The Sentinel Problem for a Multi-hop Sensor Network

Dimitri Marinakis, Sue Whitesides
Department of Computer Science, University of Victoria
dmarinak@gmail.com, sue@uvic.ca

***Abstract*— In the context of a multi-hop sensor network alarm application, we define the *Sentinel Problem*: How can a network of simple devices with limited communication ability signal the occurrence of an event that is capable of disabling the sensors?**

We present both deterministic and probabilistic methods for solving this problem, and evaluate the methods based on algorithmic correctness, false positive rates, latency, and implementation potential.

***Index Terms*— wireless sensor networks, broadcast scheduling, stochastic algorithms, alarm propagation**

I. Introduction

We present an alarm application problem that operates on a multi-hop wireless sensor network. The problem asks: how can a network of simple devices with limited communication ability signal the occurrence of an event that is capable of disabling the sensors? We call this the Sentinel Problem. [1]

We assume that each network component, or sentinel, emits an occasional status message under normal circumstances but signals an alarm event by ceasing to transmit. Specifically, a sentinel that has not sensed an alarm event may select to send a broadcast message containing only its identity according to some schedule, but if an alarm event occurs in the region of that device, then the sentinel no longer transmits. A sentinel that recognizes that one of its neighbours has ceased to transmit will also cease to transmit. In this manner the alarm condition propagates throughout the network until it ultimately reaches a gateway device where an appropriate action can be taken; *e.g.* alerting higher level processes or, in the case of a hybrid sensor network / mobile robot system such as that proposed by Meger *et al.* [1], initiating investigatory behaviour.

A sensor network employing the sentinel protocol could be deployed, for example, to detect a potential emergency event such as a fire or chemical leak, where the sensor itself might become damaged and cease to operate. In contrast to the sentinel approach, an alarm system could rely on a sensor initiated transmission. For example, a system could be built over a flood based protocol such as that proposed by Rahman *et al.* [2]. The flood based system would, however, be vulnerable to an event capable of destroying a sensing device before it could transmit.

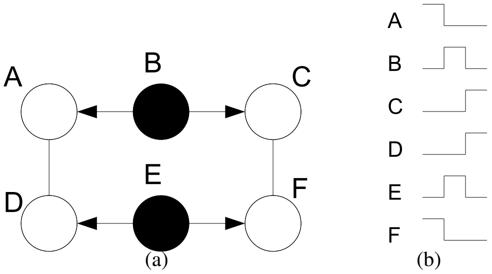

Fig. 1. (a) a maximal broadcasting set with broadcasting devices shown in black and arrows depicting the tranfer of messages, (b) a Gant chart showing a deterministic schedule for this network.

Overview and Results: In the remainder of the paper we first give a formal description of the Sentinel Problem and then present and analyze both a deterministic and probabilistic solution to the problem. We then evaluate these two approaches through experiments conducted using a network simulator. Our results suggest that the probabilistic solution performs adequately and should be easier to implement.

II. The Sentinel Problem on Graphs

We model the multi-hop network as a graph $G = (V, E)$ in which each vertex $v \in V$ represents a network component and each edge $e \in E$ denotes a two-way communication link; *i.e.* the devices i and j are neighbours if $e_{ij} \in E$. We assume that if an alarm event occurs in the region near sensor i it will enter a *triggered* state and will cease to communicate for the remainder of the problem instance.

We assume the following constraints on communication. If device i transmits, then a neighbouring device j will receive the message from i if and only if during the communication j itself is not transmitting and i is the only neighbour of j that is transmitting. This constraint provides a simple way to model congestion issues such as the hidden terminal problem when communicating within a single radio frequency band and has been used before; *e.g.* by Chen *et al.* in [3]. See Bharghavan *et al.* [4] for further information on the hidden terminal problem.

We assume that all devices maintain synchronized clocks and may select to time their communications to occur during a particular timeslice. All communications are assumed to take equal time. Note that this assumption of synchronized clocks is common in the area of wireless sensor network research, and there are a number of techniques that could be used to accomplish this task; see Sivrikaya and Yener [5] for a survey. Additionally, we assume the existence of only one

[1] The name of the problem is motivated by the 1951 Arthur C. Clark short story ‘The Sentinel of Eternity’ in which a beacon on the moon ceases to transmit a signal when its force field is breached.

radio frequency channel, although a variant of the problem could allow devices to switch between a finite number of channels.

Devices that have not directly sensed the alarm event in their region may select to be in a transmitting, or *armed* state, during which they may occasionally emit their broadcast message according to some algorithm, or they may select to be in a non-communicating *alarmed* state in which they stay silent. We define a device in the network to be *silent* when it is either in the alarmed state, or the triggered state.

We assume that the network graph G is connected. If it is not, each connected component can be considered separately.

We evaluate potential solutions according to the following criteria:

1) *Correctness:* When a device $i \in V$ is triggered will all devices $j \in V$ fall silent with probability 1 as the elapsed time since the triggering, $t \to \infty$?
2) *Network False Positive Rate:* What is the probability that *every* device $i \in V$ falls silent when no event has occurred; *i.e.* when there is no device $j \in V$ that has entered the triggered state?
3) *Latency:* How long, when successfully detected, does it take for all devices $i \in V$ to fall silent when an event occurs; *i.e.* when a device $j \in V$ enters the triggered state?
4) *Implementation Potential:* As a practical matter, the processing that occurs on an individual device should be minimal, and ideally accomplished without a floating point processor, and only limited memory and code space.

III. The Deterministic Approach: Broadcast Scheduling

One way to solve the Sentinel Problem is to assign device specific communication schedules, such that each device receives at least one message from each armed neighbour every M timeslices. If a neighbour i is not heard from by a device j during the timeslice allocated to i, then device j will fall silent. We will argue that this approach is not ideal when one considers some of the implementation details required for coordination purposes, however, it provides a benchmark for evaluating other approaches.

A. Background

The class of problems related to assigning a timeslot to each component in a wireless network for congestion avoidance is referred to as *broadcast scheduling*. Such problems were considered as early as the mid-eighties by Chlamtac and Kutten [6], for example. Later in that decade, Ramaswami and Parhi [7] showed that the problem of finding a minimum length schedule that allows each device to hear from each neighbour is NP-complete. The authors presented a centralized heuristic for the problem as well as a token based, distributed approach. Ramanathan and Llyod [8] improved on earlier broadcast scheduling algorithms and included a simulation-based anaylsis. More recent scheduling work such as that by Ephremides and colleagues consider variants of the problem where there are multiple channels, *e.g.* [9], or power considerations, *e.g.* [10].

We now briefly describe the broadcast schedule assignment heuristic presented by Ramaswami and Parhi [7] and apply it to the Sentinel Problem. The authors define a *broadcasting set* to be a set of nodes that can broadcast simultaneously without congestion, and they define a *maximal broadcasting set* to be a broadcasting set such that if any node is added, it is no longer a broadcasting set.

The heuristic for finding a broadcast schedule presented in [7] assigns each device to the first slot in the schedule in which it will not interfere with any nodes already assigned to that slot. Once all devices are allocated, it revisits the schedule and ensures that each slot contains a maximal broadcasting set. Fig. 1 shows an example of a graph and the schedule that results from applying this heuristic.

B. The Broadcast Scheduling Algorithm (BSA)

When using a broadcast scheduling approach for the Sentinel Problem, one obtains a broadcast schedule $\Lambda = \{\lambda_i\}, i \in V$ for each device in the network. This schedule could be obtained by the method described above, or some other technique. Each device knows its own schedule and that of each of its neighbours. Should a neighbour j of i fail to transmit during one of its assigned slots in λ_j, then the device i would fall silent.

An alternative, simpler implementation would be that a device falls silent if a neighbour is not heard from for $M = |\Lambda|$ timeslots. The advantage of this simpler approach is that it does not require storing the schedule for each neighbour on each device. We will refer to this second variant as the Broadcast Scheduling Algorithm (BSA) in the remainder of the paper.

C. Analysis

1) Correctness: It can be seen that the deterministic algorithm is correct given our problem formulation. Once a single device $i \in V$ enters the triggered state and falls silent, its neighbours will enter the alarmed state within $M = |\Lambda|$ timeslots, and the alarm state will propagate to all components connected to i in the network.

2) Network False Positive Rate: The false positive rate for the deterministic algorithm is zero. No device will fall silent unless one of its neighbours has legitimately entered either the alarmed state or the triggered state.

3) Latency: A worst case latency bound is DM, where M is the length of the broadcast schedule and D is the diameter of the communication graph G. It should be possible to find a tighter bound for some graph classes by considering the local topology; *i.e.* in some regions of the network, the communication schedule might allow each device to communicate with all its neighbours in less than M timeslices.

4) Implementation Potential: The approach of deterministically assigning a topology dependent broadcast schedule to each network component could be challenging to implement in a real sensor network application. One approach would be to assign device specific schedules from a centralized point such as a gateway device. This could be done using a two phase method: in the first part information regarding the communication topology would be collected using neighbour tables and flooding; and in the second part a centralized algorithm would determine the appropriate schedule and assign it.

Another approach would be to use a distributed algorithm for assigning broadcast schedules such as the one presented by Ramaswami and Parhi in [7]. This type of distributed method relies on token passing and would also require considerable implementation complexity.

In the next section we will present a probabilistic solution to the Sentinel Problem with a potentially much simpler implementation.

IV. The Probabilistic Approach

As opposed to the deterministic approach, we propose to address the Sentinel Problem by assigning a probability of broadcasting per timeslice to each device in the network; *i.e* $P = \{p_i\}, \forall i \in V$. This will require synchronizing the devices, but otherwise is quite easy to implement using, for example, a XOR shift key as a pseudorandom number generator. In order to decide when to switch from the armed state to the alarmed state, each device keeps track of how long it has been since it has heard from each of its neighbours. If this time exceeds a device dependent threshold γ then the device switches states.

A. Background

This approach is motivated by research that considers the application of stochastic techniques to other aspects of sensor networks such as flooding, *e.g.* the work of Sasson *et al.* [11], or data aggregation, *e.g.* the work of Boyd *et al.* [12]. Also related are distributed, low complexity approaches to scheduling such as the recent work of Tang *et al.* [13].

B. Two Algorithms

1) The Basic Probabilistic Algorithm (BPA): Each device maintains a neighbour table with one entry for each of its neighbours along with an associated count. At each timeslice, all the counts are incremented by one. Also at each timeslice, with probability p_i, a device broadcasts its unique media access control (MAC) identification. Upon receiving a message from a neighbour j, device i sets the count associated with j in its neighbour table to zero. Should any count in its table exceed the threshold γ, device i enters the alarmed state and falls silent forever. We will refer to this approach in the remainder of the paper as the Basic Probabilistic Algorithm (BPA).

2) The Neighbour Table Exchange Probabilistic Algorithm (NTXPA): At the cost of communicating more information, the basic approach described above can be improved as follows. Like before, each device maintains a neighbour table with an associated count. During each timeslice, with probability p_i, a device broadcasts its neighbour table along with its unique MAC identification. Upon receiving a message from a neighbour j, a device i sets the count of all neighbours in common with j to the lesser of the value reported by j and what is present in the table. Should any count in its table exceed threshold γ, device i falls silent. In contrast to the basic algorithm, when in the alarmed state, the device continues to update its neighbour table, and if the maximum count falls beneath threshold γ it re-enters the armed state, and continues to broadcast according to p_i once more. We will refer to this approach in the remainder of the paper as the Neighbour Table Exchange Probabilistic Algorithm (NTXPA).

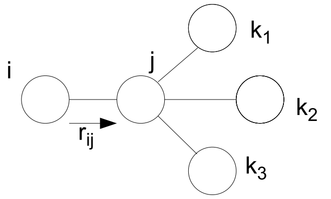

Fig. 2. A graph in which all devices affect the receive probability r_{ij}. In order to improve the performance over this directed link from node i to node j, we must increase the emission probability of node i and decrease the emission probabilities of devices j and k_1, k_2 and k_3.

C. Algorithm Issue: two heuristics for assigning values to P

One question with this approach is how to select appropriate values for P. As a preliminary to this question we can consider the impact of P on the probability of one device communicating with another. We define $G' = (V, R)$ to be a weighted, directed graph in which the edge weights correspond to the probability of device j receiving a message from a neighbouring device i during any given timeslice t; *i.e.* $R = \{r_{ij}\}, \forall i, j \in E$. We can calculate the value of r_{ij} as follows:

$$r_{ij} = p_i(1 - p_j) \prod_{k \in N(j), k \neq i} (1 - p_k), \quad i, j \in E \qquad (1)$$

where $N(x)$ returns the neighbours of x according to G.

If we consider the BPA variant of the probabilistic algorithm described above, we can see that each link $r \in R$ is critical. In fact, for the basic variant, we can write the probability of device i falling silent when all devices $j \in N(i)$ are in the armed state as:

$$\omega_i = 1 - \prod_{j \in N(i)} 1 - (1 - r_{ji})^{\gamma} \quad . \qquad (2)$$

The situation is more subtle in the neighbour table exchange variant of the algorithm; however, we can see that throughput over each link is a desirable property since each link has the potential to reduce the probability of a false positive.

1) The Max Min R Heuristic (MMRH): Since the effectiveness of the alarm system depends on recognizing when a neighbour has stopped transmitting, a reasonable question to consider is how to select maximal values for $P = \{p_i\}, i \in V$ subject to the constraint that we get the best performance over the worst link in the network. We define the worst link r_{min} to be $\min r_{ij}, i, j \in R$.

To find the $\max \min(R)$ we propose a gradient ascent algorithm in which we increase the value of the minimum link at each iteration. To do this we use the partial derivatives of r_{min} with respect to each $p_i \in P$:

$$\nabla r_{min} = \Big(\frac{\partial r_{min}}{\partial p_1}, \frac{\partial r_{min}}{\partial p_2}, \ldots \frac{\partial r_{min}}{\partial p_n}\Big)$$

where $n = |V|$.

From Equation (1), however, it can be observed that only the partials for p_i, p_j and p_k , $k \in N(j), k \neq i$ are non-zero for the gradient ∇r_{min} (Fig. 2). Considering each of these sets of partials in turn gives us the following equations. For the device i initiating the communication over the minimum link r_{ij}:

$$\begin{aligned}\frac{\partial r_{ij}}{\partial p_i} &= \frac{\partial}{\partial p_i} p_i(1-p_j) \prod_{k \in N(j), k \neq i} (1-p_k) \\ &= (1-p_j) \prod_{k \in N(j), k \neq i} (1-p_k) \\ &= \frac{r_{ij}}{p_i} \quad . \end{aligned} \tag{3}$$

For the receiving device j over the minimum link r_{ij}:

$$\begin{aligned}\frac{\partial r_{ij}}{\partial p_j} &= \frac{\partial}{\partial p_j} p_i(1-p_j) \prod_{k \in N(j), k \neq i} (1-p_k) \\ &= -p_i \prod_{k \in N(j), k \neq i} (1-p_k) \\ &= \frac{-r_{ij}}{1-p_j} \quad . \end{aligned} \tag{4}$$

and similarily, for the devices $k \in N(j)$ where $k \neq i$ we have:

$$\begin{aligned}\frac{\partial r_{ij}}{\partial p_k} &= \frac{\partial}{\partial p_k} p_i(1-p_j)(1-p_k) \prod_{q \in N(j), q \neq i, q \neq k} (1-p_q) \\ &= -p_i(1-p_j) \prod_{q \in N(j), q \neq i, q \neq k} (1-p_q) \\ &= \frac{-r_{ij}}{1-p_k} \quad . \end{aligned} \tag{5}$$

The following gradient ascent style algorithm could be used for maximizing the r_{min} value of the network:

```
$P_0$ = initial guess
for $t$ = 1:NumIterations
    calculate $r_{min}$ as a function of $P_{t-1}$ and $G$
    $P_t = P_{t-1} + \alpha(\nabla r_{min})$
end
```

Here, α is an appropriate selected value between 0 and 1 and ∇r_{min} is defined by Equations (3), (4) and (5). The above approach, however, will not improve links that are not minimal during some iteration of the algorithm.

We can alternate the boosting of the minimum link with an update that attempts to equalize the performance of the minimum links affected by each p_i value. This can be done by considering the partial derivative of each p_i with respect to R and constructing an 'equalizing' gradient ∇R_{eq}. As opposed to the gradient with respect to one link, we use the gradient across all links:

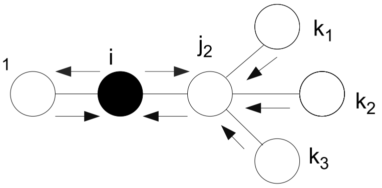

Fig. 3. If p_i is adjusted upwards, then the performance across links (i, j_1) and (i, j_2) will improve, while that of links (j_1, i), (j_2, i), (k_1, j_2), (k_2, j_2) and (k_3, j_2) will decrease.

$$\nabla R_{eq} = \Big(\frac{\partial R_{eq}}{\partial p_1}, \frac{\partial R_{eq}}{\partial p_2}, \ldots \frac{\partial R_{eq}}{\partial p_n}\Big)$$

where $n = |V|$.

Let us consider the partials for all links R with respect to a single p_i:

$$\frac{\partial R}{\partial p_i} = \sum_{i,j \in R} \frac{\partial r_{ij}}{\partial p_i} \quad .$$

The partials for p_i are only non-zero for $r_{ij}, j \in N(i)$ and $r_{ji}, j \in N(i)$ and $r_{kj}, k \in N(j), k \neq i$. This is because adjusting the p value of an individual node i will only affect the weight (or r value) of certain links in the graph $G' = (V, R)$ (see Fig. 3).

In a manner similar to the derivation of Equations (3), (4) and (5), we can write the partial of a single r with respect to p_i. The first type of r with a non-zero partial derivative that we consider are the outbound links from i to j:

$$\frac{\partial r_{ij}}{\partial p_i} = \frac{r_{ij}}{p_i} \quad . \tag{6}$$

For the inbound links from j to i where $j \in N(i)$ we have:

$$\frac{\partial r_{ji}}{\partial p_i} = \frac{-r_{ji}}{1-p_i} \tag{7}$$

and similarily, for the links from k to j where $k \in N(i), k \neq i$ we have:

$$\frac{\partial r_{ki}}{\partial p_i} = \frac{-r_{kj}}{1-p_i} \quad . \tag{8}$$

Indeed, we can categorize the directed links affected by p_i into those with a positive partial derivative:

$$R_{i+} = \{r_{ij}\}, j \in N(i) \tag{9}$$

and those with a negative partial derivative:

$$R_{i-} = \{r_{ji}\}, j \in N(i) \cup \{r_{kj}\}, k \in N(i), k \neq i \quad . \tag{10}$$

To arrive at our equalizing gradient ∇R_{eq} we now consider how to adjust each $p_i \in P$ such that the $min(R_{i+})$ and the $min(R_{i-})$, as defined by Equations (9) and (10), are equal. To do so we first select the overall minimum r value effected by a particular p_i $u_i = \min(R_{i-} \cup R_{i+})$. We then consider the trajectory of each link with a partial of the opposite sign (with respect to p_i); if $u_i \in R_{i+}$ then $\hat{R}_i = R_{i-}$ and otherwise $\hat{R}_i = R_{i+}$. We now compute the set of potential p_i values W at which the trajectory of the minimum link

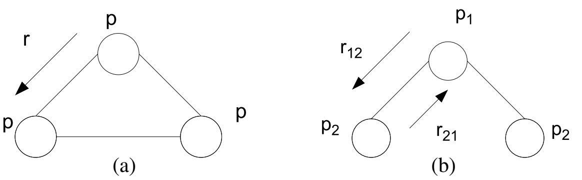

Fig. 4. Two communication topologies where: (a) by symmetry each device has the same p and the values of r are the same across all directed edges and (b) there are two values of p to consider and two potential r values to consider for each directed edge.

u_i and each link with a partial of the opposite sign $r \in \hat{R}_i$ would intersect. We take the minimum of these intersection values as our target value $p'_i = \min(W)$. The component of the gradient ∇Eq for p_i is then equal to $p'_i - p_i$. Our update becomes:

$$P_t = P_{t-1} + \alpha(\nabla r_{min} + \nabla R_{eq}) \quad .$$

We will refer to this gradient-based approach for selecting the maximal values of P such that the largest r_{min} value is obtained as the Max Min R Heuristic (MMRH).

2) The Max Neighbourhood Degree Heuristic: As an alternative to the MMRH approach described in the previous section, we consider the following heuristic for assigning values to P. Let the probability of a node broadcasting during a timeslice be inversely proportional to the max degree of itself and all its neighbours plus one:

$$p_i = \frac{1}{\max \delta(k) + 1}, k \in \{N(i), i\} \quad . \qquad (11)$$

The motivation for this heuristic is to limit the emission rate of each device to that of the most overloaded device in its neighbourhood.

As opposed to the MMRH heuristic, this heuristic can be easily calculated in a distributed manner. The application of this calculation would require each device running either BPA or NTXPA to include in its status message the size of its neighbour table; *i.e.* its degree. This degree would be recorded in a table for each neighbour, allowing the calculation in Equation (11). We refer to this approach as the Max Neighbourhood Degree Heuristic (MNDH). Later, we will show through simulations that this heuristic approach for selecting the values of P performs almost as well as the MMRH.

Example assignment of P values: Let us consider maximizing the performance over the worst link for the two simple topologies shown in Fig. (4).

For the case of a three node ring topology, Equation 1 becomes the following: $r = p(1-p)^2$. By setting the derivative to zero it can be shown that $p = 1/3$ optimizes the best performance over the worst link. This is intuitively natural; each device spends a third of its time emitting and two thirds of its time in receive mode.

For the case of a three node star topology, we have two r values to consider when we substitute p_1 and p_2 into Equation (1): $r_{12} = p_1(1-p_2)$ and $r_{21} = p_2(1-p_1)(1-p_2)$. By setting $r_{12} = r_{21}$, we can take the derivative with respect to p and obtain the values $p_1 = 1 - \frac{\sqrt{2}}{2}$ and $p_2 = \sqrt{2} - 1$.

Running the gradient-based MMRH gives the value reported above as one would expect. The heuristic MDHA gives the value of $p = 1/3$ for all nodes for both the graphs shown in Fig. 4.

D. Algorithm Issue: the assignment of threshold γ

Once the P values have been selected, we must choose a γ value for our network. Obviously there will be a tradeoff between the false positive rate and the latency that occurs when a true event is sensed. One way to assign a γ value is to set the threshold such that the probability of the worst device $i \in V$ suffering a false positive is less than ϵ; *i.e.* select the lowest discrete value for γ such that $\max\{\omega_i\} > \epsilon$ where ω_i is the false positive rate of device i.

Equation (2) gives us the value for ω_i if the basic approach to the Sentinel Problem (BPA) is used. From there we can write:

$$\begin{aligned} 1 - \omega_i &= \prod_{k \in N(i)} 1 - (1 - r_{ki})^{\gamma} \\ \log(1 - \omega_i) &= \sum_{k \in N(i)} \log\left(1 - (1 - r_{ki})^{\gamma}\right) \\ &> \delta(i) \log\left(1 - (1 - r_{min})^{\gamma}\right) \\ &> \delta(i) \frac{-(1 - r_{min})^{\gamma}}{1 - (1 - r_{min})^{\gamma}} \qquad (12) \end{aligned}$$

by using the fact that $r_{ki} > r_{min}, \forall k, i \in E$ by definition of r_{min} and the fact that $\log(1 + x) > x/1 + x$.

By setting $1 - \epsilon$ to the right side of Equation (12) and working through some algebra, we can show that if we select:

$$\gamma >= \left\lceil \frac{\log\left(\frac{\log(1-\epsilon)}{\delta(i)}\right) - \log\left(\frac{\log(1-\epsilon)}{\delta(i)} - 1\right)}{\log(1 - r_{min})} \right\rceil, i \in V \qquad (13)$$

then we can be sure that $\omega_i <= \epsilon, i \in V$.

This calculation is more challenging if the information exchanging (NTXA) variant of the probabilistic algorithm is used and depends on how many neighbours each device has in common with its other neighbours. We will show experimentally, however, that the NTXA appears to do as well or better than the BPA for a particular network topology, suggesting that Inequality (13) provides an upper bound on the γ value required for a given ϵ.

E. Analysis

1) Correctness: It can be seen that the probabilistic algorithm variants are correct given our problem formulation for finite values of γ. Once a single device $i \in V$ enters the triggered state and falls silent, its neighbours will enter the alarmed state in less than γ timeslots, and the alarm state will propagate to all components connected to i in the network.

2) Network False Positive Rate: The probability of a false positive for the probabilistic algorithm variants is greater than zero. Specifically, it depends on: the variant of the algorithm employed (BPA or NTXPA); the manner of selecting P values (MMRH or MNDH); the value assigned to γ; and both the number of devices and communication topology of the network. We will investigate these relationships further through numerical simulations in a later section.

3) Latency: For both probabilistic variants, a worst case latency bound is γD where D is the diameter of the communication graph G.

4) Implementation Potential: Using the Neighbour Table Exchange Probabilistic Algorithm (NTXPA) with the P value assigned in a distributed manner using the Max Neighbourhood Degree Heuristic (MNDH) would be relatively straightforward to implement. One key advantage of this approach is that it does not necessarily require synchronized clocks. Although we have made the synchronized assumption for ease of analysis (and simulation), the concept would still work without it. The same is not true for any approach based on an assigned broadcast schedule.

One challenge is the assignment of an appropriate γ value. In practice, one could envision this value being set prior to the deployment of the network based on rules of thumb that depend on an estimation of the max degree of the communication graph of the network and the number of devices. Equation (13) is a start towards such a heuristic.

In the next section we evaluate the performance of the various probabilistic and deterministic algorithms for solving the Sentinel Problem.

V. Experiments

In order to evaluate solutions to the Sentinel Problem we programmed a simulation of the network model described in Section II using the both the BPA and NTXPA algorithms (see Sections IV-B.1 and IV-B.2 respectively). The simulation takes as input: a network communication graph G, an assignment of broadcast probabilities per timeslice for each device $p_i \in P$; a vector of devices in the triggered state Z; and a maximum simulation length in timeslices T. The output of the simulation is the number of timeslices the simulation runs before the network falls silent; *i.e.* before each device in the network enters the alarmed state (or the triggered state). In the event that the network does not fall silent, the value T is returned.

We then analysed various aspects of our solutions to the Sentinel Problem using a class of graphs we will refer to as *disk graphs*. The graphs are obtained by selecting random points uniformly at random in some bounded region of the plane as the location of the vertices, and assigning an edge between any two vertices if the pair-wise distance between their associated locations is less than the given communication radius. These types of graphs are commonly used as models in sensor network research; see, for example, the work of Gandham *et al.* [14].

A. Assigning Values to P: MMRH vs. MNDH

One interesting observation was that whether values were assigned to P using either the Max Min R Heuristic (MMRH) or the Max Neighbourhood Degree Heuristic (MNDH) the results were similar. We ran a number of trials using the simulator with the T value set to some fixed number of timesteps. The number of trials successfully completed without the network falling silent, (suffering a network false positive), was recorded for different values of γ for both MMRH and MNDH. The results were observed to be similar across disk graphs of various size and edge density. For example, see Fig. 5.

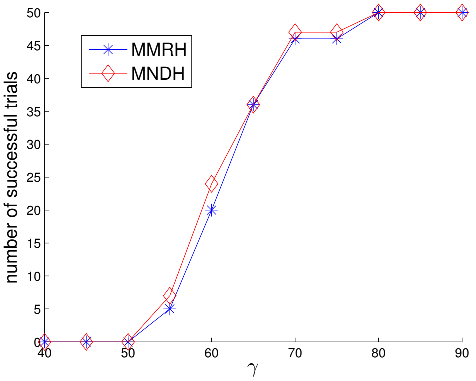

Fig. 5. The number trials out of 50 without a network false positive as a function of the γ value selected for the two heuristics of assigning P using the NTXPA. A successful trial ran for 10000 timesteps. The simulation was run on the graph of Fig. 6b.

A phase transition phenomenon was observed in these experiments as the value assigned to γ was varied. For any specific value of γ, all trials either suffered a network false positive or else all trials completed successfully. Only at a handful of γ values were mixed results seen. Phase transitions are not uncommon in sensor network behaviour; see, *e.g.*, the work of Krishnamachari *et al.* [15]. Pragmatically, this means that for any particular network and a specified false positive free run time T, one requires a γ value safely past the transition point. For example, a value of $\gamma = 100$ would likely be acceptable for the example shown in Fig. 5. For reference, the length of the broadcast schedule $|\Lambda|$ obtained for the graph used in this experiment was 13 using the BSA method described in Section III-B.

B. Evaluation of Deterministic and Probabilistic Approaches

The Neighbour Table Exchange Probabilistic Algorithm (NTXPA) outperformed the Basic Probabilisitic Algorithm (BPA) on simulations using the same graph and P values. For example, the experiment presented in Fig. 7 shows histograms of simulation run times before a network false positive for a number of trials on graphs of three different edge densities. On all trials, it can be seen that the NTXPA algorithm obtains a longer run time on average. The better performance of the NTXPA algorithm is not unexpected since this approach has the opportunity to augment its own neighbour table with information collected by its neighbours. It can also be seen that as the density of the graph increases, the average run time before a false positive for both techniques decreases.

For both the BPA and NTXPA variants of the probabilistic approach, the latency between a device detecting an activity of interest (entering the triggered state) and the network

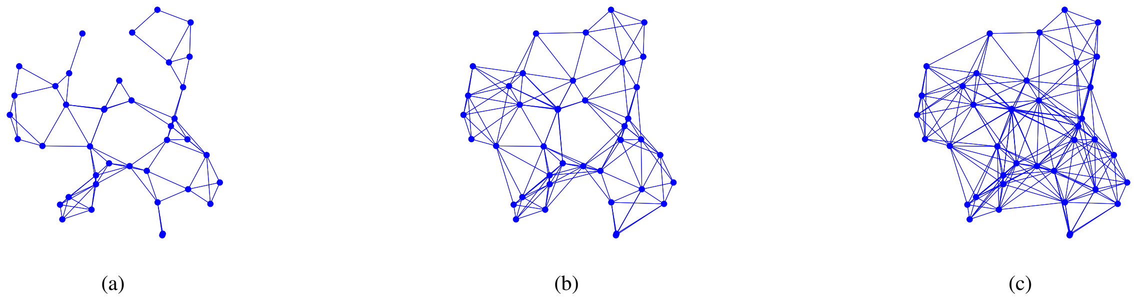

Fig. 6. Connectivity graphs for a 40 node network. Graphs were constructed by selecting 40 points uniformly at random within a radius of 1 unit from the origin. Points within (a) 0.4 units, (b) 0.5 units, and (c) 0.6 units were connected. The graphs have: (a) 87 edges; (b) 142 edges; and (c) 212 edges. (The length of broadcast schedule $M = |\Lambda|$ obtained using the BSA method described in Section III-B is: (a) 9; (b) 13; and (c) 19.)

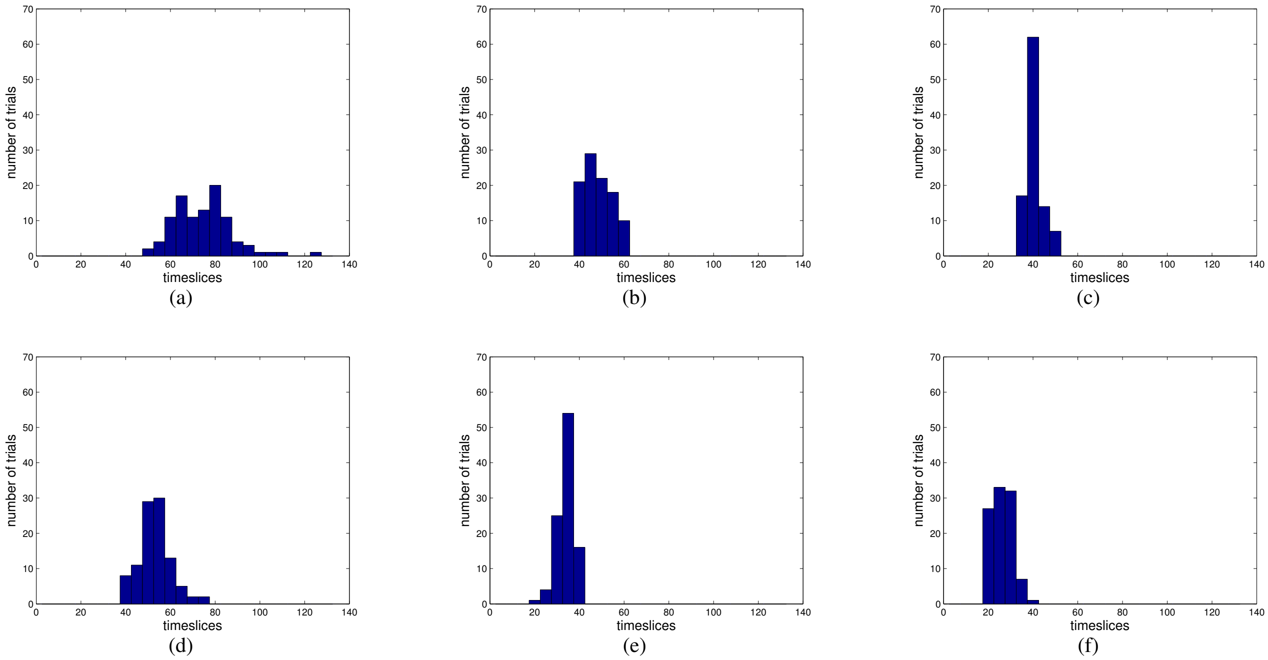

Fig. 7. Histograms showing the distribution of run times in timeslices before a network false positive using value of $\gamma = 20$. Values for P were obtained using the MNDH. Each plot shows the results of 100 trials. Simulation results shown in (a), (b) and (c) were obtained by using the NTXPA (see Section IV-B.1) on the graphs of Fig. 6a, Fig. 6b, and Fig. 6c respectively. Simulation results shown in (d), (e) and (f) were obtained by using the BPA (see Section IV-B.1) on the graphs of Fig. 6a, Fig. 6b and Fig. 6c respectively.

falling silent is determined by the topology of the network graph and the value assigned to γ. Fig. 8 shows the result of an experiment examining this issue. Note that latency of up to 500 timeslices was observed in this experiment. This can be contrasted to an upper limit of 104 timeslices if the BSA was used (the diameter of the graph used in the experiment was eight).

VI. Discussion and Future Work

In this paper we have presented a graph-based formulation of the *Sentinel Problem* together with a number of potential approaches for solving the problem. We show that the problem can be solved using known broadcast scheduling techniques, but the application will require solving complex implementation issues. We presented probabilistic alternatives that would be easier to implement in practice, although it appears the probabilistic algorithms have the disadvantage of either suffering from occasional false positives, or long latency times.

Our analysis and experiments have relied on several common assumptions regarding RF communications in sensor networks such as a fixed communication radius. Such assumptions are common in sensor network research, but are not necessarily valid. See Kotz *et al.* [16] for results on the experimental validation of common wireless simulation assumptions. Specifically, there are several key areas of possible improvement for our network communication model. Below we list three areas:

1) *Probability of reception over a communication link:* Our model assumes a link success probability of either 0 or 1. More accurate would be to allow the probability of occasional failure even over good links.
2) *Range edge effects:* We assume that a device has

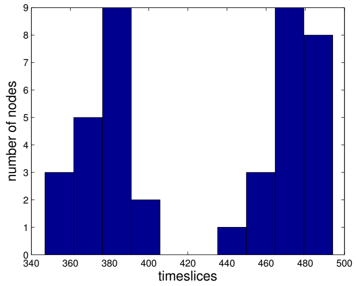

Fig. 8. Histogram of the number of timeslices between setting a single device to the triggered state and the network falling silent for a number of trials. One trial was conducted with each device as the trigger. The simulation was run using $\gamma = 100$ on the graph shown in Fig. 6b with NTXPA and P values assigned using the MNDH. For each trial, the simulation was first run for 1000 timesteps to 'burn-in' before the triggering device was set to the triggered state.

a known set of neighbours; however, depending on deployment details, there will typically be a large number of devices that are near the edge of their range and for which communication is intermittent. A possible approach would be to threshold on received signal strength indication (RSSI), but even this approach could result in some devices that move above or below the threshold due to dynamic radio frequency conditions.

3) *More accurate model of congestion:*
We assume that any two simultaneously transmitting neighbours will interfere with each other. The result depends, however, on the relative power of the signals at the receiving point. For example, if the signal of a near neighbour is several tens of decibels more powerful than the transmission of a simultaneously transmitting distant neighbour, is likely that the device will be able to receive the nearer neighbour's transmission.

The improvements listed above would allow a more thorough assessment of our approach and would be a prudent preliminary step before addressing implementation on a hardware platform.

Additionally, although we have introduced probabilistic alternatives to broadcast scheduling in the context of solving the Sentinel Problem, it is possible that the approach might have a more general application as an easily distributed and more easily implemented alternative to time division multiple access (TDMA). A variant of this type of approach might be appropriate for data acquisition applications where the devices obtain power from the grid and latency is not as important as ease of deployment, adaptability, and the efficient use the resources available on the sensor platform; *i.e.* memory, computational power, and code space.

Finally, multi-channel versions of the Sentinel Problem would be interesting to consider and would tie in with work such as that by Giannoulis *et al.* [17].

VII. Conclusions

In this paper we presented a multi-hop sensor network alarm application that we call the *Sentinel Problem*. We showed that the problem can be solved using known broadcast scheduling techniques and suggested some probabilistic alternatives. Through simulations and analysis we compared the performance and discussed the merits of the various approaches. Aspects of our probabilistic approach show promise but require further assessment using more realistic network models.

References

[1] D. Meger, D. Marinakis, I. Rekleitis, and G. Dudek, "Inferring a probability distribution function for the pose of a sensor network using a mobile robot," in *Proc. of ICRA*, Kobe, Japan, May 2009.

[2] A. Rahman, M. Hoque, F. Rahman, S. K. Kundu, and P. Gburzynski, "Enhanced partial dominant pruning EPDP based broadcasting in ad hoc wireless networks." *Journal of Networks*, vol. 4, no. 9, pp. 895–904, 2009.

[3] X. Chen, X. Hu, and J. Zhu, "Data gathering schedule for minimal aggregation time in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 5, no. 4, 2009.

[4] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "Macaw: A media access protocol for wireless lans," in *ACM SIGCOMM*, 1994.

[5] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, July-Aug. 2004.

[6] I. Chlamtac and S. Kutten, "A spatial reuse TDMA/FDMA for mobile multi-hop radio networks," in *IEEE Proc. of INFOCOM*, March 1985.

[7] R. Ramaswami and K. Parhi, "Distributed scheduling of broadcasts in a radio network," in *IEEE Proc. of INFOCOM*, vol. 2, April 1989, pp. 497–504.

[8] S. Ramanathan and E. L. Lloyd, "Scheduling algorithms for multihop radio networks," *IEEE/ACM Transactions on Networking*, vol. 1, no. 2, pp. 166–177, April 1993.

[9] K. Sayrafian-Pour and A. Ephremides, "Interference-free time-frequency broadcast scheduling in multihop packet radio networks," in *Proc. of IEEE Wireless Communications and Networking Conference, (WCNC)*, vol. 1, 2000, pp. 106 – 111.

[10] T. A. ElBatt and A. Ephremides, "Joint scheduling and power control for wireless ad hoc networks," *IEEE Trans. on Wireless Communications*, vol. 3, no. 1, pp. 74–85, January 2004.

[11] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," in *Proc. of IEEE Wireless Communications and Networking Conference, (WCNC 2003)*, 2003.

[12] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *Special issue of IEEE Trans. on Information Theory and IEEE ACM Trans. on Networking*, vol. 52, no. 6, pp. 2508–2530, June 2006.

[13] S.-J. Tang, X. Wu, X. Mao, Y. Wu, P. Xu, G. Chen, and X.-Y. Li, "Low complexity stable link scheduling for maximizing throughput in wireless networks," in *Proc. of IEEE Com. Society Conf. on Sensor, Mesh and Ad Hoc Communications and Networks, SECON'09*, June 2009, pp. 1 – 9.

[14] S. Gandham, E. Dawande, and R. Prakash, "Link scheduling in sensor networks: Distributed edge coloring revisited," in *IEEE Proc. of INFOCOM*, vol. 4, Miami, March 2005, pp. 2492–2501.

[15] B. Krishnamachari, S. Wicker, and R. Bejar, "Phase transition phenomena in wireless ad hoc networks," in *IEEE Proc. of Global Telecommunications Conference, GLOBECOM '01*, vol. 5, San Antonio, 2001, pp. 2921–2925.

[16] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," in *Proc. of the ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*. ACM Press, October 2004, pp. 78–82.

[17] A. Giannoulis, T. Salonidis, and E. Knightly, "Congestion control and channel assignment in multi-radio wireless mesh networks," in *IEEE Com. Society Conf. on Sensor, Mesh and Ad Hoc Communications and Network (SECON)*, San Francisco, CA, June 2008, pp. 350–358.