I. RSSI and Ad-Hoc Networking

Portions adapted from
Kamin Whitehouse
And Crossbow

Design Principles

- Node-level Resolution
- Scalable Deployment
- Event-driven
- Simple and Approximate Operation
Radio Ranging – Connectivity

Data courtesy Alec Woo, Ganesan, et al
Radio Ranging – Signal Strength

Outdoors: Density Plot of Distance Estimates: RSSI 24.9407% error; 118.5437 cm error

Radio Ranging – Signal Strength

Indoors: Density Plot of Distance Estimates: RSSI 43.8991% error; 169.5373 cm error
Radio Ranging – Signal Strength

Average Signal Strength over Distance

Distance (ft) vs. Signal Strength (V)

- Large Error
- Small Error
Radio Ranging – Signal Strength

- Error equation:

\[
\text{error (cm)} \approx \frac{\text{noise (dB)}}{\text{Attenuation rate (dB) cm}}
\]
Localization Accuracy

Empirical Probability of Error (kernel smoothing)

Error (relative to maximum range)

Resolution of Forces

Localization Accuracy
Localization Accuracy

![Localization Accuracy Graph]

Evaluation

- Node-level Resolution
- Scalable Deployment
- Event-driven
- Simple and Approximate Operation
II. MultiHop Networking

Overview:

• Characteristics of wireless mesh networks
• TinyOS Status of MultiHop Protocols

Ad-Hoc Routing
(Self configuring)

• Links are not reliable over the long term
• Links change dynamically
• Requires networking topology that also dynamically changes.
• Low energy requirements limit types of protocols. Powered networks can afford to expend a lot more energy to manage links.
• Broadcasting is energy and time inefficient
• Protocols where the motes dynamically determine the best parent are attractive.
Mote Msgs Constrain Sleep Time

Computation of required sleep time to achieve duty cycle given number of TOS msgs to rcv/xmit while awake

<table>
<thead>
<tr>
<th>Specifications</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Msg Size</td>
<td>40</td>
<td>bytes</td>
</tr>
<tr>
<td>Msg Preamble</td>
<td>16</td>
<td>bytes</td>
</tr>
<tr>
<td>Baud Rate</td>
<td>38400</td>
<td>baud</td>
</tr>
<tr>
<td>Duty Cycle</td>
<td>0.5</td>
<td>%</td>
</tr>
<tr>
<td># of msgs to rcv/xmit during wake time</td>
<td>5</td>
<td>msg</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Computed Values</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to xmit/rcv 1 msg</td>
<td>11.7</td>
<td>msec</td>
</tr>
<tr>
<td>Time to rcv/xmit all msgs</td>
<td>116.7</td>
<td>msec</td>
</tr>
<tr>
<td>Required sleep time to maintain duty cycle</td>
<td>23.22</td>
<td>sec</td>
</tr>
</tbody>
</table>

Motes closer to base station get more messages

Radio Link Behavior #1

- Radio contour plot shows received radio strength varies significantly.

From Alec Woo, To be Published
Radio Link Behavior #2

Static links show variability in receive strength over time
• Local null effects, people, … influence quality of link

From Alec Woo, To be Published

Broadcasting (flooding)

• Backward links
• Longs links
• Stragglers (no reception)
• Clustering
• Asymmetric links

Flooding at 19.2K baud Tos Packet, 156 motes
• 80msec to receive then transmit
• Each mote transmits once:
  - 156*40msec = 6.25 sec (no mdlm delay)
  - Probably ~13 seconds with mdlm delay
  - Large probability of pckt collision

13x12 grid (156 motes)
Simulation

- TinyOS supports PC simulation of networks.
- Best way to start, prove out code before real implementation
- Reasonably good results between simulation and real world

![Simulation Diagram]

TinyOS and Multihop

- Complex routing protocols
- Complex energy management
- Complex time management
- No single multihop stack; application dependant
- TinyOS will allow users to wire in different protocols with minimal effort.
- TinyOS will allow users to compare different protocols for reliability and efficiency.
- Time/Power management is being added to these protocol (3-4 months)
MultiHop Components

- Routing beacon from base station to establish route paths back.
- Find best parent to forward messages:
  - Routing Table: List of best neighbors and routing info. Sram constrained.
  - Table Management: Eviction/Insertion of neighbors into routing table
  - Parent Selection: Decide on parent to forward messages.
  - Cycle Detection: Avoid loops (i.e. forwarding msg to child instead of parent)
- Timer: Period update of routing tables, messaging…
- Power management

Figure 9: Message flowchart to illustrate the different components for routing.
Multihop Throughput

- Can achieve ~90% packet throughput over 5-6 hops with good protocols.
- Link estimators critical to good multihop. Need to change quickly as link quality changes.

Simulated Results: From Alec Woo, To be Published

Some TinyOS Multihop Stacks

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Comments</th>
<th>Status</th>
<th>Pwr Mng</th>
<th>Authors</th>
<th>TinyOS Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surge</td>
<td>Used in TinyDB, GSK.</td>
<td>Being replaced by Alec Woo</td>
<td>none</td>
<td>UCB</td>
<td>apps/surge</td>
</tr>
<tr>
<td>DSDV</td>
<td>Mote version of standard algorithm. Good results.</td>
<td>Released and stable</td>
<td>***</td>
<td>Intel</td>
<td>contrib/hsn</td>
</tr>
<tr>
<td>TinyDiff</td>
<td>Now deployed in James Preserve.</td>
<td>Doesn’t support mote base station (1-2 months)</td>
<td>*</td>
<td>UCLA-CENS</td>
<td>contrib/TinyDiff</td>
</tr>
<tr>
<td>Alec Woo</td>
<td>PhD research. A lot of work on estimators. May be best when done.</td>
<td>Should be released in next few months.</td>
<td>***</td>
<td>Alec Woo UCB</td>
<td>Not released yet</td>
</tr>
</tbody>
</table>

* Power management implementation in progress
** May be released with power management
*** Currently using long radio preambles