BTnodes

Scaling it up
Networking using the BTnode Platform

Jan Beutel
Outline

Wireless Sensor Networks – visions and current status

Example: Constructing network topologies using Bluetooth

– BTnode – Ad hoc networking prototyping platform
– robust, self-healing tree topology TreeNet Algorithm
– implementation requirements and issues

Lessons learned
Wireless Sensor Networks visions

Large scale of proposed systems
- centralized, decentralized, clustered
- very few, many, massive amounts
- functionally rich, constrained
- homo-, heterogeneous
- self-configuring, managed
- failure tolerant, QoS

Smart Dust [Kahn1999]
Paintable Computing [Butera1999]
Picoradio [Rabaey1999]
Terminodes [Hubaux1999]
Amorphous Computing [Abelson2001]
Specnet [Arvind2003]
Diffusion [Estrin2000]
WINS [Pottie2000]
Prototyping Wireless Sensor Networks

A myriad of interacting devices
– sensor node heterogeneity
– sensing and actuation
– user interaction

Smart everyday objects
by attaching sensor nodes:
– self aware
– context sensitive
– cooperative
– integration into computing environment

NCCR-MICS Terminodes

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
Wireless Sensor Network systems today

Sub mm scale, super high density all the way to layered, semi infrastructure dependant iPAQ/PC architecture nodes.
BTnode prototyping platform

Lightweight wireless communication and computing platform based on a Bluetooth radio module and a microcontroller.

Bluetooth has the advantage of
- availability today for experimentation
- compatibility to interface to consumer appliances
- an abstract, standardized high level digital interface
Bluetooth architecture details

Integrated hardware features
- 8-Bit RISC, max. 8 MIPS, 128 kB Flash, 64 kB SRAM, 180 kB data cache
- operating from 3 cell batteries
- generic sensor interfaces

Event-driven lightweight OS
- standard C language
- system software available as library

<table>
<thead>
<tr>
<th>Current bill of material</th>
<th>50 parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parts</td>
<td>60 USD</td>
</tr>
<tr>
<td>Assembly</td>
<td>5 USD</td>
</tr>
<tr>
<td>Bluetooth</td>
<td>45 USD</td>
</tr>
</tbody>
</table>

Unit cost @ 200 units 110 USD
Other BTnode applications

Many successful BTnode applications
- The Lighthouse location system [Roemer2003]
- Smart product monitoring [Siegemund2002]
- Bluetooth enabled appliances [Siegemund2003]
- Smart It’s friends [Siegemund2003]
- XHOP/R-DSR multihop prototype [Beutel2002]
- Distributed positioning – TERRAIN implementation [Frey2003]
- Physical activity detection network [Junker2003]
- Better avalanche rescue through sensors [Michahelles2002]
- Wearable unit with reconfigurable modules [Plessl2003]
- Undergrad projects with Lego Mindstorms [Blum2003]
- …

Mostly relying on simple point to point data links
Constructing large network topologies

How to construct an ad hoc network topology with Bluetooth

- large network, many devices
- all devices connected, supporting transparent multihop transport

XHOP prototype

TreeNet topology
Constructing network topologies

**Scatternet formation algorithms**
- many theoretical studies and simulations often far away from reality
- improvements in the current Bluetooth voting draft specification v1.2

**BlueStars** [Basagni2002/3], **BlueRing** [Lin2003] ... 
- make assumptions on physical prerequisites not available today
- assume “perfect” connection performance
- assume symmetric data availability on nodes

**Ad hoc network topologies only in simulations**
- usually all using the same underlying physical models
- often lacking realistic distributed system models for large networks
- limited access to appropriate hardware devices
BTnode networking – definitions

Four states
- IDLE
- MASTER
- SLAVE
- MASTERSLAVE

Useful operations
- inquiry() – find other nodes
- connect() – open connection
- roleSwitch() – change MS relation
- sendData() – data transport

Hardware limitations on the BTnodes/Bluetooth
- max. 7 active slaves in one Piconet
- while in inquiry() and connect() a node is not visible
- while in SLAVE or MASTERSLAVE a node is not visible
- while in SLAVE or MS a node cannot do inquiry() or connect()
- inquiry() and connect() have long delays and no a priori guarantee

Bluetooth only defines single hop Master-Slave data transport
Distributed Bluetooth Piconets

Distributed *inquiry()* and *connect()* is a problem
- nodes are uncoordinated
- limited visibility
- asymmetry: inquired node doesn’t notice

*Inquiry()* and *connect()* have long delays
- state change in remote node goes unnoticed
- average delay in seconds [Kasten2001]
- no a priori guarantee for success

*Inquiry()* and *connect()* are highly nondeterministic (both in timing and function)
TreeNet simple tree construction

Every node executes algorithm
– until a single tree is reached

Formation of large topologies
– robustness
– simplicity
– redundancy
– distribution
– self-healing

Demonstrated with 40 nodes at NCCR-MICS annual review
Lessons Learned

A. A 7 line high level algorithm leads to about 2000 lines of code.

B. It is very difficult to test, debug, deploy and evaluate a large amount of devices.
A. Code size and complexity

Lockup issues
– might not fully connect if multiple max_degree roots form
– distributed inquiry() and connect() problem

Performance issues
– simple greedy algorithm reduces inquiry() and connect()
– highly non-deterministic behavior

Basic underlying infrastructure
– data storage and exchange
– timing and time-stamping
– connection/link management

Leads to about 2000 lines of additional code!
B. Large scale distributed deployment

So why do we actually need even more lines of code?

- additional system software + debugging + visualization + monitoring
- stepwise testing and deployment
- result in an ~87 kB program (un-optimized)

Other problems we had with deployment

- cables
- batteries
- mounting/casing
- (re-)programming
- debugging of a distributed concurrent system
- developing for stepwise deployment
- visualization/analysis
- online access to nodes
- ...

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
It is hard to deploy anywhere beyond 10-20 nodes today.

Coordinated methods, concepts and tools are missing today.
Acknowledgements

**BTnode/TreeNet collaborators**
- Oliver Kasten, Friedemann Mattern, Matthias Ringwald, Kay Römer, Frank Siegemund
- Regina Bischoff, Roger Wattenhofer, Aaron Zollinger
- Jan Beutel, Matthias Dyer, Lennart Meier, Martin Hinz, Lothar Thiele

**Related publications**

**Thanx for material to**
- Deborah Estrin, Jeremy Elson, Ivo Locher, Mani Srivastava, UCLA
- David Culler, Rob Szewczyk, Paul Lewis, Jan Rabaey, Brian Boser, UC Berkeley
- Ralph Kling, Intel Research
To probe further...

http://www.btnode.ethz.ch