REPORT HOME CHALLENGE #2

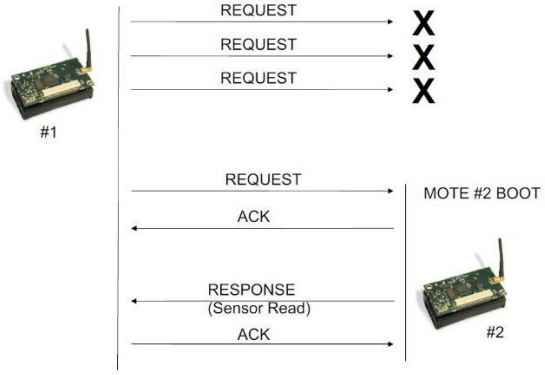
Costa Giovanni 10735519

Polo Enrico 10528160

Salani Francesco 10748052

Repository link: https://github.com/Salani-Polimi/HomeChallange2-CostaPoloSalani.git

SCOPO: Simulazione in TOSSIM di un sistema di Acknowledgments tra due motes, come riportato nella seguente figura.



IMPLEMENTAZIONE

1) **PACCHETTO**
   Il pacchetto è implementato tramite una struct all'interno del file .h. Esso contiene tre interi: counter della richiesta, type e data.

2) **INTERFACCE**

   Le interfacce utilizzate sono:
   - Boot;
   - Receive;
   - AMSender;
   - Timer;
   - SplitControl;
   - Packet;
   - PacketAcknowledgments;

3) **COLLEGAMENTI AppC.nc**

   Le componenti sono:
   - MainC, sendAckC as App;
   - New AMSenderC (AM_MY_MSG);
   - New AMReceiverC (AM_MY_MSG);
   - New TimerMilliC() as Timer*1*;

- ActivateMessageC;
- Collegamenti interfacce (facendo attenzione a App.PacketAcknowledgments-> ActivateMessageC);

4) LOGICA C.nc

Il mote 1 si accende e invia periodicamente (1s) pacchetti, type REQ e counter pari al numero di request, al mote 2, richiedendo l'ACK per ciascuno.
Il mote 2 si accende dopo 5s i primi pacchetti verranno persi. Non appena riceve correttamente un pacchetto, invia l'ACK ad 1 che terminerà la trasmissione di REQs. Inviato l'ACK il mote 2 invia un pacchetto type RESP, counter pari al campo counter relativo al payload cui fa riferimento infine il campo *data* viene riempito con un valore prelevato dal sensore di temperatura (*Fake Sensor*).
I punti principali sono l'implementazione dell'invio/ricezione dell'ACK. È stato utilizzato il modulo PacketAcknowledgments in particolare tramite tramite i *command*:

1) *requestAck*: setta il flag per la richiesta dell'Ack;
2) *wasAcked*: verifica la ricezione dell'Ack sul pacchetto precedentemente inviato;

Quando il mote 2 riceverà l'Ack relativo alla sua RESP allora viene spento il sistema (Figura 1), Idle listening power consumption, altrimenti si procederà al reinvio dello stesso pacchetto.
In particolare è stato notato che, seppur con scarsa probabilità, non sempre il mote due riceve l'ACK relativo alla RESP (Figura 2). Il problema è stato risolto usando una la bool *data_lock* che permette di riutilizzare il valore letto dal Fake Sensor in caso di non corretta ricezione dell'Ack, altrimenti di riaccedere alla lettura dal sensore per un nuovo dato. Il secondo caso non si verifica mai per via dello spegnimento della radio.

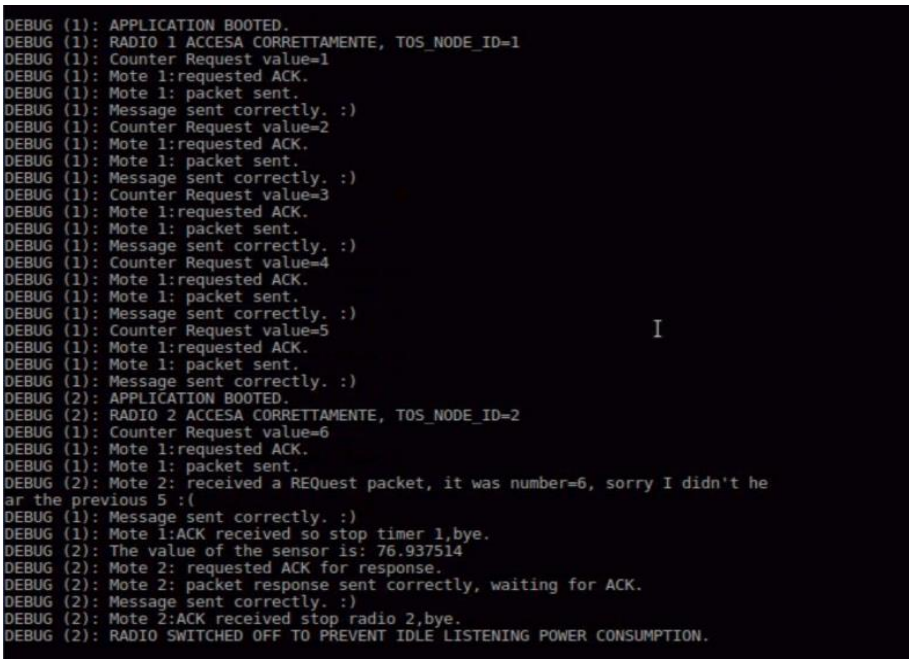


Figura 1

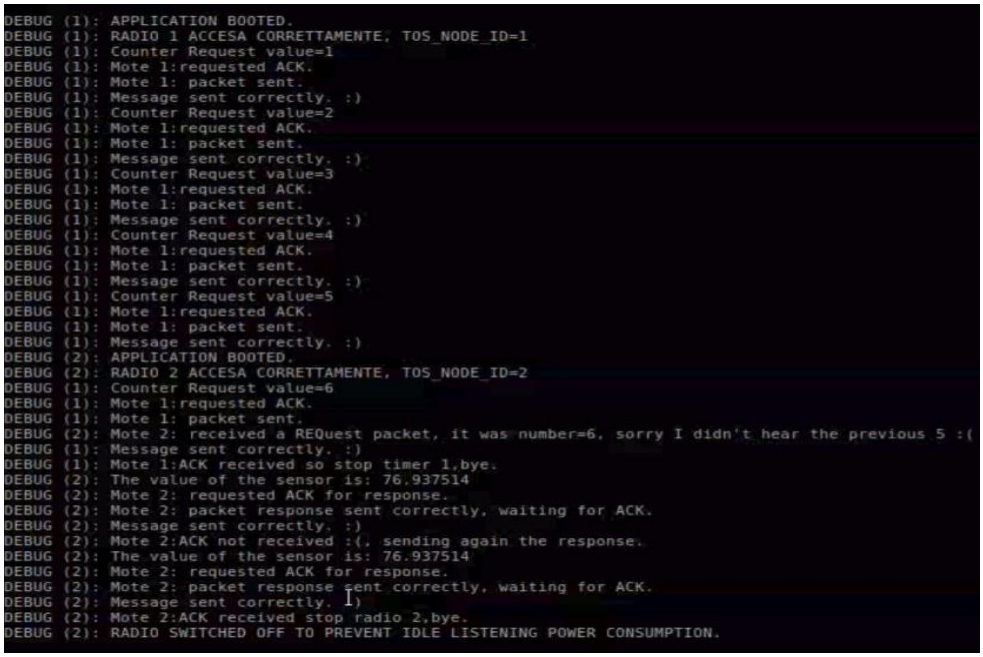

DEBUG (1): APPLICATION BOOTED.
DEBUG (1): RADIO 1 ACCESA CORRETTAMENTE, TOS_NODE_ID=1
DEBUG (1): Counter Request value=1
DEBUG (1): Mote 1:requested ACK.
DEBUG (1): Mote 1: packet sent.
DEBUG (1): Message sent correctly. :)
DEBUG (1): Counter Request value=2
DEBUG (1): Mote 1:requested ACK.
DEBUG (1): Mote 1: packet sent.
DEBUG (1): Message sent correctly. :)
DEBUG (1): Counter Request value=3
DEBUG (1): Mote 1:requested ACK.
DEBUG (1): Mote 1: packet sent.
DEBUG (1): Message sent correctly. :)
DEBUG (1): Counter Request value=4
DEBUG (1): Mote 1:requested ACK.
DEBUG (1): Mote 1: packet sent.
DEBUG (1): Message sent correctly. :)
DEBUG (1): Counter Request value=5
DEBUG (1): Mote 1:requested ACK.
DEBUG (1): Mote 1: packet sent.
DEBUG (1): Message sent correctly. :)
DEBUG (2): APPLICATION BOOTED.
DEBUG (2): RADIO 2 ACCESA CORRETTAMENTE, TOS_NODE_ID=2
DEBUG (1): Counter Request value=6
DEBUG (1): Mote 1:requested ACK.
DEBUG (1): Mote 1: packet sent.
DEBUG (2): Mote 2: received a REQuest packet, it was number=6, sorry I didn't hear the previous 5 :(
DEBUG (1): Message sent correctly. :)
DEBUG (1): Mote 1:ACK received so stop timer 1,bye.
DEBUG (2): The value of the sensor is: 76.937514
DEBUG (2): Mote 2: requested ACK for response.
DEBUG (2): Mote 2: packet response sent correctly, waiting for ACK.
DEBUG (2): Message sent correctly. :)
DEBUG (2): Mote 2:ACK not received :(, sending again the response.
DEBUG (2): The value of the sensor is: 76.937514
DEBUG (2): Mote 2: requested ACK for response.
DEBUG (2): Mote 2: packet response sent correctly, waiting for ACK.
DEBUG (2): Message sent correctly. :)
DEBUG (2): Mote 2:ACK received stop radio 2,bye.
DEBUG (2): RADIO SWITCHED OFF TO PREVENT IDLE LISTENING POWER CONSUMPTION.

Figura 2