**SAF**: a **S**imilarity-based **A**daptable **F**ramework for Answering Queries in Sensor Networks based on Time Series Forecasting

Daniela Tulone  
UCLA  
tulone@ee.ucla.edu  
(joint work with Sam Madden)
Introduction

- WSN offer a potential to collect large amount of data from remote locations
  - large variety of applications: environmental and industrial monitoring, agriculture...

- Data is often collected at sink, and analyzed
- Tools to facilitate data collection (e.g., Cougar, TinyDB, Directed Diffusion, etc...)

2/1/2007
The problem

Problem: **efficiently** answer queries at sink

```
SELECT nodeID temp > 30
from sensorlist error < 2
```
Our approach

- **Goal:** conserve energy by reducing transmissions
- **Approach:** probabilistic
  - local/global models
  - linear time series → compact representation of data
Our approach

- **Goal**: reduce communication
- **Approach**: probabilistic
  - local/global time series models

SELECT nodeID temp > 30 C from sensorlist conf > 95%
error < 1 C

Answer queries via global model without communicating

2/1/2007
SAF contributions

- SAF: framework for approximate *query answering*
  - Detects *data similarity & anomalies* via *lightweight* time series models

- SAF extends previous approaches in key aspects:
  - Remarkably reduces communication cost
  - Enhances robustness
  - Able to classify *anomalies* (isolated anomalies, persistent variations in the data distribution, periods of noisy data)

- Detects *node similarity* and *clusters nodes* at no additional cost!
  - Definition of similarity based on *data model*, not raw data
  - Efficient clustering algorithm, *provably optimal* in the number of clusters
  - More general than geographic-based similarity

- Analytical evaluation and simulations based on traces of real sensor data
Outline of the talk

- Overview of SAF
  - Local time series models
  - Monitoring/adapting algorithms

- Centralized algorithms
  - Query algorithm
  - Model-based similarity
  - Clustering algorithm

- Simulation results

- Other applications
Outline of the talk

Overview of SAF
- Local time series models
- Monitoring/adapting algorithms

- Centralized algorithms
  - Query algorithm
  - Model-based similarity
  - Clustering algorithm

- Simulation results

- Other applications

2/1/2007
SAF overview

**Phase 1**

Steps Phase 1:

1. Sample values every $T$ time units
2. Learn model

Sample values periodically. Learn model.
SAF overview

Phase 1

Steps Phase 1:

1. Sample values every T time units
2. Learn model
3. Transmit model to sink

Sample values periodically. Learn model.

2/1/2007
Sink stores local models and answers query

1. Copy at the sink in synch with local copy
2. Model has to fit data distribution
SAF overview

Data distribution can change over time!
SAF overview

Data distribution can change over time!

Steps Phase 2:

1. Sample values every T time units
2. Monitor error, detect/classify anomalies

Phase 2

Sample every T time units. Monitor model.

2/1/2007
Data distribution can change over time

Steps Phase 2:

1. Sample values every $T$ time units
2. Monitor error, detect/classify anomalies
3. If needed re-learn model (type)

Phase 2

If needed re-learn local model.
SAF overview

Data distribution can change over time

Steps Phase 2:

1. Sample values every T time units
2. Monitor error, detect/classify anomalies
3. If needed re-learn model (type)
4. Transmit new model to sink

What is M?
Model: Time series

- Time series = set of temporal observations
  - predict current value based on previous history

- Linear time series models → ARMA
  - Benefits: compact representation and bounds, predict complex phenomena... but very costly for sensors!
    - Require long learning phase, computationally intensive!

- Need to adapt ARMA models for WSNs!
  - lightweight model with short learning phase
  - accurate → physical phenomena are complex and change over the time, and sensor data is error-prone
Modeling physical phenomena

Observation: phenomena usually change slowly!

Idea:

- Do not model $F$ over the system lifetime!
- Compute an adaptable lightweight model $M$ capable of accurately predicting $F$ during a (variable) time window $W$

$$M = \{ M_1, M_2, ..., M_i, ... \}$$

Sub-model: AR(q) model

$$F(t) = a_1 F(t-1) + ... + a_q F(t-q) + d \ N(0,1)$$
Why AR models

- Simple, tractable on motes
- Low memory requirements and computational cost
- Cheap and short learning phase
- Solve a linear system in $q$ unknowns in AR($q$)
- Linear and short fingerprint
- Avoid transmissions while learning
- Efficient to detect similarities
- Provide bounds, crucial to monitor the model and detect variations in data distribution, outliers, and inconsistencies
- Design adaptable energy-saving strategies
Our model

\[ F(t) = \text{trend}(t) + X(t) \]

Example:

\[ \text{trend}(t) = e^t + f \]

\[ X(t) = a X(t-1) + b X(t-2) + c X(t-3) + d N(0,1) \]

Stationary AR model!
Our model

\[ F(t) = \text{trend}(t) + X(t) \]

Example: linear + AR(3)
\[
\text{trend}(t) = e \cdot t + f \\
X(t) = a \cdot X(t-1) + b \cdot X(t-2) + c \cdot X(t-3) + d \cdot N(0,1)
\]

Derive lower and upper bounds for \( X(t) \)
\( X(t) \) in \([l, L]\) \( \rightarrow \) \( P(t) \) in \([\text{trend}(t)+l, \text{trend}(t)+L]\)

No periodic communication!
Only when detecting outliers and updating models
On the average 150 transmissions over a week
F(t) = trend(t) + X(t) is a **class** of models

Which type of model, AR(q) and trend function, should the sensor node compute?

**Model metrics:**

- **Model accuracy** → **uncertainty** and **error probability**
  
  Lemma *Prediction error* \(|P(t) - v|\) is smaller than \(e = a \cdot d\), with error probability smaller than \(1/a^2\).

- **Cost of learning/adapting**
  
  - Computational cost and memory storage

- **Communication cost**: cost of sending **new model coefficients**
Model monitoring

- Periodically read value $v$,
- Compute $\text{err}=|P(t) - v|$, $e = a \ast d$

Outliers

Monitor model

Model fits data

Monitor model

2/1/2007
Outline of the talk

- Overview of SAF
  - Local time series models
  - Monitoring/adapting algorithms

Centralized algorithms
- Query algorithm
- Model-based similarity
- Clustering algorithm

- Simulation results
- Other applications

2/1/2007
Answering queries

Query(Q)

for each node i do
  if Q.err > err_i & trend(t) satisfies Q.cond
    add i into query sensorlist L
  else
    add N into sendlist S

Contact nodes in S and unstable list

The query answer is provably correct.

2/1/2007
Benefits of model-based approach

- Flexibility:
  - some models might be more accurate
  - areas of interests
- Dynamic model
- Tunable accuracy
- Detect similarities...

2/1/2007
Data similarity

- Detecting and tracing data similarity is relevant for a number of sensor applications:
  - Redundancy detection → decrease duty cycle, re-task sensors
  - Intrusion and anomaly detection
  - Scientific studies, detect spatial-temporal correlations
    - Seismic or volcanic applications, etc...
    - Temperature isolane

- Clustering in **dynamic conditions** consumes energy!
  - Computing and maintaining clusters requires node **coordination**, leader election etc...

2/1/2007
Data similarity in PAQ

- Exploit geographical correlations
  - nearby sensors are likely to be similar
- Group “similar sensors” into clusters
  - $|v_1 - v_2| < a$

- no leader election
- no need for consistent views
Model-based similarity

- **Idea:** detect similarity not based on raw data but on prediction values
- Node i and j are a-similar if
  \[ |P_i(t) - P_j(t)| < a \]
- Bound prediction value, P(t) in \([l, L]\)
  - Transform a complex problem into 1-dim problem
Advantages of model-based clusters

- Clustering algorithm
  - Efficient, $O(n \log n)$
  - Provably optimal number of clusters
- More general than geographical clusters
  - Larger clusters 2 vs. 4
- Nodes are not aware of membership
  - variations in cluster membership require no additional communication
  - Useful for tracking clusters
  - Sink can dynamically tune parameter $a$
- Suitable for mobile networks!
Other SAF components

- Tunable rate
- Dynamic strategies
- Low duty cycle
- Hierarchical framework (HSAF)
Outline of the talk

- Overview of SAF
  - Local time series models
  - Monitoring/adapting algorithms

- Centralized algorithms
  - Query algorithm
  - Model-based similarity
  - Clustering algorithm

Simulation results

- Other applications

2/1/2007
Simulation-based results

- Trace of real data: 1 month of temperature, voltage, pressure...
- 50 sensor nodes, in-door, air conditioning
Model stability

Time vs. Max. Error and Error, Sensor 45

Time vs. Temperature, Sensor 45

2/1/2007
User-error vs. transmissions

Error Threshold vs. Total Transmissions, Node 45

Error Threshold vs. Model Stability Time, Node 45

2/1/2007
Clustering

- Number of clusters over the system lifetime: 50 sensors, similarity within 1 degree
- Comparison between ideal clusters and our approach: avg. 2.4, avg. 4.2
Model updates vs. user error

Number of model updates over a month using different local models and different error threshold.

<table>
<thead>
<tr>
<th>Error</th>
<th>SAF</th>
<th>Kalman</th>
<th>Appr. Caching</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>1464</td>
<td>2323</td>
<td>2521</td>
</tr>
<tr>
<td>0.3</td>
<td>987</td>
<td>1515</td>
<td>1664</td>
</tr>
<tr>
<td>0.4</td>
<td>782</td>
<td>1218</td>
<td>1245</td>
</tr>
<tr>
<td>0.5</td>
<td>643</td>
<td>931</td>
<td>940</td>
</tr>
<tr>
<td>0.6</td>
<td>570</td>
<td>816</td>
<td>782</td>
</tr>
<tr>
<td>0.7</td>
<td>547</td>
<td>696</td>
<td>644</td>
</tr>
<tr>
<td>0.8</td>
<td>520</td>
<td>620</td>
<td>559</td>
</tr>
</tbody>
</table>
SAF features

- Consumes little energy
  - Few transmissions + low duty cycle
  - Short learning phase
- Suitable for mobile networks
  - More robust to communication failures
  - No additional overhead
- Ability of deriving strong data properties
  - Outlier detection, data inconsistencies, data similarities
- Adaptable strategies
  - Tunable data rate, areas of geographical interest
## SAF vs. previous approaches

<table>
<thead>
<tr>
<th></th>
<th>SAF</th>
<th>BBQ</th>
<th>Ken</th>
<th>Kalmar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptability</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Robust to comm. fail.</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Outlier</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Learning</td>
<td>½ - 1 h</td>
<td>7-15 days</td>
<td>few days</td>
<td>_</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>20160</td>
<td>2880</td>
<td></td>
</tr>
<tr>
<td>Data similarities</td>
<td>Yes</td>
<td>correlations</td>
<td>correlations</td>
<td>No</td>
</tr>
</tbody>
</table>


2/1/2007
Outline of the talk

- Overview of SAF
  - Local time series models
  - Monitoring/adapting algorithms

- Centralized algorithms
  - Query algorithm
  - Model-based similarity
  - Clustering algorithm

- Simulation results

Other applications
Time synchronization

- Time series models can be applied to improve the accuracy and robustness of the clock
  - Reduce the error growth by exploiting information on clock deviation
    - Novel viewpoint
    - Different metrics, clock deviation, cumulative drift rate.

\[
\begin{align*}
\text{time est.} & & 2\Gamma & & t & & 3\Gamma \\
\text{max error} & < \eta & & & & & \\
\epsilon & & T & & T + \Delta H & & \epsilon + \rho \Delta H
\end{align*}
\]
Time synchronization

- **Deterministic methods**
  - Reduce error growth by a factor of 2 or more
  - Refinement of optimality result for external clock synch

- **Probabilistic method (time series models)**
  - Highly flexible

- **Benefits:**
  - Reduce the frequency of clock adjustments → energy saving
  - Robustness: node failures, isolations, network partitions, malicious failures
  - It can work in synergy with other clock synch protocol

Fault detection and diagnosis

- Detect sensor failures
  - Distinguish between faults and unexpected behavior of the phenomenon

- Diagnosis failures:
  - Stack-at, out-of-range, calibration faults, complex failures

- Study geographical and temporal scope of failures
Image sensing and interpretation

- **Cyclops project:**
  - Each nest box is equipped with a camera
  - Goal: detect presence/absence birds, derive information useful for scientific studies

- **Times series forecasting:**
  - Object detection algorithm → model background
  - Derive patterns of bird behavior

- **Other potential applications:** data management in mobile networks, object tracking...
Conclusions

- Proposed SAF framework for approximate query answer and outlier and similarity detection
  - Lightweight and adaptable time series models
  - Showed suitability simple model + monitor

- Main features of SAF:
  - Consume little energy
  - Provide information regarding outliers, variations in the data distribution, periods of data inconsistency
  - Provably correct
  - Adaptable: provides user-controllable error
  - Suitable for mobile networks