



UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Corso di Laurea in Ingegneria dell'Informazione

**STRATEGIE DI GESTIONE ENERGETICA IN
TERMINALI RADIOMOBILI ALIMENTATI DA
SORGENTI RINNOVABILI**

Laureando

Giulia Mansutti

Relatore

Leonardo Badia

ANNO ACCADEMICO 2012/2013

Indice

1	Introduzione	1
2	Background	4
2.1	Catene di Markov a tempo discreto	5
2.2	Transizioni del sistema	8
2.3	Comportamento asintotico	10
2.3.1	Probabilità di stato	10
2.3.2	Probabilità di stato asintotiche	11
3	Il sistema	13
3.1	Il modello generale	13
3.2	Politiche di gestione energetica	14
3.2.1	<i>Reward media</i> e politica ottima	15
3.2.2	Politiche implementate	16
4	Implementazione e risultati numerici	19
4.1	Implementazione matlab	19
4.1.1	Matrice di transizione dei pacchetti dati	20
4.1.2	Matrice di transizione dei i quanti di energia	21
4.1.3	Vettore delle probabilità stazionarie	24
4.1.4	<i>Reward media</i> normalizzata	24
4.2	<i>Reward media</i> vs. capacità del buffer	26
4.3	Simulazioni dell'andamento del sistema	28
5	Conclusioni e sviluppi futuri	31
5.1	Conclusioni	31
5.2	Sviluppi futuri	32

INDICE

iii

Bibliografia

34

Sommario

Questo lavoro si propone di studiare l'efficienza di diverse politiche di gestione energetica, adottabili in terminali radiomobili alimentati da sorgenti rinnovabili (Energy Harvesting Devices), sulla base dell'assunzione che i processi degli arrivi dei quanti di energia e dei pacchetti dati che il dispositivo deve gestire siano modellati da due catene di Markov a tempo discreto.

L'obiettivo finale è quello di confrontare tra loro le diverse politiche implementate per determinare quale sia la più efficiente in termini di informazione trasmessa, al variare della capacità della batteria utilizzata dal dispositivo. Si fornisce inoltre un contributo originale estendendo il modello a considerare anche il mancato arrivo di pacchetti dati.

Capitolo 1

Introduzione

L'imponente sviluppo della tecnologia wireless, che vanta tra le sue applicazioni telefoni cellulari, computer portatili, reti di sensori e dispositivi biomedicali, è stato favorito in larga misura dall'avanzamento tecnologico che ha portato ad una significativa riduzione delle dimensioni e del prezzo delle batterie utilizzate dai terminali. Tuttavia, la sempre maggiore diffusione di dispositivi wireless ha generato un'altra esigenza primaria: trovare un modo per evitare o ridurre drasticamente la necessità di ricaricare, mantenere e sostituire le batterie utilizzate dai terminali [17]. Si pensi per esempio ad una rete formata da molti sensori che devono necessariamente essere wireless a causa della loro collocazione in un luogo in cui non è economicamente conveniente portare una rete di distribuzione elettrica (per esempio una stazione di monitoraggio meteo): la ricarica e la sostituzione di un grande numero di batterie costituisce un problema non indifferente.

Nonostante i passi avanti compiuti dalla tecnologia, il miglioramento delle prestazioni delle batterie, in termini di densità di energia, non ha un andamento tanto promettente (ovvero esponenziale in funzione del tempo) quanto quello di altre figure di merito. In Fig. 1.1 è riportato un grafico che confronta, nel caso di una tecnologia mobile ormai consolidata quale un computer portatile, l'andamento in funzione del tempo di diverse metriche che stabiliscono l'efficienza del dispositivo [17]; come si può vedere, l'aumento della densità di energia della batteria registra l'andamento più lento.

In questo scenario si è cominciato a guardare con interesse a dispositivi in grado di sfruttare l'energia proveniente dall'ambiente esterno raccogliendola e immagazzinandola sotto forma di energia elettrica [4], [7]: gli *energy harvesting devices* (EHD). Questi dispositivi hanno la capacità di utilizzare l'energia proveniente dall'ambiente in cui si trovano, e dunque sono in grado di operare autonomamente per lunghi periodi di tempo, riducendo la necessità di manutenzione e sostituzione.

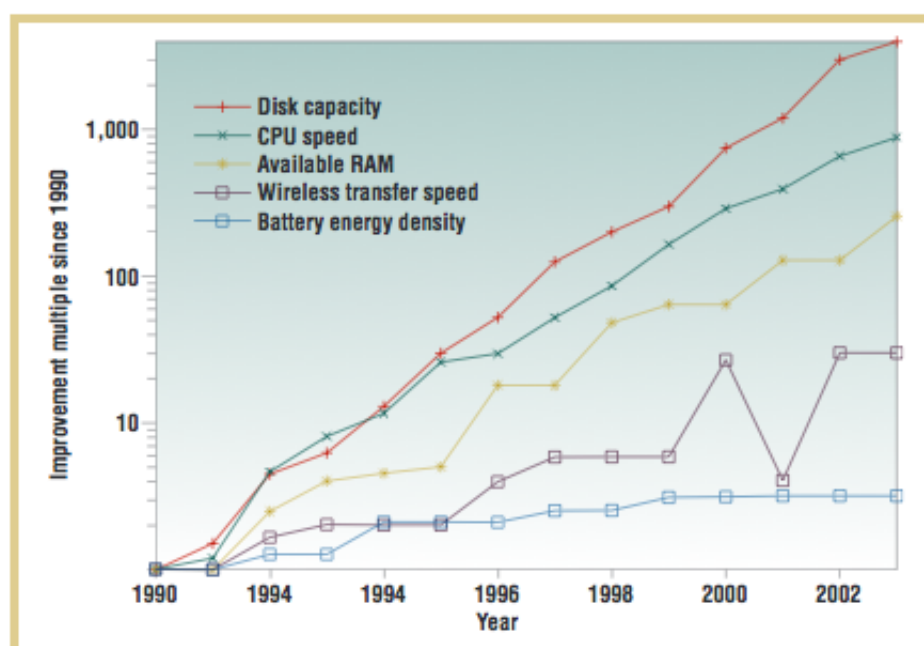


Figura 1.1: Miglioramenti relativi nella tecnologia dei personal computer dal 1990 al 2003.[17]

L'energia raccolta dagli energy harvesting devices può derivare da diverse fonti e avere diversi impieghi. Tra le fonti di energia rinnovabile utilizzate da questi dispositivi, si annoverano per esempio: quella solare, quella eolica, quella derivante dalle vibrazioni meccaniche (VEH: vibration energy harvesters [13]), dalle radiazioni elettromagnetiche e quella generata dal corpo umano [5], [16] (dal movimento o dal calore). Gli EHD possono avere diverse applicazioni nell'ambito della tecnologia wireless: vengono utilizzati per alimentare nodi di reti di sensori [6], [7], dispositivi biomedicali (sfruttando soprattutto il movimento e il calore come fonte di energia) [5], in applicazioni industriali di vario genere [13], e per alimentare dispositivi come orologi, cellulari e personal computer [17].

Gli EHD, grazie alla loro capacità di sfruttare l'energia proveniente da fonti rinnovabili al fine di compiere operazioni di rilevamento e trasmissione di dati, sono quindi particolarmente utili in situazioni in cui una sorgente di energia elettrica sicura e costante (come quella derivante da una centrale elettrica o da una batteria) non sia disponibile, oppure nel caso in cui sia troppo dispendiosa in termini di costo di installazione e manutenzione.

Consideriamo ora un terminale radiomobile che si occupa di gestire dell'informazione, ovvero che si occupa del rilevamento, dell'elaborazione, della generazione e della trasmissione di dati. Nel caso in cui il dispositivo sia alimentato da energia elettrica proveniente da sorgenti "tradizionali" quali una rete di distribuzione, l'obiettivo che generalmente si persegue è quello di minimizzare il consumo di energia, senza compromettere dei requisiti minimi in termini di performance, per esempio senza far aumentare sopra una certa soglia il ritardo nella trasmissione dei dati. Nel caso di un EHD invece, l'obiettivo che si persegue è quello di elaborare una *strategia* per gestire al meglio l'energia immagazzinata dall'ambiente [10].

Questo lavoro prende in considerazione un dispositivo che si occupa di trasmettere dati di diversa "importanza", con l'obiettivo di elaborare e confrontare tra loro diverse strategie (chiamate anche *politiche*) di gestione energetica per arrivare a determinare quale tra le diverse politiche sia più efficiente. L'analisi è stata condotta utilizzando due catene di Markov per modellare i processi degli arrivi dei quanti di energia e dei pacchetti dati, prendendo ispirazione da quanto fatto in [10], [11], [7], [9]. Il contributo originale di questa tesi consiste nell'aver considerato uno scenario nel quale in un dato istante di tempo può anche non arrivare alcun pacchetto dati da trasmettere.

Nel capitolo 2 vengono riportati alcuni concetti fondamentali alla base della teoria delle catene di Markov, così da poter comprendere meglio il lavoro svolto. Nel capitolo 3 viene presentato il modello generale del sistema analizzato e vengono definite le tre politiche implementate. Nel capitolo 4 viene brevemente descritta l'implementazione matlab del sistema per poi arrivare a presentare e commentare i risultati ottenuti. Infine il capitolo 5 raccoglie le conclusioni del lavoro e i possibili sviluppi futuri.

Capitolo 2

Background

I dispositivi radiomobili alimentati da una fonte rinnovabile, noti come Energy Harvesting Devices (EHD), si servono dell'energia fornita dall'ambiente esterno ed accumulata in una batteria, per gestire informazioni: ovvero ricevere, elaborare, generare e trasmettere dati [10] [17]. Tuttavia, l'energia proveniente da fonti rinnovabili ha l'inconveniente di non essere sempre disponibile e costante: si pensi ad un pannello fotovoltaico che sfrutta la radiazione luminosa emanata dal sole per accumulare energia, in questo caso l'approvvigionamento di energia dipende dall'ora del giorno (di notte l'energia non è disponibile) come anche dalle condizioni meteorologiche (se è nuvoloso, l'energia che raggiunge il dispositivo potrebbe essere intermittente). Proprio per la natura non costante delle sorgenti di cui si servono gli energy harvesting devices, è cruciale elaborare delle strategie di gestione energetica che rendano sempre più efficiente l'utilizzo della batteria [9] [10].

Un EHD può essere rappresentato dal modello riportato in Figura 2.1.

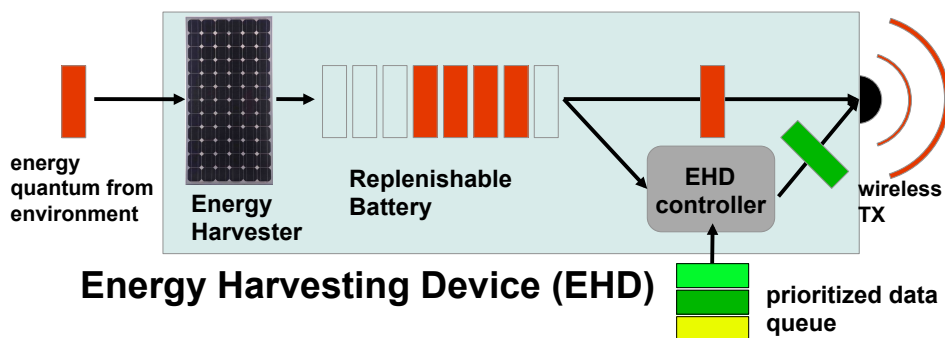


Figura 2.1: Modello di un Energy Harvesting Device [1]

L'energia perviene all'EHD sotto forma di quanti che, tramite un apposito dispositivo (un pannello solare, nell'esempio in figura) vengono trasferiti alla batteria ricaricabile, rappresentata da un buffer [10], [11]. Per quanto riguarda i dati gestiti, essi sono costituiti da pacchetti di diverso valore, dipendente dalla loro importanza; ciascuno di questi pacchetti arriva ad un dispositivo di controllo: questo dispositivo si occupa della trasmissione dei dati e quindi, in definitiva, dell'utilizzo dell'energia immagazzinata nella batteria, secondo la strategia adottata.

Vogliamo definire il concetto di strategia di gestione energetica per poi analizzarne diverse, al fine ultimo di determinare quale di queste sia più efficiente e a quali condizioni. La metrica per definire l'efficienza di una determinata politica energetica è costituita dal *guadagno* medio ottenibile adottando la politica in questione. Il guadagno può essere pensato in linea generale, come una misura della quantità media di pacchetti dati trasmettibili adottando una determinata strategia di gestione [10]: esso dipende dalla strategia adottata, dal processo degli arrivi dei quanti di energia e dal processo degli arrivi dei pacchetti dati.

In particolare esso dipende dalla probabilità che il livello di energia nella batteria assuma un certo valore a regime, ovvero dopo un intervallo di tempo sufficientemente lungo affinché le condizioni iniziali (cioè la carica della batteria all'istante iniziale) non influenzino l'evoluzione del processo.

Per poter calcolare questa probabilità, detta probabilità asintotica di stato (def 2.10) e quindi il guadagno, poiché i processi degli arrivi dell'energia e dei dati sono rappresentati da due catene di Markov tra loro indipendenti, è necessario introdurre alcuni concetti fondamentali sulla teoria delle catene di Markov. Questo capitolo si occupa per l'appunto di presentare i concetti teorici fondamentali utilizzati in questo lavoro: si comincia dalla definizione di cosa sia una catena di Markov (MC ovvero Markov Chain), per arrivare alla definizione del vettore delle probabilità di stato stazionario e di alcune sue proprietà che sono state sfruttate per implementare il sistema in matlab.

2.1 Catene di Markov a tempo discreto

Il sistema considerato in Figura 2.1 è un sistema a tempo discreto: l'asse temporale è suddiviso in intervalli (chiamati anche *passi*) in ognuno dei quali può arrivare o meno un pacchetto dati e può arrivare o meno un quanto di energia. Il pacchetto dati che perviene al dispositivo nell'intervallo k , è rappresentato da una variabile aleatoria discreta V_k che assume valori su un alfabeto finito di interi, che nel caso considerato contiene 4 valori: $\mathcal{V} =$

$\{0, v_1, v_2, v_3\}$ con probabilità p_0, p_1, p_2, p_3 , dove $V_k = 0$ indica l'evento "non è arrivato alcun pacchetto nell'intervallo k ", che si verifica con probabilità p_0 .

Ricordando la definizione di processo aleatorio a tempo discreto [3]:

Definizione Un processo aleatorio a tempo discreto è costituito da una famiglia di variabili aleatorie $\{X_n\}$, dove n assume valori in uno specifico sottoinsieme degli interi; l'insieme dei valori che $\{X_n\}$ può assumere, è detto *spazio di stato*. [3]

la sequenza costituita dalle variabili aleatorie $\{V_1, V_2, \dots, V_n, \dots\}$, dove V_i rappresenta il pacchetto dati che arriva nell'intervallo temporale i , costituisce un processo aleatorio; l'insieme \mathcal{V} dei valori assumibili dalle variabili V_i è lo *spazio di stato* del processo $\{V_k\}$. Assumiamo che le variabili V_k siano indipendenti tra loro e identicamente distribuite (i.i.d.).

Analogamente il quanto di energia che può arrivare nell'intervallo k è rappresentato da una variabile aleatoria Q_k che può assumere solo due valori $\mathcal{Q} = \{0, 1\}$ con probabilità p_0 e $p_1 = 1 - p_0$: se $Q_k = 1$, significa che al passo k è arrivato un quanto di energia, altrimenti non è arrivato alcun quanto. Per la definizione appena data, anche la sequenza di variabili aleatorie $\{Q_1, Q_2, \dots, Q_n, \dots\}$, dove Q_i rappresenta il quanto di energia che arriva nell'intervallo i , costituisce un processo aleatorio di spazio di stato \mathcal{Q} . Anche in questo caso, assumiamo che le variabili Q_k siano i.i.d..

I processi degli arrivi dei pacchetti dati e dei quanti di energia sono due processi particolari, ovvero sono due catene di Markov [3]:

Definizione Il processo aleatorio $\{X_0, X_1, \dots\}$ è una catena di Markov (MC, ovvero Markov Chain) se gode della proprietà di Markov, ovvero se per ogni $n \geq 1$: [3]

$$P(X_{n+1} = i_{n+1} | X_n = i_n, \dots, X_0 = i_0) = P(X_{n+1} = i_{n+1} | X_n = i_n) \quad (2.1)$$

In altre parole, data una sequenza di valori assunti dal processo in diversi istanti $\{0, 1, \dots, n\}$, la probabilità che il processo assuma un determinato valore al tempo $n + 1$, dipende solamente dal valore che il processo ha assunto nel precedente istante n , e non dall'intera sequenza di valori.

Descrivere l'evoluzione dello stato della batteria nel tempo, ovvero l'evoluzione della sequenza di variabili aleatorie $\{E_k\}$ al variare di k è più complicato, in quanto esso dipende, oltre che dai processi degli arrivi dei quanti di energia e dei pacchetti dati, anche da come è modellata la batteria, ovvero dalla sua capacità massima e_{max} che determina lo spazio di stato $\mathcal{E} = \{0, 1, \dots, e_{max}\}$ di $\{E_k\}$; inoltre dipende anche dalla strategia di gestione

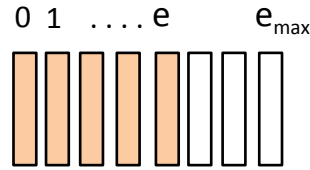


Figura 2.2: Modello di una batteria di capacità massima e_{max} , il cui livello di energia è pari a e .

energetica adottata dal dispositivo, nella misura in cui questa determina la probabilità di trasmettere un determinato pacchetto (e quindi di consumare un quanto di energia) quando il dispositivo si trova in un certo stato.

L'evoluzione della carica in una batteria di capacità e_{max} , come quella in Figura 2.2 può essere quindi visto come un *processo di decisione di Markov* (MDP, cioè Markov Decision Process), ovvero [14]:

Definizione Un processo di decisione di Markov (MDP) è una 4-tupla $(S, A, P_a(S_{ij}, S_{lm}), R_a(S_{ij}, S_{lm}))$ dove [14]:

- S è un insieme finito di stati; in questo caso $S = \{S_{ij}\}_{i \in \mathcal{E}, j \in \mathcal{V}}$ con $S_{ij} = (E_i, V_j)$, ovvero ogni stato è formato da una coppia costituita da un possibile livello di energia nella batteria $E_i \in \mathcal{E} = \{0, \dots, e_{max}\}$ e da un possibile pacchetto dati $V_j \in \mathcal{V} = \{0, v_1, v_2, v_3\}$.
- A è un insieme finito di azioni; in questo caso $A = \{\text{trasmettere } V_1, \text{trasmettere } V_2, \text{trasmettere } V_3, \text{non trasmettere } V_1, \text{non trasmettere } V_2, \text{non trasmettere } V_3\}$.
- $P_a(S_{ij}, S_{lm}) = P[S(n+1) = S_{lm} | S(n) = S_{ij}, a_n = a]$ è la probabilità che lo stato al tempo $n+1$ sia $S(n+1)$ dato che al tempo n lo stato era $S(n)$ ed è stata compiuta l'azione a .
- $R_a(S_{ij}, S_{lm})$ è il guadagno immediato che si ottiene passando dallo stato s allo stato $s+1$ compiendo l'azione a . Per esempio se al tempo n la batteria ha 3 quanti di energia e non viene ricevuto alcun ulteriore quanto, ma viene invece ricevuto un pacchetto dati di valore $v=3$ e si decide di compiere l'azione $\{\text{trasmettere } v\}$ (consumando quindi un quanto di energia per la trasmissione di v), il guadagno immediato che si ottiene passando dallo stato $(3, v)$ allo stato $(2, v)$ è 3.

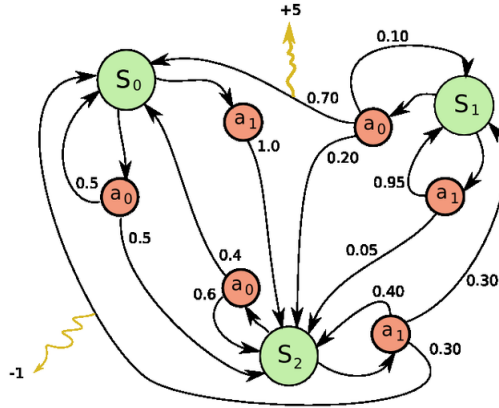


Figura 2.3: Esempio di MDP: gli stati sono evidenziati in verde, le azioni in rosso, mentre le frecce gialle indicano il guadagno immediato ottenibile passando da uno stato ad un altro tramite la particolare azione a_i .

2.2 Transizioni del sistema

Le due MC che rappresentano gli arrivi dei quanti di energia e dei pacchetti dati e il MDP che rappresenta l'energia nella batteria, hanno spazi di stato rispettivamente $\mathcal{Q} = \{0, 1\}$, $\mathcal{V} = \{0, v_1, v_2, v_3\}$ e $\mathcal{E} = \{0, \dots, e_{max}\}$. Per determinare l'evoluzione dei tre processi, dobbiamo conoscere la descrizione statistica dell'evoluzione di ciascun processo da uno stato ad un altro.

Per fare questo richiamiamo la definizione di matrice di transizione in k passi per una MC di N stati [2]:

$$P(k) = \begin{bmatrix} P_{0,0}(k) & P_{0,1}(k) & \dots & P_{0,j}(k) & \dots & P_{0,N}(k) \\ P_{1,0}(k) & P_{1,1}(k) & \dots & P_{1,j}(k) & \dots & P_{1,N}(k) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{i,0}(k) & P_{i,1}(k) & \dots & P_{i,j}(k) & \dots & P_{i,N}(k) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{N,0}(k) & P_{N,1}(k) & \dots & P_{N,j}(k) & \dots & P_{N,N}(k) \end{bmatrix} \quad (2.2)$$

dove $P_{i,j}(k) = P[X_{n+k} = j | X_n = i]$ è la probabilità che il processo evolva dallo stato i allo stato j in k passi. E dove per definizione $P(0) = I$, poiché

in zero passi il sistema rimane nello stato in cui si trova e quindi gli unici elementi non nulli della matrice sono quelli sulla diagonale del tipo $P_{i,i}(0)$.

Prendendo in considerazione il caso $k = 1$, otteniamo la *matrice di transizione in un passo* [2]:

$$P = \begin{bmatrix} P_{0,0} & P_{0,1} & \dots & P_{0,j} & \dots & P_{0,N} \\ P_{1,0} & P_{1,1} & \dots & P_{1,j} & \dots & P_{1,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{i,0} & P_{i,1} & \dots & P_{i,j} & \dots & P_{i,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{N,0} & P_{N,1} & \dots & P_{N,j} & \dots & P_{N,N} \end{bmatrix} \quad (2.3)$$

questa matrice ha la proprietà di generare qualsiasi altra matrice di transizione in k passi. Infatti dalle equazioni di Chapman-Kolmogorov:

Proposizione 2.2.1 *Per ogni coppia di interi positivi n, k , vale che [2]:*

$$P_{i,j}(n+k) = \sum_{s=0}^N P_{i,s}(n)P_{s,j}(k), \quad i, j \in \mathcal{S} \quad (2.4)$$

che in forma matriciale diventa: $P(n+k) = P(n)P(k)$, o equivalentemente:

$$P(k) = P^k \quad (2.5)$$

segue che è sufficiente conoscere la matrice di transizione in un passo per poter determinare tutte le altre probabilità di transizione in k passi.

La matrice di transizione in un passo del processo di generazione dei pacchetti dati, è quindi definita da:

$$P_v = \begin{bmatrix} P_{v_0,v_0} & P_{v_0,v_1} & P_{v_0,v_2} & P_{v_0,v_3} \\ P_{v_1,v_0} & P_{v_1,v_1} & P_{v_1,v_2} & P_{v_1,v_3} \\ P_{v_2,v_0} & P_{v_2,v_1} & P_{v_2,v_2} & P_{v_2,v_3} \\ P_{v_3,v_0} & P_{v_3,v_1} & P_{v_3,v_2} & P_{v_3,v_3} \end{bmatrix} = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \\ p_0 & p_1 & p_2 & p_3 \\ p_0 & p_1 & p_2 & p_3 \\ p_0 & p_1 & p_2 & p_3 \end{bmatrix} \quad (2.6)$$

dove $P_{v_i,v_j} = P[V_{k+1} = v_j | V_k = v_i]$ indica la probabilità che all'istante $k+1$ arrivi il pacchetto j dato che all'istante k è arrivato il pacchetto i .

La seconda uguaglianza segue dal fatto che si è assunto che gli arrivi siano indipendenti, perciò $P_{v_i,v_j} = p_j \quad \forall i \in \{0, 1, 2, 3\}$, ovvero la probabilità

che in un certo istante arrivi il pacchetto i –esimo è indipendente da quale pacchetto sia arrivato nell'istante precedente.

La matrice di transizione per i quanti di energia è semplicemente

$$P_q = \begin{bmatrix} P_{0,0} & P_{0,1} \\ P_{1,0} & P_{1,1} \end{bmatrix} = \begin{bmatrix} p_0 & 1 - p_0 \\ p_0 & 1 - p_0 \end{bmatrix}$$

Infine la matrice di transizione per il MDP che modella la batteria, è:

$$P_e = \begin{bmatrix} P_{0,0} & P_{0,1} & \dots & P_{0,e_{max}} \\ P_{1,0} & P_{1,1} & \dots & P_{1,e_{max}} \\ \vdots & \vdots & \ddots & \vdots \\ P_{e_{max},0} & P_{e_{max},1} & \dots & P_{e_{max},e_{max}} \end{bmatrix} \quad (2.7)$$

dove $P_{i,j} = P[E_{k+1} = j | E_k = i]$ con $i, j \in \{0, \dots, e_{max}\}$; ovvero $P_{i,j}$ è la probabilità che l'energia nel buffer all'istante $k+1$ sia j dato che all'istante k era i . I singoli termini della matrice dipendono dalle probabilità delle azioni che si possono compiere da un dato stato i , ovvero dipendono dalla strategia di gestione energetica

2.3 Comportamento asintotico

La probabilità che la batteria, si trovi in un certo stato quando è a regime, ovvero dopo un tempo *sufficientemente* lungo (il significato di questo aggettivo verrà chiarito a breve), al limite tendente ad infinito, prende il nome di probabilità (di stato) asintotica. A partire dalle probabilità di transizione, è possibile calcolare le probabilità asintotiche, servendosi di un altro tipo di probabilità, dette probabilità di stato.

2.3.1 Probabilità di stato

Le probabilità di stato sono le probabilità *assolute* che il sistema si trovi in un certo stato j , a differenza delle probabilità di transizione che sono probabilità *condizionate*, ovvero probabilità che il sistema sia nello stato j al tempo $n+k$, dato che si trovava nello stato i al tempo n .

Prendendo come esempio una batteria di capacità e_{max} , il vettore delle probabilità di stato (in k passi) è:

$$p(k) = [p_0(k), p_1(k), \dots, p_{e_{max}}(k)]. \quad (2.8)$$

dove $p_j(k) = P(E_k = j)$ è la probabilità che il livello di energia nel buffer sia j all'istante k .

Dalla condizione di normalizzazione e dal teorema della probabilità totale, segue che [2]:

$$\begin{aligned}
 p(k) &= p(k)\mathbf{1} = \sum_{j=0}^{e_{max}} p_j(k) = \sum_{j=0}^{e_{max}} P[X_k = j] = \\
 &= \sum_{i=0}^{e_{max}} P[X_k = j|X_0 = i]P[X_0 = i] = \\
 &= \sum_{j=0}^{e_{max}} \sum_{i=0}^{e_{max}} p_i(0)P_{i,j}(k) = p(0)P(k) = p(0)P^k
 \end{aligned} \tag{2.9}$$

dove $p(0)$ è il vettore delle probabilità di stato iniziali, e l'ultimo passaggio è una conseguenza dell'equazione di Chapman-Kolmogorov (2.5). Dunque il comportamento delle probabilità di stato in funzione del tempo, è univocamente determinato dal vettore delle probabilità di stato iniziali e dalla matrice di transizione del sistema.

2.3.2 Probabilità di stato asintotiche

Il vettore delle probabilità di stato asintotiche, sempre considerando una batteria di capacità massima e_{max} , è definito da:

$$\pi = [\pi_0, \pi_1, \dots, \pi_{e_{max}}] \quad \text{con} \quad \pi_j = \lim_{k \rightarrow \infty} P_{i,j}(k), \quad j \in \mathcal{S} \tag{2.10}$$

In sostanza π_j fornisce la probabilità che, campionando il processo in un istante casuale arbitrariamente lontano nel tempo, si osservi il processo nello stato j . Quando esiste (e questo non è messo in discussione in questa trattazione), la probabilità asintotica di stato è indipendente dal vettore delle probabilità iniziali di stato $p(0)$.

Dalla definizione appena data (2.10) e da (2.9) segue che:

$$\pi = \lim_{k \rightarrow \infty} p(k) = \lim_{k \rightarrow \infty} p(0)P^k = p(0) \lim_{k \rightarrow \infty} P^k = p(0)P^\infty \tag{2.11}$$

Introduciamo ora un'altra definizione:

Definizione Il vettore p delle probabilità di stato, è detto *stazionario* se, a partire da p , il vettore delle probabilità di stato non evolve più nel tempo, ovvero rimane costante: $p = pP(k) = p$ per ogni $k > 0$.

Il MDP che rappresenta l'evoluzione della carica nella batteria soddisfa le condizioni (vedi teorema 7.9 in [2]) affinché valga la seguente proprietà: il vettore π è l'unico vettore delle probabilità di stato stazionarie, ovvero $\pi = \pi P$.

Da $\pi = \pi P$ segue che $\pi(I - P) = 0$. Aggiungendo a questo, la condizione di normalizzazione, otteniamo il seguente sistema:

$$\begin{cases} \pi(I - P) = 0 \\ \pi \mathbf{1} = \sum_{j=0}^{e_{max}} \pi_j = 1 \end{cases} \Rightarrow \pi_j(1 - P_{j,j}) = \sum_{i \in \mathcal{E}, i \neq j} \pi_i P_{i,j} \quad (2.12)$$

dove $\mathbf{1}$ indica il vettore di dimensione $e_{max} \times 1$ con tutti gli elementi 1. Ponendo $e_{max} = N$, risolvendo il sistema di $N + 1$ equazioni (delle quali una è combinazione lineare delle altre, e quindi ridondante), possiamo determinare tutte le componenti del vettore delle probabilità stazionarie.

Quindi conoscendo la matrice di transizione in un passo, è possibile calcolare la probabilità che il livello della batteria sia i quando il sistema è a regime.

Per calcolare il guadagno, siamo interessati a conoscere qual è la probabilità che a regime il sistema si trovi nello stato (e, v_i) , ovvero la probabilità che a regime il livello della batteria sia e , e che venga ricevuto il pacchetto v_i . Dobbiamo quindi calcolare i valori degli elementi del vettore:

$$\pi = [\pi(0, 0) \dots \pi(e_{max}, 0) \pi(0, v_1) \dots \pi(e_{max}, v_1) \dots \pi(e_{max}, v_3)] \quad (2.13)$$

Tuttavia, le due MC degli arrivi dei quanti di energia e dei pacchetti dati, sono indipendenti, quindi la probabilità che il sistema si trovi con la batteria nello stato e e con un pacchetto dati di valore v_i è semplicemente il prodotto delle due probabilità, quindi il vettore π diventa:

$$\pi = [\pi(0)p_0 \dots \pi(e_{max})p_0 \pi(0)p_1 \dots \pi(e_{max})p_1 \dots \pi(e_{max})p_3] \quad (2.14)$$

ovvero, definendo $\pi^* = [\pi(0) \dots \pi(e_{max})]$:

$$\pi = [\pi^* p_0 \pi^* p_1 \pi^* p_2 \pi^* p_3]. \quad (2.15)$$

Capitolo 3

Il sistema

3.1 Il modello generale

Il modello del sistema analizzato, che prende ispirazione da quello considerato in [10], comprende quattro componenti fondamentali: il tempo, la batteria utilizzata dal dispositivo, i pacchetti dati e i quanti di energia gestiti dal dispositivo.

1. **Il tempo.** Nel sistema considerato, il tempo è suddiviso in intervalli: un dato intervallo k va dall'istante k all'istante $k + 1$: $[k, k + 1)$, $k \in \mathbb{Z}$.
2. **La batteria.** La batteria del dispositivo è modellata come un buffer di capacità massima e_{max} , ovvero il livello di energia della batteria può assumere valori in $\mathcal{E} = \{0, 1, \dots, e_{max}\}$.
3. **I pacchetti dati.** L'informazione che il dispositivo deve gestire è stata considerata in forma di pacchetti, a ciascuno dei quali è stato assegnato un determinato valore v di valore crescente in base alla sua importanza: $v \in \mathcal{V} = \{v_1, v_2, v_3\}$, dove v_1 indica un pacchetto poco importante, v_2 uno di importanza media e v_3 uno molto importante.

Il contributo originale di questo lavoro consiste nell'aver preso in considerazione anche il caso in cui in un dato intervallo di tempo possa non arrivare alcun pacchetto dati: questa situazione è stata modellata introducendo un ulteriore valore $v_0 = 0$ che il pacchetto v assume con probabilità p_0 , ovvero p_0 coincide con la probabilità che non arrivino pacchetti dati al sistema.

In ogni intervallo temporale k , può arrivare al massimo un pacchetto dati al dispositivo, e analogamente può essere inviato al massimo un pacchetto dati dal dispositivo. Un pacchetto dati che arriva al di-

spositivo all'istante k può essere solo trasmesso oppure scartato dal dispositivo.

4. **L'energia.** L'energia arriva al dispositivo sotto forma di quanti: in ogni intervallo di tempo può arrivare al massimo un quanto di energia; quindi, come visto nel capitolo 2, indicando con Q_k il processo degli arrivi dei quanti nell'intervallo temporale k , si ha che: $Q_k \in \mathcal{Q} = \{0, 1\}$, dove $Q_k = 1$, se al tempo k arriva un quanto di energia, mentre $Q_k = 0$ altrimenti. Per trasmettere un pacchetto dati nell'intervallo k , è necessario un quanto di energia, indipendentemente dal valore del pacchetto.

Inoltre è stata fatta l'ulteriore assunzione che un quanto di energia che arriva al dispositivo nell'intervallo k può essere utilizzato per trasmettere pacchetti dati solo a partire dall'intervallo $k + 1$. Quindi se per esempio nell'intervallo k : la batteria è scarica ($E_k = 0$) e arrivano un pacchetto dati ed un quanto di energia, quest'ultimo non può essere utilizzato per trasmettere il pacchetto dati; quindi il pacchetto sarà scartato e il quanto di energia verrà accumulato nella batteria.

Il numero di quanti di energia E_{k+1} presenti al tempo $k + 1$ nella batteria, è così determinato:

$$E_{k+1} = \min\{[E_k - A_k]^+ + Q_k, e_{max}\}$$

dove A_k indica l'azione compiuta dal dispositivo all'istante k : A_k è 1 se all'istante k viene trasmesso un pacchetto dati e conseguentemente viene consumato un quanto di energia del buffer, mentre è 0 se all'istante k non viene trasmesso alcun pacchetto dati e quindi non viene consumato alcun quanto di energia.

Due situazioni limite indesiderate in cui si può trovare il buffer sono: *energy outage* e *energy overflow*. Il primo caso si verifica quando $E_k = 0$, ovvero quando il buffer è vuoto; di conseguenza se all'istante k arriva un pacchetto dati, il dispositivo non può trasmetterlo perchè non ha energia nella batteria. Il secondo caso si verifica quando $(E_k = e_{max}) \wedge (A_k = 1) \wedge (Q_k = 0)$, ovvero quando arriva un quanto di energia in un istante in cui il buffer è già pieno e dunque questo non può essere immagazzinato andando così sprecato.

3.2 Politiche di gestione energetica

Dato il modello del sistema appena delineato, si vuole rendere efficiente la gestione dell'energia accumulata dal dispositivo che, provenendo da fonti

rinnovabili, non è costante nè perfettamente prevedibile. L'efficienza di un dispositivo è definita dal rapporto tra i dati trasmessi dal dispositivo e quelli da esso trasmissibili, se disponesse sempre di energia nella batteria. Assumiamo che la struttura della batteria, il processo degli arrivi dei quanti di energia e il processo degli arrivi dei pacchetti dati siano non modificabili (e quindi non migliorabili). Per rendere efficiente il dispositivo in termini di dati trasmessi, il parametro sul quale agiremo è A_k , ovvero l'azione che il dispositivo compie al tempo k . Una politica di gestione energetica impone appunto la probabilità che il dispositivo compia una determinata azione (trasmetta/non trasmetta un determinato pacchetto v ricevuto al tempo k), dato che la sua energia nella batteria è e .

La definizione di politica di gestione energetica di un EHD richiede l'introduzione del concetto di *stato*:

Definizione lo stato di un EHD al tempo k è definito come $S_k = (E_k, V_k)$, dove E_k è il livello di energia del buffer nell'intervallo k , mentre V_k indica il valore dell'importanza del pacchetto ricevuto nell'intervallo k , oppure è zero se nell'intervallo considerato non è stato ricevuto alcun pacchetto.

Definizione Dato un dispositivo EHD, il cui stato al tempo k è $S_k = (E_k, V_k)$, una *politica* μ è una misura di probabilità, parametrizzata da S_k , sullo spazio delle azioni $\mathcal{A} = \{0, 1\}$.

Quindi dato S_k , una politica determina l'azione da svolgere al tempo k , ovvero determina $A_k \in \{0, 1\}$. Per esempio: dato $S_k = (e, v) \in \mathcal{E} \times V$, la politica μ assegna $\mu(0; e, v) = 1 - \mu(1; e, v)$, ovvero assegna una probabilità agli eventi "trasmettere il pacchetto v " e "non trasmettere il pacchetto v ", rispettivamente [10].

3.2.1 *Reward media e politica ottima*

Data una certa politica μ , la metrica utilizzata per misurare le sue prestazioni, è il *guadagno medio* (ovvero la *reward media*): il guadagno medio fornisce un'indicazione della quantità di dati che il dispositivo riesce a trasmettere, adottando una certa politica μ , quando è "a regime", ovvero per $k \rightarrow \infty$ (dove k indica un istante temporale).

Dato uno stato iniziale S_0 ed una politica μ , la *reward media* è definita come:

$$G(\mu, S_0) = \liminf_{K \rightarrow \infty} \frac{1}{K} \cdot \mathbb{E} \left[\sum_{k=0}^{K-1} A_k V_k \middle| S_0 \right] \quad (3.1)$$

dove il valore atteso è calcolato rispetto alle variabili $\{A_k, Q_k, V_k\}$, e dove A_k dipende da μ .

Una politica è ottima se massimizza la reward media definita in (3.1). Si può dimostrare [10] che la politica ottima ha una struttura a soglia, ovvero è del tipo:

$$\begin{cases} \mu(1; e, v) = 1 & v \geq v_{th}(e) \\ \mu(1; e, v) = 0 & v < v_{th}(e) \end{cases} \quad (3.2)$$

dove $v_{th}(e)$ è un opportuna soglia definita per ogni livello di energia $e \in \mathcal{E}$, $e \neq 0$ del buffer.

Si procede adesso a definire tre diverse politiche: una *non a soglia* (politica “greedy”), una *a soglia indipendente dal livello di energia* (politica “balanced”), e una *a soglia* (politica “low complexity”), con il duplice obiettivo di studiarne il comportamento singolarmente al variare della capacità massima del buffer e di confrontarle tra loro, per arrivare a mostrare come quella a soglia abbia una performance migliore, in termini di reward, rispetto alle altre due.

3.2.2 Politiche implementate

Politica *greedy*

La politica *greedy* (cioè aggressiva) è una politica che impone al dispositivo di trasmettere qualunque pacchetto dati gli arrivi, sia esso poco o molto importante, indipendentemente dal livello di energia presente nella batteria (ovviamente fino a quando il buffer non è vuoto). Formalmente per la politica *greedy* si ha che:

$$\mu(1; e, v) = 1 \quad \forall v \in V \quad \forall e \in \mathcal{E}, e \neq 0$$

Questa politica non è a soglia, o meglio, ha soglia nulla $v_{th}(e) = 0$, $\forall e \in \mathcal{E}_0$. Lo svantaggio della politica *greedy* è che, trasmettendo qualsiasi pacchetto dati, indipendentemente dal livello di energia nel buffer, aumenta la probabilità che il buffer vada in energy outage: aumenta quindi la probabilità che la batteria sia scarica, e in particolare che sia scarica quando arriva un pacchetto molto importante, che di conseguenza non può essere trasmesso.

Per esempio, se all’istante k il livello di energia nel buffer è $e = 1$, e arriva un pacchetto di valore v_1 (poco importante) mentre non arriva alcun quanto di energia, utilizzando la politica *greedy*, il dispositivo trasmette v_1 , consumando l’unico quanto di energia presente nel buffer. All’istante $k + 1$ il livello di energia nel buffer sarà quindi $e = 0$: se adesso arrivasse un pacchetto di

valore v_3 (molto importante), il dispositivo non avrebbe quanti a disposizione per la trasmissione di v_3 , che andrebbe così scartato (la situazione non cambierebbe nemmeno se all'istante $k + 1$ arrivasse un quanto di energia, perchè per ipotesi questo potrebbe essere utilizzato solo a partire dall'istante successivo).

Politica *balanced*

La politica *balanced* (bilanciata) è una politica a soglia, dove la soglia è stata scelta in modo che il tasso medio di consumo dei quanti di energia, cioè il tasso medio di trasmissione dei pacchetti dati, sia uguale al tasso di arrivo dei quanti di energia λ . Così facendo, il consumo dei quanti di energia è statisticamente bilanciato dalla generazione dei quanti stessi.

Nella pratica, per ottenere questo effetto senza complicare il modello considerato, è stata scelta $\lambda = p_{v_2} + p_{v_3}$ dove i termini a secondo membro indicano rispettivamente la probabilità di ricevere i pacchetti v_2 (media importanza) e v_3 (elevata importanza). Così facendo, la probabilità che il dispositivo trasmetta un pacchetto di media-alta importanza nell'intervallo k è uguale alla probabilità di ricevere un quanto di energia nello stesso intervallo. La politica *balanced* è definita da:

$$\begin{cases} \mu(1; e, v) = 1 & v \geq v_2 \\ \mu(1; e, v) = 0 & v < v_2. \end{cases} \quad (3.3)$$

Rispetto alla politica *greedy*, la politica *balanced* fa sì che il dispositivo assuma un comportamento più bilanciato: il tasso di consumo e di generazione dei quanti, in questo caso sono uguali; mentre utilizzando la politica *greedy*, il tasso di consumo dell'energia è maggiore del tasso di generazione della stessa, facendo aumentare la probabilità che la batteria vada in energy outage.

D'altra parte, anche la politica *greedy* ha uno svantaggio. Per esempio, supponiamo che al tempo k l'energia nel buffer sia $e = e_{max}$, e supponiamo che nello stesso istante arrivino un pacchetto v_1 ed un quanto di energia: il dispositivo scarta v_1 , non consumando alcun quanto di energia; così facendo però, il buffer è ancora pieno e non può accumulare il quanto che è appena arrivato che va quindi sprecato.

Non trasmettendo mai ad un tasso maggiore di quello di generazione dei quanti di energia, aumenta la probabilità che il buffer vada in overflow, cioè aumenta la probabilità che venga sprecata energia.

Politica *low complexity*

La politica *low complexity* è una politica a soglia che rappresenta un compromesso tra la politica *balanced* e la politica *greedy*. Essa compendia le caratteristiche desiderabili delle due: la politica *greedy* evita l'overflow, e la politica *balanced* garantisce un bilanciamento dei tassi di consumo e di generazione dell'energia quando il livello di energia nel buffer assume valori intermedi. La politica *low complexity* implementata in questo lavoro prende ispirazione da quella sviluppata in [10], dove sono state anche dimostrate alcune sue proprietà teoriche.

Assumendo $e_{max} \geq 3$ (i casi limite sono trattati subito dopo), la politica *low complexity* è costruita in modo tale da essere:

- conservativa, cioè tale da trasmettere solo il pacchetto molto importante v_3 , quando c'è poca energia nel buffer ($e = 1$); così da diminuire la probabilità che questo si trovi in *energy outage*;
- bilanciata, cioè tale da simulare la politica *balanced* trasmettendo v_2 e v_3 , quando il livello di energia nel buffer assume valori intermedi ($2 \leq e \leq e_{max} - 1$);
- aggressiva, cioè tale da trasmettere tutti i pacchetti dati (compresi quelli meno importanti) come fa la politica *greedy*, quando il livello di energia nel buffer è uguale ad e_{max} ; così da mantenere bassa la probabilità che il buffer vada in *overflow*.

La politica *low complexity*, per $e_{max} \geq 3$, è quindi espressa da:

$$\begin{cases} \mu(1; 1, v) = 1 & v \geq v_3 \\ \mu(1; e, v) = 1 & v \geq v_2, \quad 2 \leq e \leq e_{max} \\ \mu(1; e_{max}, v) = 1 & v \geq v_1 \end{cases} \quad (3.4)$$

Per quanto riguarda i casi limite: se $e_{max} = 1$ la politica *low complexity* si comporta come quella *balanced*, trasmettendo solo v_2 e v_3 ; invece se $e_{max} = 2$ la politica *low complexity* è conservativa per $e = 1$ trasmettendo solo v_3 ed è aggressiva per $e = e_{max} = 2$, trasmettendo v_1, v_2 e v_3 .

Capitolo 4

Implementazione e risultati numerici

Il modello del sistema considerato è stato utilizzato per implementare in matlab le tre diverse politiche di gestione energetica (greedy, balanced, low complexity) presentate nel capitolo precedente. Innanzitutto il sistema è stato implementato in via teorica utilizzando la teoria delle catene di Markov presentata nel capitolo 2; in secondo luogo si è proceduto alla validazione dei risultati così ottenuti tramite opportune simulazioni dell'andamento del sistema.

Sono stati perseguiti tre obiettivi: viene mostrato come i risultati ottenuti mediante l'implementazione teorica del sistema siano consistenti con quelli ottenuti mediante le simulazioni dello stesso; in secondo luogo si procede a determinare e commentare, per ciascuna delle tre politiche, l'andamento della reward media normalizzata in funzione della capacità massima del buffer; in terzo luogo, si confrontano gli andamenti della reward media normalizzata ottenuti per ciascuna delle tre diverse politiche per arrivare a determinare quale delle tre sia più conveniente e sotto quali condizioni.

4.1 Implementazione matlab

Per implementare il modello del sistema in matlab, si è assunto che i pacchetti dati ricevibili dal dispositivo siano modellati da variabili aleatorie discrete, indipendenti e identicamente distribuite. Poiché il modello è stato generalizzato assumendo che in ogni istante k possa anche non arrivare alcun pacchetto dati, per indicare il caso in cui non ne viene ricevuto alcuno, è stato introdotto il pacchetto $v_0 = 0$ con la relativa probabilità p_0 . Quindi il

dispositivo ha una probabilità p_0 di ricevere il pacchetto $v_0 = 0$, ovvero ha una probabilità p_0 di non ricevere alcun pacchetto.

Per l'implementazione del sistema sono stati assegnati i seguenti valori numerici:

v	valore	probabilità	importanza
v_0	0	$p_0 = 0.01$	–
v_1	2	$p_1 = 0.3$	bassa
v_2	4	$p_2 = 0.3$	media
v_3	6	$p_3 = 0.39$	alta

dove p_i indica la probabilità di ricevere il pacchetto v_i ad un certo tempo k .

Come i pacchetti dati, anche i quanti di energia sono rappresentati da variabili aleatorie indipendenti e identicamente distribuite; la probabilità che in un dato intervallo arrivi un quanto di energia è $\lambda = p_2 + p_3 = 0.69$.

L'implementazione matlab delle tre politiche mediante l'uso della teoria delle catene di Markov si basa su quattro passi fondamentali: la costruzione della matrice di transizione per la catena di Markov che rappresenta il processo degli arrivi dei pacchetti dati, la costruzione dell'analogica matrice per la catena di Markov che rappresenta il processo degli arrivi e delle partenze dei quanti di energia; la costruzione del vettore delle probabilità stazionarie del sistema, e la computazione della reward in funzione della capacità massima del buffer.

4.1.1 Matrice di transizione dei pacchetti dati

La matrice di transizione per il processo markoviano di generazione dei pacchetti dati, ricordando la definizione data nel capitolo 2, è data da:

$$P_v = \begin{bmatrix} P_{v_0,v_0} & P_{v_0,v_1} & P_{v_0,v_2} & P_{v_0,v_3} \\ P_{v_1,v_0} & P_{v_1,v_1} & P_{v_1,v_2} & P_{v_1,v_3} \\ P_{v_2,v_0} & P_{v_2,v_1} & P_{v_2,v_2} & P_{v_2,v_3} \\ P_{v_3,v_0} & P_{v_3,v_1} & P_{v_3,v_2} & P_{v_3,v_3} \end{bmatrix} = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \\ p_0 & p_1 & p_2 & p_3 \\ p_0 & p_1 & p_2 & p_3 \\ p_0 & p_1 & p_2 & p_3 \end{bmatrix} \quad (4.1)$$

dove $P_{v_i,v_j} = P[V_{k+1} = v_j | V_k = v_i]$, ovvero P_{v_i,v_j} indica la probabilità che all'istante $k + 1$ arrivi il pacchetto j dato che all'istante k è arrivato il pacchetto i ; e dove la seconda uguaglianza deriva dal fatto che gli arrivi sono

indipendenti, perciò la probabilità che in un certo istante arrivi il pacchetto i -esimo è indipendente da quale pacchetto sia arrivato nell'istante precedente.

La matrice di transizione dei pacchetti dati è uguale per tutte e tre le politiche analizzate.

4.1.2 Matrice di transizione dei i quanti di energia

La costruzione della matrice di transizione (a un passo) della catena di Markov del processo degli arrivi e delle partenze dei quanti di energia, è diversa a seconda della politica implementata. Come visto nel capitolo 2, la struttura generale della matrice è la seguente:

$$\begin{bmatrix} P_{0,0} & P_{0,1} & \dots & P_{0,e_{max}} \\ P_{1,0} & P_{1,1} & \dots & P_{1,e_{max}} \\ \vdots & \vdots & \ddots & \vdots \\ P_{e_{max},0} & P_{e_{max},1} & \dots & P_{e_{max},e_{max}} \end{bmatrix} \quad (4.2)$$

dove $P_{i,j} = P[E_{k+1} = j | E_k = i]$ con $i, j \in \{0, \dots, e_{max}\}$; ovvero $P_{i,j}$ è la probabilità che l'energia nel buffer all'istante $k + 1$ sia j dato che all'istante k era i .

Poiché in ogni intervallo di tempo, per quanto assunto nel capitolo 3, può arrivare al massimo un quanto di energia e può essere consumato al massimo un quanto di energia, il processo secondo il quale evolve la carica nella batteria è un caso di processo *QBD*, ovvero Quasi-Birth-and-Death process. In un processo di questo tipo le evoluzioni del sistema tra due istanti consecutivi possono essere solo di tre tipi: lo stato del sistema passa allo stato successivo (birth), lo stato rimane invariato, lo stato regredisce allo stato precedente (death) [12]. Nella matrice di transizione di un processo QBD gli unici termini non nulli sono quelli del tipo: $P_{i,i}, P_{i,i+1}, P_{i,i-1}$. Dunque la struttura generale della matrice di transizione del QBD che rappresenta l'evoluzione della carica nella batteria, è:

$$\begin{bmatrix} P_{0,0} & P_{0,1} & 0 & \dots & 0 \\ P_{1,0} & P_{1,1} & P_{1,2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & P_{e_{max},e_{max}-1} & P_{e_{max},e_{max}} \end{bmatrix} \quad (4.3)$$

Chiamiamo μ_i la probabilità media, dipendente dalla politica adottata, che venga trasmesso un pacchetto dati, ovvero la probabilità che venga consumato un quanto di energia, quando il livello di carica nella batteria è i .

Ricordando inoltre che la probabilità che arrivi un quanto di energia è λ , abbiamo che:

- $P_{i,i+1} = \lambda \cdot (1 - \mu_i)$: la probabilità che il livello di energia nel buffer aumenti di un quanto, è pari alla probabilità che si verifichi la coppia di eventi “arriva un quanto di energia \wedge non devo trasmettere alcun pacchetto dati” ($P = 1 - \mu_i$); poiché le due MC dei pacchetti dati e dei quanti di energia sono indipendenti, la probabilità che si verifichino entrambi gli eventi appena citati, è semplicemente il prodotto delle probabilità che si verifichino i singoli eventi.
- $P_{i,i} = \lambda \cdot \mu_i + (1 - \lambda) \cdot (1 - \mu_i)$: la probabilità che il livello della batteria rimanga invariato nell’intervallo k è pari alla probabilità che si verifichi una delle due coppie di eventi: “arriva un quanto di energia \wedge trasmetto un pacchetto dati” oppure “non arriva alcun quanto di energia \wedge non trasmetto alcun pacchetto dati”. Come per il caso precedente, vista l’indipendenza delle due MC, $P_{i,i}$ è data dalla somma dei prodotti delle due coppie di eventi
- $P_{i,i-1} = (1 - \lambda) \cdot \mu_i$: la probabilità che il livello di energia nel buffer diminuisca di uno, è pari alla probabilità che si verifichino gli eventi “non arriva alcun quanto di energia \wedge trasmetto un pacchetto dati”.

questo per $i \neq 0 \wedge i \neq e_{max}$.

Nel caso $i = 0$, ovvero nel caso di buffer vuoto, la probabilità che il dispositivo trasmetta un pacchetto dati è nulla poiché non c’è energia nel buffer, quindi sostituendo $\mu_i = 0$ in $P_{0,1}$ si ottiene $P_{0,1} = \lambda$; analogamente, $P_{0,0} = \lambda \cdot 0 + (1 - \lambda) \cdot (1 - 0) = (1 - \lambda)$.

Nel caso $i = e_{max}$, il buffer è in overflow, conseguentemente la probabilità che il buffer rimanga nello stato e_{max} è la somma dei contributi $P_{i,i} + P_{i,i+1}$: infatti anche quando si verificasse la possibilità di accumulare un quanto di energia e quindi passare dallo stato i allo stato $i + 1$, essendo il buffer in overflow questo non può avvenire, perché il buffer è già pieno; dunque abbiamo che $P_{e_{max},e_{max}} = 1 - P_{e_{max},e_{max}-1} = 1 - \mu(1 - \lambda)$.

Mentre il tasso degli arrivi λ è lo stesso per tutte e tre le politiche, il termine μ_i varia. Per le politiche greedy e balanced, μ è costante: $\mu_{greedy} = p_1 + p_2 + p_3$ e $\mu_{balanced} = \lambda$. Dunque la matrice di transizione per entrambe è:

$$\begin{bmatrix} 1 - \lambda & \lambda & 0 & \dots & 0 & 0 \\ (1 - \lambda)\mu & \lambda\mu + (1 - \lambda)(1 - \mu) & \lambda(1 - \mu) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & (1 - \lambda)\mu & 1 - (1 - \lambda)\mu \end{bmatrix}$$

Per la politica low complexity invece, per quanto detto nel capitolo 3 circa la sua struttura, μ dipende dal livello di energia nel buffer e dalla capacità massima dello stesso. Per semplicità si riporta la struttura della matrice solo per $e_{max} = 3$:

$$\begin{bmatrix} 1 - \lambda & \lambda & 0 & 0 \\ \mu_1(1 - \lambda) & \lambda\mu_1 + (1 - \lambda)(1 - \mu_1) & \lambda(1 - \mu_1) & 0 \\ 0 & \mu_2(1 - \lambda) & \lambda\mu_2 + (1 - \lambda)(1 - \mu_2) & \lambda(1 - \mu_2) \\ 0 & 0 & (1 - \lambda)\mu_3 & 1 - (1 - \lambda)\mu_3 \end{bmatrix}$$

con $\mu_1 = p_3$, $\mu_2 = \lambda$, $\mu_3 = p_1 + p_2 + p_3$.

L'implementazione matlab di tutta la matrice per la politica low complexity è la seguente:

```

1   Te = zeros(emax+1, emax+1);
2   Te(1,1)= 1-lambda;
3   Te(1,2)= lambda;
4   etaLow=p3;
5   etaHigh=p1+p2+p3;
6   for i=2:emax
7       if i==2
8           mu= etaLow;
9       elseif i>=3 && i<=emax
10          mu = lambda;
11      end
12      Te(i,i-1) = (1-lambda)*mu;
13      Te(i,i+1) = lambda*(1-mu);
14      Te(i,i) = lambda*mu + (1-lambda)*(1-mu);
15  end
16  if emax==1
17      mu=lambda;
18  end
19  if emax==2

```

```

20     mu=etaHigh;
21     end
22     if emax>=3
23         mu=etaHigh;
24     end
25     Te(emax+1,emax) = (1-lambda)*mu;
26     Te(emax+1,emax+1) = 1-Te(emax+1,emax);

```

4.1.3 Vettore delle probabilità stazionarie

Tramite le due matrici di transizione per i pacchetti dati e per i quanti di energia, è possibile costruire il vettore delle probabilità stazionarie, ovvero:

$$\pi = [\pi(0,0), \dots, \pi(e_{max},0), \pi(0,v_1), \dots, \pi(e_{max},v_1), \dots, \pi(e_{max},v_3)] \quad (4.4)$$

dove $\pi(e, v_i)$ indica la probabilità che dopo un numero infinito di transizioni, il sistema si trovi nello stato (e, v_i) . Per quanto detto nel capitolo 2, considerato che i due processi degli arrivi dei quanti di energia e dei pacchetti dati sono indipendenti, si ha che $\pi(e, v_i) = \pi(e)p_i$. Quindi per computare la matrice, è sufficiente trovare π_e risolvendo il sistema di $e_{max} + 1$ equazioni presentato nel capitolo 2, e moltiplicare poi opportunamente per $p_i, i \in \{0, 1, 2, 3\}$:

```

1     be = zeros(1,emax+2);
2     be(1,emax+2)=1;
3     Qe = horzcat(Te-eye(emax+1), ones(emax+1,1));
4     Pie=be/Qe; % prob stazionarie energia
5     Pi=horzcat(Pie*p0, Pie*p1, Pie*p2, Pie*p3);

```

4.1.4 Reward media normalizzata

Il guadagno (reward) medio normalizzato, nel sistema analizzato, è definito da:

$$G(S_0, \mu) = \sum_{e=1}^{e_{max}} \frac{\sum_{i=1}^3 v_i \cdot \pi(e, v_i) \cdot \mu(1; e, v_i)}{\sum_{i=3-j^*}^3 v_i \cdot p_i} \quad (4.5)$$

La quantità a numeratore indica la reward media ottenibile adottando la particolare politica μ . La quantità a denominatore corrisponde invece alla reward media massima che si otterrebbe, se si disponesse di un buffer di capacità infinita e quindi se fossimo nel caso ideale in cui sono sempre disponibili quanti di energia per la trasmissione.

Esplicitiamo il significato di j^* che, nel caso considerato in cui la probabilità di ricevere un quanto di energia è $\lambda (= p_2 + p_3)$, assume un valore pari a 1.

Il caso ideale si verifica quando la probabilità di trasmettere un pacchetto è uguale alla probabilità di riceverlo, ovvero quando in media il dispositivo è in grado di trasmettere un pacchetto ogni volta che lo riceve; in sostanza ciò si verifica se la probabilità che il dispositivo sia in energy outage è nulla. Infatti la probabilità di ricevere un dato pacchetto v_i è:

$$p_i = \pi(0, v_i) + \pi(1, v_i) + \cdots + \pi(e_{max}, v_i) \quad (4.6)$$

ma la probabilità di trasmetterlo è sempre minore di p_i :

$$p_{tx,i} = \pi(1, v_i) + \cdots + \pi(e_{max}, v_i) = p_i - \pi(0, v_i) < p_i \quad (4.7)$$

Tuttavia, poiché non abbiamo potere di azione sul processo degli arrivi dei quanti di energia, anche nel caso ideale c'è un limite non superabile: il tasso di arrivi dei quanti di energia. Se, come nel sistema considerato, la probabilità che arrivi un quanto è $\lambda = p_2 + p_3$, il caso ideale corrisponde alla situazione in cui il dispositivo trasmette v_2 e v_3 con la stessa probabilità con cui li riceve, ovvero p_2 e p_3 ; non si può considerare come caso ideale il caso in cui il dispositivo trasmette sempre v_1 , v_2 e v_3 , perché in questo caso, anche se disponessimo di una batteria di capacità infinita, il tasso delle trasmissioni sarebbe maggiore del tasso degli arrivi, quindi la probabilità che la batteria vada in energy outage non sarebbe più nulla.

Dunque nel nostro caso $j^* = 1$, quindi la reward media massima ottenibile, corrispondente alla situazione ideale in cui la probabilità di *trasmettere* v_2 o v_3 coincide con la probabilità di *ricevere* v_2 o v_3 è:

$$G_{max} = v_2 \cdot p_{tx,2} + v_3 \cdot p_{tx,3} = v_2 \cdot p_{rc,2} + v_3 \cdot p_{rc,3} = v_2 p_2 + v_3 p_3$$

Per esempio, considerando un sistema formato da un buffer di capacità massima $e_{max} = 2$, per il quale venga adottata la politica balanced definita da $\mu(1; e, v_2) = \mu(1; e, v_3) = 1 \quad \forall e \in \mathcal{E}_0$: la reward media normalizzata è data da:

$$\begin{aligned} G &= \frac{v_2 \cdot \pi(1, v_2) + v_3 \cdot \pi(1, v_3)}{v_2 p_2 + v_3 p_3} + \frac{v_2 \cdot \pi(2, v_2) + v_3 \cdot \pi(2, v_3)}{v_2 p_2 + v_3 p_3} = \\ &= \frac{v_2 [\pi(1, v_2) + \pi(2, v_2)] + v_3 [\pi(1, v_3) + \pi(2, v_3)]}{v_2 p_2 + v_3 p_3} \end{aligned} \quad (4.8)$$

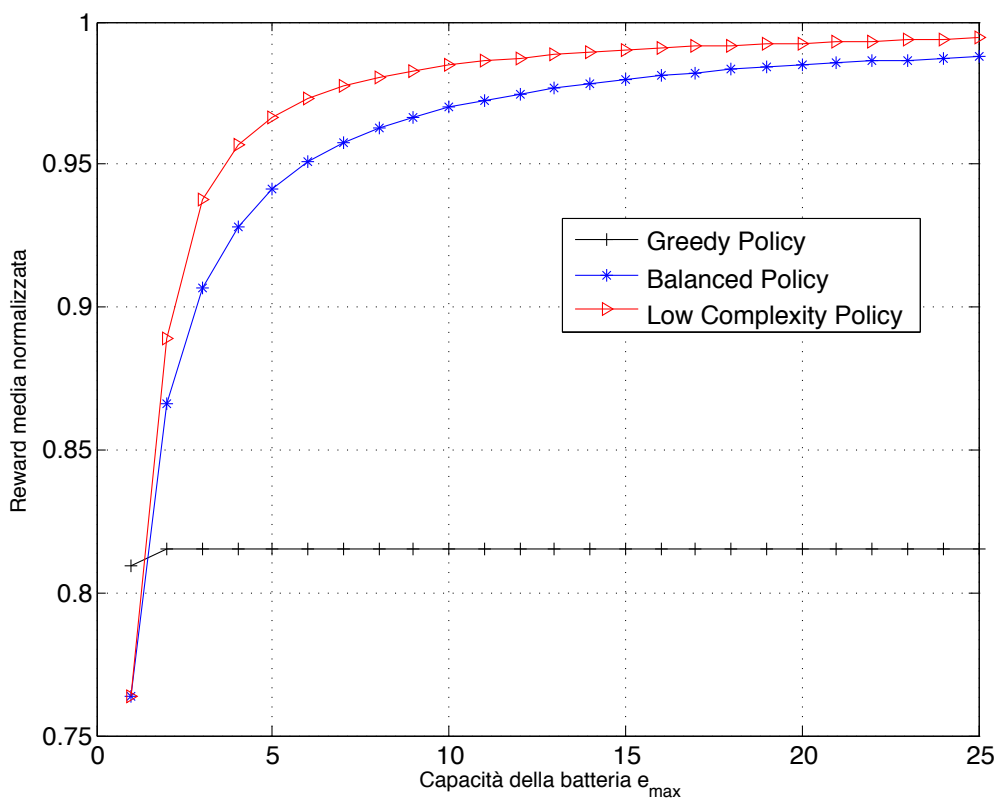


Figura 4.1: reward media normalizzata vs. capacità massima della batteria per le politiche greedy, balanced e low complexity. tasso degli arrivi: $\lambda = 0.69$

4.2 Reward media vs. capacità del buffer

L'andamento della reward media in funzione della capacità massima del buffer, per ciascuna delle tre politiche è stato riportato in figura 4.1.

Politica greedy

L'andamento della reward media ottenibile utilizzando la politica greedy, è costante a 0.82 circa e non varia al variare della capacità massima della batteria. Questo è dovuto al fatto che sotto la politica greedy, il dispositivo trasmette sempre fintanto che ha energia, conseguentemente il dispositivo rischia di trovarsi in energy outage e quindi per trasmettere sempre tutti i pacchetti anche quelli meno importanti, finisce alla fine ad avere una reward normalizzata abbastanza bassa. Nessun vantaggio si ha quindi ad aumentare la capacità sotto la politica greedy.

Politica balanced

Per quanto riguarda la politica balanced, l'andamento è ben diverso: si ha un notevole incremento della reward media all'aumentare della capacità massima della batteria per valori di e_{max} compresi tra 1 e 10; invece per valori di e_{max} maggiori di 10, il guadagno in termini di reward non è più ugualmente significativo. Tuttavia, al tendere di e_{max} all'infinito, la reward ottenibile mediante la politica balanced tende al valore ideale 1. Dunque la politica balanced ha una performance migliore sotto due aspetti rispetto alla politica greedy: innanzitutto la reward media normalizzata della politica balanced è più alta per ogni valore della capacità massima del buffer fatta eccezione per $e_{max} = 1$; in secondo luogo, l'andamento della politica balanced tende asintoticamente a 1 all'aumentare di e_{max} , dunque aumentando la capacità della batteria si possono migliorare le prestazioni del dispositivo, mentre con la politica greedy, le prestazioni non sono migliorabili anche se si potesse disporre di una batteria di capacità maggiore.

Politica low complexity

Nel caso della politica low complexity, l'andamento della reward media normalizzata in funzione della capacità massima della batteria è molto simile a quello della politica balanced (i due andamenti coincidono addirittura nel caso limite $e_{max} = 1$ per come è stata costruita la politica low complexity): all'aumentare della capacità della batteria, le prestazioni migliorano significativamente per valori di e_{max} minori di 5, mentre per valori maggiori il miglioramento è più modesto. Anche in questo caso, al tendere della capacità all'infinito, la reward media tende al valore ideale massimo ottenibile.

Confrontando gli andamenti della reward in funzione della capacità del buffer, emerge che le due politiche a soglia, ovvero quella balanced e quella low complexity, hanno prestazioni migliori rispetto a quelle della politica greedy che non è a soglia: le prestazioni relative migliorano tanto più è alta la capacità massima del buffer, raggiungendo valori anche superiori al 20%, come si può vedere dalla figura 4.2. Inoltre la politica low complexity, avendo una soglia più "raffinata" (tale che il dispositivo sia conservativo quando c'è poca energia nel buffer, bilanciato quando ce n'è media e aggressivo quando ce n'è molta) ha una performance leggermente migliore della politica balanced. Tuttavia come si può estrapolare dalla figura 4.3, questo miglioramento è modesto se confrontato con quello delle due politiche rispetto alla politica greedy, e diventa sempre più modesto all'aumentare della capacità massima della batteria, arrivando ad essere inferiore al 2.5% per valori di e_{max} maggiori di 5.

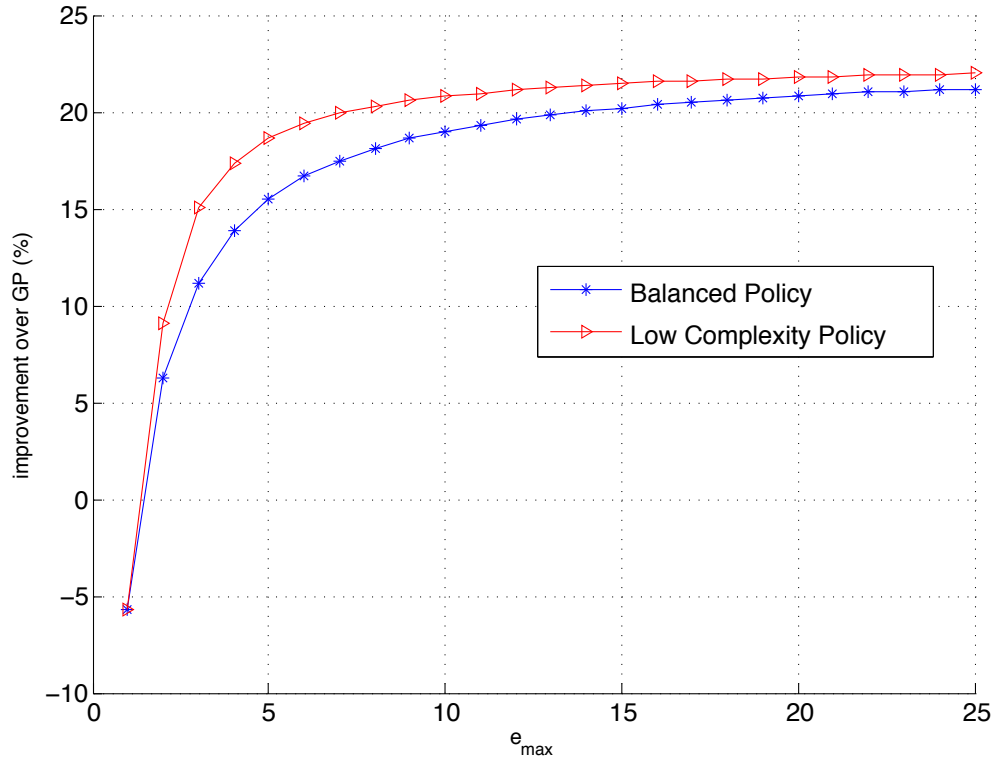


Figura 4.2: miglioramento di LCP e BP rispetto a GP

4.3 Simulazioni dell'andamento del sistema

Per verificare i risultati ottenuti tramite la teoria della catene di Markov, circa l'andamento della reward media normalizzata in funzione della capacità della batteria, sono state eseguite delle simulazioni tramite matlab.

Per ogni politica analizzata, sono state eseguite 100000 simulazioni dell'andamento del sistema, per ogni valore della capacità della batteria. Ciascuna simulazione consta di 300 transizioni tra gli stati del sistema, un numero sufficientemente grande per approssimare un valore infinito di transizioni, com'è stato rilevato facendo diverse prove al variare di questo parametro.

Il risultato delle simulazioni è presentato in figura 4.4. Si può notare come l'andamento del sistema simulato sia consistente con i risultati ottenuti dall'implementazione teorica.

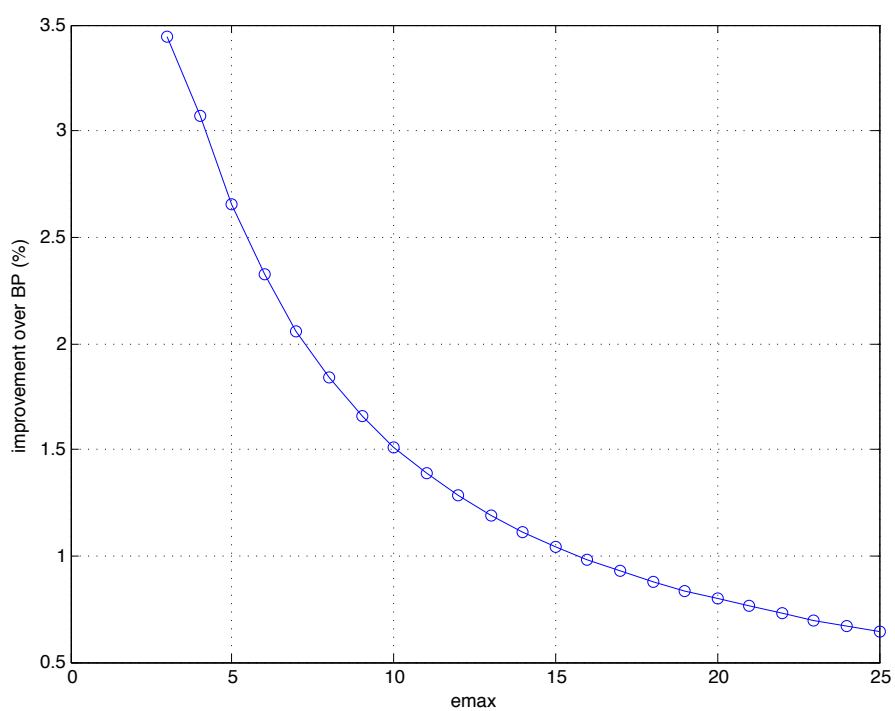


Figura 4.3: miglioramento della low complexity policy rispetto alla balanced policy

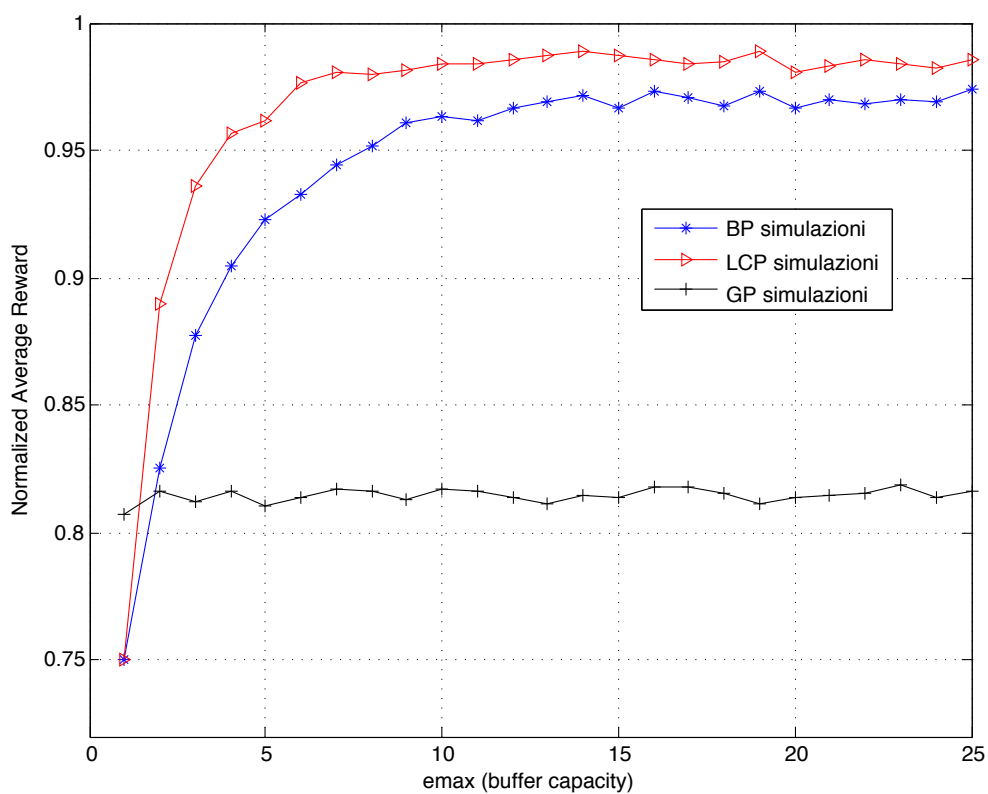


Figura 4.4: simulazioni dell'andamento della reward in funzione della capacità massima della batteria

Capitolo 5

Conclusioni e sviluppi futuri

5.1 Conclusioni

In questa tesi è stato affrontato il problema di definire politiche di gestione energetica efficienti ma non troppo complicate da realizzare per dispositivi EHD. Sono state costruite tre politiche: una non a soglia (politica *greedy*), una a soglia “semplice” (politica *balanced*) e una a soglia un più elaborata (*low complexity*); si è studiato l’andamento del guadagno medio normalizzato di ciascuna politica in funzione della capacità del buffer e poi tali andamenti sono stati confrontati tra loro. In particolare è stato ampliato il modello considerato in [10] includendo anche la possibilità che il dispositivo possa non ricevere alcun pacchetto dati in un determinato intervallo di tempo.

Dal confronto tra gli andamenti dei guadagni delle varie politiche si è potuto constatare, come supposto in partenza [10], che una politica a soglia (come quelle *balanced* e *low complexity*) ha una performance nettamente superiore rispetto ad una politica non a soglia (come la politica *greedy*). Inoltre una politica a soglia un po’ più “intelligente” della semplice politica *balanced* dimostra di avere prestazioni ancora superiori.

Nel caso di dispositivi EHD, poichè essi non dispongono di grandi capacità computazionali, l’implementazione di una politica ottima (che deve essere valutata numericamente mediante un programma lineare [15]) può diventare troppo dispendiosa a causa dell’eccessiva capacità di calcolo richiesta al dispositivo; d’altro canto adottare una politica troppo semplice come quella a soglia o quella *balanced* può compromettere eccessivamente le prestazioni. In questo scenario la politica *low complexity* può essere una valida soluzione in quanto rappresenta un buon compromesso tra prestazioni e semplicità computazionale.

Inoltre possiamo notare come le prestazioni di una politica a soglia mi-

glierino all'aumentare della capacità massima della batteria, tendendo a raggiungere il valore ideale per e_{max} che tende all'infinito, mentre ciò non accade per una politica non a soglia, la cui performance è indipendente da quanta energia possa essere accumulata nella batteria.

Confrontando i guadagni medi ottenibili adottando la politica *balanced* e quella *low complexity*, emerge come il miglioramento delle prestazioni che si ottengono adottando la seconda, sia in realtà modesto per valori della capacità massima $e_{max} > 5$, e tenda a zero al tendere della capacità massima all'infinito. Ciò si traduce in un'altra considerazione: se si dispone di una batteria di capacità medio-alta, può essere adottata la politica *balanced* al posto della *low complexity*, senza il rischio di significative perdite in termini di prestazioni.

5.2 Sviluppi futuri

In questo lavoro il processo degli arrivi dei quanti di energia e dei pacchetti dati, sono stati modellati come due catene di Markov in cui ciascuna è costituita da variabili aleatorie indipendenti e quindi scorrelate. Ovvero la probabilità che arrivi un quanto di energia all'istante k , non varia a seconda che all'istante $k - 1$ sia o non sia arrivato un quanto di energia. Così come la probabilità di ricevere un pacchetto molto importante all'istante k è indipendente da quale pacchetto sia arrivato all'istante $k - 1$.

Il prossimo passo consiste nel valutare la correlazione tra gli arrivi dei quanti di energia [8]. Per esempio, nel caso di un pannello fotovoltaico, questo scenario sarebbe adatto a rappresentare l'alternanza di notte e giorno, nella misura in cui di giorno possono arrivare quanti e mentre di notte no. In questo caso la probabilità che arrivi un quanto dato che ne è appena arrivato uno, non è uguale, bensì maggiore della probabilità che arrivi un quanto dato che all'istante precedente non ne è arrivato nessuno.

Il terzo caso da analizzare consiste invece nel tenere conto della correlazione tra gli arrivi dei pacchetti dati. In questo scenario i pacchetti dati che arrivano vanno inseriti in una coda, e all'istante k possono essere prese non due ma tre decisioni: trasmettere, scartare o aspettare. Quindi lo spazio di stato delle azioni possibili diventa $\mathcal{A} = \{trasmettere, scartare, aspettare\}$. Lo stato del sistema al tempo k viene definito da $S_k = (E_k, V_k, Q_k)$, dove le tre variabili indicano rispettivamente il livello di energia nella batteria, il valore del pacchetto dati corrente e la lunghezza della coda al tempo k . La definizione di politica non cambia: μ è una misura di probabilità su \mathcal{A} parametrizzata da S_k .

Intuitivamente, anche in questo caso la politica ottima avrà una struttura a soglia, dove però la soglia non dipenderà solo da E_k e V_k , ma anche dalla lunghezza della coda dei pacchetti dati: per esempio, se ho un pacchetto poco importante, un livello medio di energia nella batteria e la coda dei pacchetti vuota, mi converrà aspettare, se invece lo scenario è lo stesso ma la coda dei pacchetti è piena, mi converrà trasmettere.

Bibliografia

- [1] Leonardo Badia. «Transmission policies for wireless sensors networks with energy harvesting». In: *Proceedings of CTTC seminar* (2012).
- [2] Nevio Benvenuto e Michele Zorzi. *Principles of communications, networks and systems*. 2011.
- [3] John A. Gubner. *Probability and random processes for electrical and computer engineers*. 2012.
- [4] Badia L. et al. «An optimization framework for joint sensor deployment, link scheduling and routing in underwater sensor networks». In: *ACM Mobile Computing and Communication Review (MC2R)* (2007).
- [5] Paul L. e Mitcheson D. «Energy harvesting for human wearable and implantable bio-sensors». In: *Proceedings of IEEE EMBS* (2010).
- [6] K. Lin, J. Yu et al. «Helimote: enabling long-lived sensor networks through solar energy harvesting». In: *Proceedings of ACM SenSys* (2005).
- [7] Nicolò Michelusi, Leonardo Badia et al. «Correlated energy generation and imperfect state of charge knowledge in energy harvesting devices». In: *Proceedings of IWCMC* (2012).
- [8] Nicolò Michelusi, Leonardo Badia et al. «Energy generation and state-of-charge knowledge in energy harvesting devices». In: *Proceedings of IWCMC (invited paper)* (2012).
- [9] Nicolò Michelusi, Leonardo Badia et al. «Impact of battery degradation on optimal management policies of harvesting-based wireless sensor devices». In: *Proceedings of INFOCOM* (2013).
- [10] Nicolò Michelusi, Kostas Stamatiou e Michele Zorzi. «On optimal transmission policies for energy harvesting devices». In: *Proceedings of Information Theory and Applications Workshop* (2012).

- [11] Nicolò Michelusi et al. «Operation policies for energy harvesting devices with imperfect state of charge knowledge». In: *Proceedings of IEEE International conference on communications* (2012).
- [12] R. Marcel Neuts. «Matrix-geometric solutions in stochastic models: An algorithmic approach». In: Baltimore, 352pp: The Johns Hopkins University Press, 1981.
- [13] Perpetuum. *Vibration Energy Harvesting applications*. 2012. URL: <http://www.perpetuum.com/resources.asp>.
- [14] Warren B. Powell. *Approximate dynamic programming*. 2010.
- [15] K. W. Ross. «Randomized and past-dependent policies for Markov decision processes with multiple constraints». In: *Operations research* (1989).
- [16] T. Starner. «Human power wearable computing». In: *IBM systems journal, volume 35(3), pp. 618-629* (1996).
- [17] T. Starner e J. A. Paradiso. «Energy scavenging for mobile and wireless electronics». In: *Proceedings of IEEE Pervasive Computing, vol. 4, pp 18-27* (2005).