che un tasso di arrivi Poissoniano dà un maggior numero di pacchetti in coda rispetto al tasso di arrivi Bernoulliano, ma le distribuzioni sono simili se le probabilità di arrivo non sono elevate.

Il resto della tesi è organizzato in questo modo: nel capitolo 2 vengono presentate le principali meccaniche di implementazione dei sistemi ARQ e le assunzioni semplificative che possono essere utilizzate per analizzare i tempi di ritardo. Vengono quindi introdotte le catene di Markov. Nel capitolo

3 viene spiegato l'approccio utilizzato per modellare un ideal SR-ARQ come una catena di Markov e si delinea il procedimento necessario per implementare in Matlab l'analisi matematica. Si spiega poi come è possibile generalizzare l'analisi se si modificano le assunzioni. Nel capitolo 4 vengono presentati i risultati grafici ottenuti. Seguono le conclusioni nel capitolo 5.

Capitolo 2

Panoramica ARQ

2.1 ARQ, un'introduzione

I protocolli ARQ si basano sul presupposto che il ricevitore possa mandare messaggi di acknowledgment (riconoscimento) relativi ai pacchetti mandati dal trasmettitore. Se il pacchetto ricevuto è errato viene mandato un segnale di negative acknowledgment (riconoscimento negativo), detto NACK, se invece è corretto viene mandato un segnale di positive acknowledgment (riconoscimento positivo), detto ACK, quando il trasmettitore riceve un NACK deve ritrasmettere il pacchetto corrispondente. Ad ogni pacchetto, detto anche PDU (packet data unit), viene associato un sequence number (SN) che viene aggiunto al pacchetto in modo da poter riconoscere i pacchetti e sapere di quali bisogna effettuare una ritrasmissione. Si definisce round-trip time (t_{RTT}) il tempo trascorso fra l'inizio della trasmissione di un pacchetto e l'istante in cui il corrispondente messaggio di acknowledgment è completamente ricevuto dal trasmettitore, il t_{RTT} è dato dalla somma di tempo di trasmissione del pacchetto (t_p), tempo di trasmissione del ACK/NACK, tempi di propagazione e tempi di processazione. Si definisce window size m del ARQ il numero massimo di pacchetti che si possono trovare nel canale senza che si abbia ancora ricevuto acknowledgemnt per nessuno di essi. Per

analizzare i tempi è utile dividere il t_{RTT} in intervalli, con ogni intervallo lungo quanto il tempo di trasmissione t_p, in questo modo m risulta essere uguale al minimo numero di intervalli che supera il t_{RTT}.

Esistono principalmente tre diverse meccaniche per implementare le tecniche ARQ a seconda del modo in cui vengono realizzate le trasmissioni e ritrasmissioni [18]:

- Stop and Wait ARQ (SW-ARQ): Il trasmettitore invia un pacchetto ad ogni t_{RTT} e aspetta di ricevere il messaggio di acknowledgment prima di procedere con le trasmissioni successive. Se riceve un NACK ritrasmette il pacchetto precedentemente inviato, se riceve un ACK procede con la trasmissione del pacchetto successivo. Per t_{RTT} grandi questa tecnica diventa poco efficace in quanto per buona parte del tempo (cioè per $t_{RTT} - t_p$) non si effettua nessuna trasmissione.

- Go Back N ARQ (GBN-ARQ): Il trasmettitore trasmette pacchetti continuamente, in particolare trasmette m pacchetti consecutivi in ogni t_{RTT}, in questo modo il canale di trasmissione viene sempre utilizzato per trasmettere qualcosa. Quando viene ricevuto un ACK il trasmettitore continua a trasmettere pacchetti. Quando viene ricevuto un NACK il trasmettitore ricomincia a ritrasmettere pacchetti tornando indietro di m passi, vale a dire che ritrasmettere il pacchetto relativo al NACK e poi continua la ritrasmissione da lì. Il limite di questa implementazione è che tornando indietro di m passi si ritrasmettono anche i pacchetti che erano già stati ricevuti correttamente al ricevitore.

- Selective Repeat ARQ (SR-ARQ): Il trasmettitore trasmette continuamente pacchetti come nel GBN-ARQ ma quando

viene ricevuto un NACK viene ritrasmesso solamente il pacchetto relativo al NACK e poi si riprende la trasmissione dal pacchetto inviato.

Il SR-ARQ è la più efficace implementazione perché in ogni momento si usa il canale per trasmettere informazioni utili, vale a dire o per la trasmissione di un nuovo pacchetto o per la ritrasmissione di un pacchetto non ricevuto. Al contrario nel SW-ARQ ci sono momenti in cui il canale non viene utilizzato perché si sta aspettando l'acknowledgment e nel GBN-ARQ può capitare di ritrasmettere pacchetti anche se erano stati ricevuti correttamente. Come viene mostrato in [10] il throughput normalizzato di un SR-ARQ, che misura quanto efficacemente viene usato il canale, vale $S = 1 - \varepsilon$ ed è il migliore fra i throughput delle tre implementazioni nonché il migliore ottenibile dato un canale con probabilità d'errore ε. Se ad esempio $\varepsilon = 0.05$ si ha un throughput normalizzato $S = 0.95$ cioè per il 95% del tempo si mandano pacchetti utili e per il restante 5% si effettuano ritrasmissioni; dato che la probabilità di dover effettuare una ritrasmissione è il 5% non si può fare di meglio.

Per questo motivo il SR-ARQ è il più utilizzato e quindi ci limiteremo ad analizzare questo tipo di ARQ.
In particolare in un SR-ARQ i pacchetti ricevuti dal ricevitore vengono generalmente passati ai layer successivi nello stesso ordine con il quale trasmettitore li ha a sua volta ricevuti, tuttavia a causa degli errori e delle conseguenti ritrasmissioni i pacchetti possono arrivare al ricevitore in un ordine diverso da quello iniziale. Questo comporta la necessità di introdurre un buffer al ricevitore che effettui il risequenziamento [4]. Al trasmettitore invece c'è una coda che tiene tutti i pacchetti da trasmettere e per gestire le ritrasmissioni viene inserito un buffer che tiene conto dei pacchetti da ritrasmettere, in genere le

ritrasmissioni hanno priorità più alta delle trasmissioni, vale a dire che finché si devono effettuare ritrasmissioni non si possono trasmettere nuovi pacchetti.

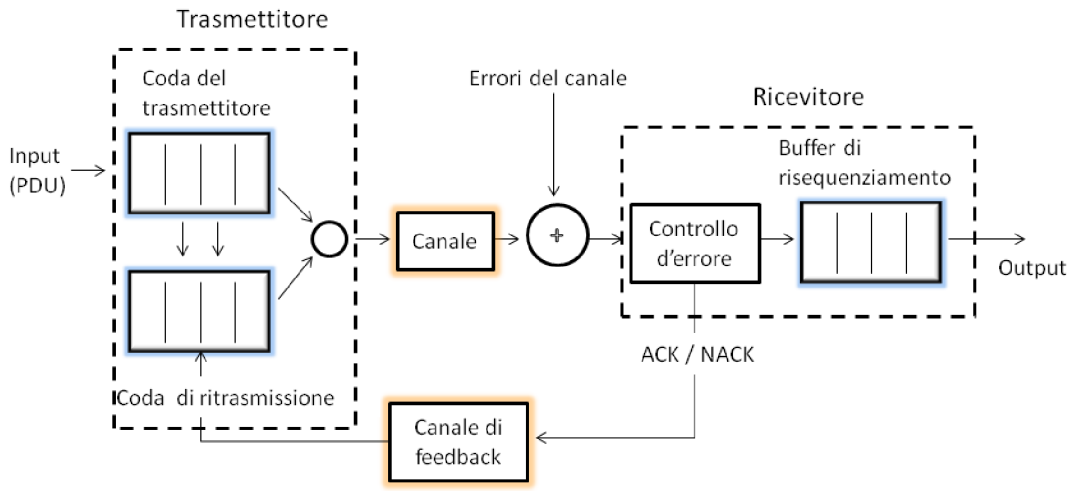

Figura 2.1: diagramma di un sistema di trasmissione basato su ARQ

Il tempo che un pacchetto perde in un SR-ARQ come quello in figura può essere diviso in tre componenti [4]:

- τ_Q ritardo d'accodamento: che è il tempo che il pacchetto perde nella coda del trasmettitore prima di essere trasmesso per la prima volta.
- τ_T ritardo di trasmissione: il tempo che passa dalla prima trasmissione fino alla corretta ricezione da parte del ricevitore, ovvero il tempo della prima trasmissione sommato ai tempi di tutte le ritrasmissioni necessarie affinché il pacchetto sia corretto.
- τ_R ritardo di risequenziamento: Il tempo perso nel buffer di risequenziamento. Ogni pacchetto deve stare nel buffer di risequenziamento finché tutti i pacchetti con indice inferiore al suo (cioè quelli trasmessi prima di lui) sono stati ricevuti

correttamente e solo successivamente può venire passato ai layer successivi.

Si definisce ritardo di consegna (delivery delay) la somma di ritardo di trasmissione e ritardo di risequenziamento, $\tau_D = \tau_T + \tau_R$. E' possibile progettare un ARQ anche con meccaniche out-of-order, i pacchetti vengono passati ai layer successivi nell'ordine con cui sono ricevuti correttamente dal ricevitore; in questo caso non è necessario il buffer di risequenziamento e il ritardo di risequenziamento diventa zero.

Per analizzare questi tempi di ritardo è comune l'introduzione di diverse ipotesi semplificative per facilitare i calcoli.

Le assunzioni più comunemente utilizzate sono:

- condizioni di heavy-traffic: si suppone che il trasmettitore abbia sempre pacchetti da trasmettere; si tratta di una buona approssimazione quando si calcola il ritardo di trasmissione o il ritardo di risequenziamento perché essendo un ipotesi peggiorativa permette di calcolare un limite superiore agli ultimi due tempi di ritardo [4], tuttavia quest'ipotesi rende impossibile calcolare il ritardo di accodamento nel trasmettitore. Per rimuovere questa assunzione bisogna considerare arrivi casuali in modo che la coda del trasmettitore abbia dimensione variabile (esiste così la possibilità che sia vuota e non ci siano pacchetti da trasmettere). In [12], ad esempio, si considera un tasso di arrivo bernoulliano.
- error-free ACK e NACK: si considera il canale di feedback privo di errori e quindi i segnali di acknowledgment vengono sempre ricevuti correttamente, vale a dire che un ACK non può venire scambiato per un NACK e viceversa. In questa

tesi useremo l'ipotesi di error-free ACK e NACK in quanto i pacchetti di feedback hanno dimensioni più piccole dei PDU e sono spesso protetti da codifiche per la correzione d'errore, rendendo la loro probabilità d'errore molto piccola [10]. Nelle situazioni in cui la probabilità d'errore del canale di feedback diventa rilevante (come ad esempio nelle trasmissioni sottomarine) si può procedere come in [2] dove viene analizzato cosa succede se si rilassa questa assunzione. Ne risulta che i tempi di ritardo aumentano ma mantengono lo stesso andamento.

- errori identicamente distribuiti (i.i.d. errors): i pacchetti vengono considerati identici e tutti con probabilità d'errore p in modo indipendente fra di loro. Tuttavia nella realtà gli errori dei pacchetti sono correlati fra loro a causa di effetti di *error burst* per i quali, quando il canale è in condizioni cattive, un intero gruppo di pacchetti consecutivi non viene ricevuto correttamente. Come si vedrà quest'approssimazione si può facilmente rimuovere modellando il canale come una catena di Markov.

- ideal SR-ARQ: si suppone che l'informazione relativa alla corretta ricezione del pacchetto sia immediatamente disponibile dopo la sua trasmissione, il che vuol dire che le ritrasmissioni vengono effettuate subito se il pacchetto non viene ricevuto. Se si usa quest'ipotesi non è possibile valutare il ritardo di risequenziamento in quanto effettuando le ritrasmissioni immediatamente i pacchetti vengono ricevuti in ordine [1].

Esistono già molti studi ed analisi sui tempi di ritardo dei sistemi ARQ. In [6] viene calcolata la distribuzione del numero di pacchetti in coda nei buffer ma non si considerano gli effetti di error burst del canale perché è usata l'ipotesi di i.i.d. errors.

In [4] viene sviluppata un'analisi dei tempi di consegna (τ_D) ma non viene calcolato il tempo di accodamento (τ_Q) perché si utilizza l'ipotesi di heavy traffic. In [11] si considera un canale markoviano con probabilità d'errore variabile nel tempo ma viene calcolato solo il valore medio dei tempi di ritardo. In [12] si considera un tasso di arrivi bernoulliano (rimuovendo l'ipotesi di heavy traffic) ma anche in questo caso si considera i.i.d. errors. In [1] vengono rimosse tutte queste assunzioni (i.i.d. errors, ideal SR-ARQ, heavy traffic) e viene calcolata, sotto l'ipotesi di arrivi bernoulliani, la distribuzione esatta dei tempi di accodamento e dei tempi di consegna. Il contributo di questa tesi consiste nell'estendere l'analisi svolta in [1] calcolando la distribuzione del tempo di accodamento anche nel caso di un tasso di arrivo poissoniano. A differenza di [1] si userà l'ipotesi di ideal SR-ARQ ma si spiegherà come rilassare quest'assunzione per calcolare i tempi di consegna.
Prima di iniziare ad analizzare un ideal SR-ARQ forniamo alcuni richiami sulle catene di Markov (ulteriori approfondimenti si trovano in [10]). Come si vedrà, il loro utilizzo permette di modellare matematicamente il canale di trasmissione o lo stato nel quale si trova il sistema eliminando diverse di queste ipotesi semplificative[1], riuscendo così a calcolare non solo i valori medi ma anche le distribuzioni complete dei parametri d'interresse.

2.2 Le catene di Markov

Una catena di Markov è un processo stocastico caratterizzato dalla proprietà di Markov, secondo la quale è possibile conoscere la descrizione statistica futura del sistema conoscendo solo lo stato presente del sistema (a prescindere da quale è stata l'evoluzione del sistema nel passato).

Le catene di Markov possono essere sia processi a tempo continuo che a tempo discreto; I sistemi ARQ si adattano

molto bene ad essere discretizzati suddividendo il tempo in slot uguali, quindi per la loro analisi si usano catene di Markov a tempo discreto.

Dato un processo aleatorio X_n a tempo discreto che assume in ogni intervallo n un valore (associato a uno stato) fra un possibile insieme finito di stati (di dimensione Z), la proprietà di Markov diventa:

$$P[X_{n_K} = c_k | X_{n_{k-1}} = c_k - 1, ...X_{n_0} = c_0] = P[X_{n_K} = c_k | X_{n_{k-1}} = c_k - 1]$$

Ovvero la distribuzione di X all'istante n_k dipende esclusivamente dalla sua distribuzione all'istante $n_k - 1$, con k intero.
Per capire quali siano le implicazioni di questa proprietà bisogna introdurre le probabilità di trovarsi in uno certo stato (state probability) in un determinato istante di tempo n e le probabilità di transizione da uno stato all'altro (transition probability).

- Si definisce probabilità di stato $p_j(n)$ la probabilità che la catena di Markov assuma valore j all' ennesimo passo del processo, $p_j(n) = P[X_n = j]$. Queste probabilità possono essere raccolte in un unico vettore $p(n) = [p_0(n), p_1(n), ..., p_Z(n)]$ dove la componente j-esima è la probabilità di trovarsi nello stato j.

- Si definisce probabilità di transizione $P_{i,j}$ la probabilità che il processo assuma lo stato j all'istante $n + 1$ dato che vale i all'istante n, cioè $P[X_n + 1 = j | X_n = i]$.

La markovianità del processo garantisce che è sufficente conoscere le probabilità di transizione per poter ricavare le probabilità di stato dell'istante $n + 1$ supponendo note quelle dell'istante

n. Infatti raggruppando opportunamente le probabilità di transizione in una matrice $\mathbf{P}$, detta matrice di transizione, si riesce a trovare una semplice legge che permette di studiare l'evoluzione del sistema:

$$p(n+1) = p(n) \cdot \mathbf{P} \tag{2.1}$$

In questo modo il prodotto di $p(n)$ con la colonna j-esima della matrice vale $\sum_{i=1}^{Z} p_i(n) P_{i,j}$ che è la probabilità che il processo assuma lo stato j all'istante $n+1$ secondo il teorema della probabilità totale.

L'utilità di poter analizzare un certo sistema tramite catene di Markov consiste nel fatto che supponendo noto lo stato iniziale $X_0 = x$, si può valutare l'intera evoluzione statistica del sistema. O si possono trovare i punti di equilibrio a cui converge il sistema cercando il vettore delle probabilità stazionario p_f che soddisfa la condizione:

$$p_f = p_f \cdot \mathbf{P} \tag{2.2}$$

.

Capitolo 3

Sistemi ARQ

3.1 La modellazione del canale

La modellazione più semplice e più utilizzata per il canale consiste in una catena di Markov discreta a due stati, usiamo la variabile $c(t)$ per indicare in quale stato ci troviamo all'istante t. Lo stato $c(t) = 1$ indica che il canale è soggetto a rumore (stato 'cattivo'), quando un pacchetto arriva al ricevitore e lo stato del canale vale 1 verrà inviato un NACK perché è necessaria la ritrasmissione, al contrario lo stato 0 indica che il pacchetto viene ricevuto correttamente perché non si verificano errori, in questo caso diciamo che il canale è in uno stato 'buono'. Questa è una buona approssimazione del canale [6], si può comunque estendere l'analisi utilizzando ipotesi aggiuntive e quindi una catena di Markov più complessa come viene fatto in [8]. Le probabilità di transizione fra gli stati sono $p_{00}, p_{11}, p_{01}, p_{10}$, con $p_{00} + p_{01} = 1$ e $p_{11} + p_{10} = 1$ (per semplicità indichiamo la probabilità di transizione $p_{[i|j]}$ con p_{ij}).

La matrice di transizione è: $P = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix}$.

Imponendo che $\mathbf{p} = \mathbf{p} \cdot P$, con $\mathbf{p}$ vettore delle probabilità di stato, e che la somma degli elementi di $\mathbf{p}$ sia 1 si

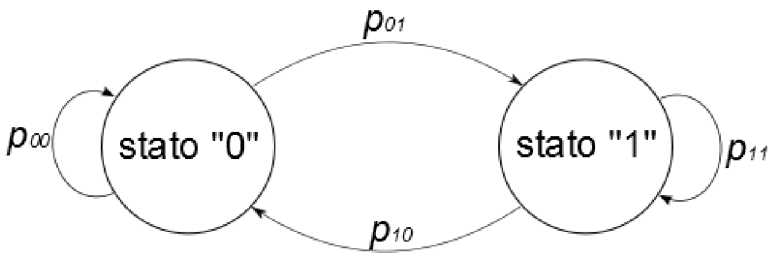

Figura 3.1: diagramma del canale

ottine:

$$\begin{cases} p_0 = p_0 p_{00} + p_1 p_{10} \\ p_1 = p_0 p_{01} + p_1 p_{11} \\ p_0 + p_1 = 1 \end{cases}$$

che una volta risolto da il valore asintotico delle probabilità d'errore $p_1 = \varepsilon = p_{01}/(p_{01} + p_{10})$, mentre invece $1/p_{10}$ è la lunghezza media dell'error burst (B), ovvero il numero medio di pacchetti compromessi quando il canale entra nello stato 'cattivo'.
L'utilità di questa modelazione consiste nel fatto che in questo modo si riesce a rimuovere l'ipotesi di i.i.d. errors. Infatti la probabilità che lo stato sia '1' dipende dalle probabilità di stato precedenti e dalla probabilità di transizione secondo $p_1 = p_0 p_{01} + p_1 p_{11}$.

3.2 Analisi di un ideal SR-ARQ

Per studiare gli ARQ tramite catene di Markov si può pensare di dividere il tempo in slot, ciascuno lungo quanto il tempo di trasmissione di un pacchetto in modo che a uno slot corrisponda una trasmissione. Assumiamo che fra uno slot di tempo e il successivo il canale possa cambiare stato secondo le sue probabilità di transizione.

Per implementare in Matlab l'analisi dei ritardi di accodamento di un SR-ARQ sono state usate le seguenti ipotesi:
- un canale modellato con catene di Markov come in **3.1** con $p_{01} \neq p_{11}$ e $p_{10} \neq p_{00}$ in modo da eliminare l'ipotesi di i.i.d. errors (ci si può ricondurre al caso i.i.d. errors ponendo $p_{01} = p_{11}$ e $p_{10} = p_{00}$)
- gli arrivi sono stati considerati prima bernoulliani e poi poissoniani. Nel caso di arrivi bernoulliani in ogni slot di tempo arriva un pacchetto con probabilità λ. Nel caso di arrivo poissoniani indichiamo con λ il numero medio di pacchetti che arrivano in uno slot di tempo. In entrambi i casi poniamo $\lambda < 1 - \varepsilon$ per non ricondurci a condizioni di heavy traffic in cui il buffer del trasmettitore è pieno.
- l'ipotesi di error-free ACK e NACK per i motivi spiegati in **2.1**
- l'ipotesi di ideal SR-ARQ per facilitare l'implementazione, anche se come si vedrà si può estendere l'analisi procedendo come in [1].
- consideriamo un buffer al trasmettitore di dimensione finita che tiene fino a q_{max} pacchetti, se la coda del buffer è piena e arriva un pacchetto supponiamo che il pacchetto venga respinto.
L'ipotesi di ideal SR-ARQ permette di sapere subito dopo la trasmissione se il pacchetto deve essere ritrasmesso o no, è come avere un SR-ARQ con windows size=1.

Il sistema ARQ può essere modellato come una catena di markov $X(t)$ che dipende da due variabili, la dimensione della coda $q(t)$ e uno stato $s(t)$ che indica la presenza o meno di un pacchetto da ritrasmettere nel canale e contemporaneamente dà informazioni sullo stato del canale. Analogamente a quanto avviene per il canale $X(t)$ cambia stato fra uno slot di tempo e il successivo secondo le sue probabilità di transizione.

Quando il canale si trova nello stato 'buono' nell'istante t ($c(t) = 0$) poniamo $s(t) = 0$. In questa situazione se avviene una trasmissione all'istante t (e questo avviene se c'è almeno un pacchetto in coda) allora la trasmissione si conclude con successo, se invece la coda del trasmettitore è vuota non avviene nessuna trasmissione. In entrambi i casi $s(t) = 0$ indica che il canale è in uno stato 'buono' e quindi un eventuale pacchetto trasmesso in quell'istante viene ricevuto correttamente. Quando il canale si trova nello stato 'cattivo' ($c(t) = 1$) le due possibilità 'coda vuota' e 'coda con almeno un pacchetto' vanno trattate distintamente. Se nella coda c'è almeno un pacchetto da trasmettere poniamo $s(t) = 1$, in questo caso la trasmissione non andrà a buon fine per via dello stato del canale. Per l'ipotesi di ideal SR-ARQ supponiamo di sapere istantaneamente che il pacchetto non è stato ricevuto e quindi lo ritrasmettiamo nello slot di tempo successivo, in questo caso diciamo che il pacchetto si trova in servizio. Quando un pacchetto è in servizio, finché il canale ha lo stato $c(t) = 1$, viene ritrasmesso in ogni slot di tempo; quando il canale passa dallo stato 1 allo stato 0 il pacchetto viene ricevuto con successo e nello slot di tempo successivo verrà trasmesso il prossimo pacchetto nella coda del trasmettitore. Ad esempio nello stato X(t)=(3,1) ci sono 4 pacchetti, 3 in coda e 1 in servizio (il canale si trova in condizioni 'cattive'). Infine se $c(t) = 1$ ma non avvengono trasmissioni (perché la coda del trasmettitore è vuota) vuol dire che non ci sono pacchetti in servizio. Tuttavia in questo caso non si può porre $s(t) = 0$ perché non si terrebbe conto del fatto che il canale si trova in uno stato 'cattivo', ne si può porre $s(t) = 1$ perché $s(t) = 1$ indica che il canale è cattivo ma anche che c'è un pacchetto in servizio nel canale. Per trattare questo caso si introduce un nuovo valore da associare a $s(t)$ (ad esempio $s(t) = -1$) per indicare che il canale è in

condizioni cattive ma non è stata effettuata nessuna trasmissione e quindi non c'è nessun pacchetto in servizio. Ipotizziamo infine che se si verifica un'arrivo nello slot t e la coda è vuota il pacchetto viene trasmesso immediatamente (nello stesso slot), con successo se $c(t) = 0$, mentre resta in servizio se $c(t) = 1$.

Associamo a ogni possibile stato una probabilità $\pi(q, s)$, chiamiamo π il vettore che racchiude tutte le probabilità di stato e $\mathbf{S}$ la matrice di transizione.

3.2.1 Matrice di transizione di un ideal-SR-ARQ

Per poter creare la matrice di transizione bisogna calcolare le probabilità di transizione che esistono fra i vari stati (q, s) da un istante $t-1$ all'istante successivo t; Queste probabilità di transizione possono essere calcolate perché noto $X(t-1)$ si può ricavare la probabilità di $X(t)$ in funzione della probabilità che si verifichi un arrivo λ e delle probabilità di transizione del canale $p_{00}, p_{11}, p_{01}, p_{10}$.

Le probabilità di transizione per $q = 0$ vanno calcolate singolarmente, mentre se $q > 0$ le transizioni fra $(q-1, s), (q, s), (q+1, s)$ con $s \in \{-1, 0, 1\}$ hanno lo stesso comportamento per ogni valore di q.

$q = 0$

$$\pi(0,-1) = (1-\lambda)p_{01}\pi(0,0)+(1-\lambda)p_{11}\pi(0,-1) \tag{3.1}$$

$$\begin{aligned}\pi(0,0) &= \lambda p_{00}\pi(0,0) + \lambda p_{10}\pi(0,-1) + (1-\lambda)p_{00}\pi(0,0)+ \\ &\quad + (1-\lambda)p_{10}\pi(0,-1) + (1-\lambda)p_{00}\pi(1,0) + (1-\lambda)p_{10}\pi(0,1) \\ &= p_{00}\pi(0,0) + p_{10}\pi(0,-1) + (1-\lambda)p_{00}\pi(1,0) + (1-\lambda)p_{10}\pi(0,1)\end{aligned} \tag{3.2}$$

$$\pi(0,1) = \lambda p_{01}\pi(0,0) + \lambda p_{01}\pi(0,-1) + (1-\lambda)p_{01}\pi(1,0) + \\ + (1-\lambda)p_{01}\pi(0,1) \tag{3.3}$$

$q > 0$

$$\pi(q,-1) = 0 \quad \forall\ q \tag{3.4}$$

$$\pi(q,0) = \lambda p_{10}\pi(q-1,1) + \lambda p_{00}\pi(q,0) + (1-\lambda)p_{10}\pi(q,1) + \\ + (1-\lambda)p_{00}\pi(q+1,0) \tag{3.5}$$

$$\pi(q,1) = \lambda p_{01}\pi(q,0) + \lambda p_{11}\pi(q-1,1) + (1-\lambda)p_{11}\pi(q,1) + \\ + (1-\lambda)p_{01}\pi(q+1,0) \tag{3.6}$$

(3.1): raggiungiamo lo stato (0,-1), solo quando non arrivano pacchetti e passiamo dallo stato $s(t-1) = 0, 1$ a $s(t) = 1$, non può essere $s(t-1) = 1$ perché in quel caso c'è un pacchetto in servizio e non può risultare $s(t) = -1$.
(3.2): si considerano le possibili transizioni che vanno in $\pi(0,0)$, si noti che non si ha mai una combinazione di λ con $\pi(0,1)$ perché ci sarebbero due pacchetti di cui solo uno (quello nel canale) viene trasmesso, l'altro resterebbe in coda, mentre invece in $\pi(0,0)$ abbiamo zero pacchetti nel sistema.
(3.3): $\pi(0,1)$ può essere causato da un arrivo e canale vuoto o da nessun arrivo e canale pieno.
(3.4): $\pi(q,-1)$ con $q > 0$ è zero perché se lo stato vale -1 il canale era 'cattivo' ma non c'era nessun pacchetto in coda.
(3.5) e (3.6): $\pi(q,s)$ possono essere ottenuti partendo da $\pi(q-1,s')$,$\pi(q,s')$, $\pi(q+1,s')$ a seconda delle probabilità di transizione del canale $p_{s's}$ e della presenza o meno di un nuovo pacchetto (λ o $(1-\lambda)$).

La matrice $\mathbf{S}$ è nella forma:

$$\mathbf{S} = \begin{pmatrix} S_1 & 0 & 0 \\ 0 & S_2 & 0 \\ 0 & 0 & S_3 \end{pmatrix}$$

In particolare ponendo
$A = \lambda p_{10}$ $B = \lambda p_{00}$ $C = (1-\lambda)p_{10}$ $D = (1-\lambda)p_{00}$
$E = \lambda p_{11}$ $F = \lambda p_{01}$ $G = (1-\lambda)p_{11}$ $H = (1-\lambda)p_{01}$
e usando (3.1),(3.2),(3.3) risulta

$$S_1 = \begin{pmatrix} G & A+C & E & 0 & 0 & 0 & \cdots \\ H & D+B & F & 0 & 0 & 0 & \cdots \\ 0 & C & G & 0 & A & E & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & D & H & 0 & B & F & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

mentre usando (3.4),(3.5),(3.6), S_2 risulta essere

$$S_2 = \begin{pmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & 0 & A & E & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & 0 & B & F & 0 & 0 & 0 & \cdots \\ \cdots & 0 & C & G & 0 & A & E & \cdots \\ \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & 0 & D & H & 0 & B & F & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Infine usando (3.4),(3.5),(3.6) e imponendo che se ci sono già q_{max} pacchetti in coda i nuovi arrivi vengono respinti, risulta

$$S_3 = \begin{pmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & 0 & 0 & 0 & 0 & 0 & \\ \cdots & B & F & 0 & 0 & 0 & \\ \cdots & C & G & 0 & A & E & \\ \cdots & 0 & 0 & 0 & 0 & 0 & \\ \cdots & D & H & 0 & B & F & \\ \cdots & 0 & 0 & 0 & A+C & E+G & \end{pmatrix}$$

Come verifica si può controllare che la somma degli elementi in ogni riga di S deve dare 1, infatti i valori contenuti nella riga i-esima sono le probabiltà che si verifichi una transizione a partire dallo stato i-esimo.

Usando **S** si trova π_f, vettore delle probabilità di stato asintotiche, in modo analogo a come viene fatto nella sezione **3.1**. Conoscendo le probabilità di stato asintotiche si può trovare la descrizione statisica della coda del trasmettitore. La probabilità che ci siano q utenti in coda è: $Prob[q] = \sum_{s=-1}^{1} \pi_f(q,s)$. Per calcolare il tempo di accodamento di un pacchetto che è arrivato all'istante t è sufficente conoscere il numero di pacchetti che devono essere trasmessi prima di lui, vale a dire tutti i pacchetti che si trovano davanti a lui nella coda. Il tempo perso nella coda t_Q sarà uguale al tempo di tutte le trasmissioni e ritrasmissioni dei pacchetti che vengono trasmessi prima di lui. Definiamo $S(1)$ e $S(0)$ che sono la matrice S quando λ vale rispettivamente 1 e 0 (arrivo certo e nessun arrivo). Supponiamo che il pacchetto di

cui vogliamo calcolare il tempo di accodamento arrivi all'istante t. In seguito al suo arrivo il vettore delle probabilità di stato diventa $\Delta = \pi_f S(1)$. Si può suppore che non avvengano ulteriori arrivi dopo l'istante t perché gli arrivi successivi non influenzano il tempo di accodamento del pacchetto. In questo modo sappiamo che il pacchetto viene trasmesso quando la coda si svuota ($q = 0$). A questo punto si può calcolare il numero di slot di tempo necessari a raggiungere lo stato $(0, s)$ utilizzando $S(0)$, perché abbiamo supposto che non ci siano ulteriori arrivi. Chiamiamo e_Q il vettore di dimensione pari a quella di π_f che vale 1 in corrispondenza degli stati con $q = 0$ e vale 0 nelle restanti posizioni.
La probabilità che la coda si svuoti dopo uno slot di tempo è $\Delta \cdot S(0) \cdot e_Q$. Generalizzando, la probabilità che siano necessari meno di $\tau + 1$ slot di tempo per raggiungere lo stato $(0, s)$ è

$$C_Q[\tau] = \Delta \cdot S(0)^\tau \cdot e_Q \tag{3.7}$$

di conseguenza $Prob[\tau_Q = \tau] = \begin{cases} C_Q[0], & \text{se } \tau{=}0 \\ C_Q[\tau] - C_Q[\tau - 1], & \text{se } \tau{>}0 \end{cases}$

Nel caso di arrivi poissoniani il numero di pacchetti x che può arrivare in uno slot di tempo segue la distrubuzione di Poisson:

$$Prob(x) = e^{-\lambda}\frac{\lambda^x}{x!} \tag{3.8}$$

Poiché $\lambda < 1 - \varepsilon$, la probabilità che ci siano più di 8 o 9 arrivi diventa trascurabile quando si esegue l'implementazione in Matlab. Il calcolo delle probabilità di transizione è analogo a quello svolto nel caso di arrivi bernoulliani: si considerano le transizioni che possono portare a uno stato (q_i, s_i) partendo da un altro stato (q_j, s_j) e si calcola le probabilità di ciascuna transizione combinando opportunamente le probabilità degli

arrivi poissoniani con le probabilità di transizione del canale. In questo caso la matrice di transizione S risulta essere più densa perché da un generico stato (q, s) non esistono solamente le transizioni verso gli stati con $q-1$ o $q+1$ utenti in coda (come nel caso di arrivi bernoulliani), ma anche le transizioni verso stati con più di $q+1$ utenti in coda (in corrispondenza di 2 o più arrivi).
In Matlab si può creare il vettore π associando un numero intero (che fa da indice) a ogni stato (q, s), ad esempio $i = 3q + (s+1) + 1$. La dimensione del vettore π è 3Q, con $Q = q_{max} + 1$. La dimensione della matrice di transizione $\mathbf{S}$ è $3Q \times 3Q$. Una volta calcolate $Prob[q]$ e $Prob[\tau_Q]$ si possono tracciare le distribuzioni degli utenti in coda e dei tempi di accodamento.

3.3 Analisi di un non-ideal SR-ARQ

In [1] oltre alle ipotesi già rimosse nella sezione **3.2** si rilassa anche l'ipotesi di ideal SR-ARQ riuscendo in questo modo a calcolare anche il tempo di consegna τ_D.

L'impostazione generale è la stessa dell'ideal-SR-ARQ, l'unica differenza è che bisogna aspettare un t_{RTT} per avere l'informazione relativa al successo o insuccesso della trasmissione (ACK o NACK).
In caso di NACK si suppone che la ritrasmissione avvenga dopo un t_{RTT} che equivale a m slot (m=window size). Nel canale ci sono sempre m pacchetti e per ognuno di essi bisogna sapere se è necessario o meno ritrasmetterlo, per tracciare il loro stato si può usare un vettore $\mathbf{b}$ di dimensione m. All'ultimo elemento b_m del vettore è associato il pacchetto trasmesso nell'istante attuale t, mentre al primo b_1 è associato il pacchetto trasmesso

nell'intervallo $t-m+1$. Ad ogni elemento del vettore è associato il valore 0 se il corrispondente pacchetto non necessita di ritrasmissione, mentre il valore 1 indica che il pacchetto deve essere ritrasmesso; si sceglie il valore a seconda dello stato del canale nello slot in cui è stato trasmesso il pacchetto, se era cattivo si ha 1, se era buono si ha 0.

Quindi per analogia con il caso dell'ideal SR-ARQ lo stato del sistema è dato da $(q(t), \mathbf{b}, c(t))$. Si nota che b_m e $c(t)$ sono correlati in quanto $c(t) = 0$ implica $b_m = 0$ mentre $c(t) = 1$ implica $b_m = 1$ se è stata effettuata una trasmissione e $b_m = 0$ se non è stata effettuata una trasmissione (perché la coda era vuota e non ci sono stati arrivi).
Si possono raggruppare $c(t)$ e b_m in unica variabile $s(t)$ che funziona in modo analogo alla variabile $s(t)$ dell'ideal-SR-ARQ, $s(t) = 0$ se lo stato è buono, $s(t) = 1$ se lo stato è cattivo e avviene una trasmissione, $s(t) = -1$ se lo stato è cattivo ma non avvengono trasmissioni. In questo modo risulta $X(t) = (q(t), b_1, b_2, ..., b_{m-1}, s(t))$

Fra l'istante $t-1$ e t il vettore $\mathbf{b}$ evolve in modo deterministico (fatta eccezione per l'ultima posizione di $\mathbf{b}$ che dipende dallo stato del canale): $b_j(t) = b_{j+1}(t-1)$ per $1 \leq j \leq m-2$ e $b_{m-1} = 0$ se $s(t) = 0, -1$, $b_{m-1} = 1$ se $s(t) = 1$. Mentre q(t) e s(t) dipendono dai valori di q(t-1) e s(t-1) e possono avere diversi valori a seconda dell'arrivo o meno di pacchetti e delle probabilità di transizione del canale. Ad esemprio dato $X(t-1) = (q, b_1, b_2, ..., b_{m-1}, s)$ lo stato di arrivo è $X(t) = (q + b_1, b_2, ..., b_{m-1}, u[s-1], d)$ con probabilità $\lambda p_{|s|d}$.
Il numero di possibili stati del sistema è $3Q2^{b_{m-1}}$ (anche in questo caso $Q = q_{max} + 1$) perché $s \in -1, 0, 1, q \in 0, 1, .., q_{max}$ e $\mathbf{b}$ è un vettore binario di $2^{b_{m-1}}$ elementi, considerando le possibili transizioni si calcola $\pi(q, b_1, b_2, ..., b_{m-1}, s)$ per ogni stato e si

crea la matrice di transizione che avrà dimensione $3Q2^{b_{m-1}}$.

Una volta note le $\pi(q, b_1, b_2, ..., b_{m-1}, s)$ si può calcolare $P[q]$ e i tempo t_Q, inoltre dato che sono stati modellati tutti i pacchetti nel canale tramite il vettore **b** si può anche calcolare i tempi di delivery.

Il tempo t_Q si calcola come il tempo necessario per raggiungere lo stato (0,**a**,s) (dove **a** è un vettore binario di m-1 elementi) (in modo analaogo a quanto fatto nella sezione **3.2**). Affinché un pacchetto possa essere passato ai layer successivi è necessario che tutti i pacchetti inviati prima di lui siano stati ricevuti correttamente (meccanica in-order); questo vuol dire che se avviene una trasmissione all'istante t, se il pacchetto viene ricevuto, deve restare nel buffer di risequenziamento fino a quando tutti i pacchetti con indice inferiore (quelli inviati prima di lui) vengono ricevuti correttamente. Chiamiamo t_G il tempo totale perso nel sistema da parte del pacchetto. t_G è il tempo necessario affinché tutti i pacchetti trasmessi precedentemente vengano ricevuti correttamente, questo avviene quando si raggiunge lo stato (0, **0**, s) perché quando il vettore **a** diventa un vettore di zeri tutti i pacchetti sono stati ricevuti correttamente. Come nel calcolo di t_Q i pacchetti arrivati successivamente al pacchetto arrivato nell'istante t non influiscono sul tempo di consegna: le ritrasmissioni hanno priorità maggiore delle trasmissioni, ed è sufficiente che solo i pacchetti inviati precedentemente vengano ricevuti correttamente. Il tempo di consegna è $t_D = t_G - t_Q$.

Capitolo 4

Risultati per un ideal SR-ARQ

La distribuzione del numero di pacchetti in coda è stata rappresentata usando la funzione di distribuzione complementare (complementary cumulative distribution function, ccdf). La $ccdf(q)$ è la probabilità che ci siano più di q pacchetti in coda e si ottiene utilizzando le $P[q]$. $ccdf(q) = \sum_{i=q+1}^{q_{max}} P[i]$. Anche il tempo di accodamento viene rappresentato con una *ccdf*. In questo caso $ccdf(\tau_Q)$ è la probabilità che il pacchetto debba aspettare più di τ_Q slot di tempo.

I paramtri utilizzati (tranne dove è specificato diversamente) sono:
- $B = 5$, lunghezza dell'error burst
- $\varepsilon = 0.1$, probabilità d'errore del canale
- $q_{max} = 100$, dimensione massima della coda del trasmettitore.

La ccdf è stata rappresentata per valori di q che vanno da 0 a 80. Cambiando il valore di q_{max} (ma sempre con $q_{max} > 100$) si nota che le distribuzioni nell'intervallo [0,80] non subiscono variazioni. Questo consente di affermare che le *ccdf* ottenute sono una buona approssimazione delle *ccdf* nel caso di buffer di dimensione illimitata.

4.1 Pacchetti in coda e tempi di accodamento

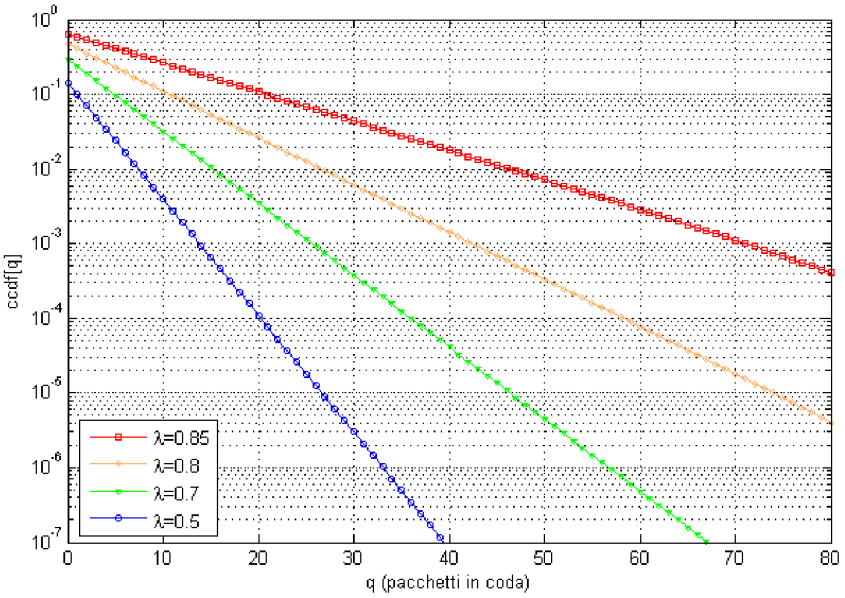

Figura 4.1: $ccdf(q)$, arrivi Bernoulliani

Le figure 4.1 e 4.2 sono state ottenute considerando un tasso degli arrivi Bernoulliano con probabilità di arrivo λ. Nella figura 4.1 è presentata la $ccdf(q)$, nella 4.2 la $ccdf(\tau_Q)$. Le due distribuzioni sono strettamente correlate, infatti benché la $ccdf(\tau_Q)$ sia più lenta a calare della $ccdf(q)$ si nota che le due distribuzioni hanno lo stesso andamento. Tale somiglianza si nota anche in tutti i grafici successivi, per questo motivo verrano presentate solo le $ccdf(q)$, sapendo che le $ccdf(\tau_Q)$ hanno lo stesso comportamento qualitativo. Questa correlazione fra le due distribuzioni dipende dal fatto che il tempo d'accodamento che un pacchetto deve aspettare è proporzionale al numero di pacchetti che sono in coda davanti a lui. Le due figure mettono in evidenza che l'aumentare di λ porta ad avere più pacchetti in

coda (e quindi tempo di accodamento maggiore). Quando λ si avvicina a $1-\varepsilon$ la probabilità di avere molti pacchetti in coda aumenta considerevolmente. Di fatto quanto λ supera $1-\varepsilon$ il numero di pacchetti che arriva nel sistema è maggiore del numero di pacchetti che vengono trasmessi con successo e quindi il buffer tenderà a saturare.

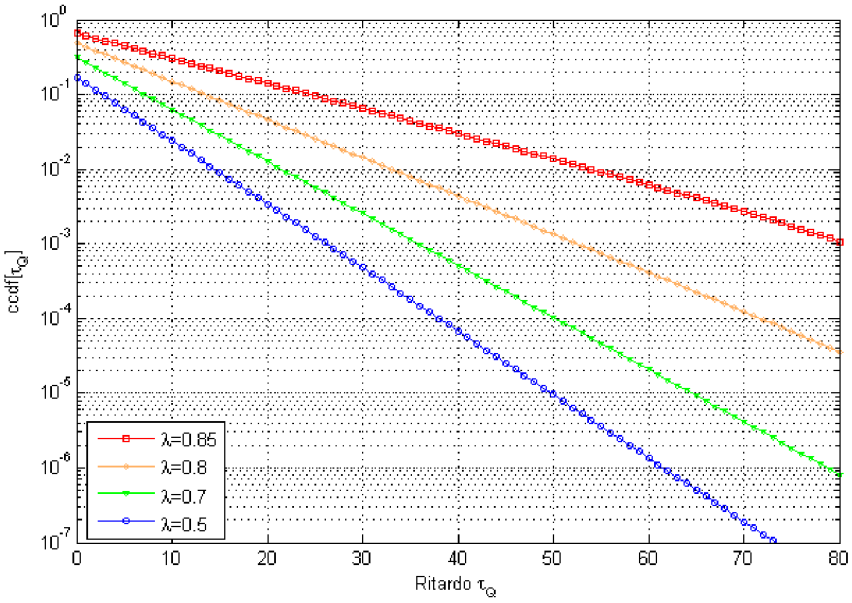

Figura 4.2: $ccdf(\tau_Q)$, arrivi Bernoulliani

4.2 Confronto fra arrivi Bernoulliani e arrivi Poissoniani

Nella figura 4.3 vengono confrontati tassi di arrivo Bernoulliani con tassi di arrivo Poissoniani. Si nota che a parità di λ il buffer del trasmettitore tende ad essere più pieno se gli arrivi sono Poissoniani. Questo può essere attribuito al fatto che nel caso di arrivi Poissoniani possono arrivare più pacchetti

contemporaneamente (eventualità che non è possibile con arrivi Bernoulliani). Nel caso di tasso degli arrivi Poissoniano l'arrivo di più pacchetti in un certo istante implica che negli slot di tempo successivi ci saranno di certo pacchetti da trasmettere, inoltre negli slot di tempo successivi può verificarsi l'arrivo di uno o più pacchetti; esiste quindi la possibilità che si formino accumuli di pacchetti a prescindere dalle condizioni del canale. Al contrario nel caso di arrivi Bernoulliani può arrivare al massimo un pacchetto in ogni slot di tempo, di conseguenza è meno probabile che si formino accumuli di pacchetti nel buffer (succede solo se il canale è in condizioni 'cattive').

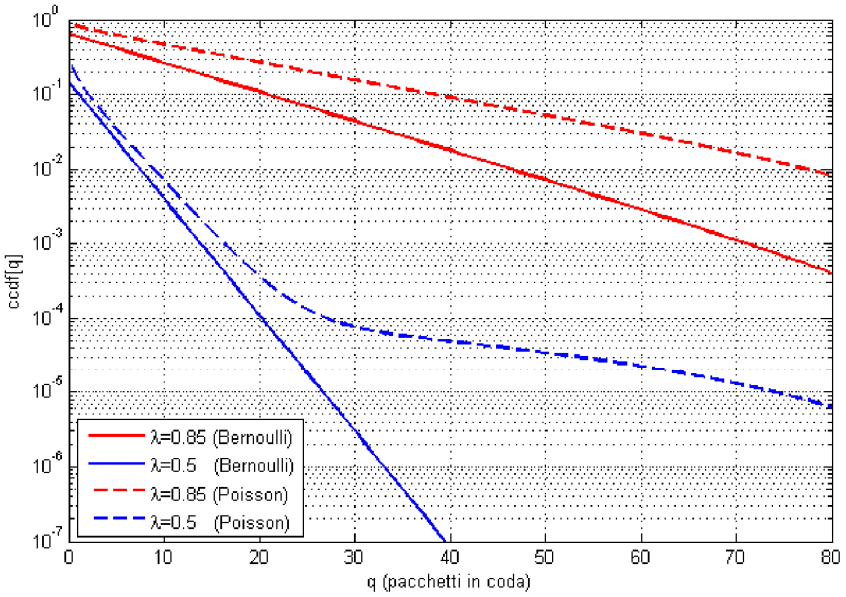

Figura 4.3: $ccdf(q)$, arrivi Bernoulliani e arrivi Poissoniani

4.3 Confronto fra i.i.d. errors e errori correlati

Per valutare quanto possa essere influente l'ipotesi di i.i.d. errors è stato effettuato il confronto fra errori indipendenti ($p_{01} = p_{11} = \varepsilon$

e $p_{10} = p_{00} = 1 - \varepsilon$) e errori dipendenti ($p_{01} \neq p_{11}$ e $p_{10} \neq p_{00}$, $\varepsilon = 0.1$, $B = 5$) al variare di λ. I grafici 4.4 e 4.5 mostrano chiaramente che la coda della funzione di distribuzione è più alta nel caso di errori dipendenti. Questo avviene per via dell' error burst B: quando il canale entra nello stato 'cattivo' ci rimane per un periodo sufficientemente grande da compromettere la trasmissione di un intero gruppo di pacchetti successivi. Di conseguenza è più probabile che si formino accumuli di pacchetti in coda.

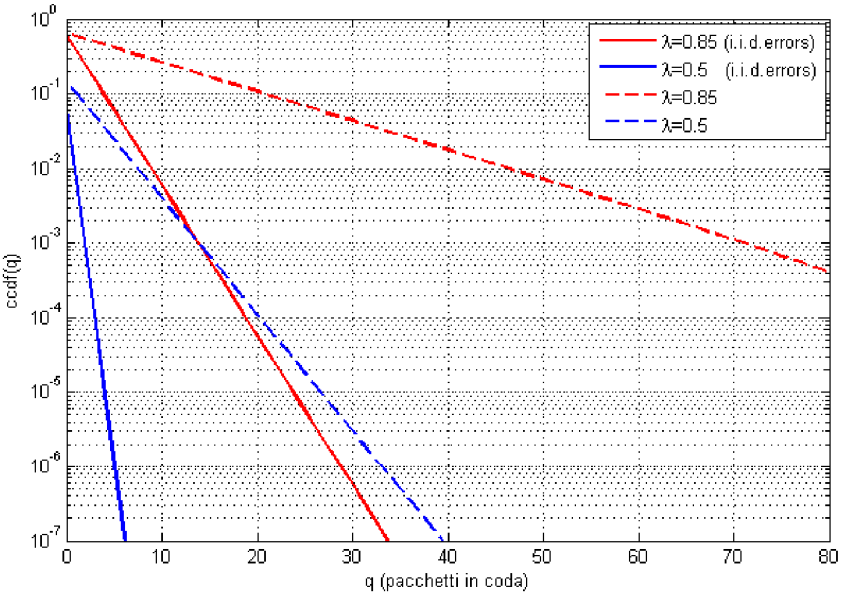

Figura 4.4: $ccdf(q)$, arrivi Bernoulliani

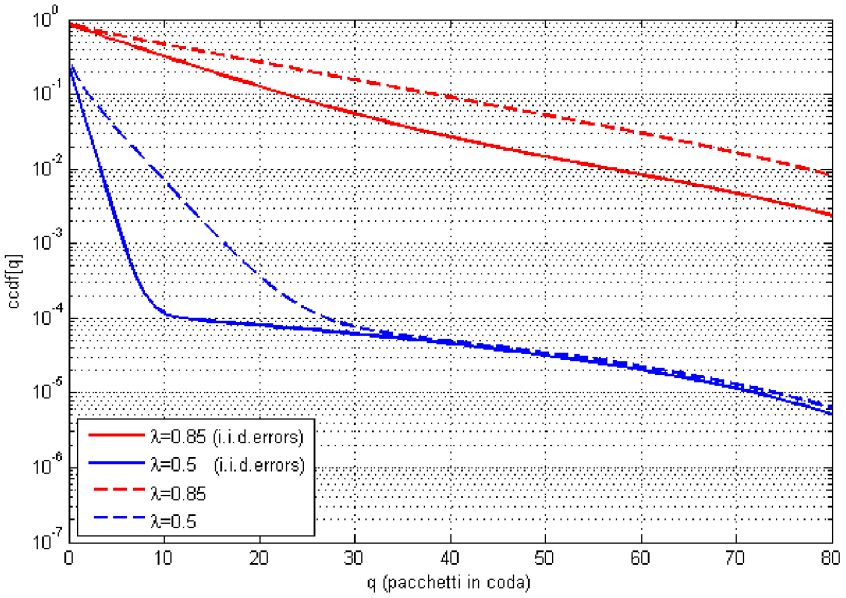

Figura 4.5: $ccdf(q)$, arrivi Poissoniani

4.4 Effetto dell'error burst B e della probabilità d'errore ε

Per avere un'idea di quantosia rilevante l'error burst B si può considerare la ccdf(q) (in caso di arrivi Bernoulliani) per diversi valori di B. Nelle figure 4.6 e 4.7 sono presentati i risultati per B che varia fra 5 e 25 ($\lambda = 0.5$, $\varepsilon = 0.1$). L'aumento dell'error burst implica un maggiore numero di pacchetti che devono essere ritrasmessi (in media) quanto il canale è in condizioni 'cattive'. Ne consegue la presenza di più pacchetti nel buffer del trasmettitore.
Nella figura 4.8 vengono fatti variare B ed ε contemporaneamente nel caso di arrivi bernoulliani, facendo aumentare questi parametri il numero di pacchetti in coda è maggiore. Ne risulta che un aumento di B è particolarmente influente sulla parte finale della distribuzione, che risulta essere più alta rispetto al

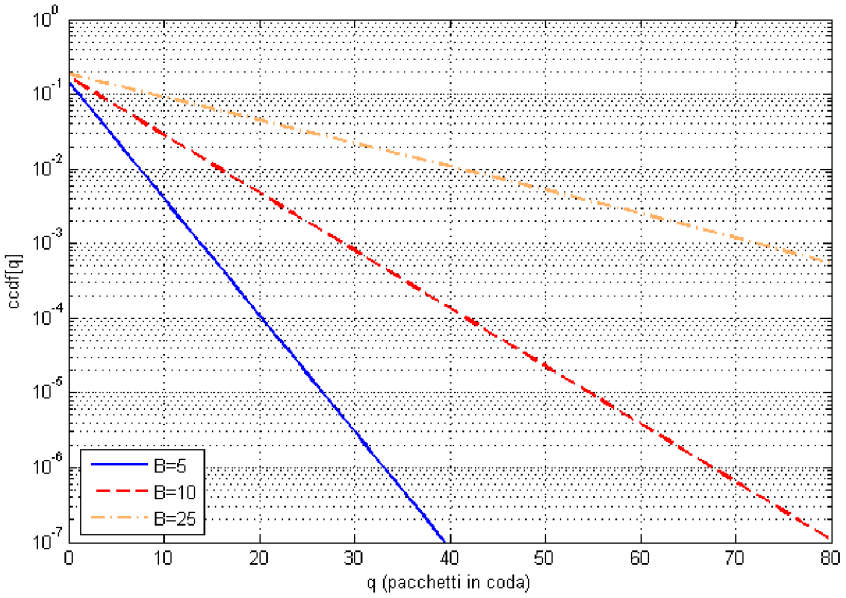

Figura 4.6: $ccdf(q)$, arrivi Bernoulliani

caso in cui aumenta ε. Si può concludere che sia B che ε sono parametri importanti per la dimensione della coda ed in particolare un error burst grande aumenta la probabilità di avere tanti pacchetti nel buffer.

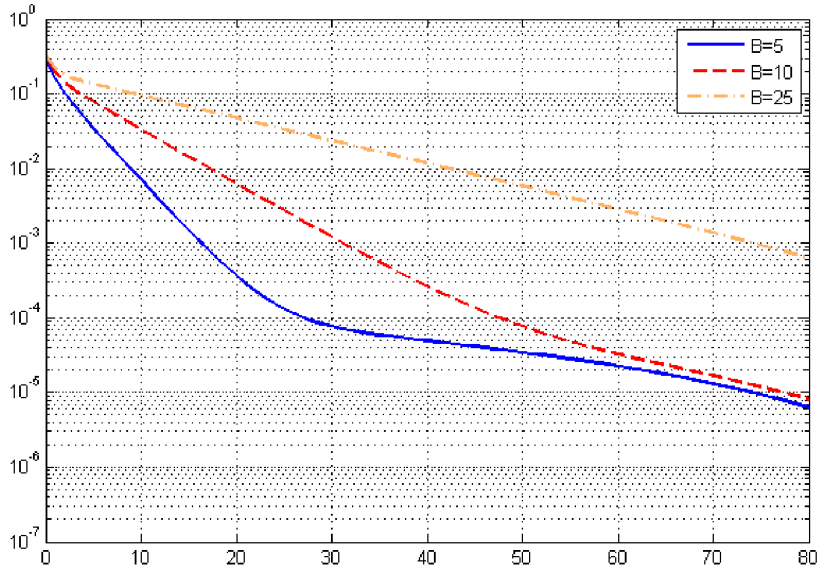

Figura 4.7: $ccdf(q)$, arrivi Poissoniani

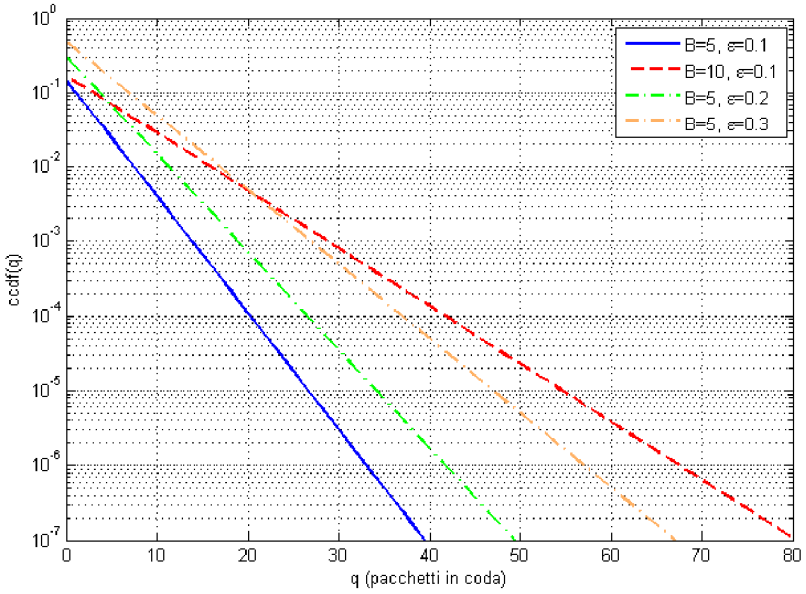

Figura 4.8: $ccdf(q)$, arrivi Bernoulliani

Capitolo 5

Conclusioni

In questa tesi sono state analizzate le distribuzioni dei pacchetti in coda e dei tempi di accodamento nel caso di un ideal SR ARQ. I risultati grafici mostrano che le distribuzioni dei tempi hanno lo stesso andamento delle distribuzioni dei pacchetti in coda. Sono stati confrontati tassi di arrivo poissoniani con tassi di arrivo bernoulliani, osservando che nel caso di arrivi poissoniani si ottengono tempi e code più grandi a causa di fenomeni di accumulo dei pacchetti. Il confronto fra i.i.d. errors e errori correlati ha mostrato che le distribuzioni assumevano andamenti differenti nei due casi; questo indica che la correlazione fra gli errori e gli effetti di error burst influiscono visibilmente sui tempi di accodamento e devono essere considerati per effettuare un'analisi accurata.

Inoltre, al di là dei singoli risultati grafici ottenuti, va sottolineato che i metodi di analisi qui presentati hanno validità generale e possono essere estesi a piacimento a seconda del tipo di analisi da effettuare. Si possono cambiare le assunzioni sul tipo di arrivi (ad esempio tasso di arrivo con distribuzione geometrica), si possono calcolare i tempi di consegna come spiegato nella sezione **3.3**, ma si possono anche utillizare procedimenti simili a quello presentato per analizzare altri tipi

di sistemi ARQ, come in [17] dove si studia il throughput di un HARQ o in [3] dove si analizza un SR Type II HARQ. In ogni caso la modellazione tramite catene di Markov permette un'analisi accurata dei parametri d'interessere nei sistemi ARQ.

Bibliografia

[1] L.Badia, M.Rossi, M.Zorzi, "SR ARQ Packet delay statistics on Markov channels in the presence of a variable arrival rate," *IEEE Transactions on Wireless Communication*, Vol 5, pp.1639-1644, July 2006

[2] L.Badia, "On the effect of feedback errors in Markov models for SR ARQ packet delays," *IEEE GLOBECOM*, November 2009

[3] L.Badia, M.Levorato, M.Zorzi, "Markov analysis of selective repeat type II hybrid ARQ using block codes," *IEEE Transactions on Communications*, Vol 56, pp.1434-1441, September 2008

[4] M.Rossi, L.Badia, M.Zorzi, "Exact statistics of ARQ packet delivery delay on Markov channels with finite round-trip delay," *Proc. IEEE Globecom*, vol.6, pp.3356-3360, 2003

[5] M.Rossi, L.Badia, M.Zorzi, "On the delay statistics of an aggregate of SR-ARQ packets over Markov channels with finite round-trip delay," *Proc. IEEE Wireless Comm. and Networking*, vol.3, pp.1773-1778, march 2003

[6] M.Zorzi, R.R.Rao, L. B. Milstein, "Error statistics in data transmission over fading channels," *IEEE Trans. Commun.*, vol 46, pp. 1468-1477, 1998.

[7] Z.Rosbert, M. Sidi, "Selective-repeat ARQ: the joint distribution of the transmitter and the receiver resequencing buffer occupancies," *IEEE Trans. Commun.*, vol. 38, no.9, pp.1480-1438, 1990.

[8] M.Rossi, L.Badia, M. Zorzi, "SR-ARQ delay statistics on N-state Markov channels with finite round trip delay," *Proc. IEEE Globecom 2004.*

[9] M.Rossi, L.Badia and M. Zorzi, "SR ARQ delay statistics on N-state Markov channels with non-instantaneous feedback," *IEEE Transaction on Wireless Communication*, vol 5, pp. 1526-1536, June 2006.

[10] N.Benvenuto, M.Zorzi, "Principles of communications network and system"

[11] J.G.Kim and M.M.Krunz, "Delay analysis of Selective Repeat ARQ for a Markovian source over a wireless channe," *IEEE Trans. Veh. Tecgbik.*, vol.49, no. 5, pp.1968-1981, 2000.

[12] M.E.Anagnostu and E.N.Protonotarios, "Performance analysis of the Selective-Repeat ARQ protocol," *IEEE Trans. Commun.*, vol.34, no. 2, pp. 127-135, 1986.

[13] J.Chang and T.Yang, "End-to-end delay of adaptive Selective Repeat ARQ protocol," *IEEE Trans. Commun.*, vol.42, pp. 2926-2928, 1994.

[14] B.L.Long, E.Hossain, and A.S.Alfa, "Queuing analysis for radio link level scheduling in a multi-rate TDMA wireless network," *IEEE Globecom 2004*, vol. 6, Dallas, 2004.

[15] L.Badia, "On the impact of correlated arrivals and errors on ARQ delay terms," *IEEE Trans. Commun.*, vol. 57, pp. 334-338, Feb 2009.

[16] R.Cam. and C.Leung, "Throughput analysis of some ARQ protocols in the presence of feedback errors," *IEEE Trans. Commun.*, vol. 45, no. 1, pp. 35-44, Jan. 1997.

[17] Q.Zhang and S.A. Kassam, Hybrid ARQ with selective combining for fading channel, *IEEE J. Select. Areas Commun.*, vol.17, no. 5, pp. 867-880, May 1999.

[18] S.Lin, D.J.Costello, and M.J.Miller, Automatic-Repeat-reQuest error control schemes, *IEEE Commun. Mag.*, vol. 22, no. 12, pp. 5-17, 1984.