

Loose notes and random ideas Alternating Bit Protocol Unit Testing TinyOS  
Cooja Combination TinyOS and Cooja LibReplay Eclipse: Yeti Github

Gebruikte tools: Sublime Atom (en waarom niet VIM), Cooja, telosb motes, ...

Wat is moeilijkheid ontwerpen algoritme TinyOs (distributed networking algemeen): klassiek debugging, op F7 drukken en lijn per lijn afgaan, werkt niet. Speciale debugging tools nodig.

Context vermelden: onderzoek paper gesubsidieerd via KARYON project; wat is het, doel, welk puzzelstukje is dit algoritme en resultaten in het grotere geheel?

Acknowledge problems that arose during the work: no or little support to be found for TinyOS. E.g.: I can add this in the part about future work → in presentation of current state of TinyOS by Phil Levis (Lessons Learned from 10 years of TinyOS development), he mentions the general move away from TinyOS to more accessible and supported OS' such as Contiki. This leads to the conclusion that future work will have to convert my code to Contiki-compatible code. ALSO, on tinyos wiki itself found this:

“Generally, the best place to start is with the authors of the paper describing it. Even if they don’t maintain an implementation, chances are they are aware of existing versions of it. Emailing tinyos-help isn’t usually effective.” [protocolhelp][[http://tinyos.stanford.edu/tinyos-wiki/index.php/FAQ#I\\_am\\_looking\\_for\\_code\\_for\\_protocol\\_or\\_system\\_.22X.22:howdo\\_I\\_find\\_it.3F](http://tinyos.stanford.edu/tinyos-wiki/index.php/FAQ#I_am_looking_for_code_for_protocol_or_system_.22X.22:howdo_I_find_it.3F)]

Contiki vs TinyOS: [contikivstinyos](#)

Central research question.

Thesis is an original contribution to knowledge: You have identified a worthwhile problem or question which has not been previously answered You have solved the problem or answered the question

Keep to the point A concise paper or thesis requires keeping the main points in mind—ONLY include background information, data, discussion that is relevant to these points

You may develop computer programs, prototypes, or other tools as a means of proving your points, but remember, the thesis is not about the tool, it is about the contribution to knowledge

Why using TinyOS and not Contiki for example? Ask Elad why he suggested TinyOS.

In original paper of algorithm are a few “lemma’s” described, which form the proof for some of the statements made concerning the advantages of the algorithm. In “expected results” section we can use these lemma’s to form an idea of the results we’re expecting to see out of the simulations and experiments.

Researchers describe WSNs as

“An exciting emerging domain of deeply networked systems of low-power wireless motes with a tiny amount of CPU and memory, and large federated networks for high-resolution sensing of the environment.” [M. Welsh, D. Malan, B. Duncan, T. Fulford-Jones, S. Moulton, “Wireless Sensor Networks for Emergency Medical Care,” presented at GE Global Research Conference, Harvard University and Boston University School of Medicine, Boston, MA, Mar. 8, 2004.]<sup>11</sup>

Thesis Outline

## **1. Abstract**

## **2. Introduction**

## **3. Background and Literature review**

### **3.1 Current state of the art**

### **3.2 Theory (explaining the theoretical working of the algorithm according to the paper given)**

## **4. Problem statement/research question**

### **4.1 Assignment**

### **4.2 Goals (~targets, objectives)**

- Implement algorithm, full freedom to make own design
- Benchmark the algorithm by performing simulations and establishing performance
- Optimize the algorithm using the data acquired by the simulations
- Provide proof of self-stabilization concept by using LibReplay to insert, duplicate and reorder packets during communication
- Write report, the goal is here to make sure other researchers can continue my work in case they want to implement the algorithm in a bigger application.

### **4.3 Challenges**

Debugging TinyOS applications: “Bug hunting in sensor networks is challenging: Bugs are often prompted by a particular, complex concatenation of events. Moreover, dynamic interactions between nodes and with the environment make it time-consuming to track and reproduce a bug.” Although LibReplay (see Methods section) offers the ability to debug sensor network applications like in sequential programming environments, it is still not feasible to do this consistently for a lot of bugs during development. To use LibReplay for even the smallest

bugs would require a lot of time. All this means that I had to search for other debugging methods during my work, such as responding to the compiler output and using printf statements. [libreplay](#)

TinyOS documentation often contradicting because of different versions used, and the up-to-date information is for a large part spread out over the internet in the form of GitHub issues and commits, TinyOS-help mailing list, etc.

Abandoned TinyOS support “Generally, the best place to start is with the authors of the paper describing it. Even if they don’t maintain an implementation, chances are they are aware of existing versions of it. Emailing tinyos-help isn’t usually effective.” [protocolhelp][[http://tinyos.stanford.edu/tinyos-wiki/index.php/FAQ#I\\_am\\_looking\\_for\\_code\\_for\\_protocol\\_or\\_system\\_.22X.22:howdo\\_I\\_find\\_it.3F](http://tinyos.stanford.edu/tinyos-wiki/index.php/FAQ#I_am_looking_for_code_for_protocol_or_system_.22X.22:howdo_I_find_it.3F)]

## **5. Methods (Describing How You Solved the Problem or Answered the Question)**

## **6. Results**

### **6.1 Data**

### **6.2 Interpretation: analysis of results**

## **7. Conclusion**

## **8. Future work/Recommendations**

Switch to Contiki

## **9. Acknowledgements**

Elad, Olaf Daniel, David, Henning, Robin

KU Leuven Peter Karsmakers, Patrick Coleman, Hilde Lauwereys, Patricia Van Genechten, Isabelle Moons

## **10. References**

## **11. Appendices**