Pattern Matching and Detection in Extremely Resource-Constrained Wireless Sensor Networks

Michael Zoumboulakis

Birkbeck College, University of London

March 2013
Pattern Matching and Detection in Extremely Resource-Constrained Wireless Sensor Networks

Outline

1 Motivation
   - Introduction
   - Examples
   - Characteristics of the Problem
   - Justification
Outline

1 Motivation
   - Introduction
   - Examples
   - Characteristics of the Problem
   - Justification

2 Our Approach
   - Symbolic Conversion
Outline

1 Motivation
   ■ Introduction
   ■ Examples
   ■ Characteristics of the Problem
   ■ Justification

2 Our Approach
   ■ Symbolic Conversion

3 Contributions
   ■ Efficient Pattern Matching and Detection in WNSs
   ■ Matching Algorithms
Outline

1. Motivation
   - Introduction
   - Examples
   - Characteristics of the Problem
   - Justification

2. Our Approach
   - Symbolic Conversion

3. Contributions
   - Efficient Pattern Matching and Detection in WNSs
   - Matching Algorithms

4. Spatial Detection
Outline

1 Motivation
   - Introduction
   - Examples
   - Characteristics of the Problem
   - Justification

2 Our Approach
   - Symbolic Conversion

3 Contributions
   - Efficient Pattern Matching and Detection in WNSs
   - Matching Algorithms

4 Spatial Detection

5 Results
**Pattern**: An ordered list of sensor observations revealing interesting or unusual activity in the monitored object.
Example
Example
Characteristics

- Lists of sensor observations
- *Pattern Matching* refers to the problem of finding occurrences of one or more user-submitted template patterns.
- *Pattern Detection* refers to the problem of discovering sustained changes or local anomalies.
Why do we need a different pattern matching and detection abstraction for WSNs?

- Matching or detecting patterns in sensor observations is a common requirement in a number of domains.
Why do we need a different pattern matching and detection abstraction for WSNs?

- Matching or detecting patterns in sensor observations is a common requirement in a number of domains.
  - Precision Agriculture, Volcanic Monitoring, SHM, Pervasive Healthcare, Disaster Detection, etc.
Why do we need a different pattern matching and detection abstraction for WSNs?

- Matching or detecting patterns in sensor observations is a common requirement in a number of domains.
  - Precision Agriculture, Volcanic Monitoring, SHM, Pervasive Healthcare, Disaster Detection, etc.
- Only a few in-network approaches are available.
Why do we need a different pattern matching and detection abstraction for WSNs?

- Matching or detecting patterns in sensor observations is a common requirement in a number of domains.
  - Precision Agriculture, Volcanic Monitoring, SHM, Pervasive Healthcare, Disaster Detection, etc.
- Only a few in-network approaches are available.
- The alternative of using thresholds and a programmatic description of patterns as series of logically combined conditions is neither efficient nor scalable.
On-line data-mining

**Central Idea**: Use a well-understood & established algorithm for the online conversion of sensor readings to characters. Reduce matching and detection to a string matching problem.

**Why?**
Our Approach

On-line data-mining

**Central Idea:** Use a well-understood & established algorithm for the online conversion of sensor readings to characters. Reduce matching and detection to a string matching problem.

**Why?**

- Vast collection of algorithms.
- Rich Pattern Detection Capabilities.
- Scalable & Resource-efficient solution.
Symbolic Aggregate approXimation (SAX)
Symbolic Aggregate approXimation (SAX)
Symbolic Aggregate approXimation (SAX)
Contributions

- Developed temporal domain in-network Pattern Matching and Detection algorithms that reduce network communication.
- Demonstrated that the algorithms are suitable for extremely resource-constrained WNSs by integer arithmetic refactoring and dynamic sampling frequency adjustments.
- Introduced an iterative algorithm, based on a geometric computation and a Kalman filter that detects spatial patterns inside the network in a collaborative manner.
Exact and Approximate or Non-parametric Detection

- **Exact** matching is performed when a user is interested in a specific pattern.
- **Approximate** matching is performed when a user is interested in a pattern and other patterns similar to it.
- **Non-parametric** matching is performed when a user does not provide a pattern.
Multiple Pattern Matching (using Suffix Arrays)

Store all interests in a suffix array: speeds up searches & makes the solution scalable.

- . . .
- efcg_haaagdbeabc
- fcg_haaagdbeabc
- gdba_abc
- _ghaaagdbeabc
- _haaagdbeabc
- . . .
Pattern Matching and Detection in Extremely Resource-Constrained Wireless Sensor Networks

Contributions

- Matching Algorithms

Non-parametric Detection
Non-parametric Detection

Train on this
Non-parametric Detection
Many WSN nodes exhibit the *boredom punctuated by panic* property.

Sampling at a constant rate when very little happens is wasteful.

Instead, adjust sampling frequency by continuously monitoring string distances.
The process that generates characters can be viewed as a Markov Chain of desirable order (depending on the app requirements).

Use a portion of the time series that is known to be normal to generate transition probabilities.

Allows to compute probabilities for various sequences.
The Spatial Pattern Detection Problem

We considered the problem of estimating the location and intensity of “dirty bomb” dispersion attacks inside the network. Also known in literature as the *Inverse Problem* where dispersion parameters must be estimated from sensor-observed data.
An example
An example
The spatial detection algorithm was evaluated through simulations and detection accuracy was satisfactory but heavily dependent on node placement and topology.

The work on spatial detection opened a number of future work directions:

- Alter the gas dispersion model and consider multiple sources.
- Evaluate against competitive techniques such as TDOA.
- Implement the algorithm for WSNs.
Selected Results from Emulation

- We compared our temporal detection algorithm to two competitive approaches — Exponential Weighted Moving Average (EWMA) and Real-time Seismic Amplitude Measurement (RSAM).
- We found improved accuracy (true positive ratio of 92.7%) and a reduction of false positives (a ratio of 3.8%).
- We found detection accuracy degrading gracefully in relationship to decreasing SNR.
- For some time-critical applications we established that the average detection latency is low (0.086 secs in a physiological data set).
Selected Results from Deployment

- An integer only implementation showed a factor of ten runtime improvement in comparison with a FP implementation.
- The Dynamic Sampling Frequency Adjustment algorithm resulted in 64% fewer timer ticks in comparison with a fixed sampling scheme.
- The execution profile was sufficiently low to support use in a new class of energy harvesting nodes powered from RFID readers.
### Software Timing Model

#### Arithmetic (Integer)

<table>
<thead>
<tr>
<th>Operation</th>
<th>16-bit</th>
<th>32-bit</th>
<th>64-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increment</td>
<td>6,734</td>
<td>12,824</td>
<td>67,271</td>
</tr>
<tr>
<td>Addition</td>
<td>7,955</td>
<td>15,319</td>
<td>88,301</td>
</tr>
<tr>
<td>Subtraction</td>
<td>7,969</td>
<td>15,310</td>
<td>87,985</td>
</tr>
<tr>
<td>Multiplication</td>
<td>145,020</td>
<td>159,060</td>
<td>316,755</td>
</tr>
<tr>
<td>Division</td>
<td>226,265</td>
<td>709,720</td>
<td>3,519,055</td>
</tr>
<tr>
<td>Remainder</td>
<td>225,375</td>
<td>706,875</td>
<td>3,540,080</td>
</tr>
</tbody>
</table>

#### Bitwise

<table>
<thead>
<tr>
<th>Operation</th>
<th>16-bit</th>
<th>32-bit</th>
<th>64-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>7,965</td>
<td>15,285</td>
<td>88,095</td>
</tr>
<tr>
<td>OR</td>
<td>7,985</td>
<td>15,325</td>
<td>88,360</td>
</tr>
<tr>
<td>XOR</td>
<td>7,955</td>
<td>15,315</td>
<td>88,310</td>
</tr>
<tr>
<td>SHIFT</td>
<td>56,735</td>
<td>62,880</td>
<td>573,665</td>
</tr>
</tbody>
</table>

#### Floating Point Arithmetic

- Assignment & Cast: 2,687,535
- Addition: 3,438,530
- Subtraction: 3,499,920
- Multiplication: 4,841,490
- Division: 4,041,785

#### Array Comparisons & Swaps

- Straight Comp: 104,095
- Comp C Fcn: 83,315
- Comp TOS Fcn: 83,335
- Comp TOS task: 159,055
- Swap C Macro: 93,085
- Swap C Function: 83,320
Thank you!

Questions?