



SAPIENZA
UNIVERSITÀ DI ROMA

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

Facoltà di Ingegneria dell'informazione, informatica e statistica

Tirocinio Formativo Attivo

Classe Informatica L-31

Candidato

Leonardo Emili

Matricola 1802989

Relatore

Prof. Chiara Petrioli

Tutor del Tirocinante

Dr. Georgia Koutsandria

Tutor Coordinatori

Prof. Chiara Petrioli

Dr. Georgia Koutsandria

Anno Accademico 2019/2020

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

TFA. Relazione di tirocinio. Sapienza – Università di Roma

© 2020 Leonardo Emili. Tutti i diritti riservati

Questa tesi è stata composta con \LaTeX e la classe Sapthesis.

Email dell'autore: emili.1802989@studenti.uniroma1.it

Indice

1	Introduzione	1
2	Scenario di riferimento	3
3	Il protocollo GREEN-WUP	5
4	Soluzioni proposte	9
5	Valutazione delle prestazioni	11
6	Conclusioni	13
	Bibliografia	15

Capitolo 1

Introduzione

Nell'ultimo decennio abbiamo assistito ad uno sviluppo portentoso del settore dell'Internet of Things che ha contribuito alla diffusione di un enorme numero di dispositivi wireless. Il numero di questi dispositivi ha registrato crescite costanti negli anni e loro applicazioni sono ormai infinite. Essi trovano impiego in ambito domestico dove realizzano l'automatizzazione dei compiti quotidiani, in quello sanitario in cui monitorano lo stato di salute dei pazienti e in ambito sottomarino dove gli obiettivi spaziano da quello di realizzare campionamenti di dati sino a creare vaste reti di comunicazione sottomarine. L'immissione di questo enorme contingente di dati nella rete ha reso possibili nuove interpretazioni e lo sviluppo di soluzioni che puntano a migliorare la qualità della vita delle persone, anche in quei settori noti esser di difficile comprensione. Ad esempio, l'impiego di sensori IoT in ambito di monitoraggio del crosta terrestre ha reso disponibili informazioni fino a prima sconosciute sulla presenza di terremoti e tsunami, migliorando radicalmente la nostra percezione degli eventi sismici nel mondo.

Con lo sviluppo massivo dell'IoT nuove sfide sono emerse a minare la solidità degli approcci usati. Attualmente le soluzioni impiegate nello sviluppo dei dispositivi wireless puntano a realizzare comunicazioni con basse latenze e che siano altamente efficienti in termini energetici. In ambito di ricerca, lo stato corrente dell'arte punta a realizzare tecnologie hardware e software che implementino questo binomio. Un particolare settore dell'IoT si occupa di realizzare vaste reti di nodi sensori interconnessi, note come *Wireless Sensor Networks*, che realizzano una comunicazione wireless con consumi minimi al fine di prolungare i tempi di servizio dell'intera rete. In questo caso, le principali problematiche sono chiaramente settoriali e sono identificate dallo specifico

campo di applicazione: come la richiesta di tecnologie altamente performanti per i monitoraggio clinici oppure di altre che garantiscano elevate lifetime nel caso di sensori ambientali. Tuttavia esistono problematiche comuni a queste tipologie di reti, ad esempio spesso si richiede che l'intera rete sia connessa e che tutti i nodi siano, anche parzialmente, a conoscenza della topologia della rete. Nel caso, si richiede l'impiego di protocolli che ne prevedano il costante aggiornamento e favoriscano una comunicazione efficiente. Inoltre, in queste reti la principale fonte di consumo energetico risiede nella comunicazione e nell'attesa che la precede, è quindi di vitale importanza prevedere che i nodi della rete siano oggetto di scaricamento delle batterie che li alimentano. Infatti, la morte prematura di un nodo può spesso avere conseguenze più grandi della semplice disconnessione dello stesso, poichè frequentemente si tratta di reti *multi-hop* che realizzano la comunicazione passando attraverso nodi intermedi, si può verificare perfino la disconnessione dell'intera rete.

Nel seguito verranno descritte le principali problematiche che motivano ricerca di nuove idee per la costruzione di protocolli di rete efficienti. Nel capitolo seguente analizzeremo lo scenario di riferimento e le principali considerazioni che nel tempo sono state adottate e che ad oggi definiscono il panorama delle Wireless Sensor Networks.

Nel capitolo 3 verrà presentato uno specifico protocollo di rete per sistemi IoT: il protocollo GREEN-WUP. In questo lavoro ci concentreremo sullo studio dei principi che regolano il design dei protocolli di rete e sulla loro applicazione in riferimento al protocollo in questione. Analizzeremo in dettaglio il comportamento del protocollo, descrivendone i principali vantaggi e le problematiche.

Nel capitolo 4 verrà presentata una variante del protocollo originale che punta a risolvere le problematiche osservate nel capitolo precedente. Le modifiche apportate saranno descritte in dettaglio sia a livello teorico che a livello di implementazione all'interno di Castalia, simulatore utilizzato a cui faremo diffusamente riferimento all'interno di questa relazione.

Nel capitolo 5 saranno analizzate le prestazioni della variante proposta e quelle del protocollo originale. In particolare, verranno mostrati i comportamenti di entrambe le implementazioni all'interno di uno stesso ambiente reale ed infine nel Capitolo 6 trarremo le conclusioni di quanto osservato.

Capitolo 2

Scenario di riferimento

Sino a qualche anno fa, la principale tendenza nello sviluppo di soluzioni ai problemi menzionati nel Capitolo 1 consisteva nell'introduzione di cicli di attività dei nodi. Il cosiddetto *duty cycling*, definito come la frazione di tempo in cui un nodo è attivo, aveva come obiettivo quello di minimizzare i momenti in cui i nodi sensori sono accessi senza alcun processamento di dati attivo. Seppur questo approccio permetta di prolungare la durata della vita della rete, incrementa in maniera considerevole i ritardi nelle comunicazioni, dal momento che queste possono avvenire solamente all'interno delle finestre di attività delle coppie dei nodi coinvolti. Tuttavia l'utilizzo del *duty cycling* non permette di risolvere tutti i problemi in quanto le comunicazioni possono ancora avvenire nei momenti di inattività dei nodi. Dualmente, esistono ancora i periodi di tempo in cui i nodi consumano energia pur non ricevendo o inviando dati.

In questo panorama vengono introdotte le *wake up radios*, in grado di abbattere i consumi energetici mantenendo bassi i ritardi nelle comunicazioni. La principale differenza rispetto all'approccio tradizionale consiste nell'introdurre un'antenna secondaria per i soli messaggi di wake-up costantemente attiva. La nuova antenna è progettata per avere dimensioni ridotte e consumi energetici nettamente inferiori rispetto a quelli della radio principale. Al momento della ricezione di un messaggio di wake-up è possibile attivarla e ricevere il pacchetto dati. Di fatto questa tecnologia implementa uno schema di comunicazione *on demand*: un nodo può inviare un pacchetto dati ad un nodo dormiente semplicemente posticipandone l'invio a quello di una sequenza di wake-up che, svegliando il ricevente, lo abilita alla ricezione. In questo modo i nodi possono rimanere attivi per il solo tempo minimo necessario a svolgere l'attività richiesta e lasciare spenta la radio principale nel tempo rimanente. Dal punto vista

energetico questo approccio è altamente efficiente e non richiede alcuna forma di sincronizzazione che è generalmente non desiderata in quanto implicitamente introduce overhead dovuto alla gestione di timer aggiuntivi.

Più recentemente la tendenza è quella di equipaggiare i nodi sensori, dotati di wake up radios, con moduli per l'*energy harvesting* come ulteriore supporto energetico alla loro durata di attività. In particolare, i nodi sono in grado di utilizzare l'energia fornita dalle pile elettriche di cui sono forniti e di derivarne nuova mediante turbine eoliche, pannelli solari e generatori termoelettrici di cui sono equipaggiati. La ricerca ha dimostrato che l'impiego di nodi dotati di energy harvesting [1] e che sono abilitati alle wake up radios [2] ha portato notevoli incrementi nelle performance delle Wireless Sensor Networks.

Inoltre, l'innovativa tecnologia delle wake up radios può essere combinata con il cosiddetto *semantic addressing*. L'idea in questo caso è di limitare ad un sottoinsieme di nodi la ricezione di una sequenza di wake up, evitando quindi che questa sia recepita da tutti i nodi presenti all'interno del range di comunicazione. Il principio fondante è quello di scegliere una o più opportune sequenze di wake up e di assegnarle ad un certo sottoinsieme di nodi della rete. A meno di errori di interpretazione della sequenza di wake up, al momento della ricezione tutti e soli i nodi desiderati saranno svegliati. In questo scenario, i consumi energetici risultano essere ulteriormente ridotti poichè viene virtualmente azzerato il numero dei nodi che sono riattivati a seguito di sequenze di wake up non a loro destinate ma che si trovano all'interno del loro range.

Questo lavoro si concentra sullo studio di protocolli delle cosiddette *green wireless sensor network*, ovverosia di quei protocolli che si basano sulle tecnologie sopra esposte e che implementano la comunicazione tra nodi minimizzandone gli sprechi energetici al fine di massimizzare i tempi di attività e il corretto funzionamento delle comunicazioni all'interno delle reti. Nella letteratura scientifica sono presenti molti protocolli di questo tipo (es. CTP-WUR, GREENROUTES, WHARP) e ciascuno di loro punta a risolvere diverse problematiche. Nel seguito verrà analizzato in dettaglio il comportamento del protocollo GREEN-WUP che utilizza energy harvesting e wake up radio per realizzare un'opportuna scelta del percorso da seguire per permettere la ricezione dati da parte del sink node.

Capitolo 3

Il protocollo GREEN-WUP

Il protocollo GREEN-WUP si inserisce nel contesto dei protocolli delle *green wireless sensor network* e pone tra i suoi principali obiettivi la massima efficienza energetica della rete. Esso impiega le tecnologie di wake up radio, energy harvesting e semantic addressing. Inoltre, è di tipo *converge casting* ed è basato sull'assegnazione di hop count a ciascun nodo per poter distribuire i pacchetti dati all'interno della rete.

Algoritmo 1 Sender in GREEN-WUP

Require: $level > 0$

```

while buffer is not empty do
   $dataPkt \leftarrow \text{buffer.pop}()$ 
  if level = 1 then
    send  $dataPkt$  to the sink node and wait for some time  $\delta_{ACK}$  to receive the
    relative ACK packet
  else
     $k \leftarrow \text{maxEnergyClass}$ 
     $acked \leftarrow \text{false}$ 
    while  $k > 0$  and not  $acked$  do
       $address_{RTS} \leftarrow \text{createWurAddress}(\text{level} - 1, k)$ 
      send  $address_{RTS}$  and wait for nearby nodes to wake up
       $rtsPkt \leftarrow \text{createRTS}(\text{self})$ 
      send  $rtsPkt$  in broadcast and wait for some time  $\delta_{CTS}$ 
      if CTS packet is received within  $\delta_{CTS}$  then
        send a wake up signal using the address contained inside the first CTS
        packet received and wait for the receiver to wake up
        send  $dataPkt$  and wait for some time  $\delta_{ACK}$ 
        if ACK packet is received within  $\delta_{ACK}$  then
           $acked \leftarrow \text{true}$ 
        end if
      end if
       $k \leftarrow k - 1$ 
    end while
  end if
end while

```

Algoritmo 2 Receiver in GREEN-WUP

Require: the node detects a wake up signal $address_{RTS}$

```

activate the main radio and wait for some time  $\delta_{RTS}$ 
if RTS packet is received within  $\delta_{RTS}$  then
   $\delta_{JITTER} \leftarrow \text{random value in } [0, \text{maxJitter})$ 
  wait for  $\delta_{JITTER}$  and then send a wake up signal using the address contained
  inside  $rtsPkt$ 
   $ctsPkt \leftarrow \text{createCTS}(\text{self})$ 
  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

```

Capitolo 4

Soluzioni proposte

Algoritmo 3 Sender nella variante

Require: $level > 0$

```

while buffer is not empty do
   $dataPkt \leftarrow \text{buffer.pop}()$ 
  if  $level = 1$  then
    send  $dataPkt$  to the sink node and wait for some time  $\delta_{ACK}$  to receive the
    relative ACK packet
  else
     $k \leftarrow \text{maxEnergyClass}$ 
     $acked \leftarrow \text{false}$ 
    while  $k > 0$  and not  $acked$  do
       $u \leftarrow \text{maxBufferClass}$ 
      while  $u > 0$  and not  $acked$  do
         $address_{RTS} \leftarrow \text{createWurAddress}(level - 1, k, u)$ 
        send  $address_{RTS}$  and wait for nearby nodes to wake up
         $rtsPkt \leftarrow \text{createRTS}(\text{self})$ 
        send  $rtsPkt$  in broadcast and wait for some time  $\delta_{CTS}$ 
        if CTS packet is received within  $\delta_{CTS}$  then
          send a wake up signal using the address contained inside the first CTS
          packet received and wait for the receiver to wake up
          send  $dataPkt$  and wait for some time  $\delta_{ACK}$ 
          if ACK packet is received within  $\delta_{ACK}$  then
             $acked \leftarrow \text{true}$ 
          end if
        end if
         $u \leftarrow u - 1$ 
      end while
       $k \leftarrow k - 1$ 
    end while
  end if
end while

```

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 δ_{RTS}

if RTS packet is received within δ_{RTS} **then**

$\delta_{RANDOM} \leftarrow$ random value in $[0, 1)$

$\delta_W \leftarrow \delta_{RANDOM} \cdot maxJitter$

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

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

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

wait for δ_{JITTER} and then send a wake up signal using the address contained inside $rtsPkt$

$ctsPkt \leftarrow createCTS(self)$

send $ctsPkt$ and wait for some time δ_{DATA}

if DATA packet is received within δ_{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

Capitolo 5

Valutazione delle prestazioni

Capitolo 6

Conclusioni

Bibliografia

- [1] Stefano Basagni, Georgia Koutsandria, Chiara Petrioli. *A Comparative Performance Evaluation of Wake-up Radio-based Data Forwarding for Green Wireless Networks.*
- [2] Stefano Basagni, Valerio Di Valerio, Georgia Koutsandria, Chiara Petrioli. *Wake-up Radio-enabled Routing for Green Wireless Sensor Networks*