Usevalad Milasheuski 10816982

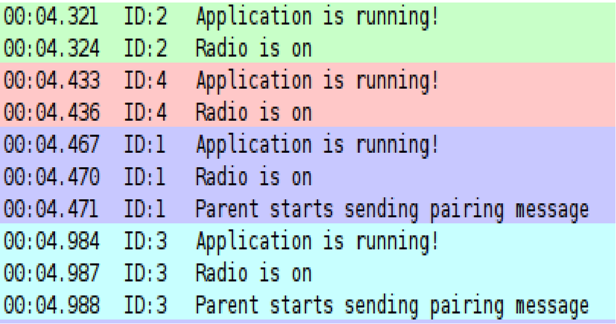
IoT Project

Smart Bracelets

0) The simulation is performed in Cooja using 4 Sky motes (make telosb). Odd nodes correspond to parents, even – to children. The code provides flexibility, meaning that only addition of an extra key would allow to add an extra pair of nodes. Log file contains the 5min of simulation different from the screenshots in the report, but still showing all the simulation requirements to be implemented in the project.

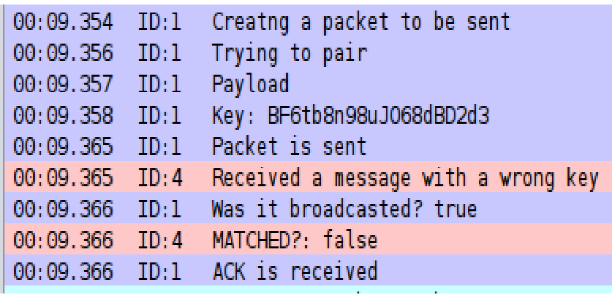
1)Pairing phase

In this part every parent node, which has an odd identifier, chooses the key for itself from an array and sends a message to be acknowledged in a broadcast way. Whoever receives the message, compares the stored key with the sent one and drops the message in case of mismatch. In case the key in the message and the stored one match, the address of the sender is stored to be used for the next phase. Then the reply is formed, containing an "OK" state, which is checked at the parent's side.



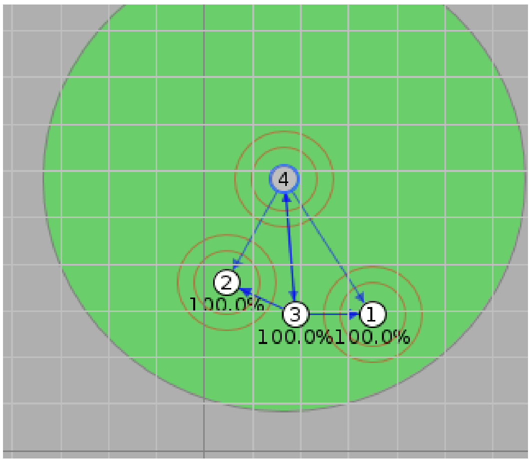
```
00:04.321  ID:2  Application is running!
00:04.324  ID:2  Radio is on
00:04.433  ID:4  Application is running!
00:04.436  ID:4  Radio is on
00:04.467  ID:1  Application is running!
00:04.470  ID:1  Radio is on
00:04.471  ID:1  Parent starts sending pairing message
00:04.984  ID:3  Application is running!
00:04.987  ID:3  Radio is on
00:04.988  ID:3  Parent starts sending pairing message
```

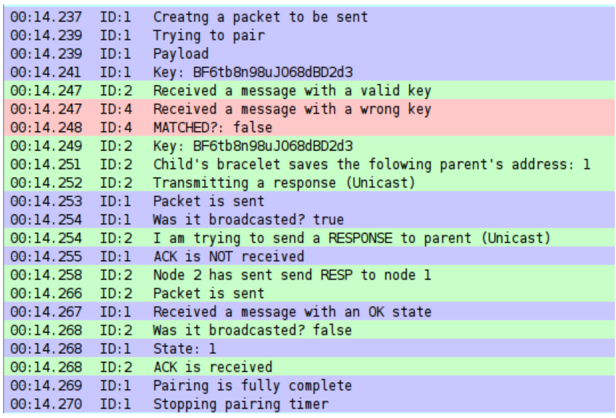
Fig1. Initial booting of the motes



```
00:09.354  ID:1  Creatng a packet to be sent
00:09.356  ID:1  Trying to pair
00:09.357  ID:1  Payload
00:09.358  ID:1  Key: BF6tb8n98uJ068dBD2d3
00:09.365  ID:1  Packet is sent
00:09.365  ID:4  Received a message with a wrong key
00:09.366  ID:1  Was it broadcasted? true
00:09.366  ID:4  MATCHED?: false
00:09.366  ID:1  ACK is received
```

Fig2. A mismatch scenario



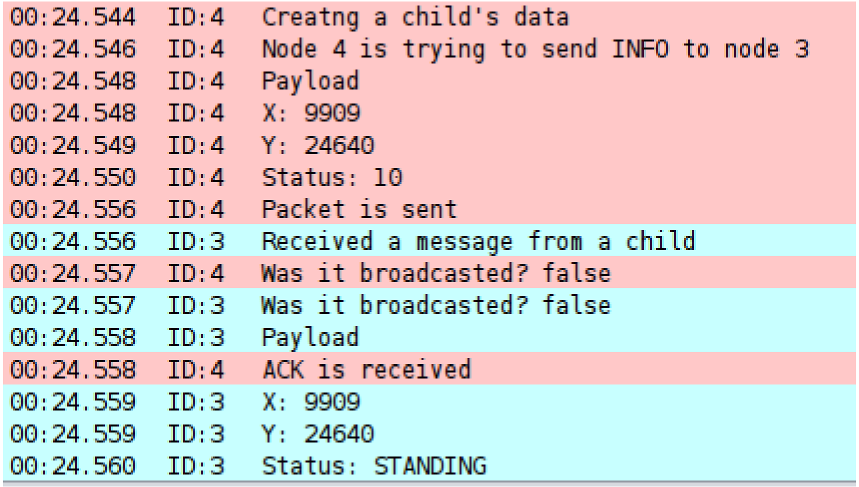


```
00:14.237  ID:1  Creatng a packet to be sent
00:14.239  ID:1  Trying to pair
00:14.239  ID:1  Payload
00:14.241  ID:1  Key: BF6tb8n98uJ068dBD2d3
00:14.247  ID:2  Received a message with a valid key
00:14.247  ID:4  Received a message with a wrong key
00:14.248  ID:4  MATCHED?: false
00:14.249  ID:2  Key: BF6tb8n98uJ068dBD2d3
00:14.251  ID:2  Child's bracelet saves the folowing parent's address: 1
00:14.252  ID:2  Transmitting a response (Unicast)
00:14.253  ID:1  Packet is sent
00:14.254  ID:1  Was it broadcasted? true
00:14.254  ID:2  I am trying to send a RESPONSE to parent (Unicast)
00:14.255  ID:1  ACK is NOT received
00:14.258  ID:2  Node 2 has sent send RESP to node 1
00:14.266  ID:2  Packet is sent
00:14.267  ID:1  Received a message with an OK state
00:14.268  ID:2  Was it broadcasted? false
00:14.268  ID:1  State: 1
00:14.268  ID:2  ACK is received
00:14.269  ID:1  Pairing is fully complete
00:14.270  ID:1  Stopping pairing timer
```

Fig3,4 Example of pairing between two motes, when all of them are in proximity to each other (similar output between motes 3-4)

2) Operation mode

Having completed the previous phase, child's node sends periodically (triggering a timer every 10s) a new type of messages containing X,Y and STATUS fields. These messages are sent in Unicast, since the address is known from the previous phase and saved in *address_coupled_device* variable. The acceptance of the message also depends on the id of the child, meaning that only messages from coupled bracelets will be parsed. For the convenience, he status is encoded with values and has the following structure: STANDING 10, WALKING 20, RUNNING 30, FALLING 40. The coordinates are generated using *call Random.rand16()* method, the status is selected by generating an integer value between 1 and 10 and observing to which region the value falls.



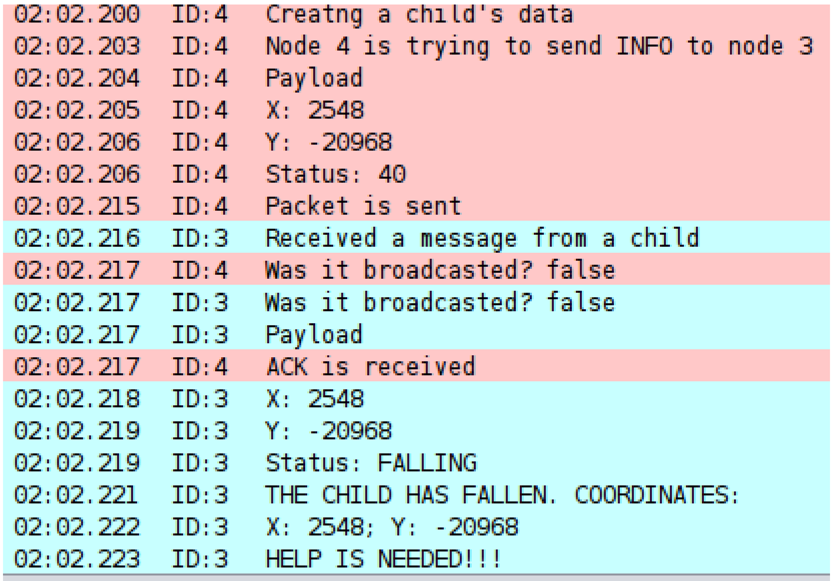
```
00:24.544  ID:4  Creatng a child's data
00:24.546  ID:4  Node 4 is trying to send INFO to node 3
00:24.548  ID:4  Payload
00:24.548  ID:4  X: 9909
00:24.549  ID:4  Y: 24640
00:24.550  ID:4  Status: 10
00:24.556  ID:4  Packet is sent
00:24.556  ID:3  Received a message from a child
00:24.557  ID:4  Was it broadcasted? false
00:24.557  ID:3  Was it broadcasted? false
00:24.558  ID:3  Payload
00:24.558  ID:4  ACK is received
00:24.559  ID:3  X: 9909
00:24.559  ID:3  Y: 24640
00:24.560  ID:3  Status: STANDING
```

Fig.5 Example of communication during the operation mode

3)Alert mode

-FALL alarm

When child’s message contains the “FALLING” status, it triggers a function which notifies a parent of the situation and reports the position.

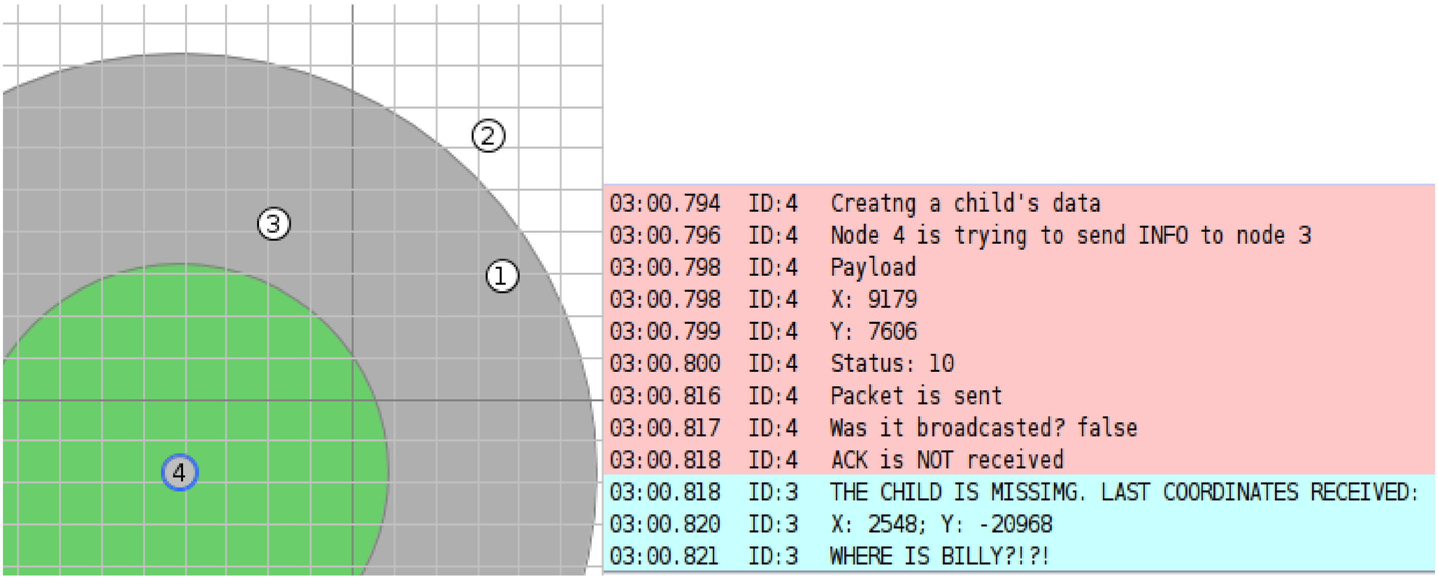


```
02:02.200  ID:4  Creatng a child's data
02:02.203  ID:4  Node 4 is trying to send INFO to node 3
02:02.204  ID:4  Payload
02:02.205  ID:4  X: 2548
02:02.206  ID:4  Y: -20968
02:02.206  ID:4  Status: 40
02:02.215  ID:4  Packet is sent
02:02.216  ID:3  Received a message from a child
02:02.217  ID:4  Was it broadcasted? false
02:02.217  ID:3  Was it broadcasted? false
02:02.217  ID:3  Payload
02:02.217  ID:4  ACK is received
02:02.218  ID:3  X: 2548
02:02.219  ID:3  Y: -20968
02:02.219  ID:3  Status: FALLING
02:02.221  ID:3  THE CHILD HAS FALLEN. COORDINATES:
02:02.222  ID:3  X: 2548; Y: -20968
02:02.223  ID:3  HELP IS NEEDED!!!
```

Fig.6 Example of the falling alarm

-MISSING alarm

To simulate this event, we can separate two coupled nodes and leave them for a minute. The logic is the following: A timer is periodically trigged every minute on the parent’s device, but a reception of a message resets the timer. If a message has not been received for a minute, the timer eventually fires triggering a function with a notification and last received coordinates. If the nodes are brought back together, operating mode will function as usual.



```
03:00.794   ID:4   Creatng a child's data
03:00.796   ID:4   Node 4 is trying to send INFO to node 3
03:00.798   ID:4   Payload
03:00.798   ID:4   X: 9179
03:00.799   ID:4   Y: 7606
03:00.800   ID:4   Status: 10
03:00.816   ID:4   Packet is sent
03:00.817   ID:4   Was it broadcasted? false
03:00.818   ID:4   ACK is NOT received
03:00.818   ID:3   THE CHILD IS MISSIMG. LAST COORDINATES RECEIVED:
03:00.820   ID:3   X: 2548; Y: -20968
03:00.821   ID:3   WHERE IS BILLY?!?!
```

Fig. 7, 8 Example of missing alarm, when node 3(parent) has not received anything from its child (node 3)

Note: For some reasons Cooja displays that ACK messages are sent to multiple entities. It might be a bug, but I have several flags which confirm that the messages are unicast and fulfill the requirements of the project.