The Emergence of Networking Abstractions and Techniques in TinyOS

CS295-1 Paper Presentation

Mert Akdere

10.12.2005
Outline

• Problem Statement & Motivation
• Background Information
  – TinyOS
  – HW Platforms
  – Sample Applications
• TinyOS Networking Development
  – Single-Hop Communication
  – Multi-Hop Communication
  – Network Services
• Discussion & Analysis of
  – Networking Abstractions
  – Common Techniques
• Questions
Problem Statement & Motivation

- EmNets have the “potential to change radically the way people interact with environment”
- Constraints of wireless sensor networks (WSN)
  - New design approaches required
- Focus of Paper:
  - Study of networking abstractions and techniques in TinyOS, a popular sensor network OS.
- Method
  - Examining CVS repositories
  - Compare practice (implementation) with theory (research)
**Background Information:** TinyOS

- A sensor network OS, used by many research groups
- **Design Issues:**
  - Modularity for adaptation to diversity in HW
  - Event-driven nature of WSN applications
  - Resource Constraints
  - Concurrency to cope with multiple info sources
  - Robustness
  - Application specific requirements
- **Events and Tasks**
- **Component Model:** Interfaces & Wiring
Background Information: HW Platforms

• Varying hardware, different designs and interfaces
  – Berkeley motes
  – iMote [Intel]
  – BTnode [ETH]

• Improvements in HW
  – to reduce the size, weight
  – to increase the computation and storage capacity
  – to decrease power consumption
**Background Information: Applications**

- WSN research is closely related to WSN applications
- Sample Applications and their requirements
  - Habitat Monitoring  --> Low energy consumption  Long term deployment
  - Shooter Localization  --> High sample rate  Fine-grained time sync.
  - Pursuer-Evader  --> Localization  Advanced Routing
TinyOS Networking
Single-Hop Communication

- Active Message [AM] abstraction
  - Message ID <---> Receiver Action
  - General and widely used
    - almost all reviewed applications used AM
- HW characteristics influence lower layers of radio stack
TinyOS Networking
Single-Hop Communication

• Rene:
  – RFM TR1000, simple radio, bit level interface, 10K data rate

• Mica:
  – RFM TR1000, increased data rate 40K
  – Link-layer immediate ACKs
  – Low-power listening by adjusting radio sample rate

• Mica2:
  – Chipcon CC 1000, byte level interface
  – Link-layer ACKs
TinyOS Networking
Single-Hop Communication

- **GenericComm**: TinyOS network stack interface

  ```
  interface SendMsg { // single-hop networking
    command result_t send(uint16_t addr, uint8_t len, TOS_MsgPtr msg);
    event result_t sendDone(TOS_MsgPtr msg, result_t success);
  }
  interface ReceiveMsg { // single-hop networking
    event TOS_MsgPtr receive(TOS_MsgPtr m);
  }
  ```

- Message buffers exchanged through pointers
- No send buffer pool
- Limited receive buffering

Keeping system responsive
Many implementations/algorithms exist for multi-hop communication in TinyOS

Classification:
- Tree-based Routing
- Intra-network routing
- Dissemination
TinyOS Networking
Multi-Hop Communication

Tree-based Routing:

- Two basic info
  - Parent node
  - Depth from root
- For routing packets through the root (end points)
- Routing tree formation, routing tree maintenance, and forwarding are determined by used protocol.
- AMROUTE, BLess, Surge, mh6, MultiHopRouter
TinyOS Networking
Multi-Hop Communication

Tree-based Routing:
TinyOS Networking
Multi-Hop Communication

Intra-Network Routing:

• For data transfer between in-network end points
• Uncommon in TinyOS apps
• Common in the Internet
• DSDV, AODV, Directed Diffusion
TinyOS Networking
Multi-Hop Communication

Broadcast and Epidemic Protocols:

• For dissemination to every node
• Simple flooding: Every node that hears a msg retransmits it
  - fast
  - high power consumption
  - not reliable
• Epidemic Protocols: Periodic information exchange
  - Reliable
  - Slower

start of a new epidemic round:
1. choose a communication partner from the set of neighboring nodes
2. execute information exchange with the chosen partner based on the style of epidemic algorithm
TinyOS Networking
Multi-Hop Communication

Common Developments:

• Neighboring node information
  - Link-state info
  - Routing meta-data

• Lossy and asymmetric link detection

• Common interface is used by most multi-hop routing algorithms

• Promiscuous communication for snooping

• Appearance of send queues
  - tolerant for transmission delays
  - load shedding
TinyOS Networking
Network Services

Power Management:

• Three main elements for power management:
  – Services can be stopped through function calls
  – HPLP.PowerManagement keeps processor in the lowest possible power mode at all times
  – Timer service works in lowest power mode.

• Effective power management without application specific input not possible
TinyOS Networking
Network Services

Time Synchronization:

• Needed for
  – communication scheduling
  – for slot coordination by TDMA style protocols
  – temporal data consistency

• Sub-millisecond accuracy obtained
Discussion and Analysis of Networking Abstractions

- **General Abstractions:**
  - AM abstraction
  - Common interface for different tree-based routing protocols

- **Specialized Abstractions:**
  - Power management and time synchronization

- **In-flux Abstractions:**
  - Epidemic propagation
  - Radio MAC

- **Absent Abstractions:**
  - Receive queues
  - Distributed cluster formation
Discussion and Analysis of Common Techniques

- Communication scheduling and snooping
  - to reduce the overall communication burden and power consumption
- Cross-layer Control
  - exposing state from a lower-level layer to upper layer
    - for reducing RAM use
    - for efficiency
- Static Resource Allocation
  - allocating buffers at compile-time
    - RAM is valuable
    - generally adopted policy
    - components reserve the amounts of memory they need
Thank You.

Questions...