Sequential Decision Algorithms for Measurement-Based Impromptu Deployment of a Wireless Relay Network along a Line

Arpan Chattopadhyay, Marceau Coupechoux, Anurag Kumar

Sequential Decision Algorithms for Measurement-Based Impromptu Deployment of a Wireless Relay Network Along a Line

Arpan Chattopadhyay, Marceau Coupechoux, and Anurag Kumar, *Fellow, IEEE*

***Abstract*—We are motivated by the need, in some applications, for impromptu or as-you-go deployment of wireless sensor networks. A person walks along a line, starting from a sink node (e.g., a base-station), and proceeds towards a source node (e.g., a sensor) which is at an *a priori* unknown location. At equally spaced locations, he makes link quality measurements to the previous relay, and deploys relays at some of these locations, with the aim to connect the source to the sink by a multihop wireless path. In this paper, we consider two approaches for impromptu deployment: (i) the deployment agent can only move forward (which we call *a pure as-you-go* approach), and (ii) the deployment agent can make measurements over several consecutive steps before selecting a placement location among them (the *explore-forward* approach). We consider a very light traffic regime, and formulate the problem as a Markov decision process, where the trade-off is among the power used by the nodes, the outage probabilities in the links, and the number of relays placed per unit distance. We obtain the structures of the optimal policies for the *pure as-you-go* approach as well as for the *explore-forward* approach. We also consider natural heuristic algorithms, for comparison. Numerical examples show that the explore-forward approach significantly outperforms the pure as-you-go approach in terms of network cost. Next, we propose two learning algorithms for the explore-forward approach, based on Stochastic Approximation, which asymptotically converge to the set of optimal policies, without using any knowledge of the radio propagation model. We demonstrate numerically that the learning algorithms can converge (as deployment progresses) to the set of optimal policies reasonably fast and, hence, can be practical model-free algorithms for deployment over large regions. Finally, we demonstrate the end-to-end traffic carrying capability of such networks via field deployment.**

***Index Terms*—As-you-go deployment, impromptu wireless networks, measurement based deployment, relay placement.**

Manuscript received February 25, 2015; revised September 07, 2015; accepted October 20, 2015; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Weber. Date of publication November 17, 2015; date of current version October 13, 2016. This work was supported by a Department of Electronics and Information Technology (DeitY, India) and NSF (USA) funded project on Wireless Sensor Networks for Protecting Wildlife and Humans, the Indo-French Centre for Promotion of Advance Research (IFCPAR), and the Department of Science and Technology (DST, India) via J. C. Bose Fellowship. The contents of this paper have been arXived at http://arxiv.org/abs/1502.06878.

A. Chattopadhyay and A. Kumar are with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560012, India (e-mail: arpanc.ju@gmail.com; anurag@ece.iisc.ernet.in).

M. Coupechoux is with Telecom ParisTech and CNRS LTCI, Department Informatique et Reseaux, 75013 Paris, France (e-mail: marceau.coupechoux@telecom-paristech.fr).

All appendices of this paper are provided as supplementary downloadable material available online at http://ieeexplore.ieee.org.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

I. INTRODUCTION

A WIRELESS sensor network (WSN) typically comprises sensor nodes (sources of measurements), a base station (or sink), and wireless relays for multihop communication between the sources and the sink. There are situations in which a WSN needs to be deployed (i.e., the relays and the sensors need to be placed) in an *impromptu or as-you-go* fashion. One such situation is in emergencies, e.g., situational awareness networks deployed by first-responders such as fire-fighters or anti-terrorist squads. As-you-go deployment is also of interest when deploying multihop wireless networks for sensor-sink interconnection over large terrains, such as forest trails (see [2] for an application of multi-hop WSNs in wildlife monitoring, and [3, Section 5] for application of WSN in forest fire detection), where it may be difficult to make exhaustive measurements at all possible deployment locations before placing the relay nodes. As-you-go deployment would be particularly useful when the network is temporary and needs to be quickly redeployed at a different place (e.g., to monitor a moving phenomenon such as groups of wildlife).[1]

Our work is motivated by the need for as-you-go deployment of a WSN over large terrains, such as forest trails, where planned deployment (requiring exhaustive measurements over the deployment region) would be time consuming and difficult. Abstracting the above-mentioned problems, we consider the problem of deployment of relay nodes along a line, between a sink node (e.g., the WSN base-station) and a source node (e.g., a sensor) (see Fig. 1), where a *single deployment agent* (the person who is carrying out the deployment) starts from the sink node, places relay nodes along the line, and places the source node where required. In applications, the location at which sensor placement is required might only be discovered as the deployment agent walks (e.g., in an animal monitoring application, by finding a concentration of pugmarks, or a watering hole).

In the perspective of an optimal *planned* deployment, we would need to place relay nodes at *all* potential locations (for example, with reference to Fig. 1, this would mean placing relays at all the four dots in between the source and the sink) and measure the qualities of all possible links in order to decide where to place the relays. This approach would provide the global optimal solution, but the time and effort required might not be acceptable in the applications mentioned earlier. With *impromptu* deployment, the next relay placement locations depend on the radio link qualities to the previously placed nodes;

[1]In remote places, cellular network coverage may not be available or practicable. Hence, a multi-hop WSN is required for monitoring purposes.

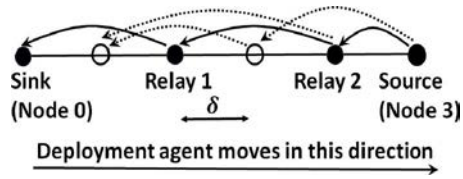

Fig. 1. A wireless relay network, placed along a line, connecting a source to a sink. The dots (filled and unfilled) denote potential locations for node placement, and are successively δ meters apart. The deployed network comprises two relays (filled dots) placed at two of the potential locations; the solid arrows show the path from the source to the sink. *The dotted arrows show some more possible links between pairs of potential locations*.

these link qualities and also the source location are discovered as the agent walks along the line. Such an approach requires fewer measurements compared to planned deployment, but, in general, is suboptimal.

In this paper, we mathematically formulate the problems of impromptu deployment along a line as *optimal sequential decision problems*. The cost of a deployment is evaluated as a linear combination of three components: the sum transmit power along the path, the sum outage probability along the path, and the number of relays deployed; we provide a motivation for this cost structure. We formulate relay placement problems to minimize the expected average cost per-step. Our channel model accounts for path-loss, shadowing, and fading.

We explore deployment with two approaches: (i) the *pure as-you-go* approach and (ii) the *explore-forward* approach. In the pure as-you-go approach, the deployment agent can only move forward; this approach is a necessity if the deployment needs to be quick. Due to shadowing, the path-loss over a link of a given length is random, and a more efficient deployment can be expected if link quality measurements at several locations along the line are compared and an optimal choice is made among these; we call this approach *explore-forward*. Explore-forward would require the deployment agent to retrace his steps; but this might provide a good compromise between deployment speed and deployment efficiency.

We formulate each of these problems as a Markov decision process (MDP), obtain the optimal policy structures, illustrate their performance numerically and compare with reasonable heuristics. Next, we propose several learning algorithms and prove that each of them asymptotically converges to the optimal policy if we seek to minimize the long run average cost per unit distance. We also demonstrate the convergence rate of the learning algorithms via numerical exploration. Finally, we demonstrate the end-to-end traffic carrying capability of such networks via field deployment.

A. Related Work

Until recently, problems of impromptu deployment of wireless networks have been addressed primarily by heuristics and by experimentation. Howard *et al.*, in [4], provide heuristic algorithms for incremental deployment of sensors in order to cover the deployment area; their problem is related to that of self-deployment of autonomous robot teams. Souryal *et al.*, in [5], address the problem of impromptu wireless network deployment by experimental study of indoor RF link quality variation; a similar approach is taken in [6]also. The authors of [7] describe a *breadcrumbs* system for aiding firefighting inside buildings. Their work addresses the same class of problems as ours, with the requirement that the deployment agent has to stay connected to k previously placed nodes in the deployment process. Their work considers the trade-off between link qualities and the deployment rate, but does not provide any optimality guarantee of their deployment schemes. Their next work [8] provides a reliable multiuser *breadcrumbs* system. Bao and Lee, in [9], study the scenario where a group of first-responders, starting from a command centre, enter a large area where there is no communication infrastructure, and as they walk they place relays at suitable locations in order to stay connected among themselves and with the command centre. However, these approaches are based on heuristic algorithms, rather than on rigorous formulations; hence they do not provide any provable performance guarantee.

In our work we have formulated impromptu deployment as a sequential decision problem, and have derived optimal deployment policies. Recently, Sinha *et al.* ([10]) have provided an algorithm based on an MDP formulation in order to establish a multi-hop network between a sink and an unknown source location, by placing relay nodes along a random lattice path. Their model uses a deterministic mapping between power and wireless link length, and, hence, does not consider statistical variability (due to shadowing) of the transmit power required to maintain the link quality over links having the same length. The statistical variation of link qualities over space requires measurement-based deployment, in which the deployment agent makes placement decisions at a point based on the measurement of the power required to establish a link (with a given quality) to the previously placed node.

We view the current paper as a continuation of our papers [11] (which provides the first theoretical formulation of measurement-based impromptu deployment) and [12] (which provides field deployment results using our algorithms).

B. Organization

The system model and notation have been described in Section II. Impromptu deployment with a pure as-you-go approach has been discussed in Section III. Section IV presents our work on the explore-forward approach. A numerical comparison between these two approaches are made in Section V. Section VI and Section VII describe the learning algorithms for the explore-forward approach approach. Numerical results are provided in Section VIII on the rate of convergence of the learning algorithms. Experimental results demonstrating the traffic carrying capability of the deployed networks are provided in Section IX, followed by the conclusion.

II. System Model and Notation

The line is discretized into steps of length δ (Fig. 1), starting from the sink. Each point, located at a distance of an integer multiple of δ from the sink, is considered to be a potential location where a relay can be placed. As the *single* deployment agent walks along the line, at each step or at some subset of steps, he measures the link quality from the current location to the previous node; these measurements are used to decide the location and transmit power of the next relay.

As shown in Fig. 1, the sink is called Node 0, the relay closest to the sink is called Node 1, and the relays are enumerated as nodes $\{1, 2, 3, \cdots\}$ as we walk away from the sink. The link whose transmitter is Node i and receiver is Node j is called

link (i, j). A generic link is denoted by e. The length of each link is an integer multiple of δ.

A. Channel Model and Outage Probability

We consider the usual aspects of path-loss, shadowing, and fading to model the wireless channel. The received power of a packet (say the k-th packet, $k \geq 1$) in a particular link (i.e., a transmitter-receiver pair) of length r is given by:

$$P_{rcv,k} = P_T c \left(\frac{r}{r_0}\right)^{-\eta} H_k W \tag{1}$$

where P_T is the transmit power, c is the path-loss at the reference distance r_0, η is the path-loss exponent, H_k denotes the fading random variable seen by the k-th packet (e.g., it is an exponentially distributed random variable for the Rayleigh fading model), and W denotes the shadowing random variable. H_k captures the variation of the received power over time, and it takes independent values over different coherence times.

The path-loss between a transmitter and a receiver at a given distance can have a large spatial variability around the mean path-loss (averaged over fading), as the transmitter is moved over different points at the same distance from the receiver; this is called shadowing. Shadowing is usually modeled as a log-normally distributed, random, multiplicative path-loss factor; in dB, shadowing is distributed with values of standard deviation as large as 8 to 10 dB. Also, shadowing is spatially uncorrelated over distances that depend on the sizes of the objects in the propagation environment (see [13]); *our measurements in a forest-like region of our Indian Institute of Science (IISc) campus established log-normality of the shadowing and gave a shadowing decorrelation distance of 6 meters (see* [12]). In this paper, *we assume that the shadowing at any two different links in the network are independent*, i.e., $W_{(e_1)}$ *is independent of* $W_{(e_2)}$ *for* $e_1 \neq e_2$. This is a reasonable assumption if δ is chosen to be at least the decorrelation distance (see [13]) of the shadowing. Thus, from our experiments in the forest-like region in the IISc campus, we can safely assume independent shadowing at different potential locations if δ is greater than 6 m. In this paper, W is assumed to take values from a set $\mathcal{W}$. We will denote by $p_W(w)$ the probability mass function or probability density function of W, depending on whether $\mathcal{W}$ is a countable set or an uncountable set (e.g., log-normal shadowing).

A link is considered to be in *outage* if the received signal power (RSSI) drops (due to fading) below $P_{rcv-min}$ (e.g., below -88 dBm, a figure that we have obtained via experimentation for the popular TelosB "motes," see [14]). Since practical radios can only be set to transmit at a finite set of power levels, the transmit power of each node can be chosen from a discrete set, $\mathcal{S} := \{P_1, P_2, \cdots, P_M\}$, where $P_1 \leq P_2 \leq \cdots \leq P_M$. For a link of length r, a transmit power γ and any particular realization of shadowing $W = w$, the outage probability is denoted by $Q_{out}(r, \gamma, w)$, which is increasing in r and decreasing in γ, w (according to (1)).

$Q_{out}(r, \gamma, w)$ depends on the fading statistics. For a link with shadowing realization w, if the transmit power is γ, the received power of a packet will be $P_{rcv} = \gamma c(r/r_0)^{-\eta} wH$. Outage is the event $P_{rcv} \leq P_{rcv-min}$. If H is exponentially distributed with mean 1 (i.e., for Rayleigh fading), then we have, $Q_{out}(r, \gamma, w) = \mathbb{P}(\gamma c(r/r_0)^{-\eta} wH \leq P_{rcv-min}) = 1 - e^{-(P_{rcv-min}(r/r_0)^{\eta}/\gamma cw)}$. The outage probability of a randomly chosen link of given length and given transmit power is a random variable, where the randomness comes from shadowing W. *Outage probability can be measured by sending a sufficiently large number of packets over a link and calculating the percentage of packets whose RSSI is below* $P_{rcv-min}$.

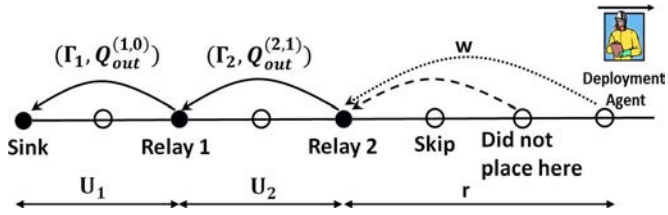

Fig. 2. Illustration of pure as-you-go deployment with $A = 1$ and $B = 3$. In this "snap-shot" of the deployment process, the deployment agent has already placed Relay 1 and Relay 2 at distances U_1 and U_2, has set their transmit powers to Γ_1 and Γ_2, thereby achieving outage probabilities $Q_{out}^{(1,0)}$ and $Q_{out}^{(2,1)}$ (links shown by solid arrows). Having placed Relay 2, he skips the next location (since $A = 1$); based on measurements made at the next location (dashed arrow), the algorithm advises him to not place a relay and move on. The diagram shows the agent in the process of evaluating the next location at $r = 3\delta$ distance from Relay 2 (dotted arrow). Based on these measurements, the deployment agent will decide whether to place a relay at $r = 3\delta$; if a relay is not placed here, it must be placed at the next location, since $B = 3$.

B. Deployment Process and Related Notation

In this paper, we consider two approaches for deployment.

Pure as-you-go Deployment: After placing a relay, the agent skips the next A steps, and sequentially measures the outage probabilities from locations $(A + 1), (A + 2), \cdots, (A + B)$ to the previously placed node, at all transmit power levels $\gamma \in \mathcal{S}$. As the agent explores the locations $(A + 1), \cdots, (A + B - 1)$ and makes link quality measurements,[2] at each step he decides whether to place a relay there, and if the decision is to place a relay, then he also decides the transmit power for the placed relay. This has been depicted in Fig. 2. In this process, if he has walked $(A + B)$ steps away from the previous relay, or if he encounters the source location, then he must place a node. A and B will be fixed before deployment begins. □

Explore-Forward Deployment: After placing a node, the deployment agent skips the next A locations $(A \geq 0)$ and measures the outage probabilities to the previous node from locations $(A + 1), \cdots, (A + B)$, at each power level from the set $\mathcal{S}$. Then, based on these $B|\mathcal{S}|$ measurements[3] of the outage probability values, he places the relay at location $u^* \in \{A + 1, \cdots, A + B\}$, sets its transmit power to $\gamma^* \in \mathcal{S}$, and repeats the same process for placing the next relay. This procedure is illustrated in Fig. 3. If the source location is encountered within $(A+B)$ steps from the previous node, then the source is placed. □

Choice of A and B: If the propagation environment is very good, or if we need to place a limited number of relays over a long line, it is very unlikely that a relay will be placed within the first few locations from the previous node. In such cases, we can skip measurements at locations $1, 2, \cdots, A$ and make measurements from locations $(A+1), \cdots, (A+B)$. In general, the

[2]At a distance r from the previous node, he measures the outage probabilities $\{Q_{out}(r, \gamma, w)\}_{\gamma \in \mathcal{S}}$ from the current location to the previous node, where w is the realization of the shadowing in the link being evaluated.

[3]Let us denote by w_u the realization of shadowing in the potential link between the u-th location (starting from the previously placed node) and the previous node (see Fig. 3). The agent measures the outage probabilities $\{Q_{out}(u, \gamma, w_u)\}_{A+1 \leq u \leq A+B, \gamma \in \mathcal{S}}$ in order to make a placement decision.

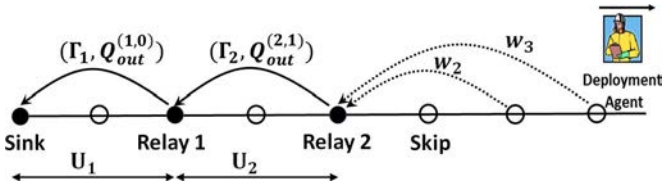

Fig. 3. Illustration of explore-forward deployment with $A = 1$ and $B = 2$. In this "snap-shot" of the deployment process, the deployment agent has already placed Relay 1 and Relay 2 at distances U_1 and U_2, has set their transmit powers to Γ_1 and Γ_2, thereby achieving outage $Q_{out}^{(1,0)}$ and $Q_{out}^{(2,1)}$ (links shown by solid arrows). Having placed Relay 2, he skips the next location (since $A = 1$). The agent then evaluates the next two locations (dotted arrows) ($B = 2$). Then, based on the measurements at these two locations, the algorithm determines which of them to place the relay at and which power level to use.

choice of A and B will depend on the constraints and requirements for the deployment. Larger A will result in faster exploration, but very large A will result in very high outage in each link. For a fixed A, a large B results in more measurements, but we can expect a better performance.

C. Traffic Model

In order to develop the problem formulation, we assume that the traffic is so low that there is only one packet in the network at a time; we call this the "lone packet model." Hence, there are no simultaneous transmissions to cause interference. This permits us to easily write down the communication cost on a path over the deployed relays. However, this assumption does not trivialize the deployment problem, since the deployment must still take into account the stochastic shadowing and fading in the links, and the effects of these factors on the number of nodes deployed and the powers they use.

The lone packet traffic model is realistic for sensor networks that carry low duty cycle measurements, or just carry an occasional alarm packet. For example, recently there has been an effort to design passive infra-red (PIR) sensor platforms that can detect intrusion of a human or animal, and also can classify whether the intruder is a human or an animal ([15]). The data rate generated by such a platform deployed in a forest will be very low. The authors in [2, Section 3.2] use only a 1.1% duty cycle for a multi-hop wireless sensor network used for the purpose of wildlife monitoring. The sensors gather data from RFID collars on the animals; hence, the traffic to be supported by the network is light. Lone packet model is also realistic for condition monitoring/industrial telemetry applications ([16]) as well, where the time between successive measurements is very large. Infrequent data model is common in machine-to-machine communication ([17]). Table 1 and Table 3 of [18] illustrate sensors whose sampling rate and the size of the sampled data packets are small; it shows data rate requirement as small as several bytes per second for habitat monitoring.

Even though the network is designed for the lone packet traffic, it will be able to carry some amount of positive traffic. See Section IX for experimental evidence of this claim; a five-hop line network deployed using one algorithm proposed in this paper, over a 500 m long line in a forest-like environment, was able to carry 127 byte packets at a rate of 4 packets per second, with end-to-end packet loss probability less than 1%, which is sufficient for the applications mentioned above.

Lone packet model is also valid when interference-free communication is achieved via multi-channel access. Recently there have been efforts to use multiple channels available in 802.15.4 radio in a network; see [19]–[22]. In a line topology, this reduces to frequency reuse after certain hops, which, in turn, mitigates interference in the network. Thus, as with the lone packet assumption, the availability of multiple channels, and appropriate channel allocation over the network, eliminates the need to optimize over link schedules.

It has been proved that design with the lone packet model can be the starting point for a design with desired positive traffic (see [23]). Network design for carrying a given positive traffic rate is left as a future research work.

D. Network Cost Structure

In this section we develop the cost that we use to evaluate the performance of a given deployment policy. Given the current location of the deployment agent with respect to the previous relay, and given the measurements made to the previous relay, a policy will provide the placement decision (in the case of pure as-you-go deployment, whether or not to place the relay, and if place then at what power, and in the case of explore-forward deployment, where among the B locations to place the relay and at which power).

Let us denote the number of placed relays up to x steps (i.e., $x\delta$ meters) from the sink by $N_x(\leq x)$; define $N_0 = 0$. Since deployment decisions are based on measurements to already placed relays, and since the path-loss over a link is a random variable (due to shadowing), we see that $\{N_x\}_{x\geq 1}$ is a random process. In this paper we have assumed that each node forwards each packet to the immediately previously placed relay (e.g., with reference to Fig. 1, the source forwards all packets to Relay 2, which, in turn, forwards all packets to Relay 1, etc.). See [11] for the considerably more complex possibility of relay skipping while forwarding packets.

When the node i is placed, the deployment policy also prescribes the transmit power that this node should use, say, Γ_i; then the outage probability over the link $(i, i-1)$, so created, is denoted by $Q_{out}^{(i,i-1)}$ (see Fig. 2 and Fig. 3). We evaluate the cost of the deployed network, up to $x\delta$ distance, as a linear combination of three cost measures:

(i) The number of relays placed, i.e., N_x.

(ii) The sum outage, i.e., $\sum_{i=1}^{N_x} Q_{out}^{(i,i-1)}$. The motivation for this measure is that, for small values of Q_{out}, the sum-outage is approximately the probability that a packet sent from the point x to the source encounters an outage along the path from the point x back to the sink.

(iii) The sum power over the hops, i.e., $\sum_{i=1}^{N_x} \Gamma_i$.

These three costs are combined into one cost measure by combining them linearly and taking expectation (under a policy π), as follows:

$$\mathbb{E}_\pi \left(\sum_{i=1}^{N_x} \Gamma_i + \xi_{out} \sum_{i=1}^{N_x} Q_{out}^{(i,i-1)} + \xi_{relay} N_x \right) \qquad (2)$$

The multipliers $\xi_{out} \geq 0$ and $\xi_{relay} \geq 0$ can be viewed as capturing the emphasis we wish to place on the corresponding measure of cost. For example, a large value of ξ_{out} will aim for a network deployment with smaller end-to-end expected outage. We can view ξ_{relay} as the cost of placing a relay.

A Motivation for the Sum Power Objective: In case all the nodes have *wake-on radios*, the nodes normally stay in sleep mode, and each sleeping node draws a very small current from the battery (see [24]). When a node has a packet, it sends a

wake-up tone to the intended receiver. The receiver wakes up and the sender transmits the packet. The receiver sends an ACK packet in reply. Clearly, the energy spent in transmission and reception of data packets governs the lifetime of a node, given that the ACK size is negligible. We assume that a fixed modulation scheme is used, so that the transmission bit rate over all links is the same (e.g., in IEEE 802.15.4 radios, that are commonly used for sensor networking, the standard modulation scheme provides a bit rate of 250 Kbps). We also assume a fixed packet length. Let t_p be the transmission duration of a packet over a link, and suppose that the node i $(1 \leq i \leq N_x)$ uses power Γ_i during transmission. Let P_r denote the packet reception power expended in the electronics at any receiving node. If the packet generation rate ζ at the source is very small, the lifetime of the k-th node $(1 \leq k \leq N_x)$ is $T_k := E/\zeta(\Gamma_k + P_r)t_p$ seconds (E is the total energy in a fresh battery). Hence, the rate at which we have to replace the batteries in the network from the sink up to distance x steps is given by $\sum_{k=1}^{N_x}(1/T_k) = \sum_{k=1}^{N_x}(\zeta(\Gamma_k + P_r)t_p/E)$. The term $\zeta P_r t_p/E$ can be absorbed into ξ_{relay}. Hence, the battery depletion rate is proportional to $\sum_{k=1}^{N_x}\Gamma_k$.

E. Deployment Objective

We assume that the distance L to the source from the sink is *a priori* unknown, and its distribution is also unknown. Hence, we assume that $L = \infty$ (deployment along a line of infinite length) and develop deployment policies that seek to minimize the average cost per step. *This setting can be useful in practice when L is large* (e.g., *a long forest trail). Also, if we seek to create networks along multiple trails in a forest, and if deployment is done serially along multiple trails, then this is effectively equivalent to deployment along a single long line, provided that the trails have statistically identical radio propagation environment. Note that, in case we deploy serially along multiple lines but use this formulation, it means that we seek to optimize the per-step cost averaged over multiple lines*.

1) Unconstrained Problem: Motivated by the cost structure and the $L = \infty$ model, we seek to solve the following:

$$\inf_{\pi\in\Pi} \limsup_{x\to\infty} \frac{\mathbb{E}_\pi \sum_{i=1}^{N_x}\left(\Gamma_i + \xi_{out}Q_{out}^{(i,i-1)} + \xi_{relay}\right)}{x} \quad (3)$$

where π is a placement *policy*, and Π is the set of all possible placement policies (to be formalized later). We formulate (3) as a long-term average cost Markov decision process (MDP).

2) Connection to a Constrained Problem: Note that, (3) is the relaxed version of the following constrained problem where we seek to minimize the mean power per step subject to a constraint on the mean outage per step and a constraint on the mean number of relays per step:

$$\inf_{\pi\in\Pi} \limsup_{x\to\infty} \frac{\mathbb{E}_\pi \sum_{i=1}^{N_x}\Gamma_i}{x}$$
$$s.t. \ \limsup_{x\to\infty} \frac{\mathbb{E}_\pi \sum_{i=1}^{N_x} Q_{out}^{(i,i-1)}}{x} \leq \overline{q} \text{ and } \limsup_{x\to\infty} \frac{\mathbb{E}_\pi N_x}{x} \leq \overline{N} \quad (4)$$

The following standard result tells us how to choose the *Lagrange multipliers* ξ_{out} and ξ_{relay} (see [25], Theorem 4.3):

Theorem 1: Consider the constrained problem (4). If there exists a pair $\xi_{out}^* \geq 0$, $\xi_{relay}^* \geq 0$ and a policy π^* such that π^* is the optimal policy of the unconstrained problem (3) under $(\xi_{out}^*, \xi_{relay}^*)$ and the constraints in (4) are met with equality under π^*, then π^* is an optimal policy for (4) also. □

III. Pure As-You-Go Deployment

A. Markov Decision Process (MDP) Formulation

Here we seek to solve problem (3), for the pure as-you-go approach. When the agent is r steps away from the previous node $(A + 1 \leq r \leq A + B)$, he measures the outage probabilities $\{Q_{out}(r, \gamma, w)\}_{\gamma\in\mathcal{S}}$ on the link from the current location to the previous node, where w is the realization of the shadowing random variable in the link being evaluated. He uses the knowledge of r and the outage probabilities to decide whether to place a node at his current location, and what transmit power $\gamma \in \mathcal{S}$ to use if he places a relay. In this case, we formulate the impromptu deployment problem as a Markov Decision Process (MDP) with state space $\{A + 1, \cdots, A + B\} \times \mathcal{W}$. At state $(r, w), (A + 1) \leq r \leq (A + B - 1), w \in \mathcal{W}$, the action is either to place a relay and select a transmit power, or not to place. When $r = A + B$, the only feasible action is to place and select a transmit power $\gamma \in \mathcal{S}$. If, at state (r, w), a relay is placed and it is set to use transmit power γ, a hop-cost of $\gamma + \xi_{out}Q_{out}(r, \gamma, w) + \xi_{relay}$ is incurred.[4]

A deterministic Markov policy π is a sequence of mappings $\{\mu_k\}_{k\geq 1}$ from the state space to the action space, and it is called a stationary policy if $\mu_k = \mu$ for all k. Given the state (i.e., the measurements), the placement decision is made according to the policy.

B. Formulation for $L \sim Geometric(\theta)$

Under the pure as-you-go approach, we will first minimize the expected total cost for $L \sim Geometric(\theta)$, and then take $\theta \to 0$; this approach provides the policy structure for the average cost problem (see [26], Chapter 4).

In the $L \sim Geometric(\theta)$ case, the deployment process regenerates (probabilistically) after placing a relay, because of the memoryless property of the geometric distribution, and because of the fact that deployment of a new node will involve measurement of qualities of new links not measured before, and the new links have i.i.d. shadowing independent of the previously measured links. The (special) state of the system at such regeneration points is denoted by $\mathbf{0}$ (apart from the states of the form (r, w)). When the source is placed at the end of the line, the process terminates. Suppose N is the (random) number of relays placed, and node $N + 1$ is the source node (as shown in Fig. 1). We first seek to solve the following:

$$\min_{\pi\in\Pi} \mathbb{E}_\pi \left(\sum_{i=1}^{N+1} \Gamma_i + \xi_{out} \sum_{i=1}^{N+1} Q_{out}^{(i,i-1)} + \xi_{relay}N \right) \quad (5)$$

We will first investigate this approach assuming finite $\mathcal{W}$.

C. Bellman Equation

Let us denote the optimal expected cost-to-go at state (r, w) and at state $\mathbf{0}$ be $J(r, w)$ and $J(\mathbf{0})$ respectively. Note that here we have an infinite horizon total cost MDP with a finite state

[4]We have taken (r, w) as a typical state for simplicity of representation; so long as the channel model given by (1) is valid, we can also take $(r, \{Q_{out}(r, \gamma, w)\}_{\gamma\in\mathcal{S}})$ as a typical state. This happens because the cost of an action depends on the state (r, w) only via the outage probabilities.

space and finite action space. The assumption P of Chapter 3 in [26] is satisfied, since the single-stage costs are nonnegative. Hence, by the theory developed in [26], we can focus on the class of stationary deterministic Markov policies.

By Proposition 3.1.1 of [26], the optimal value function $J(\cdot)$ satisfies the Bellman equation which is given by, for all $(A+1) \leq r \leq (A+B-1)$,

$$\begin{aligned}
&J(r,w) \\
&= \min \Big\{ \min_{\gamma \in \mathcal{S}} (\gamma + \xi_{out} Q_{out}(r,\gamma,w)) + \xi_{relay} + J(\mathbf{0}), \\
&\qquad \theta \mathbb{E}_W \min_{\gamma \in \mathcal{S}} (\gamma + \xi_{out} Q_{out}(r+1,\gamma,W)) \\
&\qquad + (1-\theta) \mathbb{E}_W J(r+1,W) \Big\}, \\
&J(A+B,w) \\
&= \min_{\gamma \in \mathcal{S}} (\gamma + \xi_{out} Q_{out}(A+B,\gamma,w) + \xi_{relay}) + J(\mathbf{0}) \\
&J(\mathbf{0}) \\
&= \sum_{k=1}^{A+1} (1-\theta)^{k-1} \theta \mathbb{E}_W \min_{\gamma \in \mathcal{S}} (\gamma + \xi_{out} Q_{out}(k,\gamma,W)) \\
&\quad + (1-\theta)^{A+1} \mathbb{E}_W J(A+1,W) \qquad (6)
\end{aligned}$$

These equations are understood as follows. If the current state is (r,w), $(A+1) \leq r \leq (A+B-1)$ and the line has not ended yet, we can either place a relay and set its transmit power to $\gamma \in \mathcal{S}$, or we may not place. If we place, the cost $\min_{\gamma \in \mathcal{S}} (\gamma + \xi_{out} Q_{out}(r,\gamma,w) + \xi_{relay})$ is incurred at the current step, and the cost-to-go from there is $J(\mathbf{0})$. If we do not place a relay, the line will end with probability θ in the next step, in which case a cost $\mathbb{E}_W \min_{\gamma \in \mathcal{S}} (\gamma + \xi_{out} Q_{out}(r+1,\gamma,W))$ will be incurred. If the line does not end in the next step, the next state will be a random state $(r+1,W)$ and a mean cost of $\mathbb{E}_W J(r+1,W)$ will be incurred. At state $(A+B,w)$ the only possible decision is to place a relay. At state $\mathbf{0}$, the deployment agent starts walking until he encounters the source location or location $(A+1)$; if the line ends at step $k, 1 \leq k \leq A+1$ (with probability $(1-\theta)^{k-1}\theta$), a cost of $\mathbb{E}_W \min_{\gamma \in \mathcal{S}} (\gamma + \xi_{out} Q_{out}(k,\gamma,W))$ is incurred. If the line does not end within $(A+1)$ steps (this event has probability $(1-\theta)^{A+1}$), the next state will be $(A+1,W)$.

D. Value Iteration

The value iteration for (5) is obtained by replacing $J(\cdot)$ in (6) by $J^{(k+1)}(\cdot)$ on the L.H.S (left hand side) and by $J^{(k)}(\cdot)$ on the R.H.S (right hand side), and by taking $J^{(0)}(\cdot) = 0$ for all states. The standard MDP theory says that $J^{(k)}(\cdot) \uparrow J(\cdot)$ for all states as $k \to \infty$.

E. Policy Structure: OptAsYouGo Algorithm

Lemma 1: $J(r,w)$ is increasing in r, ξ_{out} and ξ_{relay}, decreasing in w, and jointly concave in ξ_{out} and ξ_{relay}. $J(\mathbf{0})$ is increasing and jointly concave in ξ_{out} and ξ_{relay}.

Proof: See Appendix A. ∎

Next, we propose an optimal algorithm OptAsYouGo (Optimal algorithm with pure As-You-Go approach).

Algorithm 1: (OptAsYouGo Algorithm): At state (r,w) (where $A+1 \leq r \leq A+B-1$), place a relay if and only if $\min_{\gamma \in \mathcal{S}} (\gamma + \xi_{out} Q_{out}(r,\gamma,w)) \leq c_{th}(r)$, where $c_{th}(r) := \theta \mathbb{E}_W \min_{\gamma \in \mathcal{S}} (\gamma + \xi_{out} Q_{out}(r+1,\gamma,W)) + (1-\theta) \mathbb{E}_W J(r+1,W) - (\xi_{relay} + J(\mathbf{0}))$ is a threshold increasing in r. If the decision is to place a relay, the optimal power to be selected is given by $\arg\min_{\gamma \in \mathcal{S}} (\gamma + \xi_{out} Q_{out}(r,\gamma,w))$. At state $(A+B,w)$, select transmit power $\arg\min_{\gamma \in \mathcal{S}} (\gamma + \xi_{out} Q_{out}(A+B,\gamma,w))$. □

Theorem 2: Under the pure as-you-go approach, Algorithm 1 provides the optimal policy for Problem (3).

Proof: See Appendix A. ∎

Remark: The trade-off in the impromptu deployment problem is that if we place relays far apart, the cost due to outage increases, but the cost of placing the relays decreases. The intuition behind the threshold structure of the policy is that if at distance r we get a good link with the combination of power and outage less than a threshold, then we should accept that link because moving forward is unlikely to yield a better link. $c_{th}(r)$ is increasing in r. Since $Q_{out}(r,\gamma,w)$ is increasing in r for any γ, w, and since shadowing is i.i.d across links, the probability of a link (to the previous node) having desired QoS decreases as we move away from the previous node. Hence, the optimal policy will try to place relays as soon as possible if r is large, and this explains why $c_{th}(r)$ is increasing in r. Note that the threshold $c_{th}(r)$ does not depend on w, due to the fact that shadowing is i.i.d. across links.

F. Computation of the Optimal Policy

Let us write $V(r) := \mathbb{E}_W J(r,W) = \sum_{w \in \mathcal{W}} p_W(w) J(r,w)$, and $V(\mathbf{0}) := J(\mathbf{0})$. Also, for each stage $k \geq 0$ of the value iteration, define $V^{(k)}(r) := \mathbb{E}_W J^{(k)}(r,W)$ and $V^{(k)}(\mathbf{0}) := J^{(k)}(\mathbf{0})$. Multiplying both sides of the value iteration by $p_W(w)$ and summing over $w \in \mathcal{W}$, we obtain an iteration in terms of $V^{(k)}(\cdot)$ and this iteration does not involve $J^{(k)}(\cdot)$. Since $J^{(k)}(r,w) \uparrow J(r,w)$ for each r, w and $J^{(k)}(\mathbf{0}) \uparrow J(\mathbf{0})$ as $k \uparrow \infty$, we can argue that $V^{(k)}(r) \uparrow \mathbb{E}_W J(r,W) = V(r)$ for all r (by Monotone Convergence Theorem) and $V^{(k)}(\mathbf{0}) \uparrow J(\mathbf{0}) = V(\mathbf{0})$. Then we can compute $c_{th}(r)$ by knowing $V(\cdot)$ itself (see the expression of $c_{th}(r)$ in Algorithm 1); we need not keep track of the cost-to-go values $J^{(k)}(r,w)$ for each state (r,w), at each stage k. Here we simply need to keep track of $V^{(k)}(\cdot)$. Similar iterations were proposed in [11] (Section III-A-5).

G. Average Cost Problem: Optimality of OptAsYouGo

Note that the problem (5) can be considered as an infinite horizon discounted cost problem with discount factor $(1-\theta)$. Hence, keeping in mind that we have finite state and action spaces, we observe that for the discount factor sufficiently close to 1, i.e., for θ sufficiently close to 0, *the optimal policy for problem in* (5) *is optimal for the problem in* (3) (see [26, Proposition 4.1.7]). In particular, the optimal average cost per step with pure as-you-go approach, λ^*, is given by $\lambda^* = \lim_{\theta \to 0} \theta J_\theta(\mathbf{0})$ (see [26, Section 4.1.1]), where $J_\theta(\mathbf{0})$ is the optimal cost for problem (5) under pure as-you-go with the probability of the line ending in the next step is θ.

In case $\mathcal{W}$ is a Borel subset of $\mathbb{R}$, we still have a finite action space, and bounded, nonnegative cost per step. By [27, Theorem 5.5.4], one can show that the optimal average cost per step is again $\lambda^* = \lim_{\theta \to 0} \theta J_\theta(\mathbf{0})$. As $\theta \downarrow 0$, we will obtain a sequence of optimal policies (i.e., mappings from the state space to the action space), and a limit point of them will be an average cost optimal policy.

H. HeuAsYouGo: A Suboptimal Pure As-You-Go Heuristic

Algorithm 2: (HeuAsYouGo): The power used by the relays is set to a fixed value. At each potential location, the deployment agent checks whether the outage to the previous relay meets a certain predetermined target with this fixed transmit power level. After placing a relay, the next relay is placed at the last location where the target outage is met; or place at the $(A+1)$-st location (after the previously placed relay) in the unlikely situation where the target outage is violated in the $(A+1)$-st location itself. If the agent reaches the $(A+B)$-th step and if all previous locations violate the outage target, he must place the next relay at step $(A+B)$. □

HeuAsYouGo is a modified version of the heuristic deployment algorithm proposed in [5]. HeuAsYouGo is not exactly a pure as-you-go algorithm since it sometimes requires the agent to move one step back in case the outage target is violated.

IV. Explore Forward Deployment

A. Semi-Markov Decision Process (SMDP) Formulation

Let us recall explore-forward deployment from Section II-B; we denote by w_u the realization of shadowing in the potential link between the u-th location (starting from the previously placed node) and the previous node (see Fig. 3). The agent measures the outage probabilities $\{Q_{out}(u,\gamma,w_u)\}_{A+1\le u\le A+B,\gamma\in\mathcal{S}}$ from locations $(A+1)$, $\cdots,(A+B)$ at all available transmit power levels from the set $\mathcal{S}$, in order to make a placement decision.

Here we seek to solve the unconstrained problem (3). We formulate our problem as a Semi-Markov Decision Process (SMDP) with state space $\mathcal{W}^B$ and action space $\{A+1,A+2,\cdots,A+B\}\times\mathcal{S}$. The vector $\underline{w} := (w_{A+1},w_{A+2},\cdots,w_{A+B})$, i.e., the shadowing from B locations, is the state in our SMDP. In the state $\underline{w}$, an action $(u,\gamma)\in\{A+1,A+2,\cdots,A+B\}\times\mathcal{S}$ is taken where u is the distance of the next relay (from the previous relay) that would be placed and γ is the transmit power that this relay will use. In this case, a hop-cost of $\gamma+\xi_{out}Q_{out}(u,\gamma,w_u)+\xi_{relay}$ is incurred. After placing a relay, the next state becomes $\underline{w}' := (w'_{A+1},w'_{A+2},\cdots,w'_{A+B})$ with probability $g(\underline{w}') := \prod_{r=A+1}^{A+B} p_{W_r}(w'_r)$ (since shadowing is i.i.d. across links).

Let us denote, by the vector $\underline{W}(k)$, the (random) state at the k-th decision instant, and by $\mu_k(\underline{W}(k))$ the action at the k-th decision instant. For a deterministic Markov policy $\pi := \{\mu_k\}_{k\ge1}$, let us define the functions $\mu_k^{(1)}:\mathcal{W}^B\to\{A+1, A+2,\cdots,A+B\}$ and $\mu_k^{(2)}:\mathcal{W}^B\to\mathcal{S}$ as follows: if $\mu_k(\underline{w})=(u,\gamma)$, then $\mu_k^{(1)}(\underline{w})=u$ and $\mu_k^{(2)}(\underline{w})=\gamma$.

B. Policy Structure: Algorithm OptExploreLim

Note that, $\underline{W}(k)$ is i.i.d across $k, k\ge1$. The state space is a Borel space and the action space is finite. The hop cost and hop length (in number of steps) are uniformly bounded across all state-action pairs. Hence, we can work with stationary deterministic policies (see [28] for finite state space, i.e., finite $\mathcal{W}$, and [29] for a general Borel state space, i.e., when $\mathcal{W}$ is a Borel set). Under our current scenario, the optimal average cost per step, λ^*, exists (in fact, the limit exists) and is same for all states $\underline{w}\in\mathcal{W}^B$. *For simplicity, we work with finite $\mathcal{W}$, but the policy structure holds for Borel state space also.*

We next present a deployment algorithm called "OptExploreLim," an optimal algorithm for limited exploration.

Algorithm 3: (OptExploreLim Algorithm:): In the state $\underline{w}$ which is captured by the measurements $\{Q_{out}(u,\gamma,w_u)\}$ for $A+1\le u\le A+B$, $\gamma\in\mathcal{S}$, place the new relay according to the stationary policy μ^* as follows:

$$\mu^*(\underline{w}) = \arg\min_{u,\gamma}\left(\gamma+\xi_{out}Q_{out}(u,\gamma,w_u)+\xi_{relay}-\lambda^*u\right) \tag{7}$$

where λ^* (or $\lambda^*(\xi_{out},\xi_{relay})$) is the optimal average cost per step for the Lagrange multipliers (ξ_{out},ξ_{relay}). □

Theorem 3: The policy μ^* given by Algorithm 3 is optimal for the problem (3) under the explore-forward approach.

Proof: The optimality equation for the SMDP is given by (see [28], Equation 7.2.2):

$$v^*(\underline{w}) = \min_{u,\gamma}\left\{\gamma+\xi_{out}Q_{out}(u,\gamma,w_u)+\xi_{relay} -\lambda^*u+\sum_{\underline{w}'\in\mathcal{W}^B} g(\underline{w}')v^*(\underline{w}')\right\} \tag{8}$$

$v^*(\underline{w})$ is the optimal differential cost corresponding to state $\underline{w}$. The structure of the optimal policy is obvious from (8), since $\sum_{\underline{w}'\in\mathcal{W}^B} g(\underline{w}')v^*(\underline{w}')$ does not depend on (u,γ). ■

Later we will also use the notation π^* or $\pi^*(\xi_{out},\xi_{relay})$ to denote the OptExploreLim policy under the pair (ξ_{out},ξ_{relay}), since here $\pi^*=\{\mu^*,\mu^*,\mu^*,\cdots\}$.

Remark 1: The same optimal policy structure will hold for a Borel state space, by the theory presented in [29].

Remark 2: The optimal decision depends on state $\underline{w}$ only via the outage probabilities which can be easily measured.

Remark 3: For an action (u,γ), a cost $(\gamma+\xi_{out}Q_{out}(u,\gamma,w_u)+\xi_{relay})$ will be incurred. On the other hand, λ^*u is the reference cost over u steps. The policy minimizes the difference between these two for each link.

Remark 4: The policy requires the deployment agent to know λ^*, and computation of λ^* will require perfect knowledge of propagation environment (e.g., the path-loss exponent η in (1), the distribution of shadowing, etc.); see Section IV-C.

Theorem 4: $\lambda^*(\xi_{out},\xi_{relay})$ is jointly concave, increasing and continuous in ξ_{out} and ξ_{relay}.

Proof: See Appendix B. ■

Let us consider a sub-class of stationary deployment policies (parameterized by $\lambda\ge0$, $\xi_{out}\ge0$ and $\xi_{relay}\ge0$) given by:

$$\mu(\underline{w}) = \arg\min_{u,\gamma}\left(\gamma+\xi_{out}Q_{out}(u,\gamma,w_u)+\xi_{relay}-\lambda u\right) \tag{9}$$

where λ is not necessarily equal to $\lambda^*(\xi_{out},\xi_{relay})$.

Under the class of policies given by (9), let $(U_k,\Gamma_k,Q_{out}^{(k,k-1)}), k\ge1$, denote the sequence of inter-node distances, transmit powers and link outage probabilities that the optimal policy yields during the deployment process. By the assumption of i.i.d. shadowing across links, it follows that $(U_k,\Gamma_k,Q_{out}^{(k,k-1)}), k\ge1$, is an i.i.d. sequence.

Let $\overline{\Gamma}(\lambda,\xi_{out},\xi_{relay})$, $\overline{Q}_{out}(\lambda,\xi_{out},\xi_{relay})$ and $\overline{U}(\lambda,\xi_{out},\xi_{relay})$ denote the mean power per link, mean outage per link and mean placement distance (in steps) respectively, under the policy given by (9), where λ is not

necessarily equal to $\lambda^*(\xi_{out}, \xi_{relay})$. Also, let $\overline{\Gamma}^*(\xi_{out}, \xi_{relay})$, $\overline{Q}^*_{out}(\xi_{out}, \xi_{relay})$ and $\overline{U}^*(\xi_{out}, \xi_{relay})$ denote the optimal mean power per link, the optimal mean outage per link and the optimal mean placement distance (in steps) respectively, under the OptExploreLim algorithm (i.e., policy $\pi^*(\xi_{out}, \xi_{relay})$ when λ in (9) is replaced by $\lambda^*(\xi_{out}, \xi_{relay})$). By the Renewal-Reward theorem, the optimal mean power per step, the optimal mean outage per step, and the optimal mean number of relays per step are given by $\overline{\Gamma}^*(\xi_{out}, \xi_{relay})/\overline{U}^*(\xi_{out}, \xi_{relay})$, $\overline{Q}^*_{out}(\xi_{out}, \xi_{relay})/\overline{U}^*(\xi_{out}, \xi_{relay})$ and $1/\overline{U}^*(\xi_{out}, \xi_{relay})$.

Theorem 5: For a given ξ_{out}, the mean number of relays per step under the OptExploreLim algorithm (Algorithm 3), $1/\overline{U}^*(\xi_{out}, \xi_{relay})$, decreases with ξ_{relay}. Similarly, for a given ξ_{relay}, the mean outage probability per step, $\overline{Q}^*_{out}(\xi_{out}, \xi_{relay})/\overline{U}^*(\xi_{out}, \xi_{relay})$, decreases with ξ_{out} under the optimal policy.

Proof: See Appendix B. ■

Remark: The proof of Theorem 5 is quite general; the results hold for the pure as-you-go approach also.

Theorem 6: For Problem (3), under the optimal policy (with explore-forward approach) characterized by λ^* (i.e., under the OptExploreLim algorithm), we have $\mathbb{E}_{\underline{W}} \min_{u,\gamma}(\gamma + \xi_{out} Q_{out}(u, \gamma, W_u) + \xi_{relay} - \lambda^* u) = 0$.

Proof: See Appendix B. ■

C. Policy Computation

We adapt a policy iteration (from [28]) based algorithm to calculate λ^*. The algorithm generates a sequence of stationary policies $\{\mu_k\}_{k\geq 1}$ (note that the notation μ_k was used for a different purpose in Section IV-A; here each μ_k is a stationary, deterministic, Markov policy), such that for any $k \geq 1$, $\mu_k(\cdot) : \mathcal{W}^B \to \{A+1, \cdots, A+B\} \times \mathcal{S}$ maps a state into some action. Define the sequence $\{\mu_k^{(1)}, \mu_k^{(2)}\}_{k\geq 1}$ of functions as follows: if $\mu_k(\underline{w}) = (u, \gamma)$, then $\mu_k^{(1)}(\underline{w}) = u$ and $\mu_k^{(2)}(\underline{w}) = \gamma$.

Algorithm 4: The policy iteration algorithm is as follows:

Step 1) (Initialization): Start with an initial policy μ_0.

Step 2) (Policy Evaluation): Calculate the average cost λ_k corresponding to the policy μ_k, for $k \geq 0$. λ_k is equal to the quantity (by the Renewal Reward Theorem) given by the expression at the bottom of the page.

Step 3) (Policy Improvement): Find a new policy μ_{k+1} by solving the following:

$$\mu_{k+1}(\underline{w}) = \arg\min_{(u,\gamma)} (\gamma + Q_{out}(u, \gamma, w_u) + \xi_{relay} - \lambda_k u) \tag{10}$$

If μ_k and μ_{k+1} are same (i.e., if $\lambda^{(k-1)} = \lambda_k$), then stop and declare $\mu^* = \mu_k$, $\lambda^* = \lambda_k$. Otherwise, go to Step 1. □

Remark: By the theory in [28], *this policy iteration will converge (to λ^*) in a finite number of iterations, for finite state and action spaces*. For a general Borel state space (e.g., for log-normal shadowing), only asymptotic convergence to λ^* can be guaranteed.

Computational Complexity: The finite state space has cardinality $|\mathcal{W}|^B$. Then, $O(|\mathcal{W}|^B)$ addition operations are required to compute λ_k from the policy evaluation step. However, careful manipulation leads to a drastic reduction in this computational requirement, as shown by the following theorem.

Theorem 7: In the policy evaluation step in Algorithm 4, we can reduce the number of computations in each iteration from $|\mathcal{W}|^B$ to $O(B^2 M^2 |\mathcal{W}|^2)$.

Proof: See Appendix B. ■

D. HeuExploreLim: An Intuitive but Suboptimal Heuristic

A natural heuristic for (3) under the explore-forward approach is the following HeuExploreLim Algorithm (Heuristic Algorithm for Limited Explore-Forward):

Algorithm 5: (HeuExploreLim Algorithm): Under the explore-forward setting as discussed in Section IV, at state $\underline{w}$, make the decision according to the following rule:

$$(u^*, \gamma^*) = \arg\min_{u,\gamma} \frac{\gamma + \xi_{out} Q_{out}(u, \gamma, w_u) + \xi_{relay}}{u}$$

□

Under any stationary deterministic policy μ, let us denote the cost of a link by C_μ (a random variable) and the length of a link by U_μ (under any stationary deterministic policy μ, the deployment process regenerates at the placement points).

Lemma 2: HeuExploreLim solves $\inf_\mu \mathbb{E}_\mu(C_\mu/U_\mu)$.

Proof: See Appendix B. ■

Remark: This heuristic is not optimal. Our optimal policy given in Theorem (3) solves $\inf_\mu(\mathbb{E}_\mu(C_\mu)/\mathbb{E}_\mu(U_\mu))$. However, HeuExploreLim solves $\inf_\mu \mathbb{E}_\mu(C_\mu/U_\mu)$, which is, in general, different from $\inf_\mu(\mathbb{E}_\mu(C_\mu)/\mathbb{E}_\mu(U_\mu))$. Note that $\mathbb{E}_\mu(C_\mu/U_\mu) = \mathbb{E}_\mu(C_\mu)/\mathbb{E}_\mu(U_\mu)$ *if and only if* the variance of U_μ is zero. But this does not happen due to the variability in shadowing over space.

V. Comparison Between Explore-Forward and Pure As-You-Go Approaches

Let us denote the optimal average cost per step (for a given ξ_{out} and ξ_{relay}) under the explore-forward and pure as-you-go approaches by λ^*_{ef} and λ^*_{ayg}.

Theorem 8: $\lambda^*_{ef} \leq \lambda^*_{ayg}$.

Proof: See Appendix C. ■

Next, we numerically compare various deployment algorithms, in order to select the best algorithm for deployment.

A. Parameter Values Used in the Numerical Comparisons

We consider deployment for a given ξ_{out} and a given ξ_{relay}, for the objective in (3). We provide numerical results for deployment with iWiSe motes ([30]) (based on the Texas Instrument (TI) CC2520 which implements the IEEE 802.15.4 PHY in the 2.4 GHz ISM band, yielding a bit rate of 250 Kbps, with

$$\frac{\xi_{relay} + \sum_{\underline{w}} g(\underline{w}) \left(\mu_k^{(2)}(\underline{w}) + \xi_{out} Q_{out}\left(\mu_k^{(1)}(\underline{w}), \mu_k^{(2)}(\underline{w}), w_{\mu_k^{(1)}(\underline{w})}\right)\right)}{\sum_{\underline{w}} g(\underline{w}) \mu_k^{(1)}(\underline{w})}.$$

a CSMA/CA medium access control (MAC)) equipped with 9 dBi antennas. The set of transmit power levels $\mathcal{S}$ is taken to be $\{-18, -7, -4, 0, 5\}$ dBm, which is a subset of the transmit power levels available in the chosen device. For the channel model as in (1), our measurements in a forest-like environment inside the Indian Institute of Science Campus gave path-loss exponent $\eta = 4.7$ and $c = 10^{0.17}$ (i.e., 1.7 dB); see [12]. Shadowing W was found to be log-normal; $W = 10^{Y/10}$ with $Y \sim \mathcal{N}(0, \sigma^2)$, where $\sigma = 7.7$ dB. Shadowing decorrelation distance was found to be 6 meters. Fading is assumed to be Rayleigh; $H \sim Exponential(1)$.

We define outage to be the event when the received signal power of a packet falls below $P_{rcv-min} = 10^{-9.7}$ mW (i.e., -97 dBm); for a commercial implementation of the PHY/MAC of IEEE 802.15.4, -97 dBm received power corresponds to a 2% packet loss probability for 127 byte packets for iWiSe motes, as per our measurements.

We consider deployment along a line with step size $\delta = 20$ meters, $A = 0$, $B = 5$. Given $A = 0$, we chose B is the following way. Define a link to be good if its outage probability is less than 3%, and choose B to be the largest integer such that the probability of finding a good link of length $B\delta$ is more than 20%, when the highest transmit power is used (this will ensure that the agent does not measure very long links having poor outage probabilities). For the parameters $\eta = 4.7$, $c = 10^{0.17}$, $\sigma = 7.7$ dB, and 5 dBm transmit power, B turned out to be 5. If B is increased further, the probability of getting a good link will be very small. □

B. Numerical Comparison Among Deployment Policies

Assuming these parameter values, we computed (by MATLAB) the mean power per step (in mW), mean outage per step, mean placement distance (in steps), mean cost per step and mean number of measurements made per step, for the four deployment algorithms presented so far. The results are shown in Table I. In order to make a fair comparison, *we used the mean power per node for OptAsYouGo as the fixed node transmit power for HeuAsYouGo, and the mean outage per link of OptAsYouGo as the target outage for HeuAsYouGo*. The mean number of measurements per step is defined as the ratio of the mean number of links evaluated for deployment of one node and the mean placement distance (in steps). The numerator of this ratio is $B = 5$ for explore-forward algorithms (since $A = 0$). OptAsYouGo makes one measurement per step, but HeuAsYouGo makes more than one measurements per step since the agent often evaluates a bad link, takes one step back and places the relay.[5]

We notice that the average per-step cost (COST in Table I) of OptExploreLim (OEL) is the least. OEL uses the least mean power per step (POW column), places nodes the widest apart (DIST column), and the mean outage per step (OUT column) is second to lowest. On the other hand, OEL requires about twice as many measurements per step as compared to OptAsYouGo.[6] Hence, we can conclude that the algorithms based on the explore-forward approach significantly outperform the algorithms based on the pure-as-you-go approach, at the cost of slightly more measurements per step. Hence, for applications that do not require very rapid deployment, such as deployment along a long forest trail for wildlife monitoring, explore-forward is a better approach to take. Thus, for the learning algorithms presented later, we will consider only the explore-forward approach. However, under the requirement of fast deployment (e.g., emergency deployment by first responders), pure as-you-go or deployment without measurements (as in [10]) might be more suitable.

TABLE I
NUMERICAL COMPARISON AMONG VARIOUS ALGORITHMS FOR $\xi_{out} = 100$ AND $\xi_{relay} = 1$. ABBREVIATIONS: OEL—OPTEXPLORELIM, HEL—HEUEXPLORELIM, OAYG—OPTASYOUGO, HAYG—HEUASYOUGO. POW—MEAN POWER (IN mW UNIT) PER STEP, OUT—MEAN OUTAGE PER STEP, DIST—MEAN PLACEMENT DISTANCE, COST—MEAN COST PER STEP, MEAS—MEAN NUMBER OF MEASUREMENTS PER STEP

Algorithm	POW (mW)	OUT	DIST (steps)	COST	MEAS
OEL	0.1955	0.001969	2.2859	0.8312	2.1873
HEL	0.2432	0.002507	2.673	0.8684	1.8706
OAYG	0.2904	0.003607	1.5	1.265	1
HAYG	0.3318	0.001752	1.313	1.267	1.6969

VI. OPTEXPLORELIMLEARNING: LEARNING WITH EXPLORE-FORWARD, FOR GIVEN ξ_{out} AND ξ_{relay}

Based on the discussion in Section V, we proceed, in the rest of this paper, with developing learning algorithms based on the policy OptExploreLim (to solve problem (3)). We observe that the optimal policy (given by Algorithm 3) can be completely specified by the optimal average cost per step λ^*, for given values of ξ_{out} and ξ_{relay}. But the computation of λ^* requires policy iteration. Policy iteration requires the channel model parameters η and σ, and it is computationally intensive. In practice, these parameters of the channel model might not be available. Under this situation, the agent measures $\{Q_{out}(u, \gamma, w_u) : A + 1 \leq u \leq A + B, \gamma \in \mathcal{S}\}$ before deploying each relay, but he has to *learn* the optimal average cost per step in the process of deployment, and, use the corresponding updated policy each time he places a new relay. In order to address this requirement, we propose an algorithm which will maintain a running estimate of λ^*, and update it each time a relay is placed. The algorithm is motivated by the theory of Stochastic Approximation (see [31]), and it uses, as input, the measurements made for each placement, in order to improve the estimate of λ^*. We prove that, as the number of deployed relays goes to infinity, the running estimate of average network cost per step converges to λ^* almost surely.

After the deployment is over, let us denote the length, transmit power and outage values of the link between node k and node $(k-1)$ by u_k, γ_k and $Q_{out}^{(k,k-1)}$. After placing the $(k-1)$-st node, we will place node k, and consequently u_k, γ_k and $Q_{out}^{(k,k-1)}$ will be decided by the following algorithm.

Algorithm 6: (OptExploreLimLearning): Let $\lambda^{(k)}$ be the estimate of the optimal average cost per step after placing the k-th relay (sink is node 0), and let $\lambda^{(0)}$ be the initial estimate. In the process of placing relay $(k+1)$, if the measured outage probabilities are $\{Q_{out}(u, \gamma, w_u) : A + 1 \leq u \leq A + B, \gamma \in \mathcal{S}\}$, then place relay $(k+1)$ using the following policy:

$$(u_{k+1}, \gamma_{k+1}) = \arg\min_{u,\gamma} \left(\gamma + \xi_{out} Q_{out}(u, \gamma, w_u) + \xi_{relay} - \lambda^{(k)} u \right)$$

[5]For planned deployment, we will have to evaluate all possible potential links; from each potential location, we need to measure link quality to $B = 5$ preceding potential locations, which is not feasible.

[6]A more detailed comparison among the algorithms can be found in Appendix C, along with elaborate discussion.

After placing relay $(k+1)$, update $\lambda^{(k)}$ as follows (using the measurements made in the process of placing relay $(k+1)$):

$$\begin{aligned}&\lambda^{(k+1)}\\&=\lambda^{(k)}+a_{k+1}\min_{u,\gamma}\Big(\gamma+\xi_{out}Q_{out}(u,\gamma,w_u)+\xi_{relay}-\lambda^{(k)}u\Big)\\&=\lambda^{(k)}+a_{k+1}\left(\gamma_{k+1}+\xi_{out}Q_{out}^{(k+1,k)}+\xi_{relay}-\lambda^{(k)}u_{k+1}\right)\end{aligned} \tag{11}$$

$\{a_k\}_{k\geq 1}$ is a decreasing sequence such that $a_k > 0\ \forall k \geq 1$, $\sum_k a_k = \infty$ and $\sum_k a_k^2 < \infty$. One example is $a_k = 1/k$. □

Theorem 9: If we employ Algorithm 6 in the deployment process, we will have $\lambda^{(k)} \to \lambda^*$ almost surely.

Proof: By Theorem 6, under OptExploreLim, we have $\mathbb{E}_{\underline{W}} \min_{u,\gamma}(\gamma + \xi_{out}Q_{out}(u,\gamma,W_u) + \xi_{relay} - \lambda^* u) = 0$; this leads to the stochastic approximation update in Algorithm 6. The detailed proof can be found in Appendix D. ■

While Algorithm 6 utilizes the general stochastic approximation update, Algorithm 7 ensures that the iterate $\lambda^{(k)}$ is the actual average network cost per step up to the k-th relay.

Algorithm 7: Start with any $\lambda^{(0)} > 0$. Let, for $k \geq 1$, $\lambda^{(k)}$ be the average cost per step for the portion of the network already deployed between the sink and the k-th relay, i.e.,

$$\lambda^{(k)} = \frac{\sum_{i=1}^{k}\left(\gamma_i + \xi_{out}Q_{out}^{(i,i-1)} + \xi_{relay}\right)}{\sum_{i=1}^{k} u_i}$$

Place the $(k+1)$-st relay according to the following policy:

$$\begin{aligned}&(u_{k+1},\gamma_{k+1})\\&\quad=\arg\min_{u,\gamma}\left(\gamma+\xi_{out}Q_{out}(u,\gamma,w_u)+\xi_{relay}-\lambda^{(k)}u\right)\end{aligned}$$

□

Corollary 1: Under Algorithm 7 in the deployment process, we will have $\lambda^{(k)} \to \lambda^*$ almost surely.

Proof: See Appendix D. ■

VII. OptExploreLimAdaptiveLearning With Constraints on Outage Probability and Relay Placement Rate

In Section VI, we provided a stochastic approximation algorithm for relay deployment, with given multipliers ξ_{out} and ξ_{relay}, without knowledge of the propagation parameters. Let us recall that Theorem 1 tells us how to choose the Lagrange multipliers ξ_{out} and ξ_{relay} (if they exist) in (3) in order to solve the problem given in (4). However, we need to know the radio propagation parameters (e.g., η and σ) in order to compute an optimal pair $(\xi_{out}^*, \xi_{relay}^*)$ (if it exists) so that both constraints in (4) are met with equality. In real deployment scenarios, these propagation parameters might not be known. Hence, in this section, we provide a sequential placement and learning algorithm such that, as the relays are placed, the placement policy iteratively converges to the set of optimal policies for the constrained problem displayed in (4). The policy is of the OptExploreLim type, and the cost of the deployed network converges to the optimal cost. We modify the OptExploreLimLearning algorithm so that a running estimate $(\lambda^{(k)}, \xi_{out}^{(k)}, \xi_{relay}^{(k)})$ gets updated each time a new relay is placed. The objective is to make sure that the running estimate $(\lambda^{(k)}, \xi_{out}^{(k)}, \xi_{relay}^{(k)})$ eventually converges to the set of optimal $(\lambda^*(\xi_{out},\xi_{relay}), \xi_{out}, \xi_{relay})$ tuples as the deployment progresses. Our approach is via two time-scale stochastic approximation (see [31, Chapter 6]).

A. OptExploreLim: Effect of Multipliers ξ_{out} and ξ_{relay}

Consider the constrained problem in (4) and its relaxed version in (3). We will seek a policy for the problem in (4) in the class of OptExploreLim policies (see (7)). Clearly, there exists at least one tuple $(\overline{q}, \overline{N})$ for which there exists a pair $\xi_{out}^* > 0, \xi_{relay}^* > 0$ such that, under the optimal policy $\pi^*(\xi_{out}^*, \xi_{relay}^*)$, both constraints are met with equality. In order to see this, choose any $\xi_{out} > 0, \xi_{relay} > 0$ and consider the corresponding optimal policy $\pi^*(\xi_{out}, \xi_{relay})$ (provided by OptExploreLim). Suppose that the mean outage per step and mean number of relays per step, under the policy $\pi^*(\xi_{out}, \xi_{relay})$, are q_0 and n_0, respectively. Now, if we set the constraints $\overline{q} = q_0$ and $\overline{N} = n_0$ in (4), we obtain one instance of such a tuple $(\overline{q}, \overline{N})$.

On the other hand, there exist $(\overline{q}, \overline{N})$ pairs which are not feasible. One example is the case $\overline{N} = 1/(A+B)$ (i.e., inter-node distance is always $(A+B)$), along with $\overline{q} < (\mathbb{E}_W Q_{out}(A+B, P_M, W)/(A+B))$, where P_M is the maximum available transmit power level at each node. In this case, the outage constraint cannot be satisfied while meeting the constraint on the mean number of relays per step, since even use of the highest transmit power P_M at each node will not satisfy the per-step outage constraint.

Definition 1: Let us denote the optimal mean power per step for problem (4) by γ^*, for a given $(\overline{q}, \overline{N})$. The set $\mathcal{K}(\overline{q}, \overline{N})$ is defined as follows:

$$\begin{aligned}\mathcal{K}(\overline{q},\overline{N}) := \{&(\lambda^*(\xi_{out},\xi_{relay}),\xi_{out},\xi_{relay}) :\\&\frac{\overline{\Gamma}^*(\xi_{out},\xi_{relay})}{\overline{U}^*(\xi_{out},\xi_{relay})} = \gamma^*, \frac{\overline{Q}_{out}^*(\xi_{out},\xi_{relay})}{\overline{U}^*(\xi_{out},\xi_{relay})} \leq \overline{q}\\&\frac{1}{\overline{U}^*(\xi_{out},\xi_{relay})} \leq \overline{N}, \xi_{out} \geq 0, \xi_{relay} \geq 0\Bigg\}\end{aligned}$$

where the optimal average cost per step of the unconstrained problem (3) under OptExploreLim is $\lambda^*(\xi_{out}, \xi_{relay})$. □

$\mathcal{K}(\overline{q}, \overline{N})$ can possibly be empty (in case $(\overline{q}, \overline{N})$ is not a feasible pair). *Hence, we make the following assumption which ensures the non-emptiness of* $\mathcal{K}(\overline{q}, \overline{N})$.

Assumption 1: The constraint parameters $\overline{q}$ and $\overline{N}$ in (4) are such that there exists at least one pair $\xi_{out}^* \geq 0, \xi_{relay}^* \geq 0$ for which $(\lambda^*(\xi_{out}^*, \xi_{relay}^*), \xi_{out}^*, \xi_{relay}^*) \in \mathcal{K}(\overline{q}, \overline{N})$. □

Remark: Assumption 1 implies that the constraints are consistent (in terms of achievability). If $\xi_{out}^* > 0, \xi_{relay}^* > 0$, it would imply that both of the constraints are active. If $\xi_{out}^* = 0$, it would imply that we can keep the mean outage per step strictly less than $\overline{q}$ by using the minimum available power at each node, while meeting the constraint on the relay placement rate. The optimal policy in Algorithm 3, under $\xi_{out} = 0$, will place relays with inter-relay distance $(A+B)$ steps, and use the minimum available power level at each node. $\xi_{out}^* = \infty$ implies that the outage constraint cannot be met even with the highest power level at each node, under the relay placement rate constraint. Similar arguments apply to ξ_{relay}^*. □

We now establish some structural properties of $\mathcal{K}(\overline{q}, \overline{N})$.

Theorem 10: If $\mathcal{K}(\overline{q}, \overline{N})$ is non-empty, then:

- Suppose that there exists $\xi_{out}^* > 0, \xi_{relay}^* > 0$ such that the policy $\pi^*(\xi_{out}^*, \xi_{relay}^*)$ satisfies both constraints in (4) with

equality. Then, there does not exist $\xi'_{out} \geq 0, \xi'_{relay} \geq 0$ satisfying (i) $(\lambda^*(\xi'_{out}, \xi'_{relay}), \xi'_{out}, \xi'_{relay}) \in \mathcal{K}(\overline{q}, \overline{N})$, and (ii) $(\overline{Q}^*_{out}(\xi'_{out}, \xi'_{relay})/\overline{U}^*(\xi'_{out}, \xi'_{relay})) < \overline{q}$ or $(1/\overline{U}^*(\xi'_{out}, \xi'_{relay})) < \overline{N}$.

- If there exists a $\xi'_{relay} \geq 0$ such that $(\lambda^*(0, \xi'_{relay}), 0, \xi'_{relay}) \in \mathcal{K}(\overline{q}, \overline{N})$, then, $\forall \xi_{relay} \geq 0$, we have $(\lambda^*(0, \xi_{relay}), 0, \xi_{relay}) \in \mathcal{K}(\overline{q}, \overline{N})$. □

Proof: See Appendix E, Section A. ■

Assumption 2: The shadowing random variable W has a continuous probability density function (p.d.f.) over $(0, \infty)$; for any $w \in (0, \infty)$, $\mathbb{P}(W = w) = 0$. One example could be log-normal shadowing. □

Theorem 11: Suppose that Assumption 2 holds. Under the OptExploreLim algorithm, the optimal mean power per step $\overline{\Gamma}^*(\xi_{out}, \xi_{relay})/\overline{U}^*(\xi_{out}, \xi_{relay})$, the optimal mean number of relays per step $1/\overline{U}^*(\xi_{out}, \xi_{relay})$ and the optimal mean outage per step $\overline{Q}^*_{out}(\xi_{out}, \xi_{relay})/\overline{U}^*(\xi_{out}, \xi_{relay})$, are continuous in ξ_{out} and ξ_{relay}.

Proof: See Appendix E, Section B. ■

Remark: Note that, by Theorem 11, we need not do any randomization (see [32] for reference) among deterministic policies in order to meet the constraints with equality.

B. OptExploreLimAdaptiveLearning Algorithm

Algorithm 8: This algorithm iteratively updates $\lambda^{(k)}, \xi^{(k)}_{out}, \xi^{(k)}_{relay}$ after each relay is placed. Let $(\lambda^{(k)}, \xi^{(k)}_{out}, \xi^{(k)}_{relay})$ be the iterates after placing the k-th relay (the sink is called node 0), and let $(\lambda^{(0)}, \xi^{(0)}_{out}, \xi^{(0)}_{relay})$ be the initial estimates. In the process of deploying the k-th relay, if the shadowing (which is measured indirectly only via $Q_{out}(u, \gamma, w_u)$ for $A + 1 \leq u \leq A + B$ and $\gamma \in \mathcal{S}$) is $\underline{w} = \{w_{A+1}, \cdots, w_{A+B}\}$, then place the k-th relay according to the following policy:

$$(u_k, \gamma_k) = \arg\min_{u,\gamma} \left(\gamma + \xi^{(k-1)}_{out} Q_{out}(u, \gamma, w_u) + \xi^{(k-1)}_{relay} - \lambda^{(k-1)} u\right) \tag{12}$$

After placing the k-th relay, let us denote the transmit power, distance (in steps) and outage probability from relay k to relay $(k-1)$ by γ_k, u_k and $Q_{out}(u_k, \gamma_k, w_{u_k})$. After placing the k-th relay, make the following updates (using the measurements made in the process of placing the k-th relay):

$$\begin{aligned}
\lambda^{(k)} &= \lambda^{(k-1)} + a_k \min_{u,\gamma} \Big(\gamma + \xi^{(k-1)}_{out} Q_{out}(u, \gamma, w_u) \\
&\qquad + \xi^{(k-1)}_{relay} - \lambda^{(k-1)} u\Big) \\
\xi^{(k)}_{out} &= \Lambda_{[0,A_2]} \left(\xi^{(k-1)}_{out} + b_k \left(Q_{out}(u_k, \gamma_k, w_{u_k}) - \overline{q} u_k\right)\right) \\
\xi^{(k)}_{relay} &= \Lambda_{[0,A_3]} \left(\xi^{(k-1)}_{relay} + b_k (1 - \overline{N} u_k)\right)
\end{aligned} \tag{13}$$

where $\Lambda_{[0,A_2]}(x)$ denotes the projection of x on the interval $[0, A_2]$. A_2 and A_3 need to be chosen carefully; the reason is explained in the discussion later in this section (along with a brief discussion on how A_2 and A_3 have to be chosen).

$\{a_k\}_{k\geq 1}$ and $\{b_k\}_{k\geq 1}$ are two decreasing sequences such that $a_k, b_k > 0$, $\forall k \geq 1$, $\sum_k a_k = \infty$, $\sum_k a_k^2 < \infty$, $\sum_k b_k = \infty$, $\sum_k b_k^2 < \infty$ and $\lim_{k\to\infty}(b_k/a_k) = 0$. In particular, we can use $a_k = C_1 k^{-n_1}$ and $b_k = C_2 k^{-n_2}$ where $C_1 > 0$, $C_2 > 0$, $1/2 < n_1 < n_2 \leq 1$. □

Note that, for $(\xi_{out}, \xi_{relay}) \in [0, A_2] \times [0, A_3]$, we have $0 < \lambda^*(\xi_{out}, \xi_{relay}) \leq (P_M + A_2 + A_3)$. Let us define the set $\hat{\mathcal{K}}(\overline{q}, \overline{N}) := \mathcal{K}(\overline{q}, \overline{N}) \cap ([0, (P_M + A_2 + A_3)] \times [0, A_2] \times [0, A_3])$ which is a subset of $\mathcal{K}(\overline{q}, \overline{N})$.

Theorem 12: Under Assumption 1, Assumption 2 and under proper choice of A_2 and A_3, the iterates $(\lambda^{(k)}, \xi^{(k)}_{out}, \xi^{(k)}_{relay})$ in Algorithm 8 converge almost surely to $\hat{\mathcal{K}}(\overline{q}, \overline{N})$ as $k \to \infty$.

Proof: See Appendix E, Section C. ■

Remark: Algorithm 8 induces a nonstationary policy. But Theorem 12 establishes that the sequence of policies generated by Algorithm 8 converges to the set of optimal stationary, deterministic policies (for problem (4)).

Discussion of Theorem 12:

(i) *Two timescales*: The update scheme (13) can be rewritten as a two-timescale stochastic approximation (see [31], Chapter 6). Note that, $\lim_{k\to\infty}(b_k/a_k) = 0$, i.e., ξ_{out} and ξ_{relay} are adapted in a *slower* timescale compared to λ (which is adapted in the *faster* timescale). The dynamics behaves as if ξ_{out} and ξ_{relay} are updated simultaneously in a slow outer loop, and, between two successive updates of ξ_{out} and ξ_{relay}, we update λ in an inner loop for a long time. Thus, the λ update equation views ξ_{out} and ξ_{relay} as quasi-static, while the ξ_{out} and ξ_{relay} update equations view the λ update equation as almost equilibrated.

(ii) *Structure of the iteration*: Note that, $(Q_{out}(u_k, \gamma_k, w_{u_k}) - \overline{q} u_k)$ is the excess outage compared to the allowed outage $\overline{q} u_k$ for the k-th link. If this quantity is positive (resp., negative), the algorithm increases (resp., decreases) ξ_{out} in order to reduce (resp., increase) the outage probability in subsequent steps. Similarly, if $u_k < (1/\overline{N})$, the algorithm increases ξ_{relay} in order to reduce the relay placement rate. The goal is to ensure $\lim_{k\to\infty}(\overline{Q}^*_{out}(\xi^{(k)}_{out}, \xi^{(k)}_{relay}) - \overline{q}\overline{U}^*(\xi^{(k)}_{out}, \xi^{(k)}_{relay})) = 0$ and $\lim_{k\to\infty}(1 - \overline{NU}^*(\xi^{(k)}_{out}, \xi^{(k)}_{relay})) = 0$. In the faster timescale, our aim is to ensure that $\lim_{k\to\infty} \mathbb{E}_{\underline{W}} \min_{u,\gamma}(\gamma + \xi^{(k)}_{out} Q_{out}(u, \gamma, W_u) + \xi^{(k)}_{relay} - \lambda^{(k)} u) = 0$.

(iii) *Outline of the proof*: The proof proceeds in five steps.
We first prove the almost sure boundedness of $\{\lambda^{(k)}\}_{k\geq 1}$.
Next, we prove that the difference between the sequences $\lambda^{(k)}$ and $\lambda^*(\xi^{(k)}_{out}, \xi^{(k)}_{relay})$ converges to 0 almost surely; this will prove the desired convergence in the faster timescale. This result has been proved using the theory in [31, Chapter 6] and Theorem 9.
In order to ensure boundedness of the slower timescale iterates, we have used the projection operation in the slower timescale. We pose the slower timescale iteration in the same form as a projected stochastic approximation iteration (see [33, Equation 5.3.1]).
In order to prove the desired convergence of the projected stochastic approximation, we show that our iteration satisfies certain conditions given in [33] (see [33, Theorem 5.3.1]).
Next, we argue (using Theorem 5.3.1 of [33]) that the slower timescale iterates converge to the set of stationary points of a suitable ordinary differential equation (o.d.e.). But, in general, a stationary point on the boundary of the

closed set $[0, A_2] \times [0, A_3]$ in the (ξ_{out}, ξ_{relay}) plane may not correspond to a point in $\mathcal{K}(\overline{q}, \overline{N})$. *Hence, we will need to ensure that if $(\xi'_{out}, \xi'_{relay})$ is a stationary point of the o.d.e., then $(\lambda^*(\xi'_{out}, \xi'_{relay}), \xi'_{out}, \xi'_{relay}) \in \mathcal{K}(\overline{q}, \overline{N})$. In order to ensure this, we need to choose A_2 and A_3 properly. The choice of A_2 and A_3 is rather technical, and is explained in detail in Appendix E, Section C. Here we will just provide the method of choosing A_2 and A_3, without any explanation of why they should be chosen in this way*. The number A_2 has to be chosen so large that under $\xi_{out} = A_2$ and for all $A+1 \leq u \leq A+B$, we will have $\mathbb{P}(\arg\min_{\gamma \in \mathcal{S}}(\gamma + A_2 Q_{out}(u, \gamma, W)) = P_M) > 1 - \kappa$ for some small enough $\kappa > 0$. We must also have $\overline{Q}^*_{out}(A_2, 0)/\overline{U}^*(A_2, 0) \leq \overline{q}$. The number A_3 has to be chosen so large that for any $\xi_{out} \in [0, A_2]$, we will have $\overline{U}^*(\xi_{out}, A_3) > (1/\overline{N})$ (provided that $(1/\overline{N}) < A+B$). The numbers A_2 and A_3 have to be chosen so large that there exists at least one $(\xi'_{out}, \xi'_{relay}) \in [0, A_2] \times [0, A_3]$ such that $(\lambda^*(\xi'_{out}, \xi'_{relay}), \xi'_{out}, \xi'_{relay}) \in \mathcal{K}(\overline{q}, \overline{N})$.

(iv) *Asymptotic behaviour of the iterates*: If the pair $(\overline{q}, \overline{N})$ is such that one can be met with strict inequality and the other can be met with equality while using the optimal mean power per step for this pair $(\overline{q}, \overline{N})$, then one Lagrange multiplier will converge to 0. This will happen if $\overline{q} > (\mathbb{E}_W Q_{out}(A+B, P_1, W)/(A+B))$; we will have $\xi^{(k)}_{out} \to 0$ (obvious from OptExploreLim with $\xi_{out} = 0$) in this case. Here we will place all the relays at the $(A+B)$-th step and use the smallest power level at each node. On the other hand, if the constraints are not feasible, then either $\xi^{(k)}_{out} \to A_2$ or $\xi^{(k)}_{relay} \to A_3$ (since convergence to ∞ is not possible due to projection) or both will happen.
$\mathcal{K}(\overline{q}, \overline{N})$ may have multiple tuples. But simulation results show that it has only one tuple in case it is nonempty. □

C. Asymptotic Performance of Algorithm 8

Let us denote by π_{oelal} the (nonstationary) deployment policy induced by Algorithm 8. We will now show that π_{oelal} is an optimal policy for the constrained problem (4).

Theorem 13: Suppose that Assumption 1 and Assumption 2 hold. Then, under proper choice of A_2 and A_3, the policy π_{oelal} solves the problem (4); i.e., we have:

$$\limsup_{x \to \infty} \frac{\mathbb{E}_{\pi_{oelal}} \sum_{i=1}^{N_x} \Gamma_i}{x} = \gamma^*$$

$$\limsup_{x \to \infty} \frac{\mathbb{E}_{\pi_{oelal}} \sum_{i=1}^{N_x} Q^{(i,i-1)}_{out}}{x} \leq \overline{q}, \ \limsup_{x \to \infty} \frac{\mathbb{E}_{\pi_{oelal}} N_x}{x} \leq \overline{N}$$

Proof: See Appendix E, Section D. ■

VIII. Convergence Speed of Learning Algorithms: A Simulation Study

In this section, we provide a simulation study to demonstrate the convergence rate of Algorithm 7 and Algorithm 8. The simulations are provided for $\eta = 4.7$, $\sigma = 7.7$ dB, $\delta = 20$ m, $A = 0$, $B = 5$, $c = 10^{0.17}$, $P_{rcv-min} = -97$ dBm, $\mathcal{S} = \{-18, -7, -4, 0, 5\}$ dBm (see Section II for notation and Section V-A for parameter values).

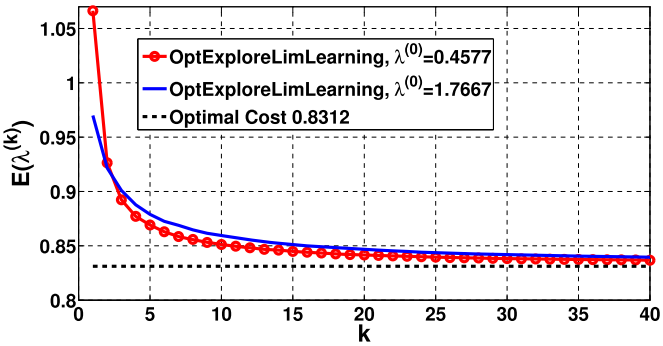

Fig. 4. Demonstration of the convergence of OptExploreLimLearning (Algorithm 7) as deployment progresses. $\lambda^{(0)}$ has not been included here.

A. OptExploreLimLearning for Given ξ_{out} and ξ_{relay}

Let us choose $\xi_{out} = 100$, $\xi_{relay} = 1$. We assume that the propagation environment in which we are deploying is characterized by the parameters as in Section V-A (e.g., $\eta = 4.7$, $\sigma = 7.7$ dB). The optimal average cost per step, under these parameter values, is 0.8312 (computed numerically).

On the other hand, for $\eta = 4$, $\sigma = 7$ dB, $\xi_{out} = 100$ and $\xi_{relay} = 1$, the optimal average cost per step is 0.4577, and it is 1.7667 for $\eta = 5.5$, $\sigma = 9$ dB. These two cases correspond to two different imperfect estimates of η and σ available to the agent before deployment starts.

Suppose that the actual $\eta = 4.7$, $\sigma = 7.7$ dB, but at the time of deployment we have an initial estimate that $\eta = 4$, $\sigma = 7$ dB; thus, we start with $\lambda^{(0)} = 0.4577$. After placing the k-th relay, the actual average cost per step of the relay network connecting the k-th relay to the sink is $\lambda^{(k)}$; this quantity is a random variable whose realization depends on the shadowing realizations over the links measured in the process of deployment up to the k-th relay. We ran 10000 simulations of Algorithm 7, starting with different seeds for the shadowing random process, and estimating $\mathbb{E}(\lambda^{(k)})$ as the average of the samples of $\lambda^{(k)}$ over these 10000 simulations. We also do the same for $\lambda^{(0)} = 1.7667$ (optimal cost for $\eta = 5.5$, $\sigma = 9$ dB).

The estimates of $\mathbb{E}(\lambda^{(k)})$, $k \geq 1$ as a function of k, for the two initial values of $\lambda^{(0)}$, are shown in Fig. 4. Also shown, in Fig. 4, is the optimal value $\lambda^* = 0.8312$ for the true propagation parameters (i.e., $\eta = 4.7$, $\sigma = 7.7$ dB). From Fig. 4, we observe that $\mathbb{E}(\lambda^{(k)})$ approaches the optimal cost 0.8312 for the actual propagation parameters, as the number of deployed relays increases, and gets to within 10% of the optimal cost by the time that 4 or 5 relays are placed, starting with two widely different initial guesses of the propagation parameters. Thus, OptExploreLimLearning could be useful even when the distance can be covered by only 4 to 5 relays.

Note that, each simulation yields one sample path of the deployment process. We obtained the estimates of $\mathbb{E}(\lambda^{(k)})$ as a function of k (by averaging over 10000 sample paths); the convergence speed will vary across sample paths even though $\lambda^{(k)} \to 0.8312$ almost surely as $k \to \infty$.

B. OptExploreLimAdaptiveLearning

In this section, we will discuss how OptExploreLimAdaptiveLearning (Algorithm 8) performs for deployment over a finite distance under an unknown propagation environment. We assume that the true propagation parameters are given in Section V-A (e.g., $\eta = 4.7$, $\sigma = 7.7$ dB). If we know

the true propagation environment, then, under the choice $\xi_{relay} = 1$ and $\xi_{out} = 100$, the optimal average cost per step will be 0.8312, and this can be achieved by OptExploreLim (Algorithm 3). The corresponding mean outage per step will be $0.0045/2.2859 = 0.001969$ (i.e., 0.1969%) and the mean number of relays per step will be 1/2.2859.

Now, suppose that we wish to solve the constrained problem in (4) with the targets $\overline{q} = 0.001969$ (i.e., 0.1969%) and $\overline{N} = 1/2.2859$, but we do not know the true propagation environment. Hence, the deployment will use OptExploreLimAdaptiveLearning with some choice of $\xi_{out}^{(0)}$, $\xi_{relay}^{(0)}$ and $\lambda^{(0)}$.

We seek to compare among the following three scenarios: (i) η and σ are completely known (we use OptExploreLim with $\xi_{relay} = 1$ and $\xi_{out} = 100$ in this case), (ii) imperfect estimates of η and σ are available prior to deployment, and OptExploreLimAdaptiveLearning is used to learn the optimal policy, and (iii) imperfect estimates of η and σ are available prior to deployment, but a corresponding suboptimal policy is used throughout the deployment without any update. For convenience in writing, we introduce the abbreviations OELAL and OEL for OptExploreLimAdaptiveLearning and OptExploreLim, respectively. We also use the abbreviation FPWU for "Fixed Policy without Update." Now, we formally introduce the following cases that we consider in our simulations:

(i) **OEL**: Here we know $\eta = 4.7$, $\sigma = 7.7$ dB, and use OptExploreLim (Algorithm 3) with $\xi_{out} = 100$, $\xi_{relay} = 1$, $\lambda^* = 0.8312$. OEL will meet both the constraints with equality, and will minimize the mean power per step.

(ii) **OELAL Case 1**: OELAL Case 1 is the case where the true η and σ (which are unknown to the deployment agent) are specified by Section V-A, but we use OptExploreLimAdaptiveLearning with $\xi_{out}^{(0)} = 75$, $\xi_{relay}^{(0)} = 1.25$ and $\lambda^{(0)} = 0.5007$, in order to meet the constraints specified earlier in this subsection. Note that, under $\xi_{out} = 75$ and $\xi_{relay} = 1.25$, the optimal mean cost per step is 0.5007 for $\eta = 4$, $\sigma = 7$ dB. Hence, we start with a wrong choice of Lagrange multipliers, a wrong estimate of η and σ, and an estimate of the optimal average cost per step which corresponds to these wrong choices. The goal is to see how fast the variables $\lambda^{(k)}$, $\xi_{out}^{(k)}$ and $\xi_{relay}^{(k)}$ converge to the desired target 0.8312, 100 and 1, respectively. We also study how close to the desired target values are the quantities such as mean power per step, mean outage per step and mean placement distance for the relay network between k-th relay and the sink node.

(iii) **OELAL Case 2**: OELAL Case 2 is different from OELAL Case 1 only in the aspect that $\lambda^{(0)} = 1.7679$ is used in OELAL Case 2. Note that, under $\xi_{out} = 75$ and $\xi_{relay} = 1.25$, the optimal mean cost per step is 1.7679 for $\eta = 5.5$, $\sigma = 9$ dB.

(iv) **FPWU Case 1**: In this case, the true η and σ are unknown to the deployment agent. The deployment agent uses $\xi_{out} = 75$, $\xi_{relay} = 1.25$ and $\lambda^* = 0.5007$ throughout the deployment process under the algorithm specified by (7). Clearly, he chooses a wrong set of Lagrange multipliers $\xi_{out} = 75$, $\xi_{relay} = 1.25$, and he has a wrong estimate $\eta = 4$, $\sigma = 7$ dB. The optimal average cost per step $\lambda^* = 0.5007$ is computed for these wrong choice of parameters, and the corresponding suboptimal policy is used throughout the deployment process without any update; this will be used to demonstrate the gain in performance by updating the policy under OptExploreLimAdaptiveLearning, w.r.t. the case where the suboptimal policy is used without any online update.

(v) **FPWU Case 2**: It differs from FPWU Case 1 only in the aspect that we use $\lambda^* = 1.7679$ in FPWU Case 2. Recall that, under $\xi_{out} = 75$ and $\xi_{relay} = 1.25$, the optimal mean cost per step is 1.7679 for $\eta = 5.5$, $\sigma = 9$ dB.

For simulation of OELAL, we chose the step sizes as follows. We chose $a_k = 1/k^{0.55}$, chose $b_k = 10000/k^{0.8}$ for the ξ_{out} update and $b_k = 1/k^{0.8}$ for the ξ_{relay} update (note that, both ξ_{out} and ξ_{relay} are updated in the same timescale). We simulated 10000 independent network deployments (i.e., 10000 sample paths of the deployment process) with OptExploreLimAdaptiveLearning, and estimated (by averaging over 10000 deployments) the expectations of $\lambda^{(k)}, \xi_{out}^{(k)}, \xi_{relay}^{(k)}$, mean power per step $(\mathbb{E}_{\pi_{oelal}} \sum_{i=1}^{k} \Gamma_i)/(\mathbb{E}_{\pi_{oelal}} \sum_{i=1}^{k} U_i)$, mean outage per step $(\mathbb{E}_{\pi_{oelal}} \sum_{i=1}^{k} Q_{out}^{(i,i-1)})/(\mathbb{E}_{\pi_{oelal}} \sum_{i=1}^{k} U_i)$ and mean placement distance $(\mathbb{E}_{\pi_{oelal}} \sum_{i=1}^{k} U_i)/k$, from the sink node to the k-th placed node. In each simulated network deployment, we placed 20000 nodes, i.e., k was allowed to go up to 20000. Asymptotically the estimates are supposed to converge to the values provided by OEL.

Observations From the Simulations: The results of the simulations are summarized in Fig. 5. We observe that, the estimates of the expectations of $\lambda^{(20000)}$, $\xi_{out}^{(20000)}$, $\xi_{relay}^{(20000)}$, mean power per step up to the 20000th node, mean outage per step up to the 20000th node, and mean placement distance (in steps) over 20000 deployed nodes are 0.8551, 104.0606, 1.0385, 0.2005, 0.2% (i.e., 0.002) and 2.2939 for the OELAL Case 1, whereas those quantities are supposed to be equal to 0.8312, 100, 1, 0.1955, 0.1969% (i.e., 0.001969) and 2.2859, respectively. We found similar results for OELAL Case 2 also. Hence, the quantities converge very close to the desired values. *We have shown convergence only up to $k = 50$ deployments in most cases, since the convergence rate of the algorithms in the initial phase are most important in practice*.

All the quantities except expectation of $\xi_{out}^{(k)}$ and $\xi_{relay}^{(k)}$ (which are updated in a slower timescale) converge reasonably close to the desired values by the time the 50th relay is placed, which will cover a distance of roughly 2–3.5 km. distance.

FPWU Case 1 and FPWU Case 2 either violate some constraint or uses significantly higher per-step power compared to OEL. But, by using the OptExploreLimAdaptiveLearning algorithm, we can achieve per-step power expenditure close to the optimal while (possibly) violating the constraints by small amount; even in case the performance of OELAL is not very close to the optimal performance, it will be significantly better than the performance under FPWU cases (compare OELAL Case 2 and FPWU Case 2 in Fig. 5). □

The speed of convergence will depend on the choice of the step sizes a_k and b_k; optimizing the rate of convergence by choosing optimal step sizes is left for future endeavours in this direction. Also, note that, the choice of $\xi_{out}^{(0)}$, $\xi_{relay}^{(0)}$ and $\lambda^{(0)}$ will have a significant effect on the performance of the network over a finite length; the more accurate are the estimates of η and σ, and the better are the initial choice of $\xi_{out}^{(0)}$, $\xi_{relay}^{(0)}$ and $\lambda^{(0)}$, the better will be the convergence speed of OptExploreLimAdaptiveLearning.

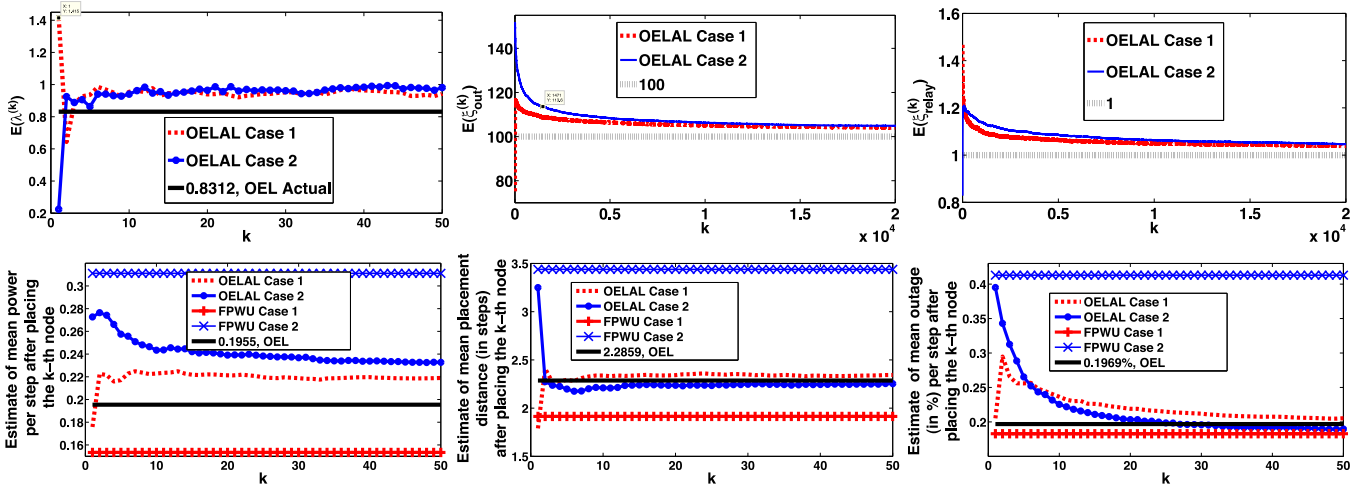

Fig. 5. Demonstration of the convergence of OptExploreLimAdaptiveLearning as deployment progresses. In the legends, "OEL" refers to the values that are obtained if OptExploreLim is used; these are the target values for OptExploreLimAdaptiveLearning. Note that, we have used line styles for ξ_{out} and ξ_{relay} updates, that are different from the line styles of other four plots. Also note that, outage probabilities are shown in percentage and not in decimal.

IX. Physical Deployment Experiments

For completeness, we briefly summarize experimental results that were reported in our conference paper [12]. We performed an actual deployment experiment along a long tree-lined road in our campus (not exactly a straight line, which is the reality in a forest) with iWiSe motes equipped with 9 dBi antennas. We chose $\xi_{out} = 100$, $\xi_{relay} = 1$, $B = 5$ steps, $\delta = 50$ meters, and $\mathcal{S} = \{-7, -4, 0, 5\}$ dBm. We used the packet error rate (PER) of a link as a substitute for outage probability; this does not violate the assumptions of our formulation. For $\eta = 4$, $\sigma = 7$ dB, $\xi_{out} = 100$, $\xi_{relay} = 1$, the optimal average cost per step is 1.0924 (computed numerically). Taking $\lambda^{(0)} = 1.0924$, we performed a real deployment experiment with OptExploreLimLearning. The deployed network (along with power levels, outage probabilities and link lengths) is shown in Fig. 6. The sink is denoted by the "house" symbol. The algorithm placed relays at successive distances of 150 m, 50 m, 50 m, 100 m, and 150 m, thereby covering 500 m until the source was placed. The two short (50 m long) links are created due to significant path-loss at the turn in the road. After deployment, we used the last placed node as the source and sent periodic traffic (at various rates) from the source to the sink. The end-to-end packet loss probability increases with arrival rate (Fig. 6); this happens due to carrier sense failures and collisions because of simultaneous transmissions from different nodes. At very low arrival rate, the loss probability is 0 (but the sum PER under the lone packet model is not 0). This happens since there are link level retransmissions and since the outage durations are relatively short; in case a packet encounters an outage in a link, the retransmission attempts succeed with high probability. *The results demonstrate that, even though the design was for the lone packet model, the network can carry 4 packets/second (packet size is 127 bytes) with $P_{loss} \leq 1\%$, which is sufficient for many applications. Hence, network design with the lone packet model assumption is reasonable for those applications*.

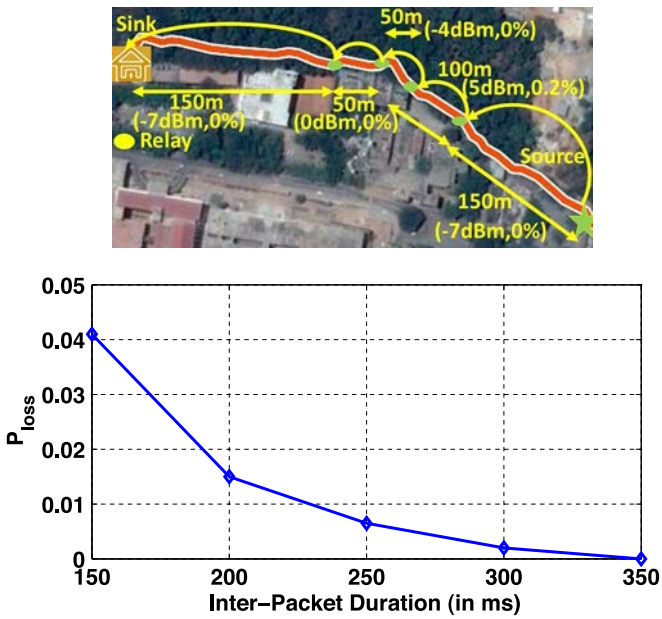

Fig. 6. Actual deployment along a long tree-lined road in the Indian Institute of Science Campus using OptExploreLimLearning with iWiSe motes, $\xi_{out} = 100$, $\xi_{relay} = 1$: five nodes (including the source) are placed; link lengths, transmit powers, and % outage probabilities are shown; the plot shows variation of end-to-end loss probability with inter-packet duration, for periodic traffic generated from the source. Picture and plot are taken from [12].

X. Conclusion

We have developed several approaches for as-you-go deployment of wireless relay networks using on-line measurements, under a very light traffic assumption. Each problem was formulated as an MDP and its optimal policy structure was studied. We also studied a few learning algorithms that will asymptotically converge to the corresponding optimal policies. Numerical and experimental results have been provided to illustrate the performance and trade-offs.

This work can be extended or modified in several ways: (i) Networks that are robust to node failures and long term link variations would either require each relay to have multiple neighbours (i.e., the deployment would need to be multi-connected), or the nodes can choose their transmit powers adaptively as the environment changes. (ii) It would be of interest to develop deployment algorithms for 2 and 3 dimensional regions, where a team of agents cooperates to carry out the deployment. (iii) We have assumed very light traffic conditions in our design (what we call "lone packet" traffic), but our experiments show that these designs can carry a

useful amount of positive traffic. It will be of interest, however, to develop deployment algorithms that can provide theoretical guarantees to achieve desired traffic rates.

References

[1] A. Chattopadhyay, M. Coupechoux, and A. Kumar, "Sequential decision algorithms for measurement-based impromptu deployment of a wireless relay network along a line," 2015 [Online]. Available: http://arxiv.org/abs/1502.06878

[2] V. Dyo *et al.*, "Evolution and sustainability of a wildlife monitoring sensor network," in *Proc. 8th ACM SenSys*, 2010, pp. 127–140.

[3] A. A. A. Alkhatib, "A review on forest fire detection techniques," *Int. J. Distrib. Sensor Netw.*, vol. 2014, 2014, Article ID 597368.

[4] M. Howard, M. J. Matarić, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," *Auton. Robots*, vol. 13, no. 2, pp. 113–126, 2002.

[5] M. R. Souryal, J. Geissbuehler, L. E. Miller, and N. Moayeri, "Real-time deployment of multihop relays for range extension," in *Proc. ACM MobiSys*, 2007, pp. 85–98.

[6] T. Aurisch and J. Tölle, "Relay placement for ad-hoc networks in crisis and emergency scenarios," in *Proc. IST Symp.*, 2009, pp. 2-1–2-10.

[7] H. Liu *et al.*, "Automatic and robust breadcrumb system deployment for indoor firefighter applications," in *Proc. ACM MobiSys*, 2010, pp. 21–34.

[8] H. Liu *et al.*, "Efficient and reliable breadcrumb systems via coordination among multiple first responders," in *Proc. 22nd IEEE PIMRC*, 2011, pp. 1005–1009.

[9] J. Q. Bao and C. Lee, "Rapid deployment of wireless ad hoc backbone networks for public safety incident management," in *Proc. IEEE GLOBECOM*, 2007, pp. 1217–1221.

[10] A. Sinha, A. Chattopadhyay, K. P. Naveen, M. Coupechoux, and A. Kumar, "Optimal sequential wireless relay placement on a random lattice path," *Ad Hoc Netw.*, vol. 21, pp. 1–17, 2014.

[11] A. Chattopadhyay, M. Coupechoux, and A. Kumar, "Measurement based impromptu deployment of a multi-hop wireless relay network," in *Proc. 11th IEEE WiOpt*, 2013, pp. 349–356.

[12] A. Chattopadhyay *et al.*, "Impromptu deployment of wireless relay networks: Experiences along a forest trail," in *Proc. IEEE MASS*, 2014, pp. 232–236.

[13] P. Agrawal and N. Patwari, "Correlated link shadow fading in multi-hop wireless networks," 2008 [Online]. Available: http://arxiv.org/abs/0804.2708

[14] A. Bhattacharya *et al.*, "SmartConnect: A system for the design and deployment of wireless sensor networks," in *Proc. 5th IEEE COMSNETS*, 2013, pp. 1–10.

[15] R. Upadrashta *et al.*, "An animation-and-chirplet based approach to intruder classification using PIR sensing," in *Proc. 10th IEEE ISSNIP*, 2015, pp. 1–6.

[16] B. Aghaei, "Using wireless sensor network in water, electricity and gas industry," in *Proce. 3rd IEEE Int. Conf. Electron. Comput. Technol.*, 2011, pp. 14–17.

[17] T. Adame, A. Bel, B. Bellalta, J. Barcelo, and M. Oliver, "IEEE 802.11ah: The WiFi approach for M2M communications," *IEEE Wireless Commun.*, vol. 21, no. 6, pp. 144–152, Dec. 2014.

[18] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. ACM WSNA*, 2002, pp. 88–97.

[19] S. Lohier, A. Rachedi, I. Salhi, and E. Livolant, "Multichannel access for bandwidth improvement in IEEE 802.15.4 wireless sensor networks," 2011 [Online]. Available: https://hal-enpc.archives-ouvertes.fr/hal-00680871/document

[20] N. Abdeddaim, F. Theoleyre, F. Rousseau, and A. Duda, "Multichannel cluster tree for 802.15.4 wireless sensor networks," in *Proc. 23rd IEEE PIMRC*, 2012, pp. 590–595.

[21] E. Toscano and L. L. Bello, "Multichannel superframe scheduling for IEEE 802.15.4 industrial wireless sensor networks," *IEEE Trans. Ind. Inf.*, vol. 8, no. 2, pp. 337–350, May 2012.

[22] A. Bardella, N. Bui, A. Zanella, and M. Zorzi, "An experimental study on IEEE 802.15.4 multichannel transmission to improve RSSI-based service performance," in *Proc. 4th REALWSN*, 2010, pp. 154–161.

[23] A. Bhattacharya and A. Kumar, "QoS aware and survivable network design for planned wireless sensor networks," 2014 [Online]. Available: http://arxiv.org/abs/1110.4746

[24] M. Vodel and W. Hardt, "Energy-efficient communication in distributed, embedded systems," in *Proc. 9th RAWNET & IEEE WiOpt*, 2013, pp. 641–647.

[25] F. J. Beutler and K. W. Ross, "Optimal policies for controlled markov chains with a constraint," *J. Math. Anal. Appl.*, vol. 112, pp. 236–252, 1985.

[26] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA, USA: Athena Scientific, 2007, vol. II.

[27] O. Hernandez-Lerma and J. B. Lasserre, *Discrete-Time Markov Control Processes Basic Optimality Criteria*. New York, NY, USA: Springer, 1996.

[28] H. C. Tijms, *A First Course in Stochastic Models*. New York, NY, USA: Wiley, 2003.

[29] O. Vega-Amaya and F. Luque-Vsquez, "Sample-path average cost optimality for semi-Markov control processes on Borel spaces: Unbounded costs and mean holding times," *Appl. Math.*, vol. 27, no. 3, pp. 343–367, 2000.

[30] Automation Systems Technology Centre (ASTeC), "Wireless sensor network," [Online]. Available: http://www.astec.org.in/astec/content/wireless-sensor-network

[31] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[32] D.-J. Ma and A. M. Makowski, "A class of steering policies under a recurrence condition," in *Proc. 27th IEEE CDC*, 1988, pp. 1192–1197.

[33] H. J. Kushner and D. S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. New York, NY, USA: Springer-Verlag, 1978.