Localized Radio Sleep Protocol

Ben Duggan

October 3, 2018

1 Purpose

The nRF24 radio modules are a great relatively low power module. However, when attaching the radios to a device ran on batteries, where up time is crucial, performance must sometimes be sacrificed by turning off the radios to ensure that the system lasts long enough to be useful. For this reason this protocol has been designed to allow radios to power down all at once and then all turn on after a predetermined amount of time. The two parts to this protocol are the server node (Raspberry Pi, master node or node 0) and the readers (child nodes nodes 1-127).

2 Method

The servers radio will always be kept on. This is to help ensure that reconnections can happen quickly and more serious problems dont occur. By default all radios are on and only power down when the reader does so or when the server tell the reader to. When a reader has no reads in its queue it will tell the server that it is able to sleep. If every node in the sub-tree, with this node as the root, have sent a sleep command then the every node in the sub-tree will be told to go to sleep. These nodes will power down for the predefined amount of time but if there are children in this tree that already sleeping a time will be used that ensures they can reconnect when the they power on.

3 Pseudocode

3.1 server

3.2 reader

Algorithm 1 Server Periodic Radio Sleep

```
$sleepTime \leftarrow$ length sleep in milliseconds
if A message of type $'S'$ is sent and the first char is "Y" then
    Record that this node can sleep
    if Every child of this node can sleep then
        Check so see if any child is currently sleeping in BFS order. If they are use $sleepTime$ - duration of time that node has been sleeping as the sleep time.
        Put each child to sleep in DFS order, recording that it's sleeping and the time it went to sleep.
    end if
end if
if A message of type $'S'$ is sent and the first char is "N" then
    Record that this node cannot sleep.
end if
```

Algorithm 2 Reader Localized Radio Algorithm

```
$sleepTime \leftarrow$ length sleep in milliseconds received from the server
$sleepTimeRecieved \leftarrow$ time in milliseconds when the sleep command was received
if milliseconds()-$sleepTimeRecieved > sleepTime$ then
    $radio$.powerUp()
    $radio$.requestAddress()
end if
if $message$ containing sleep code $'S'$ is received and first char in $message$ is a "S" then
    $sleepTime = message.timeToSleep$
    $sleepTimeRecieved =$ milliseconds()
    $network$.releaseAddress()
    $radio$.powerDown()
end if
if $readQueue$.length() == 0 and a sleep command hasn't been sent then
    if sleep command hasn't been sent then
        Send $server$ sleep command $'S'$ with the first char of message "Y"
    end if
else
    if sleep command has been sent then
        Send $server$ sleep command $'S'$ with the first char of message "N"
    end if
end if
```