This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.

Cost-Aware Reactive Monitoring in Resource-Constrained Wireless Sensor Networks

Mohammad S. Talebi*, Ahmad Khonsari*† and Reyhaneh Jabarvand‡
* School of Computer Science, IPM, Tehran, Iran
† ECE Department, University of Tehran, Tehran, Iran
‡ CE Department, Sharif University of Technology, Tehran, Iran
Emails: {mstalebi, ak}@ipm.ir, jabarvand@ce.sharif.edu

***Abstract*—Motivated by applications of sensor networks, there has been growing interest in monitoring large scale distributed systems. In these applications, we usually wish to monitor a global system condition defined as a function of local network elements parameters. In this paper, we study *Reactive Monitoring* in sensor networks, which has the benefit of operating in a decentralized manner. Our primary concern in adopting such a monitoring paradigm is reducing the communication cost which is the dominant factor of energy drain in wireless sensor networks. In this study, we address the reactive aggregate monitoring problem by casting the underlying threshold assignment as an optimization problem. This allow us to propose a distributed algorithm to set local thresholds on each sensor node to be adapted to the statistics of the events measured by spatially scattered sensor nodes. Through simulation, we illustrate that the proposed threshold assignment technique can significantly reduce the communication overhead of the monitoring mechanism in sensor networks.**

***Index Terms*—Wireless Sensor Networks, Reactive Monitoring, Threshold Assignment, Optimization**

I. Introduction

In recent years there has been a surge of interest in usage of wireless sensor networks. Wireless sensor networks are expected to be exploited in a wide variety of applications, ranging from data collection to monitoring in different scenarios. Emergence of microsensors based on MEMS technology has made it possible to deploy a large networked system comprising of battery-operated sensor nodes. Using sensor networks, today it is quite a conventional task to implement a ubiquitous architecture, for gathering, processing and transferring of data from an environment ranging from urban sites to wide rural areas [1].

Amongst the wide variety of applications, monitoring is a subject of primary concern in current and emerging deployments of wireless sensor networks. The significance of monitoring in wireless sensor network is increasing not only for accounting and management, but also for revealing anomalies and malicious attacks. In a monitoring scenario, either we are interested in supervising the network itself, e.g. traffic, routing optimization, anomaly detection in data networks, or the environment that network deployed in it, e.g. wildlife behavior, or behavior of moving objects in sensor networks. In this context, we are observing the emergence of monitoring applications for urban sensing, e.g. pollution monitoring in urban areas, noise tracking in residential areas, to name a few [2]-[3].

One can distinguish between two fundamental classes of monitoring mechanisms. In the *Statistical Monitoring* mechanism, the specified management criteria and future trend of the network are evolved by requiring each agent (node) to send the raw stream of its measured data to the central node for further processing. On the other hand, *Reactive Monitoring* mechanism relies on local threshold defined all over the network, which are responsible for detecting criteria violation locally and transmitting appropriate queries to the central node. Reactive monitoring is essentially designed to alleviate the communication burden of monitoring tasks in distributed resource constrained scenarios, and therefore is very appropriate for battery-operated wireless sensor networks [4].

Even with reactive monitoring, the communication cost of the monitoring application can be high enough to waste the scarce power resources of wireless nodes. Thus, our primary concern in adopting such a monitoring paradigm is reducing the communication cost which is the dominant factor of energy drain in wireless sensor networks. The major contribution of this work is to exploit the reactive monitoring mechanism for usage in wireless sensor networks. Our primary concern is to reduce the communication burden emanated from transmitting monitoring queries from sensor nodes. The pure reactive monitoring mechanisms introduced so far, are relying on a general architecture and has not exploited the spatial distribution of sensor nodes over the target environment. Sensor nodes are scattered or deployed over a geographically heterogeneous environment and thereby experience statistically diverse events which primarily depend upon the spatial distribution of their locations. Therefore, it proves quite promising to adapt the structure of the underlying monitoring mechanism to the specific statistics of the event monitored by each sensor node.

We particularly concentrate on a wide class of monitoring applications called *threshold counts* [5], in which the essential criteria of the monitoring application is to check whether an aggregate of the form $\sum_i x_i$ has exceeded a global parameter of the network, say a simple threshold K. In particular, we aim at determining the local thresholds based on the statistics of local events. Towards this end, we formulate the threshold assignment problem as an optimization problem

978-1-4244-2948-6/09/$25.00 ©2009 IEEE

whose objective is to minimize the communication overhead of the network. This allows us to design a self-tuning distributed threshold assignment algorithm which determines the optimal thresholds based on the statistics of local events known by nodes a priori.

The remainder of this paper is organized as follows: Section II reviews related works. Section III defines the system model, states some preliminaries regarding the reactive monitoring mechanism and formulates the optimization problem. Section IV provides the optimal solution to the threshold assignment problem. Section V presents a distributed threshold assignment algorithm based on the solution to the optimization problem and addresses the convergence behavior of the algorithm. Section VI validates experimental evaluation of the proposed algorithm. Finally, Section VII concludes the study.

II. Related Works

In recent years, continuous query processing for monitoring distributed data streams has received much research attention. Different query types have been mentioned in this environment includes quantiles[6], joins[7], etc. Monitoring aggregate threshold queries in networks initially mentioned by the pioneering work of Raz *et al* [4]. They introduced installing local mathematical constraints at remote sites and present a simple approach for threshold assignment assuming uniform data distributions of system variables. Olston *et al* [8] suggested an adaptive filter based approach for tracking error bounded values of aggregate functions. According to the precision specified in continuous query filters are assigned for local values of data objects. Keralpera *et al* [5] presented several algorithms for static and adaptive threshold setting for monitoring thresholded counts queries and analyzed the communication complexity of each algorithm. Sharfman *et al* [9] introduced a geometric approach for monitoring arbitrary threshold functions. [10] investigates a hybrid push-pull approach for monitoring global system parameters in IP networks. Authors present algorithms for selecting which elements "to push" and which "to pull" when network elements events are independent or dependent. Recent work of kashyap *et al* [11] considers the problem of non-zero slack threshold assignment which adaptively dedicate fraction of total threshold to monitoring node to absorb small local threshold crossing that eliminate the need for global system polling. All the above works assume centralized threshold computation and assignment but in this paper we propose a distributed algorithm for optimal threshold setting.

III. System Model and Problem Formulation

A. Network Model

We model the topology of the sensor network by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The vertex set $\mathcal{V} = \{1, 2, \ldots, n\}$ denotes the set of sensor nodes. For the edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, we have $(i, j) \in \mathcal{E}$ if and only if there is a connection between node i and j. We assume that there is a base station being responsible for monitoring the network. Each sensor node i, at time t continuously measures (reads) the requested local event denoted by $x_i^{(t)}$ at a fixed sampling rate, which results in a vector $\mathbf{x}^{(t)} = (x_i^{(t)}, i \in \mathcal{V})$. The goal is to detect when the aggregate value of all the measurements of the form $\sum_{i=1}^{n} a_i x_i$ crosses a predetermined threshold K.

As stated earlier, in this paper we focus on reactive monitoring schemes, in which local thresholds are set to check the aggregate condition, distributively. In particular, we exploit the Simple-Rate (SR) Algorithm introduced in [4] with some modifications, which is to be described in the next subsection.

B. The Monitoring Algorithm

In this subsection, we state the SR Algorithm to be used for aggregate computation.

The SR Algorithm consists of two parts: SR_N for distributed nodes and SR_C for central base station. Both of these parts are described as SR Algorithm [4]. The SR_N is quite simple, however the central part, i.e. SR_C is partly more complex.

The original SR Algorithm assumes uniform thresholding scheme, i.e. each node is assigned a fixed threshold δ which can be selected to be $\delta = K/n$. Therefore, the SR_C algorithm in its original form updates the next time in which global polling is needed (i.e. t_m) as following

$$t_m \leftarrow t + \frac{K - \sum_{i=1}^{n} a_i x_i^{(t)}}{n\delta}$$

In contrast to the original form of SR, we focus on a scenario with nonuniform thresholding, i.e. each sensor node i is initialized with a threshold T_i. To be consistent with the update equation of t_m, we replace δ by $\bar{T}$, i.e. the average value of vector $\mathbf{T} = (T_i, i \in \mathcal{V})$. Then, the major concern of the algorithm is to assign threshold values so as to minimize the communication cost of the SR algorithm.

First we compute the cost of the SR Algorithm. The cost of the SR consists of two components [4]. The first one is due to local events at sensor nodes.

Let $\mathcal{A}$ be the outcome that at least one node had triggered a query at time t, then

SR. Simple Rate Algorithm

SR_N
At each node i:
Initialize threshold T_i.

At time t
if $x_i^{(t)} - x_i^{(t-1)} > T_i$
  send the value $x_i^{(t)}$ to the base station.

SR_C
Initialize $t_m = 0; f = 0$
while (TRUE)
  if $t \geq t_m$ OR report received
    $f \leftarrow$ POLLALL($x_i^{(t)}$)
    $f > K$
      report ALARM
    else
      $t_m \leftarrow t + \frac{K - \sum_{i=1}^{n} a_i x_i^{(t)}}{n\bar{T}}$

SR. Simple Rate Algorithm

$$\begin{aligned}\Pr(\mathcal{A}) &= 1-\Pr\left(\bar{\mathcal{A}}\right) && (1)\\ &= 1-\prod_{i=1}^{n}\Pr\left(\bar{\mathcal{A}}_i\right) && (2)\\ &= 1-\prod_{i=1}^{n}\Pr\left(x_i^{(t)}-x_i^{(t-1)}<T_i\right) && (3)\\ &= 1-\prod_{i=1}^{n}F_i(T_i) && (4)\end{aligned}$$

where $\bar{\mathcal{A}}_i$ is the outcome that node i had not triggered a query at time t. Moreover, (2) is derived assuming that different nodes are independent and $F_i(.)$ is the Cumulative Distribution Function (CDF) of the occurrence of $x_i^{(t)}-x_i^{(t-1)}$ event at node i, to be known by node i a priori. Thus, the expected number of messages generated due to local events is given by [4]:

$$\mathbb{E}\left(\text{local messages}\right)=n\left(1-\prod_{i=1}^{n}F_i(T_i)\right) \quad (5)$$

The second component of the cost is due to the centralized polling part. The expected number of steps per poll is [4]:

$$\mathbb{E}\left(\frac{K-\sum_{i=1}^{n}a_ix_i}{n\bar{T}}\right) \quad (6)$$

Similarly, the communication cost due to second part is

$$\frac{n^2\bar{T}}{K-\mathbb{E}\left(\sum_{i=1}^{n}a_ix_i\right)} \quad (7)$$

Combining (5) and (7), the communication cost of the SR Algorithm is given by

$$J(\mathbf{T})=n\left(1-\prod_{i=1}^{n}F_i(T_i)\right)+\frac{n^2\bar{T}}{K-\mathbb{E}\left(\sum_{i=1}^{n}a_ix_i\right)} \quad (8)$$

C. Problem Formulation

In this subsection, we are ready to cast the threshold assignment as an optimization problem, whose objective is to minimize the communication cost of the algorithm. Considering constraint $\sum_i a_ix_i^{(t)}\le K$, for the average of threshold we have

$$\bar{T}=\frac{1}{n}\sum_{i=1}^{n}T_i\le\frac{K}{n} \quad (9)$$

Hence, $\bar{T}$ is at most K/n. Therefore, the second component of $J(\mathbf{T})$ is at most $\frac{nT}{K-\mathbb{E}(\sum_i^n a_ix_i)}$, which is independent from $\mathbf{T}$. Therefore, only the communication cost due to local messages depends upon $\mathbf{T}$. Then the communication cost function to be minimized is $n\left(1-\prod_{i=1}^{n}F_i(T_i)\right)$. To minimize the communication cost, it is equivalent to formulate the problem as

$$\max_{\mathbf{T}}\prod_{i=1}^{n}F_i(T_i) \quad (10)$$

subject to:

$$\sum_{i=1}^{n}a_iT_i\le K \quad (11)$$

We turn the problem (10) into a simpler equivalent form by taking the logarithm of its objective function as following:

$$\max_{\mathbf{T}}\sum_{i=1}^{n}\log F_i(T_i) \quad (12)$$

subject to:

$$\sum_{i=1}^{n}a_iT_i\le K \quad (13)$$

Problems (10) and (12) are equivalent due to the facts that the $\log(.)$ function is monotone increasing and such transformation of the objective function for an optimization problem leads to an equivalent form. For problem (12) with an affine constraint to admit a unique maximizer, it is necessary to be convex, i.e. its objective to be concave as following

$$-\frac{d^2}{dz^2}\log F_i(z)\ge\kappa_i\ge 0 \quad (14)$$

where κ_i is a positive constant to bound the curvature of $\log F_i(z)$ to guarantee the convergence, which is to be discussed later in Section V.

IV. Optimal Solution

A. Dual Formulation

We aim at solving problem (12) through its dual. We start by writing the Lagrangian of problem (12), as follows

$$L(\mathbf{T},\mu)=\sum_{i=1}^{n}\log F_i(T_i)-\mu\left(\sum_{i=1}^{n}a_iT_i-T\right) \quad (15)$$

where $\mu>0$ is the positive Lagrange multiplier associated with constraint (13). Using Karush-Kuhn-Tucker (KKT) conditions for convex optimization, to find $\mathbf{T}^*$, we should find the stationary points of the Lagrangian and satisfy complementary slackness conditions. The complementary slackness conditions for optimal primal variable $\mathbf{T}^*$ and dual variable μ^*, are

$$\mu^*\ge 0;\quad \sum_{i=1}^{n}a_iT_i^*\le T; \quad (16)$$

$$\mu^*\left(\sum_{i=1}^{n}a_iT_i^*-T\right)=0 \quad (17)$$

In order to find the stationary points of the Lagrangian, we solve

$$\nabla L(\mathbf{T}^*,\mu^*)=\mathbf{0} \quad (18)$$

where $\mathbf{0}$ is a vector with all zero. For the ith element of (18) we have

$$\begin{aligned}\frac{\partial L}{\partial T_i} &= \frac{F_i'(T_i)}{F_i(T_i)}-\mu a_i && (19)\\ &= \frac{f_i(T_i)}{F_i(T_i)}-\mu a_i && (20)\end{aligned}$$

where $f_i(z)$ is the corresponding probability density function (pdf) of $F_i(z)$.

Setting (20) to zero and doing some algebraic manipulations yields

$$T_i^* = l_i^{-1}(a_i\mu^*) \tag{21}$$

where $l_i(z) \doteq \frac{f_i(z)}{F_i(z)}$.

Now, we are ready to solve problem (12) through its dual problem which is defined as [12]:

$$\min_{\mu\geq 0} D(\mu) \equiv \min_{\mu} \max_{\mathbf{T}} L(\mathbf{T},\mu) \tag{22}$$

where $D(\mu)$ is called the dual function. Based on the results of the KKT condition mentioned above, $D(\mu) = L(\mathbf{T}^*,\mu)$ with $\mathbf{T}^*$ is given by (21).

Dual problem defined above can be solved using iterative methods. In order to obtain a distributed algorithm, we solve the dual problem (22) using Gradient Projection Method. We postpone solving (22) to the next subsection.

B. The Iterative Solution

In the sequel, we solve the dual problem using Gradient Projection Method [13]. In this respect, μ should be adjusted in opposite direction to the gradient of dual function, i.e. $\nabla D(\mu)$. Precisely speaking, in the kth iteration step, $\mu^{(k)}$ is updated as follows

$$\mu^{(k+1)} = \left[\mu^{(k)} - \gamma \frac{dD(\mu^{(k)})}{d\mu}\right]^+ \tag{23}$$

where $[z]^+ = \max\{z, 0\}$ and γ is a sufficiently small constant step size. Using the Danskin's Theorem [13], the derivative of $D(\mu)$ is given by

$$\frac{dD(\mu)}{d\mu} = K - \sum_{i=1}^{n} a_i T_i \tag{24}$$

Substituting (24) in (23), yields

$$\mu^{(k+1)} = \left[\mu^{(k)} + \gamma\left(\sum_{i=1}^{n} a_i T_i^{(k)} - K\right)\right]^+ \tag{25}$$

where $T_i^{(k)}$ is the solution to (21) for a given $\mu^{(k)}$. In (25), γ is chosen sufficiently small so as to guarantee the convergence. (21) and (25) forms an iterative solution to problem (22) and thereby (12). At each iteration step k, dual variable μ is updated based on the history of itself and the primal variables $\mathbf{T}$. Then, it would be utilized by (21) to update primal variable $\mathbf{T}$, accordingly. Therefore, after spending enough iteration steps, primal and dual variables tends to primal-optimal $\mathbf{T}^*$ and dual-optimal μ^*, respectively. We deffer the algorithm until the Section V.

V. Distributed Threshold Assignment Algorithm

A. The Algorithm

Based on the obtained iterative solution, we propose a distributed algorithm as a distributed solution to problem (12) and thereby problem (10).

The algorithm is devised by utilizing the update equation (25) and (21), over the network.

In particular, each sensor i in the iteration step k benefits from all other nodes' currently assigned threshold $\mathbf{T}_{-i}$, thanks to Flooding-like algorithms, to update the current Lagrange multiplier $\mu^{(k)}$. Since all other nodes have access to such information too, they will obtain the same value for $\mu^{(k+1)}$ and therefore, we don't introduce additional notation so as to distinguish between the realized update process.

Upon updating $\mu^{(k)}$, each sensor i calculates its threshold T_i, accordingly. The above rule will proceed until reaching some predefined notions of convergence.

The proposed algorithm is listed below as Algorithm 1.

Algorithm 1. Distributed Threshold Assignment Algorithm

Initialization

Initialize γ, a_is and K $\forall i = 1..n$.

Main Loop

Do until $\max_i |T_i^{(k+1)} - T_i^{(k)}| < \epsilon$

1.
At each sensor node, update μ as following:
$\mu^{(k+1)} =$
$\left[\mu^{(k)} + \gamma\left(\sum_{i=1}^{n} a_i T_i^{(k)} - K\right)\right]^+$
2. Update $T_i^{(k)}$ according to the following equation:
$T_i^{(k+1)} = l_i^{-1}(a_i\mu^{(k)})$

Algorithm 1. Distributed Threshold Assignment Algorithm

B. Convergence Analysis

In this subsection, we investigate the convergence behavior of the proposed algorithm. As step size has an important role on the convergence behavior of the update equation (25), we mainly focus on its effect. In other words, the conditions under which Algorithm 1 converges and performance analysis of the algorithm will be obtained with respect to the choice of step size.

The following theorem determines the necessary condition on the step size, under which Algorithm 1 converges to the arbitrarily close neighborhood of the optimum of problem (22) and thereby that of problem (10).

Theorem 1: The iterative threshold assignment algorithm proposed by (21) and (25) converges to the arbitrarily close neighborhood of the optimal point of problem (10) provided that

$$0 < \gamma \leq \frac{2\underline{\kappa}}{n\bar{a}^2} \tag{26}$$

where

$$\bar{a} \doteq \max_i a_i$$
$$\underline{\kappa} \doteq \min_i \kappa_i$$

Proof: See Appendix I for proof. ■

VI. Experimental Evaluation

We have conducted simulation experiments to evaluate the performance of the proposed algorithm. Simulation experiments are carried out using Java and MATLAB. We verify that the algorithm, locally executed on each node, may indeed achieve the desired global optimal threshold assignment.

In our simulation scenario, we consider a sensor network consisting of 100 sensor nodes. We assume that each node i takes measurements of a physical phenomenon, whose difference is Poisson distributed with the parameter λ_i. The corresponding coefficient a_i for such an illustrative experiment is set to 1. Step size is chosen to be $\gamma = .0012$.

One of the most significant issues of interest are the evolutions of primal and dual variables. Evolution of the assigned thresholds T_is for some of nodes and dual variable μ are depicted in Fig. 1(a) and 1(b), respectively. For this experiment, λ_i is chosen to be a random value, uniformly distributed in $[1, 4]$. From these figures, it is apparent that by spending less than 40 iteration steps, convergence was achieved and thereafter, μ and T_is had intangible variations.

In order to have a better understanding of how optimal thresholds are heterogeneously scattered all over the network, Fig. 2 depicts the bar plot of optimally-assigned thresholds for all nodes. As Fig. 2 implies, optimal thresholds for the dominant majority of nodes are quite different from the uniform thresholding schemes. It is also apparent that for many nodes the optimal threshold value is quite larger than the anticipated uniform threshold. Such a greater threshold would generate constraint violation less frequently and thereby helps extending the lifetime of the corresponding sensor node.

It is anticipated that threshold assignment in favor of problem (10) will reduces the communication burden of monitoring queries. To validate this claim, in another experiment we have varied parameter λ_i uniformly over the interval $[22, 34]$ and compared the overhead due to monitoring queries for both optimal and uniform thresholds. The results are shown in Fig. 3, which divulges that the number of monitoring queries represented in percentage is reduced by a factor of 25.7%. Such a tangible reduction in overhead verifies the claim that assigning local thresholds using the proposed algorithm significantly alleviates the overhead incurred by the monitoring mechanism.

VII. Conclusion

There have been many studies exploring various applications of WSNs such as monitoring. In this paper, we addressed the problem of distributed threshold selection for reactive aggregate threshold monitoring in WSNs. The nature of such networks has induced the design of asynchronous and distributed algorithms for exchanging the information among the sensor nodes. We established a connection between threshold assignment and statistical characteristics of the event being monitored over the network. We utilized this connection to design the minimal-cost network monitoring scenario by casting the threshold assignment as an optimization problem. This allowed us to formulate the minimum communication

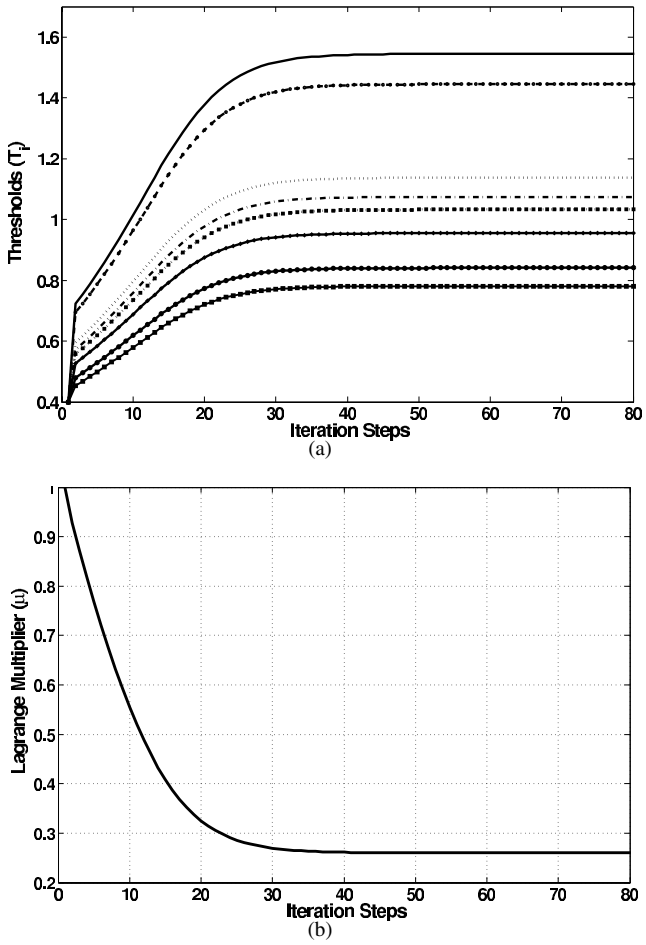

Fig. 1. Evolution of (a) Assigned Thresholds (T_i) for some nodes and (b) Lagrange Multiplier (μ)

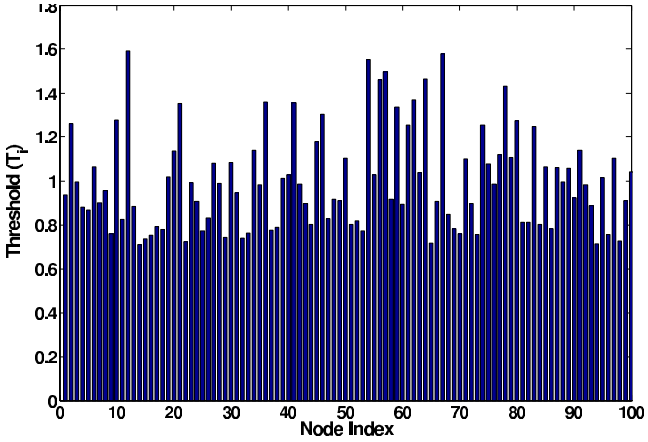

Fig. 2. Optimal Threshold for All Nodes

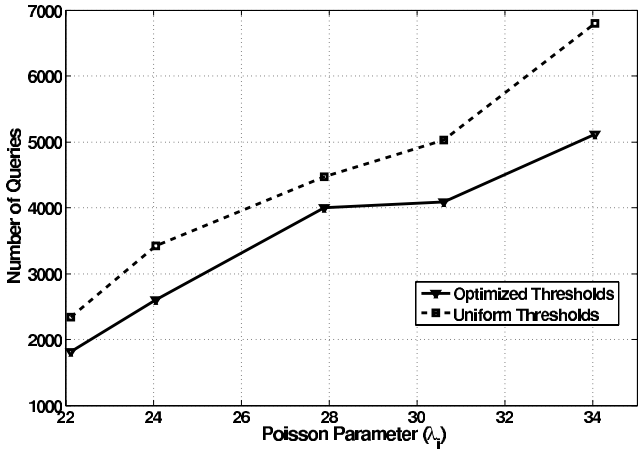

Fig. 3. Comparison Between Optimal Threshold Assignment and Uniform Threshold Assignment Schemes

cost threshold assignment as an optimization problem to be convex by holding slight conditions on the objective function.

Solving the threshold assignment problem in a distributed manner is not possible. However, we have utilized the duality theory to solve the problem which led to a distributed solution using gradient method. This allows for self-tuning thresholds: that is, the network can initialize with arbitrary initial conditions and then locally, without any central coordination converges to within arbitrarily close neighborhood of the optimal threshold values. The approach developed in this study is not restricted to the investigated scenarios and can be generalized to cover more complex ones. The results extracted from the experimental evaluation was promising and demonstrated that the achieved performance of the suggested algorithm are quite comparable to the results of the centralized approach.

REFERENCES

[1] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: Mobile networking for smart dust," *In Proceedings of MobiCom'99*, pp. 271-278, 1999.
[2] "google StreetView,"http://maps.google.com/help/maps/streetview/.
[3] "Urban Atmospheres," http://www.urban-atmospheres.net/ .
[4] M. Dilman and D. Raz, "Efficient reactive monitoring," *In Proceedings of INFOCOM'01*, pp. 1012-1019, 2001.
[5] R. Keralapura, G. Cormode, and J. Ramamirtham, "Communication-efficient distributed monitoring of thresholded counts," *In Proceedings of SIGMOD'06*, 2006.
[6] G. Cormode, M. Garofalakis, S. Muthukrishnan, and R. Rastogi, "Holistic aggregates in a networked world: Distributed tracking of approximate quantiles," *In Proceedings of SIGMOD'05*, pp. 25-36, 2005.
[7] G. Cormode and M. Garofalakis, "Sketching streams through the net: Distributed approximate query tracking." In *In Proceedings of VLDB'05*, pp. 13-24, 2005.
[8] C. Olston, J. Jiang, and J. Widom, "Adaptive filters for continuous queries over distributed data streams," *In Proceedings of SIGMOD'03*, pp. 563-574, 2003.
[9] I. Sharfman, A. Schuster, and D. Keren, "A geometric approach to monitoring threshold functions over distributed data streams," *In Proceedings of SIGMOD'06*, pp. 301-312, 2006.
[10] A. Bulut, N. Koudas, A. Meka, A. K. Singh, and D. Srivastava, "Optimization Techniques for Reactive Network Monitoring," *IEEE Transactions on Knowledge and Data Engineering*, To appear.
[11] S. Kashyap, J. Ramamirtham, R. Rastogi, and P. Shukla, "Efficient Constraint Monitoring Using Adaptive Thresholds," *In Proceedings of ICDE'06*, pp. 526-535, 2006.
[12] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge, U.K, Cambridge Univ. Press, 2004.
[13] D. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1999.

APPENDIX I: PROOF OF THEOREM 1

According to the duality theory, whenever the strong duality is held, the duality gap is zero and the optima of the dual leads to that of primal. Hence, it suffices to seek the condition on the step size under which (25) converges to a neighborhood of the dual-optimal point.

According to the Descent Lemma [13], to establish the convergence condition of the update equation, the step size must satisfy the following

$$0 < \epsilon \leq \gamma \leq \frac{2-\epsilon}{M} \tag{27}$$

where M is the constant which establishes Lipschitz continuity for the dual function $D(\mu)$. In order to satisfy Lipschitz continuity condition for $D(\mu)$, it suffices to show that the absolute value of second derivative of $D(\mu)$ is upper bounded. Considering (24), for the second derivative of $D(\mu)$, we have

$$\frac{d^2D}{d\mu^2} = -\sum_{i=1}^{n} a_i \frac{dT_i}{d\mu} \tag{28}$$

Considering (21), for $\frac{dT_i}{d\mu}$ we get

$$\frac{dT_i}{d\mu} = \frac{a_i}{l_i'(T_i)} \tag{29}$$

$$= \frac{a_i F_i^2(T_i)}{F_i'' F_i - F_i'^2}$$

$$= \frac{a_i}{\frac{d^2}{dT_i^2}\log F_i(T_i)} \tag{30}$$

Therefore, for $\frac{d^2D}{d\mu^2}$, we get

$$\frac{d^2D}{d\mu^2} = \sum_{i=1}^{n} -\frac{a_i^2}{\frac{d^2}{dT_i^2}\log F_i(T_i)}$$

The upper bound of $\left|\frac{d^2D}{d\mu^2}\right|$, can be found as following

$$\left|\frac{d^2D}{d\mu^2}\right| = \sum_{i=1}^{n} \frac{a_i^2}{\frac{d^2}{dT_i^2}\log F_i(T_i)}$$

$$\leq \sum_{i=1}^{n} \frac{a_i^2}{\kappa_i} \tag{31}$$

$$\leq \frac{n \max_i a_i^2}{\min_i \kappa_i}$$

$$\leq \frac{n\bar{a}^2}{\underline{\kappa}} \tag{32}$$

where

$$\bar{a} \doteq \max_i a_i; \quad \bar{\kappa} \doteq \min_i \kappa_i$$

Finally, by Descent Lemma we deduce that for some $\epsilon > 0$

$$0 < \gamma \leq \frac{2\underline{\kappa}}{n\bar{a}^2} \tag{33}$$

which completes the proof.