On Heterogeneous Neighbor Discovery in Wireless Sensor Networks

Hema Nair
Spring 2017
Outline

• Background
• Design Goals
• Problem Formulation
• ND Protocol Types
• HEDIS
• TODIS
• Performance Evaluation
• Conclusion
• Recommendations for Future Work
• References
A Brief Introduction to Neighbor Discovery
Find the Neighbors to Form a Network

For Power Conservation:

Keep radios off, turn on periodically
Background

Previous Work:

• Protocols not fine-grained to support heterogeneous duty cycles of nodes.

• Clock synchronization is difficult in distributed system: clock drift exists; neighbor discovery must be done asynchronously.
Design Goals

1. Guarantee neighbor discovery within a reasonable time frame. Minimize discovery latency.
2. Minimize the number of time slots for which the node is awake, to save energy.
3. Match the nodes’ awake-sleep schedules with their heterogeneous battery duty cycles as closely as possible (ie. Finer duty cycle granularity).
Problem Formulation

Neighbor Discovery (ND) Protocol determines the Pattern by which each Radio Turns **ON** or **OFF**

**Duty Cycle**: Percentage of 1 period in which a sensor/radio is active

- **Node A’s DC** = 1/6; **Node B’s DC** = 2/9
Problem Formulation

ND Protocol determines the Pattern by which each Radio Turns **ON** or **OFF**

Node A (6)  
\[
\begin{array}{cccccccccccc}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\end{array}
\]

Node B (2, 9)  
\[
\begin{array}{cccccccccccc}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
\end{array}
\]

A viable neighbor discovery protocol must either implement clock synchronization or be robust in the face of clock drifts.
Neighbor Discovery (ND) Protocol Types

• Quorum Based Protocols
  ▪ Exploits the geometry of a 2-D array

• Co-primality Based Protocols
  ▪ Exploits the Chinese Remainder Theorem (CRT)

• The **Chinese Remainder Theorem** is a theorem of number theory, which states that, if one knows the remainders of the division of an integer $n$ by several integers, then one can determine uniquely the remainder of the division of $n$ by the product of these integers, under the condition that the divisors are pairwise coprime.

• In number theory, two integers $a$ and $b$ are said to be **relatively prime, mutually prime, or co-prime**, if the only positive integer that divides both of them is 1. This is equivalent to their greatest common divisor (gcd) being 1.
Quorum ND Protocols

Example:

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Period = 16
Duty Cycle = 7/16

- Arrange time slots into a matrix formation
- Choose 1 row and column as the “on” slots
  - This ensures that nodes operating on the same duty cycle (ie. Homogeneous nodes) will discover each other within bounded time

- Example: Searchlight Protocol
Co-Prime ND Protocols

**Theorem**: A Co-primality based neighbor discovery protocol guarantees discovery for any two nodes for any amount of clock drifts if the associated integer sets (time slots) of the nodes in this network satisfy the co-prime pair property, with the worst-case discovery delay bounded by the product of the two smallest co-prime numbers, one from each set.

Example: Disco Protocol
Co-Prime ND Protocols

• Example from the theorem: 2 nodes with wakeup schedules of 3 and 2, respectively

A: 3

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

B: 2

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Time drift by 1 time slot

• In reality, each node chooses 2 different numbers and combine their wake-up slots
  ▪ Avoids the problem of 2 nodes choosing the same number: problem where co-primality compromised.

• Disco Protocol (only choose prime numbers)
TWO NEW PROTOCOLS PROPOSED:

**HEDIS (QUORUM) VS TODIS (CO-PRIME)**
HEDIS - The Optimized Quorum Protocol

• **HE**terogeneous **DIS**covery Protocol
• Asynchronous periodic slot-based protocol
• Uses \( n \times (n-1) \) time slots
  - \( n \) depends upon the desired duty cycle \( \delta \)
  - \( n \approx \frac{2}{\delta} \)
  - Duty cycle granularity: \( \frac{2}{n} \)
  - The \( n \)'s must be of the same parity to guarantee discovery
    - Protocol dictates all odd or all even
  - Example: \( n \): even protocol, \( \delta = 7\% = 7/100 \)
    - \( \Rightarrow \frac{2}{(7/100)} = 200/7 \approx 28 = n \)
    - In this case, \( 2/n = 2/28 \approx 0.071 \)
  - HEDIS guarantees neighbor discovery within bounded latency for any two nodes with the same-parity parameters \( n \) and \( m \), given any amount of clock drift between their schedules. The average discovery latency is \( O(nm) \).
**HEDIS Example**

\[ n_1 = 4: 4 \times 3 \text{ slots} \]

\[
\begin{array}{cccc}
0 & 1 & 2 & 3 \\
4 & 5 & 6 & 7 \\
8 & 9 & 10 & 11
\end{array}
\]

Red and Blue slots represent Anchor and Probing Slots during which the nodes are awake.

\[ n_2 = 6: 6 \times 5 \text{ slots} \]

\[
\begin{array}{ccccccc}
0 & 1 & 2 & 3 & 4 & 5 \\
6 & 7 & 8 & 9 & 10 & 11 \\
12 & 13 & 14 & 15 & 16 & 17 \\
18 & 19 & 20 & 21 & 22 & 23 \\
24 & 25 & 26 & 27 & 28 & 29
\end{array}
\]
TODIS – The Optimized Co-Prime Protocol

• **Triple-Odd DIScovery Protocol**
• Duty cycle granularity too high if only prime numbers are chosen to ensure co-primality (DISCO)
• Observation: all prime numbers (except for 2) are odd
  ▪ Odd integers are likely to be co-prime
• Each node chooses consecutive odd integers
  ▪ Example: \( N_a = \{33, 35\}; N_b = \{5, 7\} \)
    - \( \gcd(35, 5) = 5; \gcd(35, 7) = 7 \) => Not co-prime
    - \( \gcd(33,5) = 1 \) => co-prime; \( \gcd(33, 7) = 1 \) => co-prime
TODIS

- Two consecutive odd integers does not guarantee co-primality for many number combinations less than 100: {33, 35}, {75, 77}
  - gcd > 1
- Three consecutive odd integers will do it.
  - Smallest counterexample: {1600023, 1600025, 1600027} and {2046915, 2046917, 2046919}
- Supports duty cycle of $\frac{3(n^2-n-1)}{(n(n^2-4))} \approx \frac{3}{n}$
PERFORMANCE EVALUATION
# A Comparison of ND Protocols

<table>
<thead>
<tr>
<th>Protocol Name</th>
<th>Parameter Restriction</th>
<th>Average Discovery Delay</th>
<th>Supported Duty Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISCO</td>
<td>Prime $p_1, p_2$</td>
<td>$O(p_1p_2)$</td>
<td>$1/p_1 + 1/p_2$</td>
</tr>
<tr>
<td>U-Connect</td>
<td>Prime $p_1, p_2$</td>
<td>$O(p_1p_2)$</td>
<td>$(3p_1+1)/2p_1^2$</td>
</tr>
<tr>
<td>Searchlight</td>
<td>Power multiples of $t_1, t_2$</td>
<td>$O(t_1t_2)$</td>
<td>$2/t_1^2$</td>
</tr>
<tr>
<td>HEDIS</td>
<td>Same parity $n, m$</td>
<td>$O(nm)$</td>
<td>$2/n$</td>
</tr>
<tr>
<td>TODIS</td>
<td>Parity: Odd $n, m$</td>
<td>$O(nm)$</td>
<td>$3(n^2-n-1)/(n(n^2-4)) \approx 3/n$</td>
</tr>
</tbody>
</table>
Duty Cycle Matching - Errors

Small Duty Cycles: 1, 1/2, 1/3, ..., 1/100

Large Duty Cycles: 0%, 1%, 2%, 3%, ..., 100%
Cumm. Distribution Functions (CDFs) of Discovery Latency
Conclusion

• The paper explores the two main approaches of designing a heterogeneous ND protocol: quorum and co-primality
• HEDIS and TODIS proposed, a quorum and co-primality protocol respectively
• Both can match battery duty cycles at a high granularity (2/n and ~3/n)
• HEDIS performs better than TODIS in discovery latency
Recommendations for Future Work

Novel Information

• Co-primality: Weakness:
  - Two consecutive odd integers does not guarantee co-primality for many number combinations less than 100: 
    - \{33, 35\}, \{75, 77\}
    - \(\text{gcd} > 1\)
  - Co-primality requires 3 consecutive odd integers for constructing the wake-up schedule.

• Need to address energy consumption in Performance Evaluation.
References