Leveraging IP for Sensor Network Deployment

Simon Duquennoy, Niklas Wirström,
Nicolas Tsiftes, Adam Dunkels

IP+SN Workshop, Chicago, IL
April 11th 2011

Incremental Sensornet Deployment

Target deployments

- Heterogeneous networks (applications, hardware)
- Need for individual configuration (location, settings)
- Need for individual programming (application)

Usual deployment techniques

- Network-wide programming
- Based on dedicated, inflexible, error-prone solutions
- Many failures in practice

Incremental Sensornet Deployment

Target deployments

- Heterogeneous networks (applications, hardware)
- Need for individual configuration (location, settings)
- Need for individual programming (application)

Usual deployment techniques

- Network-wide programming
- Based on dedicated, inflexible, error-prone solutions
- Many failures in practice

What about using low-power IP?

IP-Based Deployment

Leveraging standards for robustness

- Addressing with IPv6
- Routing with RPL
- Transport with CoAP/UDP or TCP

Network layer interoperability

- All nodes able to route IPv6 traffic
- Applications built on top of IP
- Configuration built on top of IP
- Deployment built on top of IP

Methodology

Steps

- Implementation
- Simulation (based on motes emulation)
- Feasibility study: RPL, CoAP, ...
- Optimizations

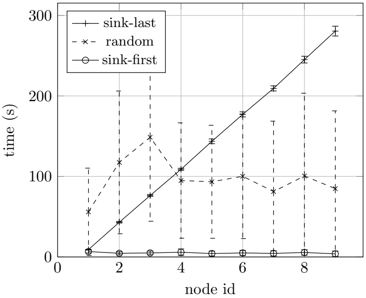
Implementation in Contiki

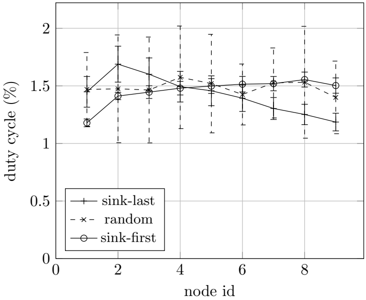
- uIPv6
- ContikiRPL
- CoAP and TCP-based deployment

Typical deployment

1. The node integrates the RPL network
2. The node is configured (location, task, settings)
3. The node downloads a program from a server

Deployment with RPL

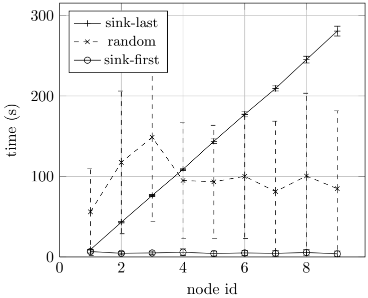


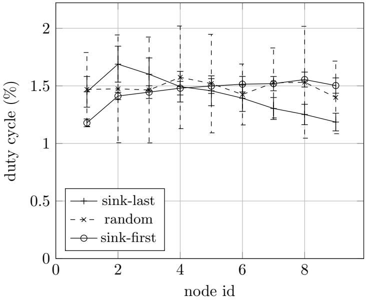


Impact of the deployment order

- Time dominated by installation interval (30 s)
- Little impact on energy

Deployment with RPL





Impact of the deployment order

- Time dominated by installation interval (30 s)
- Little impact on energy

RPL is suitable for incremental deployment

Adapting the MAC for faster download

Principle

- Radio duty-cycling slows down the transfer
- Using a more aggressive MAC during download

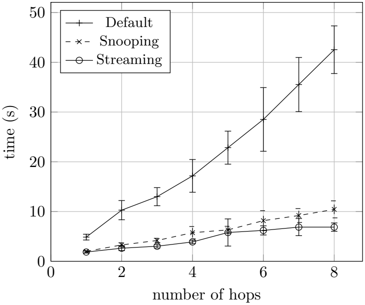
Streaming

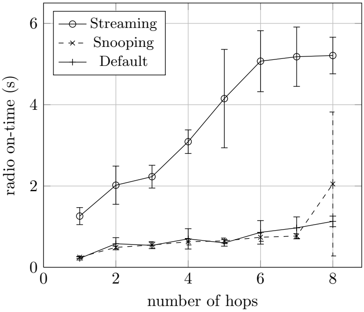
- Keep radio on 1 second after last traffic
- No more wake-up needed

Snooping

- Increase duty-cycle for 1 second after last traffic
- Shortened synchro time

Streaming and Snooping – Results

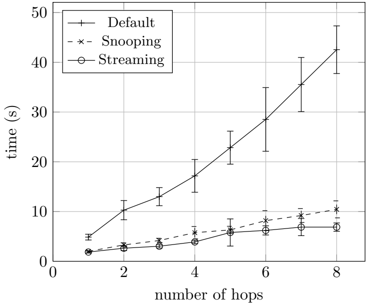


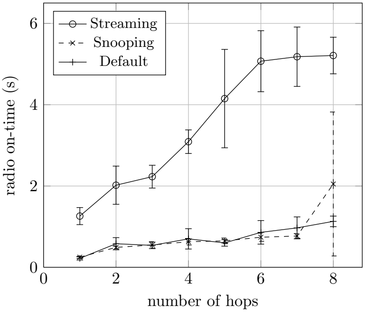


Results

- Streaming is the fastest, Default the most energy-efficient
- Snooping is a near-perfect trade-off!

Streaming and Snooping – Results



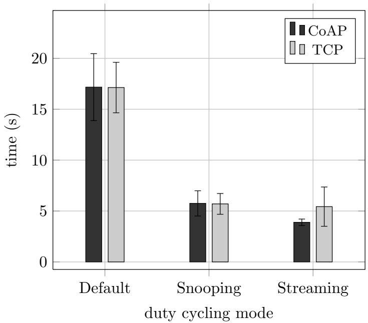


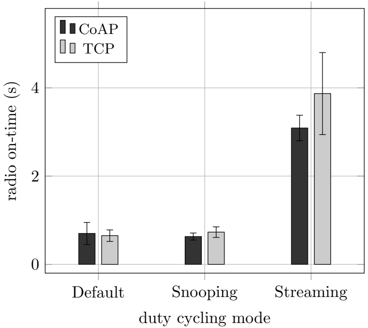
Results

- Streaming is the fastest, Default the most energy-efficient
- Snooping is a near-perfect trade-off!

Simple MAC tuning provide substantial improvements

TCP vs CoAP/UDP

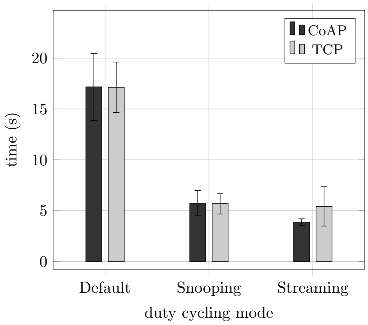


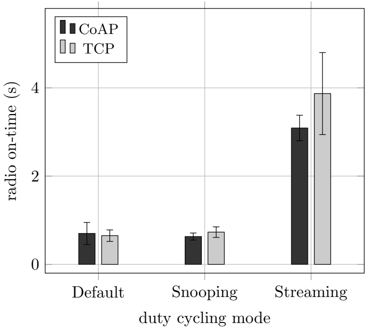


Results

- Implemented in simple packet-per-packet mode
- Both solution provide comparable results

TCP vs CoAP/UDP





Results

- Implemented in simple packet-per-packet mode
- Both solution provide comparable results

Existing transport layers are suitable

Avoiding Unicast-Based Flooding

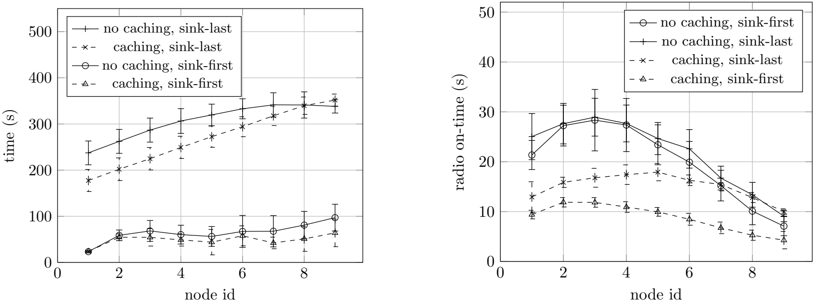
Problem

- What if many nodes need the same app.?
- This is targeted by Deluge-like solutions
- Centralized IP-based would waste energy

Idea: In-network Caching

- Nodes keep a cache of downloaded apps.
- Nodes can act as servers
- Download from nearest instead of sink
- Based on IP!

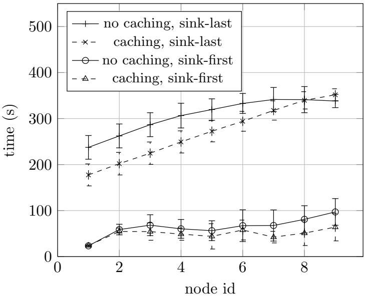
In-Network Caching – Results

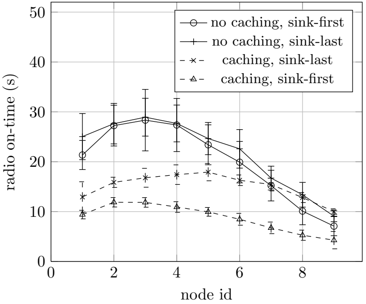


Results

- Substantial time and energy improvements

In-Network Caching – Results





Results

- Substantial time and energy improvements

All-IP facilitates rich node behavior

Conclusion & Future Work

Towards IP-based deployment

- Well-suited for heterogeneous networks
- Based on well-known/well-tested standards

Results

- IPv6, RPL, CoAP, TCP are suitable
- Simple MAC optims provide substantial improvements
- IP level optimizations (apps caching) help a lot

Towards an IP-based deployment tool

- Full scale deployment tool
- Need tesbed evaluation