Towards Optimal Event Detection and Localization in Acyclic Flow Networks

M. Agumbe Suresh, R. Stoleru, R. Denton, E. Zechman[†], B. Shihada[‡] *

Department of Computer Science and Engineering, Texas A&M University
[†]Department of Civil Engineering, Texas A&M University
[‡]Department of Computer Science, King Abdullah University of Science and Technology
{agumbe,stoleru,denton,ezechman}@tamu.edu,basem.shihada@kaust.edu.sa

Abstract. Acyclic flow networks, present in many infrastructures of national importance (e.g., oil & gas and water distribution systems), have been attracting immense research interest. Existing solutions for detecting and locating attacks against these infrastructures, have been proven costly and imprecise, especially when dealing with large scale distribution systems. In this paper, to the best of our knowledge for the first time, we investigate how mobile sensor networks can be used for optimal event detection and localization in acyclic flow networks. Sensor nodes move along the edges of the network and detect events (i.e., attacks) and proximity to beacon nodes with known placement in the network. We formulate the problem of minimizing the cost of monitoring infrastructure (i.e., minimizing the number of sensor and beacon nodes deployed), while ensuring a degree of sensing coverage in a zone of interest and a required accuracy in locating events. We propose algorithms for solving these problems and demonstrate their effectiveness with results obtained from a high fidelity simulator.

1 Introduction

Acyclic flow networks are pervasive in infrastructures of national importance, including oil & gas and water distribution systems. Water distribution, one of seven critical infrastructure systems identified in the Public Health, Security, and Bioterrorism Preparedness and Response Act [1], is particularly vulnerable to a variety of attacks, including contamination with deadly agents through intentional or accidental hazards. Contamination of water supply can have acute consequences for public health and impact economic vitality [2]. To protect consumers in a network, water security measures should include an on-line, real-time contaminant warning system of sensors to quickly identify any degradation in water quality. Efficient placement of sensors is needed to collect information for responding to a threat by identifying the location and timing of the contaminant intrusion [3] and developing strategies for opening hydrants to flush a contaminant [4]. An extensive set of studies have been conducted to develop and apply optimization-based methodologies for placing sensors in a water distribution network [5] [6].

Due to the costs of placing and maintaining sensor networks, the sensor placement problem has been traditionally solved for a limited number of sensors that often cannot provide adequate coverage of a realistic network. In addition, existing sensor technology limits the locations for placing sensors, owing to a small number of points in an underground network that are both accessible and located near a power source. Recent research has investigated placement of wireless networks to enable a

* This publication is based in part on work supported by Award No. KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST).

new approach for monitoring water distribution systems through low-cost autonomous, intelligent sensor nodes. These systems have been preliminarily tested through deployment in municipalities and laboratory settings for detecting leaks and breaks in pipe networks [7] [8]. The research presented here provides the basis for exploring a new paradigm for monitoring acyclic flow networks in general, and water distribution systems in particular, through a set of mobile sensors. Mobile sensors can provide improved coverage of pipes and nodes at a lower cost to municipalities. Our research is enabled by recent advances in wireless sensor network technologies and successful deployments of, mostly static, sensor network systems for military applications [9], emergency response [10] and habitat monitoring [11].

In our proposed solution to the research challenges posed by accurate and inexpensive event detection and localization in acyclic flow networks, a mobile sensor network, consisting of mobile *sensor nodes* and *beacons nodes* is deployed in an area (i.e., distribution system) to be monitored. The sensor nodes are equipped with sensing modalities specific to the type of threat/event they need to detect. Once they are "inserted" into the flow network, they are moved by the flow along a set of fixed paths. For energy efficiency, reduced cost, and because GPS is not available, sensor nodes can only obtain their location by proximity to a set of fixed beacon nodes with known location. These sensors travel along the edges of the flow network, sensing and recording data, i.e. events and proximity to beacon nodes. Nodes may not possess communication capabilities (e.g., acoustic modems), for costs and form factor reasons (if nodes can not communicate, their physical capture will be needed). Using the data collected by sensor nodes we aim to identify the existence and location of an event. This research addresses the problem of reducing the number of sensors and beacon nodes deployed in an acyclic flow network, while ensuring that a desired accuracy of event detection is achieved. More specifically, the contributions of this paper are as follows:

- We propose the idea of acyclic flow sensor networks, and formally define the problems for optimal event detection and localization in such networks.
- We prove that the event detection problem is NP-Hard and we propose an approximation algorithm to derive the minimal number of sensor nodes to be deployed and their deployment locations.
- We propose algorithms for optimally solving the Event Localization problem in acyclic flow networks, through an intelligent deployment of beacon nodes.
- We evaluate the performance of our solutions through extensive simulations using a high fidelity acyclic flow network simulator.

The rest of the paper is structured as follows. In Section 2 we formulate the problem of optimal event localization in acyclic flow networks, and Sections 3 and 4 propose solutions for it. We present performance evaluation results in Section 5 and review state of art in Section 6. We conclude in Section 7 with ideas for future work.

2 Preliminaries and Problem Formulation

An acyclic flow network can be best understood by considering a typical example - a water distribution system, as shown in Figure 1. Water, stored in water reservoirs or water towers, is pumped by pumpstations into a network of underground pipes. Depending on the demand of water by various consumers, the flow in pipes can change in direction and magnitude.

In a water distribution system, we are interested in identifying the point(s) in the system where an attack, e.g., chemical contamination, might be taking place. To achieve accurate and cost effective

discovery of the contamination point, we propose to deploy sensor nodes, equipped with suitable sensors (i.e., chemical sensors in our scenario) in the water distribution system.

Due to unavailability of GPS underground, a deployed sensor node can only infer its position from its proximity to points with known locations, such as beacon nodes. In this paper, we assume the availability of inexpensive sensor nodes, equipped with simple sensing modalities and with no communication capabilities. Simple sensing modality means that the sensor is capable of detecting contamination in large concentrations only, i.e., typically around the point of contamination. The lack of communication capabilities means that sensor nodes do not collaborate. In this paper, for deriving optimal event localization algorithms, we consider time-invariant flow networks, in which the flow does not change in time.

Fig. 1. An acyclic flow network example - a water distribution system, encompassing water storage (e.g., reservoir and water towers), and water distribution (i.e., a network of underground pipes).

In the remaining part of this section we formally define terms pertaining to acyclic flow networks and formulate the optimal event detection and optimal event localization problems.

2.1 Definitions

We consider a directed acyclic graph $G(V, E)$ in which every edge $(u, v) \in E$ has a non-negative, real-valued capacity denoted by $c(u, v)$, and two sets of vertices: $S = \{s_1, s_2, ..., s_k\}$ a set of *sources*, and $D = \{d_1, d_2, ..., d_k\}$ a set of *sinks*, where $S, D \subset V$.

Definition 1. *An* Acyclic Flow Network*, $\mathcal{F}$, is defined as a real function $\mathcal{F} : V \times V \to \mathbb{R}$ with the following properties:*

- *$\mathcal{F}(u, v) \leq c(u, v)$, where $c(u, v)$ is a constant. This means that the flow on an edge, cannot exceed its capacity.*
- *$\sum_{w \in V} \mathcal{F}(u, w) = \sum_{w \in V} \mathcal{F}(w, u) \forall u \in V$, unless $u \in S$ or $w \in D$, which means that the net flow of a vertex is 0, except for source and sink nodes.*

Definition 2. *A* Beacon Node *(B_i) is a node which periodically broadcasts its location. A beacon node is placed at a vertex $v_j \in V$.*

Definition 3. *A* Sensed Path *(SP_i) is a set $\{e_j \mid e_j \in E\}$ of edges through which a node n_i traveled and sensed events and proximity to beacon nodes.*

Definition 4. *An* Insertion Point *(or* Source*) for a node n_i is a vertex $v_j \in V$ at which the node is introduced into the flow network.*

Definition 5. *A* Path Synopsis *(PS_i) for a node n_i is an ordered list of events and beacons encountered by node n_i along its Sensed Path SP_i.*

Definition 6. *A* Zone of Interest *(I) is a subset of edges in graph $G(V, E)$, i.e., $I \subseteqq E$, which we are interested in monitoring. A given $\mathcal{F}$ can have multiple zones of interest.*

Definition 7. *The* Degree of Coverage *(D_c) is the fraction of I that nodes need to sense/traverse. More precisely, $0 \leq D_c \leq 1$, and at least D_c of edges in I are being traversed by sensor nodes.*

Definition 8. *The* Probability of Detection / Event Localization Accuracy *(P_d) is the probability of finding an event (or the accuracy of event localization) in Zone of Interest I. Formally, $P_d = \frac{(TP+TN)}{(TP+TN+FP+FN)}$, where TP, TN, FP and FN are true positives (i.e., an event existed and the algorithm detected it), true negatives (i.e., an event did not exist and the algorithm correctly indicated a non-existence), false positives (i.e., an event did not exist, but the algorithm detected one) and false negatives (i.e., an event existed and the algorithm failed to detect it), respectively.*

Definition 9. *A* Suspects List *is the list of edges $\{e_i \mid e_i \in E\}$ in which an event of interest was detected by a sensor node (i.e., recorded in its Path Synopsis).*

2.2 Formulations of Problems

Given the above definitions, an Acyclic Flow Network poses two interesting problems. The first one is the "Optimal Event Detection Problem", i.e., detecting the existence of an event in a zone of interest, using the least resources possible. This problem is a binary decision, i.e., an event is present or not. The second problem is the "Optimal Event Localization Problem", i.e., detecting the location of an event, using the least resources possible. From here on we will refer to "Event Detection" as "Sensing Coverage", since detecting an event requires sensing coverage. The two aforementioned problems are formally defined as follows:

Optimal Event Detection (Sensing Coverage) Problem (SCP): Given an acyclic flow network $\mathcal{F}$, a zone of interest I in $\mathcal{F}$, and degree of coverage D_c, find the smallest set $S = \{(s_i, q_i) \mid s_i \in V \wedge q_i \in \mathbb{N}\}$ of insertion points s_i (sources) where sensor nodes need to be deployed, and the smallest number q_i of sensors to be deployed at s_i, such that the union of sensed paths of all sensors represents at least D_c of I.

Optimal Event Localization Problem: Given an acyclic flow network $\mathcal{F}$, a zone of interest I in $\mathcal{F}$, with required probability for detecting an event P_d, compute the minimum number of beacon nodes that need to be deployed in $\mathcal{F}$, and their deployment locations (i.e., vertices in V) such that from $\{PS_i \mid i \leq \sum_{i=1}^{|S|} q_i\}$ (a set of path synopses) for all sensor nodes deployed in the $\mathcal{F}$, the probability of localizing an event X, detected by sensor nodes, by identifying a set of edges where the event could be present (i.e. *Suspects List*) is P_d.

It is important to build the intuition that D_c determines the number of nodes to be inserted in the flow network (i.e., sensing coverage for event detection), whereas P_d determines the number of beacons to be deployed (event localization accuracy). When P_d is high most vertices in the flow network will have beacons. So, even if the sensing coverage of zone of interest is small, if an event is detected it is localized more accurately when P_d is high (note that P_d is high and D_c is low, there is a chance that an event might not be detected). Typically, ensuring sensing coverage is less important than accurate event localization. In systems where it is sufficient

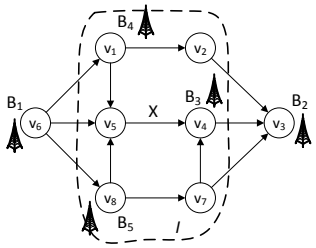


Fig. 2. Graphical representation of an acyclic flow network involving vertices/junctions, edges/pipes, and defined elements for the event detection problem: a zone of interest I, beacon nodes B_i and an event X along edge (v_5, v_4).

to know that an event occurred, P_d can even be 0, which means no beacons are required.

Example 1: For clarity of presentation, we illustrate the aforementioned concepts with an example, depicted in Figure 2. In the shown acyclic flow network, the Zone of Interest I consists of 6 edges, (v_1, v_2), (v_1, v_5), (v_8, v_5), (v_5, v_4), (v_7, v_4) and (v_8, v_7) (i.e., this is the area where events of interest might occur), and the Degree of Coverage D_c is 0.67, i.e., any four of the six edges of I must be sensed/traversed. Five beacon nodes (B_1, B_2, B_3, B_4 and B_5) are deployed to ensure a P_d of 0.67, and a number of sensor nodes are deployed at the *Insertion Point* vertex v_6. A node n_i might travel along edges (v_6, v_1), (v_1, v_5), (v_5, v_4) and (v_4, v_3) (i.e., a Sensed Path SP_i). Consequently, the *Path Synopsis* for node n_i is $PS_i = \{B_1, B_4, X, B_3, B_2\}$. A solution for event localization is a set of edges, i.e., "Suspects List", where the event might be present: $= \{(v_1, v_5), (v_5, v_4)\}$.

3 Node Placement for Optimal Sensing Coverage

We first show that SCP reduces to the Weighted Set Cover problem, a known NP-Hard problem.

Theorem 1: SCP is NP-Hard.

Proof. Take an instance of the Weighted Set-Cover (WSC) problem $(\mathcal{E}, \mathcal{V}, \mathcal{S}, w)$ where:

$$\mathcal{E} = \{e_i \mid i = 1, 2, \ldots, n\}$$
$$\mathcal{V} = \{\mathcal{V}_j \mid j = 1, 2, \ldots, m; \mathcal{V}_j \subseteq \mathcal{E}\}; \bigcup_{j=1}^{m} \mathcal{V}_j = \mathcal{E}$$
$$\mathcal{S} \subset \mathcal{V} \mid \bigcup_{\mathcal{V}_j \in \mathcal{S}} \mathcal{V}_j = \mathcal{E}$$
$$\omega : \mathcal{V} \to \mathbb{R}$$
$$W = \sum_{\mathcal{V}_j \in \mathcal{S}} \omega_j$$

where $\mathcal{E}$ is a set of elements, $\mathcal{V}$ is a set of subsets of $\mathcal{E}$ covering all elements of $\mathcal{E}$, $\mathcal{S}$ is a subset of $\mathcal{V}$ that contains all elements of $\mathcal{E}$. Each subset $\mathcal{V}_j$ has a weight ω_j. $\mathcal{S}$ is constructed such that $\mathcal{E}$ can be covered with cost W.

We construct f : WSC $\to$ SCP

$$V = \{w_j \mid j = 1, 2, \ldots, m\} \cup \{u_i \mid i = 1, 2, \ldots, n\} \cup \{v_i \mid i = 1, 2, \ldots, n\}$$
$$E = E_1 \cup E_2 \text{ where } E_1 = \{(w_j, u_i) \mid e_i \in \mathcal{V}_i\};\ E_2 = \{(u_i, v_i) \mid i = 1, 2, \ldots, n\}$$
$$I = E_2\ ;\ D_c = 1$$

$$\mathcal{F}(u, v) = \begin{cases} c(u, v) & \text{if } (u, v) \in E_2 \\ \frac{|\mathcal{V}_i|}{\omega_i} & \text{if } (u, v) \in E_1 \\ 0 & \text{if } (w, u) \notin E_1 \text{ but } (u, v) \in E_2 \end{cases}$$

where V, E, I, and D_c represent the vertices, edges, zone of interest and degree of coverage of FSN, respectively, and $\mathcal{F}(u, v)$ is the flow in edge (u, v). I can be covered with W sensor nodes.

Note that V is constructed in $O(m)$ time, E and I in $O(n+m)$ time, $\mathcal{F}$ in $O(n+m)$ time and D_c in constant time. Hence, this construction occurs in polynomial time.

Equivalence: $\mathcal{S}$ covers $\mathcal{E}$ with cost $W \iff D_c$ of I in $\mathcal{F}$ can be covered by W sensor nodes.

$\Rightarrow$ Given $\mathcal{S}$ covers $\mathcal{E}$ with cost W. Since $D_c = 1$, all edges in I need to be covered. The number of nodes to be inserted in w_j, $j = 1, 2 \ldots m$ such that all edges incident on it are covered is $\omega_j \mid j = 1, 2 \ldots m$. Any node that reaches u_i will cover the edge (u_i, v_i). Since $\mathcal{S}$ covers $\mathcal{E}$ with cost W, selecting the corresponding vertices in $\mathcal{F}$ covers all edges in I. So, if $\mathcal{S}$ covers $\mathcal{E}$ with cost W, inserting W sensor nodes in the corresponding vertices in $\mathcal{F}$, D_c of I will be covered.

$\Leftarrow$ Given D_c of I in $\mathcal{F}$ can be covered by W sensor nodes. By our definition of E, all u_i's are covered by at least one w_j. Hence, any set of u_i in the set of insertion points can be replaced by an existing w_j without increasing the cost. Notice that using our definition of E, each vertex w_j can be used to uniquely identify $V_j \in \mathcal{V}$. Further, each w_j covers a set of edges in I and each corresponding V_j covers a set of elements in $\mathcal{E}$. So, the sets corresponding to the insertion points ensure that $\mathcal{V}$ covers $\mathcal{E}$ with cost W.

In the remaining part of this section we present a heuristic solution for SCP.

3.1 Optimal Sensing Coverage Algorithm

The algorithm we propose for solving the optimal sensing coverage problem, depicted in Algorithm 1 consists of three main steps: in the first step we derive the number of sensor nodes that must be deployed at each vertex, such that edges in the zone of interest are covered (this step is reflected in Lines 1-8); in the second step we derive the minimum number of nodes required to ensure D_c coverage of zone of interest (this step is reflected in Lines 9-15); in the third step we obtain the best insertion points for the mobile sensor nodes (this step is reflected in Lines 16-26)). These steps are described in detail below.

In order to ensure sensing coverage we first derive the probability that a sensor node deployed in an acyclic flow network $\mathcal{F}$ will reach a zone of interest. The simplest case is when we have a single source in the acyclic flow network and the sensor nodes are inserted at this source. In this case, the probability p_i that a single node will traverse a particular edge in $\mathcal{F}$ is $p_i = \frac{f_i}{T_f}$, where f_i is the network flow in edge i and T_f is the total network flow (i.e., inserted at all source vertices).

In a multiple source $\mathcal{F}$, where we allow sensor nodes to be inserted at any vertex, we derive p_i using a matrix $\mathbf{M}$,

Algorithm 1 Sensing Coverage

Require: D_c, $\mathcal{F}$
1: **for** each $e_i \in E$ **do**
2: **for** each $e_j \in E$ **do**
3: $\mathbf{M}_{ij} = prob(e_i, e_j)$
4: **end for**
5: **end for**
6: **while** $\mathbf{M}^k \neq 0$ **do**
7: $\mathbf{T} += \mathbf{M}^k; k++$
8: **end while**
9: **for** each $e_i \in E$ **do**
10: **for** each $e_j \in E$ **do**
11: $i' = e_i.terminal\_vertex$;
12: $j' = e_j.terminal\_vertex$
13: $\mathbf{N}_{i',j'} = \frac{\ln(1-D_c)}{\ln(1-\mathbf{T}_{ij})} + 1$
14: **end for**
15: **end for**
16: **for** each $e_i \in V$ **do**
17: $g_i = \frac{\sum N.row(i) + max(N.row(i))}{(num\ non\ zero(N.row(i)))^{\alpha}}$
18: **end for**
19: **while** $g \neq 0$ **do**
20: i = min(g)
21: $S+ = \{v_i, max(N.row(i))\}$
22: **for** each e_j covered by v_i **do**
23: $j' = e_j.teminal\_vertex$
24: $Ni, j' = 0$
25: **end for**
26: **end while**

that describes the edge to edge transitions as follows:

$$\mathbf{M} = \begin{pmatrix} 0 & e_{12} & \dots & e_{1(|E|+r)} \\ e_{21} & 0 & \dots & e_{1(|E|+r)} \\ \dots & \dots & \dots & \dots \\ e_{(|E|+r)1} & e_{(|E|+r)2} & \dots & 0 \end{pmatrix}$$

where e_{ij} is the probability that a sensor node currently in edge i will be in edge j after passing through the next vertex (i.e., the rows of $\mathbf{M}$ represent the current edge and the columns of $\mathbf{M}$ represent the next edge, after passing through a vertex) and r is the number of source vertices in the flow network (to account for node insertion into the flow network, we add one fictitious edge to each source node). In Lines 1-5 of Algorithm 1, this matrix is constructed. The *prob* function in Line 3 calculates the probability that a sensor node currently in the edge e_i will be in the edge e_j after passing through the next vertex.

The probabilities that a node inserted in $\mathcal{F}$ will traverse a particular edge are computed in the form of a "traversal probability matrix" $\mathbf{T}$, as follows:

$$\mathbf{T} = \sum_{k=1}^{p} \mathbf{M}^k, \text{ such that } \mathbf{M}^p = \mathbf{0}_{|\mathbf{E}|+\mathbf{r},|\mathbf{E}|+\mathbf{r}} \tag{1}$$

where $\mathbf{0}_{|\mathbf{E}|+\mathbf{r},|\mathbf{E}|+\mathbf{r}}$ is the zero matrix of size $|E|+r, |E|+r$. In T, as defined in Equation 1, t_{ij} is the probability that a sensor node inserted into the edge i will traverse edge j of $\mathcal{F}$. Because $\mathcal{F}$ is acyclic, to compute the probability that a node inserted traverses a particular edge, we need to consider a p large enough such that $\mathbf{M}^p$ is a zero matrix (i.e., all sensor nodes have reached sink vertices). Lines 6 - 8 of Algorithm 1 construct the matrix $\mathbf{T}$ using $\mathbf{M}$.

Derivation of Number of Nodes Required for D_c (Step 2)

To determine ν_i - the number of nodes needed to achieve detection threshold D_c on edge e_i, we first define a random variable X for the number of sensor nodes that pass through the edge e_i. Next, notice that X will follow a binomial distribution $b(n, p)$, thus it is described by the probability mass function:

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}, x = 0, 1, \dots, n \tag{2}$$

Let's define an event A when at least one sensor node passes through an edge e_i. We aim to determine ν_i (the number of sensor nodes needed to reach the detection threshold D_c) such that $P(A) \geq D_c$. Note that $P(A) = 1 - P(A')$, where A' is the complement of A, making $P(A')$ the probability that no node will traverse the edge e_i. Using Equation (2) we obtain:

$$P(A) = 1 - P(A') = 1 - (1-p_i)^{\nu_i} \geq D_c \tag{3}$$

From Equation (3), ν_i can be obtained as:

$$\nu_i \geq \left\lceil \frac{\ln(1-D_c)}{\ln(1-p_i)} \right\rceil \tag{4}$$

When the probability p_i of reaching an edge is known we can then calculate the number of sensor nodes that need to be inserted to reach a desired detection threshold D_c. If $p_i \geq D_c$ then

one sensor node is sufficient for detection. When $p_i < D_c$ then a larger number of nodes need to be inserted to reach the detection threshold.

Using Equation (4) the traversal probability matrix $\mathbf{T}$ is converted to a "node requirement matrix" $\mathbf{N}$, in which each element $n_{i'j'}$ is the number of nodes to be inserted into the ith edge's terminal vertex $v_{i'}$ to reach jth edge's terminal vertex $v_{j'}$ with probability D_c. Formally, $\mathbf{N}$ is defined as:

$$\mathbf{N} = \begin{pmatrix} n_{11} & ... & n_{1|V|} \\ ... & n_{ij} & ... \\ n_{|V|1} & ... & n_{|V||V|} \end{pmatrix}$$

where $n_{i'j'} = max(n_{i'j'}, \left\lceil \frac{\ln(1-D_c)}{\ln(1-t_{ij})} \right\rceil)$ where v'_i is the terminal vertex of e_i and v'_j is the terminal vertex of e_j. Wherever an element of $\mathbf{T}$ is 0 (i.e. the destination edge is not reachable from the insertion edge,), a 0 is also present in $\mathbf{N}$, as a marker for unreachability. This formalism allows each edge to have a different detection threshold by applying different values of D_c to each column of $\mathbf{T}$. This may be useful in cases where detection in some edges is much more important than others. Lines 9-15 of Algorithm 1 construct $\mathbf{N}$ using $\mathbf{T}$ and the binomial distribution equation 4.

To obtain a solution for our sensing coverage problem we must select a set of "good" insertion points from the matrix $\mathbf{N}$ that cover our zone of interest in the flow network.

Insertion Point Selection Heuristic (Step 3)

Since our problem is NP-hard, we define a "goodness" metric g, to allow our algorithm to make a greedy choice. We define g as:

$$g_i = \frac{(r_i + r_m)}{\epsilon^\alpha} \text{ where } i = 1, ..., |V| \tag{5}$$

where r_i is the sum entries in row i of $\mathbf{N}$ (i.e., $r_s = \sum_j n_{ij}$), r_m is the maximum element of $\mathbf{N}$, ϵ is the number of non-zero elements in the row of $\mathbf{N}$, and α is a tuning parameter that allows us to control the importance of coverage in calculating g_i. For a single source acyclic flow network, a lower value of α may be necessary to pull focus away from the source vertex. Each vertex now has a corresponding goodness value, based on $\mathbf{N}$. The goodness vector is constructed using the goodness metric in Lines 16-18 of Algorithm 1.

The smallest g_i corresponds to the best vertex v_i (i.e., g_i relates to the number of nodes to be deployed to cover each edge). By reducing this number, we ensure that as few nodes as possible are inserted. The maximum element in the ith row of $\mathbf{N}$ is the number of nodes to be inserted at v_i. The smallest value of g_i is chosen from g in Line 20. In Line 21, that vertex is added to S, where s_i is the vertex v_i and q_i is the maximum element in row i of $\mathbf{N}$. Finally, vertices covered by v_i are removed from $\mathbf{N}$ in lines 22-25. This process is repeated, selecting vertices until a coverage threshold is met or until $\mathbf{N} = \mathbf{0}$, which indicates that all remaining vertices are unreachable. We then select $s_i = v_i$ and $q_i = max(N.row(i))$ to obtain $\{s_i, q_i\}$ to be inserted into set S. When the algorithm terminates, it produces a set of vertices, each with an associated number of nodes to be inserted into the flow network, that meet the sensing coverage requirement.

Algorithm Complexity: This greedy heuristic is an approximation to the optimal solution for node placement in a FSN. The heuristic uses the same technique used to approximate the weighted set cover problem. The approximation ratio for this heuristic was proved to be $\ln |V|$ [12]. The flows in all the edges is known, so Line 3 takes $O(1)$ time. So, Lines 1-5 take $O(|E|^2)$

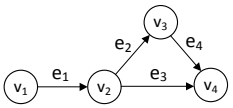


Fig. 3. Graph example

time. Time complexity of calculating $\mathbf{M}^i \times \mathbf{M}$ to get $\mathbf{M}^{i+1}$ is $O(|E|^3)$. So, Lines 6-8 take $O(|E|^3 p)$ time. Construction of $\mathbf{N}$ using $\mathbf{T}$ and the binomial distribution equation 4 takes time $O(|E|^2)$. The goodness vector is constructed using the goodness metric in time $O(|V|)$. Choosing the vertices in Lines 19-26 takes $O(|V|^2)$ The most time consuming step in the algorithm is the construction of the Traversal Probability Matrix, the time complexity of the Sensing Coverage Algorithm is $O(|E|^3 p)$. Asymptotically, this is $O(|E|^3)$.

Example 2: For clarity of presentation we exemplify how our algorithm works for a flow network shown in Figure 3, where we let the flow be equally divided in edges e_2 and e_3. e_1 is a virtual edge added to introduce incoming flow into the network. So, the traversal probability matrix $\mathbf{T}$ derived from Equation 1, and the node requirement matrix $\mathbf{N}$ with a D_c of 0.75, as defined in Equation 4 and the goodness vector when $\alpha = 2$ derived from Equation 5 will be:

$$\mathbf{M} = \begin{pmatrix} 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \mathbf{T} = \begin{pmatrix} 0 & 0.5 & 0.5 & 0.5 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \mathbf{N} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}; g = \begin{pmatrix} \infty & 1.5 & 2 & \infty \end{pmatrix} \tag{6}$$

Now, vertex v_2 is chosen since it has the smallest g. The maximum element in the second row of $\mathbf{N}$ is 2. Hence 2 nodes are inserted at vertex v_2. All edges reachable from v_1 are removed from $\mathbf{N}$. This reduces the matrix to a zero matrix. Hence, we obtain $S = \{v_2, 2\}$. Since e_1 was a virtual edge, this ensures that at least D_c edges are covered.

4 Beacon Placement for Optimal Event Localization

Our solution for the problem of optimal event localization in acyclic flow networks consists of two steps: in the first step we seek an optimal placement of beacon nodes (i.e., reduce the number of beacon nodes); in the second step, using the path synopsis collected from sensor nodes, we identify the location of an event. More formally, these steps are described as follows: i) Beacon Placement Algorithm: given $\mathcal{F}$ - an acyclic flow network and P_d - the required accuracy of event localization, find the set $B = \{b_i \mid b_i \in V\}$ of vertices where beacon nodes are to be placed such that the probability of finding an event is greater than P_d; ii) Event Localization Algorithm: given $\mathcal{F}$ - an acyclic flow network and $\{PS_i \mid i \leq \sum_{i=1}^{|S|} q_i\}$ - a set of path synopses for all sensor nodes deployed in the $\mathcal{F}$, localize an event X, detected by sensor nodes, by identifying a set of edges, i.e. *Suspects List*, where the event might be present.

In the remaining part of this section, we present our algorithms for solving the optimal event localization problem.

4.1 Beacon Placement Algorithm

The Beacon Placement algorithm is presented in Algorithm 2. This algorithm optimizes the placement of beacon nodes in the network, so that the event localization algorithm can achieve a P_d accuracy. The algorithm uses an approach similar to Breadth First Search. In Line 1, a vertex queue Q is initialized with sources of $\mathcal{F}$, the acyclic flow network. In Line 2, a beacon node is placed in all sources of $\mathcal{F}$.

Every vertex has a potential ϕ, where ϕ for a vertex v_i is the number of edges that can be localized by the event localization algorithm when v_i is reached from any of the sources. If v_j is a

Algorithm 2 Beacon Placement

Require: P_d, $G(V, E)$,
1: $Q \leftarrow V.sources$
2: $V.sources.place\_beacon$
3: $\tau \leftarrow (1 - POD)|E|$
4: **while** $Q \neq \Phi$ **do**
5: $v_i \leftarrow deque(Q); no\_beacons \leftarrow \Phi$
6: **for** each $p_j \in v_i.parents$ **do**
7: **if** $\neg\ p_j.completed$ **then**
8: $Q.insert(p_j)$
9: **end if**
10: **if** $\neg\ p_j.has\_beacon$ **then**
11: $no\_beacons.add(p_j)$
12: **end if**
13: **end for**
14: **while** $v_i.\phi < \tau$ AND $no\_beacons \neq \Phi$ **do**
15: $p_j \leftarrow no\_beacons.EXTRACT\_MAX$
16: $p_j.place\_beacon$
17: $v_i.\phi \leftarrow v_i.\phi - p_j.\phi - 1$
18: **end while**
19: **if** $v_i.\phi > \tau$ **then**
20: $v_i.has\_beacon \leftarrow true$
21: **end if**
22: **for** each $v_j \in v_i.children$ **do**
23: $v_j.\phi \leftarrow v_j.\phi + v_i.\phi + 1$
24: **if** $\neg v_j.queued$ **then**
25: $Q.enque(v_j)$
26: **end if**
27: **end for**
28: $v_i.completed \leftarrow true$
29: **end while**

Algorithm 3 Event Localization

Require: PS, N, $G(V, E)$
1: $SuspectsList \leftarrow E.$
2: BT $\leftarrow$ initialize Beacon Table
3: **for** each $n_i \in N$ **do**
4: **for** each $p \in PS_i$ **do**
5: **if** $p \neq X$ **then**
6: **if** $BT[p][p.next].path = 1$ **then**
7: **for** each $e_j \in BT[p][p.next]$ **do**
8: $SuspectsList.remove(e_j)$;
9: **end for**
10: **end if**
11: **else**
12: $BT[p][p.next].event$
13: **end if**
14: **end for**
15: **end for**
16: **for** each $b_i \in BT$ **do**
17: **if** $b_i.event = false$ **then**
18: **for** each $e_j \in b_i$ **do**
19: $SuspectsList.remove(e_j)$;
20: **end for**
21: **else**
22: $b_i.event$
23: **end if**
24: **end for**

parent of a vertex v_i, then $v_i.\phi > v_j.\phi$. Hence, we can iteratively obtain ϕ for a vertex using ϕ of its parents. When a beacon node is placed at v_j, v_i's potential will decrease. A threshold τ for ϕ is derived from P_d in Line 3.

Lines 4-29 iterate over the vertices of a graph, similar to the Breadth First Search. If the parent of a vertex v_i was not iterated over, the vertex is added back to the queue, with a priority, in Line 8. This is because we cannot make a decision about beacon placement in v_i without knowing the potential value $v_i.\phi$. We maintain a heap with the parents of v_i that do not have beacons with the key as $p_j.\phi$ in Line 11. Once we check all the parents of v_i, we are sure that the potential of v_i is computed. Now we start placing beacon nodes at the parent vertices of v_i until the potential of v_i decreases below τ. The selection of the parents is done greedily, so that as few parents as possible have beacons. This is done in Lines 14-18. If $v_i.\phi$ is still greater than τ, a beacon node is placed at v_i. Lines 22-27 add the children of v_i to the queue, similar to Breadth First Search. Once a vertex

is iterated over, it is marked as completed in Line 28. Since we consider directed acyclic graphs, Line 8 will not introduce an infinite loop.

This algorithm provides an optimal solution to the Beacon Placement problem for directed acyclic graphs, since we ensure optimal result for each subgraph of G. The time complexity of this algorithm depends on the number of times a vertex is added back in the queue and the number of parents a node has. Adding and removing parents from heap takes $O(lgn)$ time, n being number of parents. A vertex can be added back to the queue at most $O(V)$ times. There is no cyclic dependency, because the graph is directed and acyclic. The number of parents of a node is also $O(V)$. So, the time complexity of the algorithm is $O(V^3(lgV))$.

4.2 Event Localization Algorithm

The algorithm for Event Localization is presented in Algorithm 3. In line 1, we initialize the "Suspects List" (i.e., edges where an event might be present) to contain all the edges in the network. We follow an elimination method to localize events to as few edges as possible. In line 2, we initialize a Beacons Table (BT). Each entry in the BT contains the number of paths, list of edges between them and indication of whether an event is present between them. The number of paths and list of edges between each pair of beacon nodes is obtained from the graph. The event indicator is initialized to $false$. Next, in lines 3-15, we iterate over all the nodes to analyze their path synopses. For each entry in the path synopsis p of a node n_i, line 5 checks if no event was detected. If no event is detected between two beacons and there is only one path between them, then the edges in that path definitely do not have an event. Hence, line 8 eliminates such edges from the Suspects List. If an event is found in the path synopsis, we mark an event in the corresponding BT entry, in line 12.

Upon iterating over all path synopses obtained from all the nodes, the BT entries will reflect whether or not an event was detected on a path between pairs of beacon nodes. Consequently, in lines 16-24 we iterate over the entries in BT. An entry in the BT will be marked for an event only if one of the nodes detected an event between the beacons for that entry. If the entry in BT is not marked with an event, line 19 removes edges between those beacons from the Suspects List. At the end of the iteration, we will be left with the smallest possible Suspects List, i.e., the highest event localization accuracy.

The time complexity of this algorithm depends on the number of nodes, the number of beacons in each path synopsis and the number of edges between any two beacons. The number of edges between any two beacons is $O(E)$. Number of nodes is $O(V)$ and number of Beacons in the Path Synopsis is also $O(V)$. So, the worst case time of the algorithm is $O(V^2E)$.

Example 3: Consider the flow network in Figure 2. Between source v_6 and sink v_3, there are 6 possible paths. When sensing coverage is ensured, nodes are inserted in such a way that all these paths are covered. Without loss of generality (since we solve here the event localization problem), we can assume that all the nodes were inserted in the source. Let there be an event in edge (v_5, v_4). The paths P_i covered by sensors (and their path synopsis PS_i) are: $P_1 = \{v_6, v_1, v_2, v_3\}$ with $PS_1 = \{B_1, B_4, B_2\}$; $P_2 = \{v_6, v_1, v_5, v_4, v_3\}$ with $PS_2 = \{B_1, B_4, X, B_3, B_2\}$; $P_3 = \{v_6, v_5, v_4, v_3\}$ with $PS_3 = \{B_1, X, B_3, B_2\}$; $P_4 = \{v_6, v_8, v_5, v_4, v_3\}$ with $PS_4 = \{B_1, B_5, X, B_3, B_2\}$; $P_5 = \{v_6, v_8, v_7, v_4, v_3\}$ with $PS_5 = \{B_1, B_5, B_3, B_2\}$; and $P_6 = \{v_6, v_8, v_7, v_3\}$ with $PS_6 = \{B_1, B_5, B_2\}$.

In the first part of the algorithm, the following edges are removed: (v_6, v_1), (v_1, v_2), (v_2, v_3), (v_6, v_8), (v_8, v_7), (v_7, v_4), (v_7, v_3), (v_4, v_3). Next, we use the Beacon Table entries, but we cannot remove more edges. So finally, in the suspects list, we have (v_1, v_5), (v_8, v_5), (v_5, v_4), (v_6, v_5).

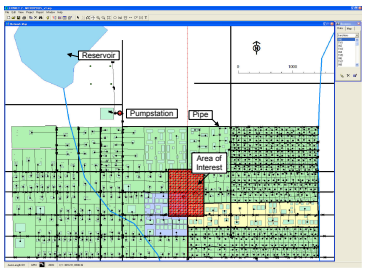


Fig. 4. FlowSim integrated with EPANET depicts a zone of Interest in the middle of the figure.

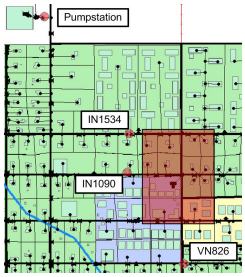


Fig. 5. Zone of Interest magnified, along with a set of Insertion Points.

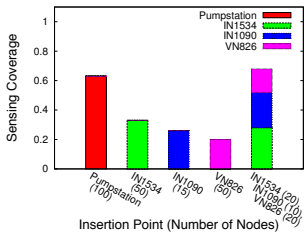


Fig. 6. Impact of number of nodes and placement on coverage.

We remark here that if we know that there was only one event in the network, we can localize the event more precisely by taking only the common edges from the BT entries that have events. In the above example, we can reduce the suspects list to (v_5, v_4), thereby achieving 100% success.

5 Performance Evaluation

For performance evaluation, we have developed FlowSim. FlowSim uses accurate simulation of sensor movement in a municipal water system, by loading results from EPANET [13] - an acyclic flow network simulator.

FlowSim was validated on simple networks for which the exact behavior is known and results can be derived theoretically. The obtained results are within a 90% confidence interval. For validation, we used Micropolis [14], a *virtual network/city model*. A map of Micropolis is shown in Figure 4, with water storage areas, a pumpstation and a flow network using interconnected water pipes. We validated the sensing coverage algorithm by considering a zone of interest (a darker vertical rectangle in Figure 5). For a degree of coverage $D_c = 0.6$ of the zone of interest, the sensing coverage algorithm produced as insertion points IN1534, IN1090, and VN826, shown in Figure 5. The sensing coverage results obtained from FlowSim are depicted in Figure 6. As shown, when the optimal number of nodes (50 nodes in total: 20 at IN1534, 10 at IN1090 and 20 at VN826) is placed at the three insertion points, we can achieve the desired sensing coverage. The achieved sensing coverage is higher than the scenario when we insert 100 nodes at the pumpstation, and higher than the scenarios the same number of nodes (i.e., 50) are all inserted at a single insertion point.

The metrics we use in our evaluation are Sensing Coverage and Event Localization Accuracy/Success. Since, to the best of our knowledge no other solutions exist for our problem, for performance evaluation, we use for comparison two Random algorithms, one that chooses randomly the insertion points (for Sensing Coverage) and one that chooses randomly the locations of beacon nodes (for Beacon Placement and Event Localization). We also investigate how different parameters affect our algorithms: α- the tuning parameter to chose good insertion points, D_c- the degree of coverage and P_d- the probability of event detection affect our metrics of interest. We consider two acyclic flow network topologies, shown in Figures 2 and 7.

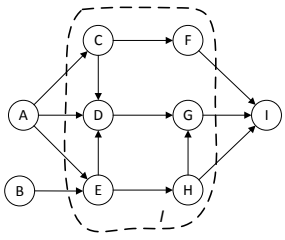


Fig. 7. Acyclic flow network for evaluation of sensing coverage.

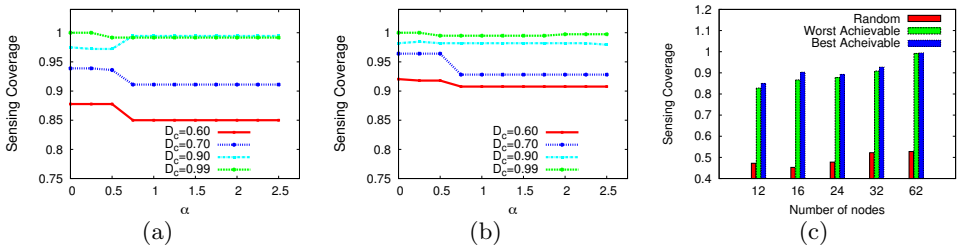


Fig. 8. (a) Evaluation of sensing coverage wrt α and D_c on graph in Figure 2; (b) Evaluation of sensing coverage wrt α and D_c on graph in Figure 7; (c) Comparison of sensing coverage with random node deployment vs our algorithm on graph in Figure 2.

All our performance evaluation results are averages of 30 simulation runs with random seed values.

5.1 Impact of α and D_c on Sensing Coverage

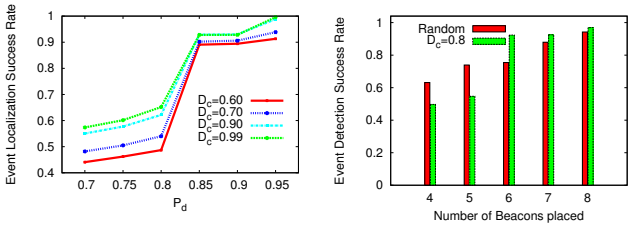
In this set of simulations we investigate how our algorithm for ensuring Sensing Coverage is affected by α and D_c, and how its performance compares with a random deployment of nodes. Since Sensing Coverage does not depend on events present, we do not consider P_d.

The performance results for the two topologies we considered are presented in Figure 8(a) and Figure 8(b). We observe that the sensing coverage reduces with higher values of α. This is because fewer nodes are inserted. As α increases, for a given D_c, we observe gradual reduction in number of nodes q_i inserted. Consider the acyclic flow network topologies, shown in Figure 2. For $D_c = 0.9$, for $\alpha = 0.5$, 1, and 10, the number of nodes inserted were 32, 17 and 16 respectively. We can therefore say that choosing a low value of α provides higher coverage at the cost of higher number of nodes. Fewer nodes does not always result in lower coverage. Eg. for $D_c = 0.5$ and $\alpha = 2.5$, for 10 nodes inserted, coverage is 0.910, whereas when $D_c = 0.6$ and $\alpha = 1.5$, for 13 nodes inserted, coverage is 0.907.

The actual Sensing Coverage is usually higher for a higher D_c. This is because, increasing the value of D_c uniformly increases the number of nodes to be inserted in any vertex. Increasing the value of D_c increases the number of inserted for a given value of α. Consider the acyclic flow network topologies, shown in Figure 2. For $\alpha = 1$, for $D_c = 0.5$, 0.6, 0.7, 0.8, 0.9 and 0.99, the number of nodes inserted were 10, 13, 17, 22, 32 and 63 respectively. We can therefore say that choosing a high value of D_c provides higher coverage at the cost of higher number of nodes.

We also evaluated our algorithms against a random node deployment, with the results depicted in Figure 8(c). From the sensing coverage algorithm we obtain the number of nodes to be deployed in a network, for given, fixed α and D_c. We compare the best and worst achievable results from our algorithm with a random deployment for a fixed number of nodes inserted. As shown in Figure 8(c), our algorithm ensures a significantly improved sensing coverage.

5.2 Impact of P_d and D_c on Event Localization



(a) (b)

Fig. 9. (a) Evaluation of event localization wrt P_d and D_c; (b) Comparison of Beacon Placement algorithm to random Beacon Placement

In this set of simulations we investigate how P_d and D_c affect the accuracy of event localization accuracy. In our simulations, we observed that for a higher value of P_d, the accuracy of Event Detection was higher. When beacon nodes were placed on all vertices, and sensing coverage was 100%, we observed that the accuracy of event localization was 100%. Considering Definition 8, for e events and s edges in the suspects list, the observed accuracy is $\frac{e+|E|-s}{|E|}$.

The results are depicted in Figure 9(a). As shown, the actual event detection success rate is lower than the D_c for lower value of P_d, but, nevertheless, it is expected, since the sensing coverage is not 100% (for 60% sensing coverage, with 70% detection probability of detection, event expected localization accuracy is ∼42%). The results of our beacon placement and random placement algorithms, are presented in Figure 9(b). For this simulation, the value of α is 0 and D_c is 0.8. Our algorithm returns a fixed number of beacons for each value of P_d. As shown, our algorithm performs better than the random deployment for higher values of P_d.

6 Related Work

To address the challenges that water infrastructure faces, following the events of 9/11 in the US, an online contaminant monitoring system was deemed of paramount importance. As consequence, the Battle of the Water Sensor Networks (BWSN) design competition was undertaken. In one BWSN project [15], the authors aim to detect the contamination in the water distribution network. The proposed approach is distinct from ours in that their aim is to select a set of points to place static sensor nodes. One other solution, similar to those proposed in the BWSN competition, which are based on static sensor nodes, strategically deployed, was Pipenet [7]. More recently, mobile sensors probing a water distribution infrastructure have been proposed [16]. The WaterWise [8] system considers nodes equipped with GPS devices.

Node mobility in an acyclic flow network might resemble node movement in a delay tolerant networking (DTN) scenario. The problem of event localization in DTN, however, has received little attention, primarily because it can be done using GPS. DTN typically involves vehicles, for which energy is not an issue. Consequently, solutions to problems similar to the event localization problem addressed in this paper have not been proposed. For completeness, we review a set of representative DTN research. Data Dolphin [17] uses DTN with fixed sinks and mobile nodes in 2D area. A set of mobile sinks move around in the area. Whenever a sink is close to a node, it exchanges information over one hop, thereby reducing the overhead of communicating multi hop and saving energy on the static nodes. A survey done by Lee et al. encompasses the state of art in vehicular networks using DTN [18]. Sensing coverage problems in DTN are handled by CarTel [19], MobEyes [20] etc. These systems use vehicles that can communicate with each other and localization is based on GPS.

Coverage problems in sensor networks have been considered before [21] [22]. These papers consider coverage problems in 2D or 3D area, unlike coverage on graphs, as in this paper. Sensing

coverage, in general, has been studied under different assumptions. [23] uses both a greedy approach, and linear programming to approximate the set covering problem. These problems consider only minimizing the number of vertices to cover edges in a graph.

7 Conclusions

This paper, to the best of our knowledge for the first time, identifies and solves the optimal event detection and localization problems in acyclic flow networks. We propose to address these problems by optimally deploying a set of mobile sensor nodes and a set of beacon nodes. We prove that the event detection in NP-Hard, propose an approximation algorithm for it, and develop algorithms for optimally solving the event localization problem. Through simulation we demonstrate the effectiveness of our proposed solutions. We leave for future work the development of algorithms for time-varying flow networks with flow direction changes.

References

1. U.S. Government Accountability Office, "Drinking water: Experts' views on how federal funding can be spent to improve security." Sept. 2004.
2. U.S. Environmental Protection Agency, "Response protocol toolbox: Planning for and responding to contamination threats to drinking water systems," *Water Utilities Planning Guide-Module 1*, 2003.
3. E. Zechman and S. Ranjithan, "Evolutionary computation-based methods for characterizing contaminant sources in a water distribution system," *J Water Resources Planning and Management*, pp. 334–343, 2009.
4. E. Zechman, "Agent-based modeling to simulate contamination events and evaluate threat management strategies in water distribution systems," *Risk Analysis*, 2011.
5. A. Ostfeld and E. Salomons, "Optimal layout of early warning detection stations for water distribution systems security," *J Water Resources Planning and Management*, pp. 377–385, 2004.
6. A. Ostfeld and et al., "The battle of the water sensor networks (BWSN): A design challenge for engineers and algorithms," *Journal of Water Resources Planning and Management*, 2006.
7. I. Stoianov, L. Nachman, S. Madden, and T. Tokmouline, "PIPENET: A wireless sensor network for pipeline monitoring," in *IPSN*, 2007.
8. A. J. Whittle, L. Girod, A. Preis, M. Allen, H. B. Lim, M. Iqbal, S. Srirangarajan, C. Fu, K. J. Wong, and D. Goldsmith, "WATERWISE@SG: A testbed for continuous monitoring of the water distribution system in singapore," in *Water Distribution System Analysis (WSDA)*, 2010.
9. T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, and L. Gu, "An energy-efficient surveillance system using wireless sensor networks," in *MobiSys*, 2004.
10. S. George, W. Zhou, H. Chenji, M. Won, Y. O. Lee, A. Pazarloglou, R. Stoleru, and P. Barooah, "DistressNet: a wireless ad hoc and sensor network architecture for situation management in disaster response," *Communications Magazine, IEEE*, vol. 48, no. 3, pp. 128 –136, 2010.
11. R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," in *ACM SenSys*, 2004.
12. R. Bar-Yehuda and S. Even, "A linear-time approximation algorithm for the weighted vertex cover problem," *Journal of Algorithms*, vol. 2, pp. 198 – 203, 1981.
13. "EPANET v2.0," Environmental Protection Agency, Tech. Rep., 2006.
14. K. Brumbelow, J. Torres, S. Guikema, E. Bristow, and L. Kanta, "Virtual cities for water distribution and infrastructure system research," in *World Environmental and Water Resources Congress*, 2007.
15. J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *KDDM*, 2007.

16. T.-t. Lai, Y.-h. Chen, P. Huang, and H.-h. Chu, "PipeProbe: a mobile sensor droplet for mapping hidden pipeline," in *SenSys*, 2010.
17. E. Magistretti, J. Kong, U. Lee, M. Gerla, P. Bellavista, and A. Corradi, "A mobile delay-tolerant approach to long-term energy-efficient underwater sensor networking," in *WCNC*, 2007.
18. U. Lee and M. Gerla, "A survey of urban vehicular sensing platforms," *Comput. Netw.*, vol. 54, pp. 527–544, March 2010.
19. B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "CarTel: a distributed mobile sensor computing system," in *SenSys*, 2006.
20. U. Lee, B. Zhou, M. Gerla, E. Magistretti, P. Bellavista, and A. Corradi, "Mobeyes: smart mobs for urban monitoring with a vehicular sensor network," *Wireless Communications*, vol. 13, no. 5, 2006.
21. A. Tahbaz-Salehi and A. Jadbabaie, "Distributed coverage verification in sensor networks without location information," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, Aug 2010.
22. S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *INFOCOM*, 2001.
23. M. Cardei, M. T. Thai, Y. Li, and W. Wu, "Energy efficient target coverage in wireless sensor networks," in *INFOCOM*, 2005.