Position and Topology Estimation in Bluetooth Ad-hoc Networks

Diploma Thesis
Urs Frey

Jan Beutel, Matthias Dyer
(Oliver Kasten, Matthias Ringwald)
Overview

- Platform – BTnodes
  - Memory Sections
  - Bluetooth – HCI Interface
- Service Routines
  - xHop
  - Bootloader
- Positioning on BTnodes
  - Hop-TERRAIN
- Conclusion
Platform - BTnodes

- Ericsson ROK Module
- AVR ATmega128
  - 8 bit RISC Architecture
  - memories
  - interfaces
- External SRAM
- LEDs, Reset Button
- Power management
Memory Sections

- **External SRAM**
  - 4 banks of 60kB
  - xCopy routine
  - e.g. bootloader data

- **AVR**
  - 128kB Flash – program
  - 4kB EEPROM - features
  - 4kB SRAM - stack
BT Layers, HCI Interface

- HCI Interface
  - e.g. Inquiry

BT Host – ATmega128

BT Hardware – Ericsson ROK
System Software

- Lightweight OS
  - event-driven application model
  - cooperative multithreading
  - device drivers (UART, RTC, ADC, ...)

- Programming
  - standard C language
  - high-level Bluetooth interface
  - emulation environment on Linux
R-DSR / xHop - Multihop

- xHop to cross piconet boundaries
- scatternets provide no solution for positioning

Header:  Payload:

<table>
<thead>
<tr>
<th>route len</th>
<th>route pos</th>
<th>route (BT addresses)</th>
<th>cmds</th>
<th>answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>6 * route_len</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Diagram:

A \rightarrow B \rightarrow C \rightarrow D
xHop to cross piconet boundaries
scatternets provide no solution for positioning

Header:        Payload:

|  4 |  2 | ABCD | get position|xhop rev |
|-----|----|------|-------------|
| route len | route pos | route (BT addresses) | cmds | answer |

A → B
C
D
R-DSR / xHop - Multihop

- xHop to cross piconet boundaries
- scatternets provide no solution for positioning

<table>
<thead>
<tr>
<th>Header:</th>
<th>Payload:</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

- route len
- route pos
- route (BT addresses)
- cmds
- answer

Diagram of nodes A, B, C, D with arrows indicating connectivity.
xHop Performance

- 57 packets
- 2 failures
- $E(X)=2.1$ s per hop

Conditions:
- inq every 180s for 6s
- 9 hops (5 different nodes)
- only one packet at any time
Bootloader

Bluetooth PC [00:80:37:17:4d:30]

BTnode [00:80:37:17:4d:02]
Initialising Data Structures...
Locally Supported PSM: 101, 103, 105, 107
Local BT address: 00:80:37:17:4d:30
[00:80:37:17:4d:30]> program
read hex file: 4930 lines; RAM: 78840 byte
read bt addresses ... reprogram 4 nodes
compose bootloader request/data packet

Request:

<table>
<thead>
<tr>
<th>len</th>
<th>status</th>
<th>version</th>
<th># nodes</th>
<th>addresses</th>
<th>CRC</th>
</tr>
</thead>
</table>

Data (modified intel hex format):

<table>
<thead>
<tr>
<th>data</th>
<th>CRC</th>
</tr>
</thead>
</table>
Bootloader

Bluetooth PC [00:80:37:17:4d:30]

Initialising Data Structures...
Locally Supported PSM: 101, 103, 105, 107
Local BT address: 00:80:37:17:4d:30

read hex file: 4930 lines; RAM: 78840 byte
read bt addresses ... reprogram 4 nodes
compose bootloader request/data packet
xterm
./btnode_system /dev/ttyS0
program

BTnode [00:80:37:17:4d:02]

Bluetooth PC [00:80:37:17:4d:30] ->
check version
accept/deny
send data

reveived bootloader request packet...check program version
send accept ...
receive program data
done ... start bootloader

minicom: /dev/ttyS0

copy program data from RAM to Flash ...
... done, reboot ...

Initializing data structures
Locally supported PSM: 101, 103, 105, 107
Local BT address: 00:80:37:174d:02
[00:80:37:17:4d:02] >
Bootloader

Bluetooth PC [00:80:37:17:4d:30]

BTnode [00:80:37:17:4d:02]

(selective) network flooding
Positioning: Measurements

- **Distance**
  - RSSI
  - ToA, TDoA

- **Angle**
  - antenna arrays

**Diagram Notes:**
- **free node** (unknown position)
- **free node** (estimated position)
- **anchor node** (known position: GPS, fixed, ...)

01/03/2003
Positioning: Algorithms I

- Discrete Ranges
  - LP (min/max values)

- Mesh Based
Positioning: Algorithms II

- Local Coordinate Systems
  - translation, rotation, flip

- Global Coordinates
Positioning: Algorithms II

- Local Coordinate Systems
  - translation, rotation, flip

- Global Coordinates
Positioning: Algorithms II

- Local Coordinate Systems
  - translation, rotation, flip

- Global Coordinates
Positioning: Hop-TERRAIN

- Implemented on BTnodes
- Two stage algorithm
  - 1\(^{st}\): hop-TERRAIN
    - broadcast anchor positions
    - triangulate
  - 2\(^{nd}\): refinement
    - triangulation with weighted least-squares
Positioning: Hop-TERRAIN

- Implemented on BTnodes
- Two stage algorithm
  - 1\textsuperscript{st}: hop-TERRAIN
    - broadcast anchor positions
    - triangulate
  - 2\textsuperscript{nd}: refinement
    - triangulation with weighted least-squares
Positioning: Hop-TERRAIN

- Implemented on BTnodes
- Two stage algorithm
  - 1st: hop-TERRAIN
    - broadcast anchor positions
    - triangulate
  - 2nd: refinement
    - triangulation with weighted least-squares
Positioning: Hop-TERRAIN

- Implemented on BTnodes
- Two stage algorithm
  - 1st: hop-TERRAIN
    - broadcast anchor positions
    - triangulate
  - 2nd: refinement
    - triangulation with weighted least-squares
Results Positioning (Refinement)

- experiment with 14 measurements
  - 6 fatal network failures
  - best time 1min 16s
  - worst time 5min 33s
  - average 3min 20s

- scheduling problem
- inquiry blocks communication
- connecting and disconnecting is slow
- simultaneously opening a connection may crash the stack

- currently least square computation only on Linux
Conclusion

- Ad-Hoc Network Services for Bluetooth
  - Bootloader useful development tool
  - xHop enables large network configurations

- Positioning on Btnodes
  - Functional in *static* environment
  - Bluetooth too slow for *mobile* positioning?
  - Extend BTnodes with ‘custom’ radio interface?
  - Extend positioning algorithms with some regional coordination/scheduling schemes?

- Bluetooth is not that simple 😊
Questions?