Project 1. Smart Bracelets

You are requested to design, implement and test a software prototype for a smart bracelet. The bracelet is worn by a child and her/his parent to keep track of the child's position and trigger alerts when a child goes too far. These bracelets are becoming more and more popular, with some commercially available prototypes on the market.

The operation of the smart bracelet couple is as follows:

1. **Pairing phase**: at startup, the parent's bracelet and the child's bracelet broadcast a 20-char random key used to uniquely couple the two devices. The same random key is pre-loaded at production time on the two devices: upon reception of a random key, a device checks whether the received random key is equal to the stored one; if yes, it stores the address of the source device in memory. Then, a special message is transmitted (in unicast) to the source device to stop the pairing phase and move to the next step.

2. **Operation mode**: in this phase, the parent's bracelet listen for messages on the radio and accepts only messages coming from the child's bracelet. The child's bracelet periodically transmits INFO messages (one message every 10 seconds), containing the position (X, Y) of the child and an estimate of his/her kinematic status (`STANDING`, `WALKING`, `RUNNING`, `FALLING`).

3. **Alert Mode**: upon reception of an INFO message, the parent's bracelet reads the content of the message. If the kinematic status is FALLING, the bracelet sends a FALL alarm, reporting the position (X, Y) of the children. If the parent's bracelet does not receive any message, after one minute from the last received message, a MISSING alarm is sent reporting the last position received.

Continue in the next page with Requirements.

Requirements

1. Implement the prototype with the O.S. of your choice (including application logic, message formats, etc...). The X, Y coordinates can be random numbers, and the kinematic status should be randomly selected according to the following probability distribution:

   $$P(STANDING) = P(WALKING) = P(RUNNING) = 0.3$$

   and

   $$P(FALLING) = 0.1$$

2. Simulate your implementation with 2 couples of bracelets at the same time. Your simulation should demonstrate that your code follows the design requirements. If you simulate using TOSSIM, you can emulate that a node goes out of range by turning it off (i.e., calling `mote.turnOff()` in python). In Cooja, you can simply move a node out of the communication range of the other one.