

# Green IoT: Design e valutazione delle prestazioni di protocolli per sistemi IoT dotati di wake up radio



Leonardo Emili

Dipartimento di Informatica  
Sapienza Università di Roma

Luglio 2020

# Presentazione dello scenario

# Presentazione dello scenario

Il settore dell'**Internet of Things (IoT)** ha registrato enormi crescite durante gli ultimi anni. Nel 2019 si contano 42.1B di dispositivi connessi e le stime prevedono di raggiungere i 50B per la fine del 2020 (fonte: *IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey*).

Quotidianamente i dispositivi IoT sono utilizzati in svariati contesti, quali: monitoraggio della situazione clinica dei pazienti, recupero di informazioni circa la presenza di fenomeni sismici e in ambito domestico in cui rendono **smart** alcuni compiti comuni alla vita delle persone.

I diversi obiettivi sono associati all'eterogeneità dei compiti che devono risolvere ed in generale corrispondono ad esigenze diverse. Ad esempio il bisogno di ottenere informazioni in tempo reale circa la situazione clinica delle persone oppure i lunghi tempi di attività richiesti in ambito di monitoraggio del territorio.

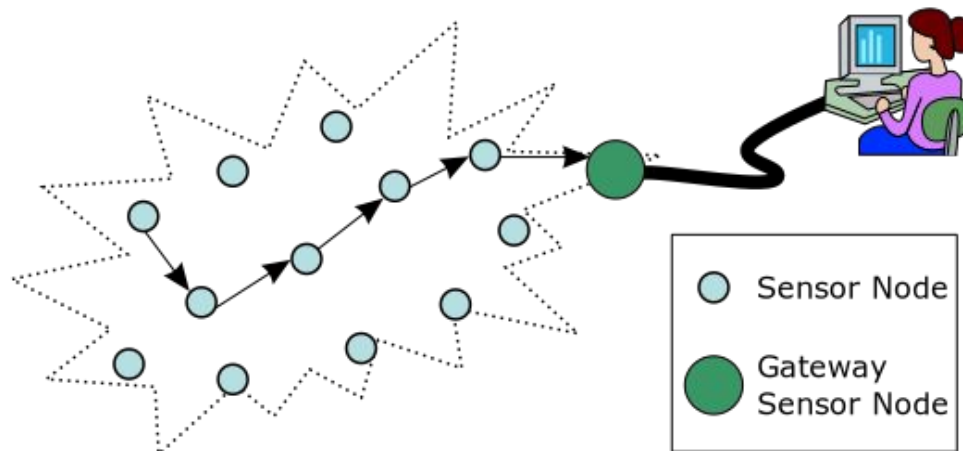
# Presentazione dello scenario: le reti di nodi sensori

In questo contesto presentiamo le **Wireless Sensor Network (WSN)**: reti di nodi sensori interconnessi il cui obiettivo consiste nella comunicazione *wireless* dei dati campionati circa l'ambiente circostante (fonte: *sciencedirect*).

Mediante protocolli di rete **multi-hop**, le WSN realizzano la trasmissione dei dati verso un nodo della rete che è responsabile del loro collezionamento. Tipicamente si fa riferimento a questo nodo con il nome di **sink node**.

L'obiettivo in questa tipologia di reti è quello di minimizzare i consumi energetici dell'intera rete al fine di estendere la *lifetime*. Le precedenti soluzioni riuscivano a garantire lifetime di diversi mesi, tuttavia gli attuali sistemi allo stato dell'arte permettono lifetime di diversi decenni (fonte: *Beyond Duty Cycling: Wake-up Radio with Selective Awakenings for Long-lived Wireless Sensing Systems*).

# Presentazione dello scenario: Wireless Sensor Network



Grafica illustrativa della comunicazione all'interno delle WSN.  
Fonte: *wikipedia*

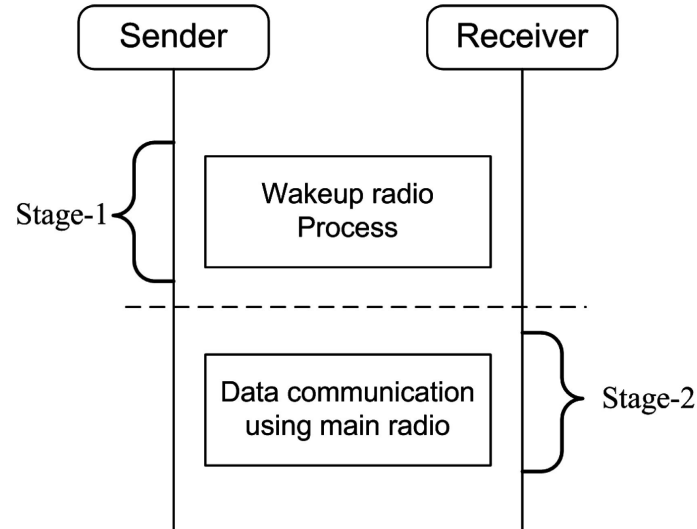
# Presentazione dello scenario: Green WSN

I principali consumi energetici nelle WSN consistono nella comunicazione tra nodi ma anche nel cosiddetto **idle listening**: ovvero il tempo in cui i nodi sono attivi con il trasmettitore pronto a ricevere senza che sia in corso alcuna comunicazione.

Una tecnica impiegata precedentemente per abbattere i consumi energetici consisteva nell'introduzione del **duty cycling** che permette ai nodi di rimanere attivi per una frazione del tempo originale. Tuttavia il questo approccio introduce ritardi nei tempi di trasmissione e non fornisce un reale beneficio alle prestazioni della rete.

In questo contesto viene introdotto il meccanismo delle **wake up radio**, il cui obiettivo consiste nell'eliminazione dei consumi energetici dovuti all'idle listening, che permette di abilitare nodi precedentemente inattivi alla ricezione di un pacchetto dati. Successivamente questa tecnica è stata ulteriormente migliorata grazie all'introduzione del **semantic addressing**, che permette di selezionare i nodi da attivare in base al loro stato, e del meccanismo di **energy harvesting** utilizzato per derivare energia da fonti energetiche esterne.

# Presentazione dello scenario: un nuovo approccio



Grafica illustrativa della procedura di wake up.

Fonte: *mdpi*

# Il protocollo GREEN-WUP



# Il protocollo GREEN-WUP: descrizione algoritmica (parte 1)

Il protocollo di rete GREEN-WUP è un protocollo del tipo **converge-casting** basato su **hop-count** abilitato alle tecnologie di wake up radio e energy harvesting (fonte: *A Novel wake-up Receiver with Addressing Capability for Wireless Sensor Nodes*).

In GREEN-WUP ciascun nodo è dotato di una coppia di indirizzi di wake up: il primo identifica **staticamente** il nodo all'interno della rete, mentre il secondo indirizzo è **dinamico** e riflette il suo stato energetico corrente. La disponibilità energetica di un nodo è rappresentata attraverso la sua discretizzazione in  $k$  classi, con  $k$  classe energetica massima.

In particolare si pone l'obiettivo di minimizzare i consumi energetici realizzando la comunicazione mediante la scelta di nodi intermedi (**relay node**) sulla base della loro disponibilità energetica. In altri termini, un nodo (**sender**) seleziona in maniera iterativa sottoinsiemi di nodi con classe energetica  $k, k-1, \dots, 1$ , sino a trovare il nodo che opererà da intermediario per la comunicazione destinata al sink node.

## Il protocollo GREEN-WUP: descrizione algoritmica (parte 2)

La fase di selezione del nodo intermedio avviene mediante lo scambio di **pacchetti di controllo**: il sender invia pacchetti **Request To Send (RTS)** per la ricerca di nodi intermedi e i nodi candidati rispondono inviando in **unicast** pacchetti **Clear To Send (CTS)** per indicare la loro disponibilità a reinstradare il pacchetto.

Una volta selezionato il relay node si procede inviando il **data packet** seguito dalla ricezione del relativo **acknowledgement packet (ACK)**. Si procede in questo modo fino ad trasmettere il pacchetto al sink node direttamente (nessuna fase di relay selection).

Per evitare collisioni relative all'invio simultaneo di pacchetti CTS, i nodi (**receiver**) calcolano prima un **jitter randomico** e poi inviano il pacchetto. Inoltre, lo scambio dei pacchetti RTS/CTS è condizionato dall'invio di sequenze di wake up che li precedono per evitare sprechi di energia.

# Il protocollo GREEN-WUP: diagramma degli stati

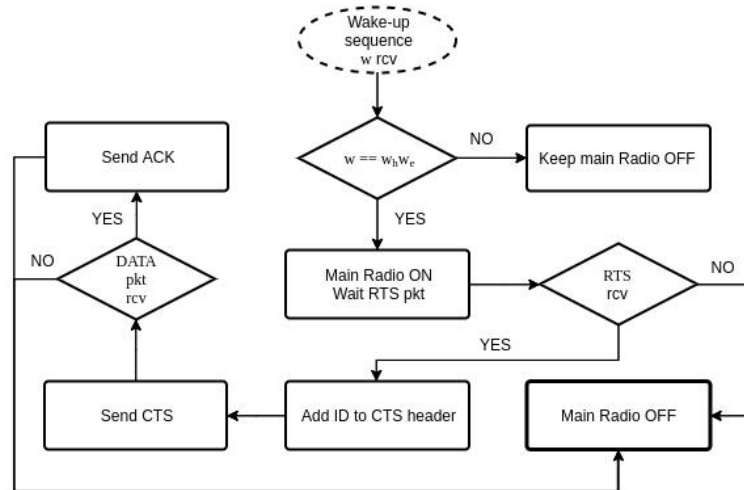
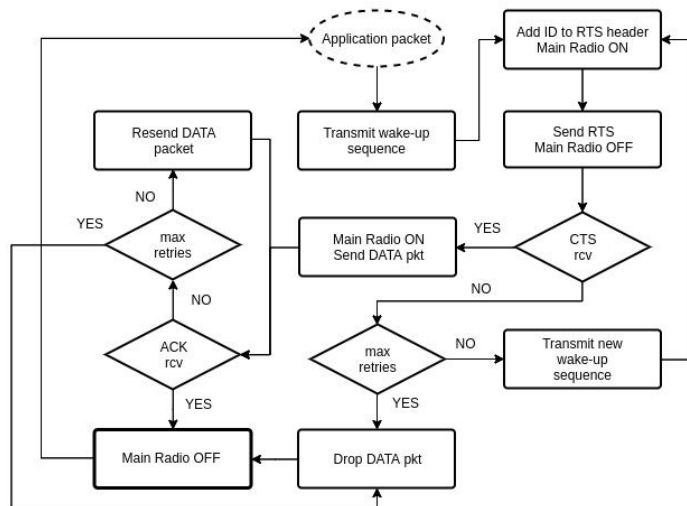


Illustrazione degli stati assunti dai nodi intermedi (hop count > 1),  
rispettivamente il sender (a sinistra) ed il receiver (a destra).

# Il protocollo GREEN-WUP: le problematiche emerse

L'implementazione originale del protocollo GREEN-WUP utilizza jitter puramente **randomici**. In altri termini i nodi rispondono inviando un pacchetto CTS senza prima considerare il loro **stato**.

L'aspetto semantico degli indirizzi di wake up considera il solo **stato energetico** del nodo. Ad esempio, dal momento che i nodi dispongono di un buffer di capacità limitata si potrebbe preferire un nodo con buffer di ricezione pieno ad un altro con buffer vuoto se questi condividono la stessa classe energetica (**buffer overflow**).

La ricezione di una sequenza di wake up durante la fase di relay selection comporta una risposta da parte dei nodi che attivano la radio principale (RX). Tuttavia se il numero dei nodi considerati è elevato si può avere uno **spreco energetico** considerevole in quanto solo un nodo sarà eventualmente selezionato come intermediario.

# La variante proposta

# La variante proposta: le modifiche applicate (parte 1)

In risposta alla mancanza dello stato del nodo  $j$  nella formulazione del jitter  $\delta$  si propone una modifica che tiene conto sia dello stato energetico che dello stato del buffer di ricezione, calcolata come:

$$\delta = (1 - e_j) \cdot k \cdot \delta_M + b_j \cdot l \cdot \delta_M + \delta_r$$

Inoltre l'idea del semantic addressing viene sviluppata al fine di mitigare il fenomeno di buffer overflow precedentemente descritto. In particolare, si codifica lo stato del buffer di ricezione all'interno dell'indirizzo di wake-up semantico per selezionare i nodi realmente abilitati alla ricezione di nuovi pacchetti.

Nella variante proposta vengono utilizzati **energy predictor** durante la fase di relay selection. I nodi rispondono ad una sequenza di wake-up attivando la radio principale (RX) con una probabilità che è proporzionale al valore di energia predetto, altrimenti lasciano la radio principale disattiva.

# La variante proposta: le modifiche applicate (parte 2)

---

**Algoritmo 4** Receiver nella variante

---

**Require:**  $s + p = 1$ , the node detects a wake up signal  $address_{RTS}$

$r \leftarrow$  random value in  $[0, maxEnergy]$

**if**  $r < predictEnergy(t)$  **then**

    activate the main radio and wait for some time  $\delta_{RTS}$

**if** RTS packet is received within  $\delta_{RTS}$  **then**

$\delta_W \leftarrow$  random value in  $[0, maxJitter)$

$\delta_E \leftarrow (maxJitter - \delta_W) \cdot s \cdot (1 - energyRatio)$

$\delta_B \leftarrow (maxJitter - \delta_W) \cdot p \cdot bufferState$

$\delta_{JITTER} \leftarrow \delta_W + \delta_E + \delta_B$

        wait for  $\delta_{JITTER}$  and then send a wake up signal using the address contained inside the RTS packet

$ctsPkt \leftarrow createCTS(self)$

        send  $ctsPkt$  and wait for some time  $\delta_{DATA}$

**if** DATA packet is received within  $\delta_{DATA}$  **then**

            send an acknowledgement packet to the sender

**end if**

**end if**

    process buffered packets if there are any, otherwise go back to sleep

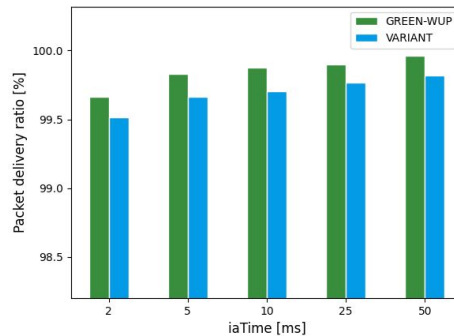
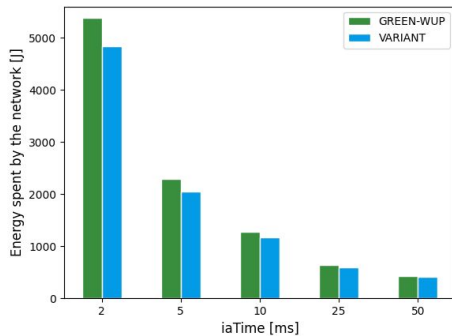
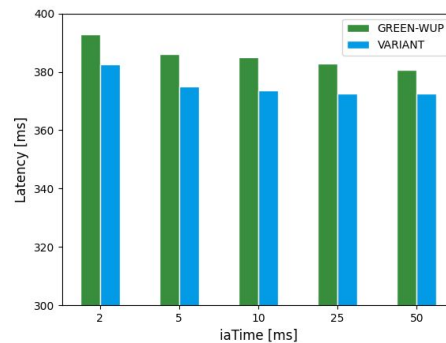
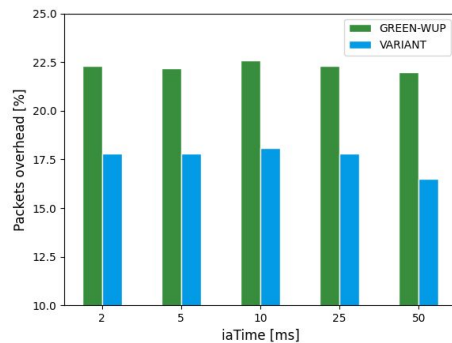
**end if**

---

# Valutazione delle prestazioni

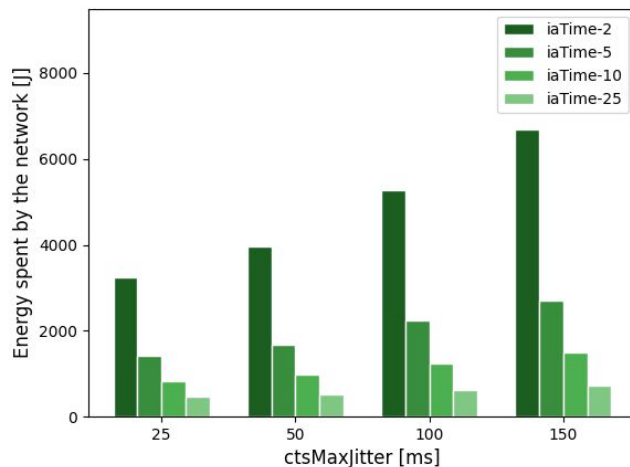
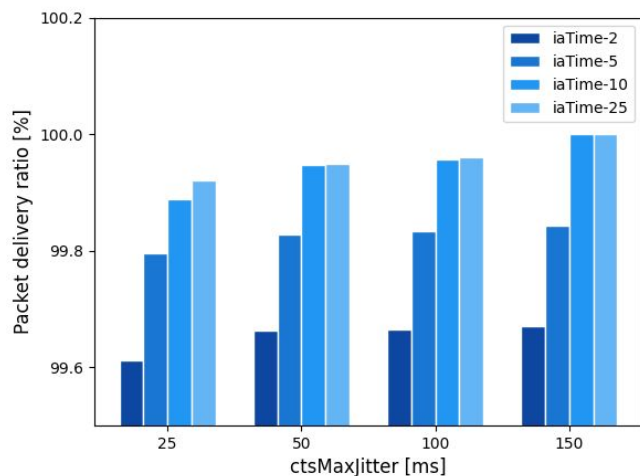


# Valutazione delle prestazioni



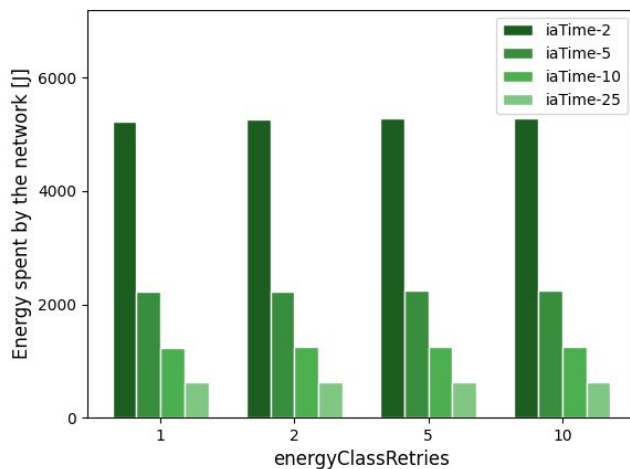
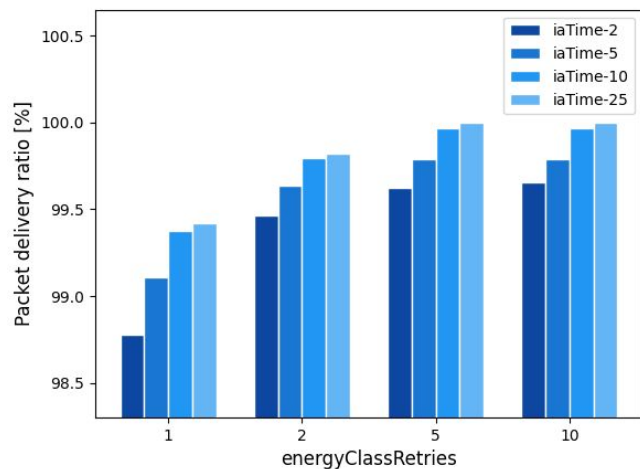
# Ottimizzazione dei parametri

# Ottimizzazione dei parametri: massimo jitter



Prestazioni della rete al variare del valore di *ctsMaxJitter*.

# Ottimizzazione dei parametri: tentativi per classe energetica



Prestazioni della rete al variare del valore di *energyClassRetries*.

# Sviluppi futuri

# Sviluppi futuri

Valutare le prestazioni del protocollo introducendo nuove modifiche che puntano ad ottenere migliori risultati in termini di efficienza energetica e di *packet delivery ratio*.

Sperimentare nuove condizioni di traffico o dello stato della rete (es. nodi altamente congestionati, link di trasmissione meno performanti).

Valutare le nuove varianti sviluppate mediante un testbed in ambiente reale (es. *SENSES Lab* ha realizzato un testbed per reti IoT dotate di wake up radio).

Grazie