Routing in Small Wireless Sensor Network

Diplomarbeit by Philipp Huppertz (phihup@gmail.com)

5th May 2008

Supervisor: Prof. Dr.-Ing. C. Gühmann
           Dipl.-Ing. S. Rein

Berlin University of Technology
Faculty IV – Electrical Engineering and Computer Science
Institute of Energy and Automation Technology
Department of Electronic Measurement and Diagnostic Technology
Overview

1) Motivation
2) Objectives / Goals
3) SPISA Platform
4) Software Framework and Radio Network Stack Design
   • MAC Protocol
5) Multihop Protocol Design
   • Flooding
     • DYMO and DSR
6) Multihop Protocol Evaluation
7) Conclusion
8) Perspective and Future Work
9) References / Bibliography
Motivation

Routing...

• is needed in multihop networks (low radio coverage)

• is an open research topic in Wireless Sensor Networks (WSNs)
  – Many protocols have been designed and evaluated using simulation tools.
  – Few have been investigated in real-world scenarios (mostly using TinyOS\cite{1}).

• Several studies\cite{2,3} show that simulation results may not be representative for real-world scenarios.

>> real-world testing may produce new insights <<
Objectives / Goals

• Implementation of general purpose ad-hoc routing protocols for the SP/ISA platform:
  – DSR\[^4\], AODV\[^5\] and DYMO\[^6\] routing protocols
  – alternative approach: Flooding protocol\[^4\]

• Optimise and adapt to needs of small WSN
  – Routing protocols were originally designed for MANETS

• Compare Performance for different scenarios
  – Goodput
  – Reliability/Packet-Loss
  – Energy Consumption
  – Complexity (code size & memory consumption)
SPISA Platform

- **Hardware**
  - Two different board revisions with different MCUs
    - dsPIC30F4013 (2 KiB RAM & 46 KiB)
    - dsPIC33FJ256GP710 (32 KiB RAM & 256 KiB)
  - Same nRF905 radio chip
    - Packet radio interface
    - Hardware CRC
    - Hardware addressing
    - Currents: RX 12.5 mA
    - TX 9 - 30 mA
    - IDLE 0.3 mA

- **Software**
  - Serial, Stack/Matrix, SD-Card, TDMA protocol, cam, etc.
  - SpisaTerm on PC-side
    (for communication with node)

4 nodes (dsPIC30Fxxx)  2 nodes (dsPIC33FJxxx)
General Software Design

- **Application framework**
  - Modular – several utility modules (timer abstraction, task queue)
  - Allows reuse of existing modules – do not reinvent the wheel
  - Applications can link to modules they need

- **Radio network stack**
  - Interrupt driven / Cooperative tasks
  - Complete stack
    - Transport layer (Files/Matrix structures)
    - Multihop protocol (Routing & Flooding protocols)
    - MAC protocol
    - nRF905 chip driver

---

![Diagram of data flow and control flow in a radio network stack](image-url)
MAC Layer Protocol

• The MAC Protocol builds the basis of the radio stack
  - The existing protocol supported only communication between two nodes!
  - CSMA / CA with binary exponential backoff and ACKs, based upon IEEE 802.15.4\cite{7}
  - Congestion is handled with ACKs (closed loop): missing ACK → backoff

• Link-Quality Characteristics are investigated
  - Needed as a basis for later multihop setup (physical placement)
  - One sender, one receiver
    • Varied distance (30 runs per distance)
    • Transmission of 1000 packets with and without ACK
Routing in a Small Wireless Sensor Network
Multihop Protocols

- Flooding protocol (as alternative to routing):
  - Simple flooding (rebroadcast unknown message)
    - Redundant broadcasts (BCs)!
  - Flooding protocols trying to minimize redundant BCs\(^8\)
    - Trade-off between reachability and minimizing redundant BCs (e.g. counter based -, stochastic flooding)

  \(>>\) makes no sense in small (sparse) WSNs \(<<\)

- Ad-hoc multihop protocol properties:
  - Route is setup ad-hoc, i.e. only when needed
  - Functionality can be divided into:
    - Route Discovery (setup, finding)
    - Route Maintenance (error handling, route shortening)
Multihop Protocols - DSR

- Dynamic Source Routing (basic features):
  - Route Discovery is done by Route Request (RREQ) Flood
    - Every node forwarding RREQ appends its address (reverse route is set up)
    - Destination node answers with a Route Reply (RREP) sent unicast via reverse route.
  - Route Maintenance consists only of route error detection
    - Route Error (RERR) packet is sent back to source via reverse route
  - Complete route is stored in each data packet!

![Route Request Diagram]

![Route Reply Diagram]
Multihop Protocols - DYMO

- Simplified Dynamic MANET On Demand: SDYMO
  - Route Discovery similar to DSR
  - Differences to DSR:
    - Forwarding nodes need valid routing table entry (next hop → dst)
    - Only one route entry for each destination possible
    - Route is only contained in RREQ/RREP packets (no caching, packet payload bigger – higher goodput)
    - DYMO uses local broadcast for RERR transmission

Route Error (DSR):

Route Error (SDYMO):
Multihop Protocol Evaluation

- Multihop application:
  - Command driven application controlled from PC (SpisaTerm)
  - Measurements are collected via radio interface

- Performance of Plainflooding, DSR and SDYMO are compared for several scenarios:
  - One sender, one receiver
  - Two sender, one receiver (concurrent transmission)
  - Two sender, two receiver (concurrent transmission)
  - Packet/Matrix transmission

- Measurements:
  - Goodput, Success/Packet Delivery Ratio, Mean Sent Packets (estimator for energy consumption), Route Setups, Route Setup Time
Multihop Protocol Evaluation Results (1)

- **Goodput** for matrix transmission scenarios using Transport Layer (bidirectional data flow – measurements repeated 30 times):

![Graphs showing Goodput for matrix transmissions involving single and concurrent transmissions from one sender to one or two receivers.](image)
Multihop Protocol Evaluation Results (2)

- **Route Setups & Setup Time** for matrix transmission scenarios using Transport Layer (bidirectional data flow – measurements repeated 30 times):

**Routing protocol performance for matrix transmissions**

![Graphs showing route setups and setup time for different scenarios.](image-url)
Conclusion – Evaluation Results

• MAC protocol:
  - nRF905 physical transmission is affected by multipath fading
  - Transitional region of link-quality
    (weak correlation between distance and reception rate)
  - Asymmetric link behaviour, also seen during multihop evaluation
    (different hop count seen at sender and receiver)

• Multihop protocols:
  - **DSR**: Highest *reliability*, high goodput
  - **SDYMO**: Highest *goodput*, weak route error handling
  - Both routing protocols:
    • High amount of route setups (unstable routes) → energy consumption
  - **Plainflooding**: lowest *complexity*, lowest goodput (rate limiting),
    energy consumption is the highest for short hop distances

>> needs a very application specific – no “best” solution <<
Perspective and Future Work

• MAC
  - Better congestion / channel usage handling (adaptive rate control\textsuperscript{[9]})
  - MAC has high energy consumption (idle listening)

  \textbf{>> sleep / rendezvous MAC <<}

• Routing Protocols
  - Shortest hop distance not the best metric\textsuperscript{[10,3]}
  - Investigate \textit{metrics based upon link quality}:
    • May improve DSR & SDYMO performance (more stable)
    • Enhance MAC protocol to allow estimation of link quality\textsuperscript{[11]}
The End

Thank you!

Hopefully there are questions!? 
Demonstration (1)

• The demo shows an exemplary measurement:
  – A transmission from node six to node one using DYMO
  – The measurement setup and data collection is automated by a SpisaTerm script, results are computed using GNU Octave

• The physical setup of the WSN can be seen in the figure below
Demonstration (2)

- The SpisaTerm script invokes the following actions:
  (The script is shown in the right-hand window in the video)
  - Reset all nodes and clear measurements
  - Setup an measurement task, i.e. a transmission from one sender to one receiver.
  - Collect measurements from all six nodes
  - Store measurements on PC

- Measurements are calculated using GNU octave are:
  (The m-file can be seen in the lower window in the video)
  - Goodput
  - Hop Distance (seen at sender and receiver)
  - Route Setup Time
References / Bibliography (1)


References / Bibliography (2)


Backup - Conclusion - Achievements

- Small framework was designed (virtual timers, tasks, etc.)
- Complete radio network stack was designed & implemented:
  - Basic CSMA/CA MAC protocol
  - Different Multihop protocols: DSR, SDYMO, Flooding
  - Transport-Layer protocol (files/matrices)
- Investigation of Link-quality using developed MAC
- Investigation of multihop protocols for different scenarios:
  - Includes design and implementation of multihop evaluation application and extension of SpisaTerm PC-application to support scripting
Backup - Application Framework

Allows reuse of modules and helper functions...

- **Components:**
  - *Radio* (described in more detail later...)
    - Complete radio network stack
  - *Stack* (by Stephan Rein & Stephan Lehmann)
    - Software stack supporting matrix handling and manipulations
  - *SD-Card* (by Stephan Lehmann)
    - Access SD-Card on SPISA rev. 4 boards + lightweight file-system
  - *Util*
    - **Task queue:** Offers task queue for cooperative multitasking
    - **Timers:** Timer abstractions and Virtual Timers
    - Serial communication: allows communication with PC (SpisaTerm) (by Stephan Lehmann)
    - Debug-, Random-, Platform- modules (general helper functions...)
Backup - Radio Network Stack

- **radiotrans** (tasks & split phase)
  - Transport layer (matrix/file)

- **multihop** (tasks & split phase)
  - DSR, DYMO, Flooding

- **basicmac**
  - CSMA with ACKs
    - (bases on unslotted CSMA described in IEEE 802.15.4\cite{7})

- **nrf905**
  - Exposes radio chip functionality

---

Routing in a Small Wireless Sensor Network    -  24
Backup – Transport Protocol

• Simple Stop-and-Wait Protocol (using ACKs)
  - Matrix/File is split up into several packets
  - RTS packet is sent informing about length – allows receiver to allocate memory
  - Supports several receive streams (receiving concurrently matrices/files from different nodes)
  - Single send stream
  - Only unicast transmissions are supported!
  - Additional packet interface to allow sending of single packets (used in multihop application to send commands)
Backup - MAC Protocol

- MAC protocol based on CSMA/CA scheme:
  - Binary exponential backoff
  - Extra backoff before first transmission
    - Avoids correlation if multiple nodes transmit periodically
  - ACK scheme used to handle congestion
    - If no ACK is received MAC goes into backoff
  - Use of ACKs can be controlled on per packet basis
  - Max. Goodput ACK / no ACK: ~12.5 / ~20 kbps
  - Hardware addressing of nRF905 unused
    - No broadcast address supported!
    - Address is abused as start frame delimiter!?
Backup – Multihop Protocol Complexity

- Code-size and Memory Consumption for the dsPIC30Fxxx boards:
  - 2 KiB RAM
  - 48 KiB ROM

<table>
<thead>
<tr>
<th></th>
<th>application</th>
<th>radio component</th>
<th>multihop module</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Plainflooding</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROM size</td>
<td>27378 bytes</td>
<td>3822 bytes</td>
<td>374 bytes</td>
</tr>
<tr>
<td>RAM size</td>
<td>754 bytes</td>
<td>328 bytes</td>
<td>84 bytes</td>
</tr>
<tr>
<td>ROM size</td>
<td>(66.8 %)</td>
<td>(9.3 %)</td>
<td>(0.9 %)</td>
</tr>
<tr>
<td>RAM size</td>
<td>(36.8 %)</td>
<td>(16.0 %)</td>
<td>(4.1 %)</td>
</tr>
<tr>
<td><strong>DSR</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROM size</td>
<td>29298 bytes</td>
<td>5088 bytes</td>
<td>1640 bytes</td>
</tr>
<tr>
<td>RAM size</td>
<td>780 bytes</td>
<td>354 bytes</td>
<td>110 bytes</td>
</tr>
<tr>
<td>ROM size</td>
<td>(71.5 %)</td>
<td>(12.4 %)</td>
<td>(4.0 %)</td>
</tr>
<tr>
<td>RAM size</td>
<td>(38.0 %)</td>
<td>(17.3 %)</td>
<td>(5.4 %)</td>
</tr>
<tr>
<td><strong>SDYMO</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROM size</td>
<td>29064 bytes</td>
<td>4928 bytes</td>
<td>1480 bytes</td>
</tr>
<tr>
<td>RAM size</td>
<td>770 bytes</td>
<td>344 bytes</td>
<td>100 bytes</td>
</tr>
<tr>
<td>ROM size</td>
<td>(71.0 %)</td>
<td>(12.0 %)</td>
<td>(3.6 %)</td>
</tr>
<tr>
<td>RAM size</td>
<td>(37.6 %)</td>
<td>(16.8 %)</td>
<td>(4.9 %)</td>
</tr>
</tbody>
</table>
Multihop Protocol Evaluation Results (1m)

- **Mean Sent Packets (estimator for energy consumption)** for matrix transmission scenarios using Transport Layer (bidirectional data flow – measurements repeated 30 times):

![Graphs showing mean sent packets for different scenarios](image_url)
Multihop Protocol Evaluation Results (1p)

- **Packet delivery ratio** for packet transmission (#1000) scenarios: (unidirectional data flow – measurements repeated 30 times)

Packet delivery ratio for packet transmissions

![Graph showing packet delivery ratio for single and concurrent transmissions](image)

**Transmission paths** (sender -> receiver)

<table>
<thead>
<tr>
<th>Transmission path</th>
<th>Packet delivery ratio [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 -&gt; 1</td>
<td>80</td>
</tr>
<tr>
<td>6 -&gt; 3</td>
<td>90</td>
</tr>
<tr>
<td>6 -&gt; 4</td>
<td>85</td>
</tr>
</tbody>
</table>

**Transmission paths** (sender -> receiver)

<table>
<thead>
<tr>
<th>Transmission path</th>
<th>Packet delivery ratio [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 -&gt; 4</td>
<td>75</td>
</tr>
<tr>
<td>5 -&gt; 3</td>
<td>90</td>
</tr>
</tbody>
</table>

**Legend**

- **PF**
- **DSR**
- **SDYMO**
- **standard deviation**
- **95 % CI**
Multihop Protocol Evaluation Results (2p)

- **Mean Sent Packets (estimator for energy consumption)** for packet transmission (#1000) scenarios:
  (unidirectional data flow – measurements repeated 30 times)

Measured number of sent packets for packet transmissions

![Graphs showing sent packets](image)