Optimal Sequential Wireless Relay Placement on a Random Lattice Path

Abhishek Sinha[a,1], Arpan Chattopadhyay[b], K.P. Naveen[b], Prasenjit Mondal[c], Marceau Coupechoux[d,2], Anurag Kumar[b]

[a] *Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139*
[b] *Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560012, India*
[c] *Cisco Networks, Bangalore, India*
[d] *Telecom ParisTech and CNRS LTCI, Dept. of Informatique et Réseaux, Paris, France*

Abstract

Our work is motivated by impromptu (or "as-you-go") deployment of wireless relay nodes along a path, a need that arises in many situations. In this paper, the path is modeled as starting at the origin (where there is the data sink, e.g., the control centre), and evolving randomly over a lattice in the positive quadrant. A person walks along the path deploying relay nodes as he goes. At each step, the path can, randomly, either continue in the same direction or take a turn, or come to an end, at which point a data source (e.g., a sensor) has to be placed, that will send packets to the data sink. A decision has to be made at each step whether or not to place a wireless relay node. Assuming that the packet generation rate by the source is very low, and simple link-by-link scheduling, we consider the problem of sequential relay placement so as to minimize the expectation of an end-to-end cost metric (a linear combination of the sum of convex hop costs and the number of relays placed). This impromptu relay placement problem is formulated as a total cost Markov decision process. First, we derive the optimal policy in terms of an optimal placement set and

[☆] The research reported in this paper is an extension of the material published in the conference paper [1].

[☆☆] This work was supported by (i) the Department of Science and Technology (India) via the Indo-Brazil Project WINSON, (ii) the Department of Electronics and Information Technology (India) and the National Science Foundation (USA) via an Indo-US collaborative project titled Wireless Sensor Networks for Protecting Wildlife and Humans, and (iii) the J.C. Bose National Fellowship (of the Govt. of India).

Email addresses: `sinhaa@mit.edu` (Abhishek Sinha), `arpan@ece.iisc.ernet.in` (Arpan Chattopadhyay), `kpnaveen@ece.iisc.ernet.in` (K.P. Naveen), `prasenjuetce@gmail.com` (Prasenjit Mondal), `marceau.coupechoux@telecom-paristech.fr` (Marceau Coupechoux), `anurag@ece.iisc.ernet.in` (Anurag Kumar)

[1] Corresponding author. This work was done while he was a Master of Engineering student in the ECE Department, Indian Institute of Science, Bangalore.

[2] This work was done while he was a Visiting Scientist in the ECE Department, Indian Institute of Science, Bangalore.

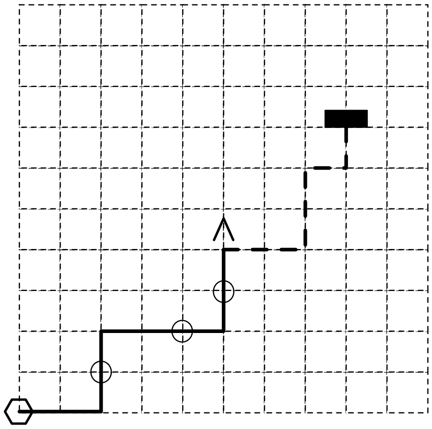


Figure 1: A wireless network being deployed as a person steps along a random lattice path. Inverted **V**: location of the deployment person; path drawn with a solid line: path already covered; circles: deployed relays; path drawn with a thick dashed line: a possible evolution of the remaining path. The source to be placed at the end is also shown as the black rectangle.

show that this set is characterized by a boundary (with respect to the position of the last placed relay) beyond which it is optimal to place the next relay. Next, based on a simpler one-step-look-ahead characterization of the optimal policy, we propose an algorithm which is proved to converge to the optimal placement set in a finite number of steps and which is faster than value iteration. We show by simulations that the distance threshold based heuristic, usually assumed in the literature, is close to the optimal, provided that the threshold distance is carefully chosen.

Keywords: relay placement, as-you-go deployment of wireless sensor networks, optimal stopping problems, placement boundary

1. Introduction

Wireless networks, such as cellular networks or multihop ad hoc networks, would normally be deployed via a planning and design process. There are situations, however, that require the impromptu (or "as-you-go") deployment of a multihop wireless packet network. Such an impromptu approach would be required to deploy a wireless sensor network for situational awareness in emergency situations such as those faced by firemen or commandos ([2, 3]). For example, as they attack a fire in a building, firemen might wish to place temperature sensors on fire-doors to monitor the spread of fire, and ensure a route for their own retreat; or commandos attempting to flush out terrorists might wish to place acoustic or passive infra-red sensors to monitor the movement of people in the building. As-you-go deployment may also be of interest when deploying a multi-hop wireless sensor network over a large terrain (such as a dense forest) in order to obtain a first-cut deployment which could then be augmented to a network with desired properties (connectivity and quality-of-service). Such quick deployment of a wireless relay network also becomes necessary if the deployment needs to be stealthy (for example, for detecting poachers or fugitives in a forest), or if the network has to be redeployed at short time intervals to track an evolving phenomenon.

With the above larger motivation in mind, in this paper we are concerned with the rigorous formulation and solution of a problem of impromptu deployment of a multihop wireless network along a random lattice path, see Fig. 1, which, while being a simple model for providing insights into a larger class of problems, could also be a reasonable model for a forest trail on flat ground. The objective is to create a multihop wireless path for packet communication from the end of the path to its beginning. The problem is formulated as an optimal sequential decision problem. The formulation gives rise to a total cost Markov decision process, which we study in detail in order to derive structural properties of the optimal policy. We also provide an efficient algorithm for computing the optimal policy.

1.1. Related Work

"As-you-go" deployment of wireless relay networks has, in the past, been motivated by "first responder" networks, a concept that has been around at least since 2001. In [3], Howard et al. provide heuristic algorithms for the problem of incremental deployment of sensors (such as surveillance cameras) with the objective of covering the deployment area. Their problem is related to that of self-deployment of autonomous robot teams and to the art-gallery problem. Creation of a communication network that is optimal in some sense is not an objective in [3]. In a somewhat similar vein, the work of Loukas et al. [4] is concerned with the dynamic locationing of robots that, in an emergency situation, can serve as wireless relays between the infrastructure and human-carried wireless devices. The problem of impromptu deployment of static wireless networks has been considered in [5, 6, 7, 8, 9]. In [5], Naudts et al. provide a methodology in which, after a node is deployed, the next node to be deployed is turned on and begins to measure the signal strength to the last deployed node. When the signal strength drops below a predetermined level, the next node is deployed and so on. Souryal et al. provide a similar approach in [6, 8], where an extensive study of indoor RF link quality variation is provided, and a system is developed and demonstrated. The work reported in [9] is yet another example of the same approach for relay deployment. More recently, Liu et al. [10] describe a "breadcrumbs" system for aiding firefighters inside buildings, and is similar to our present paper in terms of the class of problems it addresses. In a survey article [2], Fischer et al. describe various localization technologies for assisting emergency responders, thus further motivating the class of problems we consider. Bao and Lee [7] consider the problem of multiple persons, each carrying some relays, exploring an unknown region, and collaboratively placing relays to stay connected to a command center. The objective is to maximize the area they can explore while staying connected, using these relays. They propose a heuristic algorithm based on measurements between the deployed relays and between the mobile individuals.

In the literature referred to above, heuristic algorithms are proposed for relay placement. In our earlier work (Mondal et al. [1]) we took the first steps towards rigorously formulating and addressing the problem of impromptu optimal deployment of a multihop wireless network along a line at the end of which a source has to placed. The source (e.g., a sensor) placement location is discovered only as the network is deployed. A probabilistic model is used for the unknown location of the source along the line. Once placed, the sensor sends periodic measurement packets to a control centre near the start of the line. It is assumed that the measurement rate at the sensor is low, so that (with a very high probability) a packet is delivered to the control centre before the next packet is generated at the sensor. This, so called, "lone packet model" is realistic for situations in which the sensor makes a measurement every few seconds, or where the main purpose of the sensor network is to detect sporadic events, and communicate the detection to the sink.

The objective of the sequential decision problem is to minimise a certain expected per packet cost (e.g., end-to-end delay or total energy expended by a node), which can be expressed as the sum of the costs over each hop, subject to a constraint on the number of relays used for the operation. It has been proved in [1] that an optimal placement policy solving the above mentioned problem is a threshold rule, i.e., there is a threshold r^* such that, after placing a relay, if the operative has walked r^* steps without the path ending, then a relay must be placed at r^*.

1.2. Outline and Our Contributions

In this paper, we adopt the following model features from [1]: (**a**) a single operative moves step-by-step along a path, deciding to place or to not place a relay; (**b**) the length of the path is a geometrically distributed random multiple of the step size; (**c**) a source of packets is placed at the end of the path; (**d**) the lone packet traffic model applies; (**e**) the total cost of a deployment is a linear combination of the sum of convex hop costs and the number of nodes placed. We, however, extend the work presented in [1] to the two-dimensional case. The path evolves from a sink at the origin, over a lattice in the positive quadrant. The spacing between the lattice points is equal to the step size of the person deploying the network. The path evolves by stepping over the lattice in the +x, or the +y, direction. This type of movement is common for forest trails on flat ground, where the trail exhibits a steady drift towards some direction in the positive quadrant. While the expected path length is fixed by the parameter of the geometrically distributed number of steps, the "twistiness" (or the frequency of turns) in the path is parameterised by a probability of the path taking a turn at each step. We note that a similar mobility model has been used in the simulation experiments used for evaluating the heuristics reported in [7]. A radio link exists between successive nodes placed anywhere on the path (see Fig. 2), but the quality of the link would depend on the distance between

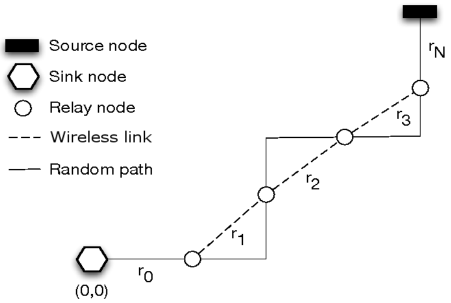


Figure 2: A depiction of relay deployment along a random lattice path.

the nodes. The lone packet model is a natural first assumption, and would be useful in low-duty cycle monitoring applications. Once the network has been deployed, an analytical technique such as that presented in [11] can be used to estimate the actual packet carrying capacity of the network.

We will formally describe our system model and problem formulation in Section 2. The following are our main contributions:

- We formulate the problem as a total cost Markov decision process (MDP), and characterize the optimal policies in terms of placement sets. We show that these optimal policies are threshold policies and thus the placement sets are characterized by boundaries in the two-dimensional lattice (Section 3). Beyond these boundaries, it is optimal to place a relay.

- Noticing that placement instants are renewal points in the random process, we recognize and prove the One-Step-Look-Ahead (OSLA) characterization of the placement sets (Section 4).

- Based on the OSLA characterization, we propose an iterative algorithm, which converges to the optimal placement set in a finite number of steps (Section 5). We have observed that this algorithm converges much faster than value iteration.

- In Section 7 we provide several numerical results that illustrate the theoretical development. The relay placement approach proposed in [5, 6, 8, 9] would suggest a distance threshold based placement rule. We numerically obtain the optimal rule in this class, and find that the cost of this policy is numerically indistinguishable from that of the overall optimal policy provided by our theoretical development. This suggests that it might suffice to utilize a distance threshold policy. However, the distance threshold should be carefully designed taking into account the system parameters and the optimality objective.

For the ease of presentation we have moved most of the proofs to the Appendix.

2. System Model

We consider a deployment person, whose stride length is 1 unit, moving along a random path in the two-dimensional lattice, placing relays at some of the lattice points of the path and finally a source node at the end of the path. Once placed, the source node periodically generates measurement packets which are forwarded by the successive relays in a multihop fashion to the control centre located at $(0, 0)$; see Fig. 2.

2.1. Random Lattice Path

Let $\mathbb{Z}_+$ denote the set of nonnegative integers, and $\mathbb{Z}_+^2$ the nonnegative orthant of the two dimensional integer lattice. Starting from $(0, 0)$ there is a lattice path that takes random turns in the +x direction, or in the +y direction (this is to avoid the path folding back onto itself, see Fig 2). Under this restriction, the path evolves as a stochastic process over $\mathbb{Z}_+^2$. When the deployment person has reached some lattice point, the path continues for one more step and terminates with probability p, or does not terminate with probability $1 - p$. In either case, the next step is in the +x direction with probability q, and in the +y direction with probability $1 - q$. Thus, for instance, $(1 - p)q$ is the probability that the path proceeds in the +x direction without ending. The person deploying the relays is assumed to keep a count of m and n, the number of steps taken in the x direction and in y direction, respectively, since the previous relay was placed. He is also assumed to know the probabilities p and q. Thus the parameter p governs the path length. If p is small then long paths will be sampled, whereas if p is large then short paths will be sampled. In either case the path length is finite w.p. 1. See Fig. 3 for an illustration of typical sample path evolutions for a given path length (in number of steps), for different turn probabilities q.

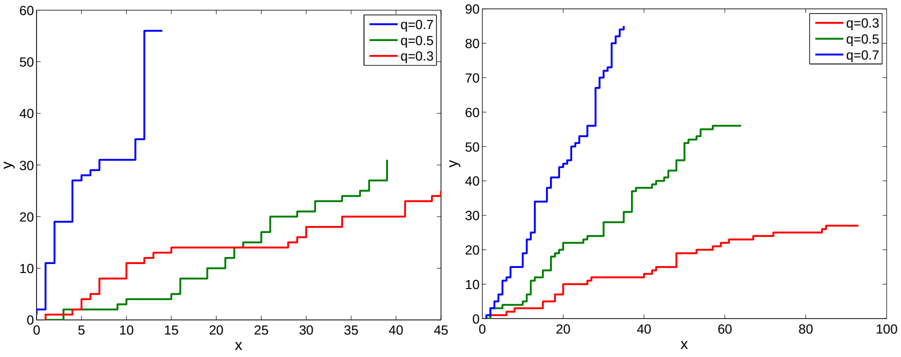


Figure 3: Sample path evolutions for 70-pitch and 120-pitch long paths with different turn probabilities. x and y index the lattice coordinates.

As mentioned earlier, the above structure, while being a simple setting for obtaining insights into the general problem of as-you-go deployment, could reasonably model a winding forest trail on flat ground. If the step sizes are small, random movement on the integral lattice is a good practical approximation to a situation such as traversing along a continuous two-dimensional random trail in a forest. For example, while deploying relays along a forest trail, the lattice pitch could be as little at 3 to 5 meters, while the deployment area could be a few millions of square meters. Thus a path along a fine-pitch lattice can adequately approximate a trail. The details of the model can be motivated as follows. The length, say, L, of the path is a priori unknown, but there is prior information (e.g., the mean distance, $\overline{L}$, along the path from the source to the sink) that, given the stride length δ, leads us to model L as a geometrically distributed number of steps.[3] The stride length δ and the mean length $\overline{L}$ can be used to obtain the parameter of the geometric distribution, i.e., the probability p that the source has to be placed at the next step.

As a practical example, the step size (the distance between points at which placement decisions are taken) can be as small as, say, 5 meters. If the expected distance to the sensor is, say, 500 meters, i.e., 100 steps, then p would be chosen to be 0.01. With this value of p, $\text{Prob}(L > 3000\text{m}) = (1-p)^{600} \approx 0.0024$. Hence, if the forest under consideration has dimensions of a few kilometers, then these forest dimensions are essentially “infinite” compared to the length of the path that is being modeled. The probability q models the “twistiness” of the path, and can also be based on prior information. If the path is known to follow a straight line, then q can be taken to be 0 (the path is a straight line proceeding in the $+\mathsf{x}$ direction), or, equivalently, 1 (the path is a straight line proceeding $+\mathsf{y}$ direction). On the other hand a path that takes frequent turns would be modelled by q close to 0.5.

We have used location independent path evolution probabilities. We note that the Markov decision setting can be extended by making the turn probabilities state dependent. However it would be hard to obtain prior information for getting such state dependent path evolution parameters. Our model uses the minimal information of expected path length, and provides a simple and computable stationary deployment policy.

The sequential decision making formulation, later in this paper, assumes the knowledge of p and q. In many situations (e.g. for deployment along a forest trail) these parameters might be estimated to some accuracy (e.g., from a map). In this paper we provide general structural results for the optimal policy for any p and q, and investigate how well a simple policy performs as compared to the optimal policy.

2.2. Traffic Model

In this paper, we assume that the source (i.e., the sensor placed at the end of the path) generates packets at a rate so low that one packet exits to the

[3] One justification for the use of the geometric distribution, given the prior knowledge, $\overline{L}$, is that it is the maximum entropy discrete probability mass function with the given mean. Thus, by using the geometric distribution, we are leaving the length of the line as uncertain as we can, given the prior knowledge of its mean.

sink before the next packet is generated. Such a, so called, “lone packet” model would be appropriate for situations in which the sensor makes low duty cycle environment measurements (a measurement every few seconds), or generates an occasional alarm packet.
As practical examples, applications such as forest fire detection, intrusion detection, animal surveillance, etc., while sensing would be performed continuously, there will be packet transmission activities only when some activity is detected. Also, there are many wireless sensor applications where continuous sensing is required but the measurement period is of the order of a few seconds (e.g., soil moisture measurement, and ambient temperature measurement). In typical deployments (using, for example, IEEE 802.15.4 radios, and 4-6 hops), the total transmission delay (sum of all one-hop delays) incurred by a packet would be only of the order of milliseconds. This essentially means that consecutive packets are unlikely to interfere with each other in time, thus justifying our lone-packet assumption. A design based on such a lone packet model could also be the starting point for a design for higher packet rates.

With such low packet arrival rates, each packet traverses the network without encountering interference from any other packet. Hence, the delay on any link, or the power required for a given link quality, depend on the path loss characteristics of that link alone.

2.3. Cost Definition

In our model, packet transmission can take place between any two successive relays even if they are not on the same straight line segment of the lattice path. Such a model is suitable, for example, when the deployment region is a thickly wooded forest where the deployment person is restricted to move only along some narrow path. The cost on each link could be the expected delay for delivering a packet across the link (taking into account the medium access and retransmission delays), or the power required to obtain a certain communication quality over the link. For example, the quality of a link can be specified in terms of the probability of the received power on the link falling below a certain minimum received power (i.e., the outage probability of the link) below which the packet error rate (PER) exceeds a desired level (e.g., a -88dBm received power, with receiver noise, could yield a PER of 3% for the packets being carried by the network; see [12]). The outage probability can be lower bounded by ensuring that the average received power across the link (averaged over shadowing and fading) meets a received power target. Such a target average received power would be obtained by deriving a margin above the minimum value of received power (e.g., the -88dBm above), from statistics of shadowing and fading. Of course, this would be a conservative approach compared to one where only a fade margin is applied and shadowing is accounted for by making measurements (see [8] for a discussion of issues related to link quality measurement). A measurement-based approach would require additional state to be maintained in the decision formulation, and is a topic of our ongoing research. We note, however, that measurements would take time at each step, and, thus, might not be feasible if the deployment has to be carried out very quickly.

It is then easily seen that for two successive relays separated by a distance r, the cost (expected delay or average power) would be a function $d(r)$. A formula for expected delay, under the IEEE 802.15.4 MAC has been derived in [13]. In our numerical work we use the power cost, $d(r) = P_m + \gamma r^\eta$, where P_m is the minimum power required, γ relates to an SNR (Signal-to-Noise Ratio) constraint, and η is the path-loss exponent. Now suppose N relays are placed such that the successive inter-relay distances are $r_0, r_1, \cdots, r_N$ (r_0 is the distance from the control centre at $(0,0)$ and the first relay, and r_N is the distance from the last relay to the source placed at the end of the path) then, under the lone packet model, we take the total cost of this placement as the sum of the one-hop costs $C = \sum_{i=0}^{N} d(r_i)$. Note that we require that all deployed relays are used in the path from the source to the sink [4].

Justification for sum of hop costs. There are two ways in which the network could be operated: (i) the nodes are always awake, or (ii) the nodes sleep-wake cycle.

(i) It is a well known fact that the current drawn by an awake radio, waiting to receive a packet, is almost as large as the current required to transmit a packet (e.g, see [15, Page 13]). A network with continuously awake nodes will have a short life span (two or three days, depending on the battery size), and will be employed for transient applications where the network is deployed and removed within a few 10s of hours. During the operational period, an objective could be to minimize the mean end-to-end packet delay. If the network cost is end-to-end delay, then, under the lone-packet model, the end-to-end mean delay is just the sum of the expected hop delays.

(ii) If the network has to last for a long time (several months) then the nodes must sleep-wake cycle, packet delays will be large, and the primary criterion of network cost would be in terms of life-time. When nodes are sleeping, then a node with a packet to transmit (the "custodian" node) requires an awake downstream relay in order to forward the packet. If nodes are equipped with *wake-on* radios then the custodian node sends a low power "wake" signal to the sleeping relay followed immediately by the packet [16][17]. Alternatively, the clocks of the neighbouring nodes could be synchronized, so that the custodian node knows when to transmit its packet so as to "catch" the downstream node just when it wakes up [18]. If the node battery energy is E joules, the transmission power required is $d(r)$, the receiver power required is P_{rcv}, the packet transmission time is t_{pkt}[5], then the life-time of a relay whose next hop node is a

[4] We have investigated the case where relay skipping is allowed in our work [14] under a measurement based setting. *After the relays are placed, the path using all the relays from the source to the sink is not necessarily the shortest path. Thus, the agent could have kept this in mind when deploying the relays in the first place. However, the complexity of the formulation and its solution is significantly more, since, at each decision point, the deployment agent needs to keep some information pertaining to all relays deployed up to any point.*

[5] Assuming a constant packet size, and a constant physical layer bit rate (the IEEE 802.15.4

distance r away is given by $T_i := \frac{E}{A(d(r)+P_{\mathsf{rcv}})t_{\mathsf{pkt}}}$, where A is the rate at which the source generates packets. It follows that if the network objective is to minimize the rate of replacing node batteries (i.e., $\sum_{0\leq i\leq N} \frac{1}{T_i}$) then the criterion should be $\min \sum_{0\leq i\leq N} d(r_i)$, whereas if the objective is to maximize the life-time then the criterion should be $\min \max_{0\leq i\leq N} d(r_i)$. However note that

$$\max_{i=1}^{N} d(r_i) = \lim_{\alpha\to\infty} (\sum_{i=1}^{N} d(r_i)^{\alpha})^{\frac{1}{\alpha}} \tag{1}$$

Hence if we solve the *min-sum* problem with a new cost-function $f(r) = d(r)^{\alpha}$, for some fixed, yet large enough α, we have a good approximate solution for the *min-max* problem via the solution of the *min-sum* problem. It is easy to verify that the function $f(r)$ also satisfies the required technical conditions. This is quite a standard technique for solving min-max problems via a min-sum relaxation and a set of useful references for this method is [19] and the references cited therein.

We now impose a few technical conditions on the one-hop cost function $d(\cdot)$: (**C1**) $d(0) > 0$, (**C2**) $d(r)$ is convex and increasing in r, and (**C3**) for any r and $\delta > 0$ the difference $d(r+\delta) - d(r)$ increases to ∞ as the argument $r \to \infty$.

(**C1**) is imposed considering the fact that it requires a non-zero amount of delay or power for transmitting a packet between two nodes, however close they may be. (**C2**) and (**C3**) are properties we require to establish our results on the optimal policies. They are satisfied by the power cost, $P_m + \gamma r^{\eta}$, and also by the mean hop delay (see [13]).

We will overload the notation $d(\cdot)$ by denoting the one-hop cost between the locations $(0,0)$ and $(x,y) \in \Re^2$ as simply $d(x,y)$ instead of $d(||(x,y)-(0,0)||)$. Using the conditions on $d(r)$ we prove the following convexity result of $d(x,y)$.

Lemma 1. *The function $d(x,y)$ is convex in (x,y), where $(x,y) \in \mathbb{R}^2$.*

Proof. Since $d(\cdot)$ is convex, non-decreasing in its argument, the proof follows by invoking the composition rule [20, Section 3.2.4]. □

We further impose the following condition on $d(x,y)$ where $(x,y) \in \Re^2$. We allow a general cost-function $d(x,y)$ endowed with the following property: (**C4**) The function $d(x,y)$ is positive, twice continuously partially differentiable in variables x and y and $\forall x, y \in \mathbb{R}_+$,

$$d_{xx}(x,y) > 0, \;\; d_{xy}(x,y) > 0, \;\; d_{yy}(x,y) > 0, \tag{2}$$

where $d_{xy}(x,y) = \frac{\partial^2 d(x,y)}{\partial x \partial y}$. These properties also hold for the mean delay and the power functions mentioned earlier.

Finally define, for $(m,n) \in \mathbb{Z}_+^2$, $\Delta_1(m,n) = d(m+1,n) - d(m,n)$ and $\Delta_2(m,n) = d(m,n+1) - d(m,n)$.

physical layer has just one bit rate).

Lemma 2. *$\Delta_1(m,n)$ and $\Delta_2(m,n)$ are non-decreasing in both the coordinates m and n.*

Proof. See Appendix A. □

2.4. Deployment Policies and Problem Formulation

A deployment policy π is a sequence of mappings $(\mu_k : k \geq 0)$, where at the k-th step of the path (provided that the path has not ended thus far) μ_k allows the deployment person to decide whether to *place* or *not to place* a relay where, in general, randomization over these two actions is allowed. The decision is based on the entire information available to the deployment person at the k-th step, namely the set of vertices traced by the path and the location of the previous vertices where relays were placed. Let Π represent the set of all policies. For a given policy $\pi \in \Pi$, let $\mathbb{E}_\pi$ represent the expectation operator under policy π. Let C denote the total hop cost (as defined earlier) and N (a random variable) the total number of relays used. We are interested in solving the following problem,

$$\min_{\pi\in\Pi} \quad \mathbb{E}_\pi C + \lambda \mathbb{E}_\pi N, \tag{3}$$

where $\lambda > 0$ may be interpreted as the cost of a relay. Solving the problem in (3) can also help us solve the following constrained problem,

$$\begin{aligned} \min_{\pi\in\Pi} & \qquad \mathbb{E}_\pi C \\ \text{Subject to:} & \quad \mathbb{E}_\pi N \leq \rho_{avg}, \end{aligned} \tag{4}$$

where $\rho_{avg} > 0$ is a constraint on the expected number of relays (we will describe this procedure in Section 6; for details see [21]).

Remark: A constraint on the expected number of relays would be applicable to a situation in which relays are deployed along multiple paths to connect a sink and a source at the ends of each path. Because of the randomness in the lengths of the paths, we will place more relays along some paths and less along others, with the total number of relays deployed being governed by the constraint on the mean number of relays. For deployment along a line, a constraint on the actual number of relays was dealt with in [13], and leads to a nonstationary deployment policy.

In Sections 3 to 5, we work towards obtaining an efficient solution to the problem in (3).

3. MDP Formulation and Solution

In this section we formulate the problem in (3) as a total cost infinite horizon MDP and derive the optimal policy in terms of optimal placement set. We show that this set is characterized by a two-dimensional boundary, upon crossing which it is optimal to place a relay.

3.1. States, Actions, State-Transitions and Cost Structure

We formulate the problem as a sequential decision process starting at the origin of the lattice path. The decision to place or not place a relay at the k-th step is based on $((M_k, N_k), Z_k)$, where (M_k, N_k) denotes the coordinates of the deployment person with respect to the previous relay and $Z_k \in \{\mathsf{e}, \mathsf{c}\}$; $Z_k = \mathsf{e}$ means that at step k the random lattice path has ended and $Z_k = \mathsf{c}$ means that the path will continue in the same direction for at least one more step. Thus, the state space is given by:

$$\mathcal{S} = \Big\{(m, n, z) : (m, n) \in \mathbb{Z}_+^2, z \in \{\mathsf{e}, \mathsf{c}\}\Big\} \cup \{\phi\}, \tag{5}$$

where ϕ denotes the cost-free terminal state, i.e., the state after the end of the path has been discovered. The action taken at step k is denoted $U_k \in \{0, 1\}$, where $U_k = 1$ is the action to place a relay, and $U_k = 0$ is the action of not placing a relay. When the state is (m, n, c) and when action u is taken, the transition probabilities are given by:

- If u is 0 then,
  (i) $(m, n, \mathsf{c}) \longrightarrow (m + 1, n, \mathsf{c})$ w.p. $(1 - p)q$
  (ii) $(m, n, \mathsf{c}) \longrightarrow (m + 1, n, \mathsf{e})$ w.p. pq
  (iii) $(m, n, \mathsf{c}) \longrightarrow (m, n + 1, \mathsf{c})$ w.p. $(1 - p)(1 - q)$
  (iv) $(m, n, \mathsf{c}) \longrightarrow (m, n + 1, \mathsf{e})$ w.p. $p(1 - q)$.
- If u is 1 then
  (i) $(m, n, \mathsf{c}) \longrightarrow (1, 0, \mathsf{c})$ w.p. $(1 - p)q$
  (ii) $(m, n, \mathsf{c}) \longrightarrow (1, 0, \mathsf{e})$ w.p. pq
  (iii) $(m, n, \mathsf{c}) \longrightarrow (0, 1, \mathsf{c})$ w.p. $(1 - p)(1 - q)$
  (iv) $(m, n, \mathsf{c}) \longrightarrow (0, 1, \mathsf{e})$ w.p. $p(1 - q)$.

If $Z_k = \mathsf{e}$ then the only allowable action is $u = 1$ and we enter into the state ϕ. If the current state is ϕ, we stay in the same cost-free termination state irrespective of the control u. The one step cost when the state is $s \in \mathcal{S}$ is given by:

$$c(s, u) = \begin{cases} d(m, n) & \text{if } s = (m, n, \mathsf{e}), \\ \lambda + d(m, n) & \text{if } u = 1 \text{ and } s = (m, n, \mathsf{c}), \\ 0 & \text{if } u = 0 \text{ or } s = \phi. \end{cases}$$

For simplicity we write the state (m, n, c) as simply (m, n).

3.2. Optimal Placement Set $\mathcal{P}_\lambda$

Let $J_\lambda(m, n)$ denote the optimal cost-to-go when the current state is (m, n). When at some step the state is (m, n) the deployment person has to decide whether to place or not place a relay at the current step. J_λ is the solution of the Bellman equation [22, Page 137, Prop. 1.1],

$$J_\lambda(m, n) = \min\{c_p(m, n), c_{np}(m, n)\}, \tag{6}$$

where $c_p(m,n)$ and $c_{np}(m,n)$ denote the expected cost incurred when the decision is to *place* and *not place* a relay, respectively. $c_p(m,n)$ is given by

$$\begin{aligned} c_p(m,n) &= \lambda + d(m,n) + (1-p)(1-q)J_\lambda(0,1) \\ &\quad +(1-p)qJ_\lambda(1,0) + pd(1). \end{aligned} \tag{7}$$

The term $\lambda + d(m,n)$ in the above expression is the one step cost which is first incurred when a relay is placed. The remaining terms are the average cost-to-go from the next step. The term $(1-p)(1-q)J_\lambda(0,1)$ can be understood as follows: $(1-p)(1-q)$ is the probability that the path proceeds in the $+\mathsf{x}$ direction without ending. Thus the state at the next step is $(0,1,\mathsf{c})$ w.p. $(1-p)(1-q)$, the optimal cost-to-go from which is $J_\lambda(0,1)$. Similarly for the term $(1-p)qJ_\lambda(1,0)$, $(1-p)q$ is the probability that the path will proceed, without ending, towards the $+\mathsf{y}$ direction (thus the next state is $(1,0,\mathsf{c})$) and $J_\lambda(1,0)$ is the cost-to-go from the next state. Finally, in the term $pd(1)$, p is the probability that the path will end, either proceeding in the $+\mathsf{x}$ direction, or in the $+\mathsf{y}$ direction, at the next step and $d(1)$ is the cost of the last link. Following a similar explanation, the expression for $c_{np}(m,n)$ can be written as:

$$\begin{aligned} &c_{np}(m,n) = \\ &\quad (1-p)qJ_\lambda(m+1,n) + (1-p)(1-q)J_\lambda(m,n+1) \\ &\quad +pqd(m+1,n) + p(1-q)d(m,n+1). \end{aligned} \tag{8}$$

We define the optimal placement set $\mathcal{P}_\lambda$ as the set of all lattice points (m,n), where it is optimal to place rather than to not place a relay. Formally,

$$\mathcal{P}_\lambda = \Big\{(m,n) : c_p(m,n) \leq c_{np}(m,n)\Big\}. \tag{9}$$

In this definition, if the costs of placing and not-placing are the same, we have arbitrarily chosen to place at that point, which is equivalent from the point of view of minimizing the total cost-to-go. This is because, if we do not place the relay at the current step and the path continues, the earliest opportunity to place a relay is at the next (random) lattice point, thus increasing our current hop-length. Hence, although we do not incur any relay-cost $\lambda > 0$ at the current step, the total expected placement cost will increase due to increased value of the current hop-cost, which is strictly increasing in hop-length. On the other hand, if we do place a relay at the current step, we will incur a placement cost of λ immediately at our current step plus a smaller hop-cost for the next hop, due to convexity of the hop-cost function $d(\cdot)$ and relay placement points being regenerative points for the process. Hence when these two costs are equal, either action is as good for minimizing the total cost.

The above result yields the following main theorem of this section which characterizes the optimal placement set $\mathcal{P}_\lambda$ in terms of a boundary.

Theorem 1. *The optimal placement set $\mathcal{P}_\lambda$ is characterized by a boundary, i.e.,* *there exist mappings $m^* : \mathbb{Z}_+ \to \mathbb{Z}_+$ and $n^* : \mathbb{Z}_+ \to \mathbb{Z}_+$ such that:*

$$\begin{aligned} \mathcal{P}_\lambda &= \bigcup_{n \in \mathbb{Z}_+} \{(m,n) : m \geq m^*(n)\} & (10) \\ &= \bigcup_{m \in \mathbb{Z}_+} \{(m,n) : n \geq n^*(m)\}. & (11) \end{aligned}$$

Proof Outline. The proof utilizes the conditions **C2** and **C3** imposed on the cost function $d(\cdot)$. First, using (7) and (8) in (9) and rearranging we alternatively write $\mathcal{P}_\lambda$ as, $\mathcal{P}_\lambda = \{(m,n) : F(m,n) \geq K\}$, where K is a constant and $F(\cdot,\cdot)$ is some function of m and n. Then, we complete the proof by showing that $F(m,n)$ is non-decreasing in both m and n. This requires us to prove (using an induction argument) that $H_\lambda(m,n) := J_\lambda(m,n) - d(m,n)$ is non-decreasing in m and n. Also, Lemma 2 has to be used here. For a formal proof see Appendix B. □

Remark: Though the optimal placement set $\mathcal{P}_\lambda$ was characterized nicely in terms of a boundary $m^*(\cdot)$ and $n^*(\cdot)$, a naive approach of computing this boundary, using value iteration to obtain $J_\lambda(m,n)$ (for several values of $(m,n) \in \mathbb{Z}_+^2$), would be computationally intensive. Our effort in the next section (Section 4) is towards obtaining an alternate simplified representation for $\mathcal{P}_\lambda$, using which we propose an algorithm in Section 5, which is guaranteed to return $\mathcal{P}_\lambda$ in a finite (in practice, small) number of steps.

4. Optimal Stopping Formulation

We observe that the points where the path has not ended, and a relay is placed, are renewal points of the decision process. This motivates us to think of the decision process after a relay is placed as an optimal stopping problem with *termination cost* $J_\lambda(0,0)$ (which is the optimal cost-to-go from a relay placement point). Let $\overline{\mathcal{P}}_\lambda$ denote the placement set corresponding to the OSLA rule (to be defined next). In this section we prove our next main result that $\mathcal{P}_\lambda = \overline{\mathcal{P}}_\lambda$.

4.1. One-Step-Look-Ahead Stopping Set $\overline{\mathcal{P}}_\lambda$

Under the OSLA rule, a relay is placed at state (m,n,c) if and only if the "cost $c_1(m,n)$ of *stopping* (i.e., placing a relay) at the current step" is less than the "cost $c_2(m,n)$ of continuing (without placing relay at the current step) for one more step, and then stopping (i.e., placing a relay at the next step)". The expressions for the costs $c_1(m,n)$ and $c_2(m,n)$ can be written as:

$$c_1(m,n) = \lambda + d(m,n) + J_\lambda(0,0)$$

and

$$\begin{aligned} &c_2(m,n) = \\ &\quad pq(d(m+1,n) + p(1-q)d(m,n+1)) + (1-p) \\ &\quad \Big(qd(m+1,n) + (1-q)d(m,n+1) + \lambda + J_\lambda(0,0)\Big). \end{aligned}$$

Then we define the OSLA placement set $\overline{\mathcal{P}}_\lambda$ as:

$$\overline{\mathcal{P}}_\lambda \quad = \quad \{(m,n) \in \mathbb{Z}_+^2 : c_1(m,n) \leq c_2(m,n)\}.$$

Substituting for $c_1(m,n)$ and $c_2(m,n)$ and simplifying we obtain:

$$\overline{\mathcal{P}}_\lambda = \Big\{(m,n) \in \mathbb{Z}_+^2 : p(\lambda + J_\lambda(0,0)) \leq \Delta_q(m,n)\Big\}, \tag{12}$$

where $\Delta_q(m,n) = q\Delta_1(m,n) + (1-q)\Delta_2(m,n)$.

Theorem 2. *The OSLA rule is a threshold policy, i.e., there exist mappings* $\bar{m} : \mathbb{Z}_+ \to \mathbb{Z}_+$ *and* $\bar{n} : \mathbb{Z}_+ \to \mathbb{Z}_+$*, which define the one-step placement set* $\overline{\mathcal{P}}_\lambda$ *as follows,*

$$\begin{aligned} \overline{\mathcal{P}}_\lambda &= \bigcup_{n \in \mathbb{Z}_+} \{(m,n) : m \geq \bar{m}(n)\} & (13) \\ &= \bigcup_{m \in \mathbb{Z}_+} \{(m,n) : n \geq \bar{n}(m)\}. & (14) \end{aligned}$$

Proof. Noticing that in (12) $\Delta_q(m,n)$ is non-decreasing in (m,n) and $p(\lambda + J_\lambda(0,0))$ is a constant, the proof follows along the lines of the proof of Theorem 1. □

Now, we present the main theorem of this section.

Theorem 3.

$$\mathcal{P}_\lambda = \overline{\mathcal{P}}_\lambda.$$

Proof. See Appendix C. □

Remark: The characterization in (12) is much simpler than the one in (9) once the value of $J_\lambda(0,0)$ is given. In the following subsection, we define a function $g(\cdot)$ and express $J_\lambda(0,0)$ as the minimum value of this function.

4.2. Computation of $J_\lambda(0,0)$

Let us start by defining a collection of placement sets indexed by $h \geq 0$:

$$\mathcal{P}(h) = \{(m,n) \in \mathbb{Z}_+^2 : p(\lambda + h) \leq \Delta_q(m,n)\}. \tag{15}$$

Referring to (12), note that $\mathcal{P}(J_\lambda(0,0)) = \overline{\mathcal{P}}_\lambda$. Let $g(h)$ denote the cost-to-go, starting from $(0,0)$, if the placement set $\mathcal{P}(h)$ is employed. Then, since $J_\lambda(0,0)$ is the optimal cost-to-go and $\mathcal{P}_\lambda \in \{\mathcal{P}(h)\}_{h \geq 0}$, we have $J_\lambda(0,0) = \min_{h \geq 0} g(h)$.

To compute $g(h)$, we proceed by defining the boundary $\mathcal{B}(h)$ of $\mathcal{P}(h)$ as follows:

$$\begin{aligned} \mathcal{B}(h) \quad = \quad & \{(m,n) \in \mathcal{P}(h) : (m-1,n) \in \mathcal{P}^c(h) \text{ or} \\ & (m, n-1) \in \mathcal{P}^c(h)\}, \end{aligned} \tag{16}$$

where $\mathcal{P}^c(h) := \mathbb{Z}_+^2 - \mathcal{P}(h)$.

Suppose the corridor ends at some $(m,n) \in \mathcal{P}^c(h) \cup \mathcal{B}(h)$, then only a cost of $d(m,n)$ is incurred. Otherwise (i.e., if the corridor reaches some $(m,n) \in \mathcal{B}(h)$ and continues), using a renewal argument, a cost of $d(m,n)+\lambda+g(h)$ is incurred, where $d(m,n)+\lambda$ is the cost of placing a relay and $g(h)$ is the future cost-to-go. We can thus write:

$$\begin{aligned} g(h) \quad &= \sum_{(m,n)\in\mathcal{P}^c(h)\cup\mathcal{B}(h)} \mathbb{P}((m,n),\mathsf{e})d(m,n) + \\ &\quad \sum_{(m,n)\in\mathcal{B}(h)} \mathbb{P}((m,n),\mathsf{c})(g(h)+\lambda+d(m,n)), \end{aligned} \tag{17}$$

where $\mathbb{P}((m,n),\mathsf{e})$ is the probability of the corridor ending at (m,n) and $\mathbb{P}((m,n),\mathsf{c})$ is the probability of the corridor reaching the boundary and continuing. Solving for $g(h)$, we obtain:

$$\begin{aligned} g(h) \quad &= \frac{1}{1-\sum_{(m,n)\in\mathcal{B}(h)}\mathbb{P}((m,n),\mathsf{c})} \left(\sum_{(m,n)\in\mathcal{P}^c(h)\cup\mathcal{B}(h)} \mathbb{P}((m,n),\mathsf{e})d(m,n) + \right. \\ &\quad \left. \sum_{(m,n)\in\mathcal{B}(h)} \mathbb{P}((m,n),\mathsf{c})(\lambda+d(m,n)) \right). \end{aligned} \tag{18}$$

The above expression is extensively used in our algorithm proposed in the next section.

We conclude this subsection by deriving the expression for the probabilities $\mathbb{P}((m,n),\mathsf{e})$ and $\mathbb{P}((m,n),\mathsf{c})$. Let us partition the boundary $\mathcal{B}(h)$ into three mutually disjoint sets:

$$\begin{aligned} \mathcal{B}^w(h) \quad &= \quad \{(m,n)\in\mathcal{B}(h) : (m-1,n)\in\mathcal{B}(h)\} \\ \mathcal{B}^s(h) \quad &= \quad \{(m,n)\in\mathcal{B}(h) : (m,n-1)\in\mathcal{B}(h)\} \\ \mathcal{B}^{null}(h) \quad &= \quad \{(m,n)\in\mathcal{B}(h) : (m-1,n)\notin\mathcal{B}(h) \text{ and} \\ & \qquad (m,n-1)\notin\mathcal{B}(h)\}. \end{aligned}$$

For a depiction of the various boundary points, see Fig. 4. Now, $\mathbb{P}((m,n),\mathsf{e})$ can be written as:

$$\mathbb{P}((m,n),\mathsf{e}) = \\ \begin{cases} \binom{m+n}{m}p(1-p)^{m+n-1}q^m(1-q)^n \\ \qquad\qquad\qquad \text{if } (m,n)\in\mathcal{P}^c(h)\cup\mathcal{B}^{null}(h) \\ \binom{m+n-1}{m}p(1-p)^{m+n-1}q^m(1-q)^n \text{ if } (m,n)\in\mathcal{B}^w(h) \\ \binom{m+n-1}{m-1}p(1-p)^{m+n-1}q^m(1-q)^n \text{ if } (m,n)\in\mathcal{B}^s(h). \end{cases}$$

This can be understood as follows. Any point $(m,n)\in\mathcal{P}^c(h)\cup\mathcal{B}^{null}(h)$ can be reached from the left or from below. $\binom{m+n}{m}$ is the number of possible paths for reaching (m,n). Each such path has to go m times in the $+\mathsf{x}$ direction (thus the

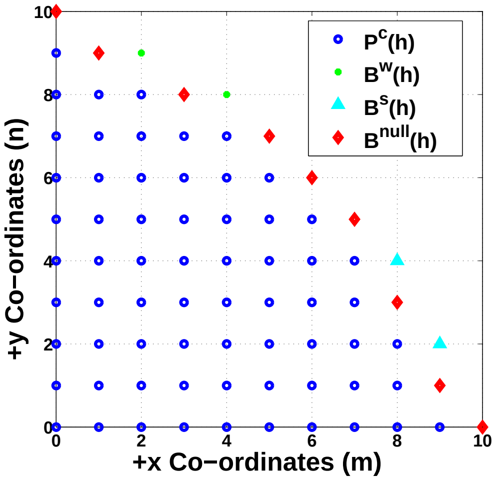


Figure 4: Example of a placement set of the form in (15): 'o' denotes lattice points outside the placement set; lattice points on the boundary can be partitioned into three sets according to the direction, from which they can be reached.

term q^m) and n times in the +y direction (thus the term $(1-q)^n$) and finally ending at (m,n) (thus the term $p(1-p)^{m+n-1}$). Any point $(m,n) \in \mathcal{B}^w(h)$ can be reached only from the point $(m, n-1)$. The probability of reaching $(m, n-1)$ without ending is $\binom{m+n-1}{m}(1-p)^{m+n-1}q^m(1-q)^{n-1}$. Then, the corridor reaches (m,n) and ends with probability $p(1-q)$. $\mathbb{P}((m,n),\mathsf{e})$ for $(m,n) \in \mathcal{B}^s(h)$ can be obtained analogously.

Similarly, $\mathbb{P}((m,n),\mathsf{c})$ can be written as:

$$\mathbb{P}((m,n),\mathsf{c}) = \begin{cases} \binom{m+n}{m}(1-p)^{m+n}q^m(1-q)^n \\ \qquad\qquad\qquad \text{if } (m,n) \in \mathcal{P}^c(h) \cup \mathcal{B}^{null}(h) \\ \binom{m+n-1}{m}(1-p)^{m+n}q^m(1-q)^n \text{ if } (m,n) \in \mathcal{B}^w(h) \\ \binom{m+n-1}{m-1}(1-p)^{m+n}q^m(1-q)^n \text{ if } (m,n) \in \mathcal{B}^s(h). \end{cases}$$

5. OSLA Based Fixed Point Iteration Algorithm

In this section, we present an efficient fixed point iteration algorithm (Algorithm 1) using the OSLA rule in (12) for obtaining the optimal placement set, $\mathcal{P}_\lambda$, and the optimal cost-to-go, $J_\lambda(0,0)$. There are two advantages of our algorithm over the naive approach of directly trying to minimize the function $g(\cdot)$ to obtain $J_\lambda(0,0)$ (recall that $J_\lambda(0,0) = \min_{h \geq 0} g(h)$):

- On the theoretical side, this iterative algorithm avoids explicit optimization altogether, which, otherwise would be performed numerically over a continuous range. Without any structure on the objective function, direct numerical minimization of $g(\cdot)$ is difficult and often unsatisfactory, as it invariably uses some sort of heuristic search over this continuous range.

- On the practical side, this algorithm is proved to converge within a finite number of iterations and observed to be extremely fast (requires 3 to 4 iterations typically).

The following is our Algorithm (Algorithm 1) which we refer to as the OSLA Based Fixed Point Iteration Algorithm.

Algorithm 1 OSLA Based Fixed Point Iteration Algorithm

```
**Require:** $0 < p < 1$, $0 \leq q \leq 1$, $\lambda \geq 0$
$k = 0$, $h^{(k)} = 0$
**while** 1 **do**
    $\mathcal{P}(h^{(k)}) \leftarrow \{(m,n) \in \mathbb{Z}_+^2 : p(\lambda + h^{(k)}) \leq \Delta_q(m,n)\}$
    Compute $g(h^{(k)})$ using (18)
    **if** $g(h^{(k)}) == h^{(k)}$ **then**
        Break;
    **end if**
    $h^{(k+1)} \leftarrow g(h^{(k)})$
    $k \leftarrow k + 1$
**end while**
**return** $g(h^{(k)})$, $\mathcal{P}(h^{(k)})$
```

We now prove the correctness and finite termination properties of our algorithm. First, we define $g^* := J_\lambda(0,0) = \min_{h \geq 0} g(h)$. Now consider a sample plot of the function $g(h)$ in Fig. 5. From Fig. 5(a) observe that whenever $h > g^*$ (which is around 150), $h > g(h)$. Also, Fig. 5(b) (where we have plotted the functions $g(h)$ and $l(h) = h$) suggests that $g(h)$ has a unique fixed point. We formally prove these results.

Lemma 3. *If $h > g^*$ then $h > g(h)$.*

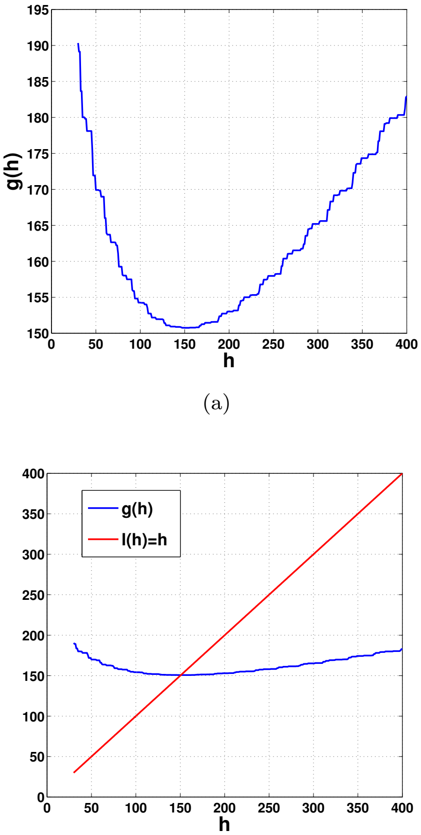
Proof. This follows from the manipulation of (18). See Appendix D for details. □

Lemma 4. *$g(h)$ has a unique fixed point.*

Proof. From (15) and (12), we observe that $\mathcal{P}(J_\lambda(0,0)) = \overline{\mathcal{P}}_\lambda$. From Theorem 3, $\overline{\mathcal{P}}_\lambda$ is the optimal placement set and thus the cost-to-go of using $\mathcal{P}(J_\lambda(0,0))$ is $J_\lambda(0,0)$, i.e., $g(J_\lambda(0,0)) = J_\lambda(0,0)$. Hence, $J_\lambda(0,0) = g^*$ is a fixed point of $g(\cdot)$. Now, any $h > g^*$ cannot be a fixed point since, in this case, $h > g(h)$ from Lemma 3. On the other hand, any $h < g^*$ is such that $h < g^* \leq g(h)$ because g^* is the optimal cost-to-go. Hence, g^* is the unique fixed point of $g(\cdot)$. □

We are now ready to prove the finite convergence property of our Algorithm.

Lemma 5. 1. *The sequence $\{h^{(k)}\}_{k \geq 1}$ (in Algorithm 1) is non-increasing, i.e., $h^{(k+1)} \leq h^{(k)}$, with the equality sign holding if and only if $h^{(k)} = g^*$.*



(a)

(b)

Figure 5: (a) Cost-to-go $g(h)$ as a function of h (b) Zoom on the cost-to-go $g(h)$ as a function of h. These plots are for $p = 0.02$, $q = 0.5$, and $\lambda = 41$.

2. *The sequence* $\{\mathcal{P}^c(h^{(k)})\}_{k\geq 1}$ *is non-increasing, i.e.,* $\mathcal{P}^c(h^{(k+1)}) \subseteq \mathcal{P}^c(h^{(k)})$, *where the containment is strict whenever* $\mathcal{P}^c(h^{(k+1)}) \subsetneq {\mathcal{P}_\lambda}^c$.

Proof. The first part of the Lemma follows from application of Lemma 3 and the second part follows from application of Lemma 4. See Appendix E for details. □

Theorem 4. *Algorithm 1 returns* g^* *and* ${\mathcal{P}_\lambda}^c$ *in a finite number of steps.*

Proof. Noting that $h^{(1)} = g(h^{(0)}) \geq g^*$ and using (15), we have ${\mathcal{P}_\lambda}^c \subseteq \mathcal{P}^c(h^{(1)})$. Either ${\mathcal{P}_\lambda}^c = \mathcal{P}^c(h^{(1)})$, in which case the algorithm terminates. Otherwise, note that both sets, ${\mathcal{P}_\lambda}^c$ and $\mathcal{P}^c(h^{(1)})$ contain a finite number of lattice points (from the definition of $\mathcal{P}(h)$ in (15)). Using Lemma 5, $\mathcal{P}^c(h^{(k)})$ converges to ${\mathcal{P}_\lambda}^c$ in at most $|\mathcal{P}^c(h^{(1)}) \setminus {\mathcal{P}_\lambda}^c| < \infty$ iterations. We can also obtain a crude upper bound for the number of iterations needed. Assuming that we start with $h^{(0)} = 0$, $|\mathcal{P}^c(h^{(1)}) \setminus \mathcal{P}_\lambda{}^c| = |\{(m,n) \in \mathbb{Z}_+^2 : p(\lambda + g^*) \leq \Delta_q(m,n) \leq p(\lambda + g(0))\}|$. Once $\mathcal{P}^c(h^{(k)})$ converges to $\mathcal{P}_\lambda$, the algorithm terminates and returns the optimal cost-to-go g^*. □

Performance comparison of Algorithm 1 with naive value iteration

To obtain the optimal placement policy directly using Eqn.(9) of section (3), we need to compute the values of $J(m,n)$ for all $(m,n) \in \mathbb{Z}_+^2$, which is usually obtained through value iterations. Since the state-space under consideration is *countably-infinite*, a direct numerical evaluation of the optimal *cost-to-go* function is computationally prohibitive. See section 3.1 of [22] for details.

On the contrary, exploiting the structure of the problem, Algorithm 1 performs a finite amount of computation per iteration (which is bounded by the number of lattice points in the set $\mathcal{P}_\lambda^c(h^{(k)}$ (See Eqn.15)) and correctly terminates after provably finite number of iterations.

6. Solving the Constrained Problem

In this section, we devise a method to solve the constrained problem in (4) using the solution of the unconstrained problem (3) provided by Algorithm 1. This method is applied in Section 7.2 where, imposing a constraint on the average number of relays, we compare the performance of a distance based heuristic with the optimal.

We begin with the following standard result which relates the solutions of the problems in (3) and (4). See also [21].

Lemma 6. *Let $\pi_\lambda^* \in \Pi$ be an optimal policy for the unconstrained problem in (3) such that $\mathbb{E}_{\pi_\lambda^*} N = \rho_{avg}$. Then π_λ^* is also optimal for the constrained problem in (4).*

Proof. (We provide this proof for completeness.) Since π_λ^* is optimal for the unconstrained problem in (3) we can write, for any $\pi \in \Pi$,

$$\mathbb{E}_{\pi_\lambda^*} C + \lambda \mathbb{E}_{\pi_\lambda^*} N \quad \leq \quad \mathbb{E}_\pi C + \lambda \mathbb{E}_\pi N.$$

Rearranging the above expression and using $\mathbb{E}_{\pi_\lambda^*} N = \rho_{avg}$, we obtain

$$\mathbb{E}_{\pi_\lambda^*} C \quad \leq \quad \mathbb{E}_\pi C + \lambda \Big(\mathbb{E}_\pi N - \rho_{avg} \Big).$$

Thus, $\mathbb{E}_{\pi_\lambda^*} C \leq \mathbb{E}_\pi C$ for any π such that $\mathbb{E}_\pi N \leq \rho_{avg}$. □

However, the above lemma is useful only when we are able to exhibit a λ such that $\mathbb{E}_{\pi_\lambda^*} N = \rho_{avg}$. The subsequent development in this section is towards obtaining the solution to the more general case.

The expected number of relays used by the optimal policy, π_λ^*, which uses the optimal placement set $\mathcal{P}_\lambda$, can be computed as:

$$\mathbb{E}_{\pi_\lambda^*} N = \frac{\sum_{(m,n)\in\mathcal{B}_\lambda} \mathbb{P}((m,n),\mathsf{c})}{1-\sum_{(m,n)\in\mathcal{B}_\lambda} \mathbb{P}((m,n),\mathsf{c})}, \tag{19}$$

where $\mathbb{P}((m,n),\mathsf{c})$ is the reaching probability corresponding to $\mathcal{P}_\lambda$ and $\mathcal{B}_\lambda$ is the boundary of $\mathcal{P}_\lambda$. A plot of $\mathbb{E}_{\pi_\lambda^*} N$ vs. λ is given in Fig. 6. We make the following observations about $\mathbb{E}_{\pi_\lambda^*} N$.

1) $\mathbb{E}_{\pi_\lambda^*} N$ decreases with λ; this is as expected, since as each relay becomes "costlier" fewer relays are used on the average.

2) Even when $\lambda = 0$, $\mathbb{E}_{\pi_\lambda^*} N$ is finite. This is because $d(0) > 0$, i.e., there is a positive cost for a 0 length link. Define the value of $\mathbb{E}_{\pi_\lambda^*} N$ with $\lambda = 0$ to be $\rho_{\max}$.

3) $\mathbb{E}_{\pi_\lambda^*} N$ vs. λ is a piecewise constant function. This occurs because the relay placement positions are discrete. For a range of values of λ the same threshold is optimal. This structure is also evident from the results based on the optimal stopping formulation and the OSLA rule in Section 4. It follows that for a value of λ at which there is a step in the plot, there are two optimal deterministic policies, $\underline{\pi}$ and $\overline{\pi}$, for the relaxed problem. Let $\underline{\rho} = \mathbb{E}_{\underline{\pi}} N$ and $\overline{\rho} = \mathbb{E}_{\overline{\pi}} N$.

We have the following structure of the optimal policy for the constrained problem:

Theorem 5. 1. *For $\rho_{avg} \geq \rho_{max}$ the optimal placement set is obtained for $\lambda = 0$, i.e., is $\mathcal{P}_0$.*

2. *For $\rho_{avg} < \rho_{max}$, if there is a λ such that (a) $\mathbb{E}_{\pi_\lambda^*} N = \rho_{avg}$ then the optimal policy is π_λ^*, or (b) $\underline{\rho} < \rho_{avg} < \overline{\rho}$ then the optimal policy is obtained by mixing $\underline{\pi}$ and $\overline{\pi}$.*

Proof. 1) is straight forward. For proof of 2)-(a), Lemma 6. Considering now 2)-(b), define $0 < \alpha < 1$ such that $(1-\alpha)\underline{\rho} + \alpha\bar{\rho} = \rho_{avg}$. We obtain a mixing policy π_m by choosing $\underline{\pi}$ w.p. $1-\alpha$ and $\bar{\pi}$ w.p. α at the beginning of the deployment. For any policy π we have the following standard argument:

$$\begin{aligned} &\mathbb{E}_{\pi_m} C + \lambda \mathbb{E}_{\pi_m} N \\ &\quad = \quad (1-\alpha)(\mathbb{E}_{\underline{\pi}} C + \lambda\underline{\rho}) + \alpha(\mathbb{E}_{\bar{\pi}} C + \lambda\bar{\rho}) \\ &\quad \leq \quad (1-\alpha)(\mathbb{E}_{\pi} C + \lambda\mathbb{E}_{\pi} N) + \alpha(\mathbb{E}_{\pi} C + \lambda\mathbb{E}_{\pi} N) \\ &\quad = \quad \mathbb{E}_{\pi} C + \lambda\mathbb{E}_{\pi} N. \end{aligned} \tag{20}$$

The inequality is because $\underline{\pi}$ and $\overline{\pi}$ are both optimal for the problem (3) with relay price λ. Thus, we have shown that π_m is also optimal for the relaxed problem. Using this along with $\mathbb{E}_{\pi_m} N = \rho_{avg}$ in Lemma 6, we conclude the proof. □

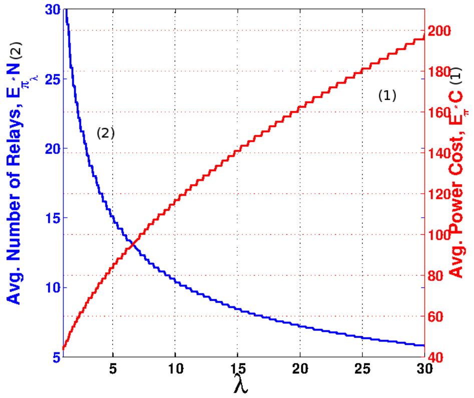


Figure 6: Average number of relays $\mathbb{E}_{\pi^*_\lambda}N$ (left scale; curve labelled (2)) and average power cost $\mathbb{E}_{\pi^*_\lambda}C$ (right scale; curve labelled (1)) as a function of λ ($p = 0.002$, $q = 0.5$ and $\eta = 2$).

7. Numerical Work

For our numerical work we use the one-hop power function $d(r) = P_m + \gamma r^\eta$, with $P_m = 0.1$, $\gamma = 0.01$. We first study the effect of parameter variation on the various costs. Next, we compare the performance of a distance based heuristic with the optimal.

7.1. Effect of Parameter Variation

In Fig. 4, we have already shown an optimal placement boundary for $p = 0.002$, $q = 0.5$, and $\eta = 3$. Since $q = 0.5$ the boundary is symmetric about the $m = n$ line.

In Fig. 6, we plot $E_{\pi^*_\lambda}N$ and $E_{\pi^*_\lambda}C$ vs. λ. The plot of $J_\lambda(0, 0)$ vs. λ is in Fig. 7. These plots are for $p = 0.002$ and $q = 0.5$. Since λ is the cost per relay, as expected, $E_{\pi^*_\lambda}N$ decreases as λ increases. We observe that $E_{\pi^*_\lambda}C$ and the optimal total cost $J_\lambda(0, 0)$ increase as λ increases. A close examination of Fig. 6 reveals that both the plots are step functions. This is due to the discrete placement at lattice points, which results in the same placement boundary being optimal for a range of λ values. Thus, as seen in Section 6, at the λ values, where there is jump in $E_{\pi^*_\lambda}N$, a random mixture of two policies is needed.

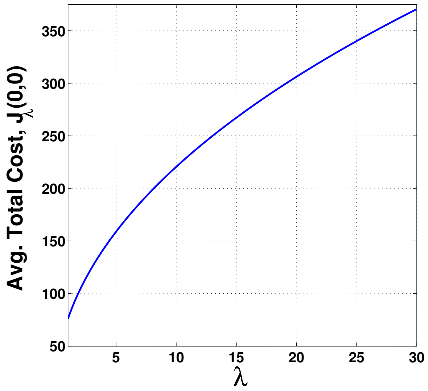


Figure 7: Average total cost $J_\lambda(0,0)$ as a function of λ ($p = 0.002$, $q = 0.5$ and $\eta = 2$).

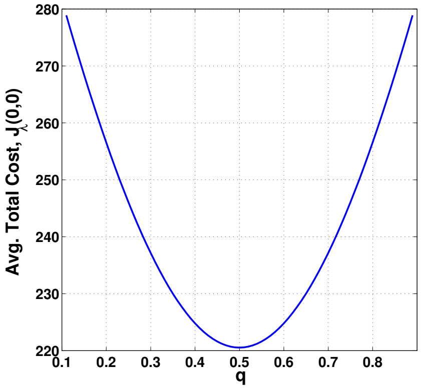


Figure 8: Average total cost $J_\lambda(0,0)$ as a function of q ($p = 0.002$ and $\eta = 2$).

Fig. 8 shows the variation of the total optimal cost $J_\lambda(0,0)$ with q. The variation is symmetric about $q = 0.5$. For a given probability p of the path ending, $q = 0.5$ results in the path folding frequently. In such a case, since the path-loss is isotropic, fewer relays are required to be placed. On the other hand, when q is close to 0 or to 1 the path takes fewer turns and more relays are needed, leading to larger values of the total cost.

In Fig. 9 we show the variation of optimal boundaries with η. As η, the path-loss exponent, increases the hop cost increases for a given hop distance. This results in relays needing to be placed more frequently. As can be seen the placement boundaries shrink with increasing η. We also notice that the placement boundary for $\eta = 2$ is a straight line; indeed this provable result holds for $\eta = 2$ for any values of p and q.

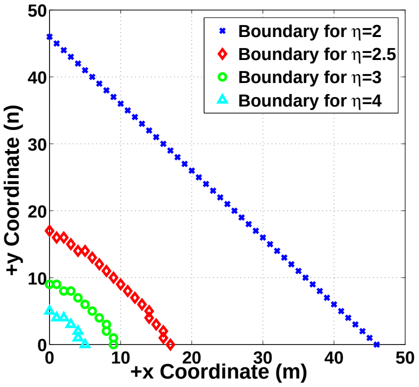


Figure 9: Boundaries for various values of the path-loss exponent η ($p = 0.002$, $q = 0.5$).

7.2. Comparison with the Distance Based Heuristic

We recall from the literature survey in Section 1 that prior work invariably proposed the policy of placing a relay after the RF signal strength from the previous relay dropped below a threshold. For isotropic propagation (as we have assumed in this paper), this is equivalent to placing the relay after a circular boundary is crossed. With this in mind, we obtained the *optimal constant distance placement policy* (called *the heuristic* hereafter) numerically in a manner similar to what is described in Section 4.2. A sample result is provided in Fig. 10, for the parameters $p = 0.002$, $q = 0.5$ and $\eta = 2$. We observe that if the path were to evolve roughly in the +x direction, or the +y direction, then the heuristic will result in many more relays being placed. On the other hand, if the path evolves diagonally (which has higher probability) then the two placement boundaries will result in similar placement decisions.

This observation shows up in Fig. 11, where we show the cost incurred by the optimal policy (for $q = 0.5$ and for $q = 1$, which corresponds to a straight line corridor) and the heuristic ($q = 0.5$) vs. ρ for the constrained problem. As expected, the cost is much larger for $q = 1$ since the path does not fold. We find that for $q = 0.5$ the optimal placement boundary and the heuristic provide costs that are almost indistinguishable at this scale. We have performed simulations by varying the system parameters and observed the same good performance of the optimal constant distance placement policy. This suggests that the heuristic policy performs well provided that the threshold distance is optimally chosen with respect to the system parameters.

8. Conclusion and Ongoing Work

We considered the problem of placing relays on a random lattice path to optimize a linear combination of average total hop cost and the average number of relays deployed. The optimal placement policy was proved to be of a threshold

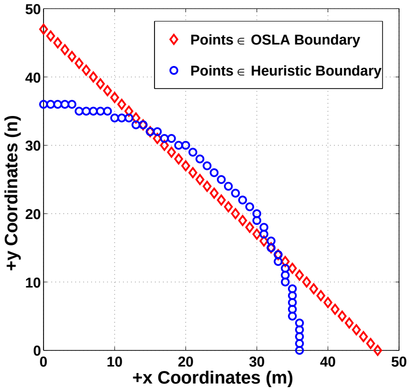


Figure 10: Boundary of the optimal placement set (OSLA boundary) and boundary derived from the heuristic policy ($p = 0.002$, $q = 0.5$ and $\eta = 2$).

type (Theorem 1). We further proved the optimality of the one-step-look-ahead (OSLA) rule (in Theorem 3). We have also devised an OSLA based fixed point iteration algorithm (Algorithm 1), which we have proved to converge to the optimal placement set in a finite number of steps. In our numerical work we assumed the hop cost to be the transmitter power. We observed that the performance (in terms of average power incurred for a given relay constraint) of the optimal policy is close to that of the distance threshold policy provided that the threshold distance is optimally chosen with respect to the system parameters.

The work that we have presented in this paper can be extended in several directions:

1. *Measurement-based placement:* When the cost of a link is taken as the power required to sustain a certain quality of communication over it, in this paper, we have taken the approach of modeling the required power by a *function* of distance, by imposing a large shadowing and fading margin. An alternative would be to make a measurement, at each step, to obtain the power required to establish a good link from that point to the previous node. This measurement, the distance from the previous node, and a statistical model for the quality of links that will be encountered as-we-go, could be used to obtain a more efficient placement. In addition, instead of evaluating just the link to the last placed node, we could also evaluate the links to the nodes placed earlier. Of course, the measurements would require time to be spent at each step, which might not be feasible for rapid deployment, as would be necessary for first responders. A measurement-based approach is a topic of our ongoing work; our early results have been presented in [14].
2. *Multiple sources:* In the work presented here, only one packet source is placed (at the end of the path). The model can be extended to the case where multiple sources might need to be placed along the path. A param-

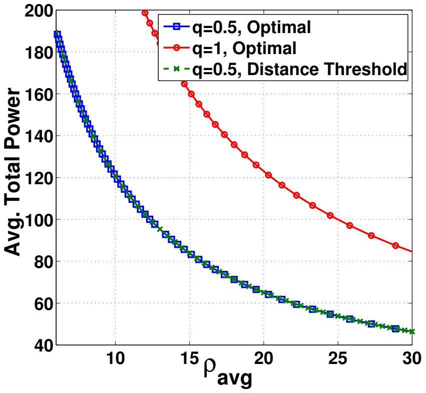


Figure 11: Average total power as a function of ρ for the optimal policy ($q = 0.5$ and $q = 1$, which corresponds to the straight line) and for the heuristic ($q = 0.5$) for $p = 0.002$ and $\eta = 2$.

eter, say $\sigma, 0 < \sigma < 1$, could be the probability that the current step is a source location given that the path has not ended thus far. When such a location is encountered, a node will necessarily have to be placed at that point. Since this would be a regeneration point for the remaining problem, we expect that results analogous to Theorem 1 and 3 can be proved. This will be a topic of our future work.

Appendix A. Proof of Lemma 2

Proof. It is easier to prove the lemma allowing the arguments m and n take values from the Real line. We have,

$$\Delta_1(x, y) = d(x + \delta, y) - d(x, y)$$

Partially differentiating both sides w.r.t. x, we get

$$\begin{aligned} \frac{\partial \Delta_1(x, y)}{\partial x} &= d_x(x + \delta, y) - d_x(x, y) \\ &= \delta d_{xx}(\zeta, y) \text{ where } x < \zeta < x + \delta \\ &> 0, \end{aligned}$$

where the equality follows from the application of Lagrange's Mean Value Theorem to the function $d_x(., y)$ and the inequality is due to assumption in (2). The above proves the fact that $\Delta_1(x, y)$ is non-decreasing in x.

To prove that $\Delta_1(x, y)$ is non-decreasing in y, we partially differentiate $\Delta_1(x, y)$ w.r.t. y and obtain

$$\begin{aligned} \frac{\partial \Delta_1(x, y)}{\partial y} &= d_y(x + \delta, y) - d_y(x, y) \\ &= \delta d_{xy}(\eta, y) \text{ where } x < \eta < x + \delta \\ &> 0, \end{aligned}$$

where the equality follows from the application of Lagrange's Mean Value Theorem to the function $d_y(.,y)$ and the inequality is due to assumption in (2). This shows that the function $\Delta_1(x,y)$ is non-decreasing in both the coordinates x and y. In a similar way it can also be shown that $\Delta_2(x,y)$ is non-decreasing in x and y under the assumption made in (2). This completes the proof. $\square$

Appendix B. Proof of Theorem 1

We begin by defining $H_\lambda(m,n) := J_\lambda(m,n) - d(m,n)$. Substituting for $c_p(m,n)$ and $c_{np}(m,n)$ (from (7) and (8), respectively) into (9) and rearranging we obtain (recall the definitions of $\Delta_1(m,n)$ and $\Delta_2(m,n)$ from Section 2):

$$\begin{aligned}
\mathcal{P}_\lambda = \\
&\Big\{(m,n):(1-p)(qH_\lambda(m+1,n)+(1-q)H_\lambda(m,n+1)) \\
&+p(q\Delta_1(m,n)+(1-q)\Delta_2(m,n)) \geq \lambda + \\
&(1-p)qJ_\lambda(1,0)+(1-p)(1-q)J_\lambda(0,1)+pd(1)\Big\}.
\end{aligned} \tag{B.1}$$

Lemma 7. *For a fixed λ, $H_\lambda(m,n)$ is non-decreasing in both $m \in \mathbb{Z}_+$ and $n \in \mathbb{Z}_+$.*

Proof. Consider a sequential relay placement problem where we have K steps to go. The corridor length is the minimum of K and of a geometric random variable with parameter p. The problem be formulated as a finite horizon MDP with horizon length K. For any given (m,n), $J_K(m,n)$, $K \geq 2$ is obtained recursively:

$$\begin{aligned}
J_K(m,n) &= \min\{c_p(m,n), c_{np}(m,n)\} \\
&= \min\{\lambda + d(m,n) + (1-p)qJ_{K-1}(1,0) + pqd(1) + \\
&\quad (1-p)(1-q)J_{K-1}(0,1) + p(1-q)d(1), \\
&\quad (1-p)qJ_{K-1}(m+1,n) + pqd(m+1,n) + \\
&\quad (1-p)(1-q)J_{K-1}(m,n+1)+p(1-q)d(m,n+1)\}.
\end{aligned}$$

For $K=1$, since the source must be placed at the next step, we have $J_1(m,n) = \min\{\lambda + d(m,n) + d(1), qd(m+1,n) + (1-q)d(m,n+1)\}$. Therefore,

$$\begin{aligned}
H_1(m,n) &:= J_1(m,n) - d(m,n) \\
&= \min\{\lambda + d(1), q\Delta_1(m,n) + (1-q)\Delta_2(m,n)\}.
\end{aligned}$$

From Lemma 2, it follows that $H_1(m,n)$ is non-decreasing in both m and n. Now we make the induction hypothesis and assume that $H_{K-1}(m,n)$ is non-decreasing in m and n. We have:

$$\begin{aligned}
H_K(m,n) &= J_K(m,n) - d(m,n) \\
&= \min\{\lambda + (1-p)qJ_{K-1}(1,0) + pqd(1) + \\
&\quad (1-p)(1-q)J_{K-1}(0,1) + p(1-q)d(1), (1-p) \\
&\quad (qH_{K-1}(m+1,n) + (1-q)H_{K-1}(m,n+1)) + \\
&\quad q\Delta_1(m,n) + (1-q)\Delta_2(m,n)\}.
\end{aligned}$$

By the induction hypothesis and Lemma 2, it follows that $H_K(m,n)$ is non-decreasing in both m and n. The proof is complete by taking the limit as $K \to \infty$. □

We are now ready to prove Theorem 1.

Proof of Theorem 1. Referring to (B.1), utilizing Lemma 7 and the Lemma 2, it follows that for a fixed $n \in \mathbb{Z}_+$, the LHS (Left Hand Side) of (B.1), describing the placement set $\mathcal{P}_\lambda$ is an increasing function of m, while the RHS (Right Hand Side) is a *finite* constant. Also, because of the assumed properties of the function $d(.)$, $\Delta_1(m,n) \to \infty$ as $m \to \infty$, for any fixed n. Hence it follows that there exists an $m^*(n) \in \mathbb{Z}_+$ such that $(m,n) \in \mathcal{P}_\lambda \quad \forall m \geq m^*(n)$. Hence we may write $P_\lambda = \bigcup_{n \in \mathbb{Z}_+} \{(m,n) | m \geq m^*(n)\}$. The second characterization follows by similar arguments. □

Appendix C. Proof of Theorem 3

We require the following lemmas to prove Theorem 3.

Lemma 8. $\mathcal{P}_\lambda \subset \overline{\mathcal{P}}_\lambda$

Proof. Suppose that $(m,n) \in \mathcal{P}_\lambda$. Then from (10) $(m+1,n) \in \mathcal{P}_\lambda$ and from (11), $(m,n+1) \in \mathcal{P}_\lambda$. Since $(m,n) \in \mathcal{P}_\lambda$, we have from (7), (8) and (9) that

$$\begin{aligned}&\lambda+d(m,n)+(1-p)qJ_\lambda(1,0)+pqd(1)+(1-p)(1-q)\times\\ &\quad J_\lambda(0,1)+p(1-q)d(1) \leq (1-p)qJ_\lambda(m+1,n)+pq\times\\ &\quad d(m+1,n)+(1-p)(1-q)J_\lambda(m,n+1)+p(1-q)d(m,n+1).\end{aligned} \tag{C.1}$$

Also we may argue that at the state $(0,0)$, it is optimal not to place. Indeed, if it had been optimal to place at the state $(0,0)$, at the next step, we return to the same state, viz., $(0,0)$. Now, because of the stationarity of the optimal policy, we would keep placing relays at the same point, and since "relay-cost" $\lambda > 0$ and $d(0,0) > 0$, the expected cost for this policy would be ∞. Hence,

$$\begin{aligned}J_\lambda(0,0) = (1-p)qJ_\lambda(1,0) + pqd(1) +\\ (1-p)(1-q)J_\lambda(0,1) + p(1-q)d(1).\end{aligned} \tag{C.2}$$

Since $(m+1,n) \in \mathcal{P}_\lambda$ and $(m,n+1) \in \mathcal{P}_\lambda$, we have (noticing that it is optimal to place at these points and utilizing (7) and (C.2)),

$$J_\lambda(m+1,n) \quad = \quad \lambda + d(m+1,n) + J_\lambda(0,0) \tag{C.3}$$

$$J_\lambda(m,n+1) \quad = \quad \lambda + d(m,n+1) + J_\lambda(0,0). \tag{C.4}$$

Now, using (C.2), (C.3) and (C.4) in (C.1), we obtain:

$$p(\lambda+J_\lambda(0,0)) \quad \leq \quad q\Delta_1(m,n)+(1-q)\Delta_2(m,n). \tag{C.5}$$

This proves that $(m,n) \in \bar{\mathcal{P}}_\lambda$ and hence $\mathcal{P}_\lambda \subset \overline{\mathcal{P}}_\lambda$ □

Using the above Lemma and from (10), (11), (13), (14) we can conclude that:

$$n^*(m) \geq \overline{n}(m) \quad \forall m \in \mathbb{Z}_+ \tag{C.6}$$

$$m^*(n) \geq \overline{m}(n) \quad \forall n \in \mathbb{Z}_+. \tag{C.7}$$

Lemma 9. *If $(m, n) \in \overline{\mathcal{P}}_\lambda$ is such that $(m, n+1) \in \mathcal{P}_\lambda$ and $(m+1, n) \in \mathcal{P}_\lambda$, then $(m, n) \in \mathcal{P}_\lambda$*

Proof. Since $(m, n) \in \bar{\mathcal{P}}_\lambda$, we have from (12),

$$p(\lambda + J_\lambda(0,0)) \leq q\Delta_1(m,n) + (1-q)\Delta_2(m,n). \tag{C.8}$$

Now $(m, n+1) \in \mathcal{P}_\lambda$, and $(m+1, n) \in \mathcal{P}_\lambda$, hence we have from (C.3) and (C.4):

$$\begin{aligned} J_\lambda(m+1,n) &= \lambda + d(m+1,n) + J_\lambda(0,0) \\ J_\lambda(m,n+1) &= \lambda + d(m,n+1) + J_\lambda(0,0). \end{aligned}$$

The expression (C.2) is always true. Now using (C.2) and the above two equations in inequality (C.8), we obtain (C.1), which proves that $(m, n) \in \mathcal{P}_\lambda$. □

Lemma 10. *If $(m, n) \in \mathcal{P}_\lambda$ (resp. $\overline{\mathcal{P}}_\lambda$), then $(m+k, n) \in \mathcal{P}_\lambda$ (resp. $\overline{\mathcal{P}}_\lambda$) and $(m, n+k) \in \mathcal{P}_\lambda$ (resp. $\overline{\mathcal{P}}_\lambda$) for any $k \in \mathbb{Z}_+$.*

Proof. The proof follows easily because the LHS of (B.1) is increasing in both m and n while the RHS is a constant. Similarly, the RHS of (12) is increasing in both m and n while the LHS is a constant. □

We can now prove the main theorem.

Proof of Theorem 3. We need to show that inequalities in (C.6) and (C.7) are equalities. For any $m \in \mathbb{Z}_+$, suppose that in (C.6) $n^*(m) > n^*(m) - 1 \geq \bar{n}(m)$. Then we have the following inclusions:

$$\begin{aligned} (m, n^*(m)) &\in \mathcal{P}_\lambda \\ (m, n^*(m) - 1) &\in \overline{\mathcal{P}}_\lambda \\ (m, n^*(m) - 1) &\notin \mathcal{P}_\lambda. \end{aligned} \tag{C.9}$$

Let us index the collection of lattice-points $(m+i, n^*(m) - 1)$ by $N_i, i \in \mathbb{Z}_+$. Since $(m, n^*(m) - 1) \in \overline{\mathcal{P}}_\lambda$, from Lemma 10, it follows that $N_i \in \overline{\mathcal{P}}_\lambda$. From (C.9), $N_0 \notin \mathcal{P}_\lambda$.

Then, the optimal policy being a threshold policy, we know that there exists a finite $k > 0$, s.t. $N_k \in \mathcal{P}_\lambda$, i.e.,

$$(m+k, n^*(m) - 1) \in \mathcal{P}_\lambda. \tag{C.10}$$

Again from Lemma 10, since $(m, n^*(m)) \in \mathcal{P}_\lambda$, we have for any $k > 0$:

$$(m+k-1, n^*(m)) \quad \in \quad \mathcal{P}_\lambda. \tag{C.11}$$

Now we see that for the point N_{k-1}, the conditions of Lemma 9 are satisfied. Hence $N_{k-1} \in \mathcal{P}_\lambda$. If $k = 1$, we already have a contradiction since $N_0 \notin \mathcal{P}_\lambda$. Otherwise for $k > 1$, using Lemma 10 and $N_{k-1} \in \mathcal{P}_\lambda$, we can show that N_{k-2} is subject to the conditions of Lemma 9 implying that $N_{k-2} \in \mathcal{P}_\lambda$. By iteration, we finally obtain that $N_0 \in \mathcal{P}_\lambda$, which contradicts (C.9) and proves the result. □

Appendix D. Proof of Lemma 3

We start by proving the following lemma.

Lemma 11. *For* any *placement set $\mathcal{P}(h)$ of the form in (15), we have:*

$$\sum_{(m,n)\in\mathcal{P}^c(h)} r(m,n)\Big(\Delta_q(m,n) - p(\lambda + g(h))\Big) \\ +d(0,0) + \lambda = 0, \tag{D.1}$$

where $r(m,n) = (1-p)^{m+n}\binom{m+n}{m}q^m(1-q)^n$.

Proof. We first introduce some notations and definitions.

Let us define a path σ as a possible realization of the corridor, starting from $(0,0)$ and let $\mathbb{P}(\sigma)$ be the probability of such a path. The set of all paths is denoted by Σ. Let Σ_{mn} denote the set of all paths that end at $(m,n) \in \mathcal{P}^c(h) \cup \mathcal{B}(h)$ and $\Sigma_{mn}(c)$ the set of all paths that hit $(m,n) \in \mathcal{B}(h)$ and continue.

Let us denote the set of edges whose both end vertices belong to the set $\mathcal{P}^c(h) \cup \mathcal{B}(h)$ by E. A path σ is completely characterized by its edge set E_σ.

The reaching probability, $r(m,n)$, of a point (m,n) is defined as the probability that a random path σ *reaches* the point (m,n) and continues for at least one step. Hence, $r(m,n) = (1-p)^{m+n}\binom{m+n}{m}q^m(1-q)^n$.

The incremental cost function $\delta : E \longrightarrow \mathbb{R}_+$ is defined as follows:

$$\delta(e) = \begin{cases} d(m+1,n) - d(m,n) = \Delta_1(m,n) \\ \qquad\qquad \text{if } e = \{(m,n),(m+1,n)\} \\ d(m,n+1) - d(m,n) = \Delta_2(m,n) \\ \qquad\qquad \text{if } e = \{(m,n),(m,n+1)\}. \end{cases} \tag{D.2}$$

For $(m,n) \in \sigma$, the incremental cost function allows us to write:

$$d(m,n) = \sum_{e \in E_\sigma \cap E} \delta(e) + d(0,0). \tag{D.3}$$

Now consider

$$\begin{aligned}
&\sum_{\mathcal{P}^c(h)\cup\mathcal{B}(h)} \mathbb{P}((m,n),\mathsf{e})d(m,n) + \sum_{\mathcal{B}(h)} \mathbb{P}((m,n),\mathsf{c})d(m,n) \\
&= \sum_{\mathcal{P}^c(h)\cup\mathcal{B}(h)} \sum_{\sigma\in\Sigma_{mn}} \mathbb{P}(\sigma)\Bigg(\sum_{e\in E_\sigma} \delta(e) + d(0,0)\Bigg) + \\
&\quad \sum_{\mathcal{B}(h)} \sum_{\sigma\in\Sigma_{mn}(c)} \mathbb{P}(\sigma)\Bigg(\sum_{e\in E_\sigma\cap E} \delta(e) + d(0,0)\Bigg) \\
&= \sum_{e\in E} \delta(e) \sum_{\sigma\in\Sigma : e\in E_\sigma} \mathbb{P}(\sigma) + d(0,0) \\
&= \sum_{e\in E} \delta(e)t(e) + d(0,0), \qquad \text{(D.4)}
\end{aligned}$$

where by $t(e)$ we denote the probability that a random path goes through the edge $e \in E$.

Now if e is horizontal, i.e., $e = \{(m,n),(m+1,n)\}, (m,n) \in \mathcal{P}^c(h)$, we have $t(e) = qr(m,n)$ and $\delta(e) = \Delta_1(m,n)$. Similarly if e is vertical, i.e., $e = \{(m,n),(m,n+1)\}, (m,n) \in \mathcal{P}^c(h)$, we have $t(e) = (1-q)r(m,n)$ and $\delta(e) = \Delta_2(m,n)$. Using these relations, we may rewrite (D.4) as follows:

$$\begin{aligned}
&\sum_{\mathcal{P}^c(h)} r(m,n)\Bigg(q\Delta_1(m,n) + (1-q)\Delta_2(m,n)\Bigg) + d(0,0) \\
&= \sum_{\mathcal{P}^c(h)} r(m,n)\Delta_q(m,n) + d(0,0). \qquad \text{(D.5)}
\end{aligned}$$

Now consider the probability $\sum_{(m,n)\in\mathcal{B}(h)} \mathbb{P}((m,n),\mathsf{c})$. It is the probability that a random path continues beyond the boundary $\mathcal{B}(h)$. Hence we may write

$$\begin{aligned}
\sum_{\mathcal{B}(h)} \mathbb{P}((m,n),\mathsf{c}) &= 1 - \sum_{\mathcal{P}^c(h)\cup\mathcal{B}(h)} \mathbb{P}((m,n),\mathsf{e}) \\
&= 1 - \sum_{\mathcal{P}^c(h)} r(m,n)p. \qquad \text{(D.6)}
\end{aligned}$$

Using (D.5) and (D.6) in (18) and simplifying, we obtain the result. □

Proof of Lemma 3. We recall the definition of $\mathcal{P}^c(h)$.

$$\mathcal{P}^c(h) = \{(m,n) \in \mathbb{Z}_+^2 : p(\lambda + h) > \Delta_q(m,n)\}. \qquad \text{(D.7)}$$

Since $h > g^*$, we immediately conclude that ${\mathcal{P}_\lambda}^c \subset \mathcal{P}^c(h)$. From (D.1) in Lemma 11, we may write for the optimal placement set $\mathcal{P}_\lambda$:

$$\begin{aligned}
\sum_{{\mathcal{P}_\lambda}^c} r(m,n)\Delta_q(m,n) &= p(\lambda + g^*) \sum_{{\mathcal{P}_\lambda}^c} r(m,n) \\
&\quad -(d(0,0) + \lambda). \qquad \text{(D.8)}
\end{aligned}$$

We may similarly write for the placement set $\mathcal{P}(h)$:

$$\begin{aligned}\sum_{\mathcal{P}^c(h)} r(m,n)\Delta_q(m,n) \quad &= \quad p(\lambda+g(h)) \sum_{\mathcal{P}^c(h)} r(m,n) \\ &\quad\quad -(d(0,0)+\lambda).\end{aligned} \tag{D.9}$$

Now, since ${\mathcal{P}_\lambda}^c \subset \mathcal{P}^c(h)$, we may expand the LHS of (D.9) as follows:

$$\begin{aligned}&\sum_{\mathcal{P}^c(h)} r(m,n)\Delta_q(m,n) \\ &\quad = \quad \sum_{\mathcal{P}_\lambda^c} r(m,n)\Delta_q(m,n) + \sum_{\mathcal{P}^c(h)\setminus\mathcal{P}_\lambda^c} r(m,n)\Delta_q(m,n) \\ &\quad < \quad \sum_{\mathcal{P}_\lambda^c} r(m,n)\Delta_q(m,n) + p(\lambda+h) \sum_{\mathcal{P}^c(h)\setminus\mathcal{P}_\lambda^c} r(m,n) \\ &\quad = \quad p(\lambda+g^*) \sum_{\mathcal{P}_\lambda^c} r(m,n) - (d(0,0)+\lambda) \\ &\quad\quad\quad + p(\lambda+h) \sum_{\mathcal{P}^c(h)\setminus\mathcal{P}_\lambda^c} r(m,n),\end{aligned} \tag{D.10}$$

where, for the inequality, we used (D.7) and for (D.10), we have substituted the value for the quantity from (D.8). We may alternatively write the RHS of (D.9) as:

$$\begin{aligned}&p(\lambda+g(h)) \sum_{\mathcal{P}^c(h)} r(m,n) - (d(0,0)+\lambda) \\ &\quad = \quad p(\lambda+g(h)) \Big(\sum_{{\mathcal{P}_\lambda}^c} r(m,n) + \sum_{\mathcal{P}^c(h)\setminus{\mathcal{P}_\lambda}^c} r(m,n) \Big) \\ &\quad\quad\quad - (d(0,0)+\lambda).\end{aligned} \tag{D.11}$$

Now comparing (D.10) and (D.11) and rearranging, we may write:

$$p(g(h)-g^*) \sum_{{\mathcal{P}_\lambda}^c} r(m,n) < p(h-g(h)) \sum_{\mathcal{P}^c(h)\setminus{\mathcal{P}_\lambda}^c} r(m,n) \tag{D.12}$$

Now $\sum_{\mathcal{P}^c(h)\setminus{\mathcal{P}_\lambda}^c} r(m,n) = 0$ if and only if $\mathcal{P}^c(h)\setminus{\mathcal{P}_\lambda}^c = \varnothing$, i.e., $\mathcal{P}(h) = \mathcal{P}_\lambda$. In this case we get $g(h) = g^* < h$. On the other hand, if $\sum_{\mathcal{P}^c(h)\setminus{\mathcal{P}_\lambda}^c} r(m,n) > 0$, since $g^* \leq g(h)$, from the inequality (D.12), we conclude that $h > g(h)$. □

Appendix E. Proof of Lemma 5

Proof. 1) Note first that $h^{(k)} \geq g^*$ for $k \geq 1$ because $h^{(k)} = g(h^{(k-1)}) \geq g^*$. Then, for $k \geq 1$, we have either $h^{(k)} = g^*$ or $h^{(k)} > g^*$. In the first case $h^{(k+1)} = g(h^{(k)}) = g(g^*) = g^* = h^{(k)}$ and we can stop, whereas in the second case, from Lemma 3 we have $h^{(k+1)} = g(h^{(k)}) < h^{(k)}$.

2) From (15), $h_2 > h_1$ implies $\mathcal{P}^c(h_1) \subseteq \mathcal{P}^c(h_2)$. Hence, as $\{h^{(k)}\}_{k \geq 1}$ is non-increasing (from Part 1)), $\{\mathcal{P}^c(h^{(k)})\}_{k \geq 1}$ is also non-increasing.

Suppose $\mathcal{P}^c(h^{(k+1)}) = \mathcal{P}^c(h^{(k)})$ then $g(h^{(k+1)}) = g(h^{(k)}) = h^{(k+1)}$ (second equality is by the definition of $\{h^{(k)}\}$), which implies $h^{(k+1)} = g^*$ (since $g(\cdot)$ has a unique fixed point, see Lemma 4). Thus, $\mathcal{P}^c(h^{(k+1)}) = {\mathcal{P}_\lambda}^c$. $\square$

References

[1] P. Mondal, K. P. Naveen, A. Kumar, Optimal Deployment of Impromptu Wireless Sensor Networks, in: Proc. of the IEEE National Conference on Communications (NCC), Kharagpur, India, 2012.

[2] C. Fischer, H. Gellersen, Location and Navigation Support for Emergency Responders: A Survey, IEEE Pervasive Computing 9 (1) (2010) 38–49.

[3] A. Howard, M. J. Matarić, S. Sukhat Gaurav, An Incremental Self-Deployment Algorithm for Mobile Sensor Networks, Kluwer Autonomous Robots 13 (2) (2002) 113–126. doi:10.1023/A:1019625207705.

[4] G. Loukas, S. Timotheou, E. Gelenbe, Robotic Wireless Network Connection of Civilians for Emergency Response Operations, in: Proc. of the IEEE International Symposium on Computer and Information Sciences (ISCIS), Istanbul, Turkey, 2008.

[5] D. Naudts, S. Bouckaert, J. Bergs, A. Schouttcet, C. Blondia, I. Moerman, P. Demeester, A Wireless Mesh Monitoring and Planning Tool for Emergency Services, in: Proc. of the IEEE Workshop on End-to-End Monitoring Techniques and Services (E2EMON), Munich, Germany, 2007.

[6] M. R. Souryal, J. Geissbuehler, L. E. Miller, N. Moayeri, Real-Time Deployment of Multihop Relays for Range Extension, in: Proc. of the ACM International Conference on Mobile Systems, Applications and Services (MobiSys), San Juan, Puerto Rico, 2007.

[7] J. Q. Bao, W. C. Lee, Rapid deployment of wireless ad hoc backbone networks for public safety incident management, in: Proc. IEEE Globecom, 2007, pp. 1217–1221.

[8] M. R. Souryal, A. Wapf, N. Moayeri, Rapidly-Deployable Mesh Network Testbed, in: Proc. of the IEEE Conference on Global Telecommunications (GLOBECOM), Honolulu, Hawai, USA, 2009.

[9] T. Aurisch, J. Tölle, Relay Placement for Ad-hoc Networks in Crisis and Emergency Scenarios, in: Proc. of the Information Systems and Technology Panel (IST) Symposium, NATO Science and Technology Organization, Bucharest, Romania, 2009.

[10] H. Liu, J. Li, Z. Xie, S. Lin, K. Whitehouse, J. A. Stankovic, D. Siu, Automatic and Robust Breadcrumb System Deployment for Indoor Firefighter Applications, in: Proc. of the ACM International Conference on Mobile Systems, Applications and Services (MobiSys), San Francisco, California, USA, 2010.

[11] R. Srivastava, A. Kumar, Performance Analysis of Beacon-Less IEEE 802.15.4 Multi-Hop Networks, in: Proc. of the IEEE International Conference on Communication Systems and Networks (COMSNETS), Bangalore, India, 2012.

[12] A. Bhattacharya, S. M. Ladwa, R. Srivastava, A. Mallya, A. Rao, D. G. R. Sahib, S. Anand, A. Kumar, Smartconnect: A system for the design and deployment of wireless sensor networks, in: Communication Systems and Networks (COMSNETS), 2013 Fifth International Conference on, IEEE, 2013, pp. 1–10.

[13] P. Mondal, Optimal Deployment of Impromptu Wireless Sensor Networks, Master's thesis, Indian Institute of Science, Bangalore (2011).

[14] A. Chattopadhyay, M. Coupechoux, A. Kumar, Measurement based impromptu deployment of a multi-hop wireless relay network, arXiv preprint arXiv:1301.3302.

[15] Chipcon AS, Cc2420 2.4 ghz ieee 802.15. 4/zigbee-ready rf transceiver data sheet (rev. 1.3) http://www. chipcon. com/files, CC2420 Data Sheet 1 (3).

[16] M. J. Miller, N. H. Vaidya, Minimizing energy consumption in sensor networks using a wakeup radio, in: Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE, Vol. 4, IEEE, 2004, pp. 2335–2340.

[17] L. Gu, J. A. Stankovic, Radio-triggered wake-up capability for sensor networks., in: IEEE Real-Time and Embedded Technology and Applications Symposium, 2004, pp. 27–37.

[18] H. Cao, K. W. Parker, A. Arora, O-MAC: a receiver centric power management protocol, in: Network Protocols, 2006. ICNP'06. Proceedings of the 2006 14th IEEE International Conference on, IEEE, 2006, pp. 311–320.

[19] R. Chen, Solution of minimax problems using equivalent differentiable functions, Computers & mathematics with applications 11 (12) (1985) 1165–1169.

[20] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[21] E. Altman, Constrained Markov Decision Processes, Chapman and Hall (CRC), 1999.

[22] D. P. Bertsekas, Dynamic Programming and Optimal Control, Vol-II, 3rd Edition, Athena Scientific, Belmont, Massachusetts, 1995.