

TinyOS 2.1: Adding Threads and Memory Protection to TinyOS

The TinyOS Alliance

(Including all members of the TinyOS 2.x Working Groups)

<http://www.tinyos.net>



TinyOS 2.1 is the next stage in the evolution of TinyOS. It takes a step towards easier and more robust application development. The most notable features include:

- **TOSThreads**: A fully preemptive application-level threads library that preserves the time-sensitive aspects of TinyOS.
- **Safe TinyOS**: A runtime memory protection service with memory safety checks.
- **Other additions**: 4-bit link estimator, FTSP, IRIS and SHIMMER support, DIP, and optional 802.15.4-compliant MAC layer.

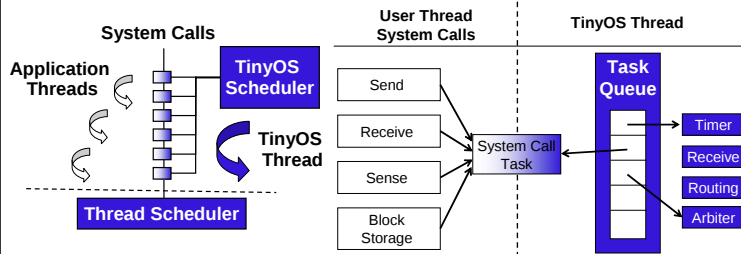
TOSThreads

Kevin Klues, Chieh-Jan Mike Liang, Jeongyeup Paek, Răzvan Musăloiu-E., Ramesh Govindan, Andreas Terzis, Philip Levis

Problem

Given motes' resource constraints, an event-based OS permits greater concurrency. However, preemptive threads offer an intuitive programming paradigm.

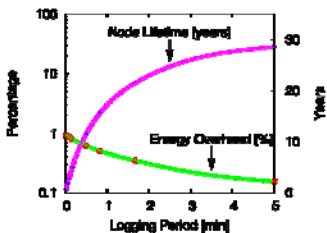
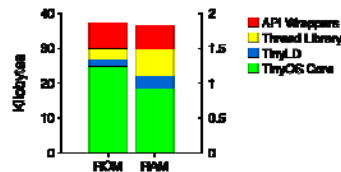
Solution: TOSThreads



TOSThreads allows fully preemptive application threads to run concurrently, making blocking calls to a single higher priority TinyOS kernel thread. Message-passing threading model does not sacrifice the underlying TinyOS event-driven model.

Evaluation

Leveraging the user/kernel boundary, the **TinyLD** component dynamically links applications in the *MicroExe* format to a static kernel.



TOSThreads context switches and system calls introduce an overhead of less than 0.92%. TinyLD requires less than 90ms on a representative sensing application.

TOSThreads has been used in various projects:

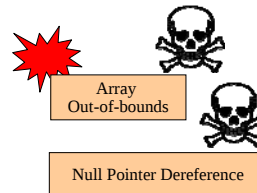
- ① **Latte**, Johns Hopkins University.
- ② **Tenet**, University of Southern California.
- ③ **MAMMARK**, University of California, Santa Cruz.
- ④ **SPINE**, Telecom Italia.

Safe TinyOS

John Regehr, Eric Eide, Nathan Coopride, Will Archer, Yang Chen, David Gay

Problem

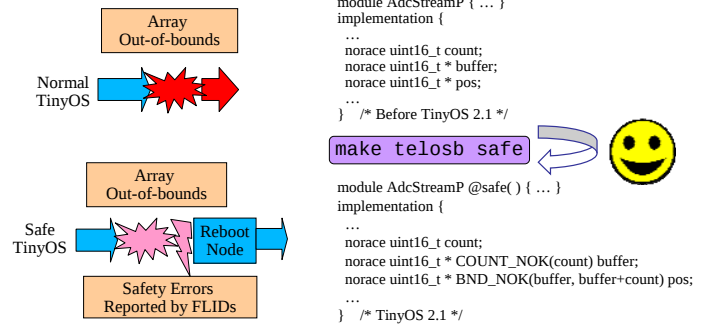
- nesC is not type or memory safe
- Motes lack hardware-based memory protection
- So...
- Pointer and array errors lead to memory corruption
- Symptoms: Motes act flaky, drop out of the network, etc.



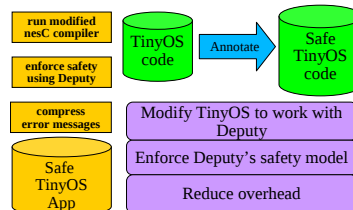
Goals:

- Trap all pointer and array errors
- Provide useful diagnostics
- Provide recovery strategies

Solution: Safe TinyOS



Implementation



Exploit Deputy:

Deputy is a source-to-source compiler for ensuring type and memory safety for C code. Code compiled by Deputy relies on a mix of compile- and runtime checks to ensure that these annotations are respected, and hence that type and memory safety are respected.

Conclusion

Our effort shows that Safe TinyOS is a practical system for the development of reliable sensor network software. Platforms currently supported: Mica2, Micaz and TelosB. Visit our webpage for more information:

<http://www.cs.utah.edu/~coop/safetinyos>

Other Additions

• Collection Tree Protocol (CTP) with the new **4-Bit link estimator**: Rodrigo Fonseca, Omprakash Gnawali, Kyle Jamieson and Philip Levis.

Flooding Time Synchronization

Protocol (FTSP): Miklós Maróti, Branislav Kusz, Gyula Simon and Ákos Lédeczi.

• Two new platforms: **IRIS** (Crossbow Inc.), and **SHIMMER** (Harvard University and Intel Corporation).

• A dissemination protocol that scales to hundreds of values, called **DIP**: Kaisen Lin and Philip Levis.

• **Optional 802.15.4-compliant MAC layer**: Gabriel Montenegro, Nandakishore Kushalnagar, Jonathan Hui and David Culler.

Acknowledgments

The institutions involved (in no particular order) include Vanderbilt University, Johns Hopkins University, Stanford University, UC Berkeley, UCLA, USC, TU Berlin, Harvard University, University of Szeged, MIT, University of Copenhagen, ETH Zürich, EPFL, University of Utah, Rincon Research Inc., Intel Research, Crossbow Inc., and Arch Rock Co. We thank the greater TinyOS community for their valuable feedback.