Kiel University
Christian-Albrechts-Universität zu Kiel

Distributed Systems
Research Group

IoT Lab 2

Multi-hop Dissemination

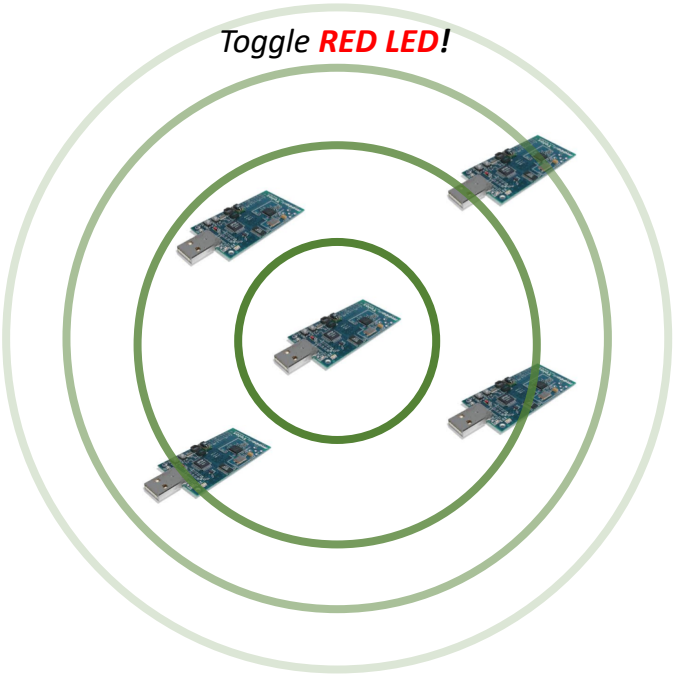
Overview

- Basic concept
- Setup
- Tasks
- Hints
- Your deliverables

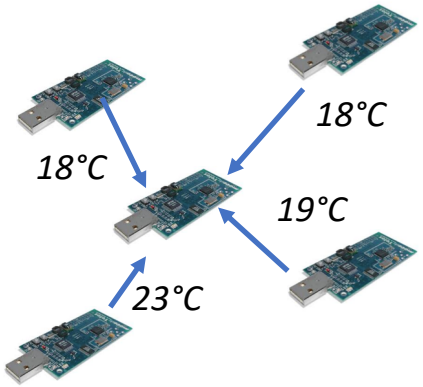
Basic Concept

Summary of Lab 1

- In the previous lab, you saw two fundamental communication models:
  - One-to-all / Dissemination
    - delivering commands
    - Network-wide configuration/code update
  - All-to-one / Collection
    - collecting sensor data to a gateway
    - Alarm detection



- However, we limited ourselves to one-hop
  - Everybody can contact everybody else
  - Many deployments are multi-hop!
    - Ex: Avalanche detection, wildlife monitoring, large factories

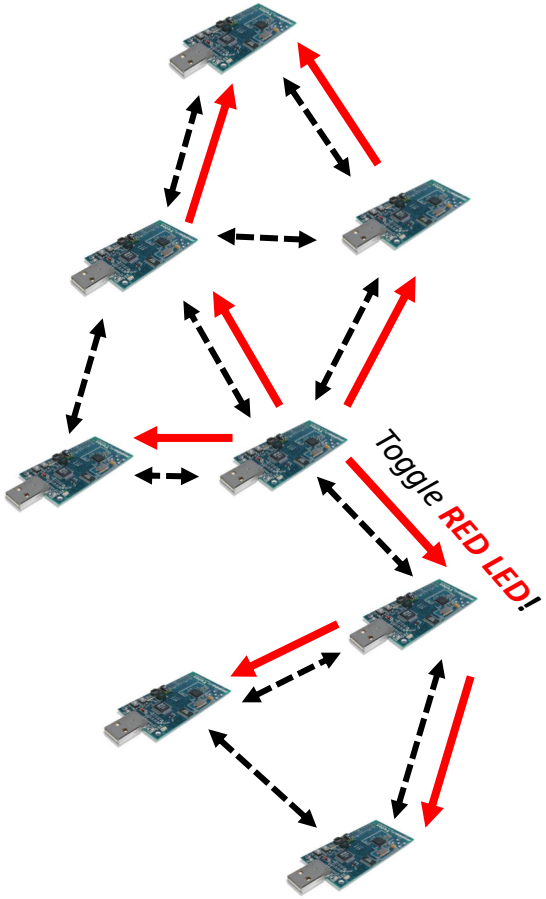


Lab 2: basic concept

- This lab will focus on **Multi-hop dissemination**
  - The **central entity** can not reach all nodes
  - Some nodes must act as relays

- How to do efficiently?
  - If we don't know the topology…
    - We can't use unicast!
  - Using broadcasts might lead to a **broadcast storm!**

- We need to evaluate our protocol
  - Reliability: how many nodes successfully received the message
  - Latency: How long did it take?

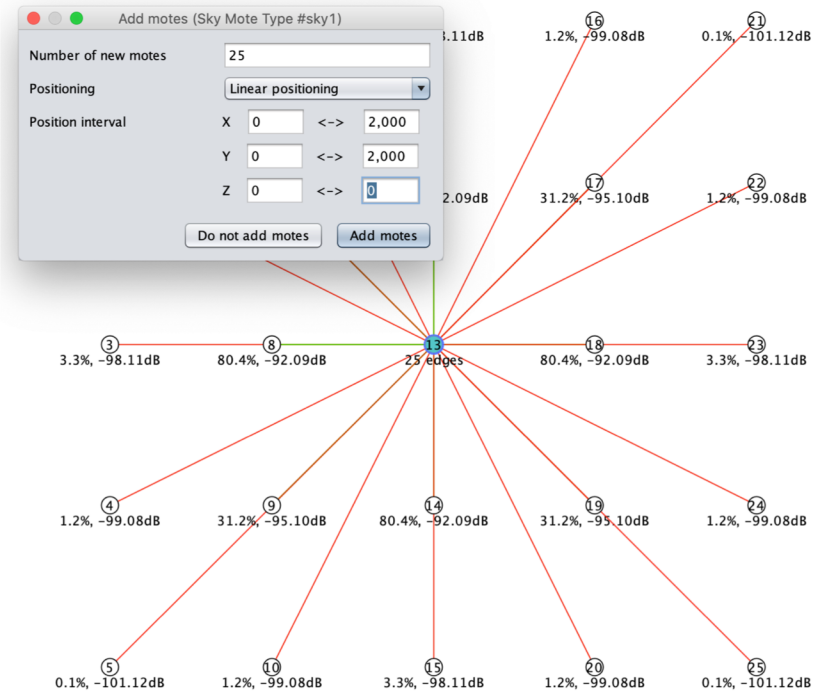


Setup

Initial Setup

- Create a new Cooja simulation
  - **Multi-path Ray-tracer Medium (MRM)**
  - **25** Sky motes
  - **Linear positioning**
    - X: 0-2000
    - Y: 0-2000

- By clicking on nodes, you should see that the probability of receiving messages depends on the distance (+ message collision)



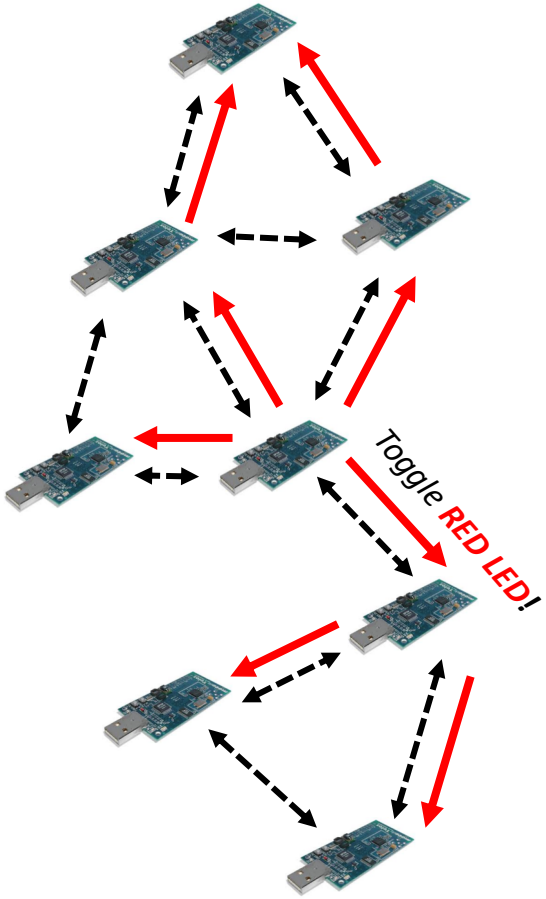
Tasks

What you must do

Part 1 – Command dissemination

25 nodes: one central node (ID 13), 24 ”slave” nodes (ID 1-12 and 14-25)

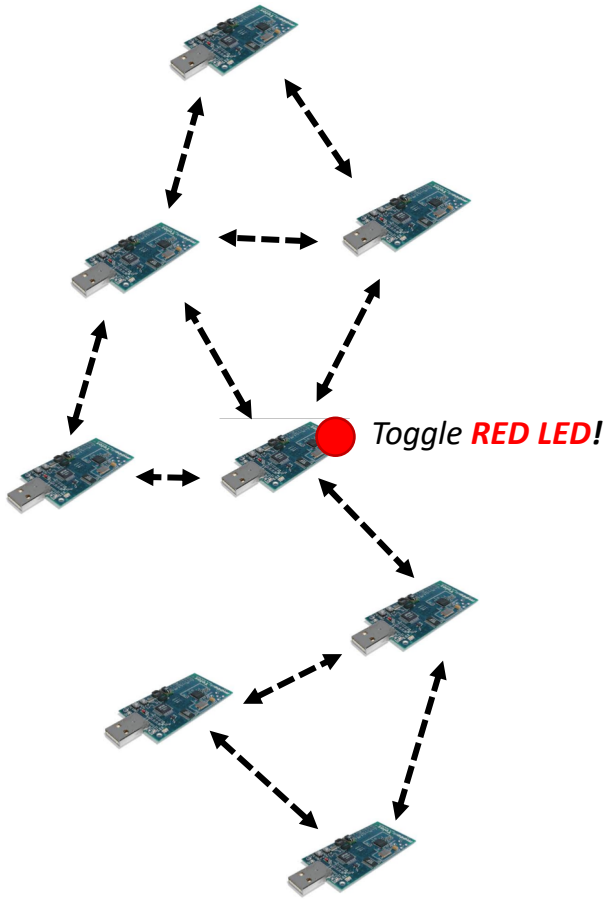
- If you followed the setup, ID 13 should be at the center, and have a 80% probability to communicate with the nearest nodes



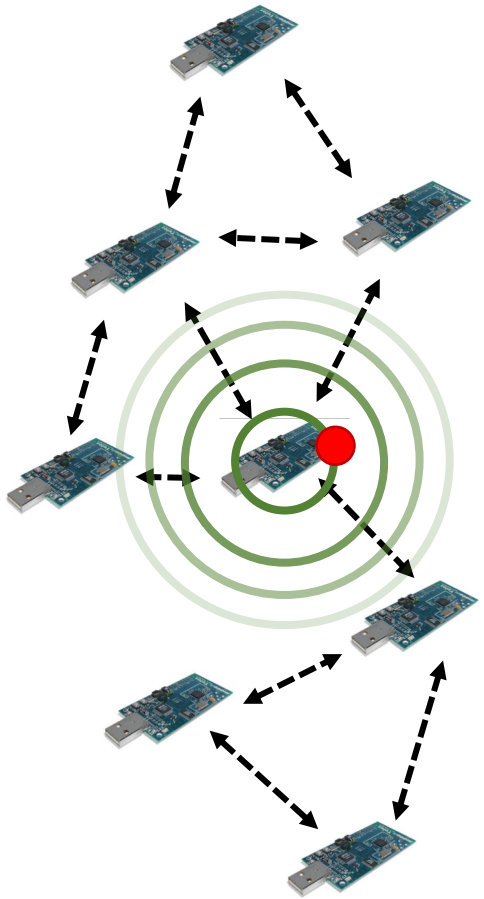
Every minute:

- The central node selects a random LED
- Sends a ”Toggle LED” command to the network
- Upon reception, a slave toggles the associated LED

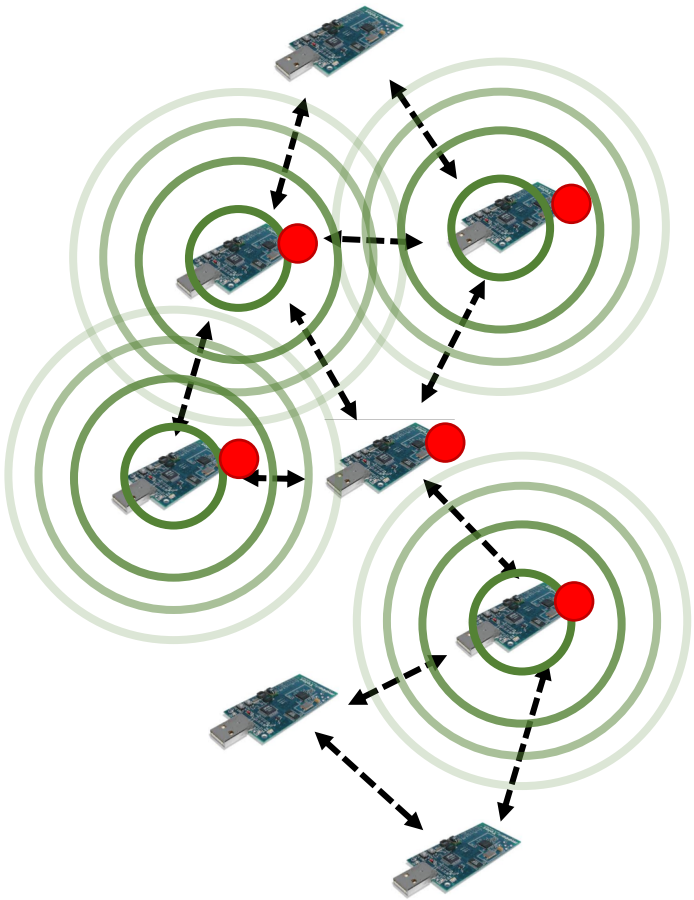
Example



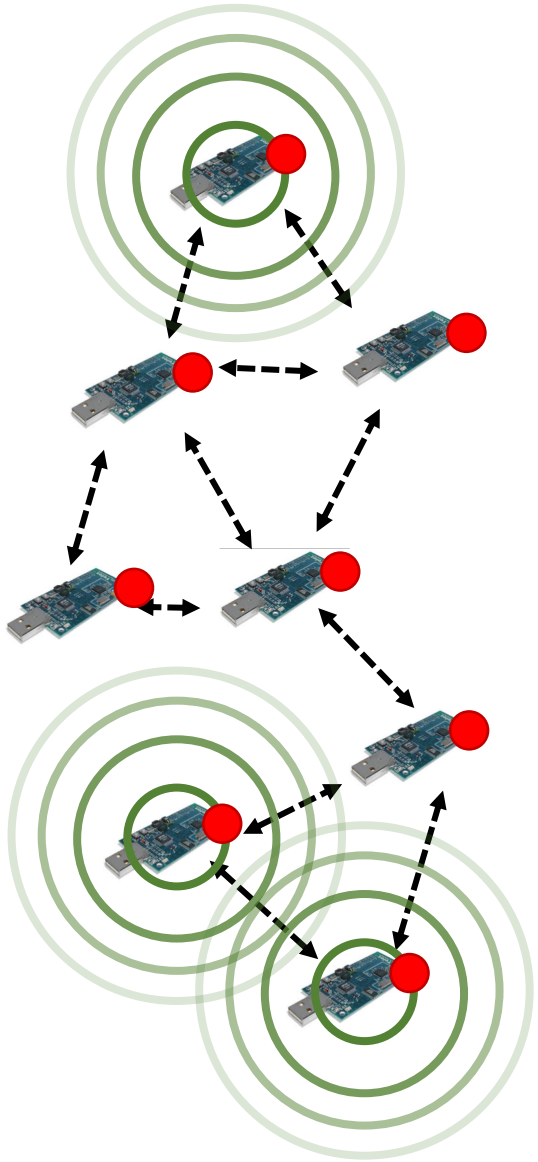
Example



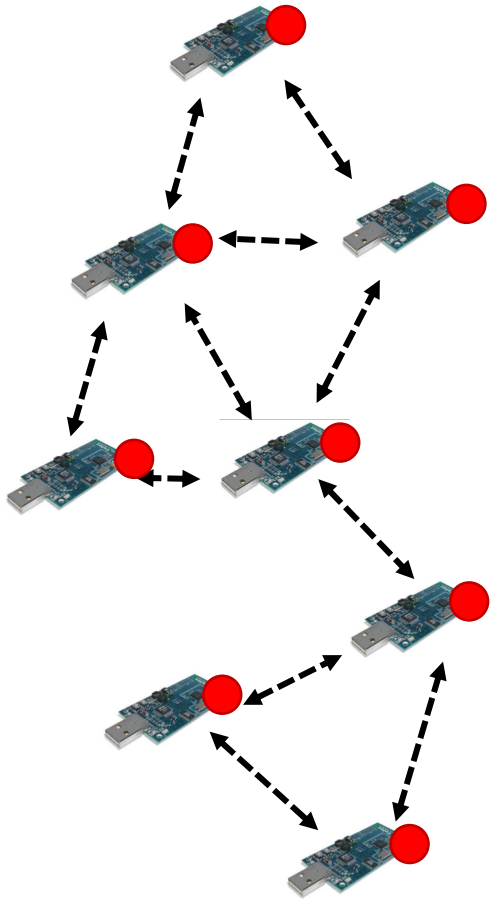
Example



Example



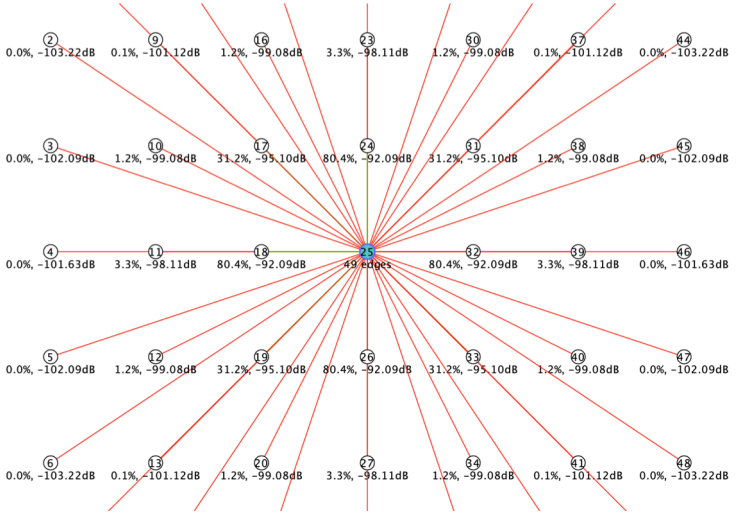
Example



Part 2 – Evaluation (1)

You will evaluate the reliability and latency of your solution, for networks with increasing diameter (= number of hops)

- Network sizes:
  - 9 nodes (linear, max: 1200, central node: 5)
  - 25 nodes (linear, max: 2000, central node: 13)
  - 49 nodes (linear, max: 2800, central node: 25)



- What to evaluate?
  - Latency: for one command, time until the last node received the command
  - Reliability: for one command, how many nodes received it?

Part 2 – Evaluation (2)

- What is a (good) evaluation?
  - A figure/plot, not just numbers thrown around
  - Should be representative of a typical execution
  - Should present the statistical aspect of running the protocol more than once (what is the mean, standard deviation or quartiles ?)
  - You should run multiple rounds and present us your methodology!!!

- How to collect the data?
  - Use printf() or LOG_INFO()
  - Cooja's mote output can be exported as .txt file (Mote output: File->save to file)
  - The output file will contain cooja's global time

Hints

Some hints

- Flooding (a succession of broadcasts) usually work well

- How to avoid broadcast storms?
  - Broadcast storms occur when a node starts a broadcast, which is in turn broadcasted by a neighbor. Then, the initial node transmits again in hope of helping the dissemination.
  - Using a unique (monitically increasing) sequence number helps!

- How to send multiple data in one packet?
  - Use a struct

You must submit

- A **video** presenting your solutions:
  - 5 to 8 min (we remove points if the video is longer than 8 min!)
  - The video must contain:
    - A demo of your solution in cooja
    - An explanation of:
      - your design choices
      - your code
      - possible corner cases (what could break your solution)?
    - **Your evaluation**
      - **Plots/figures (with axes information, how many runs were used per datapoints, etc.)**
      - **explanation of the main findings**
- Your **code**
  - Any file you might have modified
- Your **figures/plots**

How to submit?

- Upload your source code, plots and video as an archive (zip, rar) on **iLearn**
  - iLearn limits the file size to ~20Mb, if your file is too large, upload the source code and plots only on iLearn, and the video on a dropbox link given in iLearn
  - Please note that the dropbox repository is shared with everybody, so you must limit your video size to **50Mb!!!**
  - **Additionaly, you can now submit the VIDEO as a dropbox link (using your own account, no size limit) or on Youtube.**

- Like the prelab, you will work as a group (2 students), if you submit alone you need permission from us

Good luck!

Deadline: Monday, 11th May

To go further

- Tseng, Y., Ni, S., Chen, Y. *et al.* ”The Broadcast Storm Problem in a Mobile Ad Hoc Network”
- Ad-hoc On-demand Distance Vector Routing (AODV), https://www.cs.jhu.edu/~cs647/aodv.pdf
- Philip Levis, Neil Patel, David Culler, and Scott Shenker, "Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks”
- J. Lu and K. Whitehouse, "Flash Flooding: Exploiting the Capture Effect for Rapid Flooding in Wireless Sensor Networks"
- F. Ferrari, M. Zimmerling, L. Thiele and O. Saukh, "Efficient network flooding and time synchronization with Glossy”