



UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
CORSO DI LAUREA IN INGEGNERIA DELL'INFORMAZIONE

**Analysis of error control techniques
for multimedia data with
incremental representation**

Laureando:
Anna Valeria GUGLIELMI

Relatore:
Leonardo BADIA

Anno accademico 2011/2012

Indice

1	Introduzione	4
2	Related Work	7
3	Descrizione del modello	12
3.1	Descrizione del sistema e calcolo delle probabilità stazionarie	12
3.2	Calcolo delle metriche	20
4	Risultati	27
5	Conclusioni	36
	Bibliografia	38

Abstract

Questo elaborato presenta, a partire dallo studio del protocollo dell'ARQ, un modello matematico per il traffico di contenuto multimediale. Si suppone che l'informazione da trasmettere sia codificata in modo incrementale e sia trasmessa in pacchetti di diversa importanza. Si considera la trasmissione attraverso un canale binario, modellato tramite una catena di Markov che descrive l'intero sistema e che consente di ricavare le principali metriche di prestazioni. I risultati ottenuti vengono poi utilizzati per delineare le linee guida.

Capitolo 1

Introduzione

Alla trasmissione di informazioni attraverso un canale è legato il concetto di commutazione, inteso come meccanismo che permette la comunicazione tra la sorgente e la destinazione interessate dalla trasmissione. Ci sono diversi modi secondo cui può avvenire la commutazione, tra questi si può citare la commutazione di pacchetto [1], che prevede la suddivisione del contenuto informativo in piccole unità dati, chiamate pacchetti. È comune utilizzare questo tipo di commutazione, adattandola al considerare pacchetti non tutti uguali, anche per le comunicazioni multimediali. La trasmissione dei pacchetti, inoltre, può essere affetta da errore, infatti il canale stesso, attraverso il quale avviene lo scambio di informazione, può introdurre parte degli errori. Per questo motivo è necessario che ci siano dei meccanismi per il controllo d'errore che prevedano la rivelazione d'errore e la correzione d'errore nel caso in cui l'informazione trasmessa sia errata [1].

Il modello matematico proposto per il traffico multimediale usa una catena di Markov a tempo discreto di uno schema di trasmissione ARQ del tipo Selective Repeat. L'*Automatic Repeat reQuest*, ARQ, rappresenta una delle due principali tecniche usate per il controllo d'errore [1]. Questo protocollo prevede la correzione d'errore tramite la ritrasmissione del pacchetto sul quale si è verificata la rivelazione d'errore. L'altra tecnica usata, in alternativa, è la *Forward Error Correction*, FEC, che, a differenza dell'ARQ, cerca di correggere l'errore senza richiedere la ritrasmissione del pacchetto errato. L'ARQ inoltre si suddivide in tre tipologie: *stop and wait*, *go back-N* e *selective repeat*. Secondo quest'ultimo protocollo il trasmettitore ritrasmette solo ed esclusivamente le unità corrotte dagli errori, infatti, ogni qual volta si presenta una tale situazione le trasmissioni si interrompono per dare spazio alle ritrasmissioni e, avvenute queste, il processo di tra-

smissione riprende da dove si era fermato. In questo modo i dati corretti non occupano più tempo del dovuto nella trasmissione dell'intero contenuto informativo. Può capitare in alcuni casi, come per il traffico video, che si traggono vantaggi sia nell'utilizzare l'ARQ che nel servirsi della FEC, allora si parla di Hybrid ARQ [2], che altro non è che una combinazione di ARQ e FEC. I pacchetti relativi alle informazioni da trasmettere contengono della ridondanza, inserita dalla codifica di canale, utile per correggere parte degli errori introdotti durante la trasmissione lungo il canale (approccio FEC). Se gli errori da correggere sono troppi, il ricevitore chiede la ritrasmissione del pacchetto sbagliato (approccio ARQ).

Il traffico video attraverso una rete può essere pensato come la trasmissione di una sequenza di pacchetti indipendenti, così come è stato adottato dallo Joint Photographic Experts Group (JPEG) standard [3] nel caso degli M-JPEG, che rappresentano la trasmissione sequenziale di immagini JPEG. Osservando come è fatto il traffico video si è potuto constatare come questo consista di pacchetti che hanno importanza diversa e che, quindi, non sono tutti equivalenti [12]. Siccome una parte dell'informazione è autonoma e una parte è incrementale, se ne possono distinguere, di conseguenza, due diverse tipologie: *pacchetti di tipo A*, che rappresentano la parte principale dell'informazione da trasmettere e vengono codificati in modo indipendente dagli altri pacchetti, e *pacchetti di tipo B*, che invece hanno un'importanza secondaria poiché sono strettamente collegati alla corretta o errata trasmissione dei pacchetti A. La trasmissione di questi pacchetti può seguire lo schema dell' Hybrid ARQ, che è applicato in modo diverso a seconda se il pacchetto è di tipo A o di tipo B. Se i pacchetti A vengono ricevuti in modo errato ne viene richiesta la ritrasmissione, mentre se sono i B ad essere affetti da errore questi vengono semplicemente scartati.

Si possono inoltre distinguere due tipi di Hybrid ARQ: *type 1* e *type 2* [12]. Quest'ultimo prevede la trasmissione di un pacchetto attraverso un canale al quale sono associati L livelli di errore. Ad esempio per $L=5$ si hanno 5 livelli di errore e per il primo livello si può pensare di avere il 100% della correttezza riguardo all'informazione trasmessa, per il secondo livello il 75% di correttezza e procedendo così si arriva all'ultimo livello a cui corrisponde lo 0% di correttezza. In questi termini, se la prima volta che si trasmette l'informazione questa viene ricevuta affetta da errore si può, in un certo senso, compensare la

scorrettezza con una nuova trasmissione dello stesso pacchetto, affetta magari anch'essa da errore ma appartenente a un livello più basso, la cui correttezza si va a sommare alla percentuale corretta della trasmissione precedente. Nel caso preso in esame, però, viene considerato il classico canale binario con solo due livelli di errore, corretto o errato, 0 o 1. Ciò non toglie che sia possibile applicare la stessa analisi anche a sistemi in cui il canale si presenta con diversi livelli d'errore. La procedura trattata è dunque scalabile in quanto, ridimensionando gli strumenti utilizzati per realizzarla, è possibile estenderla a sistemi più complessi comprendenti anche un maggior numero di stati.

Si arriva infine a definire lo scopo di questo elaborato, che è quello di costruire, per il traffico video, un modello matematico, che verrà poi sfruttato per dare una descrizione dell'intero sistema considerato. In particolare questa descrizione avviene attraverso la definizione di 3 metriche fondamentali: il *throughput*, il *numero medio di trasmissioni* effettuate prima di ricevere correttamente un pacchetto A e la *probabilità di scarto* sempre solo di un A. Inoltre, i risultati ottenuti porteranno alla definizione di alcune importanti linee guida che permetteranno di capire come funziona e come migliorare il traffico video attraverso un canale wireless.

Il resto di questa tesi è strutturata come segue. Nel secondo capitolo vengono esposte trattazioni che hanno permesso, ponendo le basi, lo sviluppo dell'elaborato e trattazioni riguardanti argomenti simili che si concentrano sempre sul traffico video. Nel terzo capitolo viene descritto il modello in esame, vengono calcolate le probabilità stazionarie e conseguentemente le 3 metriche sopra citate. Nel quarto capitolo vengono raccolti in grafici e discussi i risultati ottenuti dall'analisi. In conclusione, con il quinto capitolo verranno messe in risalto le elaborazioni di maggiore importanza e le linee guida.

Capitolo 2

Related Work

Un sistema di comunicazione viene suddiviso, secondo il modello ISO-OSI, in 7 diversi livelli ciascuno con dei compiti specifici [1]. Uno di questi, il *data link layer*, è responsabile non solo del controllo degli accessi multipli alla rete per evitare collisioni tra i vari pacchetti informativi che la attraversano, ma anche dell'affidabilità della comunicazione attraverso la definizione di alcuni protocolli, tra cui l'ARQ. Sarà utile in un primo momento delineare a grandi linee le caratteristiche di questo protocollo, in modo poi da poter far riferimento a quanto detto per dare una descrizione accurata dei vari passi che hanno portato all'affermarsi dello studio del traffico multimediale, che oggi è molto diffuso, su una rete.

In [4] viene definito un modello matematico per il selective repeat ARQ, tramite una catena di Markov a tempo discreto con feedback non istantaneo e round trip delay assegnato, che permette l'analisi della statistica del ritardo dei pacchetti trasmessi. Lo scopo è quello di studiare l'impatto che ha il processo degli arrivi sul ritardo che interessa la trasmissione del contenuto informativo. Questo, che rappresenta il ritardo globale, è composto da diversi termini [5]; nell'analisi trattata da [4], però, viene suddiviso in due contributi: il *queueing delay*, che è il tempo speso dal pacchetto nel buffer della sorgente prima della prima trasmissione, e il *delivery delay*, che, invece, è il tempo che intercorre tra la prima trasmissione di un pacchetto e il suo essere poi pescato, una volta trasmesso, dal re-sequencing buffer del ricevitore, infatti trattandosi del selective repeat ARQ non è detto che i pacchetti vengano consegnati nell'ordine giusto. È bene notare che in realtà il delivery delay al ricevitore tiene conto anche dell'intervallo di tempo tra ciascuna trasmissione e la relativa ricezione, questo rappresenta il *ritardo di propagazione*

che va sommato dunque al delivery delay.

L'analisi, inoltre, si basa su delle assunzioni che permettono una semplificazione della trattazione. In particolare si assume che il processo degli errori sul canale sia indipendente e identicamente distribuito (*iid*), ciò comporta un legame tra la burstiness del canale e la probabilità di errore sul pacchetto. Importante è anche la *Heavy Traffic assumption*, assunzione secondo la quale il trasmettitore ha sempre qualcosa da trasmettere, cioè la coda di trasmissione è sempre piena. Grazie a quest'ultima assunzione si potrebbe non valutare il queueing delay nell'analisi della statistica in quanto si può assumere che la capacità del buffer di trasmissione sia sufficientemente elevata.

Un lavoro simile, fondato sulla Heavy Traffic assumption, è il [6], in cui ci si concentra solo sul delivery delay. Un ulteriore contributo, che generalizza questa analisi al caso in cui si considera un canale markoviano a N stati interessato da un più generale processo degli arrivi e che dà una descrizione di tutti i termini di ritardo, è dato da [7].

Si considera un approccio basato sul concetto di window per quanto riguarda il numero di trasmissioni che vengono fatte prima di conoscere il responso della prima trasmissione, il trasmettitore infatti deve aspettare fino alla fine della finestra prima di trasmettere nuovi pacchetti che, dunque, non passano attraverso il canale immediatamente dopo il loro arrivo. Il processo degli arrivi preso in esame è quello di Bernoulli, proprio come in [8]. Per concludere, un ulteriore contributo, che investiga anche sul queueing delay, può essere trovato in [9].

Con tutte queste trattazioni, e con tante altre qui non citate, si arriva alla descrizione statistica di un modello per l'ARQ che, in particolare in [4], permette di determinare le *probabilità stazionarie* del sistema in esame, da cui seguono poi le formulazioni dei vari ritardi. La loro importanza è fondamentale, perché grazie proprio ad esse si può descrivere e si possono ottenere tutte le informazioni possibili dal sistema.

Il modello ricavato da questa tesi si basa molto su ciò che è stato descritto precedentemente, da questo la sua importanza e il suo essere stato descritto così ampiamente, poiché lo studio della trasmissione di pacchetti video poggia su quello dell'ARQ relativo ad una normale trasmissione di contenuto informativo attraverso un canale. Ovviamente si tratta di applicare il modello dell'ARQ a un sistema più complesso in cui non tutti i pacchetti sono uguali e hanno la stessa importanza, ma il punto di partenza e le assun-

zioni di base sono analoghe. Si considera un sistema a tempo discreto con un time slot, cioè la durata di un pacchetto, che equivale al tempo di trasmissione di un pacchetto e si assume che i pacchetti A e B abbiano uguale tempo di trasmissione. Allo stesso tempo, però, questo elaborato va oltre a ciò che è stato fin qui descritto, perché permette di delineare anche delle linee guida che affermano come il modello ricavato sia molto efficiente. Infatti oggi il traffico video è in continua espansione. Secondo l'annuale rapporto Cisco sull'evoluzione del traffico dati su internet si stima un aumento di un fattore 4 nei prossimi anni delle comunicazioni multimediali. In internet o sugli smartphone si possono guardare, scaricare migliaia di video o ancora ci si può ad esempio videochiamare. Con il tempo lo studio del traffico multimediale ha assunto un aspetto sempre più dettagliato e specifico, conforme con il suo sempre più crescente sviluppo e mirato a trovare il giusto compromesso tra velocità di trasmissione dei diversi pacchetti e qualità del video. Molte sono le trattazioni su questo argomento, si passa dal definire solo il modello che è alla base ad elaborare il meccanismo di trasposto stesso dei vari pacchetti.

Lo scopo di questa tesi, dunque, non è solo quello di definire un modello, ma si propone anche di capire come è fatto il traffico video, infatti il considerare solo il modello di per sé non è molto interessante perché consiste di sole lunghe e noiose formule matematiche. È fondamentale, quindi, non solo risolvere il problema della modellizzazione dal punto di vista matematico, ma anche poter definire delle linee guida che permettano di capire come funziona il traffico. Nel caso specifico si scoprirà che considerare la ritrasmissione selettiva di alcuni pacchetti piuttosto che di altri, porta ad avere più efficienza per quanto riguarda il throughput, che implica dunque avere maggiore velocità nella trasmissione.

Questo lavoro riprende molto quello fatto in [10], in cui viene proposto un modello di Markov dello schema di trasmissione dei due diversi tipi di pacchetti basato sul protocollo del Selective Repeat Hybrid ARQ. Attraverso questo modello vengono poi definite due metriche: *throughput*, efficienza della trasmissione, e *goodput*, che rappresenta quanto di buono, cioè corretto, viene fatto passare attraverso il canale. Anche [11] dà un contributo significativo e segue una trattazione analoga che però ha l'obiettivo di capire il comportamento delle tecniche di controllo di errore applicate alla trasmissione video su canali wireless, definendo, tramite il confronto di diverse soluzioni, anch'essa delle linee generali per quel tipo di sistemi.

In [11], in primo luogo viene considerata una descrizione a basso livello del sistema, in cui vengono combinati il processo di generazione dei pacchetti video con il modello di un canale wireless, entrambi descritti da una catena di Markov; in secondo luogo, da una prospettiva ad alto livello, viene presentata l'evoluzione dell'intero sistema di trasmissione, sempre con una catena di Markov, descrivendo i processi di controllo d'errore che prevedono la ritrasmissione. Si introduce così una metodologia analitica per studiare le tecniche di controllo d'errore per la consegna di contenuti video e si arriva alla conclusione che per il traffico video è più efficiente l'utilizzo dell'Hybrid ARQ, nello specifico il type 2 che prevede più livelli di errore per il canale e che è ampiamente descritto in [12], piuttosto che l'utilizzo del puro schema ARQ senza considerare la distinzione dei pacchetti, che, seppure risulta applicabile, è molto meno efficiente. Viene inoltre effettuata una valutazione accurata sulla distorsione dei segnali video.

Un altro importante lavoro riguardante il traffico multimediale è il [13]. Questo paper ha esplicitamente l'intento di analizzare l'HARQ, frutto della combinazione di diversi elementi, per il traffico video. Anche in questo caso vengono distinti due tipi di pacchetti video e i risultati ottenuti non si basano su metodi analitici sviluppati, ma su delle simulazioni di trasmissioni video H.263 su un canale wireless con la protezione di errore prevista dall'HARQ.

In [14], tenendo presente che per trasmettere informazioni su di un canale bisogna usare sia la codifica di sorgente che di canale, viene presentata un'analisi teorica dell'errore quadratico medio nella *hybrid video coding* nel caso in cui la trasmissione è affetta da errore. Viene effettuata, ancora una volta, una valutazione analitica sul controllo di errore per il traffico video sul wireless. I risultati ottenuti sono stati verificati con un codificatore video H.263, viene però fatta solo un'analisi sul protocollo FEC, a differenza di questo elaborato in cui si prende in considerazione l'ARQ, e non c'è nessuna differenziazione esplicita tra i pacchetti. Risultati analoghi sono esposti in [15]. Importanti sono anche le conclusioni ottenute in [16] e in [17], che in particolare vengono anche ripresi da [11]. Il [17] tratta del mobile multicast in generale e non specificatamente al traffico video, ma, nonostante questo, è presente l'idea di applicare l'HARQ in una maniera non uniforme, simile al considerare diversi tipi di pacchetti per il flusso video. Un altro importante articolo collegato è [18], in cui si studia un metodo per poter ottimizzare il controllo

d'errore al livello applicazione considerando FEC e codifica di sorgente.

Importanti sono anche gli articoli che trattano dell'efficienza in termini di potenza nelle trasmissioni video. Ad esempio in [19] si introduce un approccio per la minimizzazione del consumo totale di potenza nel flusso video su un canale wireless, soggetto a un dato livello di qualità del servizio. Si trova che ottimizzando e regolando certi parametri questo consumo totale può ridursi sensibilmente. Anche in questo caso le affermazioni riportate in [19] sono verificate tramite dei risultati simulati. Una trattazione simile si presenta in [20], infatti si pone l'attenzione sul fatto che, se pur all'avanguardia, i portatili di oggi hanno tipicamente un'energia limitata per la computazione e per il trasferimento di informazioni in accordo con la breve vita della batteria che alimenta il dispositivo. La codifica, la decodifica e la trasmissione di contenuto video richiedono però grosse risorse, ed è per questo che in [20] viene fatta un'analisi su tutti questi aspetti in modo che si possa lavorare in condizione di minimo dispendio di potenza.

Questi sono solo alcuni dei papers che toccano l'importante analisi del traffico video o di argomenti collegati a questa, molti altri se ne potrebbero citare. Per concludere si può osservare che, nonostante quanto detto fino a questo punto sia a livello di trasposto, tutto si può traslare anche a livello MAC, in cui si considera solo il trasferimento di informazioni da un unico trasmettitore ad un unico ricevitore ed è dunque assente qualunque dimensione di rete.

Capitolo 3

Descrizione del modello

3.1 Descrizione del sistema e calcolo delle probabilità stazionarie

Si consideri un sistema costituito da un trasmettitore, un ricevitore e un canale binario lossy attraverso il quale si trasmette del contenuto multimediale. Quest'ultimo è suddiviso in pacchetti, i quali a loro volta sono di due tipi. Si può distinguere tra pacchetti indipendenti (tipo A), che sono codificati indipendentemente e che provengono da parti indipendenti dell'intera informazione da trasmettere, e pacchetti dipendenti (tipo B), che vengono codificati in modo diverso rispetto ai precedenti e vengono trasmessi dopo il loro relativo pacchetto A.

I pacchetti, quando arrivano al ricevitore, vengono processati con una codifica di canale orientata alla rivelazione d'errore, per esempio con un codice Cyclic Redundancy Check (CRC) [1]. Questo permette al ricevitore di inviare dei pacchetti di risposta, chiamati acknowledgement, che informano il trasmettitore sull'esito della trasmissione. Se il pacchetto di risposta è di tipo ack (positivo) allora l'informazione è stata trasmessa correttamente, se invece il pacchetto di risposta è di tipo nack (negativo) allora il pacchetto trasmesso è errato. Questi pacchetti di acknowledgement sono trasmessi su un canale di feedback e si suppone siano error free, cioè la loro trasmissione si può sempre considerare non affetta da errori. Il tempo impiegato per effettuare tutti questi passaggi prende il nome di *round trip time*, che è dunque dato dalla somma del tempo effettivo per trasmettere i pacchetti attraverso il canale, del tempo necessario a trasmettitore e a ricevitore per processare questi pacchetti, del ritardo di propagazione sul canale e del tempo necessario

per trasmettere i pacchetti di risposta ack e nack. Il relativo ritardo introdotto, round trip delay, viene indicato con m . In questo elaborato la window considerata consiste di 4 pacchetti trasmessi, quindi il valore di m è esattamente 4.

Si seguono due diverse procedure a seconda che il pacchetto affetto da errore sia A oppure B, sta infatti in questo la differenza rispetto al modello ottenuto studiando il puro protocollo ARQ. Nel primo caso ne viene richiesta la ritrasmissione, nel secondo caso il pacchetto viene scartato direttamente. Questo è dovuto alla diversa importanza dei due pacchetti, infatti i pacchetti A rappresentano il vero contenuto informativo e quindi, in caso di trasmissione errata, è necessaria la ritrasmissione. Inoltre si può aggiungere che è inutile ricevere correttamente i pacchetti B se i relativi pacchetti A non vengono trasmessi senza errori. A ciascun A fa riferimento e sussegue alla trasmissione un certo numero di pacchetti B. Nel caso preso in esame i pacchetti B associati a un A sono 2. Tutte queste trasmissioni implicano, all'interno del sistema, cambiamenti di stato secondo un certo valore del canale.

Il sistema considerato è rappresentato mediante una catena di Markov a tempo discreto, poiché questa si presta bene alla descrizione di molti sistemi di comunicazione. Essa permette infatti di determinare un accurato modello matematico che descrive tutto quello che governa l'intero comportamento del sistema. Innanzitutto sarà data una definizione di matrice di transizione del canale, in seguito, dopo aver delineato i principali componenti del sistema di trasmissione di traffico video, verrà messo in luce perché l'utilizzo di questo strumento si presta bene all'obiettivo di questa tesi e il modello che si trova alla base dell'analisi.

Una catena di Markov è un processo stocastico caratterizzato dalla cosiddetta proprietà di Markov, secondo la quale è possibile separare la descrizione dell'evoluzione futura del processo dal passato, dato per noto il presente. È, sostanzialmente, un sistema in cui la probabilità di uno stato i al tempo k è determinata univocamente dallo stato del sistema al tempo $(k-1)$ e non dalla storia dello stesso. Si può, quindi, definire una matrice \mathbf{P} , chiamata matrice di transizione del canale, fissato un tempo k :

$$\mathbf{P} = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix}$$

In questo caso, considerando un canale binario a soli due livelli di errore, la catena di

Markov consta di due soli stati: lo stato 0 che indica il canale buono e lo stato 1 invece che rappresenta il canale cattivo. La matrice \mathbf{P} , quindi, è una matrice 2×2 in cui l'elemento p_{00} è la probabilità del canale di passare all'istante k dallo stato 0 allo stato 0 tramite una transizione di stato (dunque è la probabilità di rimanere nello stato buono del canale), l'elemento p_{01} è la probabilità del canale di passare all'istante k dallo stato 0 allo stato 1, analogamente si può definire p_{10} la probabilità del canale di passare dallo stato 1 allo stato 0 e p_{11} la probabilità di rimanere nello stato 1. Il valore delle componenti della matrice \mathbf{P} è determinato a seconda della correlazione degli errori, che viene indicata con B , la burstiness. Infatti il canale può essere più o meno correlato: la correlazione può essere vista come indice di quanti pacchetti consecutivi trasmessi sono interessati da errore. Il caso con minore burstiness è quello in cui gli errori sono indipendenti e identicamente distribuiti, iid, si ha, così, che la probabilità di passare, in un certo istante, dallo stato buono allo stato cattivo coincide con la probabilità di rimanere nello stato cattivo e viene indicata con ε , probabilità di errore sul pacchetto, ed è indipendente dallo stato in cui ci si trova, mentre la probabilità di passare dallo stato cattivo allo stato buono coincide con la probabilità di rimanere nello stato buono ed è uguale a $(1 - \varepsilon)$. Se gli errori sono iid si può determinare anche il valore esatto della burstiness al variare di ε , questo è dato, infatti, da $B = 1/(1 - \varepsilon)$. Nel caso in cui gli errori non fossero iid resta comunque una relazione tra gli elementi di \mathbf{P} e ε , infatti, in generale, in questo elaborato si assume che $B = 1/p_{10}$ e $\varepsilon = p_{10}/(p_{01} + p_{10})$.

Una proprietà fondamentale della matrice di transizione del canale del sistema è che le righe sommano a 1, si tratta infatti di somma di probabilità. Inoltre la matrice \mathbf{P} permette di determinare le probabilità stazionarie, a cui ci si riferisce con il simbolo π , che rappresentano una sorta di equilibrio che il sistema raggiunge dopo un certo tempo tendente all'infinito. Le probabilità stazionarie caratterizzano l'intero sistema ed è proprio grazie a queste che è possibile poi analizzarlo e fornire degli strumenti che permettano di ricavare tutti i parametri descrittivi del modello. Ci sono diversi modi per determinare le π , a seconda ovviamente del numero di stati del sistema e delle transizioni tra uno stato e un altro possibili. Per questo calcolo e per analizzare il sistema e le varie metriche è stato usato un codice scritto in Matlab [21].

Si presenta ora una descrizione più approfondita del sistema considerato e il relativo

modello, che permetterà infine di ricavare le π . Questa descrizione è attuata grazie alla definizione di un vettore che dà tre tipi di informazioni sullo stato del sistema, tenendo presente che in tutti i sistemi di Markov lo stato deve esplicitare anche tutto ciò che comporta variazioni nelle transizioni. Nel caso preso in esame lo stato comprende: s che indica il valore del canale e che può assumere valore binario 0 se il canale è buono oppure 1 se il canale è cattivo, σ che invece dà informazioni sulla tipologia degli ultimi 4 pacchetti trasmessi e che assume un valore da 1 a 20, e infine un vettore \mathbf{b} di 4 elementi binari b_1-b_m , poiché $m = 4$, che indicano con lo 0 la correttezza del pacchetto trasmesso relativo a quella cella del vettore e con 1 invece il suo essere errato. In realtà la combinazione dell'ultimo elemento del vettore \mathbf{b} , b_m , e s rappresenta un unico parametro, chiamato c , che può avere 3 valori: vale 0 se il canale è buono e di conseguenza l'ultimo pacchetto trasmesso è corretto, vale invece 1 se il canale è cattivo ed è stato trasmesso un pacchetto che, se di tipo A, necessiterà poi di una ritrasmissione, e infine assume valore -1 se il canale è cattivo e non è stato trasmesso alcun pacchetto. Tutto questo si traduce nel codice Matlab nel considerare ciascuno degli stati scomposto in 4 componenti, che risultano utili per determinare le transizioni di stato: σ , b_1 che in base al suo valore determina la correttezza o meno del primo pacchetto della window data da σ , b_{23} che invece incorpora tutti i pacchetti della window esclusi il primo e l'ultimo trasmesso, e c . Facendo in modo che ci sia corrispondenza biunivoca è possibile concepire un algoritmo che permetta di passare dallo stato k -esimo alla scomposizione prima descritta. Non ha importanza come venga eseguito questo passaggio, l'importante è che ci sia biunivocità, quindi così come si è passati dallo stato alla scomposizione allo stesso tempo, in modo inverso, si deve poter passare dalla scomposizione al medesimo stato per descrivere le transizioni possibili.

Per quanto riguarda σ il suo valore ha un ruolo importante nelle transizioni. La trasmissione di un pacchetto A, generalmente, è seguita dalla trasmissione di un pacchetto B, ma questo solo se non c'è nessun pacchetto A vecchio che ha bisogno di una ritrasmissione. In questo caso infatti la ritrasmissione avrebbe la precedenza e prenderebbe il posto della trasmissione del pacchetto B che doveva avvenire e che, quindi, non avverrà più. Una volta esauriti i B_j relativi ad un certo A_j , si passa alla trasmissione di A_{j+1} e conseguentemente a quella dei due B_{j+1} . Ciascun pacchetto A ha a disposizione al più 3 trasmissioni, esclusa la prima volta che è stato trasmesso può, dunque, essere ritrasmesso

solo 2 volte. Se dopo la terza trasmissione è ancora affetto da errore viene semplicemente scartato, proprio come viene fatto per i pacchetti B non corretti. Considerato questo, partendo dallo stato in cui si ha una window di 4 pacchetti, di cui il primo è di tipo A_j , i due successivi sono i suoi relativi pacchetti B_j e l'ultimo è ancora un pacchetto A, A_{j+1} , e seguendo tutte le possibili transizioni che si possono avere a seconda che il pacchetto sia A o B e che ci sia bisogno di una ritrasmissione o meno, è possibile determinare il diagramma con tutte le transizioni che si possono avere. Il diagramma ottenuto è riportato nella Figura 3.1. Come si può notare il numero totale di transizioni possibili dettate da σ è 20. Nel diagramma in figura le frecce verdi indicano che il pacchetto A, primo elemento della window, è corretto, che corrisponde ad avere $b_1 = 0$, e non ha bisogno di ritrasmissione, mentre quelle rosse indicano che necessita di ritrasmissione in quanto errato, poiché $b_1 = 1$, e infine le frecce nere indicano una transizione forzata dovuta al fatto che il primo pacchetto della window o è di tipo B, e quindi non ha importanza indagare sulla sua correttezza, oppure è di tipo A alla terza trasmissione e dunque privo di ulteriori ritrasmissioni in caso di errore. Inoltre gli 0 indicano un pacchetto B, ovviamente alla sua prima e unica trasmissione, mentre 1, 2, 3 indicano un pacchetto A rispettivamente alla prima, seconda, terza trasmissione. Per poter usufruire delle importanti informazioni date da σ , nel codice Matlab il diagramma viene tradotto in una matrice 20×2 , qui riportata nella Tabella 3.1.

Si possono avere dunque $20 * 2^3 * 3$ stati nel sistema considerato, in totale cioè 480 stati. Infatti 20 sono i possibili valori assunti da σ , il vettore \mathbf{b} , privato dell'ultimo elemento la cui informazione viene incorporata in c , consta di sole 3 celle in cui può essere contenuto solo un valore binario e infine 3 sono i valori di c .

Prima di passare alla determinazione delle π , è opportuno capire come si effettuano le transizioni da uno stato all'altro all'interno del sistema. Innanzitutto si definisce la matrice \mathbf{T} come matrice di transizione tra gli stati. Si noti che \mathbf{T} non coincide con \mathbf{P} . Infatti, mentre \mathbf{P} permette di determinare le transizioni del canale, e quindi descrive il canale considerato, \mathbf{T} determina le transizioni tra i vari stati e mostra, a partire da uno stato, quale è la probabilità di passare in uno qualunque dei restanti 479 stati. Dunque \mathbf{T} è una matrice quadrata 480×480 che, inoltre, ha l'importante proprietà di essere sparsa e questo permette di poter risolvere il sistema con l'utilizzo di Matlab. Ovviamente, come

σ	espressione di σ	corretto	errato
1	1001	2	8
2	0010	3	3
3	0100	1	1
4	0103	5	5
5	1031	6	13
6	0310	7	7
7	3100	1	1
8	0012	9	9
9	0120	10	10
10	1201	11	15
11	2010	3	4
12	3120	10	10
13	0312	12	12
14	3103	5	5
15	2012	9	19
16	3123	20	20
17	2312	12	16
18	2310	7	14
19	0123	20	20
20	1231	18	17

Tabella 3.1: Tabella dei σ

si può intuire, queste transizioni dipendono dal valore delle 3 componenti che descrivono uno stato. Questo aspetto è molto importante perché chiarisce quale è il motivo che spinge ad usare il modello a catena di Markov e perché per avere uno stato e le relative transizioni si deve considerare l'intera tripletta $(\sigma \ b_1-b_{m-1} \ c)$ e non solo σ . In primo luogo per l'analisi non si vuole solo sapere la transizione data da σ ma anche se il canale, secondo cui avviene la trasmissione, è buono o cattivo. In secondo luogo deve valere la proprietà di Markov, il sistema dunque deve essere interamente rappresentato e deve dipendere unicamente dallo stato attuale, che deve concentrare la memoria del sistema. Una volta che sono noti σ , b_1-b_{m-1} , c al tempo t è immediato rendersi conto che si può determinare la probabilità di avere certi valori di σ , b_1-b_{m-1} , c all'istante $(t+1)$, in realtà i valori possibili sono 2. Infatti σ al tempo $(t+1)$ è determinato solo da σ e da b_1 , che certifica se è necessaria una ritrasmissione, all'istante t precedente; b_1-b_{m-1} al tempo $(t+1)$ dipende invece solo dal valore di b_2-b_{m-1} e da quello di c all'istante t (avviene, infatti, a causa di nuove trasmissioni o ritrasmissioni solo uno shift verso sinistra dei valori in $(b_1-b_{m-1} \ c)$); e infine c all'istante $(t+1)$ è dato univocamente da quello che era il valore di c sempre all'istante t . Con questo si dimostra appunto che il sistema è Markoviano e che lo stato istante per istante è determinato dal vettore (σ, b_1-b_{m-1}, c) .

Entrando nello specifico, la matrice \mathbf{T} si crea osservando che le transizioni possibili, dato un certo stato $(\sigma \ b_1-b_{m-1} \ c)$, sono 4 e sono verso:

- $\sigma_{corretto} \ b_2-b'_{m-1} \ 0$ con probabilità p_{c0}
- $\sigma_{corretto} \ b_2-b'_{m-1} \ 1$ con probabilità p_{c1}
- $\sigma_{errato} \ b_2-b'_{m-1} \ 0$ con probabilità p_{c0}
- $\sigma_{errato} \ b_2-b'_{m-1} \ 1$ con probabilità p_{c1}

dove si deve considerare o $\sigma_{corretto}$ oppure σ_{errato} , dati dalla tabella dei σ , a seconda del valore binario corrispondente a b_1 e b'_{m-1} rappresenta il vecchio valore del canale. Queste transizioni si ottengono considerando che la probabilità λ di avere un arrivo nel sistema sia pari a 1, quindi non si tiene conto di $c = -1$, poiché questo corrisponde a non avere arrivi. Si può notare, dunque, come gli elementi di \mathbf{T} altro non sono che le probabilità date dalla matrice \mathbf{P} a seconda che il canale, attraverso il quale avviene la transizione,

sia buono o cattivo. La matrice che segue è una realizzazione di \mathbf{T}

$$\mathbf{T} = \begin{pmatrix} a_{1,1} & \cdots & a_{1,25} & \cdots & a_{1,27} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & a_{1,480} \\ a_{2,1} & \cdots & \cdots & \cdots & \cdots & a_{2,37} & \cdots & a_{2,39} & \cdots & \cdots & \cdots & \cdots & a_{2,480} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{480,1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & a_{480,406} & \cdots & a_{480,408} & \cdots & a_{480,480} \end{pmatrix}$$

e si ha che $a_{1,25} = p_{00}$, $a_{2,37} = a_{480,406} = p_{10}$, $a_{1,27} = p_{01}$ e $a_{2,39} = a_{480,408} = p_{11}$, mentre $a_{1,1} = a_{2,1} = a_{480,1} = a_{1,480} = a_{2,480} = a_{480,480} = 0$ poiché non ci sono transizioni tra quegli stati.

Questo procedimento è determinante per il calcolo delle probabilità stazionarie $\pi(k)$ relative ai vari stati k . Queste, determinate le transizioni possibili, si trovano attraverso un approccio matriciale. Usare le matrici, infatti, implica una maggiore compattezza e chiarezza del codice. Nota la matrice \mathbf{T} del sistema, è possibile, dato un vettore $\boldsymbol{\pi}$ iniziale qualsiasi il cui contenuto deve sommare a 1, determinare, dopo un certo tempo elevato, quale sarà la probabilità del sistema di trovarsi in un certo stato piuttosto che in un altro. Si ha, infatti, che, per i sistemi di Markov, vale

$$\boldsymbol{\pi}\mathbf{T} = \boldsymbol{\pi} \quad (3.1)$$

da cui applicando l'inversione ad ambo i membri si ottiene

$$\mathbf{T}^T \boldsymbol{\pi}^T = \boldsymbol{\pi}^T \quad (3.2)$$

dove T indica l'operatore di trasposizione per matrici e vettori. Da questo, risolvendo il sistema composto da

$$(\mathbf{T}^T - \mathbf{I})\boldsymbol{\pi}^T = \mathbf{0} \quad (3.3)$$

e

$$\sum_{i=1}^{480} \pi_i = 1 \quad (3.4)$$

si ricava il vettore contenente tutte le $\pi(k)$ per ogni stato k .

3.2 Calcolo delle metriche

È già stata evidenziata, nel precedente paragrafo, l'enorme importanza delle probabilità stazionarie, in quanto esse permettono di determinare ulteriori importanti informazioni sul sistema. Tra queste si prendono in esame fondamentalmente 3 metriche:

- il throughput, che indica quanta informazione viene realmente "buttata" attraverso la rete ed è dunque una sorta di efficienza della trasmissione;
- il numero medio di trasmissioni fatte prima di ricevere correttamente un pacchetto A;
- la probabilità di scarto di un pacchetto A.

Per quanto riguarda il calcolo del throughput ci sono due diversi modi che si possono seguire, in entrambi i casi, però, bisogna tener presente che, trattandosi di un indice di efficienza della trasmissione, i vari contributi sono positivi e non nulli solo nel caso in cui la trasmissione del pacchetto A va a buon fine. Un metodo consiste nel calcolare direttamente il contributo al throughput di tutti i pacchetti nel caso in cui il primo pacchetto della window di un certo σ sia di tipo A. In questo modo si calcola subito il contributo dato dal pacchetto A e da i suoi relativi pacchetti B. Un secondo metodo, che è adottato in questo elaborato, prevede invece il calcolo del throughput relativo unicamente al primo pacchetto della window sia che esso sia di tipo A che di tipo B. Considerata una delle 20 raffigurazioni di stato in Figura 3.1, se il primo pacchetto della window è di tipo A e questo è stato trasmesso correttamente, allora si ha contributo positivo al throughput pari alla probabilità stazionaria di trovarsi in quello stato. Viceversa se il pacchetto A è affetto da errore si ha contributo nullo. Per i pacchetti B il meccanismo funziona in maniera leggermente diversa, bisogna considerare infatti che l'essere errato di un A priva di qualsiasi importanza la correttezza dei relativi B. Per questo motivo se il primo pacchetto della window è di tipo B si deve contare un contributo positivo al throughput solo nel caso in cui sia esso che il relativo pacchetto A vengano trasmessi correttamente. Questo implica un ulteriore problema: solo dell'esito del suddetto pacchetto B si sa tutto, infatti è sufficiente controllare il valore di b_1 , se il suo valore è 0 allora il pacchetto è corretto, in caso contrario è errato. Nell'eventualità che si presenti $b_1 = 1$ dunque non ha senso indagare sulla correttezza del pacchetto A di riferimento poiché il contributo è nullo, ma se invece si presenta il caso $b_1 = 0$, del pacchetto A si può sapere o direttamente se è corretto o errato, e allora è immediato determinare il contributo, oppure l'informazione a disposizione non è sufficiente e bisogna fare una stima. Seguiranno ora alcuni esempi che chiariranno i concetti appena esposti.

Si consideri $\sigma = 0010$. Il primo pacchetto è di tipo B e, guardando la tabella dei σ , si sa già che il relativo pacchetto A è corretto perché, se così non fosse, si sarebbe avuta una ritrasmissione di quel pacchetto A e ci si troverebbe in $\sigma = 0012$. Il contributo al throughput, dunque, è non nullo solo se $b_1 = 0$, nel qual caso il pacchetto B è corretto.

Si consideri ora $\sigma = 0012$. Il primo pacchetto è ancora di tipo B ma, apparentemente, nulla si sa sulla correttezza del pacchetto A di riferimento. Si intuisce, guardando sempre la Tabella 3.1, che precedentemente il pacchetto A non era corretto e ciò ha implicato la sua ritrasmissione (nello specifico la sua seconda trasmissione). Si può però "sbirciare" il suo stato attuale, che viene dato dall'ultimo elemento del vettore \mathbf{b} . Se il suo valore è 0 allora si sa che il pacchetto A considerato risulterà poi corretto e se anche il pacchetto B è corretto allora il contributo sarà positivo, altrimenti necessiterà di una nuova ritrasmissione e quindi si deve fare una stima su quello che potrebbe succedere dopo m , cioè 4, trasmissioni quando si presenterà l'esito della nuova ritrasmissione. In pratica quando il pacchetto A in gioco verrà trasmesso per la terza e ultima volta si hanno due casi: il pacchetto viene ritrasmesso correttamente e il contributo sarà positivo, sempre se anche il tipo B è corretto, altrimenti si avrà contributo nullo non avendo altre trasmissioni a disposizione. In poche parole, se si è in 0010 e il pacchetto B è corretto allora il contributo al throughput sarà sicuramente pari a $\pi(k)$, altrimenti nullo. Se si è nello stato con $\sigma = 0012$ e il pacchetto B non è corretto, il contributo è nullo, se il pacchetto B è invece corretto e se anche A lo è il contributo è pari a $\pi(k)$, ma se A non è corretto allora il contributo sarà frazionario, nel caso specifico sarà pari alla probabilità che dopo l'ultima ritrasmissione A sia corretto. Per calcolare questa probabilità si possono prendere in esame tutte le possibili transizioni oppure notare che, in maniera più semplice, è sufficiente considerare il giusto elemento della matrice \mathbf{P}^4 , che dà le probabilità di avere un certo valore del canale dopo la terza trasmissione. L'elemento da considerare è quello che si trova in seconda riga e prima colonna, poiché interessano appunto le transizioni che portano dall'avere un pacchetto errato ad averne uno corretto che non ha più bisogno di essere ritrasmesso. Quindi il contributo da considerare è il valore di quell'elemento moltiplicato per la $\pi(k)$ relativa a quello specifico stato. Ovviamente i due diversi procedimenti usati negli esempi si possono estendere a tutti quei valori di σ al variare dello stato k che permettono di fare le stesse considerazioni.

Una variante, simile ma non del tutto uguale, dell'ultimo esempio è il caso in cui $\sigma = 0120$ (analogo è il discorso per $\sigma = 0123$). Ancora una volta il primo pacchetto della window è B, ma ora il pacchetto A relativo si trova nella penultima cella del vettore \mathbf{b} ed è alla seconda trasmissione. Quindi si procede come prima, solo che in questo caso per fare la stima non si deve considerare \mathbf{P}^4 ma \mathbf{P}^3 e soprattutto bisogna considerare che si ha contributo non nullo sia che l'ultimo elemento del vettore \mathbf{b} contenga uno 0 sia che contenga un 1, ammesso che il pacchetto B sia corretto. Così si avrà che se $b_1 = 0$ e $b_4 = 0$ il contributo è dato dal prodotto tra l'elemento di prima riga e prima colonna di \mathbf{P}^3 , poiché la transizione di stato avviene con canale buono, e la relativa $\pi(k)$, mentre se $b_1 = 0$ e $b_4 = 1$ allora il contributo è il prodotto tra l'elemento di seconda riga, poiché il canale è cattivo, e prima colonna e sempre la $\pi(k)$ dello stato k considerato. In tutti gli altri casi si avrà contributo nullo.

I tre esempi commentati esaurisco tutti i diversi casi che si possono presentare per il calcolo dei contributi al throughput con un B come primo elemento della window. Il valore del throughput medio si ottiene sommando tutti i vari contributi calcolati per ogni k , che varia tra 1 e 480. Nel prossimo capitolo, alla luce dei risultati ottenuti, si porrà l'attenzione su particolari considerazioni relative al throughput.

Si passa ora alla descrizione del ragionamento che ha portato alla definizione e al calcolo del numero medio di trasmissioni necessario prima di ricevere correttamente un pacchetto A.¹ Calcolate le probabilità stazionarie dei vari stati, come si è potuto notare anche per il calcolo del throughput medio, si possono ottenere le probabilità stazionarie di eventi che raggruppano diverse cose che possono capitare nel sistema sommando valori specifici di $\pi(k)$. Ad esempio, se si considera $\pi(\text{pacchetti A})$ come la probabilità PASTA² con cui è possibile che entrando nel sistema ci si trovi davanti alla trasmissione corretta di un pacchetto A, questa si ottiene sommando le $\pi(k)$ relative a tutti i σ in cui il primo pacchetto della window è A e contemporaneamente il valore di b_1 è 0, che denota la correttezza del pacchetto trasmesso. Se, considerando sempre lo stesso esempio, non

¹In questo elaborato si prende in esame solo il numero di trasmissioni dei pacchetti A poiché, come più volte ripetuto, questi hanno maggiore importanza. Ovviamente lo stesso discorso e lo stesso procedimento può essere utilizzato se si volessero prendere in considerazione i pacchetti B.

²La proprietà PASTA (Poisson Arrivals See Time Averages) è di fondamentale importanza e vale per i sistemi di Markov. Questa permette la descrizione in termini probabilistici dei vari sistemi di comunicazione e una sua spiegazione accurata si può trovare in [18].

ci fosse distinzione tra i pacchetti A e i B, allora la $\pi(\text{pacchetti A})$ coinciderebbe con il throughput. Nel caso specifico, poiché il protocollo alla base del modello studiato è Selective Repeat ARQ, questo calcolo risulta estremamente semplice rispetto a quello che verrà trattato. Infatti per il SR ARQ il throughput è dato da $(1 - \varepsilon)$ e rappresenta il comportamento migliore possibile da parte del sistema poiché ritrasmette in modo selettivo i pacchetti a seconda che siano corretti o errati. In realtà, trattandosi di probabilità condizionate all'avere il pacchetto A corretto, si ha che

$$\frac{\pi(\text{pacchetti A})}{\pi(\text{tutti A})} = 1 - \varepsilon \quad (3.5)$$

dove $\pi(\text{tutti A})$ è data da tutti i contributi di quei σ con un A come primo elemento della window. Però per il calcolo del throughput si tiene conto anche dei pacchetti B e questo ha portato a un'altra serie di calcoli e considerazioni diverse.

Ritornando al numero medio di ritrasmissioni, note le $\pi(k)$ si è in grado di determinare le seguenti probabilità condizionate:

- $\pi(1|A)$ probabilità che entrando nel sistema si trova un pacchetto A alla prima trasmissione;
- $\pi(2|A)$ probabilità che entrando nel sistema si trova un pacchetto A alla seconda trasmissione;
- $\pi(3|A)$ probabilità che entrando nel sistema si trova un pacchetto A alla terza trasmissione;

Essendo delle probabilità condizionate, queste si trovano come rapporto tra la somma di tutte le $\pi(k)$ per ogni stato k con un A in principio che è alla prima (seconda o terza a seconda che si voglia calcolare $\pi(1|A)$, $\pi(2|A)$ o $\pi(3|A)$) trasmissione e la somma di tutte le $\pi(k)$ con k relativo a un σ con un A come primo pacchetto della window. Queste non sono esattamente le probabilità che servono per ciò che si vuole determinare, ma sono utili perché permettono di calcolare le probabilità che quando arriva un pacchetto A questo si faccia una, due o tre trasmissioni prima di essere trasmesso correttamente, e dunque consegnato, o scartato. Proprio questi parametri incidono nel calcolo del numero di trasmissioni di un pacchetto A; infatti è evidente che le due probabilità coincidono,

basti pensare al fatto che se un pacchetto A viene trasmesso una volta non è detto che sia l'ultima volta che effettui una trasmissione perché può risultare in errore e quindi bisognoso di una nuova trasmissione.

In generale, indicando con p_1 la probabilità che il pacchetto A venga trasmesso esattamente una volta prima di essere consegnato, p_2 la probabilità che A venga trasmesso esattamente due volte e p_3 quella che indica che il pacchetto A è stato trasmesso esattamente 3 volte prima di essere consegnato, si ha che

$$\pi(1|A) = \frac{p_1 + p_2 + p_3}{p_1 + 2p_2 + 3p_3} \quad (3.6)$$

$$\pi(2|A) = \frac{p_2 + p_3}{p_1 + 2p_2 + 3p_3} \quad (3.7)$$

$$\pi(3|A) = \frac{p_3}{p_1 + 2p_2 + 3p_3} \quad (3.8)$$

Per capire il perché di queste equazioni basti pensare al senso fisico, infatti un pacchetto A che fa tre trasmissioni viene visto tre volte nel sistema, cioè quando è alla prima, alla seconda e alla terza trasmissione. Analogamente un pacchetto che viene trasmesso due volte viene visto due volte nel sistema quando appunto è alla prima e alla seconda trasmissione, e un pacchetto che è trasmesso una volta viene visto una sola volta. Il denominatore delle 3 equazione discusse rappresenta proprio il numero medio di trasmissioni fatte da un pacchetto A prima di essere consegnato. Per determinare questa incognita dunque è sufficiente invertire le formule in modo da esplicitare i 3 parametri incogniti p_1 , p_2 e p_3 . Nel codice Matlab invece, una volta considerati i contributi a seconda di σ e calcolate dunque le varie probabilità menzionate sopra, è stato usato un procedimento matriciale.

Infine si tratta ora il caso della probabilità di scarto di un pacchetto A. Ricordando che un A viene scartato solo quando, dopo la terza trasmissione, è ancora affetto da errore, la descrizione di questa metrica è alquanto semplice. La probabilità di scarto è determinata univocamente da tutti quei σ che presentano un pacchetto A alla terza trasmissione come primo elemento della window e che hanno il relativo b_1 pari a 1, indicando così che il pacchetto è ancora errato: *solo* in questo caso il pacchetto in questione viene scartato.

Allora è sufficiente considerare i contributi in termini di $\pi(k)$, per ogni k , riguardanti solo i σ dei casi menzionati sopra e poi sommarli per avere il valore numerico della probabilità di scartare A .

Per concludere è utile introdurre ciò che verrà approfondito attraverso i risultati nel prossimo capitolo: si trova che tutte le metriche descritte variano al variare di ε e della burstiness del canale.

Capitolo 4

Risultati

In questa sessione vengono riportati i risultati ottenuti sfruttando l'analisi precedentemente fatta. In un primo momento verrà discusso come variano il throughput, il numero medio di trasmissioni per i pacchetti A e la probabilità di scartare questi al variare di ε e della burstiness.

In Figura 4.1 si prende in esame il throughput. I valori di ε considerati sono (0.01 0.02 0.03 0.05 0.07 0.1 0.15 0.2 0.25 0.3), mentre quelli della burstiness ($Biid$ 5 10 20). Si trova che per il valore di ε non ha senso andare sopra a 0.3, così come è sufficiente prendere come limite inferiore 0.01, in quanto per valori di probabilità più basse il throughput verrebbe sempre unitario. Inoltre, per quanto riguarda B il caso migliore si ha quando gli errori sono iid, nel qual caso B dipende da ε . Poiché per $\varepsilon = 0.3$ si ha che $Biid = 1.42$, si può affermare che per $B > 2$ il canale è significativamente più correlato rispetto al caso $Biid$. In poche parole, quando $B = Biid$ non c'è correlazione, quando $B = 5$ c'è correlazione confrontabile con la lunghezza della window, con $B = 10$ la correlazione è più lunga e con $B = 20$ è molto più lunga della window (5 volte il valore della window).

Rappresentando l'efficienza della trasmissione, il valore massimo che può assumere il throughput è 1, che si ha quando la probabilità di errore sul pacchetto è nulla. Idealmente all'aumentare di ε , si pensa che il throughput dovrebbe crollare ed assumere valori sempre più bassi. Il grafico 4.1 valorizza questa ipotesi. Si può notare che all'aumentare di ε , nei vari casi corrispondenti ai diversi valori di B considerati, il throughput diminuisce, infatti il contributo al throughput di un pacchetto è nullo quando questo viene trasmesso non correttamente. A parità di ε , invece, il throughput aumenta all'aumentare di B , escludendo per ora il caso iid, perché infatti per valori elevati di B il canale è come se

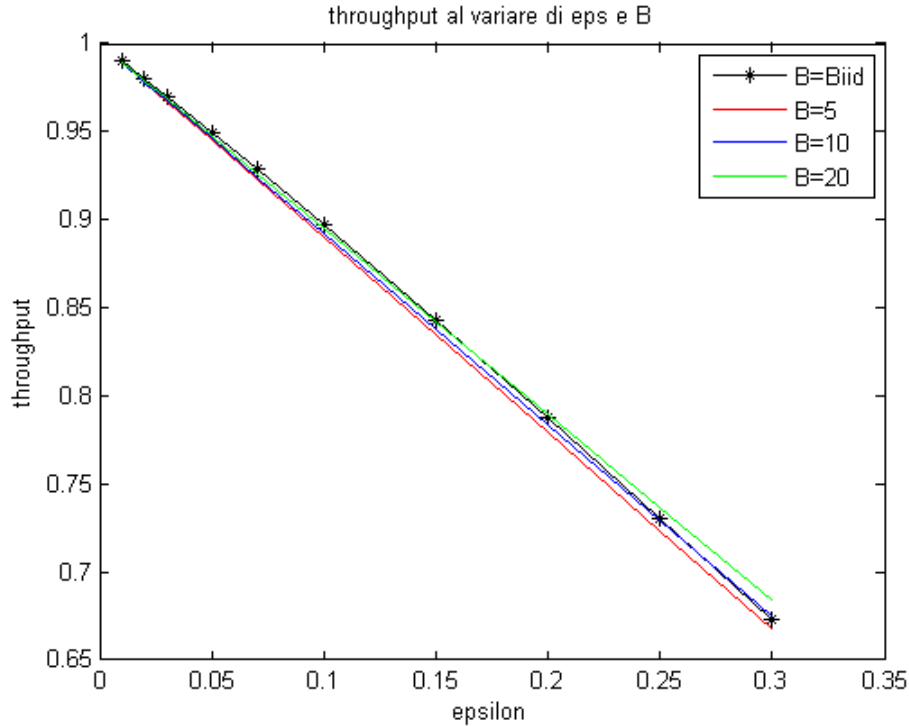


Figura 4.1: Il throughput al variare della probabilità d'errore e della burstiness

fosse suddiviso in grandi blocchi, per alcuni di essi le trasmissioni relative sono affette da errori, per altri invece avvengono secondo un canale buono dando un grosso contributo al throughput, questo aspetto si vede meglio in Figura 4.5. Qualcosa in più si può dire sul caso iid: dal grafico 4.1 emerge che non è detto che per valori di B più piccoli si abbia un throughput minore. Ad esempio per $\varepsilon = 0.3$ nel caso iid si ha che $B = 1.42$ e se si confronta il valore corrispondente assunto dal throughput con quello assunto per $B = 5$ si osserva che il primo ha un valore maggiore. Ciò non è solo dovuto al fatto che il caso iid rappresenta il caso migliore (perché se così fosse allora si dovrebbe avere un valore più grande anche di quello relativo a $B = 20$ ma, a causa del discorso fatto prima sui blocchi, così non è), ma è dovuto all'avere, per $B = 5$, l'intera window di pacchetti trasmessi su un canale correlato. Questo si può notare bene anche nella Figura 4.2, in cui viene graficato il throughput normalizzato rispetto a $(1 - \varepsilon)$. Partendo dal presupposto che questo non può assumere valori superiori a $(1 - \varepsilon)$, si nota che se B è elevato, eccetto sempre nel caso iid, il throughput ha un valore più grande. Avendo un valore di throughput più vicino ad 1 si velocizza la trasmissione, perché non si perde molto tempo per le ritrasmissioni, ma

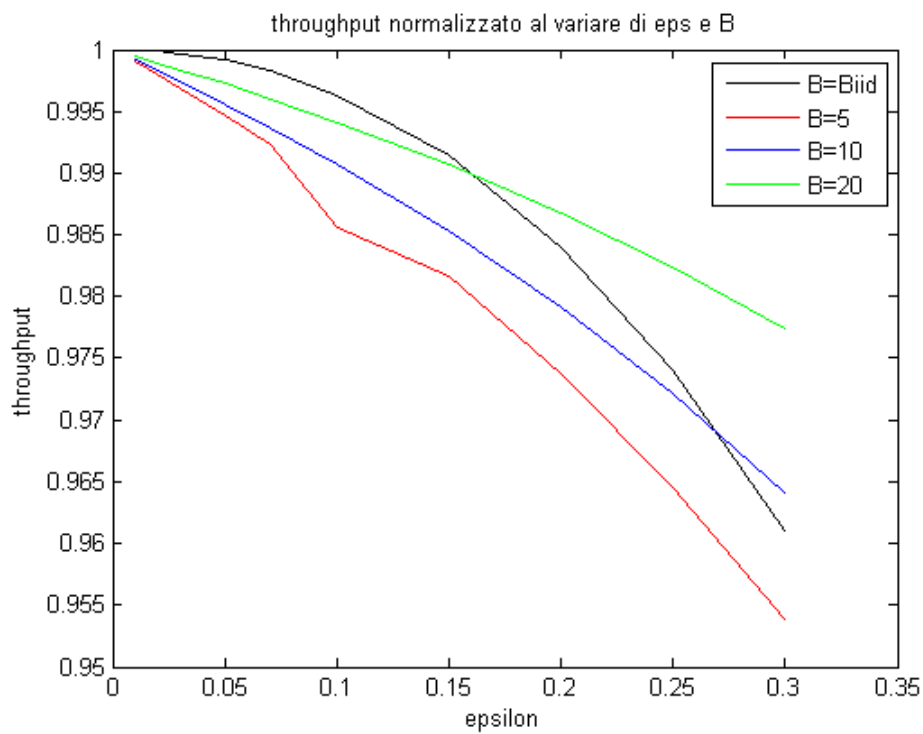


Figura 4.2: Il throughput normalizzato al variare della probabilità d'errore e della burstiness

allo stesso tempo avviene una perdita in termini di qualità del video in quanto si possono perdere alcune delle immagini principali corrispondenti a quei pacchetti A trasmessi in modo errato o addirittura scartati a causa della elevata burstiness. Allora per ovviare a questo problema si può pensare di usare una tecnica diversa da quella adottata. Inoltre, nel caso iid, se non ci fosse distinzione tra i pacchetti A e B, poiché il protocollo usato è Selective Repeat ARQ, si avrebbe che

$$\text{throughput} = 1 - \varepsilon. \quad (4.1)$$

Considerando, però, la distinzione presentata, al valore precedentemente citato va sommato il contributo dato dalla presenza dei pacchetti B. Infatti per i pacchetti A il contributo al throughput è esattamente $(1 - \varepsilon)$, mentre per i B è $(1 - \varepsilon)^2$, poiché sono due i B associati ad un A. Allora, nel caso in cui non ci siano ritrasmissioni, si può concludere che

$$\text{throughput} = \frac{1}{3}(1 - \varepsilon) + \frac{2}{3}(1 - \varepsilon)^2. \quad (4.2)$$

Questo rappresenta il valore del throughput medio, che è dato semplicemente dalla media pesata dei contributi dei pacchetti A e dei B, dove $\frac{1}{3}$ è la probabilità che si presenti un A e $\frac{2}{3}$ è la probabilità che si presenti un B. In realtà, per quanto detto fino a qui, si devono considerare anche le ritrasmissioni, allora il risultato cambia. Per quanto riguarda la non ritrasmissione del pacchetto A in quanto corretto, la probabilità che si presenti un A e quella che si abbia invece un B sono esattamente le stesse discusse nel caso in cui non si abbiano mai ritrasmissioni. Se invece si ha la ritrasmissione di un pacchetto A allora le probabilità cambiano. Si ha infatti che la probabilità di avere un pacchetto A trasmesso 2 volte è $(\frac{1}{3} + \frac{1}{3}\varepsilon)$, mentre quella di avere un suo relativo B è $(\frac{2}{3} - \frac{1}{3}\varepsilon)$. Allo stesso modo se si considera che il pacchetto A venga trasmesso 3 volte, la probabilità che si abbia un A è $(\frac{1}{3} + \frac{1}{3}\varepsilon + \frac{1}{3}\varepsilon^2)$ e quella di avere un B è $(\frac{2}{3} - \frac{1}{3}\varepsilon - \frac{1}{3}\varepsilon^2)$. Chiamata p_A la probabilità di avere un pacchetto A trasmesso 3 volte e p_B la probabilità di uno dei B di quel pacchetto A, si può concludere che il valore del throughput medio è ancora la media pesata, con probabilità però diverse dal caso precedente, tra il contributo di A e quello di B:

$$\text{throughput} = p_A(1 - \varepsilon) + p_B(1 - \varepsilon^3)(1 - \varepsilon)^2. \quad (4.3)$$

In Figura 4.3, invece, si considera come varia il numero di trasmissioni da fare per i pacchetti A, prima di essere consegnati o scartati, al variare di ε e di B , si noti che i

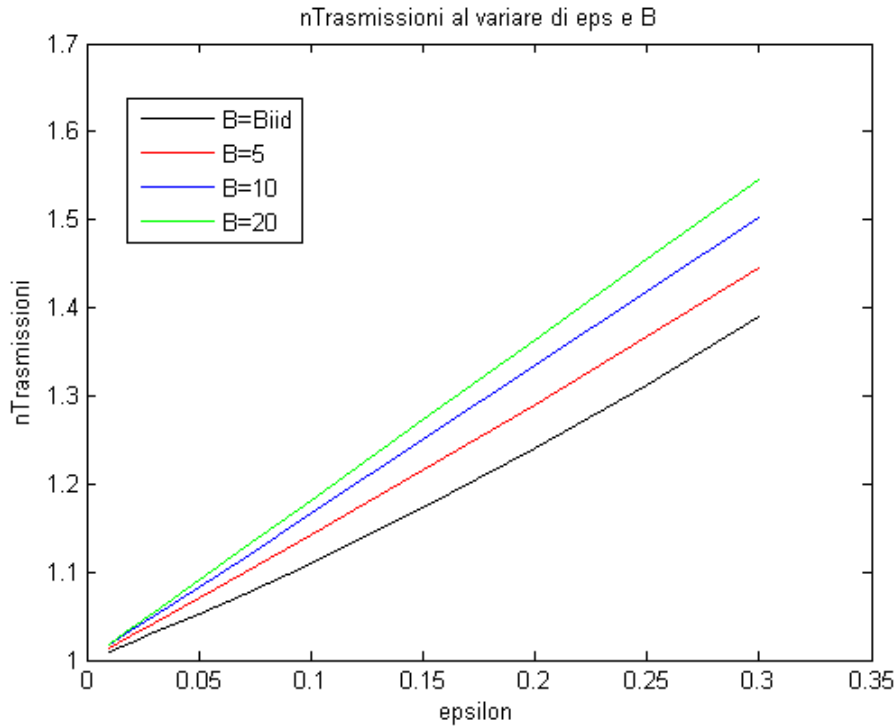


Figura 4.3: Il numero medio di trasmissioni per i pacchetti A al variare della probabilità d'errore e della burstiness

diversi valori sono ancora quelli usati per determinare la Figura 4.1 e la Figura 4.2. Dal grafico, come è logico che sia, si può constatare che, all'aumentare di ε , il numero medio di trasmissioni da fare aumenta. Questo ha senso perché, ovviamente, se c'è maggiore probabilità che un pacchetto venga trasmesso in modo errato, allora segue che saranno necessarie più ritrasmissioni, sempre al più 2, prima che venga consegnato al ricevitore correttamente. Anche all'aumentare di B si ha che il numero medio di trasmissioni, a parità di ε , aumenta. Per dimostrare la correttezza di questa affermazione è sufficiente ricordare cosa rappresenta la burstiness. Infatti, poiché B dice quanti pacchetti trasmessi consecutivamente sono affetti da errore, maggiore è il numero di questi pacchetti maggiore sarà la probabilità di "beccare" un pacchetto A errato, di conseguenza maggiore sarà il numero di trasmissioni necessarie affinché quel pacchetto venga poi consegnato correttamente. Si può inoltre notare che, anche considerando un valore della burstiness molto alto, è sufficiente che un pacchetto A venga trasmesso al più una volta e mezza prima di essere ricevuto senza errori e che il ritardo, dovuto alla ritrasmissione, è comunque

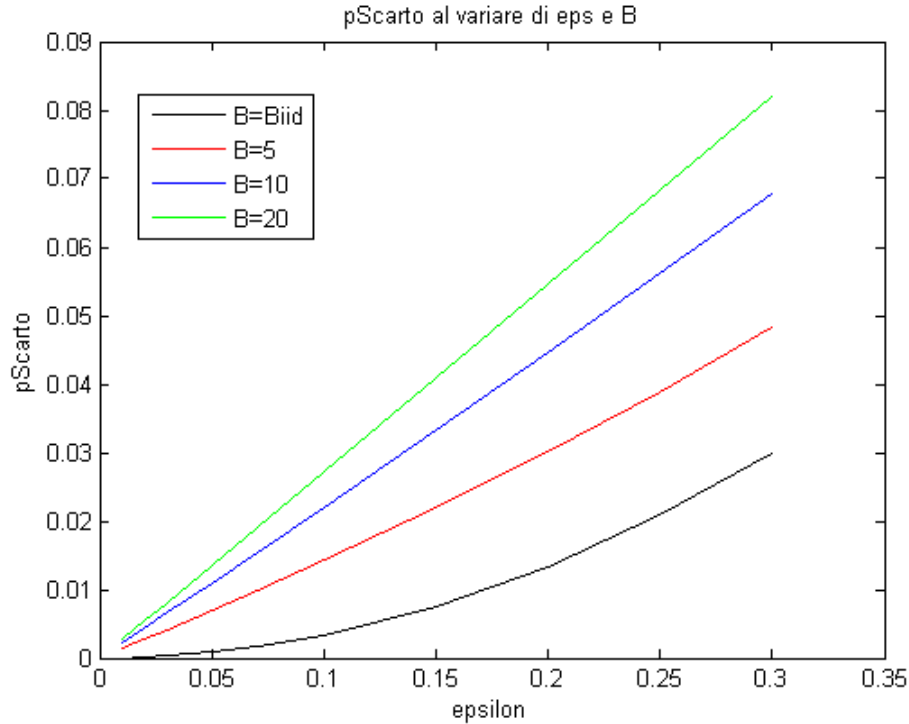


Figura 4.4: La probabilità di scarto di un pacchetto A al variare della probabilità d'errore e della burstiness

contenuto. Questo perché, in un certo senso, il fatto che la ritrasmissione di un pacchetto A prenda il posto della trasmissione del pacchetto B che doveva avvenire, compensa il fatto stesso che sia avvenuta una ritrasmissione.

Infine, per concludere questa prima trattazione dei risultati ottenuti, si prende in esame la probabilità di scarto di un pacchetto A al variare di ϵ e B , ancora una volta negli stessi intervalli di valori già usati. Come si osserva dalla Figura 4.4, per la probabilità di scarto di A si possono fare le stesse considerazioni fatte per il numero medio di trasmissioni necessarie ad un pacchetto A. Non dovrebbe stupire questa assunzione in quanto c'è un legame tra le 2 metriche: un pacchetto A viene scartato proprio quando, dopo la terza trasmissione, risulta ancora errato. Allora segue che, così come un aumento di ϵ o di B comporta un aumento del numero medio di trasmissioni, allo stesso modo comporta anche un aumento della probabilità di trovare un pacchetto A alla terza trasmissione e quindi di scartarlo perché affetto da errori avendo ϵ e B valori via via più grandi. Nello specifico caso iid, si ha che il valore della probabilità di scarto è esattamente ϵ^3 , proprio

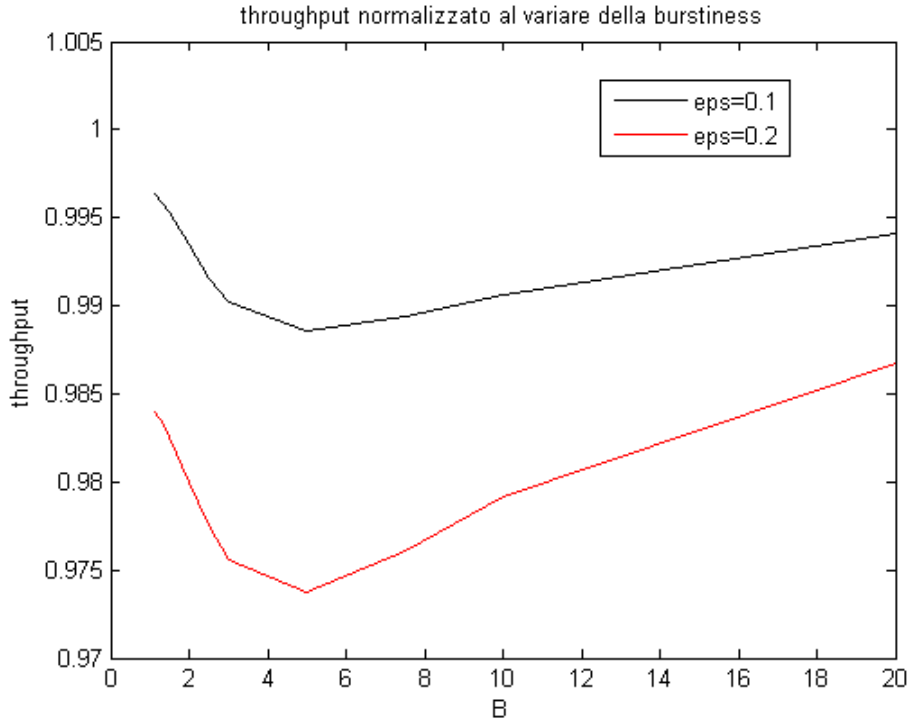


Figura 4.5: Il throughput al variare della burstiness per $\epsilon=0.1$ e $\epsilon=0.2$

perché tutte e 3 le trasmissioni non devono andare a buon fine.

Si passa ora a discutere i risultati ottenuti per le 3 metriche considerando due valori fissi per ϵ , cioè $\epsilon=0.1$ e $\epsilon=0.2$, e facendo variare B in un certo intervallo di valori. Per $\epsilon = 0.1$ B varia nel seguente slot di valori (1.11111 1.35 1.5 2 2.5 3 5 7.5 10 20), mentre per $\epsilon = 0.2$ B varia in (1.25 1.35 1.5 2 2.5 3 5 7.5 10 15 20), dove il primo valore di entrambi gli intervalli corrisponde al caso iid. Per quanto riguarda il throughput viene presentato direttamente il grafico del valore normalizzato rispetto a $(1 - \epsilon)$, in quanto questo permette di mettere bene in luce quanto verrà detto.

Si può notare in Figura 4.5 che inizialmente il throughput decresce all'aumentare di B , finché, raggiunto un valore compreso tra 4 e 6, riprende a salire per saturare verso $(1 - \epsilon)$. C'è, dunque, un picco verso il basso in corrispondenza di B pari a un valore intermedio tra 4 e 6. Questo risultato è sensato in quanto si ha che l'intera window di pacchetti trasmessi con quel valore di B è sempre interessata da un canale correlato, quindi il contributo al throughput è minore perché l'intera window può essere affetta da errore. Il motivo secondo il quale poi, per valori di B superiori, il throughput ricomincia

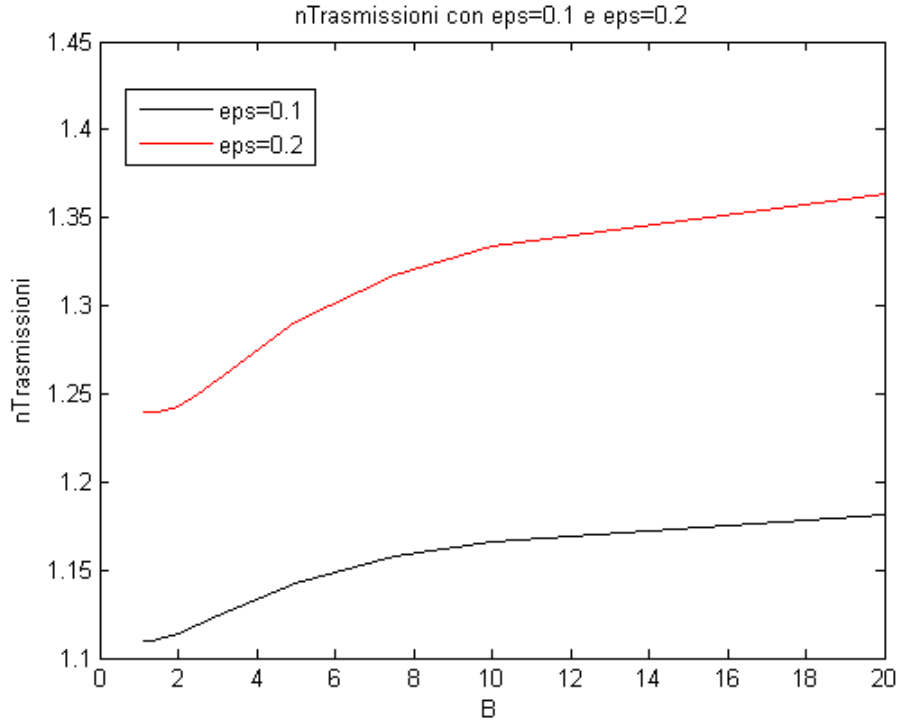


Figura 4.6: Il numero medio di trasmissioni per un pacchetto A al variare di B con $\epsilon=0.1$ e $\epsilon=0.2$

a salire è esattamente quello discusso riguardo all'avere un canale suddiviso in grandi blocchi. In Figura 4.5 si può osservare ancora una volta come, in generale, per valori di ϵ più grandi si abbia che il throughput assume valori minori.

La Figura 4.6 e la Figura 4.7, rispettivamente, considerano l'andamento del numero di trasmissioni che interessano un pacchetto A e della probabilità che questo venga scartato, sempre al variare di B e per $\epsilon = 0.1$ e $\epsilon = 0.2$.

Nuovamente, i due grafici possono essere descritti allo stesso modo e si arriva alle medesime conclusioni discusse per le Figure 4.3 e 4.4. Fissati i due valori di ϵ , si può notare meglio l'andamento monotono crescente non lineare delle 4 curve rispetto a B . La velocità di crescita del numero medio di trasmissioni e della probabilità di scarto non è dunque uniforme: per valori di B maggiori di circa di 8, nel caso delle trasmissioni, e maggiori circa di 10, per la probabilità di scarto, la crescita, all'aumentare di B , acquista un ritmo più lento. Questo può essere ancora dovuto alla suddivisione in blocchi del canale in presenza di un valore elevato di burstiness. Infatti per i pacchetti trasmessi in

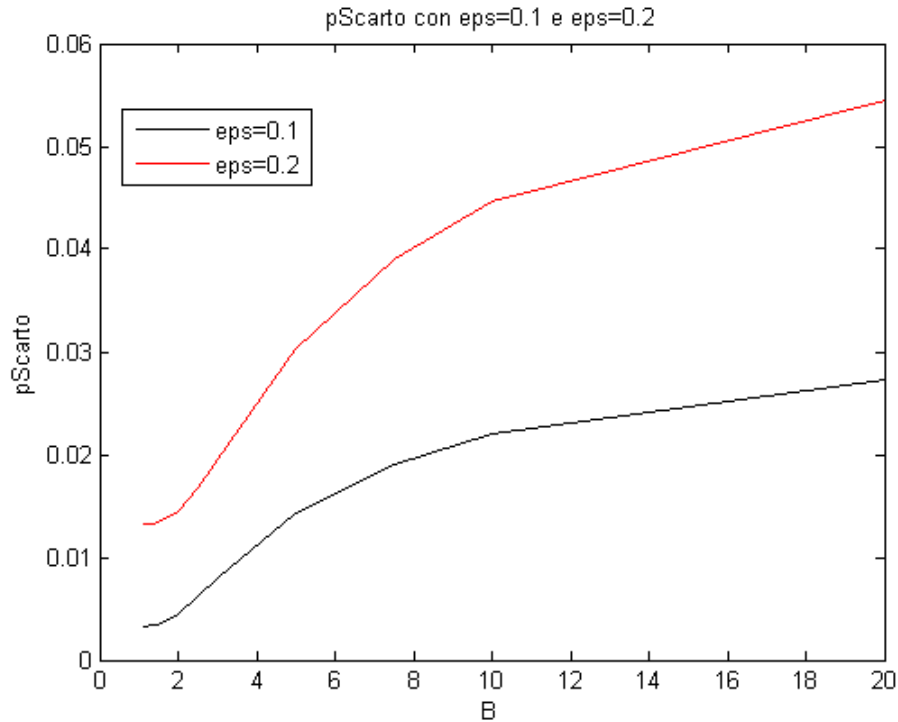


Figura 4.7: La probabilità di scarto di un pacchetto A al variare di B con $\epsilon=0.1$ e $\epsilon=0.2$

quei blocchi che sono interessati da canale buono non saranno necessarie ritrasmissioni e di conseguenza sarà anche piccolo il valore della relativa probabilità di scarto, questo ovviamente compensa il grande valore invece assunto dalle 2 metriche nei blocchi in cui il canale si presenta molto correlato. Infine, come è ovvio che sia, nel caso corrispondente a $\epsilon = 0.2$ il numero medio di trasmissioni da fare, così come la probabilità di scartare un pacchetto A, è significativamente maggiore rispetto al caso con $\epsilon = 0.1$.

Si può concludere affermando dunque che queste due ultime figure descrivono, specificamente in relazione a due valori fissati di ϵ , quanto già detto per le metriche in questione.

Capitolo 5

Conclusioni

In questa tesi è stata presentata un'elaborazione analitica, basata sulla teoria dei processi markoviani, per studiare il controllo d'errore, tramite le ritrasmissioni, per il traffico multimediale costituito da pacchetti codificati in modo diverso. Lo scopo è stato quello di descrivere le principali caratteristiche della trasmissione video sul wireless. Questa analisi ha permesso, infatti, lo studio di quei parametri che determinano l'efficienza della trasmissione dei pacchetti video attraverso un canale binario.

In accordo con lo sviluppo e la diffusione del traffico video, questo lavoro mette in luce importanti linee guida. Innanzitutto è utile delineare i due aspetti principali di questa analisi che hanno portato ai risultati ottenuti: in primo luogo l'assunzione di suddividere i pacchetti del traffico video in due diverse categorie, questa è un'idea di modello importante perché si evince dall'aver osservato come è fatto il traffico video, in secondo luogo l'applicazione delle ritrasmissioni selettive solo per i pacchetti principali, questa è invece un'idea di progetto di protocolli che porta delle conseguenze nella progettazione dell'ARQ. Sono questi due elementi che permettono di definire un modello molto efficiente. Infatti, come si può notare anche in Figura 4.2 del capitolo precedente, il throughput, normalizzato rispetto a $(1 - \varepsilon)$, assume sempre valori maggiori rispetto a 0.9 e molto prossimi a 1, anche nel caso di canale molto correlato. Inoltre si può constatare come avere valori elevati di B per il canale ha importanti conseguenze sulla velocità di trasmissione dei pacchetti video. Infatti, più B è elevato, più veloce sarà la trasmissione poiché maggiore è il valore del throughput medio. Questo però va a discapito della qualità del video stesso. Avere un valore di B troppo elevato, infatti, può implicare la perdita di un pezzo di video affetto da errore, proprio perché il canale è correlato. Questo inconvenien-

te, è però in parte compensato dal fatto che la ritrasmissione è selettiva e interessa solo i pacchetti A, che comporta, come ulteriore conseguenza, un ritardo complessivo limitato, oltre che avere un valore del throughput medio maggiore. Infatti la ritrasmissione di un pacchetto A prende il posto della trasmissione di un B, quindi complessivamente non influisce molto sul tempo di trasmissione dell'intero contenuto informativo, ma allo stesso tempo aumenta la qualità video senza far calare significativamente il throughput. Ovviamente avere un valore di B basso ha una conseguenza opposta, quindi maggiore qualità per il video ma minore velocità di trasmissione.

Per concludere è dunque fondamentale capire l'importanza assunta, per l'analisi, dalle ritrasmissioni selettive, ma allo stesso tempo anche dal modello e dalle assunzioni che stanno alla base; inoltre, essendo il procedimento adottato del tutto generale, questa analisi può essere estesa anche a modelli più complessi semplicemente tarando in modo diverso, a seconda dei casi, tutti gli strumenti utilizzati.

Bibliografia

- [1] N. Benvenuto, M. Zorzi. *Principles of communications networks and systems* Wiley, 2011.
- [2] S. Kallel. *Complementary punctured convolutional (CPC) codes and their applications*. IEEE Trans. Commun., vol. 43, no. 6, pp. 2005-2009, Jun. 1995.
- [3] G. K. Wallace. *The JPEG still picture compression standard*. Communications of the ACM, vol. 34, no. 4, pp. 30-44, 1991.
- [4] L. Badia, M. Rossi, M. Zorzi. *SR ARQ Packet Delay Statistics on Markov Channels in the Presence of Variable Arrival Rate*. Wireless Communications, IEEE Transaction on, vol. 5, pp. 1639-1644, July 2006.
- [5] J. G. Kim and M. M.Krunz. *Delay analysis of Selective Repeat ARQ for a Markovian source over a wireless channel*. IEEE Trans. Veh. Technol., vol 49, no. 5, pp. 1968-1981, 2000.
- [6] M. Rossi, L. Badia, and M. Zorzi. *Exact statics of ARQ packet delivery delay over Markov channels with finite round-trip delay*. Proc. IEEE Globecom 2003, vol. 6, pp. 3356-3360, 2005.
- [7] M. Rossi, L. Badia, and M. Zorzi. *On the delay statistics of an aggregate of SR-ARQ packets over Markov channels with finite round-trip time*. IEEE WCNC 2003, vol. 3, Mar. 2003, pp. 1773-1778.
- [8] M. E. Anagnostu and E. N. Protonotarios. *Performance analysis of the Selective-Repeat ARQ protocol*. IEEE Trans. Commun., vol. 34, no. 2, pp. 127-125, 1986.

- [9] 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, Dec. 2004, pp. 4061-4065.
- [10] L. Badia, M. Levorato, M. Zorzi. *Analysis of Selective Retransmission Techniques for Differentially Encoded Data*. Communications, 2009. ICC '09. IEEE International Conference on, pp. 1-6, Jun. 2009.
- [11] L. Badia, N. Baldo, M. Levorato, M. Zorzi. *A Markov Framework for Error Control Techniques Based on Selective Retransmission in Video Transmission over Wireless Channels*. Selected Areas in Communications, IEEE Journal on, vol. 28, pp. 488-500, April 2010.
- [12] L. Badia, M. Levorato, and M. Zorzi. *Markov analysis of selective repeat type II hybrid ARQ using block codes*. IEEE Trans. Commun., vol. 56, no. 9, pp. 1434-1441, Sep. 2008.
- [13] H. Liu and M. E. Zarki. *Performance of H.263 video transmission over wireless channels using hybrid ARQ*. IEEE J. Sel. Areas Commun., vol. 15, no. 9, pp. 1775-1786, Dec. 1997.
- [14] K. Stuhlmuller, N. Farber, M. Link, and B Girod. *Analysis of video transmission over lossy channels*. IEEE J. Sel. Areas Commun., vol. 18,no. 6, pp. 1012-1030, Jun. 2000.
- [15] B. Girod, N. Farber. *Feedback-based error control for mobile video transmission*. Proc., IEEE, vol. 87, no. 10, pp. 1707-1723, Oct. 1999.
- [16] D. Heyman, A. Tabatabai, and T. Lakshman. *Statistical analysis and simulation study of video teleconference traffic in ATM networks*. IEEE Trans. Circuits Syst. Videon Technol., vol. 2, no. 1, pp. 49-59, Mar 1992.
- [17] X. Zhang and Q. Du. *Adaptive low-complexity erasure-corretting code-based protocols for QoS-driven mobile multicast services over wireless networks*. IEEE Trans. Veh. Technol., vol. 55, no. 5, pp. 1633-1647, Sep. 2006.

- [18] F. Zhai, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos. *Rate-distortion optimized hybrid error control for real-time packetized video transmission*. IEEE Trans. Image Process., vol. 15, no. 1, pp. 40-51, Jan. 2006.
- [19] Xiaoan Lu, Yao Wang, and Elza Erkip. *Power efficient H.263 video transmission over wireless channels*. IEEE Image Proc., vol. 1, pp. 533-536, 2002.
- [20] Prathima Agrawal, Jyu-Cheng Chen, Shaline Kishore, Parameswaran Ramanathan, Krishna Sivalingam. *Battery power sensitive video processing in wireless network*. IEEE , vol. 1, pp. 116-120, Sep. 1998.
- [21] www.themathworks.com. *Matlab*.