



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**



**DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE**

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA DELL'INFORMAZIONE

**“POLICY DI GESTIONE ED EFFICIENZA ENERGETICA PER SISTEMI
BASATI SU ENERGY HARVESTING”**

Relatore: Leonardo Badia

Laureando: Nicola Toffolo

ANNO ACCADEMICO 2019 - 2020

Data di laurea 23/09/2020

Indice

1	Introduzione	1
2	Sviluppo di una politica battery-saving in un sistema basato su energy harvesting	2
2.1	Nozioni fondamentali	2
2.2	Modello	3
2.2.1	Caratterizzazione del livello di carica della batteria Q_k . . .	3
2.2.2	Introduzione a policy e problema di Ottimizzazione	4
2.3	Ottimizzazione	6
2.3.1	Risoluzione algoritmica	7
2.4	Estrapolazione delle probabilità di deperimento dai dati	7
2.4.1	Modello deterministico	8
2.4.2	Modello stocastico	9
2.5	Risultati	10
2.6	Estensione a batterie reali	15
3	Soluzione power-saving per WSN con energy harvesting	16
3.1	Nozioni fondamentali	16
3.2	Modello	17
3.2.1	Circuito EAI	17
3.2.2	Nodo sensore	18
3.2.3	Programma energy-aware	18
3.2.4	Calcolo di V_{end} e dell'energia richiesta	19
3.3	Sperimentazione	20
3.3.1	Strumentazione	20
3.3.2	Configurazione	21
3.4	Risultati	22
3.4.1	Configurazione I	22
3.4.2	Configurazione II	23
3.4.3	Configurazione III	23
3.4.4	Confronto	24
4	Conclusioni	24
	Riferimenti bibliografici	25

1 Introduzione

La sempre maggiore e più rapida diffusione di reti di dispositivi in grado di comunicare tra loro e dotati di capacità di rilevamento (Wireless Sensor Networks [2]) ha portato con sé la difficoltà nel gestire la loro autonomia da un punto di vista energetico. Essi devono infatti garantire un funzionamento idealmente perpetuo senza la necessità di un controllo costante; tale obiettivo può essere raggiunto usando tecniche di energy harvesting (EH) e un buon uso dell'energia immagazzinata localmente [14]. Si può allora pensare a una batteria ricaricabile come un buffer nel quale viene stoccata energia grazie al processo di EH e chiamare *energy-aware operation policy* un algoritmo che si occupi della gestione di quest'ultima al fine di assicurare l'operatività del dispositivo nel tempo. Quest'elaborato, tuttavia, non considera alcune peculiarità di una batteria reale (*battery leakage* [4], possibili perdite di energia accumulata [23], conoscenza non precisa dello stato di carica [13]) e si vuole piuttosto concentrare sul deterioramento della stessa e la conseguente diminuzione della sua capacità iniziale, variabile a seconda di come essa è stata usata [11]. Tale fenomeno è infatti dovuto a importanti cicli di scarica e, più la batteria viene scaricata, più velocemente avverrà il suo deterioramento, in particolare per le batterie a ioni di litio (Li-Ion). Verrà quindi proposto un modello markoviano che tenga esplicitamente conto di questo aspetto, ne sarà confermata l'ottima sovrapposizione con una batteria reale e verrà infine descritto un problema di ottimizzazione sulla durata della batteria [12].

Si passerà quindi ad analizzare un problema tipico dei WSN basati su EH, cioè l'ingente richiesta di energia che li caratterizza. Questa determina spesso un gap tra l'energia generata dall'harvester e quella necessaria al WSN per portare a termine le operazioni ad esso assegnate; si rende così necessario ridurre i consumi del nodo e aumentare la quantità di energia da accumulare. Gli argomenti sono organizzati in quest'ordine:

- Nella sezione 2.1 vengono riportati alcuni dei modelli già presenti di cui viene considerata la possibile degradazione della batteria.
- Nella sezioni 2.2 e 2.3 si definisce e si sviluppa il problema di ottimizzazione.
- Nella sezione 2.4 vengono ricavate le probabilità di deperimento da dati sperimentali, nella sezione 2.5 alcuni risultati numerici.
- Nella sezione 2.6 si discute una possibile estensione del modello alla realtà.
- Nella sezione 3.1 si passa all'analisi del secondo problema già accennato, introducendo la soluzione che verrà in seguito studiata.
- Nella sezione 3.2 viene descritto e analizzato il nuovo modello.
- Nella sezione 3.3 viene illustrata la strumentazione utilizzata per la sperimentazione.

- Nella sezione 3.4 si accerta, con differenti configurazioni degli strumenti, l'efficienza del suddetto modello,
- La sezione 4 conclude l'elaborato.

2 Sviluppo di una politica battery-saving in un sistema basato su energy harvesting

2.1 Nozioni fondamentali

Si osservi innanzitutto la Fig. 1: in un sistema di questo tipo, il *carico* è costituito dall'hardware che ha bisogno di alimentazione, la quale viene fornita dalla *power processing unit*, che si interfaccia con la batteria e la fonte di energia esterna. Vengono ora fornite alcune definizioni:

- *cycling*: processo di carica/scarica della batteria.
- *Depth of Discharge* (DoD): percentuale D di carica ceduta dalla batteria durante la sua scarica.
- C_0 : capacità nominale della batteria.
- N_{cyc} : numero di cicli che garantisce una batteria con DoD $D = 1$ prima che C_0 scenda al di sotto di una soglia prestabilita.
- $Q(N_{cyc})$: carica ottenuta dalla batteria dopo N_{cyc} alla DoD D .

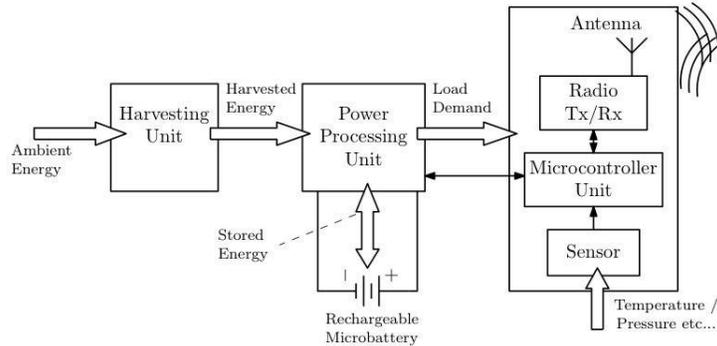


Figura 1: Diagramma a blocchi di un nodo sensore basato su EH [12]

Ci si potrebbe allora aspettare la seguente relazione

$$Q(N_{cyc}) = N_{cyc} \cdot C_0 \cdot D \quad (1)$$

ma ciò non accade perché da un lato la batteria va incontro a un deperimento che porta la sua capacità C_0 a livelli insostenibili, dall'altro perché il processo

stesso di deperimento mostra una dipendenza dall'uso della batteria [17], [20], [21]: se i cicli di scarica presentano una DoD non eccessiva, il ciclo di vita della batteria sarà allungato. Un modello empirico che può invece esser coerente con i dati di una buona gamma di batterie è

$$N_{cyc}(D) = N_{cyc,0} \cdot e^{\alpha(1-D)} \quad (2)$$

con $N_{cyc,0}$ che sta a indicare il numero di cicli con una DoD del 100%, e α una costante propria della batteria.

2.2 Modello

Si pensi a un sistema diviso in slot temporali in cui lo slot k -esimo è l'intervallo di tempo $[k\Delta t, k\Delta t + \Delta t)$, $k \in \mathbb{Z}^+$, Δt è la durata di uno slot. La batteria sia modellata come un buffer con capacità iniziale C_0 e sia uniformemente quantizzata a un certo numero di livelli di carica, con passo di quantizzazione $\Delta c \ll C_0$. Sia inoltre $q_{max} = \lfloor \frac{C_0}{\Delta c} \rfloor$ il massimo numero di quanti accumulabili a capacità nominale e si denoti l'insieme dei possibili livelli di carica con $Q = \{0, 1, \dots, q_{max}\}$. Poniamo poi $Q_{max}(k)$ la capacità della batteria al tempo k , e sia Q_k il livello di carica della batteria nel medesimo istante. Perciò, $Q_{max}(k) \leq Q_{max}(k-1)$ e $Q_{max}(0) = q_{max}$. Q_k ha la seguente evoluzione

$$Q_{k+1} = \min\{[Q_k - A_k]^+ + B_k, Q_{max}(k+1)\} \quad (3)$$

dove $[x]^+ = \max(x, 0)$. In questa relazione:

- $\{B_k\}$ è il processo di harvesting e prende valori in $\mathbf{B} = \{0, 1, \dots, B_{max}\}$. È possibile definire anche un *processo degli scenari* S_k modellandolo con una catena di Markov stazionaria con probabilità di transizione $p_S(s_{k+1}|s_k) \triangleq \Pr(S_{k+1} = s_{k+1}|S_k = s_k)$ e a valori in \mathbf{S} con stati finiti. Dato $S_k \in \mathbf{S}$, B_k è ottenuto da \mathbf{B} secondo la distribuzione $p_B(b_k|s_k) \triangleq \Pr(B_k = b_k|S_k = s_k)$. Se si chiama $\pi_S(s)$ la distribuzione di S_k in stato stazionario, allora il tasso medio di harvesting è

$$\bar{b} \triangleq \sum_{s \in \mathbf{S}} \pi_S(s) \sum_{b \in \mathbf{B}} b p_B(b|s) \quad (4)$$

- $\{A_k\}$ è il processo delle azioni, a valori in $\mathbf{A} = \{0\} \cup \{A_{min}, A_{min+1}, \dots, A_{max}\}$, gestito dall'*Energy Harvesting Device controller* (EHD). A_{min} e A_{max} rappresentano rispettivamente il minimo e il massimo che il carico può richiedere.

2.2.1 Caratterizzazione del livello di carica della batteria Q_k

Ponendo l'attenzione sulla relazione (3), si ha che se $A_k > Q_k$ il nodo sensore cerca di ottenere più energia di quanta ne sia presente nella batteria: viene così precocemente esaurita e il nuovo stato di carica Q_{k+1} eguaglierà la quantità di energia coinvolta nel processo di harvesting nello slot di tempo presente. Si

assume quindi che l'EHD richiederà sempre $A_k \leq Q_k$. Altra importante considerazione è che tra batteria e EHD non vengono scambiati quanti di *carica*, bensì di *energia*, e tali quantità sono proporzionali se e solo se la batteria mantiene una differenza di potenziale costante nel suo ciclo di vita [11]. Passiamo al processo di deperimento della batteria; innanzitutto definiamo lo *stato di salute della batteria*, H_k , a valori in $\mathbf{H} = \{0, 1, \dots, H_{max}\}$. Per un certo H_k , la capacità della batteria al tempo k è

$$Q_{max}(k) = \left\lfloor \frac{H_k}{H_{max}} q_{max} \right\rfloor \quad (5)$$

e l'insieme di possibili livelli di carica è $Q(H_k) = \{0, 1, \dots, \left\lfloor \frac{H_k}{H_{max}} q_{max} \right\rfloor\}$. Ora, la probabilità di transizione tra lo stato $H_k = h$ e il successivo $H_{k+1} = h - 1$ è

$$p_H(h; q) \triangleq \Pr(H_{k+1} = h - 1 | H_k = h, Q_k = q) \quad (6)$$

e non è possibile che avvenga una transizione tra stati tra loro non contigui: la probabilità di rimanere nello stato h è di conseguenza $1 - p_H(h; q)$. Si considerino ora le seguenti assunzioni importanti:

- a) $p_H(h; q) > 0, \forall h \in \mathbf{H}, q \in \mathbf{Q}(h)$
- b) $p_H(h; q) \ll 1, \forall h \in \mathbf{H}, q \in \mathbf{Q}(h)$
- c) $p_H(h_1; q_1) \geq p_H(h_2; q_2), \forall h_2 \geq h_1, q_2 \geq q_1$

Quindi, a) indica che prima o poi si raggiungerà lo stato $H_k = 0$, e cioè la batteria giungerà a fine vita; b) che il processo di deperimento avviene in tempi molto più lunghi di quelli di *cycling* e comunicazione con l'EHD; c) che il suddetto processo aumenta di velocità più la batteria è degradata [11].

Resta essenziale osservare che, poiché la capacità della batteria diminuisce in continuazione, il modello (5), operando una quantizzazione, porta i valori che può assumere la capacità al discreto e perciò se H_{max} ha un valore piccolo, il modello non potrà essere compatibile con la realtà; d'altra parte, se questo valore è eccessivo, l'Ass. b) non è più assicurata. Ora, lo stato dell'EHD al tempo k sia $\mathbf{Z}_k = (Q_k, H_k, S_{k-1})$, a valori in $Q \times H \times S$. Mentre H_k può essere stimato da $Q_{max}(k)$, S_{k-1} si può valutare attraverso il criterio Maximum A Posteriori: nel nostro caso, $\hat{S}_{k-1} = \arg \max_s \Pr(S_{k-1} = s | B_0, \dots, B_{k-1})$. Infine, per semplicità, si consideri \mathbf{Z}_k perfettamente noto al controller EHD (si veda [18] per una possibile estensione al caso generale).

2.2.2 Introduzione a policy e problema di Ottimizzazione

Noto $\mathbf{Z}_k = (Q_k, H_k, S_{k-1})$, l'EHD determina $A_k \in \mathbf{A}$ al tempo k secondo una determinata *policy* μ_{H_k} , la quale è una misura di probabilità sullo spazio delle azioni \mathbf{A} . La ricompensa ottenuta nello slot k , nello stato \mathbf{Z}_k e azione A_k è

$$g(A_k, Q_k) = \begin{cases} 0, & A_k > Q_k, \\ g^*(A_k), & A_k \leq Q_k, \end{cases} \quad (7)$$

$g^*(A_k)$ è funzione non decrescente di A_k e vale $g^*(0) = 0$. Denotiamo anche gli *hitting times degli stati di salute* [4] come

$$K_h = \min\{k \geq 0 : H_k = h\}, h \in H \quad (8)$$

con K_h variabile aleatoria. Inoltre, data una policy μ e uno stato iniziale $\mathbf{Z}_0 = (Q_0, H_{max}, S_{-1})$, è possibile definire

- la *ricompensa media totale*, cioè la ricompensa complessiva media ottenuta nello stato di salute h :

$$G_\mu^{tot}(h, \mathbf{Z}_0) = \mathbb{E} \left[\sum_{k=K_h}^{K_{h-1}-1} g(A_k, Q_k) \middle| \mathbf{Z}_0 \right] \quad (9)$$

- la *durata della batteria*, cioè il numero di slot temporali che ci si aspetta la batteria spenda nello slot h :

$$T_\mu(h, \mathbf{Z}_0) = \mathbb{E}[K_{h-1} - K_h | \mathbf{Z}_0] \quad (10)$$

- la *ricompensa media per slot temporale*, cioè la ricompensa complessiva media ottenuta nello stato di salute h in un singolo slot di tempo :

$$G_\mu(h, \mathbf{Z}_0) = \frac{G_\mu^{tot}(h, \mathbf{Z}_0)}{T_\mu(h, \mathbf{Z}_0)} \quad (11)$$

Per quanto concerne la scelta di $g^*(A_k)$, si può pensare a una funzione concava crescente di A_k [15] e, se $g^*(A_k) = \log_2(1 + \sigma A_k / \bar{b})$, allora $g^*(A_k)$ è la capacità del canale gaussiano secondo Shannon, con σ un parametro di scala dell'SNR [3] e $G_\mu(h, \mathbf{Z}_0)$ è il throughput che ci si attende nello stato h . Inoltre, se avviene che $g^*(A_k) = \chi(A_k \geq \bar{A})$ per un certo \bar{A} , dove $\chi(\cdot)$ è la funzione indicatrice, $G_\mu(h, \mathbf{Z}_0)$ corrisponde alla probabilità che il carico \bar{A} sia gestibile nello stato h . Si intenda quindi G^* come un minimo valore di *Quality of Service* (QoS); se $G_\mu(H_{max}, \mathbf{Z}_0) \geq G^*$, la *durata di vita di una batteria* governata da una policy μ è

$$T_\mu(G^*, \mathbf{Z}_0) = \sum_{h \geq h_\mu^*} T_\mu(h, \mathbf{Z}_0) \quad (12)$$

con

$$h_\mu^* = \max\{h : G_\mu(h, \mathbf{Z}_0) < G^*\} + 1 \quad (13)$$

indice del più basso stato di salute h in cui è possibile garantire il QoS; se per $h = H_{max}$ questo non può esser assicurato, $T_\mu(G^*, \mathbf{Z}_0) = 0$. Il fatto che $G_\mu(H_{max}, \mathbf{Z}_0) \geq G^*$ garantisce di raggiungere il requisito di QoS perlomeno nello stato di salute più in alto e si noti che tale requisito è per ogni stato $h \geq h_\mu^*$, pertanto è più stretto di un generico requisito sul QoS *medio* sulla vita della batteria. Il problema di ottimizzazione consiste perciò nello sviluppare una policy μ^* che vada a massimizzare la durata di vita della batteria, non svincolandosi dalla richiesta di un QoS garantito G^* :

$$\mu^* = \arg \max_{\mu} T_\mu(G^*, \mathbf{Z}_0) \quad (14)$$

2.3 Ottimizzazione

Viene ora sviluppato il problema (14). Tre definizioni per iniziare:

- si assume che l'EHD lavori indefinitamente in uno stato $h \in H$, senza venir assorbito dallo stato inferiore. Allora la *distribuzione in stato stazionario* di $(q, s) \in Q(h) \times S$, nello stato h e con policy μ_h è

$$\pi_{\mu_h}^h(q, s) = \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=0}^{K-1} P^{(k)}(q, s | \mathbf{Z}_0) \quad (15)$$

con \mathbf{Z}_0 stato iniziale e $P^{(k)}(q, s | \mathbf{Z}_0) = \Pr(Q_k = q, S_{k-1} = s | \mathbf{Z}_0, p_H(h; \cdot) = 0)$.

- denotando la ricompensa stimata nello stato (q, s) come $\mathbb{E}_{\mu_h(\cdot; q, s)}[g(A, q)] = \sum_{a \in A} \mu_h(a; q, s)g(a, q)$, allora la *ricompensa stimata per stato* è:

$$\hat{G}_{\mu_h}(h) = \sum_{(q, s) \in Q(h) \times S} \pi_{\mu_h}^h(q, s) \mathbb{E}_{\mu_h(\cdot; q, s)}[g(A, q)] \quad (16)$$

- la *durata dello stato di salute* h :

$$\hat{T}_{\mu_h}(h) = \frac{1}{\sum_{(q, s) \in Q(h) \times S} \pi_{\mu_h}^h(q, s) p_H(h; q)} \quad (17)$$

La (16) può essere pensata come la ricompensa media a lungo termine per slot temporale nello stato di salute h , la (17) come l'inverso della probabilità media a lungo termine di transizione verso lo stato inferiore $h - 1$; una formalizzazione è la seguente:

Lemma 1 [12] *Sia $p_H^*(h) = \max_q p_H(h; q)$. Se $p_H^*(h) \rightarrow 0$,*

$$G_\mu(h, \mathbf{Z}_0) = \hat{G}_{\mu_h}(h) + \mathcal{O}(p_H^*(h)) \quad (18)$$

e

$$T_\mu(h, \mathbf{Z}_0) = \hat{T}_{\mu_h}(h) + \mathcal{O}(1) \quad (19)$$

L'espressione (19) è un'ottima approssimazione, in quanto se $\max_q p_H(h; q) \rightarrow 0$, $T_\mu(h, \mathbf{Z}_0) \rightarrow +\infty$. Lo stesso vale per la (18), con il fattore additivo $\mathcal{O}(p_H^*(h))$ che va a zero con velocità non inferiore di $\max_q p_H(h; q)$. Possiamo allora manipolare la (12) e ottenere

$$T_\mu(G^*, \mathbf{Z}_0) \simeq \sum_{h \geq h_\mu^*} \hat{T}_{\mu_h}(h) \quad (20)$$

con

$$h_\mu^* = \max\{h : \hat{G}_{\mu_h}(h) < G^*\} + 1 \quad (21)$$

Importante sottolineare l'indipendenza di $\hat{T}_{\mu_h}(h)$ e $\hat{G}_{\mu_h}(h)$ dallo stato \mathbf{Z}_0 , in quanto vi è dipendenza solo dalla distribuzione (15). Quindi, sostituendo nella (14) la (20), otteniamo

$$\mu^* = \arg \max_{\mu} \sum_{h \geq h_{\mu}^*} \hat{T}_{\mu_h}(h). \quad (22)$$

2.3.1 Risoluzione algoritmica

Viene proposto ora un possibile algoritmo per trovare la policy ottima μ^* [12].

```

h ← Hmax
flag ← true
while flag=true and h>0 do
   $\mu_h^* \leftarrow \arg \max_{\mu_h} \hat{T}_{\mu_h}(h)$ 
  if  $\sum_{(q,s) \in Q(h) \times S} \pi_{\mu_h}^h(q,s)(\mathbb{E}_{\mu_h(\cdot; q,s)}[g(A,q)] - G^*)$  then
    if h = 1 then
      flag ← false
       $h_{\mu^*}^* \leftarrow 1$ 
    else
      h ← h - 1
    end if
  else
    flag ← false
     $h_{\mu^*}^* \leftarrow h + 1$ 
  end if
end while
 $\mu^* \leftarrow (\mu_h^*)_{h \geq h_{\mu^*}^*}$ 

return  $\mu^*$ 

```

dove μ^* è la policy ottima, essendo $T_{\mu^*}(G^*, \mathbf{Z}_0) \simeq \sum_{h \geq h_{\mu^*}^*} \hat{T}_{\mu_h^*}(h)$. La policy μ_h^* porta alla massimizzazione della durata dello stato h , al contempo garantendo un minimo requisito di QoS. L'algoritmo determina anche $h_{\mu^*}^*$ per la policy ottima e per ottenere quest'ultima combina le policy $(\mu_h^*)_{h \geq h_{\mu^*}^*}$.

2.4 Estrapolazione delle probabilità di deperimento dai dati

Siano $p_H(h; q)$ le probabilità di deperimento della batteria, ottenibili sfruttando il modello (2) e i dati forniti dal produttore. La dipendenza dalla salute h non è facilmente calcolabile, ma può essere trascurata in quanto non molto forte. Possiamo allora chiamare queste probabilità $p_H(q)$, così da evidenziare unicamente la dipendenza dalla carica q .

2.4.1 Modello deterministico

Per la costruzione del modello deterministico, sfruttiamo il modello (2), considerando di contare $N_{cyc}(D)$ finché la capacità della batteria non scende a un valore determinato. Assumiamo che il processo di deperimento sia descritto dalla *funzione del tasso di deperimento di capacità* $\rho(q\Delta c/C_0)$, al livello di carica $q\Delta c \in [0, C_0]$, con Δc quanto di carica e q il livello di carica normalizzato al quanto Δc . Se allora la batteria lavora per δ secondi al livello di carica $q\Delta c$, la capacità diminuisce di $\delta\rho(q\Delta c/C_0)$ mAh. Ammettiamo quindi che, per $\theta, \zeta > 0$, valga

$$\rho(q\Delta c/C_0) = \zeta e^{\theta(1-q\Delta c/C_0)} \quad (23)$$

Mostreremo la bontà di questa ipotesi in relazione al modello (2). Sia C_n la capacità della batteria all'inizio del ciclo di scarica/carica n -esimo. Se la DoD assume valore D , nell' n -esimo ciclo la batteria si scarica da C_n a $C_n - C_0 D$ per ricaricarsi poi a $C_{n+1} \leq C_n$. Ponendo che il deperimento della batteria nell' n -esimo ciclo sia denotata da $\Delta_\rho(D, C_n) = C_n - C_{n+1}$, se la corrente nei cicli di scarica/carica è I e se $\Delta_\rho(D, C_n) \ll 2DC_0$ (il processo di deperimento è di molto inferiore alla quantità di carica scambiata in ogni ciclo), la durata dell' n -esimo ciclo di scarica/carica è

$$T_n = \frac{2DC_0 - \Delta_\rho(D, C_n)}{I} \simeq \frac{2DC_0}{I} \quad (24)$$

e il livello di carica $Q_n(t)\Delta c, t \in (0, T_n)$ ha la seguente evoluzione

$$\text{Fase di scarica: } Q_n(t)\Delta c = C_n - It, t \in (0, \frac{T_n}{2}) \quad (25)$$

$$\text{Fase di carica: } Q_n(t)\Delta c = C_n - DC_0 + I(t - \frac{T_n}{2}), t \in (\frac{T_n}{2}, T_n) \quad (26)$$

e la capacità istantanea della batteria nell' n -esimo ciclo, $C_n(t)$, segue

$$\frac{dC_n(t)}{dt} = -\rho\left(\frac{Q_n(t)\Delta c}{C_0}\right), t \in (0, T_n) \quad (27)$$

con le condizioni $C_n(0) = C_n, C_n(T_n) = C_{n+1}$. Integrando la carica scambiata in un ciclo si ha

$$C_{n+1} = C_n + \int_0^{T_n/2} C'_n(\tau) d\tau + \int_{T_n/2}^{T_n} C'_n(\tau) d\tau \quad (28)$$

sostituendo in quest'espressione la (27), usando per ρ la (23) e per $Q_n(t)$ la (25) e la (26), otteniamo

$$\Delta_\rho(D, C_n) = \frac{2C_0\zeta}{I\theta} e^{\theta(1-C_n/C_0)} (e^{\theta D} - 1) \quad (29)$$

Poiché $N_{cyc}(D) = \min\{n : C_n < xC_0\}$ e poiché il numero di cicli che portano ad avere una piccola diminuzione della capacità $dC \ll C_0$ da $C \in (0, C_0]$ a $C - dC$

sono $dC/\Delta_\rho(D, C)$, possiamo fare la seguente approssimazione

$$N_{cyc}(D) \simeq \int_{x_{C_0}}^{C_0} \frac{1}{\Delta_\rho(D, C)} dC \quad (30)$$

sostituendo nella (30) la (29), si ha

$$N_{cyc}(D) \simeq \int_{x_{C_0}}^{C_0} \frac{I\theta e^{\theta(C/C_0-1)}}{2C_0\zeta(e^{\theta D}-1)} dC = \frac{I\theta \cdot C_0}{2C_0\zeta \cdot \theta} \cdot e^{\theta C/C_0} \Big|_{x_{C_0}}^{C_0} \text{ e perciò}$$

$$N_{cyc}(D) \simeq \left[\frac{I}{2\zeta} \frac{1 - e^{-\theta(1-x)}}{1 - e^{-\theta D}} \right] e^{-\theta D} \quad (31)$$

con la quantità tra parentesi quadre funzione decrescente di D . Allora

$$\hat{N}_{cyc}(D) \triangleq \frac{I}{2\zeta} \frac{1 - e^{-\theta(1-x)}}{1 - e^{-\theta}} e^{-\theta D} \leq N_{cyc}(D) \quad (32)$$

osservando infine il modello (2) e supponendo $N_{cyc}(D) \simeq \hat{N}_{cyc}(D)$ si ottiene

$$\alpha = \theta \text{ e } \zeta = \frac{I}{2N_{cyc,0}} \frac{1 - e^{\alpha(1-x)}}{e^\alpha - 1} \quad (33)$$

Si osservi che il modello (31) non segue perfettamente la (2): se $D \rightarrow 0$ si ha $N_{cyc}(D) \rightarrow \infty$ e ciò avviene in quanto abbiamo assunto che la DoD D sia più importante in relazione al deperimento della batteria in un singolo ciclo.

2.4.2 Modello stocastico

Per trovare le probabilità di deperimento $p_H(q)$ calcoliamo prima il tempo che impiega la batteria a passare dallo stato h (con capacità $\frac{h}{H_{max}}C_0$) allo stato $h-1$ (con capacità $\frac{h-1}{H_{max}}C_0$), per poi porre a confronti i tempi di deperimento deterministici con quelli medi. Sia $q\Delta c$ il livello di carica a cui lavora la batteria che segue il modello deterministico in precedenza analizzato. La capacità iniziale è $C(0) = \frac{h}{H_{max}}q_{max}\Delta c$ e, se la capacità in funzione del tempo è descritta dalla relazione $C(t) = C(0) - \rho(q\Delta c/C_0)t$, essa passa allo stato di salute inferiore con capacità $\frac{h-1}{H_{max}}q_{max}\Delta c$ in un tempo

$$T_{det}(q) = \frac{q_{max}\Delta c}{H_{max}\rho(q\Delta c/C_0)}. \quad (34)$$

Invece, nel modello stocastico, indicando con Δt la durata di un slot, il tempo *medio* che impiega la batteria a decadere in uno stato di salute inferiore è

$$T_{sto}(q) = \frac{\Delta t}{p_H(q)}. \quad (35)$$

Eguagliando la (34) e la (35) si scopre

$$p_H(q) = \gamma \exp \left\{ \alpha \left(1 - \frac{q}{q_{max}} \right) \right\} \quad (36)$$

dove $\gamma = \frac{H_{max}}{q_{max}} \frac{\zeta \Delta t}{\Delta c}$. La (36) è in accordo sia con a) (se $\gamma \neq 0$) che con c). Se poi $\gamma \ll 1$, nemmeno b) è contrastata. Per concludere, è opportuno notare come il valore assoluto di γ non inficia la soluzione del problema di ottimizzazione visto in 2.3.1, che diventa

$$\mu_h^* = \arg \min_{\mu_h} \sum_{(q,s) \in Q(h) \times S} \pi_{\mu_h}^h(q,s) \exp \left\{ \alpha \left(1 - \frac{q}{q_{max}} \right) \right\} \quad (37)$$

tale che

$$\sum_{(q,s) \in Q(h) \times S} \pi_{\mu_h}^h(q,s) (\mathbb{E}_{\mu_h(\cdot; q,s)}[g(A,q)] - G^*) \geq 0. \quad (38)$$

2.5 Risultati

Vengono presentati ora alcuni risultati numerici che sanciscono la bontà della scelta del modello stocastico elaborato. Consideriamo una batteria con $q_{max} = 500$ livelli di carica e $H_{max} = 50$ stati di salute. Interpolando i valori dei data-sheet delle due batterie prese in esame (MS920SE [1], MEC201-10P [19]) e usando il modello (2), si ottiene $N_{cyc} = 100$, $\alpha \simeq 2.88$ per il dispositivo MS920SE e $N_{cyc} = 5000$, $\alpha \simeq 3.33$ per il dispositivo MEC201-10P. Si calcolano quindi le $p_H(q)$, scegliendo per γ un valore sufficientemente piccolo per i motivi visti in precedenza.

Le curve in Fig. 2 dimostrano la coerenza tra modello stocastico e modello esponenziale (2) e sono state ottenute scaricando e ricaricando ciclicamente le batterie con diversi valori di DoD D . Si nota la poca differenza tra modello deterministico e stocastico, a rimarcare che il primo può ben descrivere il deperimento di una batteria reale nel corso del tempo. Anche il modello stocastico non si discosta notevolmente dalla curva sperimentale.

Consideriamo ora il processo degli scenari $\{S_k\}$, modellato da una catena di Markov a due stati con spazio $\mathcal{S} = \{C, B\}$ e probabilità di transizione $p_S(B|B) = p_S(C|C) = 0.96$, con C e B ad indicare lo scenario "cattivo" e quello "buono" [9]. Nello scenario $S_k = C$ non viene raccolta energia ($B_k = 0$), in $S_k = B$ si ha deterministicamente $B_k = 20$. Il *tasso medio di harvesting* è allora $\hat{b} = 10$. Utilizziamo come funzione delle ricompense (7) la funzione $g^*(A_k) = \log_2(1 + \sigma A_k / \bar{b})$ con $\sigma = 10$; sia lo spazio delle azioni $\mathcal{A} \triangleq \{0\} \cup \{A_{min}, \dots, A_{max}\}$ con $A_{min} = 10$ e $A_{max} = 20$. Prendiamo ora in esame le seguenti policy [12]:

- *Constant Load Lifetime Unaware Policy* (CLLUP), che sostiene un carico costante di $A_k = \bar{A}$ quanti di carica, senza tenere in considerazione del livello di carica della batteria, e rimane inattivo in caso di mancanza di energia. Tale policy richiede una minima comunicazione tra il controller EHD e la Power Processing Unit (non è richiesta la conoscenza di Q_k).
- *Lifetime Unaware Policy* (LUP), che massimizza la ricompensa media a lungo termine per il valore effettivo della capacità della batteria, non valutandone l'intaccamento futuro; a differenza della precedente, richiede l'esatta conoscenza di Q_k .

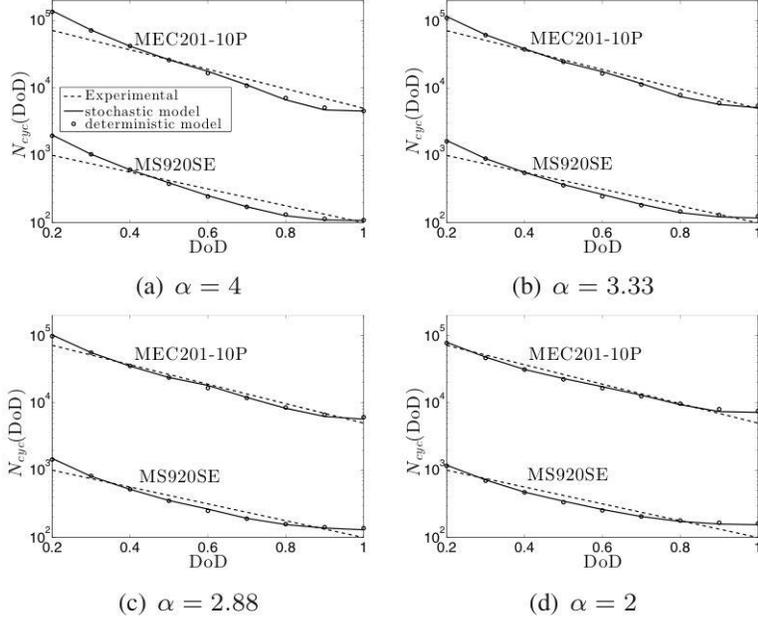


Figura 2: N_{cyc} in relazione a DoD . La capacità della batteria MS920SE viene portata fino al 50% del suo valore iniziale; quella della batteria MEC201-10P fino all'80% del valore iniziale [12]

- *Lifetime Aware Optimal Policy* (LAOP), che è la policy ottima trovata come soluzione di (14) con l'algoritmo proposto in 2.3.1.
- *Constant Load Lifetime Aware Policy* (CLLAP), che sostiene un carico costante di $A_k = \bar{A}$ quanti di carica se il livello di carica è sopra una certa DoD e rimane inattivo altrimenti. Se poi la capacità diminuisce fino a un valore che non riesce a soddisfare la DoD richiesta, si dichiara un errore.

Sia LAOP che CLLAP sono policy che tengono in considerazione la degradazione della capacità della batteria; inoltre per CLLUP e CLLAP scegliamo $\bar{A} = \bar{b}$. Data una policy e un QoS G^* la durata di vita della batteria è calcolata come in (12); la *ricompensa minima* assicurata dalla policy μ è definita come $G_{min}(\mu, G^*) = \min_{h \geq h_\mu^*} G_\mu(h, \mathbf{Z}_0)$, riprendendo le definizioni viste in (13) e (11). Questa indica la ricompensa media per slot data per certa nell'arco di vita della batteria. I seguenti grafici (Fig. 3 e Fig. 4) mostrano i risultati di una simulazione.

Si evidenzia con facilità la similarità tra le curve associate al modello deterministico e a quello stocastico: una policy come LAOP, che guarda a un modello di deperimento stocastico, riesce a descrivere bene il processo anche se quest'ultimo è deterministico. Come si vede osservando la Fig. 3, LUP fa meglio di

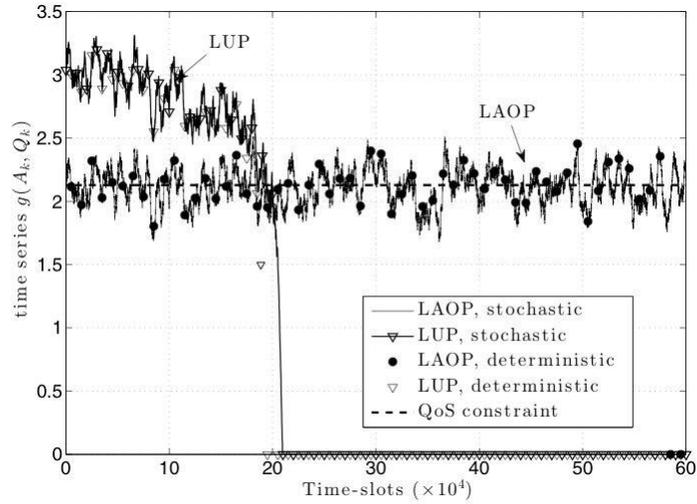


Figura 3: Serie delle ricompense per il dispositivo MS920SE, confronto tra modello deterministico e stocastico (QoS $G^* = 2.13$, il 70% della ricompensa massima pari a ≈ 3.02) [12]

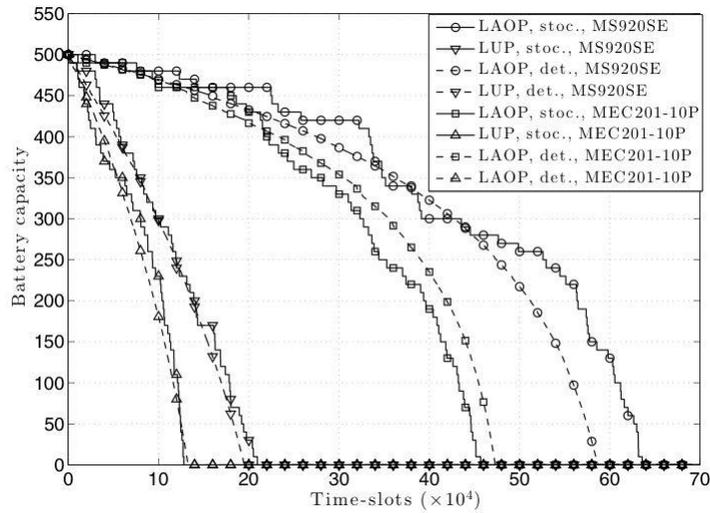


Figura 4: Diminuzione della capacità nel tempo, confronto tra modello deterministico e stocastico (QoS $G^* = 2.13$) [12]

LAOP solo in $[0, 18 \times 10^4]$, sfruttando tutti i livelli di carica a sua disposizione per ottenere la massima ricompensa e così deteriorando significativamente in tempi più brevi la capacità della batteria. LAOP riesce invece a prolungare la

vita della batteria, gestendo il processo più intelligentemente. Potrebbe anche essere reso più forte il requisito sul QoS, così da contrastare con un margine le fluttuazioni che spesso portano la ricompensa $g(A_k, Q_k)$ al di sotto di G^* . Nella Fig. 5 viene graficata la ricompensa minima $G_{min}(\mu, G^*)$ in funzione della durata di vita della batteria normalizzata alla sua durata massima, che sarebbe la durata di vita della batteria quando questa è completamente carica. Di nuovo, LAOP si dimostra capace di garantire una vita maggiore per la batteria rispetto a LUP, indifferente alla degradazione della capacità. Le medesime considerazioni possono essere esposte nel confronto CLLAP/CLLUP e inoltre CLLAP, pur garantendo ricompense inferiori a LAP, prolunga ancor di più la vita del dispositivo impedendo cicli di scarica importanti e, in generale, non permettendo scariche oltre un certo valore di DoD. Si può quindi affermare che più lunga è la vita della batteria, inferiore sarà la ricompensa minima ottenuta; infatti, mentre la prima è massimizzata con cicli di scarica/carica non eccessivamente profondi, la seconda è esaltata da cicli di scarica non banali, che portano ad accorciare la vita del dispositivo.

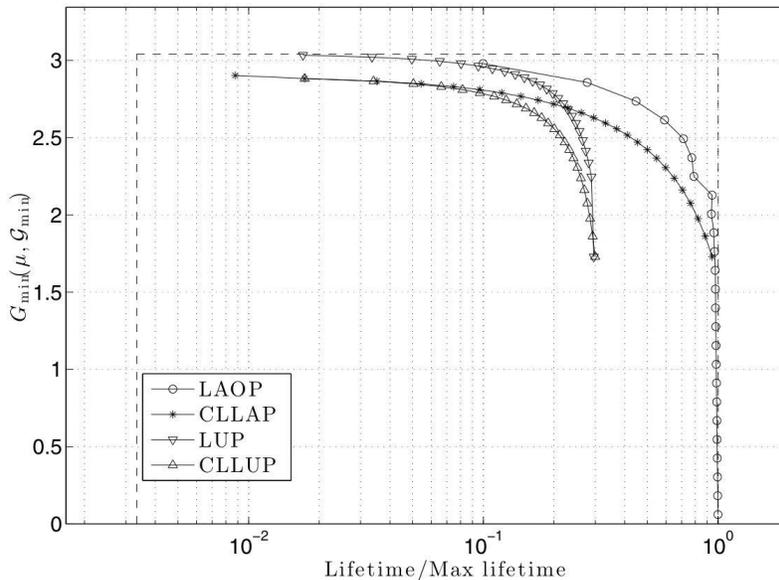


Figura 5: Ricompensa minima nel tempo in relazione alla durata di vita della batteria normalizzata per il dispositivo MS920SE; le linee tratteggiate indicano la durata di vita minima e massima e la ricompensa massima [12]

Nella Fig. 6 viene graficata la durata di ogni stato di salute $h \in H$. Si capisce subito che l'espressione della durata di vita approssimata non presenta un errore apprezzabile nei confronti di quella non approssimata. Inoltre, LAOP massimizza questa durata per ogni stato di salute, in quanto tale policy viene ricavata attraverso l'algoritmo presentato precedentemente, che minimizza per ogni stato

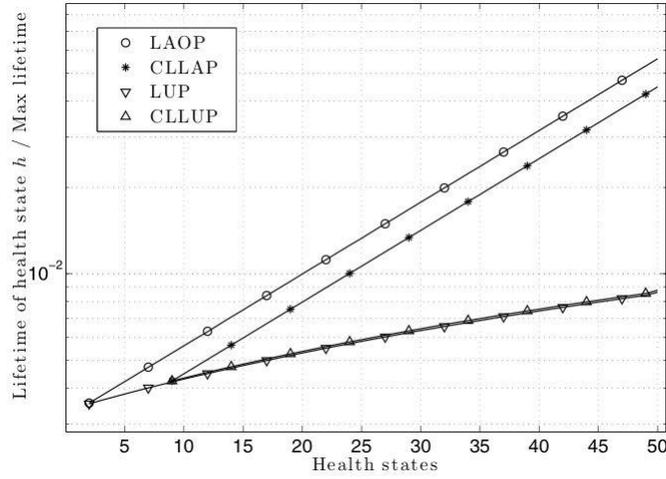


Figura 6: Durata di vita normalizzata per ogni stato di salute. Le linee rappresentano la durata esatta(12), i simboli quella approssimata(17) [12]

$h \in H$, le probabilità di deperimento della batteria che deve rispettare un QoS minimo. Ancora, LUP determina per ogni stato h una vita decisamente più breve, in accordo alla sua "avidità" nei confronti della ricompensa. Lo stesso si può dire per CLLAP e CLLUP; conformemente all'Ass. c), più la batteria è degradata, più velocemente avanza il processo di degradazione.

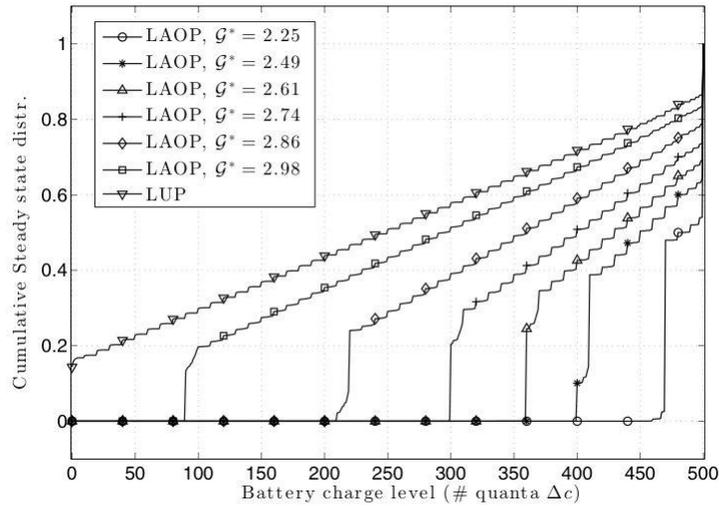


Figura 7: Distribuzione di stato stazionario cumulativa dei livelli di carica al livello di salute più alto H_{max} per il dispositivo MS920SE [12]

Infine, nella Fig. 7 viene graficata la distribuzione di stato stazionario cumulativa dei livelli di carica, per lo stato di salute H_{max} , per le policy LUP e LAOP e molteplici valori di QoS. La distribuzione di stato stazionario per LUP è caratteristica di tutti i livelli di carica, in quanto tale policy non tiene conto dei meccanismi di degradazione della batteria, al contrario di LAOP, per la quale si ha distribuzione di stato stazionario solo nei livelli di carica più alti, così da rallentare i suddetti meccanismi. Oltre a ciò, più è forte una richiesta sul QoS, più la distribuzione di stato stazionario si diffonde ai livelli inferiori, dovendo la batteria garantire prestazioni migliori e perciò subire cicli di scarica più importanti.

2.6 Estensione a batterie reali

Si cercherà ora di estendere il modello (3) a una batteria reale. Un modello più vicino alla realtà potrebbe essere [12]

$$Q_{k+1} = \min\{[Q_k - A_k - L_k]^+ + B_k, Q_{max}(k+1)\} \quad (39)$$

dove con L_k è il costo complessivo dell'energia nello slot k , ad eccezione del controllo A_k , quindi il costo di eventuali perdite, calcolo e attivazione dell'hardware quando il nodo ritorna inattivo. Sia L_k una variabile aleatoria con distribuzione $p_L(L_k|Q_k, A_k, I_k)$ a valori in $\mathbf{L} = (0, 1, \dots, L_{max})$, quindi dipendente dal livello di carica Q_k , dall'azione A_k e dallo stato di inattività I_k ; quest'ultimo, definito come $I_k = \chi(A_{k-1} = 0)$, indica se il nodo era attivo ($I_k = 0, A_{k-1} > 0$) o meno ($I_k = 1, A_{k-1} = 0$) nello slot precedente. Inoltre, il processo di harvesting $\{B_k\}$ può esser generalizzato così da includere il *modello di Markov generalizzato* di ordine $L = 1$, si veda [9]. Come in (3), $\{S_k\}$ è una catena di Markov con probabilità di transizione $p_S(s_{k+1}|s_k)$. Dato $S_k \in S$, B_k viene scelto da \mathbf{B} in accordo alla distribuzione $p_B(b_k|s_k, b_{k-1}) \triangleq \Pr(B_k = b_k|S_k = s_k, B_{k-1} = b_{k-1})$. In questo modello, lo stato nello slot k è $\mathbf{Z}_k = (Q_k, H_k, S_{k-1}, B_{k-1}, I_k)$, a valori in $Q \times H \times S \times B \times \{0, 1\}$, e la policy μ_{H_k} è una misura di probabilità sullo spazio delle azioni \mathbf{A} , parametrizzata da $(Q_k, S_{k-1}, B_{k-1}, I_k)$; ad esempio la probabilità di richiedere a quanti di carica alla batteria è $\mu_{H_k}(a, Q_k, S_{k-1}, B_{k-1}, I_k)$. La ricompensa istantanea ottenuta nello slot k , nello stato \mathbf{Z}_k con azione A_k , diventa

$$g(A_k, Q_k, L_k) = \begin{cases} 0, & A_k > Q_k - L_k, \\ g^*(A_k), & A_k \leq Q_k - L_k, \end{cases} \quad (40)$$

La policy μ_{H_k} può esser ottimizzata con l'algoritmo della sezione 2.3.1, cosicchè si ottiene

$$\mu_h^* = \arg \min_{\mu_h} \sum_{(q,s,b,i) \in Q(h) \times S \times B \times \{0,1\}} \pi_{\mu_h}^h(q, s, b, i) p_H(h; q) \quad (41)$$

tale che

$$\sum_{(q,s,b,i) \in Q(h) \times S \times B \times \{0,1\}} \pi_{\mu_h}^h(q, s, b, i) (\mathbb{E}_{\mu_h(\cdot; q, s, b, i)}[g(A, q, L)] - G^*) \geq 0. \quad (42)$$

3 Soluzione power-saving per WSN con energy harvesting

3.1 Nozioni fondamentali

Per sopperire al secondo problema presentato nell'introduzione vi sono molteplici approcci. Quello più pratico è l'inserimento di un modulo di gestione dell'energia (PMM) tra WSN ed energy harvester; questo può, a titolo di esempio, comandare degli *switch* che, se opportunamente attivati, possono dar vita a un oscillatore *LC* temporaneo grazie agli energy harvesters e a componenti quali induttori [8],[5] o condensatori [10], [25]. L'aumento di potenza sarebbe causato dall'aumento delle forme d'onda della tensione dall'energy harvester. Un'altra via praticabile è l'adattamento del carico per le impedenze interessate, il quale permette a sua volta di ottenere il massimo trasferimento di potenza. Tuttavia, il WSN, dovendo svolgere molteplici azioni, può esser considerato come un carico dinamico [6] e quindi una connessione diretta WSN-PMM ridurrebbe estremamente la quantità di energia coinvolta nel processo di harvesting [22]. Si sceglie quindi di usare uno *low sleep power consumption wireless microcontroller* (MCU); spesso però accade che, pur garantendo una buona efficienza, gli MCU vengono collegati a memorie particolarmente capienti e a numerosi dispositivi, fatto che porta il consumo energetico dei primi ad essere non trascurabile durante le fasi di inattività. Tale condizione potrebbe portare a problemi di attivazione per WSN che sfruttano l'EH nei momenti in cui non è possibile raccogliere energia sufficiente, in quanto l'MCU non si attiverebbe e non potrebbe controllare i circuiti cui è collegato e, fatto ancor più grave, gli stessi circuiti continuerebbero a sottrarre energia dal condensatore che la immagazzina (si veda la Fig. 8), non permettendo all'MCU di attivarsi in futuro.

Per risolvere tale difficoltà sono nate delle interfacce *energy-aware* (EAI), comunicanti con PMM e WSN. In [24] e in [7] viene esposto l'uso di un EAI per accendere e spegnere un nodo in base a delle soglie di tensione scelte in precedenza. Questo metodo però non è efficiente, poiché costringerebbe il WSN a stare acceso se non è ancora stata raggiunta la soglia di spegnimento, anche se questo ha già portato a termine tutte le operazioni, oppure potrebbe causarne lo spegnimento mentre esso è in piena attività. Verrà quindi proposto un programma *energy-aware*, capace di decidere se l'energia stoccata nel condensatore è sufficiente o meno al WSN per svolgere la seguente operazione e trasmettere i dati associati. Insieme all'EAI, esso verrà adottato su un sistema basato su energy harvesting e verrà mostrata sperimentalmente la superiore sostenibilità di un simile sistema combinato in confronto ai precedenti.

3.2 Modello

La Fig. 8 rappresenta un diagramma a blocchi del WSN alimentato da EH che viene studiato in questo capitolo; si noti che, a differenza del modello analizzato nel precedente capitolo, l'energia viene immagazzinata in un condensatore e non in una batteria ricaricabile. Il rispettivo circuito stampato (PCB) è costituito da un harvester piezoelettrico che sfrutta un composito in macro-fibra (MFC) per ottenere energia vibrazionale dall'ambiente e caricare il condensatore C_s , dal quale viene alimentato il nodo sensore.

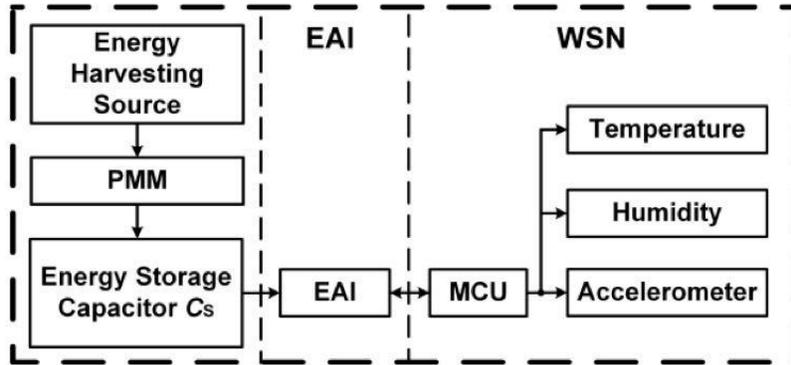
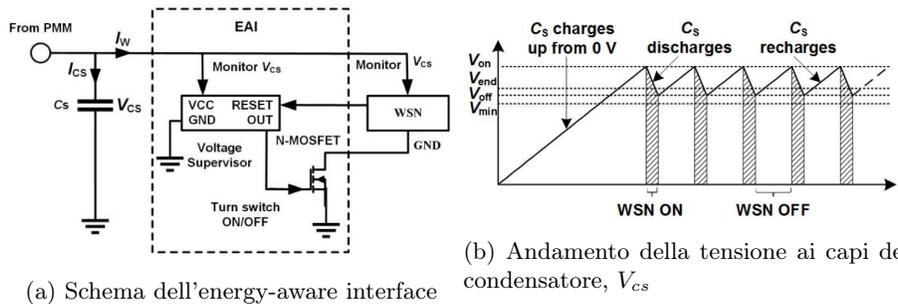


Figura 8: Diagramma a blocchi di un nodo sensore basato su EH con EAI [16]

3.2.1 Circuito EAI

Lo schema del circuito EAI, composto da un sensore di tensione per monitorare il potenziale ai capi del condensatore V_{Cs} e un transistor N-MOS per accendere e spegnere il WSN, si trova in Fig. 9a



(b) Andamento della tensione ai capi del condensatore, V_{Cs}

Figura 9: [16]

In Fig. 9b viene graficato l'andamento di V_{Cs} dovuto all'EAI; i valori di tensione notevoli sono:

- V_{on} , la soglia prefissata imposta dal sensore di tensione alla quale il WSN si accende e comincia la sua attività.
- V_{end} , la soglia calcolata dall'MCU sotto la quale il WSN non potrà ottenere dal condensatore energia sufficiente a compiere il prossimo task.
- V_{off} , il valore di tensione per cui il WSN viene spento in seguito alla trasmissione dei dati elaborati.
- V_{min} , la minima tensione a cui il nodo può operare, sotto la quale possono verificarsi malfunzionamenti.

Per avere un margine d'errore, si pone $V_{end} > V_{off}$ e $V_{off} > V_{min}$. All'inizio del processo, il condensatore C_s è scarico e viene caricato finché la tensione ai suoi capi non arriva a V_{on} . A questo punto, il pin di uscita dell'EAI viene impostato a uno stato "alto", così da accendere l'N-MOS e collegare il sistema a terra grazie a un cammino a bassa impedenza. In questo stato, il WSN può ricevere una corrente I_w da C_s , chiamando I_{cs} la corrente in input al condensatore. Durante l'attività il nodo monitora costantemente V_{cs} e decide se in C_s vi è abbastanza energia per portare a termine il prossimo task (cioè se $V_{cs} > V_{end}$) tramite il programma energy-aware. Se ciò non avviene, il nodo termina la sua attività e viene spento il transistor; avviene quindi la disconnessione dal condensatore e il WSN entra nella sua fase di inattività, permettendo che il primo venga ricaricato. Il sistema ritorna in attività solo quando V_{cs} arriva a V_{on} , e quanto appena detto si ripete.

3.2.2 Nodo sensore

Lo schema del WSN in esame è mostrato in Fig. 10; è costituito da un MCU (Jennic JN5148), 128 kB di ROM, 128 kB di RAM e tre sensori analogici, un accelerometro (ADXL335 a 3 assi), un sensore di temperatura (MCP9700) e uno di umidità (HIH-5030). I pin di uscita dei sensori sono collegati ai convertitori analogico digitale in ingresso all'MCU, ma, poiché questo dispone solo di 4 pin in input e i sensori garantiscono 5 letture (l'accelerometro dà una lettura per l'asse x, una per l'asse y e una per l'asse z), il sensore di umidità e quello di temperatura condividono il medesimo pin ADC4 e un transistor P-MOS viene acceso o spento dal controllore per evitare conflitti e corruzione di dati. Una precauzione in più viene presa per l'accelerometro, che offre una lettura di dati continua a causa della natura dei dati che deve trasmettere, i quali possono cambiare radicalmente in istanti temporali vicini: non essendo V_{cs} costante durante la fase attiva, il sistema viene corredato di un dispositivo (ISL21080CIH325) in grado di mantenere una tensione fissata (2.5 V nel nostro caso) indipendentemente dal carico.

3.2.3 Programma energy-aware

Il diagramma dell'operazione è sintetizzato in Fig. 11. Inizialmente il WSN è inattivo finché non vi è abbastanza energia nel condensatore per far funziona-

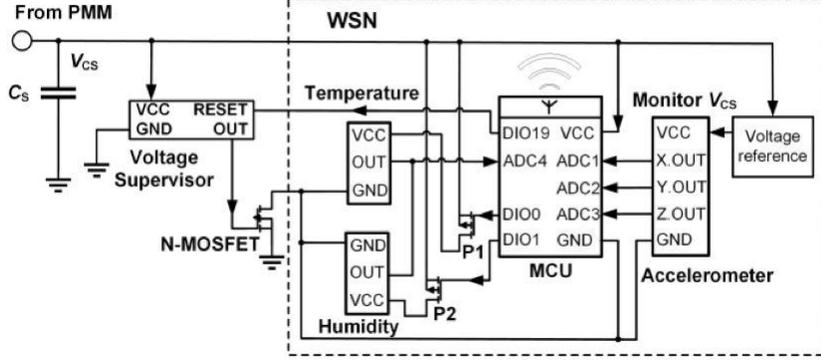


Figura 10: Schematico del nodo sensore dotato di EAI [16]

re l'MCU. Quando ciò avviene, l'N-MOS si accende e con esso gli interruttori nell'MCU collegati alla sensoristica. Viene rilevata una lettura dal sensore di umidità, quindi, non prima dello spegnimento di quest'ultimo, una lettura dal sensore di temperatura. In seguito, vengono prese le tre misure di posizione dall'accelerometro; l'MCU continua a ricevere misure da quest'ultimo per un numero di volte predeterminato (a titolo di esempio, decidiamo di ricevere 48 letture). Ogni lettura ha un peso di 2 byte e viene scritta nella RAM. Successivamente, viene misurato V_{cs} e, se $V_{cs} > V_{end}$, si continuano a leggere le misurazioni dell'accelerometro, altrimenti l'MCU trasmette tutti i dati conservati e invia un segnale al sensore di tensione affinché spenga il transistor N-MOS e il WSN torni ad essere inattivo. Il tutto si ripete ciclicamente. È importante sottolineare che in questo programma vi è una sola trasmissione dei dati alla fine della fase attiva, al fine di risparmiare energia il più possibile.

3.2.4 Calcolo di V_{end} e dell'energia richiesta

La V_{end} può essere calcolata dalla seguente espressione

$$\frac{1}{2}CV_{end}^2 - \frac{1}{2}CV_{min}^2 \geq E_{req} = E_{tx,2} + (N + 1) \cdot E_{tx,48} + E_{samp,48} + E_{reset} \quad (43)$$

dove E_{req} è l'energia minima di cui il WSN necessita per compiere i suoi task nel prossimo ciclo, $E_{tx,2}$ l'energia per trasmettere le misure di umidità e temperatura dalla RAM, $E_{tx,48}$ l'energia per trasmettere le 48 misure di accelerazione, sempre dalla RAM, $E_{samp,48}$ l'energia per campionare i 48 dati dell'accelerometro ed E_{reset} l'energia per inviare il segnale di *reset*. N è il numero di cicli delle attività del nodo. Inoltre, V_{min} è impostato al minimo valore che garantisce un funzionamento corretto dell'MCU, cioè 2.4 V. Un *source meter* viene usato per il calcolo di V_s e I_s , in input al WSN, e il consumo energetico dovuto allo

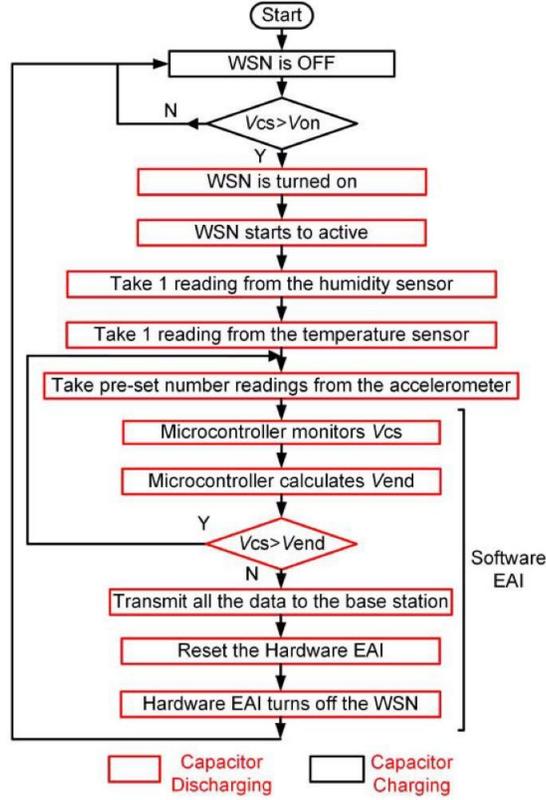


Figura 11: Schema del programma energy-aware [16]

svolgimento di un task porato a compimento in tempo t_n è

$$E_{task}(t_n) = \sum_{k=1}^n V_s(t_k) I_s(t_k) \Delta T \quad (44)$$

con $V_s(t_k)$ e $I_s(t_k)$ la tensione e la corrente istantanea del WSN al tempo t_k , Δt il passo di campionamento e n il numero di campioni in t_n ; si noti che con il pedice E_{task} si indica una singola operazione del tipo "tx,2", "tx,48", etc. La capacità C_s può allora esser determinata dalla (43) ricavando dalla (44) l'energia di cui il nodo ha bisogno per portare a termine le operazioni richieste.

3.3 Sperimentazione

3.3.1 Strumentazione

Sono stati eseguiti dei test sperimentali sul WSN basato su EH precedentemente descritto. Il diagramma in Fig. 12 mostra gli strumenti usati. Vengono usate

due interfacce LabVIEW, una per mostrare i dati trasmessi dai sensori alla stazione base posta a 4 m di distanza, l'altra per conoscere in tempo reale i valori di V_{cs} e I_w , misurati da due source meter (Keithley 2612B). Un harvester piezoelettrico che adopera un composito in macro-fibra è assicurato a un lato di una lastra di fibra di carbonio e recupera energia vibrazionale; dal lato opposto, il circuito stampato con WSN e PMM. Infine, un Instron E10000 ElectroPuls applica un carico vibrazionale sulla lastra.

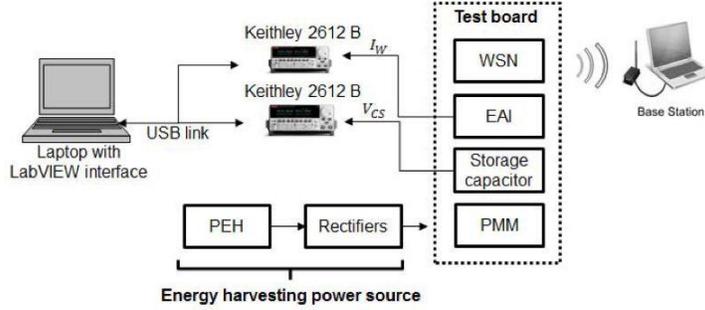


Figura 12: Diagramma a blocchi degli strumenti usati [16]

3.3.2 Configurazione

Per evidenziare il vantaggio di un approccio energy-aware, sono state predisposte 3 diverse configurazioni. La configurazione I è basata sullo schema in Fig.8, mentre la II è identica alla prima ma senza blocco EAI, così da capire l'importanza di quest'ultimo in presenza di poca energia ambientale. La configurazione III consiste in un WSN collegato direttamente a un alimentatore DC e privo di EAI, per evidenziare quanto l'EAI sia efficiente nella fase di inattività del nodo. Nella configurazione III, inoltre, l'alimentatore DC sostituisce l'MFC, l'EAI e il PMM; l'MCU viene inoltre programmato in maniera tradizionale per quanto concerne il periodo di inattività, sfruttando un timer regolato da un oscillatore resistivo-capacitivo. Quando avviene un evento di tipo "wake up", il timer imposta il periodo dopo cui il sistema entrerà nella fase di inattività, e, per rendere il confronto più corretto, le fasi di attività e inattività della configurazione III sono le stesse della configurazione I. La tensione dell'alimentatore DC è pari a 3.16 V, corrispondente a V_{on} della configurazione I, e la corrente in ingresso è di 8 mA, la quantità minima che permette il corretto funzionamento in tale setup. Definiamo la *corrente media di inattività*

$$I_{sleep} = \frac{\sum_{k=1}^n I_w(t_k) \Delta t}{T_{sleep}} \quad (45)$$

che corrisponde alla corrente media del WSN durante il periodo di inattività T_{sleep} , con $I_w(t_k)$ la corrente di inattività al tempo t_k , Δt il passo di campionamento e n il numero di campioni in T_{sleep} .

3.4 Risultati

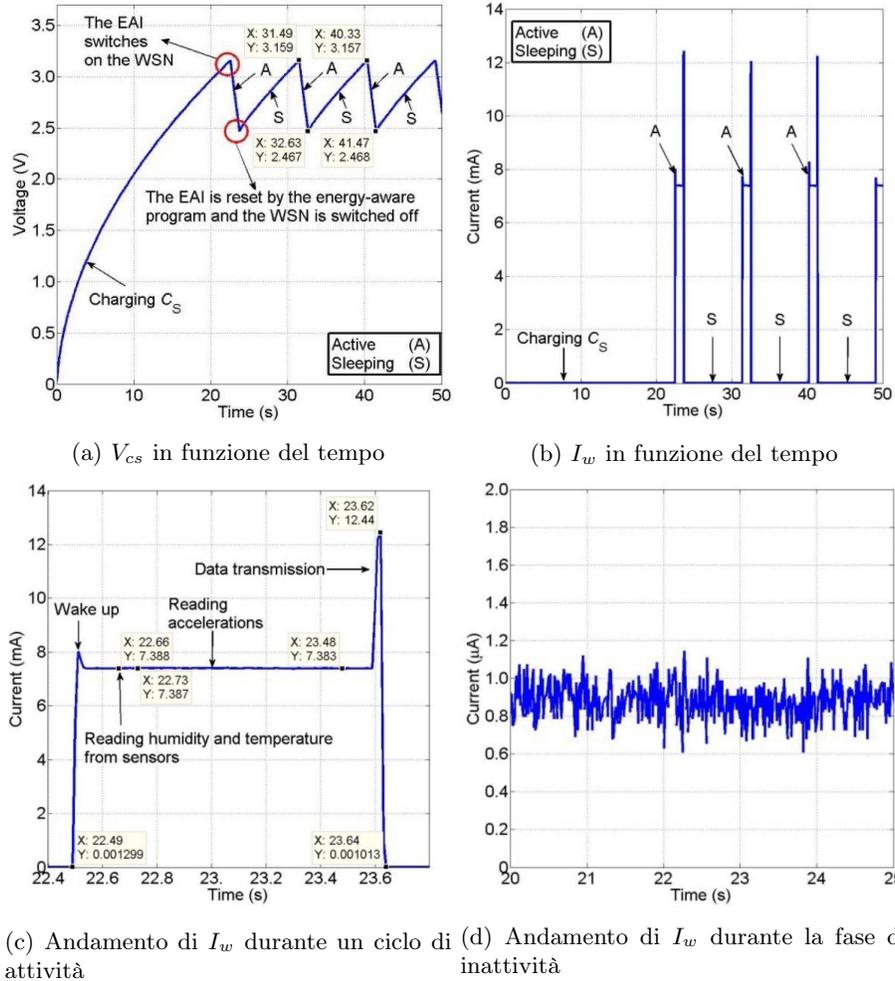


Figura 13: [16]

3.4.1 Configurazione I

Si osservi la Fig. 13. In questa configurazione, il condensatore viene portato alla tensione $V_{cs} = 3.16$ V dopo 22.56 s. In questo arco di tempo, $V_{cs} < V_{on}$, perciò il WSN è spento. Una volta che però le due tensioni si eguagliano, l'N-MOS viene acceso dall'energy-aware interface, cosicché il condensatore carica il nodo: V_{cs} inizia a calare e si ha un picco di I_w (Fig. 13a e 13b) e il WSN si accende per circa 1.15 s, eseguendo i task ad esso assegnati (Fig. 13c). Nel

mentre, l'energy-aware program calcola V_{end} ($\simeq 2.58$ V) e, appena V_{cs} scende al di sotto di questa soglia, l'EAI spegne il nodo, prima di arrivare a V_{min} ($\simeq 2.4$ V), così da lasciare al WSN margine per trasmettere i dati elaborati. Nella fase di inattività il WSN consuma una corrente media pari a $I_{sleep} = 0.95\mu\text{A}$, calcolata con la (45) (Fig. 13d): la ragione per cui il consumo energetico è così contenuto è la presenza dell'EAI, che disabilita il nodo, l'MCU e i circuiti ad essi collegati.

3.4.2 Configurazione II

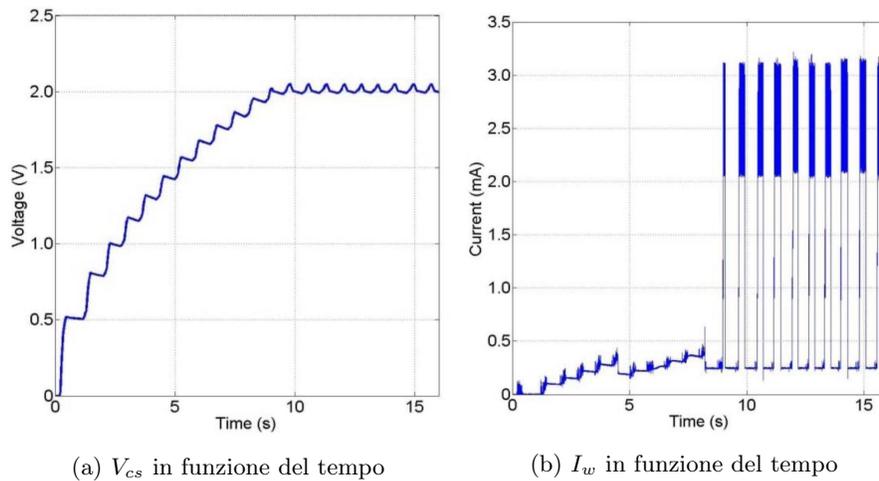
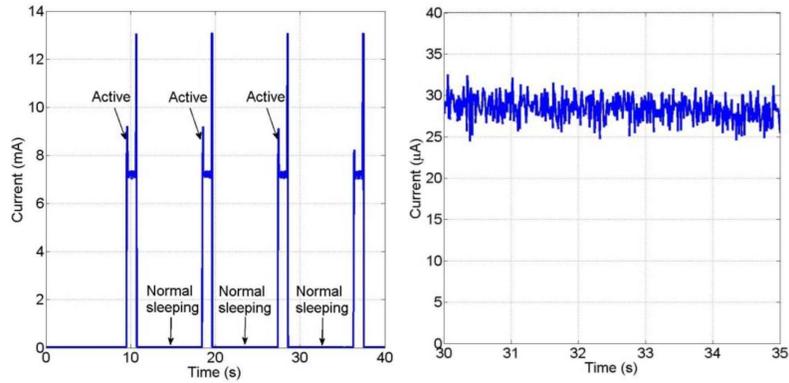


Figura 14: [16]

Nella seconda configurazione, in Fig. 14, V_{cs} raggiunge una tensione di 2.0 V per poi oscillare attorno ad essa (Fig. 14a). In questa situazione, l'MCU si accende e alcune delle sue componenti, quali ADC, DAC e comparatori riescono a funzionare, altre però non riescono nemmeno ad avviarsi, essendo la minima tensione di lavoro dell'MCU pari a 2.4 V. Nonostante ciò, le componenti sopracitate continuano a restare attive e l'MCU continua a ricevere energia. Inoltre il WSN durante la fase di attività riesce a ottenere una corrente di circa 3 mA (Fig. 14b), che scende poco dopo a circa 0.3 mA; in quanto la corrente media necessaria al funzionamento del nodo è di 8 mA, si può affermare che la configurazione priva di EAI non riesce a garantire nemmeno l'avvio. Ciò avviene perché l'energia immagazzinata in C_s viene istantaneamente consumata dal WSN e non viene invece accumulata fino al momento in cui non ve ne è abbastanza per assicurare l'operatività del sistema.

3.4.3 Configurazione III

Nella terza configurazione si può notare che il consumo di corrente nella fase attiva non si discosta eccessivamente da quello della prima configurazione, poiché



(a) Andamento di I_w durante un ciclo di attività
 (b) Andamento di I_w durante la fase di inattività

Figura 15: [16]

l'MCU è stato programmato per eseguire compiti simili tra i due setup. A variare è invece la corrente media nella fase di inattività, che, nella configurazione III, è di circa $28.3 \mu\text{A}$ (Fig. 15b); un consumo che è più di 29 volte maggiore rispetto a quello della configurazione I, dovuto alla presenza del timer interno e dell'oscillatore, usato per gestire i tempi di fase attiva e inattiva.

3.4.4 Confronto

Riassumendo i risultati ottenuti nei test sperimentali, si ha che:

- nella configurazione I, la presenza di un'interfaccia energy-aware e dell'associato programma garantisce da un lato il completo funzionamento del nodo, dall'altro un importante risparmio energetico nella fase di inattività;
- nella configurazione II, priva di EAI, V_{cs} non riesce a raggiungere il valore minimo per attivare correttamente il sistema, che non può quindi essere considerato stabile. Questo accade perché senza l'EAI non è possibile accumulare nel condensatore l'energia generata durante la fase di inattività;
- la configurazione III è estremamente più dispendiosa rispetto alla I nella fase di inattività, ma garantisce comunque piena operatività.

4 Conclusioni

Nell'elaborato è stato analizzato l'impatto che possono avere differenti policy di gestione energetica su una batteria ricaricabile e, in particolare, sulla diminuzione della sua capacità nominale nel corso del tempo, che rende un eventuale sistema basato sull'energy harvesting non più affidabile. È stato poi proposto

e caratterizzato un modello generale, idoneo a un processo di ottimizzazione [12]. Il modello, stocastico, è stato sviluppato da dati ottenuti dai produttori di dispositivi e sulla base di un modello deterministico. L'ottimizzazione, declinata come un problema di massimo sulla durata di vita di una batteria che deve garantire un minimo QoS, viene affrontata per mezzo di un algoritmo che tiene conto del deperimento della capacità durante il periodo di attività della batteria. I risultati numerici mostrano poi come sia possibile per policy "lifetime-aware" garantire un certo QoS e, al contempo, prolungare la vita della batteria e quindi dell'intero sistema, a differenza di policy il cui unico scopo è la massimizzazione del QoS stesso e che intaccano quindi in maniera pesante la durata del sopracitato dispositivo. Si è quindi passati allo studio di un metodo per assicurare di risparmiare energia in un sistema basato su EH e renderlo quindi maggiormente affidabile ed efficiente nel lungo termine [16]; lo sviluppo di un'interfaccia energy-aware corredata da una soluzione software si è dimostrata essere efficace per risolvere il problema. Le sperimentazioni hanno infatti mostrato che, grazie ad un approccio di questo tipo, è possibile da un lato colmare il divario tra energia richiesta dal WSN e prodotta dall'harvester, immagazzinandola intelligentemente in un condensatore, dall'altro permettere al nodo di essere alimentato da una corrente quasi trascurabile nella fasi di inattività, così da accrescere considerevolmente l'efficienza del sistema.

Riferimenti bibliografici

- [1] Seiko instruments inc. micro battery product catalogue. 2012. www.sii.co.jp/compo/catalog/battery_en.pdf.
- [2] I. F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.
- [3] T. M. Cover and J. A. Thomas. Elements of information theory. 2006.
- [4] B. Devillers and D. Gündüz. A general framework for the optimization of energy harvesting communication systems with battery imperfections. *Journal of Communications and Networks*, 14(2):130–139, 2012.
- [5] Alwyn D.T. Elliott and Paul D. Mitcheson. Implementation of a single supply pre-biasing circuit for piezoelectric energy harvesters. *Procedia Engineering*, 47:1311 – 1314, 2012. 26th European Conference on Solid-State Transducers, EUROSENSOR 2012.
- [6] Fariborz Entezami, Meiling Zhu, and Christos Politis. How much energy needs for running energy harvesting powered wireless sensor node? *Energy Harvesting and Systems*, 3(3):197 – 203, 01 Aug. 2016.
- [7] M. Ferrari, F. Cerini, and V. Ferrari. Autonomous sensor module powered by impact-enhanced energy harvester from broadband low-frequency

- vibrations. In *2013 Transducers Eurosensors XXVII: The 17th International Conference on Solid-State Sensors, Actuators and Microsystems (TRANSDUCERS EUROSENSORS XXVII)*, pages 2249–2252, 2013.
- [8] Daniel Guyomar and Mickaël Lallart. Recent progress in piezoelectric conversion and energy harvesting using nonlinear electronic interfaces and issues in small scale implementation. *Micromachines*, 2(2):274–294, Jun 2011.
- [9] C. K. Ho, P. D. Khoa, and P. C. Ming. Markovian models for harvested energy in wireless communications. In *2010 IEEE International Conference on Communication Systems*, pages 311–315, 2010.
- [10] T. Huang, M. Du, Y. Kang, R. Peng, K. Chen, Y. Lin, T. Tsai, C. Lee, L. Chen, and J. Chen. 120% harvesting energy improvement by maximum power extracting control for high sustainability magnetic power monitoring and harvesting system. *IEEE Transactions on Power Electronics*, 30(4):2262–2274, 2015.
- [11] D. Linden and T. B. Reddy. Handbook of batteries. *Journal of Communications and Networks*, 2002.
- [12] N. Michelusi, L. Badia, R. Carli, L. Corradini, and M. Zorzi. Energy management policies for harvesting-based wireless sensor devices with battery degradation. *IEEE Transactions on Communications*, 61(12):4934–4947, 2013.
- [13] N. Michelusi, K. Stamatiou, L. Badia, and M. Zorzi. Operation policies for energy harvesting devices with imperfect state-of-charge knowledge. In *2012 IEEE International Conference on Communications (ICC)*, pages 5782–5787, 2012.
- [14] D. Niyato, E. Hossain, M. M. Rashid, and V. K. Bhargava. Wireless sensor networks with energy harvesting technologies: a game-theoretic approach to optimal energy management. *IEEE Wireless Communications*, 14(4):90–96, 2007.
- [15] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener. Transmission with energy harvesting nodes in fading wireless channels: Optimal policies. *IEEE Journal on Selected Areas in Communications*, 29(8):1732–1743, 2011.
- [16] T. Ruan, Z. J. Chew, and M. Zhu. Energy-aware approaches for energy harvesting powered wireless sensor nodes. *IEEE Sensors Journal*, 17(7):2165–2173, 2017.
- [17] H. N. Seiger. Effect of depth of discharge on cycle life of near-term batteries. In *16th Intersociety Energy Conversion Engineering Conference*, volume 1, pages 102–110, January 1981.

- [18] L. Badia, A. Botta, L. Lenzini, “A genetic approach to joint routing and link scheduling for wireless mesh networks,” *Ad Hoc Networks*, vol. 7, no. 4, pp. 654-664, June 2009.
- [19] L. Badia, M. Levorato, F. Librino, and M. Zorzi, “Cooperation techniques for wireless systems from a networking perspective,” *IEEE Wireless Communications Magazine*, April 2010.
- [20] L. H. Thaller, “Expected cycle life vs. depth of discharge relationships of well-behaved single cells and cell strings,” *J. Electrochem. Soc.*, vol. 130, no. 5, pp. 986, 1983.
- [21] L. Canzian, L. Badia, and M. Zorzi, “Relaying in wireless networks modeled through cooperative game theory,” *Proc. IEEE CAMAD*, 2011.
- [22] R. Torah, P. Glynne-Jones, M. Tudor, T. ODonnell, S. Roy, and S. Beeby, “Self-powered autonomous wireless sensor node using vibration energy harvesting,” *Measurement Science and Technology*, 19(12):125202, Oct. 2008.
- [23] K. Tutunçuoğlu and A. Yener, “Communicating using an energy harvesting transmitter: Optimum policies under energy storage losses,” available at: <https://arxiv.org/abs/1208.6273>, 2012.
- [24] M. Zhu, V. Marsic, and S. Williams, “Wireless sensor communication system with low power consumption for integration with energy harvesting technology,” *Telfor Journal*, 4(2):89-94, 2012.
- [25] J. Zhang, P. Li, Y. Wen, F. Zhang, and C. Yang, “A management circuit with upconversion oscillation technology for electric-field energy harvesting,” *IEEE Transactions on Power Electronics*, vol. 31, no. 8, pp. 5515-5523, 2016.
- [26] L. Canzian, L. Badia, and M. Zorzi, “Promoting cooperation in wireless relay networks through Stackelberg dynamic scheduling,” *IEEE Trans. Commun.*, vol. 61, no. 2, pp. 700-711, 2012.
- [27] E. A. Jorswieck, L. Badia, T. Fahldieck, E. Karipidis, and J. Luo, “Spectrum sharing improves the network efficiency for cellular operators,” *IEEE Commun. Mag.*, vol. 52, no. 3, pp. 129-136, Mar. 2014.
- [28] L. Badia, “On the effect of feedback errors in Markov models for SR ARQ packet delays,” Proceedings IEEE Globecom, Honolulu (HI), USA, December 2009.
- [29] X. Wu, J. Yang, and J. Wu, “Optimal status update for age of information minimization with an energy harvesting source,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 2, no. 1, pp. 193-204, 2018.

- [30] F. Yuan, Q.T. Zhang, S. Jin, and H. Zhu, "Optimal Harvest-Use-Store Strategy for Energy Harvesting Wireless Systems," *IEEE transactions on wireless communications*, vol. 14, no. 2, pp. 698-710, 2015.
- [31] L. Badia, "A Markov analysis of selective repeat ARQ with variable round trip time," *IEEE Commun. Lett.*, vol. 17, no. 11, pp. 2184–2187, Nov. 2013.
- [32] N. Michelusi, L. Badia, R. Carli, K. Stamatiou, and M. Zorzi, "Correlated energy generation and imperfect state-of-charge knowledge in energy harvesting devices," *Proc. IWCMC Conference*, pp. 401-406. IEEE, 2012.
- [33] D. Altinel and G. K. Kurt, "Modeling of multiple energy sources for hybrid energy harvesting IoT systems," *IEEE Internet Things*, vol. 6, no. 6, pp. 10846–10854, 2019.
- [34] E. Gindullina, L. Badia, and X. Vilajosana, "Energy modeling and adaptive sampling algorithms for energy harvesting powered nodes with sampling rate limitations," *Trans. Emerg. Telecommun. Techn.*, vol. 31, no. 3 (2020): e3754.
- [35] L. Badia, E. Feltre, and E. Gindullina, "A Markov model accounting for charge recovery in energy harvesting devices," *Proc. IEEE WCNC Workshops*, 2017.
- [36] I. Ahmed, M. M. Butt, C. Psomas, A. Mohamed, I. Krikidis and M. Guizani, "Survey on Energy Harvesting Wireless Communications: Challenges and Opportunities for Radio Resource Allocation," *Elsevier Computer Networks*, vol. 88, pp. 234-248, 2015.
- [37] B. Da Lio, A. V. Guglielmi, and L. Badia, "Markov models for electric vehicles: the role of battery parameters and charging point frequency," *Proc. IEEE CAMAD*, pp. 207-210, 2015.
- [38] T. Zou, S. Lin, Q. Feng and Y. Chen, "Energy-Efficient Control with Harvesting Predictions for Solar-Powered Wireless Sensor Networks," *Sensors*, vol. 16, no. 53, 2016
- [39] L. Badia, "On the impact of correlated arrivals and errors on ARQ delay terms," *IEEE Transactions on Communications*, vol. 57, no. 2, pp. 334–338, February 2009.
- [40] R. Valentini, M. Levorato and F. Santucci, "Aging aware random channel access for battery-powered wireless networks," *IEEE Wirel. Comm. Lett.*, vol. 5, no. 2, pp. 176-179, 2016.
- [41] K. Tutuncuoglu, A. Yener, S. Ulukus, "Optimum policies for an energy harvesting transmitter under energy storage losses," *IEEE J. Sel. Ar. Commun.*, vol. 33, no. 3, pp. 467-481, 2015.

- [42] R.-S. Liu, P. Sinha, and C. Koksall, “Joint energy management and resource allocation in rechargeable sensor networks,” *Proc. IEEE Infocom*, pp. 1-9, 2010.
- [43] J. Yang and S. Ulukus, “Optimal packet scheduling in an energy harvesting communication system,” *IEEE Trans. Commun.*, vol. 60, no. 1, pp. 220-230, 2012.
- [44] M. Gatzianas, L. Georgiadis, and L. Tassiulas, “Control of wireless networks with rechargeable batteries,” *IEEE Trans. Wireless Commun.*, vol. 9, pp. 581–593, Feb. 2010.
- [45] Leonardo Badia and Giulia Mansutti, “Evaluation of operation policies for energy harvesting sensor nodes with variable data traffic,” *Proceedings of European Wireless 2015*, pp. 1-6, 2015.
- [46] E. Gindullina and L. Badia, “Towards self-control of service rate for battery management in energy harvesting devices,” *Proc. IEEE ICC Workshops*, pp. 355–360, 2017.
- [47] E. Gindullina and L. Badia, “An optimization framework for energy topologies in smart cities,” *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2018.
- [48] E. Gindullina, L. Badia, and D. Gunduz, “Average age-of-information with a backup information source,” *International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–6, IEEE, 2019.
- [49] O. Tan, D. Gunduz, and H. V. Poor, “Increasing smart meter privacy through energy harvesting and storage devices,” *IEEE J. Sel. Areas Commun.*, vol. 31, no. 7, pp. 1331–1341, 2013.
- [50] N. Jaggi, K. Kar, and A. Krishnamurthy, “Rechargeable sensor activation under temporally correlated events,” *Wireless Networks*, vol. 15, pp. 619–635, Jul. 2009.