# Summary of current research results (09.12.2018) in adaptive sampling

## Demanded Authors

### Abbadi

* Power-aware Query Processing Over Sensor Networks
  + Publishing Year: 2003
  + Contribution
    - Introduction of precision bound framework for for query processing over sensor networks (power-aware queries).
  + Summary:
    - nodes aggregate values they receive to reduce communication cost
    - prediction of values at nodes and sink so not every sensed value needs to be sent
    - all in all nothing really new, and the prediction function is a generic linear regression
  + Related Work
    - S. Madden and M.J. Franklin. Fjording the stream: An architecture for queries over streaming sensor data. International Conference on Data Engineering (ICDE 2002), March 2002.
    - Anantha Chandrakasan Wendi Rabiner Heinzelman and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. HICSS, January 2000.
    - Minimum energy multi-hop routing protocols
      * M. Ettus. System capacity, latency and power consumption in multihop-routed ss-cdma wireless networks. Radio and Wireless Conference (RAWCON ’98), August 1998.
      * Mike Woo Suresh Singh and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. Mobile Computing and Networking, pages 181–190, 1998.
      * Meng T. and Volkan R. Distributed network protocols for wireless communication. In Proc. IEEEE ISCAS, May 1998
      * Timothy J. Shepard. A channel access scheme for large dense packet radio networks. SIGCOMM, pages 219–230, 1996
    - Data Fusion
      * D. Hall. Mathematical techniques in multisensor data fusion. Artech House, 1992.
      * L. Klein. Sensor and data fusion concepts and applications. SPIE Optical Engr Press, 1996
    - LEACH, TAG, COUGAR
    - Interval Sampling
      * C. Olston and J. Widom. Offering a precision-performance tradeoff for aggregation queries over replicated data. In VLDB2000, September 2000.
* BINOCULAR: A System Monitoring Framework
  + Publishing Year: 2004
  + Contributions:
    - BINOCULAR is a monitoring system where users pose continuous queries to monitor the physical environment. Therefore, it is a multi-query processing platform.
    - BINOCULAR models the readings of sensors as a linear system to observe readings of all sensors with a small set of sensor readings. Therefore, it is an energy efficient monitoring system.
    - BINOCULAR improves the quality of the answers of queries by reducing the noise over sensor readings using linear observers.
    - BINOCULAR balances energy consumption among sensors while extending their lifetime
  + Summary:
    - readings can be apporximated by a linear model, thus only a subset of sensors needs to be queried
      * sensors are split into working and sleeping sensors
      * to prevent error accumulation, all sensor nodes need to be queried every D time units
    - that framework is very highlevel with questions remaining unanswered at the end
    - not continued
  + Related Work:
    - Data Filtering:
      * A. Jain, E. Chang, and Y. F. Wang. Adaptive stream resource management using kalman filters. In Proceedings of the 2004 ACM SIGMOD Intl. Conf. on Management of Data, 2004. (
      * C. Olston, J. Jiang, and J. Widom. Adaptive filters for continuous queries over distributed data streams. In Proc.of ACM SIGMOD Intl. Conf. on Management of Data, San Diego, California, USA, June 2003.
    - Adaptive Sampling
      * S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. The design of an acquisitional query processor for sensor networks. In Proceedings of the 2003 ACM SIGMOD Intl. Conf. on Management of Data, pages 491–502. ACM Press, 2003
      * B. Hull, K. Jamieson, and H. Balakrishnan. Bandwidth managment in wireless sensor networks. In Intl. Conf. on Embedded Networked Sensor Systems, Los Angeles, California, USA, November 2003.
      * I. Lazaridis, Q. Han, X. Yu, S. Mehrotra, N. Venkatasubramanian, D. V. Kalashnikov, and W. Yang. QUASAR: Quality aware sensing architecture. ACM SIGMOD Record, 33(1):26–31, Mar. 2004.
      * A. D. Marbini and L. E. Sacks. Adaptive sampling mechanisms in sensor networks. In London Communications Symposium, London, UK, 2003.
      * J.-Y. Pan and S. S. amd Christos Faloutsos. Fastcars: Fast, correlation-aware sampling for network data mining. In GLOBECOM 2002 - IEEE Global Telecommunications Conf., pages 2167– 2171, Taipei, Taiwan, November 2002.
    - Load Shedding
      * N. Tatbul, U. Cetintemel, S. Zdonik, M. Cherniack, and M. Stonebraker. Load shedding in data streams. In 29th Intl. Conf. on Very Large Data Bases (VLDB), pages 309–320, Berlin, Germany, September 2003.

### Hauswirth

* Transmitting and Gathering Streaming Data in Wireless Multimedia Sensor Networks within Expected Network Lifetime
  + Summary:
    - a routing algorithm for short lived WSNs of multimedia data
    - e.g. volcano eruption
  + Contributions:
    - first routing algorithm that focusses on providing multimedia streaming data in WSNs
    - adjustment of transmission radius of sensor nodes for minimzing transmission delay and maximizing streaming volume within an expected network lifetime
    - guaranteed to find routing paths if they exist
    - optimize routing path to find the one with the least amount of hops
    - better solution to hole bypassing
      * Hole-bypassing: Dynamic holes may occur if several nodes in a small area overload due to multimedia transmission. Efficiently bypassing dynamic holes is essentially necessary
* Related Work
  + Geographic Routing in Wireless Multimedia Sensor Networks
    - TPGF: geographic routing in wireless multimedia sensor networks is newer (2010) and more cited (187)
  + Transmitting Streaming Data in Wireless Multimedia Sensor Networks with Holes
  + Cross Layer Optimization for Data Gathering in Wireless Multimedia Sensor Networks within Expected Network Lifetime (2010, 18 cit) is a continuation of the work

## Quick Overview of Interesting Papers I have not yet summarized

### Decentralised Control of Adaptive Sampling in Wireless Sensor Networks

* Contributions:
  + information metric based upon Fisher information and Gaussian process regression to express information content of sensors observations
  + Decentralized Control Algorithms which represent trade-off between computational cost and optimality (Quality of Data)
* Summary
  + Algorithms are fed with information metric and exploit temporal correlation within the data from a sensor to select the most informative sampling points
  + With highly time correlated data which is changing slowly, low sampling frequencies are possible

### ASAP: An adaptive sampling approach to data collection in sensor networks

* Contributions
  + ASAP: use a dynamically changing subset of the nodes as samplers such that the sensor readings of the sampler nodes are directly collected, whereas the values of the non-sampler nodes are predicted through the use of probabilistic models that are locally and periodically constructed
* Summary:
  + First, sensing-driven cluster construction is used to create clusters within the network such that nodes with close sensor readings are assigned to the same clusters.  
    Second, correlation-based sampler selection and model derivation are used to determine the sampler nodes and to calculate the parameters of the probabilistic models that capture the spatial and temporal correlations among the sensor readings.  
    Last, adaptive data collection and model-based prediction are used to minimize the number of messages used to extract data from the network

### An Optimization Approach for Adaptive Monitoring in IoT Environments

* Contribution
  + We propose a novel approach that updates the choice of metrics to be monitored and their optimal monitoring frequencies with event-based and/or environmental-based triggers while taking into account resource constraints
* Continuation of work in:
  + Monitoring services in the Internet of Things: an optimization approach

### An Energy Aware Adaptive Sampling Algorithm for Energy Harvesting WSN with Energy Hungry Sensors

* Contributions
  + An energy aware ASA optimized for energy harvesting nodes achieving self-sustainability with power hungry sensors is presented.
  + Precise energy model for a WSN network with power-hungry sensors and energy harvesting capabilities.
  + Evaluation of the proposed approach within two application domains, namely SHM and monitoring of beehives.
  + Simulation evaluation using long term in-field measurements and comparison with the ASA (without energy awareness) and the fixed sampling rate (conventional method) in terms of energy durability.
* Summary
  + “When the battery level is above a critical level, defined by the user (Xlevel), the proposed algorithm can use the sampling rate of any ASA. As the battery level goes below the critical level, the algorithm becomes more energy conservative by reducing the sampling rate. Thus, the node manages its activity in the network according to its energy levels.”

### Energy minimization by exploiting data redundancy in real-time wireless sensor networks

* A communication [scheduling mechanism](https://www.sciencedirect.com/topics/computer-science/scheduling-mechanism) for a duty cycled data collection tree network.
* Contributions:
  + We propose a lightweight node level slack determination scheme by exploiting approximate, bounded-loss data collection in sensor networks. Each node independently decides whether the currently sampled data is significantly different from the previous reading. If the data is not significantly different, it does not need to be transmitted. This smarter way of not transmitting a repetitious [data packet](https://www.sciencedirect.com/topics/engineering/data-packet) has two advantages: firstly, communication [energy savings](https://www.sciencedirect.com/topics/engineering/energy-conservation) and secondly, the unused time slot (slack) can be used by other nodes to reduce their communication energy.
  + We propose smart slack distribution schemes. The proposed schemes detail the effective utilization of the resource (slack) left unused by nodes having redundant message. The schemes work in a distributed manner such that [control overhead](https://www.sciencedirect.com/topics/computer-science/control-overhead) is minimized while the energy savings can be increased.

### ASample: Adaptive Spatial Sampling in Wireless Sensor Networks

* Prevent under or over sampling of signals in regions
* Contribution
  + This paper develops an efficient distributed technique for identifying under- and over-sampled regions. Our proposed approach (ASample) utilizes current measurement and SNs placements, (a) to insert new sampling locations or (b) to remove redundant spatial samples, in order to exactly meet the accuracy requirements. Our approach is holistic as it is valid for both over-sampling and under-sampling profiling. Accordingly, it simplifies network-level decisions on moving nodes from the over-sampled regions to the under-sampled ones and for tuning the network sampling resolution according to the changes in the monitored phenomena

### Optimal Rate Schedules with Data Sharing in Energy Harvesting Communication Systems

* Contribution
  + This paper introduces a rate scheduling problem for energy harvesting wireless devices that transmit equired data of requests with the goal of minimizing the completion time. We exploit the data sharing among data requests from the platform, e.g., a participatory sensing system, to actively enhance the energy utilization of the wireless device.
  + We first study a closely related min-energy problem that aims to minimize the energy consumption within a given deadline while transmitting all required data. By decomposing the original problem into two simplified known sub-problems, we derive the optimal offline algorithm BOTTLENECK-SELECT that minimizes the energy consumption or determines that no feasible solution exists within the given deadline.
  + Then, by adopting BOTTLENECK-SELECT as a building block, we develop an optimal offline algorithm for the completion time minimization problem. The idea is to use BOTTLENECK-SELECT to narrow down the lower bound and upper bound of the minimum completion time, and then precisely locate the optimal solution.
  + We also design an event-driven online heuristic algorithm to deal with the dynamic energy and request arrivals. Simulation results validate that its performance is close to the optimal offline solution.
* Continuation of work
  + On Minimizing Sensing Time via Data Sharing in Collaborative Internet of Things

## Algorithms I already summed up

### AdaM: an Adaptive Monitoring Framework for Sampling and Filtering on IoT Devices

**Adaptive sampling** is the process of dynamically adjusting the sampling rate to the current metric evolution, such that when stable phases in a metric stream are detected, the sampling rate is reduced to ease processing and energy consumption

**Adaptive filtering** is the process of dynamically adapting the filter range to follow the metric evolution without requiring beforehand for users to guess what filter range should be applied

AdaM

* runs on source device
* Goal:
  + find a sampling interval which produces a metric stream which differs from a metric stream with a minimum fixed sampling interval by less than a user specified imprecision value
  + balance between efficiency and accuracy
  + same thing with filter ranges: a fixed filter range would be inefficient and thus an adaptive filter range should be found
    - same algotithm/process as with the sampling interval

Arguments against fixed sampling intervals

* small interval: a lot of data is generated and has to be send through the network and processed which is linked with higher resource expenditure
* big interval: sudden events may be not sensed which would fail to produce meaningful insights

Metric stream evolution is computed by a moving average

* algorithm uses a expanded version of Exponential Weighted Moving Average (EWMA) called Probabilistic Exponential Weighted Moving Average (PEWMA) which detects sudden spikes after long stable phases better then EWMA
* probabilistic means, that the evolution of the metric stream has a probability to follow a modelled distribution

User interaction in algorithm

* aggressiveness in approach with lambda as a multiplicity factor
* alpha as the weighting factor in PEWMA

Testing of algorithm

* testing was conducted using the MAPE (Mean Absolute Percentage error), cpu-cycles, outgoing network traffic and energy consumption
* tested on: AdAM, L-SIP, FAST and i-EWMA
* filtering in AdAM produces near to no overhead
  + network traffic reduction of 69% is achieved
* Core statement:
  + "AdaM succeeds in reducing data volume by 74%, energy consumption by at least 71%, while accuracy is, in all cases, greater than 89% and with filtering enabled, greater than 83%."
* FAST samples more aggressivly, meaning it uses larger sampling intervals
  + this reduces network traffic and energy expenditure
  + however a great deal of accuracy is sacrificed

### ADMin: Adaptive monitoring dissemination for the internet of things

Definitions:

*Adaptive Dissemination (similar (or the same?) to adaptive filtering)*

* process of not sending sensed values when they have a slight aberration from predicted values in order to reduce communication overhead

the paper takes the AdaM algorithm and extends it with data seasonality handling

* "Seasonality is defined as the tendency of the metric stream to exhibit behavior that repeats itself every L periods (e.g., hourly, daily)"
* trend and seasonality knowledge can be added to the estimation process, so that the algorithm can adapt to unexpected abrupt and volatile changes in the metric stream

Evaluation results

* tested features are:
  + shift detection accuracy
    - true positives, i.e. number of correctly detected shifts
    - false positives, i.e. number of false alarms
  + shift detection delay
    - time difference between detecting shift and actual time of occurence
  + data volume reduction (exception ADWIN, does not reduce data) and accuracy on receiver side
  + total energy consumption
* all metrics are compared to algorithms (or frameworks)
  + ADWIN, G-SIP, LANCE
* Shift Accuracy Detection
  + ADMin is on top with ADWIN following suit
  + rest falls behind with significantly varying between traces
    - G-SIP and LANCE have both a high share of false positives while admin, enriched with seasonality knowledge, keeps the false alarm ratio under 10%
* Shift detection time
  + ADMin outperforms other algorithms by at least 29%
  + when tested on traces with irregular seasonality, ADMin enriched with optimal seasonality from ComCube, reduces shift detection time by at least 67%
* Rest
  + ADMin is able to reduce data volume by at least 71% which accounts for a reduction in energy consumption of at least 83%
    - ADWIN saples to much and the other two algorithms have a high false alarm ratio, thus enabeling the network more often and consuming more energy
  + accuracy at the receiver to at least 86%, increasing to at least 91% when seasonality behavior is acknowledged by the estimation model.

### SIP: The Spanish Inquisition Protocol—Model Based Transmission Reduction for Wireless Sensor Networks

Basics

* data is encoded using a model which is shared by the transmitter and the sink node
  + data is sent as a vector to the sink. Further data points are predicted by the sink if the source node does not send any data
* data is only transmitted when the sink predicts the data with error value > *epsilon*
  + source node keeps track of what the sink node knows and sends an update message if prediction error is too high
* no clock snychronisation is required
* algorithm is model agnostic
  + a model and a filter have to be chosen
  + like for models: piece-wise linear
  + for the filter: depends on the model used and on the requirements of the application
    - basic filtering: Exponetionally Weighted Moving Average
      * good if processing and memory overhead need to be minimized
    - Others: LMS, NLMS, Kalman Filter (KF) or Extended KF (EKF)
      * more computationally costly and harder to tune (source)

### L-SIP: Edge Mining the Internet of Things

Three variants of SIP are presented in this paper: L-SIP, classicAct and Bare Necesseties (BN)

Edge Mining:

*"Processing of sensory data near or at the point at which it is sensed, in order to convert it from a raw signal to contextually relevant information."*

Assumption of paper:

*"a WSN is a network of individual smart sensing devices that are discrete sources of data and information, and that transmit over a shared ad-hoc wireless network."*

Other Aspects:

* data reduction on bit and packet level
  + compression focusses on number of bits to be transmitted
  + edge mining focusses of number of packages to be transmitted
  + *"saving packets is more important than saving bits"*
    - overhead for package receival is substantial (checksums and headers checking, radio powering up and down, packet aknowledgement (somtimes), energy expenditure when forwarding package by other nodes)
* privacy is enhanced as signals are transformed at the point of acquisition
* important for the logativity of a network using L-SIP, is the network layer
  + for the multihop transport protocol the authors recommend Collection Tree Protocol (CTP) and for the MAC layer the Low Power Listenting Protocol
  + however better results (energy savings) are achieved (experiments by the authors) with Synchronous interval listening approaches such as TSMP

L-SIP

* reconstruct original signal in sensing applications with some error bound
  + keyword: timeseries forecasting
* encodes the state as a point in time, the value and the rate of change
* estimate new state (different methods of state estimation can be used, like Exponentially Weighted Moving Average (EWMA), Normalised Least Mean Squares (NLMS) or a Kalman Filter (KF). EWMA or double EWMA (dEWMA)

dEWMA filtering:

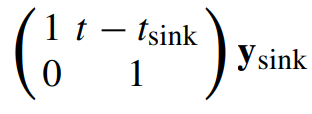
x' 1 ← αz + (1 − α) (x1 + x2dt)

x' 2 ← β x' 1 − x1 /dt + (1 − β) x2

(Update filtered estimates of value x1 and rate of change x2. dt denotes the time interval between samples.)

predict sink state

y' ←



 (linear extrapolation)

simplify

y ← x (no simplification)

eventful?

yes if |y' 1 − y1 | > ε

(The measurement is eventful if the value estimate y1 differs from the prediction y' 1 by at least some threshold ε.)

ClassAct

* human posture recognition approach
* transmits only posture (i.e. standing, walking, running, sitting) but not the accelerometer readings
* event detection is based on when that posture changes

BN

* summary of relative time spent in different postures states
* summarizes state estimator over time in terms of a histogram
* event detection here is when the distribution changes significantely

G-SIP

* generalized form of SIP
* adds heartbeat threshold
  + now transmits reading not only if model of sink node does not predict value with certain error marging, but also when a specific time elapsed since the last transmission
* simplifies state vector x (to be transmitted signal in original SIP algorithm) to y
  + *"state vector x may contain much more than needs to be transmitted, it is often useful to generate a simplified form y."*
* additional sequence number is transmitted to potentionally identify lost packages

### LMS: An Adaptive Strategy for Quality-Based Data Reduction in Wireless Sensor Networks

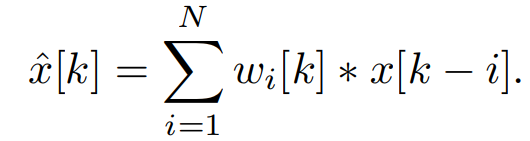
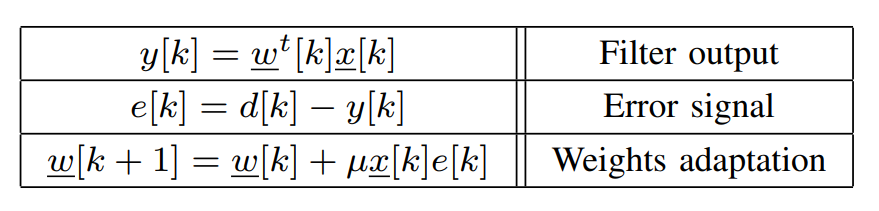
Basics:

* also a prediciton based filtering concept
* node reads values and compares them against it's estimate
* when the error is smaller than the user set error (emax), then the reading is discarded and no data is sent to the sink node
  + assumption, that no values are lost due to network errors
* the sink node then interpretes that it's prediction is good and takes in the predicted value into it's reading vector
* the source node also takes the predicted value into it's reading vector thus keeping both models in sync without synchronizing

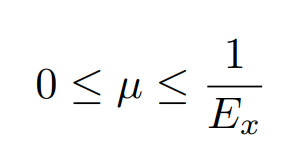
Network model

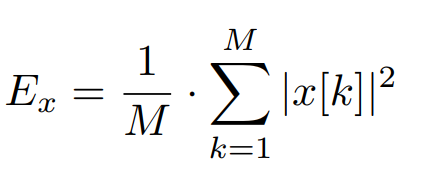
* algorithm can be used with different network/topology models

Algorithm

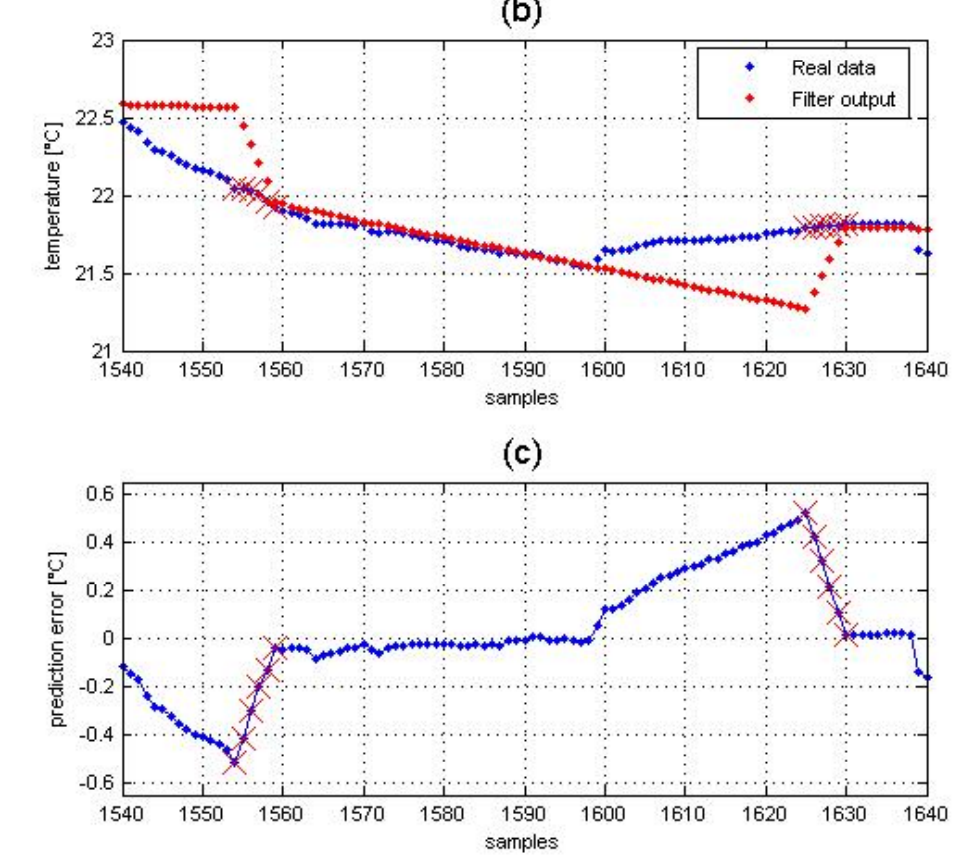
* is an adaptive filter
  + does not require a-priori knowlegde on the signal statistics
* the lms algorithm takes the previous input value and uses the current input value as the reference signal
* the filter weight is set initially to 0 and is adapted with each iteration on the basis of the actual prediction error
* 
* step size µ and filter lenght need to be set in the beginning
  + the filter length is an important parameter for the computational load
  + it exists an theoretically computable optimal filter length; increasing filter length beyond the optimal length will lead to performance loss
* lms ensures the same outcome of different instances of the filter when the inital weight, step size and filter length are the same for all instances
* workflow
  + 

Algorithm at work in the sensor

* sensor nodes have 3 modes: initialization, normal and stand-alone
  + the initialization mode
    - is started once when receiving a user query
    - node starts collecting data and reports it to the sink
    - sink and node compute estimations of the step size
      * step size has to converge
      * 

* 
* Ex is the mean input power and M the number of interations used to train the filter
* a approximation of Ex is computed because Ex is time varying
* a optimal filter length can be computed, too, however the paper reserves this work for future research
* normal mode
  + node and sink start to perfom prediciton on collected values
  + node keeps reporting the sensed value to the sink when the predicted value does produce an error higher then the user specified error
  + both sink and node run the prediciton algorithm on the sensed values to update their weights
  + once the prediction error goes below the error margin the node switches to stand alone mode
* stand alone mode
  + node collects data and computes prediction
  + when prediction error is lower than boundry, the sensed value gets discarded and the predicted value is fed into the filter
    - filter state of node thus remains consitent with filter state of sink
  + sink assumes when not receiving any values that it's prediction is good enough and keeps on predicting
    - a heatbeat or whatever needs to be possibly implemented, so that the sink nows the node still exists
    - assumption that no signals are lost
  + feeding filter with own prediction causes an error of 0
    - the filter wont get updated saving ressources
  + when error gets too high, node switches back to normal mode
* an outlier detection mechanism could be implemented
  + a prediction whose error is larger then some threshhold (not the normal error threshhold emax the user sets) gets discarded as an outlier

Evalutaion



* because of the tracking capability of lms, the error does not exceed it's boundries and fluctuates between zero

Open issues

* communication model -> signal loss would let the sink produce predictions above the error boundry and set the filters of node and sink out of sync
* node failures -> sink has to know when a node is shutdown or whatever
  + => heartbeat
* spatial prediciton -> cluster heads receiving data from different nodes
  + more potential to optimize

### Model-Driven Data Acquisition in Sensor Networks

Key point: A correlation-aware probabilistic model is integrated into a database of sensors. User queries the database and the model tries to answer the query with predictions with error boundries set by the user. The model tries to query the minimal amount of sensors by exploiting temporal, spatial and other correlations. If error is too high, additional sensors are queried.

My two cents:

* paper kinda bashes tinydb and cougar as outdated and presents itself as an advancement
* on a side note, paper was curated (?) by madden so it kinda wants to expand the fields tinydb didn't address enough

Key insights form paper:

1. Misrepresentation of Data
   * sensornets are not sensor databases as they do not represent a real life model of the environment, but show only a fraction, i.e. a descreet point in time, of the values of a phenomena
   * thus only samples of the observed phenomena are really gathered
2. Inefficient approximate queries
   * a completist’s approach is not optimal for sensor data acquisition, as sensor can only deliver approximate results
   * thus, continous querying is not efficient as it does not improve the result substationally

Contribution from paper:

* models for real world processes are introduced to deliver answers to queries and not bother the sensor
* sensor is only used when the models do not answer the query suffeciently good
* spacial coherency, i.e. a single sensor reading can improve the model induced accuracy for another sensor if it is in some proximity to the read one
* prototype BBQ -> uses model based on time-varying multivariate Gaussians
  + authors argue, that their approach is general with regards to the model if it is more or less complex

Basics of algorithm:

* The queries include error tolerances and target confidence bounds that specify how much uncertainty the user is willing to tolerate
* system decides, based on the model, what sensors to ask
  + afterwards an observation plan (how and in which order sensors are queried) is build
  + the readings are used to update the model and answer the query in the specified confidence intervals
* correlations between different attributes can be captured
  + e.g. when voltage and temperature are highly correlated and sampling temperature is much more energy consuming than sampling voltage, then sampling voltage and inferring the value for the temperatur from a model makes much more sense
  + spatial and temporal corelations can be used too

Experiments of algorithm with BBQ as model:

* User would ask a query, with a fraction of results 1 - delta not deviating by an error margin epsilon
  + i.e. users ask a query with error margin 0.5 degrees (temperature) with 95% confidence interval
* Results by BBQ are compared to tiny db and a approximate caching approach
  + approximate caching: energy costs induced due to communication are lowered because values that do not change a lot are not send through the network
  + however, the cost of sensing is not reduced because data is acquired continuosly
* with reasonable error boundary and confidence interval, energy reduction per query can be reduced by a factor of 40 (5.4 J to 150 mj)

Possible extensions are given in the paper (by the authors):

* conditional plans
* more complex models
* outliers (are not easily detectable with a correlation based model as outliers are inherently uncorrelated events)
* support for dynamic networks (network topology can change, e.g. sensors get added or removed)
* continuous queries (nodes push readings periodically at specified times )

Related work

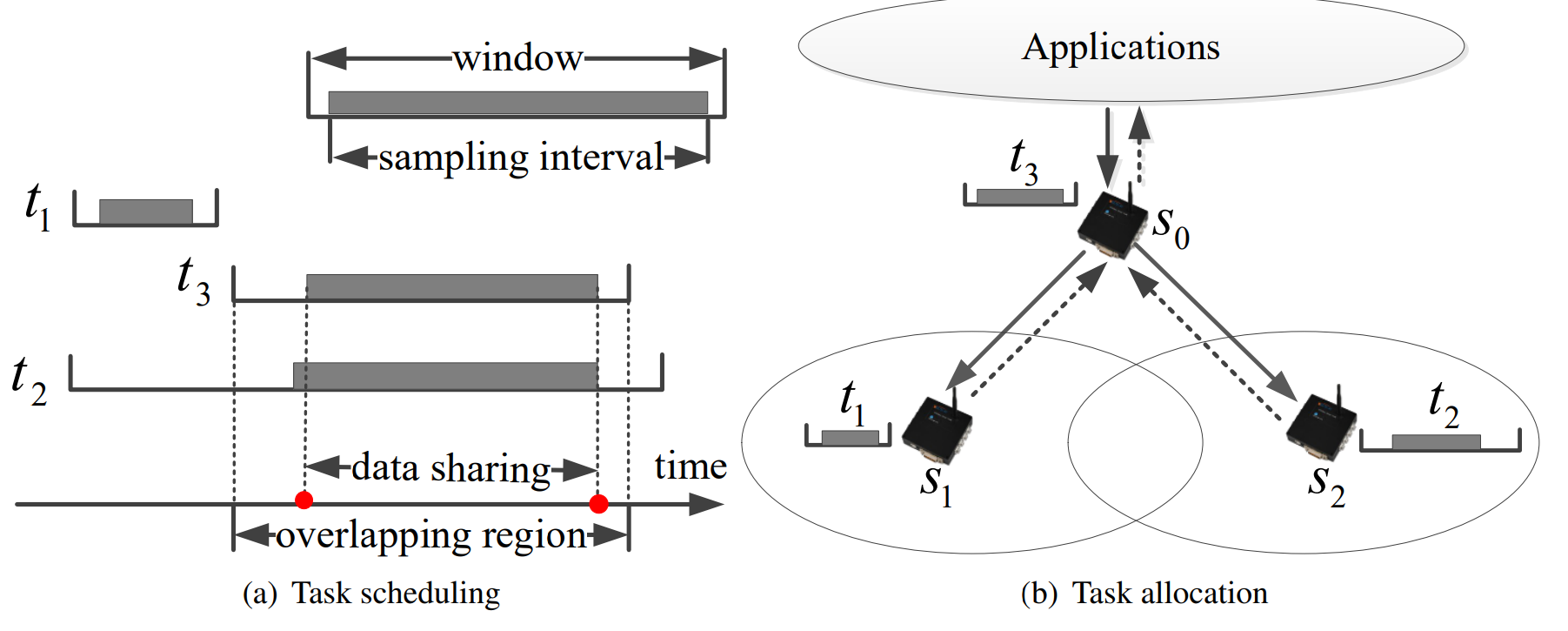
* a lot of sources, however, all from the 2004 period

### CATS: Cooperative Allocation of Tasks and Scheduling of Sampling Intervals for Maximizing Data Sharing in WSNs

Overview:

* two algorithms are presented: CATS and COMBINE
* COMBINE: maximize data sharing among overlapping tasks
* CATS: jointly allocates tasks and schedules sampling intervals for maximazing data sharing in entire network

Basics:

* Task allocation:
  + applications would inject sampling tasks into the sensor network
    - the tasks have a window and a sampling interval
    - the sampling interval can be adjusted dynamically in the window
    - 
  + because t2 and t3 have an overlapping region, t3 should be allocated to s2 thus maximizing data sharing
* Task scheduling
  + the process of scheduling tasks in a node, while task allocation schedules tasks in the network
  + the authors underline that task scheduling is presented in related work, however, only in combination with task allocation is task scheduling possible in wsns

COMBINE:

* basically combine is used to find combinations between overlapping tasks
* If a task set which consists of more than one task is sent to the network by an application
* all possible combinations (sampling window has to be overlapping) of tasks are found and the value of the data sharing between them is computed
  + with value of data sharing is meant the length of the overlapping windows
* the maximum value of the combinations is found and the original tasks in the combination are replaced by the combination
* steps above are repeated until no overlapping tasks are found
  + so original tasks are tried to be combined with already found combinations
* a task set is returned which has no overlapping tasks
* everytime a combination is found, a quad list is produced and saved in runtime
  + when memory is heavily restricted in a network an adjusted algorithm can be used
  + i.e.: same algorithm, but no quad list is used, instead, found combinations are included into the task list and the original elements of the combination are removed from the task list
  + process is repeated until no sharing potential is left and the new task set is returned

Prune:

* an algorithm which is straight forward designed
* pruning is a suboptimal because it allocates tasks to candidate sensors only once and removes the combinations with the lowest data sharing value

CATS:

* note: there is no step 1 and step 2 when using those algorithms
  + the authors wrote the paper in a dedacting manner, i.e. combine is presented first and cats is used afterwards with combine because prune turned out to be not optimal
  + thus cats is the actual algorithm presented here which uses  combine iteratively
* do COMBINE but on a global scale
* for that, a global quad list is maintained where one quad is a combine operation
* locate the quad with the maximum value of data sharing
* assign the overlapping tasks in the best quad to r out of k candidate sensors nodes
* assign the rest (not overlapping tasks) to their sensor nodes

Experimental Evaluation:

* the sensor network the experiments were conducted in, is a 5x10 sensor testbed with the network forming a rectangle
* collision problems were not present
* one node for collection and one node for disseminating tasks
  + rest of the nodes are used only as a link

**For data sharing across a single node**:

* the combine algorithm is mainly compared to the GA algorithm of

[**Application-aware data collection in Wireless Sensor Networks**]

* because combine computes the best data sharing value for every step, GA gets outperformed
  + data loss rate and energy consumption is higher with GA in comparison to COMBINE by 30% and 35% respectively
* the authors deployed additionally another, real sensor network with a general topology
* cats was tested against random and prude, both algorithms being developed by the authors
* cats was always on top

### MuSA: Multivariate Sampling Algorithm for Wireless Sensor Networks

Summary:

* Problem: optimizing sample sizes for multivariate data in WSNs
* algorithm: sample set is generated by used component analysis
  + set is ranked for representativeness and part of the data is transmitted
* experimental results: positive (authors manage to improve network performance), however algorithm is not tested against other algorithms (maybe because no sampling algorithms exit for multivariate data?
* use cases: sensor nodes which have multiple sensors and thus sense multiple attributes of a phenomenon (i.e., temperature, humidity…)
* advatages:
  + flexible, as distribution technique agnostic
  + good performance
* limitations:
  + localization of nodes is not considered
  + data has to be stationary (not change distribution between epochs)
* comparability:

univariate data: sensornode monitores temperature

multivariate data: sensor node monitores temperature, humidity, pressure …

Basics of Algorithm:

* uses component analysis to rank the multivariate data sensed considering only the first component scores (copied)
  + different component analysis techniques can be used (like PCA, ICA, and robust PCA)
* Based on this ranking, the sampling is performed, alleviating redundancies and maintaining the data representativeness (copied)
* data has to be stationary within the observation window (not sure what is meant with that)
  + stationary probably means that data has the same distribution among intersecting epochs
    - distribution primarily sets the structure of the mean, variance and covariance
  + if not, efficiency of algorithm is reduced
* V' is a set of data produced by observing a phenomenon
  + p is number of variables from set (like temperature, humidity etc), o is the number of sensors and n is number of data points (or samples over V')
  + goal: reduce n to n' -> n' < n
* proposed function: components are transformed, ranked, and sampled (into new set V'')
  + for transformation: PCA, ICA, and robust PCA
* Sink node has to reconstruct data set, through network localization information (?)
* to check how good the reconstruction was => ANOVA and Rerror
  + Rerror = maximum relative absolute error between averages of original data si and the reduced one (copied)
* so what basically happens is the original data is transformed into components, which have the same distribution of the original data
* indexes of first component are sorted
* indexes are used to sample (extract) original data set into a dataset with reduced samples n'
* according to authors, the complexity of the algorithm is O(pn'#hops)
  + where p is the number of variables, n' the reduced sample, and #hops the largest route in the network

Notes:

* localization of nodes is not considered
  + authors point out, that all nodes execute the same localization algorithm so energy expenditure and processing increases at all nodes proportionally

Evaluation:

* distributions used: Gaussian, the Skew Gaussian and the Student t
* data is pseudo real as the individual measurements that lead to the data are not availiable
* sampling strategies are application specific, authors used n, n/2 and log2(n)
  + with log2(n) being an unrealistic parameter only used for reference purposes
* musa was evaluated with:
  + different component analysis techniques
  + different network settings, i.e. number of source nodes, general number of nodes and data generated at source nodes
    - a flat network with shortest path tree routing was used
* aspects evaluated were:
  + network behaviour, specifically
    - energy consuption
    - packet delay
  + data representation, specifically:
    - relative absolute error
    - variance
* relative error was ~5%; decreasing with increasing amount of data
* the authors conclude with positive results of their algorithm
* MuSA was not compared to other algorithms in it's performance