Réseaux de Collecte Dynamique
Thomas Watteyne

22 novembre 2007
Quatrième workshop CNRS RECAP, Montpellier
Thomas Watteyne

CITI Lab, INRIA
Lyon, France
Isabelle Augé-Blum

France Telecom R&D
Grenoble, France
Mischa Dohler

Energy-Efficient Self-Organization in Wireless Sensor Networks

Web: Google me!
- Wireless Sensor Networks
- Energy-Efficiency
- Self-Organization
- Experimentation
Wireless Sensor Networks and Self-Organization
"Self-organization can be defined as the emergence of system-wide adaptive structure and functionality from simple local interactions"

C. Bettstetter
"Bio-inspired" protocols
Size of a communication range
Energy problem

Customized GTSNetS simulator

Gradient routing

1hopMAC
Cross Layering

"The suboptimality and inflexibility of [the layering approach] result in poor performance for WSANs, due to constraints of low energy consumption and low latency. Therefore, instead of having individual layers, we may need cross-layering where layers are integrated with each other". Ian Akyildiz, 2004.
2
Energy Efficiency through MAC-layer design
1hop-MAC

- Election of a neighbor as next hop
- Preamble sampling with micro-frames for energy efficiency
3

Self-Organization using Virtual Coordinates
Routing protocols

- data-centric
  - flooding (1988)
  - gossiping (1988)
  - Dir. Diff. (2000)
  - ...

- hierarchical
  - LEACH (2000)
  - TEEN (2001)
  - PEGASIS (2002)
  - ...

- geographic
  - MFR (1984)
  - GAF (2001)
  - ...

- easy
- implicit organization

- energy inefficient?
- organized
- slow/overhead
- location-awareness
Greedy geographic routing

may fail

If it doesn't fail, near to shortest path
Timeline

• 1999: Pure geographic routing


• 2004: Self-Organization through clustering


• 2004-2005: Infer location from location-aware anchors


• 2006: Infer location from location-**un**aware anchors

Introducing virtual coordinates

**Real positions**

**Virtual positions** = random positions
Routing with virtual coordinates

**Real coordinates**
- Each node knows its real coordinates
- Each node learns its neighbors' real coordinates
- Each node learns the sink's real coordinates

**Virtual coordinates**
- Each node knows its virtual coordinates
- Each node learns its neighbors' virtual coordinates
- Each node knows the sink's virtual coordinates, which are known by all a priori
Positioning inaccuracy

When positioning is not perfect, creating a planar graph disconnects the network, and GFG/GPSR fail.
Our proposal

Record path in the packet

1. never send a packet to a neighbor whom you have already sent a packet to;

2. send a packet back to a neighbor (i.e. he has sent you a packet before) only if there are no other neighbors you have never communicated with;

3. if you have several choices of neighbors whom you can send back a packet, pick the neighbor who has sent you a packet last.
Example
Results with perfect positioning

100% delivery ratio

Same hop count as GFG/GPSR
Results with **random virtual coord.**

![Graph showing results with random virtual coord.](image)
Centroïd transformation

1. Exchange virtual coordinates with your neighbors
2. Calculate the point of gravity of your neighbor's coordinates
3. Update your virtual coordinates with this position

• simultaneous rounds at network initialization
After some centroid runs
After some centroid runs
Proof-of-concept Experimentation
The communication stack

- **Application**
  - connectivity graph discovery

- **Routing**
  - 3rule routing
  - virtual coordinates

- **Medium Access Control**
  - 1-hopMAC

- **Physical layer**

- **Energy efficiency**
Scenario
Scenario

light sensor
Overview of the different protocols

Base Station
• Periodically sends Data Request, and waits for answer.

Plane
• Receives the Data Request, periodically sends Broadcast Request.
• waits for Data from the network.
• wait for Data Request, answers with Data.

Nodes
• upon receiving Broadcast request, broadcast the network.
• Source node identifies itself and sends Data using 3rule routing/virtual/1-hopMAC.

All nodes perform preamble sampling while idle.
Impact of a mobile sink

Real positions

Virtual positions = random positions
**Maximum Speed of the plane**

Talking to the base station

\[
\nu_{max} = \frac{50}{T_{DR} + D_{DRp} + D_{DATA}} \\
\approx \frac{500}{500 \text{ km} \cdot \text{h}^{-1}}
\]
Maximum Speed of the plane

Talking to the network

Worst Case Broadcast: 8 hops
Bad Case Routing: 10 hops

\[ v_{max} = \frac{150}{8 \cdot (W_{BR} + D_{BRp}) + B_{SRC} + 10 \cdot (D_{RRp} + W_{RR} + D_{DATA})} \approx 150 \text{ km/h} \]
Maximum Speed of the plane

[Graph showing miss ratio vs. sink speed]

miss ratio
Hardware – motes

Atmel AtMega128L

EM2420

EW2420 (×20)
Ember

Development Kit
THInk Is Not a Kernel

- Minimal Operating System
- Component Based
- Compiler
- Architectural Langage + C
Energy consumption of 1-hop MAC

![Graph showing energy consumption over time for nodes A, B, and C]

<table>
<thead>
<tr>
<th>EM2420 module</th>
<th>0dBm</th>
<th>-25dBm</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{sleep}$</td>
<td>8.018 mW</td>
<td>2.735 mW</td>
</tr>
<tr>
<td>$P_{poll}$</td>
<td>8.629 mW</td>
<td>3.300 mW</td>
</tr>
<tr>
<td>$P_{listen}$</td>
<td>65.833 mW</td>
<td>61.030 mW</td>
</tr>
<tr>
<td>$P_{Tx}$</td>
<td>66.156 mW</td>
<td>32.807 mW</td>
</tr>
<tr>
<td>$P_{Rx}$</td>
<td>70.686 mW</td>
<td>65.444 mW</td>
</tr>
</tbody>
</table>

![Table showing energy consumption for each module state]
Demo!

10 August 2007
Alpe d'Huez, French Alps

- Dominique Barthel
- Michaël Gauthier
- Thomas Watteyne
It works!

WiFly demo
10 August 2007
Alpe d'Huez, France
References


Thomas Watteyne

thomas.watteyne@orange-ftgroup.com

http://perso.citi.insa-lyon.fr/twatteyn/