For all assignments, a TSCH network with variable number of nodes *n* is assumed, where *n-1* send data periodically to the sink. At each step, when recording delays and schedules, capture screenshots to document the values to refer later, when answering the questions.

Assignment 1. TSCH-Minimal

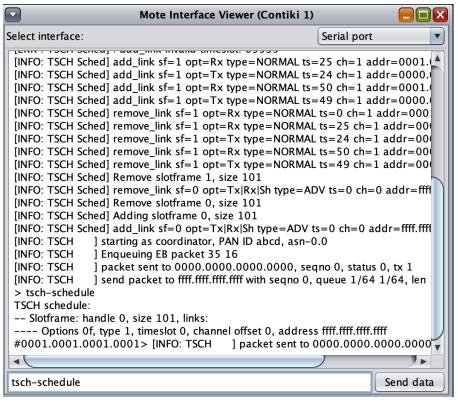
Step 0. Joining process

Using "examples/tsch-simple/tsch-simple.csc" scenario, run the simulation with two nodes (a sink and a transmitter mote). Pause the simulation and read the log file "Mote output" from the beginning. Figure out how node 1 and node 2 join the network.

HINT: Filter for "TSCH" in "Mote output" and understand the joining process.

HINT: Only SF0 is used in this setup. Ignore logs related to SF1.

Continue the simulation till you see that packets are being received by the sink (node 1) and pause again. Check local schedules of each node using `tsch-schedule` command in the respective serial port, see screenshot below. Enter the command on both interfaces, continue the simulation for a second to execute the command, and then stop again to check the output. Identify which cells are used for communication between two nodes.



Step 1. Two-node network

Continue the simulation and record the per-packet delays as well as the average end-to-end delays over multiple packets (please also note them down for further assignments). Are the results expected?

HINT: Filter for "node 2 with delay" in "Mote output" to see delay of each packet.

HINT: Once the required number of packets (50 by default, see `MIN_PACKETS` in `sink.c`) from the leaf node (i.e., node 2) are received by the sink, the average delay is printed in the log.

Step 2. Star topology

Open the "star.csc" scenario with a network configured in a star topology with 3 nodes. Record average end-to-end delays of each node and compare the results to the linear network from step 1.

Assignment 1. Questions

1. Identify the following parameters used in the 2-node scenario.

Question	node 1 (sink)	node 2
Channel used when booting up		
Hopping sequence length		
Slotframe size		

Question	node 1 (sink)	node 2
Starts as a coordinator		
Starts sending Enhanced Beacons (EBs)		
PAN ID		
Minimal cell location (SF0)		
MAC address at the minimal cell		
Channels scanned before joining	n/a	
EB size		
First EB received on channel	n/a	

2. What is the slotframe duration in seconds?

HINT: Time slot duration is 10 ms (see `os/net/mac/tsch/tsch-timeslot-timing.c`)

3. What is the average packet arrival rate per slotframe at node 2?

HINT: Packet sending rate from each mote is `SEND_INTERVAL (2 * CLOCK_SECOND)` (see `mote.c`)

4. What kind of traffic is sent over the minimal cell?
5. Why does the delay per packet vary so much in step 1?

HINT: Packets are generated with jitter of one SF (see `mote.c`)

6. Why is the average delay per node higher in star topology compared to the linear topology from step 1?

Assignment 2. TSCH-Dedicated

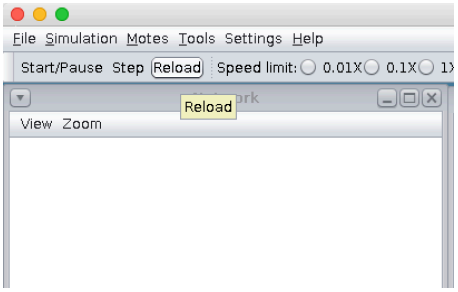
Step 1. Switching to dedicated cells

Set these flags in `project-conf.h` to 0:

```
#define TSCH_SCHEDULE_CONF_WITH_6TISCH_MINIMAL 0
#define TSCH_CONF_INIT_SCHEDULE_FROM_EB 0
```

Run the simulation for a linear network from step 1 of the first assignment and record the TSCH schedules and the average delays. Discuss with your team partner to understand the type and number of cells scheduled.

HINT: Once you modify `project-conf.h`, delete the "build" folder from "tsch-simple" directory and reload "tsch-simple.csc" scenario to apply changes.



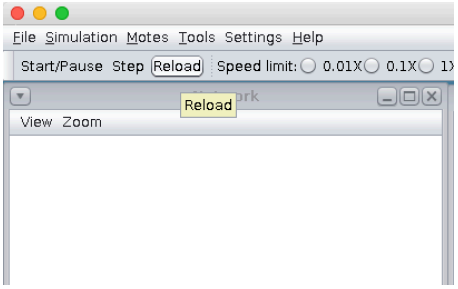
Question	node 1 (sink)	node 2
List TX cells (option 01)		
List RX cells (option 02)		

Step 2. Adding more dedicated cells

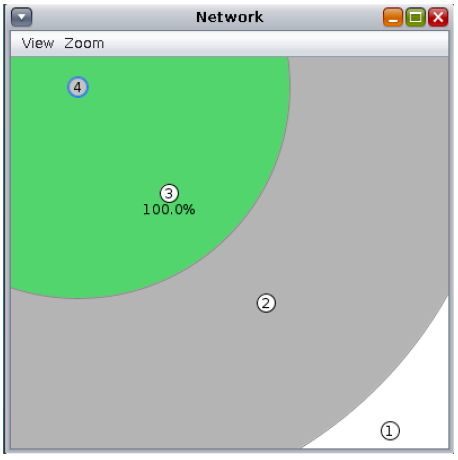
Increase `NUM_CELLS_PER_LINK` in `common.h` to 5 and check the average delay for each node, comparing the result to the previous step. Delete the "build" folder and reload "tsch-simple.csc" scenario to apply changes.

HINT: Use `tsch-schedule` command to verify that the number of cells scheduled is exactly 5 both at the sink and node 2.

Now add 3 more motes and compare the average delays of each node.



HINT: Increase the number of nodes and align them in a linear network (max 5 nodes), so that each node has only two neighbors. Ensure node 1 is the sink and arrange the rest of nodes as shown in the example. Record average end-to-end delays of packets from each node at the sink and compare the results to the previous task. Discuss with your team partner the end-to-end delay of the leaf node compare to nodes closer to the sink.



Step 3. Scheduling approach

Check `initialize_tsch_schedule()` in `common.h` and understand how the cells are scheduled between each pair of nodes.

Assignment 2. Questions

1. When adding dedicated cells, what is the factor of reduction of average delay? Explain the reasons for this.
2. Explain reasons for delay differences you notice per node in a linear network. To improve average end-to-end delays, is it better to add more cells at nodes closer to the sink or those further away? Justify your answer.
3. What are the pros and cons of the scheduling approach used in this example?

Assignment 3. 6top Protocol (6P)

For these tasks use "tsch-sixtop\tsch-sixtop.csc" scenario.

Step 1. Understanding 2-step transaction

Run the simulation with two nodes (sink and mote) and explain the process of adding a cell with 6P. Record the delays and compare to the first task of the previous section.

HINT: filter the contents of the mote output window by using queries like `ID:[1]` or `ID:[1,2]` to limit the output with respect to specified nodes only. Look for `sf-simple: Send a 6P Add Request` .

Check the schedule of node 2 with `tsch-schedule` and verify cell allocation.

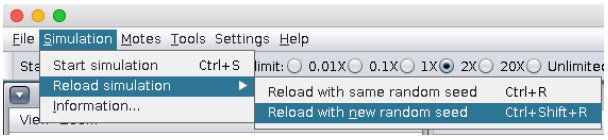
Step 2. Adding more cells

In `mote.c` set `NUM_CELLS_TO_ADD` to 3 and check the schedule of node 2 with `tsch-schedule` .

HINT: Reload the simulation when changing `NUM_CELLS_TO_ADD` and run for about 60 s, then pause and check with `tsch-schedule` .

Step 3. Linear topology and cell allocations

Now add 3 more motes to create a linear topology following the approach from step 2 in "TSCH-Dedicated". Set `TSCH_SCHEDULE_CONF_DEFAULT_LENGTH` to 101 in `project-conf.h` and `NUM_CELLS_TO_ADD` to 3. Run the simulation until all motes have added the cells (several simulated minutes). Check cell locations and compare them to step 2 from "TSCH-Dedicated". Reload the simulation with a different seed and note cell locations again.



Note down TX and RX cells of each node across different repetitions in the table below.

Repetitions	node 1 (sink)	node 2	node 3
1			
2			
3			

Step 4. Star topology [optional]

Open the "sixtop-star" scenario, set `NUM_CELLS_TO_ADD` to 3 and `TSCH_SCHEDULE_CONF_DEFAULT_LENGTH` to 11 in `project-conf.h` . Run the simulation for at least 4 (simulated) minutes and check the schedule at the sink, discuss your observations.

Explain the difference of `initialize_tsch_schedule()` from `common.h` to the "TSCH-Minimal" scenario.

Assignment 3. Questions

1. What are the cells allocated by 6P transactions during step 1 for the communication between node 1 and node 2?
2. Is there an upper limit on `NUM_CELLS_TO_ADD` ? Justify your answer.
3. Why do cell locations change when simulation is reloaded with different seeds?