Kiel University
Christian-Albrechts-Universität zu Kiel

Distributed Systems
Research Group

IoT Lab 3

Multi-hop Collection

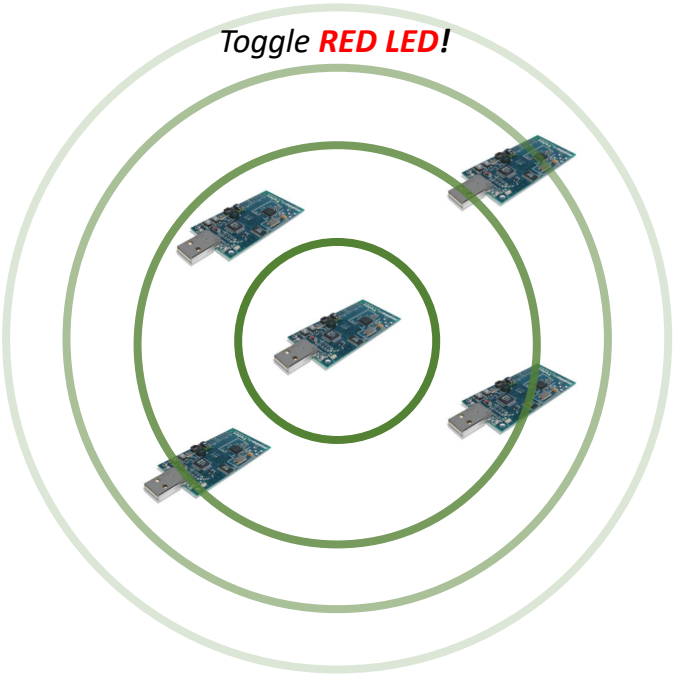
Overview

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

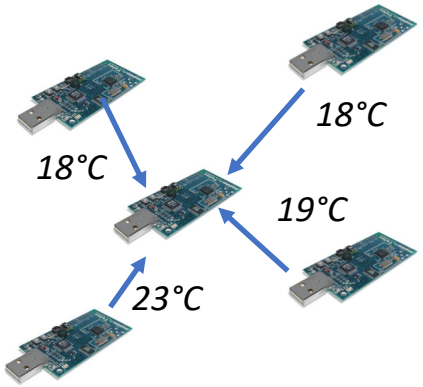
Basic Concept

Summary of Lab 1

- In the first 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

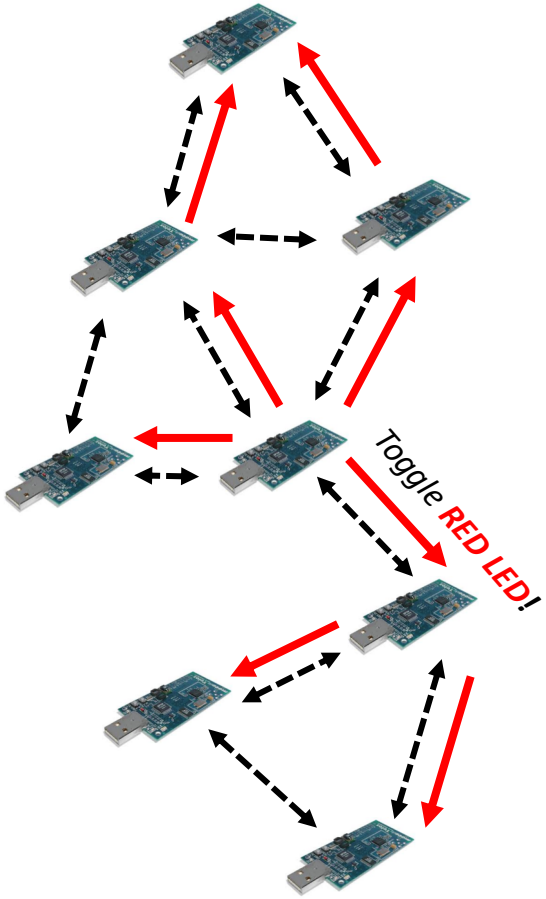


- However, we limited ourselves to one-hop
  - Everybody can contact everybody else
  - Many deployments are multi-hop!
    - Ex: Avalanche detection, wildlife monitoring, large factories



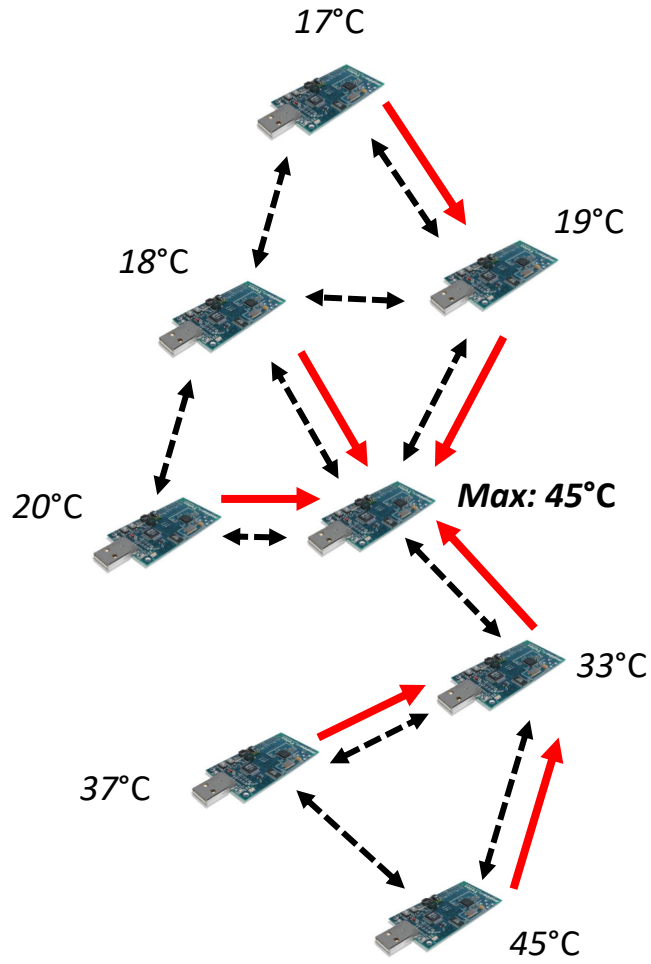
Summary of Lab 2

- Lab 2 introduced multi-hop topologies and you have implemented a multi-hop dissemination solution
  - Multi-hop: a node can not reach all other nodes of the network
- Multi-hop dissemination is used to:
  - Disseminate commands, new software, control packets
  - Build routes in AODV
  - Synchronize clocks, etc.
- One simple solution?
  - Flooding using broadcasts!



Lab 3: basic concept

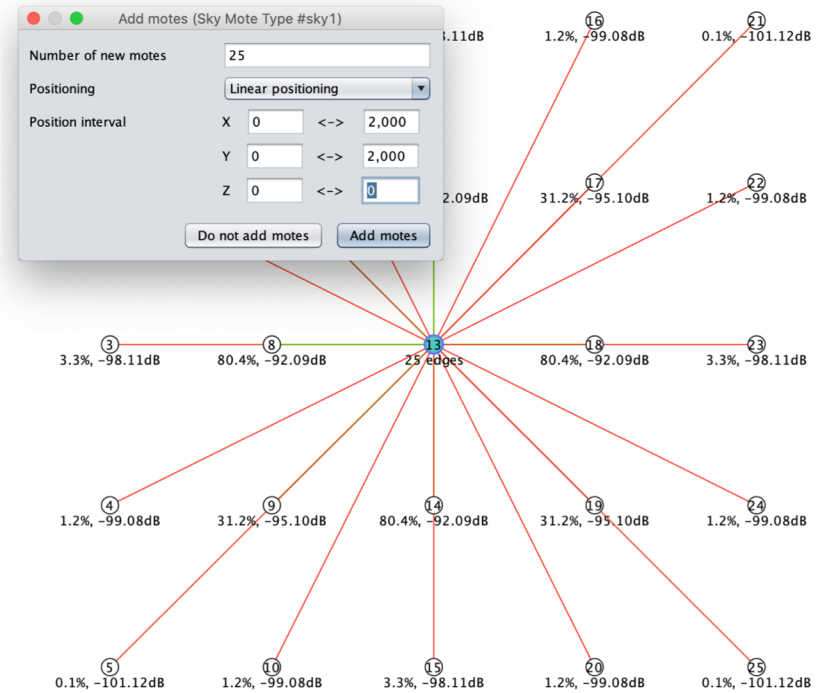
- This lab will focus on **Multi-hop collection**
  - Again, a **central entity** can not reach all nodes and must rely on relays
- What is the goal?
  - **Collect** the **maximum** temperature recorded by a network of sensors
  - Be as efficient as possible
- Again, we will evaluate our protocol



Setup

Initial Setup

- Create a new Cooja simulation
  - **Multi-path Ray-tracer Medium (MRM)**
  - **Mote startup delay: 0 ms**
  - **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)

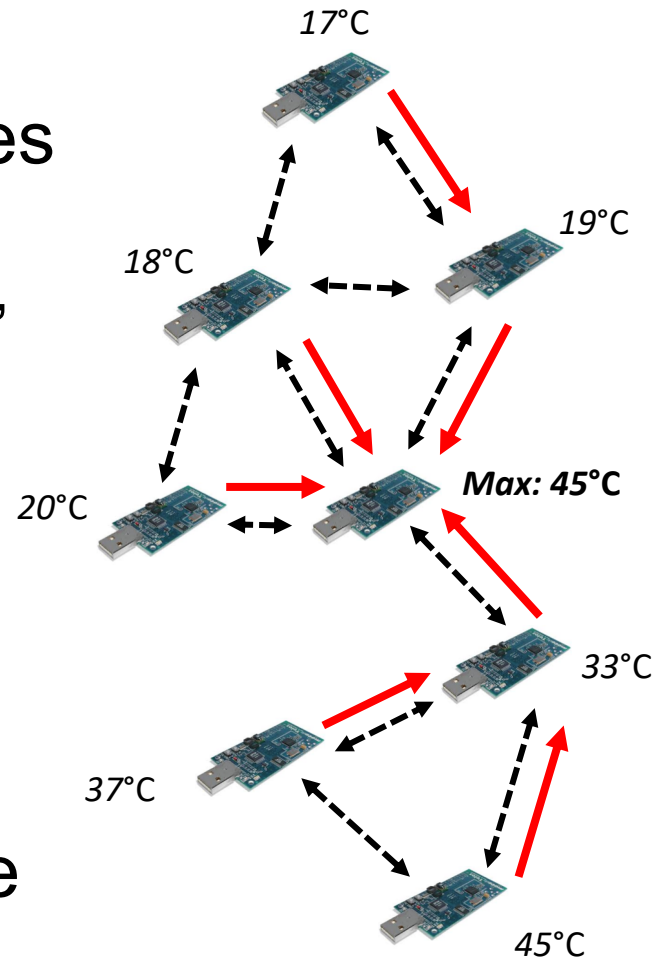
Tasks

What you must do

Part 1 – Temperature collection

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

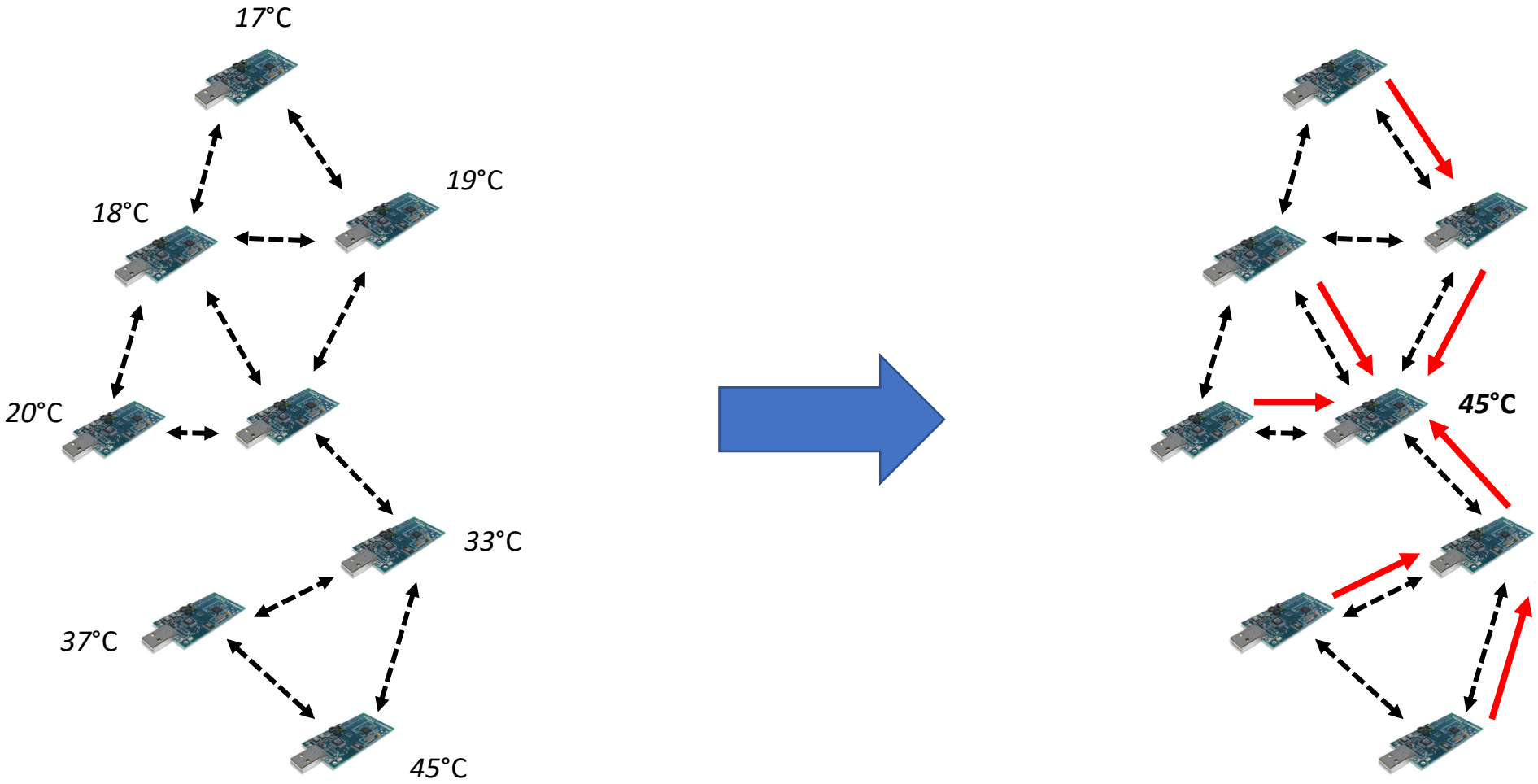
- If you followed the setup, ID 13 should be at the center, and have a 80% probability to communicate with the nearest nodes



Every 2 minutes:

- Slave nodes generate a **random** temperature between -100 and 100 degrees
- The temperature is propagated to the central node
- The central node displays the **maximum** temperature received
  - Unlike Lab 1, the central node does not have to print all received values

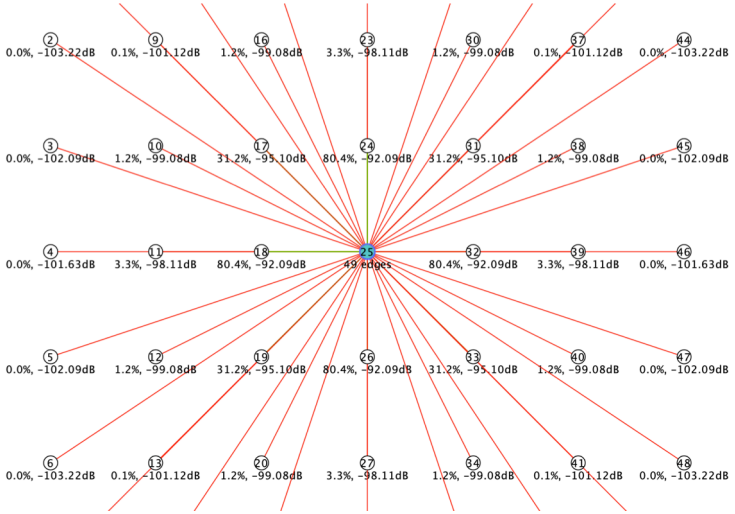
Example



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

- How to solve this lab?
  - There are many approaches possible for part 1, it is up to you how to design your preferred solution
  - Some common ways include:
    - Build a routing tree, and then use unicasts
    - Use flooding for each value
    - Use in-network processing (i.e., aggregate data along the way)
  - We give you additional reading materials that you can get inspiration from

- How to avoid the initial random initialization for part 2?
  - When creating the simulation, set the *mote startup delay* to 0 ms

Your deliverables

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 video does not have to be perfect, don't spend too much time editing, a few pauses and hesitations are perfectly normal
- 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, 25th May

To go further

- *O. Gnawali, R. Fonseca, et al.* ”Collection tree protocol”. 2009
- RFC 6550, RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, 2012
- *F. Ferrari, M. Zimmerling, et al.* ”Low-power wireless bus”. 2012
- *O. Landsiedel, F. Ferrari, et al.* ”Chaos: versatile and efficient all-to-all data sharing and in-network processing at scale”. 2013