Time synchronised communication

In this assignment you will use the WSNSimPy simulator to implement a simple star network consisting of a base station and 10 devices, where the communication of the 10 devices to the base station is time synchronised in a schedule.

The base station starts the process of scheduling the network by broadcasting a "Hello" message. It will broadcast the "Hello" message 5s after booting. This is called the "discovery" period.

When the devices receive the "Hello" they will announce their identity to the base station. They must do so in *scatterred manner to avoid collisions*, so each device will take a small random timeout at the start before transmitting their id. The timeout should be less than 100ms.

OBS: Use delayed execution instead of `yield self.sim.timeout` when scatterring the transmission of the device id.

The base station will wait for 5 seconds for device ids. Then it considers the "discovery" period closed and computes the schedule for the devices. As this is a star network the schedule is simple - allocate one slot to each device. We assume that the slot duration is known to both base station and devices and it is 100ms.

The base station then creates a schedule dissemination PDU that contains the following fields

- `num_devices`: number of devices discovered
- `dev_slots`: list of (device-id, slot number), where slot number starts at 0
- `start_delay`: delay before communication can begin.

On receiving the schedule, devices will

- retrieve their slot number from the list
- they will then start transmitting at `delay + slot_number*100ms`.

OBS: Collisions are still possible so some devices may not receive the schedule. In that case devices will not participate in transmitting data.

Before making a transmission devices will schedule the next one (delayed execution) for "(number of devices)*100msaa, which is the length of the schedule frame. It is important to schedule the next transmission at the start of the slot in order to preserve the schedule.

PDU types:

```
* ``BS-HELLO``. No other fields than shown below. Broadcast.
* ``DEV-HELLO``. Content is device id. Unicast
* ``SCHED``. Sent from BS to devices (broadcast). Contents as above.
* ``DATA``. Sent from device to BS (unicast). No other fields than below.
```

All PDUs should have a "type" field with values as above. The type values should be strings. PDUs should also include "source" and "dest" fields set to the id of the source and destination nodes.

Further instructions and tasks

- you will need two classes: one for the base station and another for the generic devices

- set up th network in a `100x100` area with the base station in the middle and devices randomly deployed in the area
- configure the transmission range (`tx_range`) of all devices to 150
- run the simulation for 50 seconds
- after running the simulation print the nuber of collisions observed by the base station with `bs.phy.stat.total_collision`
  - run the simulation several times - do you observe collisions at the base station? do all devices receive their schedule? explain why - if you use the `self.log` function you should see the simulation time for each log, you can use it to determine if collisions are likely happening
- explain how the reliability of the network scheduling process could be improved to make sure that all devices have a schedule.

Sample output

```
Node #1  [   0.00000] Waiting for HELLO
Node #2  [   0.00000] Waiting for HELLO
Node #3  [   0.00000] Waiting for HELLO
Node #4  [   0.00000] Waiting for HELLO
Node #5  [   0.00000] Waiting for HELLO
Node #6  [   0.00000] Waiting for HELLO
Node #7  [   0.00000] Waiting for HELLO
Node #8  [   0.00000] Waiting for HELLO
Node #9  [   0.00000] Waiting for HELLO
Node #10 [   0.00000] Waiting for HELLO
Node #0  [   5.00000] Discovery starts
Node #10 [   5.00051] Received PDU from 0: BS-HELLO
Node #2  [   5.00051] Received PDU from 0: BS-HELLO
Node #8  [   5.00051] Received PDU from 0: BS-HELLO
Node #4  [   5.00051] Received PDU from 0: BS-HELLO
Node #7  [   5.00051] Received PDU from 0: BS-HELLO
Node #6  [   5.00051] Received PDU from 0: BS-HELLO
Node #1  [   5.00051] Received PDU from 0: BS-HELLO
Node #3  [   5.00051] Received PDU from 0: BS-HELLO
Node #5  [   5.00051] Received PDU from 0: BS-HELLO
Node #9  [   5.00051] Received PDU from 0: BS-HELLO
Node #0  [   5.02146] Received PDU from 9: DEV-HELLO
Node #0  [   5.02306] Received PDU from 6: DEV-HELLO
Node #0  [   5.02802] Received PDU from 4: DEV-HELLO
Node #0  [   5.03588] Received PDU from 3: DEV-HELLO
Node #0  [   5.04068] Received PDU from 5: DEV-HELLO
Node #0  [   5.04344] Received PDU from 8: DEV-HELLO
Node #0  [   5.05656] Received PDU from 10: DEV-HELLO
Node #0  [   5.08307] Received PDU from 7: DEV-HELLO
Node #0  [  10.00000] Schedule: [(9, 0), (6, 1), (4, 2), (3, 3), (5, 4), (8, 5), (10, 6), (7, 7)]
Node #10 [  10.00246] Received PDU from 0: SCHED
Node #10 [  10.00246] Transmitting in slot 6
Node #2  [  10.00246] Received PDU from 0: SCHED
Node #2  [  10.00246] Didn't get a slot :(
Node #8  [  10.00246] Received PDU from 0: SCHED
Node #8  [  10.00246] Transmitting in slot 5
Node #4  [  10.00246] Received PDU from 0: SCHED
Node #4  [  10.00246] Transmitting in slot 2
Node #7  [  10.00246] Received PDU from 0: SCHED
Node #7  [  10.00246] Transmitting in slot 7
Node #6  [  10.00246] Received PDU from 0: SCHED
Node #6  [  10.00246] Transmitting in slot 1
Node #1  [  10.00246] Received PDU from 0: SCHED
Node #1  [  10.00246] Didn't get a slot :(
Node #3  [  10.00246] Received PDU from 0: SCHED
Node #3  [  10.00246] Transmitting in slot 3
Node #5  [  10.00246] Received PDU from 0: SCHED
Node #5  [  10.00246] Transmitting in slot 4
Node #9  [  10.00246] Received PDU from 0: SCHED
Node #9  [  10.00246] Transmitting in slot 0
Node #0  [  14.90000] Data period starting
Node #0  [  15.00285] Received PDU from 9: DATA
Node #0  [  15.10285] Received PDU from 6: DATA
Node #0  [  15.20285] Received PDU from 4: DATA
Node #0  [  15.30285] Received PDU from 3: DATA
Node #0  [  15.40285] Received PDU from 5: DATA
Node #0  [  15.50285] Received PDU from 8: DATA
Node #0  [  15.60285] Received PDU from 10: DATA
n1 stats: TX=2 RX=358Collisions=2
```

The assignment is worth 25 points of the final grade.

The MTU plagiarism policy penalises plagiarism (sharing of code) equally for all parties involved.