Asynchronous Stochastic Approximation Based Learning Algorithms for As-You-Go Deployment of Wireless Relay Networks along a Line

Arpan Chattopadhyay, Avishek Ghosh, and Anurag Kumar

arXiv:1709.01566v1 [cs.NI] 5 Sep 2017

Abstract—We are motivated by the need, in emergency situations, for impromptu (or "as-you-go") deployment of multihop wireless networks, by human agents or robots (e.g., unmanned aerial vehicles (UAVs)); the agent moves along a line, makes wireless link quality measurements at regular intervals, and makes on-line placement decisions using these measurements. As a first step we have formulated such deployment along a line as a sequential decision problem. In our earlier work, reported in [1], we proposed two possible deployment approaches: (i) the pure as-you-go approach where the deployment agent can only move forward, and (ii) the explore-forward approach where the deployment agent explores a few successive steps and then selects the best relay placement location among them. The latter was shown to provide better performance (in terms of network cost, network performance and power expenditure), but at the expense of more measurements and deployment time, which makes explore-forward impractical for quick deployment by an energy constrained agent such as a UAV. Further, since in emergency situations the terrain would be unknown, the deployment algorithm should not require a-priori knowledge of the parameters of the wireless propagation model. In [1] we, therefore, developed learning algorithms for the explore-forward approach.

The current paper fills in an important gap by providing deploy-and-learn algorithms for the pure as-you-go approach. We formulate the sequential relay deployment problem as an average cost Markov decision process (MDP), which trades off among power consumption, link outage probabilities, and the number of relay nodes in the deployed network. While the pure as-you-go deployment problem was previously formulated as a discounted cost MDP (see [1]), the discounted cost MDP formulation was not amenable for learning algorithms that are proposed in this paper. In this paper, first we show structural results for the optimal policy corresponding to the average cost MDP, and provide new insights into the optimal policy. Next, by exploiting the special structure of the average cost optimality equation and by using the theory of *asynchronous* stochastic approximation (in single and two timescale), we develop two learning algorithms that asymptotically converge to the set of optimal policies as deployment progresses. Numerical results show reasonably fast speed of convergence, and hence the model-free algorithms can be useful for practical, fast deployment of emergency wireless networks.

Index Terms—Wireless networks, impromptu network deployment, as-you-go relay placement, relay placement by UAV, Markov decision process, stochastic approximation.

1 Introduction

In emergency situations, such as fires in large buildings or forests, or houses in a flooded neighbourhood (without electric power and telecom infrastructure), there is a need to quickly deploy wireless networks for situation monitoring. Such networks could be deployed by first responders (e.g., fire-fighters moving through a burning building [2]), or by robots (e.g., unmanned aerial vehicles (UAVs) hopping over the rooftops of flooded homes or flying over a long road [3], [4], [5]), or by forest guards along forest trails ([1]).[1] Typically, such networks would have one or more *base-stations*, where the command and control would reside, and to which the measurements from the sensors in the field would need to be routed. For example, in the case of the fire-fighting example, the base-station would be in a control truck parked outside the building. Evidently, in such emergency situations, there is a need for "as-you-go" deployment algorithms as there is no time for network planning. As they move through the affected area, the first-responders would need to deploy wireless relays, in order to provide routes for the wireless sensors for situation monitoring.

With the above motivation for quick deployment of multihop wireless networks, in our work, in the present and earlier papers ([1], [9], [10]), we have considered the particular situation of as-you-deployment of relays along

The research reported in this paper 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, by an Indo-French Centre for Promotion of Advance Research (IFCPAR) funded project, and by the Department of Science and Technology (DST, India), via a J.C. Bose Fellowship.

Arpan Chattopadhyay is with the Electrical Engineering department, University of Southern California, Los Angeles. Avishek Ghosh is with the EECS department, UC Berkeley. Anurag Kumar is with the Department of ECE, Indian Institute of Science (IISc), Bangalore. This work was done when Arpan Chattopadhyay and Avishek Ghosh were with the Department of ECE, IISc. Email: arpanc.ju@gmail.com, avishek.ghosh38@gmail.com, anurag@ece.iisc.ernet.in

All appendices are provided in the supplementary material.

1. See [6] and [7, Section 5] for application of multihop wireless sensor networks in wildlife monitoring and forest fire detection. [8] illustrates a future possibility where drones deploy high speed, solar-powered access points on the roofs of city buildings in order to provide high speed internet connection. *The drone can land on the ground or on a rooftop for link quality measurements, and can again take off.*

a line, starting from a base-station, in order to connect a source of data (e.g., a sensor) whose location is revealed (or is itself placed) only during the deployment process. Figure 1 depicts our model for as-you-go deployment along a line, and also illustrates the difference between planned deployment and as-you-go deployment. As-you-go deployment along a line is motivated by the need for quick deployment of relay networks along long forest trails by humans or mobile robots, and relay network deployment along a long straight road by human agents or UAVs. In practice, the location of the data source would be a-priori unknown, as the deployment agent would also need to select locations at which to place the sensors. Yet, as the deployment agent traverses the line, he or she (or it) has to judiciously deploy wireless relays so as to end up with a viable network connecting the data source (e.g., the sensor) to the sink. In a planned approach, all possible links could be evaluated; in an as-you-go approach, however, the agent needs to make decisions based on whatever links can be evaluated as deployment progresses.

Motivated by the need for as-you-go deployment of wireless sensor networks (WSNs) over large terrains, such as forest trails, in our earlier work [1] we had considered the problem of multihop wireless network deployment along a line, where a single deployment agent starts from a sink node (e.g., a base-station), places relays as the agent walks along the line, and finally places a source node (e.g., a sensor) where required. We formulated this problem as a measurement based sequential decision problem with an appropriate additive cost over hops. In order to explore the range of possibilities, we considered two alternatives for measurement and deployment: (i) the explore-forward approach: after placing a node, the deployment agent explores several potential placement locations along the next line segment, and then decides on where to place the next node, and (ii) the pure as-you-go approach: the deployment agent only moves forward, making measurements and committing to deploying nodes as he goes.

As expected, in [1] we found that the explore-forward approach yields better performance (in terms of the additive per hop cost (see [1, Section V]); but, of course, this approach takes more time for the completion of deployment. Hence, explore-forward is prohibitive when soldiers or robots need to quickly deploy a relay network along a forest trail or along a long road. In addition, a deployment agent such as a UAV would be limited by its fuel, and it would be desirable to complete the mission as quickly as possible, without many fuel consuming manoeuvres. Thus, pure as-you-go is the only option for network deployment by UAVs along long roads (see [3] for practical network deployment along a road by a UAV). Further, in an emergency situation, the algorithm cannot expect to be given the parameters of the propagation environment; this gives rise to the need for deploy-and-learn algorithms.

In [1], although we introduced explore-forward and pure as-you-go approaches, we developed learning algorithms for explore-forward alone. However, with the above motivation, our current paper fills in an important gap by proposing online learning algorithms for pure as-you-go deployment. We mathematically formulate the problem of *pure as-you-go deployment* along a line as an *optimal sequential decision problem* so as to minimize the expected average cost per step, where the cost of a deployment is a linear combination of the sum transmit power, the sum outage probability and the number of relays deployed. We formulate the problem as a Markov decision process (MDP) and obtain the optimal policy structure. Next, we propose two learning algorithms (based on asynchronous stochastic approximation) and prove their asymptotic convergence to the optimal policy for the long-run average cost minimization problem. Finally, we demonstrate the convergence rate of the learning algorithms via numerical exploration.

The new contributions of this paper, in relation to [1], are discussed in Section 1.2.

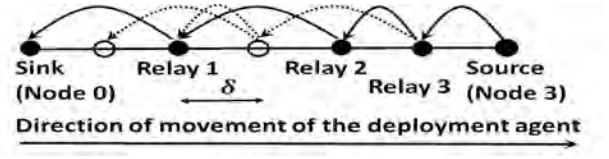

Figure 1: A line network connecting a source (e.g., a sensor) to a sink (e.g., a control centre) via relay nodes. The dots in between (filled and unfilled) denote potential relay locations, and are spaced δ meters apart. The deployed network consists of three relays (dots labeled Relay 1, 2, and 3) placed at three potential locations. The solid arrows show the multi-hop path from the source to the sink. The unfilled dots represent locations where no relay was placed. *The dotted arrows represent some other possible links between pairs of potential locations.* In case of planned deployment, link qualities between all potential location pairs need to be measured. But, in as-you-go deployment, the agent only measures the qualities of link from his (or its) current location to the previously placed nodes.

1.1 Related Work

Prior work on the problem of impromptu deployment of WSN consists of mostly heuristic algorithms validated by experimentation. For example, the authors of [11] address this problem by studying experimentally the variation in indoor link quality. The authors of [12] also took a similar approach. The authors of [13] provide heuristics for deploying (incrementally) sensors so that a certain area is covered (e.g., self-deployment of autonomous robot teams). Bao and Lee, in [14], address the problem of a group of first responders starting from a base station (e.g., a command center) and placing relay nodes while walking through a region devoid of communication infrastructure, in order to stay connected among themselves as well as with the base station. Liu et al., in [2], describe a *breadcrumbs* system meant for firefighters operating inside a building; this paper is in similar spirit with ours, but their goal is just to maintain connection with k previously placed nodes. This work was later extended by them in [15] which provides a reliable

multiuser breadcrumbs system. However, all the above works are based on heuristic algorithms, rather than on rigorous formulations; hence they do not provide any provable performance guarantee. A nice survey on rapid deployment of post-disaster networks is available in [16]. Sensor network deployment by UAVs have also been studied in literature (see [4], [5]).

In our current paper, we have formulated as-you-go deployment as an MDP, found structural results for the optimal policy, and proposed learning algorithms to solve the sequential decision problems without using any prior knowledge of the radio propagation parameters. The use of MDP to formulate as-you-go deployment was first proposed by Mondal et al. in [17]. This work was later extended by Sinha et al. in [18], where the authors have provided an algorithm derived from an MDP formulation, so as to create a multi-hop wireless relay network between a sink and a source located at an unknown location, by placing relay nodes along a random lattice path. However, these papers do not consider spatial variability of wireless link qualities due to shadowing, which allows them to develop deployment algorithms that place the next relay based on the distance from the previously placed relay.

The spatial variation of link qualities due to shadowing requires measurement-based deployment; here the deployment agent makes placement decision at a given location based on the link quality to the previously placed node. Measurement-based as-you-go deployment was formulated first in [9], and was later extended in [1]. The authors of [1] have proposed two possible approaches for deployment along a line: (i) the *pure as-you-go approach* and (ii) the explore-forward approach. [1] has provided MDP formulations and policy structures for both approaches; transition probabilities of the MDPs depend on the radio propagation parameters in the environment, and, in practice, these parameters are not known to the agent prior to deployment. Hence, [1] also provides *learning algorithms for the explore-forward approach*, that converge asymptotically to the set of optimal deployment policies as more and more measurements are made in course of deployment. One of these learning algorithms was used for actual network deployment (see [1] and [10]). Design of a two-connected network to guard against node and link failures was discussed in [19], but it did not provide any learning algorithm.

We also developed, in [20], as-you-go deployment algorithms for deploying a multi-relay line network, so that the end-to-end achievable rate is maximized; but it was done for an information-theoretic, full-duplex, multi-relay channel model where the nodes carry out decode-and-forward relaying. However, devices with such sophisticated relaying capability is not yet available for full commercial use. On the other hand, our current paper designs deployment algorithms for networks carrying packetized data, which is common in present day wireless standards.

1.2 Contributions of this paper, in relation to [1]:

(i) New deploy-and-learn algorithms: Our current paper provides learning algorithms for the pure as-you-go approach (Algorithm 2 and Algorithm 3), whereas [1] provides learning algorithms only for explore-forward. The learning algorithms are required to discover the optimal deployment policy as deployment progresses, for the situation where no prior accurate knowledge on the statistical nature of radio propagation environment is available. Learning algorithms for pure as-you-go deployment is an important requirement since the pure as-you-go deployment approach is more suitable for very fast deployment over a large region. In fact, the number of measurements in explore-forward deployment can be double or triple than that of pure as-you-go ([1, Section V]) for practical deployment; this makes pure as-you-go a better choice for emergency network deployment by soldiers or commandos or energy-constrained autonomous agents such as robots and UAVs.

Unlike [1], the learning algorithms presented in this paper make use of *asynchronous stochastic approximation*, where different iterates are updated at different time instants (in the learning algorithms proposed in [1], all iterates are updated when a new relay is placed). We provide formal proof for the convergence of our proposed learning algorithms to the optimal deployment policy for pure as-you-go deployment; these proofs require a significant and non-trivial novel mathematical analysis (compared to [1]) in order to address many technical issues that arise in the proofs.

In other words, the most important contributions of the current paper w.r.t. [1], are the newly proposed learning algorithms for pure as-you-go deployment and their convergence proofs, which are new to the literature and addresses the problem of very fast deployment.

Interestingly, one of the learning algorithms proposed in this paper exhibits a nice separation property between estimation and control, which is not present in the learning algorithms presented in [1].

(ii) Average cost MDP formulation: [1] formulates the pure-as-you deployment problem for a line having a random length $L \sim Geometric(\theta)$ (mean is $\frac{1}{\theta}$), i.e., $\mathbb{P}(L = l) = (1-\theta)^{l-1}\theta$ where $l \in \{1, 2, \cdots, \infty\}$; the average cost optimal policy is obtained by taking $\theta \to 0$. Clearly, this requires value iteration to compute the optimal policy prior to deployment. This also requires the knowledge of radio propagation parameters, since they determine the transition probabilities of the MDP. On the other hand, our present paper establishes the structure of the optimal policy by using the average cost optimality equation (see (5)) with necessary modification; it turns out that such a formulation along with the special structure of the problem enables us to propose very simple learning algorithms to find the optimal policy, irrespective of whether the radio propagation parameters are known apriori or not. Thus, the average cost MDP formulation is a precursor to the learning algorithms (Algorithm 2

and Algorithm 3) presented later in this paper. Some new interesting properties of the value functions and the policy structure are also proved in the current paper, which were not present in [1] because the problem was formulated as discounted cost MDP in [1].

(iii) Additional measurements to facilitate learning: The pure-as-you go approach considered in our current paper is not exactly the same as that described in [1]. In [1], the agent makes a link quality measurement from the current location to the immediate previous node that he had placed. On the contrary, in the pure as-you-go approach described in our present paper, the agent measures link qualities from the current location to all previously placed nodes that are located within a certain distance. This is done to facilitate learning the optimal policy. The exact reason behind using this variation of pure as-you-go deployment will be explained in Section 4.1.

(iv) Bidirectional traffic: In Section 2.5, we explain how the deployment algorithms presented in this paper can be adapted to the case where each link in the network has to carry data packets in both directions.

1.3 Organization

The rest of the paper is organized as follows. The system model has been described in Section 2. MDP formulation for pure-as-you deployment has been provided in Section 3. The learning algorithms have been proposed in Section 4 and Section 5. Convergence speed of the learning algorithms are demonstrated numerically in Section 6, after which the conclusion follows. The proofs of all theorems are provided in the appendices available as supplementary material.

2 System Model

In this section, we describe the system model assumed in this paper. It has to be noted that the system model and notation used in this paper are similar in many aspects to those of [1]; a significant difference in the system model will be found in the deployment procedure as described in Section 2.2 (deployment process), and in Section 2.5 (bi-directional traffic). The channel model (Section 2.1), traffic model (Section 2.4) and deployment objective (Section 2.3) subsections are almost similar to the respective sections in [1]; but we describe the system model here in detail to make this paper self-contained.

We assume that the line (i.e., the road or the forest trail along which the network is deployed) is discretized into steps (starting from the sink), each having length δ. The points located at distances $\{k\delta\}_{k\in\{1,2,3,\cdots\}}$ are called potential locations; the agent is allowed to place nodes only at these potential locations. As the *single* deployment agent walks along the line, at each potential location, the agent measures the link quality from the current location to the previously placed nodes that are within a certain distance from the current location; placement decisions are made based on these measurements.

After deployment, as shown in Figure 1, the sink is called Node 0, and the relays are enumerated as nodes $\{1, 2, 3, \cdots\}$ as we move away from the sink. A link whose transmitter is Node i and receiver is Node j is called link (i, j).

2.1 Wireless Channel Model

We consider a wireless channel model where, for a link (i.e., a transmitter-receiver pair) with length r and transmit power γ, the received power of a packet (say the k-th packet) is given by:

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

Here c is the path-loss at a reference distance r_0, and η is the path-loss exponent. The fading random variable seen by the k-th packet is H_k (e.g., H_k is exponentially distributed for Rayleigh fading); it takes independent values over different coherent times. W denotes the shadowing random variable that captures the (random) spatial variation in path-loss. In this paper, W is assumed to take values from a set $\mathcal{W}$, and we denote by $g(w)$ the probability mass function or probability density function of W, depending on whether $\mathcal{W}$ is countable or uncountable. We assume that the transmit power of each node comes from a discrete set, $\mathcal{S} := \{P_1, P_2, \cdots, P_M\}$, where the power levels are arranged in ascending order.

Shadowing becomes spatially uncorrelated if the transmitter or receiver is moved by a certain distance that depends on the sizes of the scatterers in the environment (see [21]). It was shown experimentally that, in a forest-like environment, shadowing has log-normal distribution (i.e., $\log_{10} W \sim \mathcal{N}(0, \sigma^2)$ where σ is the standard deviation of log-normal shadowing) and the shadowing decorrelation distance can be as small as 6 meters (see [10]). In this paper, we assume that the step size δ is chosen to be more than the shadowing decorrelation distance; this allows us to assume that the shadowing at any two different links in the network are independent.

The k-th packet is said to see an *outage* in the link if $P_{rcv,k} \leq P_{rcv-min}$, where $P_{rcv-min}$ is a threshold depending on the modulation scheme and the properties of the receiving node. For example, $P_{rcv-min}$ can be chosen to be -88 dBm for the TelosB "motes" (see [22]), and -97 dBm for iWiSe motes (see [23]). For a link with length r, transmit power γ and shadowing realization $W = w$, the outage probability is denoted by $Q_{out}(r, \gamma, w)$; it is increasing in r and decreasing in γ, w. $Q_{out}(r, \gamma, w) = \mathbb{P}(P_{rcv,k} \leq P_{rcv-min})$ depends on the fading statistics; if H is exponentially distributed with mean 1 (i.e., for Rayleigh fading), then $Q_{out}(r, \gamma, w) = \mathbb{P}(\gamma c(\frac{r}{r_0})^{-\eta} wH \leq P_{rcv-min}) = 1 - e^{-\frac{P_{rcv-min}(\frac{r}{r_0})^{\eta}}{\gamma c w}}$. The outage probability of a randomly chosen link (with given r and γ) is a random variable, with the randomness coming from shadowing W. Outage probability can be measured by sending a large number of packets over a

link and calculating the fraction of packets with received power less than $P_{rcv-min}$.

2.2 Pure As-You-Go Deployment Process

After placing a relay, the agent starts measuring the link qualities from the next B locations *one by one* (the value of B is fixed prior to deployment). At any given location, the agent uses the measurements from the current location to make a placement decision; the agent does not make measurements from all of those B locations in order to place a new relay.

At any given location, the agent measures the link qualities from the given location to all previously placed nodes that are within $B\delta$ distance from the current location; see Figure 2. Let us assume that the agent is standing at a distance $k\delta$ from the sink. Let $\mathcal{I}_k := \{r \in \{1, 2, \cdots, B\} :$ a relay was placed at a distance $(k-r)\delta$ from the sink$\}$. Then, the agent at this location will measure the outage probabilities $\{Q_{out}(r, \gamma, w_r)\}_{\gamma \in \mathcal{S}, r \in \mathcal{I}_k}$ (w_r is the realization of shadowing in a link of length r steps).

However, at each location, only the link quality to the *immediately* previous node is used to decide whether to place a relay there or to move on to the next step. If the decision is to place a relay, then the agent also decides which transmit power $\gamma \in \mathcal{S}$ to use at that particular node. If the decision is not to place a relay, the agent moves to the next step. In this process, if he reaches B steps away from the previous relay, or if the source location is encountered, then he must place a node there.

It is important to note that, while the measurement to the *immediately* previous node is used to make a placement decision, other measurements made in this process provide useful information about the statistical characteristics of the radio propagation environment (more precisely, the probability distribution of $Q_{out}(r, \gamma, \cdot)$ for $r \in \{1, 2, \cdots, B\}, \gamma \in \mathcal{S}$), and those measurements are used to learn the optimal deployment policy as described in Section 4 and Section 5. But if the radio propagation parameters (such as η and σ) are exactly known, i.e., if the probability distribution of $Q_{out}(r, \gamma, \cdot)$ is known exactly, then these additional measurements will not be required (since shadowing is i.i.d. across links, these measurements will not provide any information about the link quality between the current location and the immediately previous node); this situation has been explored in Section 3, where measurement is made only to the previously placed relay node.

Choice of B: In general, the choice of B depends on the constraints and requirements for the deployment. Large B results in better performance at the expense of more measurements. One can simply choose B to be the largest integer such that, the probability that a randomly chosen wireless link with length $B\delta$ respects a certain outage constraint, is larger than some pre-specified target. This will make sure that the probability of obtaining a workable link is small in case the agent reaches a location that is more than $B\delta$ distance away from the previously placed node.

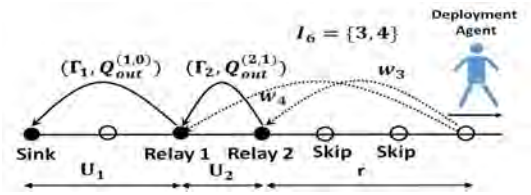

Figure 2: Illustration of pure as-you-go deployment with learning for $B = 4$. Here the deployment agent has already placed Relay 1 and Relay 2, and the corresponding inter-relay distances are U_1 and U_2. The placed relays use transmit powers Γ_1 and Γ_2, thereby achieving outage probabilities $Q_{out}^{(1,0)}$ and $Q_{out}^{(2,1)}$ (in the links shown by solid arrows). After placing Relay 2, the agent measured the link qualities from the next location to the sink, Relay 1 and Relay 2 (since $B = 4$) and the algorithm advised him not to place a node there. Then the deployment agent moved to the next location (which is at a distance of 2δ from Relay 2) and measured the link qualities to Relay 1 and Relay 2 (but not to the sink since $B = 4$). In this *snap-shot* of the deployment process, the agent is evaluating the next location at $r = 3\delta$ distance from Relay 2 (see the dotted arrows). Since $B = 4$, the agent measures the link qualities from the current location to both Relay 1 and Relay 2; this corresponds to $\mathcal{I}_6 = \{3, 4\}$ (see Section 2.2 for the definition of $\mathcal{I}_6$), since the distances to Relay 2 and Relay 1 from the current location are 3δ and 4δ respectively. Based on these measurements, the deployment agent will decide whether to place a relay at $r = 3\delta$ or not, and the transmit power of the node in case the decision is to place; if the decision is not to place a relay here, then a relay must be placed at the next location (since $B = 4$), and the agent would be at a distance of $B\delta$ from the last placed relay (i.e., Relay 2).

2.3 Network Cost Minimization Objective

We first define the cost that we use to evaluate the performance of any deployment policy. A deployment policy π takes as input the distance of the current location of the agent from the previous relay and the link quality to the previously placed node, and provides the placement decision for that location and transmit power (if the decision is to place a relay) as output.

We denote the number of relays placed up to x steps from the sink by N_x, and let us define $N_0 = 0$. Since deployment decisions are based on measurements of (random) outage probabilities, $\{N_x\}_{x \geq 1}$ is a random process.

After the deployment is over, let us denote by Γ_i the transmit power used by node i, and by $Q_{out}^{(i,i-1)}$ the outage probability over the link $(i, i-1)$ (see Figure 2). Note that, Γ_i and $Q_{out}^{(i,i-1)}$ are random variables since shadowing between various potential location pairs are random variables, whose exact realization is known only after measurement. Given the measurement values (i.e., the information available to the deployment agent) and the deployment policy, one can find the exact realizations of Γ_i and $Q_{out}^{(i,i-1)}$.

The expected cost of the deployed network up to $x\delta$

distance is given by a sum of hop costs as follows:

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

which is the expectation (under policy π) of a linear combination of the sum power $\sum_{i=1}^{N_x}\Gamma_i$, the sum outage $\sum_{i=1}^{N_x}Q_{out}^{(i,i-1)}$, and the number of relays N_x. For small outage probabilities, the sum-outage $\sum_{i=1}^{N_x}Q_{out}^{(i,i-1)}$ is approximately equal to the probability that a packet sent from the point x to the sink encounters an outage along the path (see also Section 2.4 for a better understanding of the outage cost in light of the traffic model). The sum power $\sum_{i=1}^{N_x}\Gamma_i$ is proportional to the battery depletion rate in the network, in case wake-on radios are used (see [1, Section II] for a detailed discussion).

The multipliers $\xi_{out} \geq 0$ and $\xi_{relay} \geq 0$ capture the emphasis we place on $\sum_{i=1}^{N_x}Q_{out}^{(i,i-1)}$ or N_x. A large value of ξ_{out} will aim for deployment with smaller end-to-end expected outage. ξ_{relay} can be viewed as the cost of placing a relay.

Since the distance L to the source from the sink is not known prior to deployment, we simply assume that $L = \infty$. This assumption is practical when the distance of the source from the sink is large (e.g., deployment along a long forest trail). $L = \infty$ is also equivalent to the scenario where deployment is done serially along multiple trails in a forest, provided that the radio propagation environment in various trails are statistically identical; we deploy serially along multiple lines but use this formulation to minimize the per-step cost averaged over all the lines.

Next, we define the optimization problems that we seek to address in this paper.

2.3.1 The Unconstrained Problem

We seek to solve the following problem:

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

where Π is the set of all possible placement policies. We formulate (3) as an average cost MDP.

2.3.2 The Constrained Problem

(3) is the relaxed version of the following constrained problem:

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

Here we seek to minimize the mean power per step subject to constraints on the mean outage per step and the mean number of relays per step.

It turns out that (3) is the relaxed version of the constrained problem (4), with ξ_{out} and ξ_{relay} as the Lagrange multipliers. The constrained problem can be solved by solving the unconstrained problem, under proper choice of the Lagrange multipliers. The following theorem tells us how to choose the *Lagrange multipliers* ξ_{out} and ξ_{relay} (see [24], Theorem 4.3):

Theorem 1: For 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 if the constraints in (4) are met with equality under the policy π^*, then π^* is an optimal policy for the constrained problem (4) as well. □

2.4 Traffic Model

Motivated by our prior work reported in [17], [18], [9], [1], we assume that the traffic in the network is so light that there is only one packet in the network at a time; this model is called the "lone packet model" (or the *zero traffic* model). This model results in collision-free transmissions, since there are no simultaneous transmissions in the network. As a result, we can easily write down the communication cost in the line network as a sum of hop costs (Section 2.3).

It has been formally shown that network design under the lone packet model may be necessary for designing a network with positive traffic carrying capability (see [25, Section II]). We can easily adapt the result of [25, Section II] to show that, for a finite line network, if a target end-to-end packet delivery probability has to be achieved under positive traffic, then it is necessary to achieve that target under lone packet traffic. Now, the end-to-end packet error rate under lone packet traffic is approximately equal to the sum outage; this justifies the sum outage cost in (3) and the outage constraint in (4). Network design for a given positive traffic rate is left for future research.

In a line network, if interference-free communication is achieved via multi-channel access and frequency reuse after several hops, then the traffic model essentially becomes lone packet. There have been recent efforts to use multiple channels available in 802.15.4 radio in WSN; see [26], [27], [28], [29].

The lone packet traffic model is realistic for WSNs carrying low duty cycle measurements, or just an occasional alarm packet. For example, recently developed passive infra-red (PIR) sensor platforms can detect and classify human or animal intrusion ([30]); such sensors deployed in a forest generate very low data. The paper [6, Section 3.2] uses 1.1% duty cycle for a multi-hop WSN for wildlife monitoring; the sensors gather data from RFID collars tied the animals, and generate light traffic. Very light traffic model is also realistic for condition monitoring/industrial telemetry applications ([31]), where infrequent measurements are taken. Very light traffic model is also common in machine-to-machine communication ([32]). The paper [33, Table 1, Table 3] illustrate sensors with small sampling rate and sampled data size; it shows several bytes per second data rate requirement for habitat monitoring.

We assume that data packets traverse the network in a hop-by-hop fashion, without skipping any intermediate relay. Later we will explain in Section 3.4 why we do not consider the possibility of relay skipping in this paper; the reason is increased computational complexity without a very significant gain in network performance.

2.5 Extension to Bi-Directional Traffic Flow

Let us consider the situation where the traffic is still lone packet, but a packet can flow towards either direction along the line network with equal probabilities. In such cases, one can define the cost of link $(i, i-1)$ as $\Gamma_{i,forward}+\Gamma_{i-1,reverse}+\xi_{out}Q_{out}^{(i,i-1,forward)}+\xi_{out}Q_{out}^{(i-1,i,reverse)}+\xi_{relay}$, where $\Gamma_{i,forward}$ is the transmit power used from node i to node $(i-1)$, and $\Gamma_{i-1,reverse}$ is the transmit power used from node $(i-1)$ to node i. Similar meanings apply for the outage probabilities $Q_{out}^{(i,i-1,forward)}$ and $Q_{out}^{(i-1,i,reverse)}$, under transmit power levels $\Gamma_{i,forward}$ and $\Gamma_{i-1,reverse}$, respectively. It has to be noted that the shadowing between two potential locations in forward and reverse directions, $W_{forward}$ and $W_{reverse}$, may not necessarily be independent. But the shadowing random variable pair $(W_{forward}, W_{reverse}) \in \mathbb{R}_+^2$ between two potential locations have a joint distribution, and this pair assumes independent and identically distributed (i.i.d.) value in $\mathbb{R}_+^2$ if either the transmitter or the receiver is moved beyond the shadowing decorrelation distance (which is smaller than the step size δ). Hence, with this new link cost, our formulation (3) can easily be adapted to deploy a network carrying bi-directional traffic. In the process of deployment, the agent has to measure link qualities in both forward and reverse directions in such situation. The action at each step is to decide whether to place a relay; if the decision is to place a relay, then the agent also decides the transmit power levels used in that link along the forward and the reverse directions.

Since the design for bi-directional traffic carrying network is mathematically equivalent to the design for unidirectional traffic carrying network, *we will consider only unidirectional traffic for the rest of this paper.*

3 Formulation for Known Propagation Parameters

Throughout this section, we will assume that we seek to solve the unconstrained problem given in (3), and that the radio propagation parameters (such as η and the standard deviation σ for log-normal shadowing) are known prior to deployment. We formulate the problem as an average cost MDP, and develop a threshold policy for deployment. In the process, we also discover some interesting properties of the value function, which do not follow from the discounted cost formulation.

Note that, we assume throughout this section that measurement only to the immediately previous node is used to make a placement decision at any given location. Measurement to more than one previous nodes will be used later in order to develop the learning algorithms.

3.1 Markov Decision Process (MDP) Formulation

When the deployment agent is r steps away from the previous node ($r \in \{1, 2, \cdots, B\}$), the agent measures the outage probabilities $\{Q_{out}(r, \gamma, w)\}_{\gamma \in \mathcal{S}}$ on the link from the current location to the previous node,[2] where w is the realization of shadowing in that link. Then the algorithm decides whether to place a relay there, and also the transmit power $\gamma \in \mathcal{S}$ in case it decides to place a relay. We formulate the problem as an average cost MDP with state space $\{1, 2, \cdots, B\} \times \mathcal{W}$, where a typical state is of the form $(r, w), 1 \leq r \leq B, w \in \mathcal{W}$. If $r \leq B-1$, the action is either to place a relay and select a transmit power, or not to place. If $r = B$, the only feasible action is to place and select a transmit power $\gamma \in \mathcal{S}$. If a relay is placed at state (r, w) and if a transmit power γ is chosen for it, then a hop-cost of $\gamma + \xi_{out}Q_{out}(r, \gamma, w) + \xi_{relay}$ is incurred.[3]

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

3.2 Optimal Policy Based on Average Cost Optimality Equation

We will first derive the structure of an optimal policy based on the average cost optimality equation (ACOE). Let λ^* (or $\lambda^*(\xi_{out}, \xi_{relay})$) be the optimal average cost per step for the unconstrained problem (3) under the pure as-you-go deployment approach, and let $v^*(r, w)$ be the differential cost for the state (r, w), where $1 \leq r \leq B$ and $w \in \mathcal{W}$. The average cost optimality equation for our MDP is as follows (by the theory of [34, Chapter 4], for the case of finite $\mathcal{W}$, and by the theory developed in [35, Chapter 5], when $\mathcal{W}$ is a Borel subset of the real line):

$$\begin{aligned} v^*(r,w) &= \min\Big\{ \min_{\gamma\in\mathcal{S}}(\gamma + \xi_{out}Q_{out}(r,\gamma,w)) + \xi_{relay} - \lambda^* \\ &\quad + \sum_{w'} g(w')v^*(1,w'), -\lambda^* + \sum_{w'} g(w')v^*(r+1,w')\Big\} \\ &\quad \forall 1 \leq r \leq B-1 \\ v^*(B,w) &= \min_{\gamma\in\mathcal{S}}(\gamma + \xi_{out}Q_{out}(B,\gamma,w)) + \xi_{relay} - \lambda^* \\ &\quad + \sum_{w'} g(w')v^*(1,w') \end{aligned} \tag{5}$$

where $g(w)$ was defined (in Section 2.1) to be the probability mass function or probability density function of shadowing W.

The ACOE (5) can be explained as follows. When the state is (r, w), the deployment agent can either place or may not place a relay. If he places a relay, he will incur a stage cost of $\min_{\gamma\in\mathcal{S}}(\gamma + \xi_{out}Q_{out}(r, \gamma, w)) + \xi_{relay}$ and

2. Note that, for the time being, we will ignore the measurements made to other nodes from the set $\mathcal{I}_k$.

3. We have taken (r, w) as a typical state for the sake of simplicity in representation; for the channel model given by (1), we can also take $(r, \{Q_{out}(r, \gamma, w)\}_{\gamma\in\mathcal{S}})$ as a typical state, since the cost of an action depends on the state (r, w) only via the outage probabilities.

the next (random) state is $(1, W')$, where W' has p.m.f. or p.d.f. $g(w')$. If he does not place, then he incurs 0 cost at that step and the next state is $(r+1, W')$. When at state (B, w), he can only place a relay and incur a cost of $\min_{\gamma\in\mathcal{S}}(\gamma+\xi_{out}Q_{out}(B,\gamma,w))+\xi_{relay}$ at that stage and the next (random) state is $(1, W')$. Note that, $\min_{\gamma\in\mathcal{S}}$ appears in the single-stage cost because choice of transmit power of the placed node is also a part of the action, and a transmit power is chosen so that the single-stage cost for a placed relay is minimized.

Note that, by multiplying both sides of (5) with $g(w)$ and taking summation over w, we obtain the following:

$$\begin{aligned} V(r) &= \mathbb{E}_W \min\Big\{ \min_{\gamma\in\mathcal{S}}(\gamma+\xi_{out}Q_{out}(r,\gamma,W))+\xi_{relay}-\lambda^* \\ &\quad +V(1), -\lambda^*+V(r+1)\Big\} \forall 1\le r\le B-1 \\ V(B) &= \mathbb{E}_W \min_{\gamma\in\mathcal{S}}(\gamma+\xi_{out}Q_{out}(B,\gamma,W))+\xi_{relay}-\lambda^*+V(1) \end{aligned} \tag{6}$$

where $V(r)=\sum_w g(w)v^*(r,w)\forall 1\le r\le B$. Now, it is easy to see that if any $V(\cdot)$ satisfies (6), then $V(\cdot)+c$ for any constant number c also satisfies (6). Hence, we can put $V(1)=\lambda^*$ in (6) and obtain:

$$\begin{aligned} V(r) &= \mathbb{E}_W \min\Big\{ \min_{\gamma\in\mathcal{S}}(\gamma+\xi_{out}Q_{out}(r,\gamma,W))+\xi_{relay}, \\ &\quad V(r+1)-V(1)\Big\} \forall 1\le r\le B-1 \\ V(B) &= \mathbb{E}_W \min_{\gamma\in\mathcal{S}}(\gamma+\xi_{out}Q_{out}(B,\gamma,W))+\xi_{relay} \end{aligned} \tag{7}$$

Remark: Let $c(r,W):=\min_{\gamma\in\mathcal{S}}(\gamma+\xi_{out}Q_{out}(r,\gamma,W))+\xi_{relay}$ be the (random) cost incurred if we place a relay at a distance r from the previous relay. (7) shows the criteria for optimality to be $V(r)=\mathbb{E}_W\min\{c(r,W), V(r+1)-V(1)\}$ for $r\le B-1$ and $V(B)=\mathbb{E}_W c(B,W)$. We will see in Algorithm 1 that, by solving this system of (nonlinear) equations, one can find the optimal policy; there is no need to compute the differential cost for each state explicitly. Also, (7) will be particularly useful when we develop online deploy-and-learn algorithms in later sections, using the theory of stochastic approximation.

Theorem 2: There exists a unique vector $\underline{V}^*=[V^*(1), V^*(2), \cdots, V^*(B)]^T$ satisfying (7). Also, $V^*(r)\ge rV^*(1)$ for all $r\in\{1,2,\cdots,B-1\}$ and $V^*(r)$ is increasing in r.

Proof: See Appendix A. □

3.2.1 Policy Structure

Algorithm 1 specifies the optimal decision when the agent is r steps away from the previously placed node and the shadowing realization from the current location to the previously placed node is w.

Theorem 3: The policy given by Algorithm 1 is optimal for the unconstrained problem in (3). The threshold $c_{th}(r)$ is increasing in r.

Proof: From (5), the optimal policy is to place a relay at state (r,w) if the cost of placing is less than the cost of not placing. Hence, the policy structure follows from equations (5), (6) and (7). $c_{th}(r)$ is increasing in r since $V^*(r+1)$ is increasing in r. □

Input: ξ_{out}, ξ_{relay}, $\underline{V}^*$.
Output: Placement decision at each step.
Pre-compute: The threshold values
$c_{th}(r):=V^*(r+1)-V^*(1)$ for all $1\le r\le B-1$.
Initialization: $r=1$ (distance from the previous node)
while $1\le r\le B$ **do**
  Measure $Q_{out}(r,\gamma,w)\forall\gamma\in\mathcal{S}$;
  if $r\le B-1$ *and*
   $\min_{\gamma\in\mathcal{S}}(\gamma+\xi_{out}Q_{out}(r,\gamma,w))+\xi_{relay}\le c_{th}(r)$
   then
    Place a new relay and use transmit power
     $\arg\min_{\gamma\in\mathcal{S}}(\gamma+\xi_{out}Q_{out}(r,\gamma,w))$;
    Move to next step and set $r=1$;
  else if $r\le B-1$ *and*
   $\min_{\gamma\in\mathcal{S}}(\gamma+\xi_{out}Q_{out}(r,\gamma,w))+\xi_{relay}> c_{th}(r)$
   then
    Do not place a relay and move to next step;
    $r=r+1$;
  else
    Place a new relay (since $r=B$);
    Use transmit power
     $\arg\min_{\gamma\in\mathcal{S}}(\gamma+\xi_{out}Q_{out}(B,\gamma,w))$;
    Move to next step;
    Set $r=1$.
  end
end

Algorithm 1: OptAsYouGo Algorithm

We denote the optimal policy given by Algorithm 1 by $\pi^*(\xi_{out},\xi_{relay})$.

3.3 Some properties of the optimal cost

Let us consider a sub-class of stationary deployment policies (parameterized by $\underline{V}$, $\xi_{out}\ge 0$ and $\xi_{relay}\ge 0$) where $\underline{V}^*(\cdot)$ in Algorithm 1 is replaced by any vector $\underline{V}$. Under this sub-class of policies, let us denote by $(U_k,\Gamma_k,Q_{out}^{(k,k-1)}), k\ge 1$, the sequence of inter-node distances, transmit powers and link outage probabilities (see Figure 2). Since shadowing is i.i.d. across links, the deployment process probabilistically restarts after each relay placement. Hence, $(U_k,\Gamma_k,Q_{out}^{(k,k-1)}), k\ge 1$, is an i.i.d. sequence. Let $\overline{\Gamma}(\underline{V},\xi_{out},\xi_{relay})$, $\overline{Q}_{out}(\underline{V},\xi_{out},\xi_{relay})$ and $\overline{U}(\underline{V},\xi_{out},\xi_{relay})$ denote the mean power per link, mean outage per link and mean placement distance (in steps) respectively, under this sub-class of policies. We denote by $\overline{\Gamma}^*(\xi_{out},\xi_{relay})$, $\overline{Q}^*_{out}(\xi_{out},\xi_{relay})$ and $\overline{U}^*(\xi_{out},\xi_{relay})$ the optimal mean power per link, the optimal mean outage per link and the optimal mean placement distance (in steps) respectively, under Algorithm 1, where $\underline{V}^*$ is used instead of any general $\underline{V}$.

Now, the optimal mean power per step, the optimal mean outage per step, and the optimal mean number of relays per step are given by $\frac{\overline{\Gamma}^*(\xi_{out},\xi_{relay})}{\overline{U}^*(\xi_{out},\xi_{relay})}$, $\frac{\overline{Q}^*_{out}(\xi_{out},\xi_{relay})}{\overline{U}^*(\xi_{out},\xi_{relay})}$ and $\frac{1}{\overline{U}^*(\xi_{out},\xi_{relay})}$ (by the Renewal-Reward theorem).

Theorem 4: The optimal average cost per step for problem (3), $\lambda^*(\xi_{out}, \xi_{relay})$, is concave, increasing and Lipschitz continuous in $\xi_{out} \geq 0$, $\xi_{relay} \geq 0$.

Proof: See Appendix A. □

Theorem 5: $\underline{V}^* = (V^*(1), V^*(2), \cdots, V^*(B))$ is Lipschitz continuous in (ξ_{out}, ξ_{relay}).

Proof: See Appendix A. □

Theorem 6: For a given ξ_{out}, the mean number of relays per step under Algorithm 1, $\frac{1}{\overline{U}^*(\xi_{out},\xi_{relay})}$, decreases with ξ_{relay}. Similarly, for a given ξ_{relay}, the optimal mean outage per step, $\frac{\overline{Q}^*_{out}(\xi_{out},\xi_{relay})}{\overline{U}^*(\xi_{out},\xi_{relay})}$, decreases with ξ_{out}.

Proof: The proof is exactly same as the proof of [1, Theorem 5]. □

3.4 A note on the objective function in (3)

Even though the deployment policy developed in this section uses only the measurements made to the immediately previous placed node in order to make a placement location, we will see in subsequent sections that measurements to all placed relay nodes located within B steps from the current location of the agent will be used for on-line learning of the optimal deployment policy. A question that naturally arises is whether we can do better with the additional measurements (when the propagation parameters are known and the optimal policy can be computed prior to deployment); this might require skipping some already placed relay nodes after the deployment is over. The possibility of relay skipping was considered in [9]; in the current paper, we briefly describe a similar formulation in our context and explain why we rule out the possibility of relay skipping.

Let us consider deployment up to x steps. After the deployment is over, we construct a directed acyclic graph over the deployed nodes (including the sink) as follows. Links are all directed edges from each node to every node with smaller index and located within a distance of B steps. Hence, if i and j are two nodes with $i > j$ and $\sum_{k=j+1}^{i} U_k \leq B$, there is a link (i, j) between them. Consider all directed acyclic paths from node N_x to the sink over this graph. Let us denote by $\mathbf{p}$ any arbitrary directed acyclic path, and by $\mathcal{E}(\mathbf{p})$ the set of (directed) links of the path $\mathbf{p}$. We also define $\mathcal{P}_x := \{\mathbf{p} : (i,j) \in \mathcal{E}(\mathbf{p}) \implies N_x \geq i > j \geq 0, \sum_{k=j+1}^{i} U_k \leq B\}$. Let us denote a generic link (edge) on this graph by e, and the transmit power and outage probability on edge e by $\Gamma^{(e)}$ and $Q^{(e)}_{out}$.

Let us consider the following problem:

$$\min_{\pi\in\Pi} \lim \sup_{x\to\infty} \frac{\mathbb{E}_\pi\left(\min_{\mathbf{p}\in\mathcal{P}_x} \sum_{e\in\mathcal{E}(\mathbf{p})} \left(\Gamma^{(e)} + \xi_{out} Q^{(e)}_{out}\right) + \xi_{relay} N_x\right)}{x} \tag{8}$$

We call $\sum_{e\in\mathcal{E}(\mathbf{p})} \left(\Gamma^{(e)} + \xi_{out} Q^{(e)}_{out}\right)$ the length of the path $\mathbf{p}$, and $\min_{\mathbf{p}\in\mathcal{P}_x} \sum_{e\in\mathcal{E}(\mathbf{p})} \left(\Gamma^{(e)} + \xi_{out} Q^{(e)}_{out}\right)$ the length of the shortest path.

Formulation of problem (8) as an MDP will require as the typical state the distance of all nodes located within B steps from the current location, the realization of shadowing to all these nodes (through the measured outage probabilities), and the lengths of the shortest paths from all these nodes to the sink. A similar situation was considered in [9]. It turns out that the state space becomes very large (the number of all possible lengths of shortest paths grows to ∞ as $x \to \infty$, even when the set $\mathcal{W}$ of possible values of shadowing is finite), and the policy computation becomes numerically intensive; but the numerical results of [9] show that the margin of performance improvement achieved via this formulation (instead of the formulation used earlier in this section) is not significant. Hence, in this paper, we only consider formulation (3) and proceed with it.

4 OptAsYouGoLearning: Learning with Deployment for Given Multipliers

Note that, for any given values of ξ_{out} and ξ_{relay}, the optimal policy given by Algorithm 1 can be completely specified by the vector $\underline{V}^*$. But, the computation of $\underline{V}^*$ requires the agent to solve a system of nonlinear equations (which is computationally intensive), and these nonlinear equations can be specified only when the channel model parameters (e.g., path-loss exponent η and standard deviation σ for log-normal shadowing) are known apriori. However, in practice, these parameters may not be available prior to deployment. Under this situation, the deployment agent has to *learn* the optimal policy as deployment progresses, and use the corresponding updated policy at each step to make a placement decision. In order to address this requirement, we propose an algorithm which will maintain a running estimate of $\underline{V}^*$, and update this estimate at each step (using new measurements made at each step). Using the theory of Asynchronous Stochastic Approximation (see [36]), we show that, as the number of deployed relays goes to infinity, the running estimate converges to $\underline{V}^*$ almost surely. From (7) (and the notation defined immediately after (7)), we see that the optimal $\underline{V}^*$ is the unique real zero of the system of equations: $\mathbb{E}_W \min\{c(r,W), V(r+1) - V(1)\} - V(r) = 0$ for $r \leq B-1$ and $\mathbb{E}_W\, c(B,W) - V(B) = 0$. We use asynchronous stochastic approximation so that the iterates $\{\underline{V}^{(k)}\}_{k\geq 0}$ converge asymptotically to this unique zero.

4.1 OptAsYouGoLearning Algorithm

Suppose that the deployment agent is standing k steps away from the sink node. At the k-th step, the agent makes a placement decision and then performs a learning operation. Let us recall the deployment process (see Section 2.2 and Figure 2) and notation: $\mathcal{I}_k := \{r \in \{1, 2, \cdots, B\} :$

a relay was placed at a distance $(k-r)\delta$ from the sink}}. For the learning operation, $\mathcal{I}_k \subset \{1, \cdots, B\}$ denotes the set of the values of r for which links from the current location to the placed relay r steps backwards are measured, and for which $V(r)$ is updated, when the agent is at a distance $k\delta$ from the sink. Clearly, for each $k \geq 1$, $\mathcal{I}_k$ is a random set. Let us denote by $\underline{V}^{(k)}$ the estimate of $\underline{V}^*$ after an update (i.e., a learning operation) is made at the k-th step from the sink. At step k (after a placement decision is made), $V^{(k-1)}(r)$ for $r \in \mathcal{I}_k$ is updated to $V^k(r)$, and it is not updated for $r \notin \mathcal{I}_k$ (which means that $V^{(k)}(r) = V^{(k-1)}(r)$ for $r \notin \mathcal{I}_k$). Let us define $\nu(r,k) := \sum_{i=1}^{k} \mathbb{I}\{r \in \mathcal{I}_i\}$ the number of times the estimate of $V^*(r)$ is updated up to the k-th step.

Note that, Algorithm 1 requires the agent to measure link quality only to the previous node, whereas the learning algorithm presented in this section involves link quality measurement to more than one previous nodes (unlike our prior paper [1]). *This is necessary because, if we make measurement only to last relay, then, depending on the initial estimate $\underline{V}^{(0)}$, there could arise a situation that the inter-relay distance never equals to B steps in the entire deployment process, which implies that $V^{(0)}(B)$ will never be updated, thereby converging to an unintended policy. Making measurements to all previously placed nodes located at distance less than $B\delta$ from the current location ensures that $\liminf_{k\to\infty} \frac{\nu(r,k)}{k} > 0$ almost surely, which is required for the convergence proof.*

The OptAsYouGoLearning algorithm is provided in Algorithm 2.

Theorem 7: Under Algorithm 2, $V^{(k)}(r) \to V^*(r)$ almost surely for all $1 \leq r \leq B$.

Proof: See Appendix B. □

Discussion of Algorithm 2:

(i) *The basic idea:* From (7) (and the notation defined immediately after (7)), we see that the optimal $\underline{V}^*$ is the unique real zero of the system of equations: $\mathbb{E}_W \min\{c(r,W), V(r+1) - V(1)\} - V(r) = 0$ for $r \leq B-1$ and $\mathbb{E}_W\, c(B,W) - V(B) = 0$. We use asynchronous stochastic approximation so that the iterates converge asymptotically to this unique zero.

(ii) *Asynchronous stochastic approximation:* In standard stochastic approximation techniques, all iterates are updated at the same time. However, the pure as-you-go deployment scheme does not allow the deployment agent to update all iterates at each step. Since only a subset $\mathcal{I}_k \subset \{1, \cdots, B\}$ of iterates can be updated at step k, we have to use *asynchronous* stochastic approximation.

(iii) The proof of Theorem 7 exhibits a nice separation between the estimation and control. In other words, the iterates will asymptotically converge to $\underline{V}^*$ (and the policy will converge to the optimal policy) even when the placement decisions are not made according to the proposed threshold policy (but the measurement and update scheme should be unchanged); but it may not yield the optimal cost for problem (3) since we do not use the optimal policy at each stage. However, this nice separation property will not hold in next section when we vary ξ_{out} and ξ_{relay} in order to solve the constrained problem (4).

(iv) Note that, since the state space of the MDP in Section 3 is large (potentially infinite and even uncountable), it will not be easy to use traditional Q-learning

---

Input: ξ_{out}, ξ_{relay}, and a decreasing positive sequence $\{a(n)\}_{n\geq 1}$ such that $\sum_{n=1}^{\infty} a(n) = \infty$, $\sum_{n=1}^{\infty} a^2(n) < \infty$.
Output: Placement decision at each step.
Initialization: $r' = 1$ (distance from the previous node), $k = 1$ (distance of the current location from the sink), initial estimate $\underline{V}^{(0)}$.
while $1 \leq r' \leq B$ **do**
  Find $\mathcal{I}_k := \{r \in \{1,2,\cdots,B\}$: relay placement at $(k-r)\delta$ distance from sink}};
  Find $\nu(r,k) := \sum_{i=1}^{k} \mathbb{I}\{r \in \mathcal{I}_i\} \forall r \in \{1,2,\cdots,B\}$;
  Measure $Q_{out}(r,\gamma,w_r) \forall \gamma \in \mathcal{S}, r \in \mathcal{I}_k$;
  if $r' \leq B-1$ *and* $\min_{\gamma\in\mathcal{S}}(\gamma + \xi_{out} Q_{out}(r',\gamma,w_{r'})) + \xi_{relay} \leq -V^{(k-1)}(1) + V^{(k-1)}(r'+1)$ **then**
    Place a new relay and use transmit power $\arg\min_{\gamma\in\mathcal{S}}(\gamma + \xi_{out} Q_{out}(r',\gamma,w_{r'}))$;
    Do the following updates:

$$
\begin{aligned}
& V^{(k)}(r) \\
= \; & V^{(k-1)}(r) + a(\nu(r,k))\mathbb{I}\{r \in \mathcal{I}_k\}\Big[\min\Big\{\min_{\gamma}(\gamma + \xi_{out} Q_{out}(r,\gamma,w_r)) + \xi_{relay}, -V^{(k-1)}(1) \\
& + V^{(k-1)}(r+1)\Big\} - V^{(k-1)}(r)\Big], \forall 1 \leq r \leq B-1 \\
& V^{(k)}(B) \\
= \; & V^{(k-1)}(B) + a(\nu(B,k))\mathbb{I}\{B \in \mathcal{I}_k\}\Big[\min_{\gamma}(\gamma + \\
& \xi_{out} Q_{out}(B,\gamma,w_B)) + \xi_{relay} - V^{(k-1)}(B)\Big] \qquad (9)
\end{aligned}
$$

    Move to next step and set $r' = 1$;
  else if $r' \leq B-1$ *and* $\min_{\gamma\in\mathcal{S}}(\gamma + \xi_{out} Q_{out}(r',\gamma,w_{r'})) + \xi_{relay} > -V^{(k-1)}(1) + V^{(k-1)}(r'+1)$ **then**
    Do not place, do the same updates as (9);
    Move to next step and do $r' = r' + 1$;
  else
    Place a new relay (since $r' = B$);
    Use transmit power $\arg\min_{\gamma\in\mathcal{S}}(\gamma + \xi_{out} Q_{out}(B,\gamma,w_B))$;
    Do the same updates as (9);
    Move to next step and set $r' = 1$.
  end
  k=k+1;
end

Algorithm 2: OptAsYouGoLearning Algorithm

algorithms. In fact, all the state action-pairs in a Q-learning algorithm need to repeat comparably often over infinite time horizon to guarantee the desired convergence, but this may not happen in case of infinite state space (arising out of infinite $\mathcal{W}$). On the other hand, Algorithm 2 provides a learning algorithm with provable convergence guarantee while having only B number of iterates.

5 OptAsYouGoAdaptiveLearning for the Constrained problem

In Section 4, we provided a deploy-and-learn algorithm for given ξ_{out} and ξ_{relay}. However, 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 constrained problem (4). But we need to know the radio propagation parameters (e.g., η and σ) in order to compute a pair $(\xi^*_{out}, \xi^*_{relay})$ that satisfies the condition given in Theorem 1. In practice, these parameters may not be known. Hence, we provide a sequential placement algorithm such that, as deployment progresses, the placement policy (updated at each step) converges to the set of optimal policies for the constrained problem (4). We modify the OptAsYouGoLearning algorithm so that a running estimate $(\underline{V}^{(k)}, \xi^{(k)}_{out}, \xi^{(k)}_{relay})$ gets updated at each step, and asymptotically converges to the set of optimal $(\underline{V}^*(\xi_{out}, \xi_{relay}), \xi_{out}, \xi_{relay})$ tuples. This algorithm is based on two time-scale stochastic approximation (see [37, Chapter 6]).

5.1 Some Useful Notation and Assumptions

In this subsection, we will introduce some assumptions and notation (these were provided in [1, Section VII], but are repeated here for completeness).

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

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

□

Note that, the pair $(\overline{q}, \overline{N})$ can be infeasible. For example, if $\overline{N} = \frac{1}{B}$ (i.e., inter-node distance is B) and $\overline{q} < \frac{\mathbb{E}_W Q_{out}(B, P_M, W)}{B}$ (P_M is the maximum available transmit power), the outage constraint cannot be satisfied while meeting the constraint on the mean number of relays per step, even by using the maximum transmit power P_M.

$\mathcal{K}(\overline{q}, \overline{N})$ is empty if $(\overline{q}, \overline{N})$ is infeasible. *In this paper, we assume that $\mathcal{K}(\overline{q}, \overline{N})$ is non-empty (i.e., $(\overline{q}, \overline{N})$ is a feasible pair), which is true for feasible pairs of $\mathcal{K}(\overline{q}, \overline{N})$:*

Assumption 1: $\overline{q}$ and $\overline{N}$ are such that there exists at least one pair $\xi^*_{out} \geq 0, \xi^*_{relay} \geq 0$ such that $(\underline{V}^*(\xi^*_{out}, \xi^*_{relay}), \xi^*_{out}, \xi^*_{relay}) \in \mathcal{K}(\overline{q}, \overline{N})$. □

Assumption 2: The probability density function (p.d.f.) of the shadowing random variable W is continuous over $(0, \infty)$; i.e., $\mathbb{P}(W = w) = 0$ for any $w \in (0, \infty)$ (e.g., log-normal shadowing). □

Theorem 8: Under Assumption 2 and Algorithm 1, the optimal mean power per step $\frac{\overline{\Gamma}^*(\xi_{out}, \xi_{relay})}{\overline{U}^*(\xi_{out}, \xi_{relay})}$, the optimal mean placement rate $\frac{1}{\overline{U}^*(\xi_{out}, \xi_{relay})}$ and the optimal mean outage per step $\frac{\overline{Q}^*_{out}(\xi_{out}, \xi_{relay})}{\overline{U}^*(\xi_{out}, \xi_{relay})}$, are continuous in (ξ_{out}, ξ_{relay}).

Proof: See Appendix C. □

Remark: Theorem 8 implies that there is no need to do any randomization among deterministic policies (unlike [38]) in order to meet the constraints with equality.

5.2 OptAsYouGoAdaptiveLearning Algorithm

The basic idea behind this algorithm (Algorithm 3; see next page) is to vary $\xi^{(k)}_{out}$ and $\xi^{(k)}_{relay}$ at a much slower rate than $\underline{V}^{(k)}$, as if $\xi^{(k)}_{out}$ and $\xi^{(k)}_{relay}$ are varied in an outer loop and $\underline{V}^{(k)}$ is varied in an inner loop. If the outage in a newly created link is larger than the budgeted outage for a link with that length, then ξ_{out} is increased with the hope that subsequent links will have smaller outage; the opposite is done in case the outage in a newly created link is smaller. On the other hand, if a newly created link is shorter than $\frac{1}{\overline{N}}$, then ξ_{relay} is increased, otherwise it is decreased.

Notation in Algorithm 3: $\Lambda_{[0,A_1]}(x)$ denotes the projection of x on the interval $[0, A_1]$. Let the power, outage and link length of the new relay (if placed) at the k-th step be Γ_{N_k}, $Q^{(N_k, N_k-1)}_{out}$ and U_{N_k} (recall that N_k is the number of nodes placed up to the k-th step). Note that, $\mathbb{I}\{N_k = N_{k-1} + 1\}$ is the indicator that a relay is placed at the k-th step.

Theorem 9: Under Assumption 1, Assumption 2 and under proper choice of A_1 and A_2, we have $(\underline{V}^{(k)}, \xi^{(k)}_{out}, \xi^{(k)}_{relay}) \rightarrow \mathcal{K}(\overline{q}, \overline{N})$ almost surely for Algorithm 3.

Proof: See Appendix C.

We complete the proof in four steps. First, we show that the difference between $\underline{V}^{(k)}$ and $\underline{V}^*(\xi^{(k)}_{out}, \xi^{(k)}_{relay})$ converges to 0 almost surely. This proves the desired convergence in the faster timescale. Next, we pose the slower timescale iteration as a projected stochastic approximation iteration (see [39, Equation 5.3.1]). Next, we show that the slower timescale iteration satisfies some conditions given in [39] (see [39, Theorem 5.3.1]). Finally, we argue (using Theorem 5.3.1 of [39]) that the slower timescale iterates converge to the set of stationary points of a suitable ordinary differential equation.

It is to be noted that while the proof to some extent follows the outline of the proof of [1, Theorem 12], significantly new nontrivialities arise in our work as compared to the proof of [1, Theorem 12]. For example, we had to prove the boundedness of the faster timescale iterates separately, since the asynchronous updates in the

Input: Two positive numbers A_1 and A_2 appropriately chosen, two decreasing positive sequences $\{a(n)\}_{n\geq 1}$ and $\{b(n)\}_{n\geq 1}$ such that $\sum_{n=1}^{\infty} a(n) = \infty$, $\sum_{n=1}^{\infty} a^2(n) < \infty$, $\sum_{n=1}^{\infty} b(n) = \infty$, $\sum_{n=1}^{\infty} b^2(n) < \infty$ and $\lim_{n\to\infty} \frac{b(\lfloor \frac{n}{B} \rfloor)}{a(n)} = 0$.
Output: Placement decision at each step.
Initialization: $r' = 1$ (distance from the previous node), $k = 1$ (distance of the current location from the sink), initial estimates $\underline{V}^{(0)}$, $\xi_{out}^{(0)}$, $\xi_{relay}^{(0)}$.
while $1 \leq r' \leq B$ **do**
  Find $\mathcal{I}_k := \{r \in \{1, 2, \cdots, B\} :$ relay placed at $(k-r)\delta$ distance from sink$\}\}$;
  Find $\nu(r,k) := \sum_{i=1}^{k} \mathbb{I}\{r \in \mathcal{I}_i\} \forall r \in \{1, 2, \cdots, B\}$;
  Measure $Q_{out}(r, \gamma, w_r) \forall \gamma \in \mathcal{S}, r \in \mathcal{I}_k$;
  if $r' \leq B-1$ *and* $\min_{\gamma\in\mathcal{S}}(\gamma + \xi_{out}^{(k-1)} Q_{out}(r', \gamma, w_{r'})) + \xi_{relay}^{(k-1)} \leq -V^{(k-1)}(1) + V^{(k-1)}(r'+1)$ **then**
    Place a new relay and use transmit power $\arg\min_{\gamma\in\mathcal{S}}(\gamma + \xi_{out}^{(k-1)} Q_{out}(r', \gamma, w_{r'}))$;
    Do the following updates:

$$
\begin{aligned}
&V^{(k)}(r) = V^{(k-1)}(r) + a(\nu(r,k))\mathbb{I}\{r \in \mathcal{I}_k, r < B\} \\
&\Big[\min\Big\{\min_{\gamma}(\gamma + \xi_{out}^{(k-1)} Q_{out}(r, \gamma, w_r)) + \xi_{relay}^{(k-1)}, \\
&-V^{(k-1)}(1) + V^{(k-1)}(r+1)\Big\} - V^{(k-1)}(r)\Big] \\
&V^{(k)}(B) = V^{(k-1)}(B) + a(\nu(B,k))\mathbb{I}\{B \in \mathcal{I}_k\} \\
&\Big[\min_{\gamma}(\gamma + \xi_{out}^{(k-1)} Q_{out}(B, \gamma, w_B)) + \xi_{relay}^{(k-1)} \\
&-V^{(k-1)}(B)\Big] \\
&\xi_{out}^{(k)} = \Big[\xi_{out}^{(k-1)} + b(N_k)\mathbb{I}\{N_k = N_{k-1}+1\} \\
&\Big(Q_{out}^{(N_k, N_k-1)} - \overline{q} U_{N_k}\Big)\Big]_0^{A_1} \\
&\xi_{relay}^{(k)} = \Big[\xi_{relay}^{(k-1)} + b(N_k)\mathbb{I}\{N_k = N_{k-1}+1\} \\
&\Big(1 - \overline{N} U_{N_k}\Big)\Big]_0^{A_2}
\end{aligned}
\tag{10}
$$

    Move to next step and set $r' = 1$;
  else if $r' \leq B-1$ *and* $\min_{\gamma\in\mathcal{S}}(\gamma + \xi_{out}^{(k-1)} Q_{out}(r', \gamma, w_{r'})) + \xi_{relay}^{(k-1)} > -V^{(k-1)}(1) + V^{(k-1)}(r'+1)$ **then**
    Do not place, and perform updates as in (10);
    Move to next step and set $r' = r' + 1$;
  else
    Place a new relay (since $r' = B$);
    Use power $\arg\min_{\gamma\in\mathcal{S}}(\gamma + \xi_{out}^{(k-1)} Q_{out}(B, \gamma, w_B))$;
    Do the same updates as (10);
    Move to next step and set $r' = 1$.
  end
  k=k+1;.
end

Algorithm 3: OptAsYouGoAdaptiveLearning

faster timescale do not allow us to mimic the proof of [1, Theorem 12]. Similarly there are many other steps which require significant novel additional mathematical analysis compared to [1, Theorem 12]. Hence, in this proof, we proved intermediate results wherever necessary, and skipped some steps if they follow from the proof of [1, Theorem 12]. □

Choice of A_1 and A_2: A_1 and A_2 need to be chosen carefully, otherwise the iterates $(\xi_{out}^{(k)}, \xi_{relay}^{(k)})$ may converge to undesired points on the boundary of $[0, A_1] \times [0, A_2]$. In general, a stationary point on the boundary of $[0, A_1] \times [0, A_2]$ may not correspond to a point in $\mathcal{K}(\overline{q}, \overline{N})$. Hence, we borrow a scheme from [1] for choosing A_1 and A_2 which ensures that, if $(\xi'_{out}, \xi'_{relay})$ is a stationary point of the o.d.e., then $(\underline{V}^*(\xi'_{out}, \xi'_{relay}), \xi'_{out}, \xi'_{relay}) \in \mathcal{K}(\overline{q}, \overline{N})$. The number A_1 has to be chosen so large that, for all $u \in \{1, 2, \cdots, B\}$, we will have $\mathbb{P}(\arg\min_{\gamma\in\mathcal{S}}(\gamma + A_1 Q_{out}(u, \gamma, W)) = P_M) > 1 - \kappa$ for some small enough $\kappa > 0$. We also need the condition that $\frac{\overline{Q}_{out}^*(A_1,0)}{\overline{U}^*(A_1,0)} \leq \overline{q}$. The number A_2 has to be chosen so large that, for any $\xi_{out} \in [0, A_1]$, we will have $\overline{U}^*(\xi_{out}, A_2) > \frac{1}{\overline{N}}$ (when $\frac{1}{\overline{N}} < B$). The numbers A_1 and A_2 have to be chosen so large that there exists at least one pair $(\xi'_{out}, \xi'_{relay})$ for which $(\underline{V}^*(\xi'_{out}, \xi'_{relay}), \xi'_{out}, \xi'_{relay}) \in \mathcal{K}(\overline{q}, \overline{N})$. □

Discussion of Algorithm 3:

(i) *Two timescales:* The update scheme (10) is based on two-timescale stochastic approximation (see [37, Chapter 6]). Since $\lim_{n\to\infty} \frac{b(\lfloor \frac{n}{B} \rfloor)}{a(n)} = 0$, we can say that ξ_{out} and ξ_{relay} are adapted in a *slower* timescale, and $\underline{V}$ is updated in a *faster* timescale, as if ξ_{out} and ξ_{relay} are updated in a slow outer loop, and, $\underline{V}$ is updated in an inner loop.

(ii) *Structure of the iteration:* The slower timescale iteration involves updating ξ_{out} and ξ_{relay} based on whether the corresponding constraints are violated in a link (after placing a relay); if a constraint is violated by a newly created link, then the corresponding Lagrange multiplier is increased to counterbalance it in subsequent node placements. The goal is to meet both constraints with equality (if possible) in the long run.

(iii) *Asymptotic behaviour of the iterates:* If $\overline{q} > \frac{\mathbb{E}_W Q_{out}(B, P_1, W)}{B}$; we will have $\xi_{out}^{(k)} \to 0$; here the policy places all the relays at the B-th step and uses the smallest power P_1 at each node. If the constraints are not feasible, then either $\xi_{out}^{(k)} \to A_1$ or $\xi_{relay}^{(k)} \to A_2$ or both happens.
Simulation results show that $\mathcal{K}(\overline{q}, \overline{N})$ has only one tuple in case the pair $(\overline{q}, \overline{N})$ is feasible. □

5.3 Asymptotic Performance of Algorithm 3

Though Algorithm 3 induces a nonstationary policy, Theorem 9 states that the sequence of policies generated by Algorithm 3 converges to the set of optimal stationary, deterministic policies for the constrained problem (4).

Let π_{oaygal} denote the (nonstationary) deployment policy induced by Algorithm 3.

Theorem 10: Under Assumption 1, Assumption 2 and proper choice of A_1 and A_2, we have:

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

Proof: The proof is similar to [1, Theorem 13]. □

6 Convergence Speed of Learning Algorithms: A Simulation Study

In this section, we provide a simulation study for the convergence rate of Algorithm 2 and Algorithm 3.

6.1 Parameter Values Used in the Simulation

For simulation, we consider a deployment environment similar to that considered in [1, Section VIII]. The details of the simulation environment are provided below.

We assume that deployment is done with iWiSe motes ([23]) equipped with 9 dBi antennas. $\mathcal{S}$, the set of transmit power levels, is taken to be $\{-18, -7, -4, 0, 5\}$ dBm, which is a subset of available transmit power levels for iWiSe motes. Under the channel model as given by (1), our measurements in a forest-like environment gave $\eta = 4.7$ and $c = 10^{0.17}$ (i.e., 1.7 dB); the experimental details can be found in [10]. From the statistical analysis of the measurement data, we also showed that shadowing W follows log-normal distribution in such a forest-like environment; $W = 10^{\frac{Y}{10}}$ with $Y \sim \mathcal{N}(0, \sigma^2)$, where $\sigma = 7.7$ dB was obtained from our data analysis. Shadowing decorrelation distance was calculated as 6 meters; hence we consider deployment with $\delta = 20$ meter. The fading turned out to be Rayleigh fading.

Outage is defined to be the event when a packet is received at a power level below $P_{rcv-min} = 10^{-9.7}$ mW (i.e., -97 dBm); for a commercial implementation of IEEE 802.15.4, received power -97 dBm results in a 2% packet loss probability for 127 byte packets for iWiSe motes (obtained from measurements).

We choose B in the following way. We define a link to be workable if it has an outage probability less than 3%. B is chosen to be the largest integer such that the probability of finding a workable link of length $B\delta$ is greater than 20%, under 5 dBm transmit power. For the parameters $\eta = 4.7$ and $\sigma = 7.7$ dB, and 5 dBm transmit power, B turned out to be 5.

It is important to note that, the radio propagation parameters (e.g., η and σ) and modeling assumptions (e.g., log-normal shadowing) are obtained and validated using field data collected via extensive measurements in a forest-like environment; the details of these experiments can be found in [10]. Hence, in this paper, we evaluate our algorithms only via MATLAB simulation of an environment that has radio propagation model and parameters obtained from experiments in [10]. This is done by generating random channel gains in MATLAB, for the wireless links that need to be measured in course of the deployment process.

The performance variation of OptAsYouGo algorithm with (ξ_{out}, ξ_{relay}) has been demonstrated numerically in [1, Section V, Appendix C], which comply with Theorem 4 and Theorem 6.

6.2 OptAsYouGoLearning for Given Multipliers

Here we study the rate of convergence for OptAsYouGoLearning with $\xi_{out} = 125$, $\xi_{relay} = 2$. Let us assume that the propagation environment, in which deployment is being carried out, is characterized by the parameters given in Section 6.1 (i.e., $\eta = 4.7$, $\sigma = 7.7$ dB etc.). The optimal average cost per step, under these parameter values, is $\lambda^* = V^*(1) = 1.85$ (computed numerically).[4]

We numerically study the performance of the following three types of algorithms: (i) η and σ are known prior to deployment (the agent uses the fixed optimal policy with $\xi_{relay} = 2$ and $\xi_{out} = 125$ in this case), (ii) the agent has imperfect estimates of η and σ deployment, and OptAsYouGoLearning is used to update the policy as deployment progresses, and (iii) the agent has imperfect estimates of η and σ deployment, but the corresponding suboptimal policy is used along the infinite line without any update. We use the abbreviations OAYGL and OAYG for OptAsYouGoLearning and Optimal Algorithm for As-You-Go deployment (i.e., Algorithm 1), respectively. Also, following the terminology in [1], we use the abbreviation FPWU for "Fixed Policy without Update."

Next, we formally explain the various cases considered in our simulations:

- **(i) OAYG:** Here the agent knows $\eta = 4.7$, $\sigma = 7.7$ dB prior to deployment, and uses Algorithm 1 with $\xi_{out} = 125$, $\xi_{relay} = 2$.
- **(ii) OAYGL Case 1:** Here the true $\eta = 4.7$ and $\sigma = 7.7$ dB are unknown to the deployment agent. But the agent has an initial estimate $\eta = 5$, $\sigma = 8$ dB. Hence, he starts deploying using a $\underline{V}^{(0)}$ which is optimal for these imperfect estimates of η and σ, and $\xi_{out} = 125$, $\xi_{relay} = 2$. He updates the policy using the OptAsYouGoLearning algorithm as deployment progresses.
- **(iii) OAYGL Case 2:** This is different from OAYGL Case 1 only in the aspect that here deployment starts with the optimal policy for $\eta = 4$, $\sigma = 7$ dB.
- **(iv) FPWU Case 1:** Here the true η and σ are unknown prior to deployment, and the agent has an initial estimate $\eta = 5$, $\sigma = 8$ dB. The agent computes $\underline{V}^*$ for these imperfect initial estimates and $\xi_{out} = 125$, $\xi_{relay} = 2$, and uses this policy throughout the deployment process without any update. This case will demonstrate the gain in performance by

4. These values of ξ_{out} and ξ_{relay} are chosen because they can produce reasonable values of placement rate, mean power per step and mean outage per step, which can be used in practical networks. However, these values are chosen only for illustration purposes, and the choice will vary depending on the requirement for deployment.

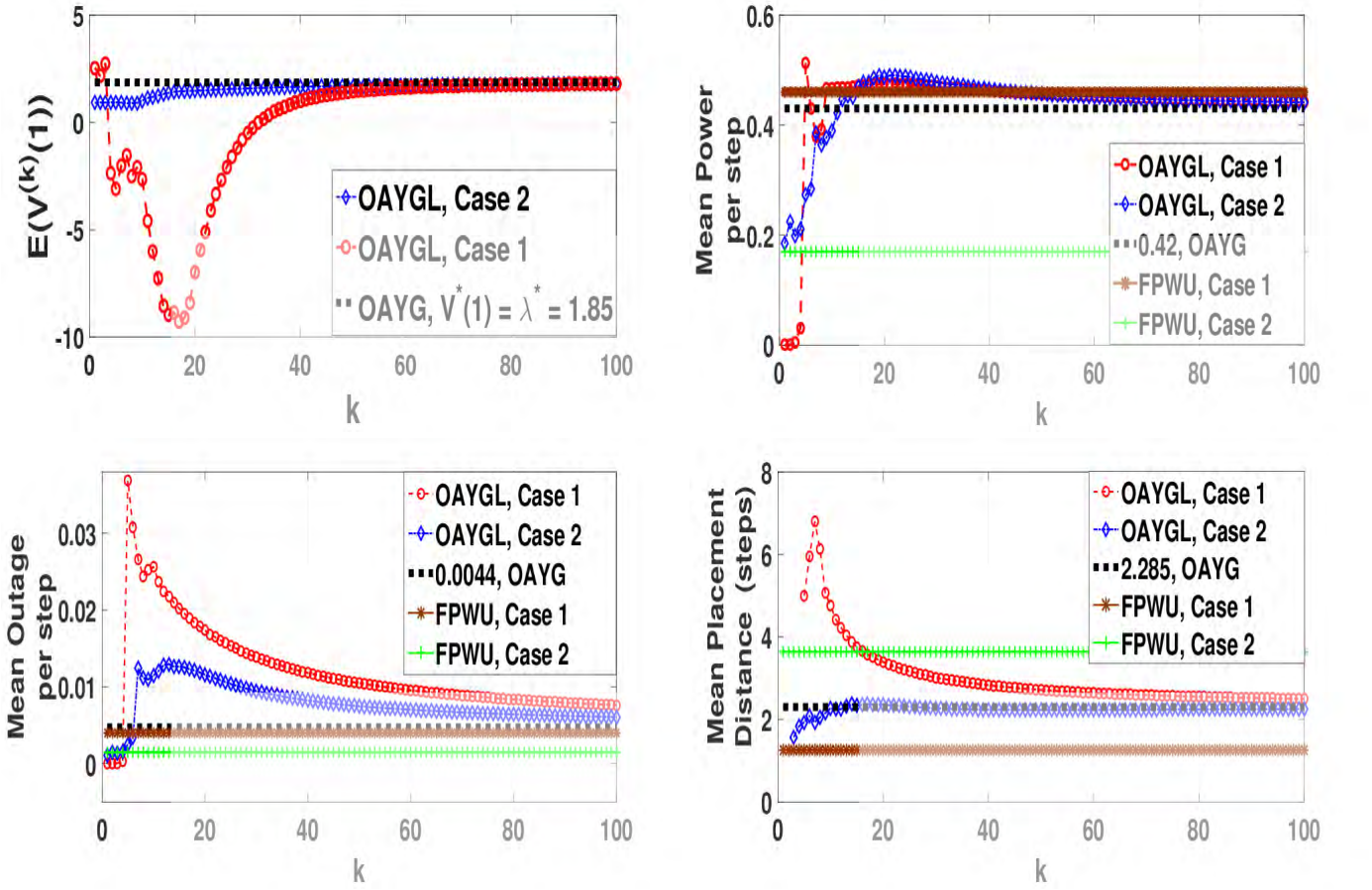

Figure 3: Convergence speed of OptAsYouGoLearning (OAYGL) with the number of steps, k. In the legends, "OAYG" refers to the values that are obtained if Algorithm 1 is used; these are the target values for OptAsYouGoLearning.

updating the policy under OptAsYouGoLearning, w.r.t. the case where the suboptimal policy is used throughout the deployment process.

(v) FPWU Case 2: It differs from FPWU Case 1 only in the aspect that here the agent has initial estimates $\eta = 4$, $\sigma = 7$ dB.

For simulation of OAYGL, we chose $a(k) = \frac{120}{k}$. We simulated 2000 independent network deployments (i.e., 2000 sample paths of the deployment process) with OptAsYouGoLearning, and estimated (by averaging over 2000 deployments) the expectation of $V^{(k)}(1)$, mean power per step (i.e., $\frac{\sum_{j=1}^{N_k} \Gamma_j}{k}$), mean outage per step (i.e., $\frac{\sum_{j=1}^{N_k} Q_{out}^{(j,j-1)}}{k}$) and mean placement distance (i.e., $\frac{k}{N_k}$), in the part of the network between the sink node to the k-th step. The results are summarized in Figure 3. Asymptotically the estimates are supposed to converge to the values provided by OAYG.

Observations: We observe that the estimate of $\mathbb{E}(V^{(k)}(1))$ approaches the optimal cost $\lambda^* = V^*(1) = 1.85$ (for the actual propagation parameters), as k increases, and gets to within 10% of the optimal cost by the time where $k = 35$ to 40 (within a distance of 800 meters), while starting with two widely different initial guesses of the propagation parameters. The estimates of mean power per step, mean outage per step and mean placement distance also converges very fast to the corresponding values achieved by OAYG. It also shows that, if the performance of the initial imperfect policy (FPWU) is significantly different than that of OAYG, then OptAsYouGoLearning will provide closer performance to OAYG, as compared to FPWU (see the mean placement distance plot).

Note that, even though Theorem 7 guarantees almost sure convergence, the convergence speed will vary across sample paths. But here we demonstrate speed of convergence after averaging over 2000 sample paths.

6.3 OptAsYouGoAdaptiveLearning

Now we will demonstrate the performance of OptAsYouGoAdaptiveLearning (Algorithm 3) for deployment over a finite distance under an unknown propagation environment. We again assume that the true propagation parameters are given by $\eta = 4.7$, $\sigma = 7.7$ dB. For these parameters, under the choice $\xi_{relay} = 2$ and $\xi_{out} = 125$, the optimal average cost per step will be $\lambda^* = 1.85$, which can be achieved by OAYG (Algorithm 1). OAYG in this case will yield a mean placement distance of 2.285 steps, a mean outage per step of $\frac{0.0101}{2.285} = 0.0044$, and a mean power per step of 0.423 mW.

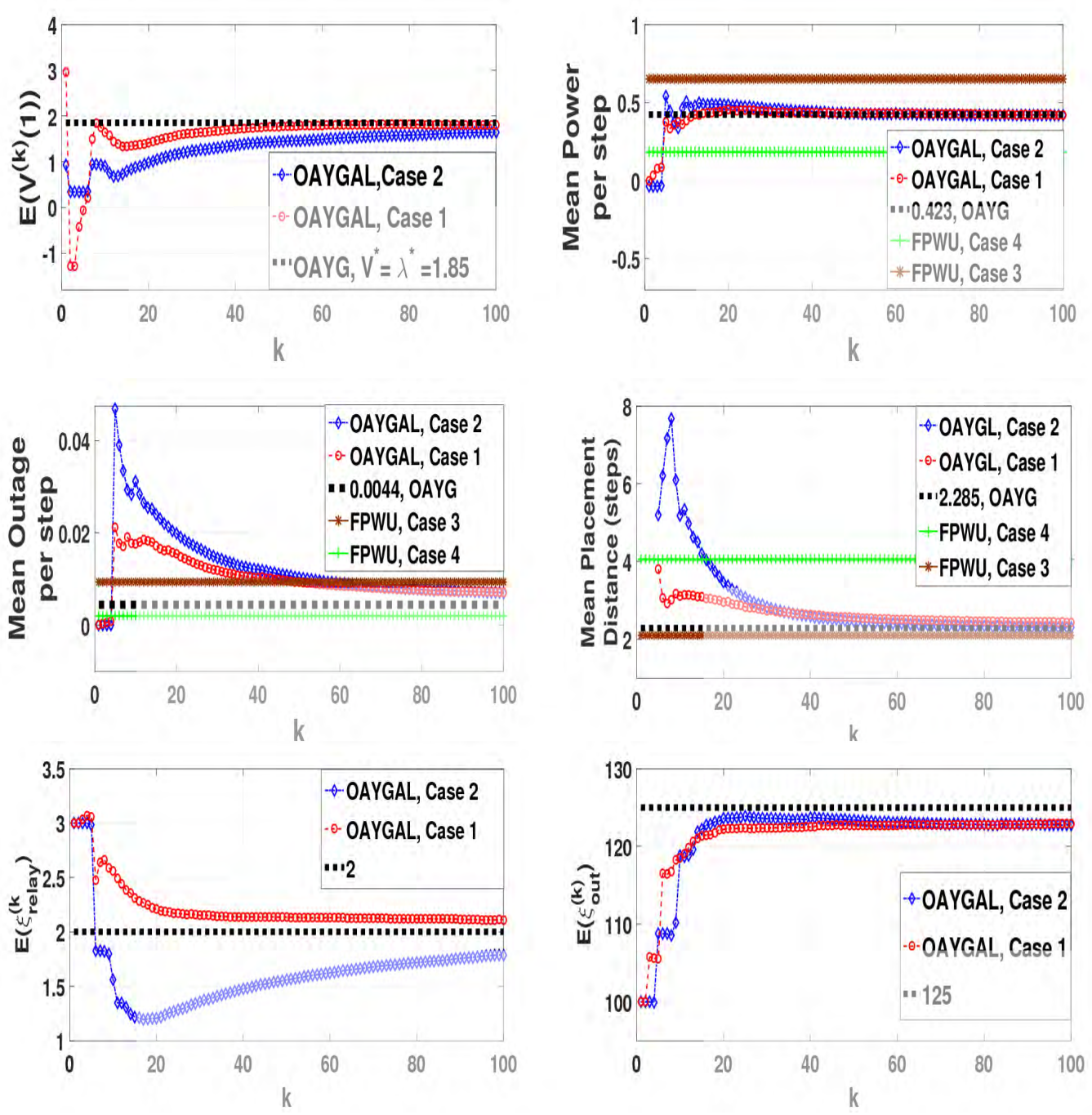

Figure 4: Convergence speed of OptAsYouGoAdaptiveLearning (OAYGAL) with the number of steps, k. In the legends, "OAYG" refers to the values that are obtained if Algorithm 1 is used; these are the target values for OptAsYouGoAdaptiveLearning. Evolution of $\xi_{out}^{(k)}$ and $\xi_{relay}^{(k)}$ are shown for a longer time, since they converge slowly to their respective target values.

Now, let us suppose that we need to solve the constrained problem in (4) with the targets $\overline{q} = 0.0044$ and $\overline{N} = \frac{1}{2.285}$, but the true η and σ of the environment are unknown to us. Hence, we need to employ OptAsYouGoAdaptiveLearning (we use the abbreviation OAYGAL for it); as compared to OptAsYouGoLearning, we need to make an additional choice of $\xi_{out}^{(0)}$ and $\xi_{relay}^{(0)}$.

We consider the following cases in our simulations:

(i) OAYG: This is same as in Section 6.2

(ii) OAYGAL Case 1: Here the true $\eta = 4.7$ and $\sigma =$ 7.7 dB are unknown to the deployment agent. But the agent has an initial estimate $\eta = 5$, $\sigma = 8$ dB. Hence, he starts deploying using a $\underline{V}^{(0)}$ which is optimal for these imperfect estimates of η and σ, and $\xi_{out}^{(0)} = 100$, $\xi_{relay}^{(0)} = 3$. He updates the policy using the OptAsYouGoAdaptiveLearning algorithm as deployment progresses.

(iii) OAYGAL Case 2: This is same as OAYGAL Case 1, except that the agent starts deploying using a policy

corresponding to the wrong initial estimate $\eta = 4$, $\sigma = 7$ dB (under $\xi_{out}^{(0)} = 100$, $\xi_{relay}^{(0)} = 3$).

(iv) FPWU Case 3: Here the agent uses $\xi_{out} = 100$, $\xi_{relay} = 3$, and uses the corresponding optimal policy for the imperfect estimates $\eta = 5$, $\sigma = 8$ dB, throughout the deployment process.

(v) FPWU Case 4: This is similar to FPWU Case 3; the only difference is that the optimal policy for the imperfect estimates $\eta = 4$, $\sigma = 7$ dB is used throughout deployment.

For simulation of OAYGAL, we chose the step sizes as follows. We took $a(k) = \frac{1}{k^{0.55}}$, $b(k) = \frac{100}{k^{0.8}}$ for the ξ_{out} update and $b(k) = \frac{1}{k^{0.8}}$ for the ξ_{relay} update (however, both ξ_{out} and ξ_{relay} are updated in the same timescale). We simulated 2000 independent network deployments (i.e., 2000 sample paths of the deployment process) with OptAsYouGoLearning, and estimated (by averaging over 2000 deployments) the expectations of $V^{(k)}(1)$, mean power per step, mean outage per step mean placement distance, $\xi_{out}^{(k)}$ and $\xi_{relay}^{(k)}$, in the part of the network between the sink node to the k-th step. The results are summarized in Figure 4 (see previous page).

Observations: Under OAYGAL Case 1 the estimates of the expectations of $V^{(2000)}(1)$, $\xi_{out}^{(2000)}$, $\xi_{relay}^{(2000)}$, mean power per step up to the 2000^{th} step, mean outage per step up to the 2000^{th} step, and mean placement distance over 2000 steps are 1.8479, 124.89, 2.01, 0.4222, 0.04403 and 2.2852, whereas the corresponding target values are 1.85, 125, 2, 0.4223, 0.00441 and 2.2857, respectively. Similarly, for OAYGAL Case 2 also, the quantities converge close to the target values. In practice, the performance metrics are reasonably close to their respective target values within 100 steps (i.e., 2 kms).

FPWU Case 3 and FPWU Case 4 either violate some constraint or uses significantly higher per-step power compared to OAYG. But, by using OptAsYouGoAdaptiveLearning, we can achieve mean power per step close to the optimal while (possibly) violating the constraints by small amount. However, performance of OAYGAL is significantly closer to the target compared to FPWU. □

The speed of convergence will depend on the choice of $a(k)$ and $b(k)$, of $\xi_{out}^{(0)}$, $\xi_{relay}^{(0)}$ and the initial estimates of η and σ. However, optimizing convergence speed over step size sequences is left for future research.

7 Conclusion

In this paper, we have formulated the problem of pure-as-you-go deployment along a line, under a very light traffic assumption. The problem was formulated as an average cost MDP, and its optimal policy structure was studied analytically. We also proposed two learning algorithms that asymptotically converge to the corresponding optimal policies. Numerical results have been provided to illustrate the speed of convergence of the learning algorithms.

While this paper provides an interesting set of results, it can be extended or modified in several ways: (i) One can attempt to develop deployment algorithms for 2 dimensional regions, where multiple agents cooperate to carry out the deployment. (ii) One can also attempt to develop deployment algorithms that can provide theoretical guarantees on the data rate supported by the deployed networks (instead of assuming that the traffic is lone packet). (iii) The optimization of the rate of convergence for the learning algorithms by proper choice of the step sizes is also a challenging problem. We leave these issues for future research endeavours.

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," *published in IEEE/ACM Transactions on Networking, available in http://www.arxiv.org/abs/1502.06878*, vol. 24, no. 5, 2016.

[2] H. Liu, J. Li, Z. Xie, S. Lin, K. Whitehouse, J. A. Stankovic, and D. Siu, "Automatic and robust breadcrumb system deployment for indoor firefighter applications," in *MobiSys*. ACM, 2010.

[3] *http://robotics.eecs.berkeley.edu/~pister/29Palms0103/*.

[4] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme, "Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle," in *IEEE International Conference on Robotics and Automation*. IEEE, 2004, pp. 3602–3608.

[5] D. Anthony, J. Ore, C. Detweiler, and E. Basha, "Controlled sensor network installation with unmanned aerial vehicles," in *ACM Conference on Embedded Network Sensor Systems (SenSys)*. ACM, 2014, pp. 348–349.

[6] V. Dyo, S. Ellwood, D. Macdonald, A. Markham, C. Mascolo, B. Pasztor, S. Scellato, N. Trigoni, R. Wohlers, and K. Yousef, "Evolution and sustainability of a wildlife monitoring sensor network," in *Proceedings of SenSys 2010*. ACM, 2011, pp. 127—140.

[7] A. A. A. Alkhatib, "A review on forest fire detection techniques," *International Journal of Distributed Sensor Networks (a journal published by Hindawi Publishing Corporation)*, vol. 2014.

[8] *www.cnet.com/news/inside-historic-nokia-bell-labs-tomorrows-5g-network-tech/*.

[9] A. Chattopadhyay, M. Coupechoux, and A. Kumar, "Measurement based impromptu deployment of a multi-hop wireless relay network," in *Proc. of the 11th Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE, 2013.

[10] A. Chattopadhyay, A. Ghosh, A. Rao, B. Dwivedi, S. Anand, M. Coupechoux, and A. Kumar, "Impromptu deployment of wireless relay networks: Experiences along a forest trail," *Proceedings of the IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS), 2014, a detailed version available in http://arxiv.org/abs/1409.3940*.

[11] M. Souryal, J. Geissbuehler, L. Miller, and N. Moayeri, "Real-time deployment of multihop relays for range extension," in *Proc. of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*. ACM, 2007, pp. 85–98.

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

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

[14] J. Bao and C. Lee, "Rapid deployment of wireless ad hoc backbone networks for public safety incident management," in *Global Telecommunications Conference (GLOBECOM)*. IEEE, 2007, pp. 1217–1221.

[15] H. Liu, Z. Xie, J. Li, K. Whitehouse, J. Stankovic, S. Lin, and D. Siu, "Efficient and reliable breadcrumb systems via coordination among multiple first responders," in *Proc. of PIMRC*. IEEE, 2011, pp. 1005–1009.

[16] K. Miranda, A. Molinaro, and T. Razafindralambo, "A survey on rapidly deployable solutions for post-disaster networks," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 117–123, 2016.

[17] P. Mondal, K. Naveen, and A. Kumar, "Optimal Deployment of Impromptu Wireless Sensor Networks," in *Proc. of the IEEE National Conference on Communications (NCC),2012*. IEEE, 2012.

[18] A. Sinha, A. Chattopadhyay, K. Naveen, M. Coupechoux, and A. Kumar, "Optimal sequential wireless relay placement on a random lattice path," *Ad Hoc Networks Journal (Elsevier).*, vol. 21, pp. 1–17, 2014.

[19] A. Ghosh, A. Chattopadhyay, A. Arora, and A. Kumar, "As-you-go deployment of a 2-connected wireless relay network for sensor-sink interconnection," in *International Conference on Signal Processing and Communications (SPCOM)*. IEEE, 2014.

[20] A. Chattopadhyay, A. Sinha, M. Coupechoux, and A. Kumar, "Deploy-as-you-go wireless relay placement: An optimal sequential decision approach using the multi-relay channel model," *Accepted in IEEE Transactions on Mobile Computing, available in http://ieeexplore.ieee.org/document/7463497/*.

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

[22] A. Bhattacharya, A. Rao, D. G. R. Sahib, A. Mallya, S. Ladwa, R. Srivastava, S. Anand, and A. Kumar, "Smartconnect: A system for the design and deployment of wireless sensor networks," in *Proc. of the 5th International Conference on Communication Systems and Networks (COMSNETS)*. IEEE, 2013.

[23] *http://www.astec.org.in/astec/content/wireless-sensor-network*.

[24] F. J. Beutler and K. W. Ross, "Optimal policies for controlled markov chains with a constraint," *Journal of Mathematical Analysis and Applications*, vol. 112, pp. 236–252, 1985.

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

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

[27] N. Abdeddaim, F. Theoleyre, F. Rousseau, and A. Duda, "Multichannel cluster tree for 802.15.4 wireless sensor networks," in *Proc. of the 23th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2012, pp. 590—595.

[28] E. Toscano and L. Bello, "Multichannel superframe scheduling for IEEE 802.15.4 industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 2, pp. 337 —350, 2012.

[29] 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. of the 4th international conference on Real-world wireless sensor networks (REALWSN)*, 2010, pp. 154—161.

[30] R. Upadrashta, T. Choubisa, V. S. Aswath, A. Praneeth, A. Prabhu, S. Raman, T. Gracious, and P. Kumar, "An animation-and-chirplet based approach to intruder classification using PIR sensing," in *Proceedings of IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2015, pp. 1–6.

[31] B. Aghaei, "Using wireless sensor network in water, electricity and gas industry," in *Proceedings of the 3rd IEEE International Conference on Electronics Computer Technology*, 2011, pp. 14—17.

[32] T. Adame, A. Bel, B. Bellalta, J. Barcelo, and M. Oliver, "IEEE 802.11ah: the WiFi approach for M2M communications," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 144—152, 2014.

[33] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of Wireless Sensor Network Applications (WSNA)*. ACM, 2002, pp. 88–97.

[34] D. Bertsekas, *Dynamic Programming and Optimal Control, Vol. II*. Athena Scientific, 2007.

[35] O. Hernandez-Lerma and J. B. Lasserre, *Discrete-Time Markov Control Processes Basic Optimality Criteria*. Springer, 1996.

[36] S. Bhatnagar, "The borkar-meyn theorem for asynchronous stochastic approximations," *Systems and Control Letters*, vol. 60, no. 7, pp. 472–478, 2011.

[37] V. S. Borkar, *Stochastic approximation: a dynamical systems viewpoint*. Cambridge University Press, 2008.

[38] D.-J. Ma and A. M. Makowski', "A class of steering policies under a recurrence condition," in *Proc. of the 27th Conference on Decision and Control (CDC)*. IEEE, 1988.

[39] H. Kushner and D. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, 1978.

[40] N. Salodkar, A. Bhorkar, A. Karandikar, and V. Borkar, "An online learning algorithm for energy efficient delay constrained scheduling over a fading channel," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 732–742, 2008.

[41] J. Abounadi, D. Bertsekas, and V. S. Borkar, "Learning algorithms for markov decision processes with average cost," *SIAM Journal on Control and Optimization*, vol. 40, pp. 681–698, 2001.

[42] V. S. Borkar and K. Soumyanath, "A new analog parallel scheme for fixed point computation, part 1: Theory," *IEEE Transactions on Circuits and Systems I*, vol. 44, pp. 351–355, 1997.

[43] W. Rudin, *Principles of Mathematical Analysis, Third Edition*. McGraw-Hill International Editions, 1976.

[44] C. Lakshminarayanan and S. Bhatnagar, "A stability criterion for two timescale stochastic approximation schemes," *Available in http://stochastic.csa.iisc.ernet.in/www/research/files/ttsastb.pdf*, 2014.

Arpan Chattopadhyay obtained his B.E. in Electronics and Telecommunication Engineering from Jadavpur University, India in the year 2008, and M.E. and Ph.D in Telecommunication Engineering from Indian Institute of Science, Bangalore, India in the year 2010 and 2015, respectively. He is currently working in Electrical Engineering department, University of Southern California, as a postdoctoral researcher. His research interests include design, resource allocation, control and learning in wireless networks and cyber-physical systems.

Avishek Ghosh obtained his B.E. in Electronics and Telecommunication Engineering from Jadavpur University, India in 2012, and M.E. in Telecommunication from Indian Institute of Science, Bangalore, India in the year 2014. He is currently doing his Ph.D in the department of EECS of UC Berkeley. His research interests include networks and machine learning.

Anurag Kumar (B.Tech., Indian Institute of Technology (IIT) Kanpur; PhD, Cornell University, both in Electrical Engineering) was with Bell Labs, Holmdel, N.J., for over 6 years. Since then he has been on the faculty of the ECE Department at the Indian Institute of Science (IISc), Bangalore; he is at present the Director of the Institute. His area of research is communication networking, and he has recently focused primarily on wireless networking. He is a Fellow of the IEEE, the Indian National Science Academy (INSA), the Indian National Academy of Engineering (INAE), and the Indian Academy of Sciences (IASc). He was an associate editor of IEEE Transactions on Networking, and of IEEE Communications Surveys and Tutorials.

Supplementary Material
Title: "Asynchronous Stochastic Approximation Based Learning Algorithms for As-You-Go Deployment of Wireless Relay Networks along a Line"
Authors: Arpan Chattopadhyay, Avishek Ghosh, anurag Kumar

APPENDIX A
FORMULATION FOR KNOWN PROPAGATION PARAMETERS

Proof of Theorem 2: From (7), $V(B)$ is unique for fixed ξ_{out} and ξ_{relay}. Hence, we can say that $V(B)$ is a continuous and decreasing function of $V(1)$. Now, let us assume that $V(r+1)$ is continuous and decreasing in $V(1)$ for some $r, 1 \leq r \leq B-1$. Let us recall (7) for $V(r)$. Since $V(r+1)$ is continuous and decreasing in $V(1)$ by our induction hypothesis, it is evident from (7) that $V(r)$ is also continuous and decreasing in $V(1)$. Proceeding in this way, we can write $V(1) = \phi(V(1))$ where $\phi(\cdot)$ is continuous and decreasing in $V(1)$. But $V(1)$ is continuous and strictly increasing in $V(1)$. Hence, $V(1) = \phi(V(1))$ has a unique fixed point $V^*(1)$. Now, from (7), $V(B-1)$ is unique since $V(1) = V^*(1)$ is unique and $V(B)$ is unique. Proceeding backwards in this way, we can show that we have a unique $V^*(r)$ for all r.

Now, from (7), we find that $V^*(r) \leq -V^*(1)+V^*(r+1)$, i.e., $V^*(r+1) \geq V^*(r)+V^*(1)$ for all $r \in \{1, 2, \cdots, B-1\}$. Also, $V^*(1) = \lambda^* > 0$ and it is unique. This proves the second part of the theorem. □

Proof of Theorem 4: Let us denote the mean power per link, mean outage per link and mean placement distance (in steps) under a stationary policy π by $\overline{\Gamma}_\pi$, $\overline{Q}_{out,\pi}$ and $\overline{U}_\pi$. Then, by Renewal-Reward Theorem, we have $\lambda^*(\xi_{out}, \xi_{relay}) = \inf_\pi \frac{\Gamma_\pi + \xi_{out}\overline{Q}_{out,\pi} + \xi_{relay}}{\overline{U}_\pi}$. The numerator is affine and increasing in ξ_{out} and ξ_{relay}, and the denominator is independent of ξ_{out} and ξ_{relay}. Hence, $\lambda^*(\xi_{out}, \xi_{relay})$ is concave, increasing in ξ_{out} and ξ_{relay}, since the pointwise infimum of increasing affine functions of (ξ_{out}, ξ_{relay}) is increasing and jointly concave in (ξ_{out}, ξ_{relay}). Now, any increasing, concave function is continuous. Hence, $\lambda^*(\xi_{out}, \xi_{relay})$ is continuous in (ξ_{out}, ξ_{relay}). Also, it is easy to see that $\lambda^*(\xi_{out}, \xi_{relay})$ is Lipschitz in each argument with Lipschitz constant 1.

Proof of Theorem 5: By Theorem 4, $V^*(1) := \lambda^*$ is Lipschitz continuous in (ξ_{out}, ξ_{relay}). By (7), $V^*(B)$ is Lipschitz continuous in (ξ_{out}, ξ_{relay}). Hence, by (7), $V^*(B-1)$ is also Lipschitz continuous in (ξ_{out}, ξ_{relay}). Thus, by using backward induction, we can show that $V^*(r)$ is Lipschitz continuous in (ξ_{out}, ξ_{relay}) for all $1 \leq r \leq B$.

APPENDIX B
OPTASYOUGOLEARNING: LEARNING WITH PURE AS-YOU-GO DEPLOYMENT, FOR GIVEN LAGRANGE MULTIPLIERS

Proof of Theorem 7: We can rewrite (9) as follows:

$$V^{(k)}(r) = V^{(k-1)}(r) + a(\nu(r,k))\mathbb{I}\{r \in \mathcal{I}_k\}\left[f_r(\underline{V}^{(k-1)}) + M_k(r)\right] \tag{11}$$

where, for all $1 \leq r \leq B-1$

$$f_r(\underline{V}^{(k-1)}) = \mathbb{E}_W\Big[\min\Big\{\min_\gamma(\gamma + \xi_{out}Q_{out}(r,\gamma,W)) + \xi_{relay}, -V^{(k-1)}(1) + V^{(k-1)}(r+1)\Big\} - V^{(k-1)}(r)\Big]$$

$$M_k(r) = \Big[\min\Big\{\min_\gamma(\gamma + \xi_{out}Q_{out}(r,\gamma,w_r)) + \xi_{relay}, -V^{(k-1)}(1) + V^{(k-1)}(r+1)\Big\} - V^{(k-1)}(r)\Big] - f_r(\underline{V}^{(k-1)})$$

and

$$f_B(\underline{V}^{(k-1)}) = \mathbb{E}_W\Big[\min_\gamma(\gamma + \xi_{out}Q_{out}(B,\gamma,W)) + \xi_{relay} - V^{(k-1)}(B)\Big]$$

$$M_k(B) = \Big[\min_\gamma(\gamma + \xi_{out}Q_{out}(B,\gamma,w_B)) + \xi_{relay} - V^{(k-1)}(B)\Big] - f_B(\underline{V}^{(k-1)})$$

Let $\underline{M}_k := (M_k(1), \cdots, M_k(B))$. Let us denote the σ-field $\mathcal{F}_k := \sigma(\underline{V}_i, \mathcal{I}_i, \underline{M}_i, i \leq k-1)$; it is the information available to the deployment agent before making any decision at the k-th step. Clearly, the update equations fall under the category of Asynchronous Stochastic Approximation algorithms (see [36]). In order to see whether $\underline{V}^{(k)} \to \underline{V}^*$ almost surely, we will first check whether the five assumptions mentioned in [36] are satisfied.

Checking Assumption 1 of [36]: For each $r, 1 \leq r \leq B$, $V(r)$ gets updated at least once in every B steps. Hence, $\liminf_{k\to\infty} \frac{\nu(r,k)}{k} \geq \frac{1}{B} > 0$ almost surely. Hence, the assumption is satisfied.

Checking Assumption 2 of [36]: If we choose $\{a(k)\}_{k\geq 1}$ to be a bounded, decreasing sequence with $\sum_k a(k) = \infty$ and $\sum_k a^2(k) < \infty$, this condition will be satisfied.

Checking Assumption 3 of [36]: Not applicable to our problem since before updating $\underline{V}^{(k)}$ the deployment agent knows $\underline{V}^{(k-1)}$.

Before checking the other two conditions, we will establish a lemma. Let us consider the following system of o.d.e-s:

$$\dot{V}_t(r) = \kappa_t(r) f_r(\underline{V}_t) \ \ \forall r \in \{1, 2, \cdots, B\} \tag{12}$$

where $\kappa_t(r) \in (0, 1]$ for all r and t. By Theorem 2, this system of o.d.e-s has an unique stationary point $\underline{V}^*(\xi_{out}, \xi_{relay})$.

Lemma 1: $\underline{V}^*(\xi_{out}, \xi_{relay})$ is a globally asymptotically stable equilibrium for the system of o.d.e-s (12). Also,

$\underline{V} = 0$ is a globally asymptotically stable equilibrium for (12) when γ, ξ_{out} and ξ_{relay} are replaced by 0 in the definition of $f_r(\underline{V})$ for all $r \in \{1, 2, \cdots, B\}$.

Proof: Note that, by Theorem 2, $\underline{V}^*(\xi_{out}, \xi_{relay})$ is the unique stationary point for (12). Now, the proof for this lemma follows from similar line of arguments as in the appendix of [40] (which uses results from [41] and [42]). □

Checking Assumption 4 of [36]: It is easy to see that $f_r(\underline{V})$ is Lipschitz in $\underline{V}$ for each r; this satisfies Assumption 4(i). Let us consider the ODE (12) with $0 < \kappa_t(r) \leq 1$ corresponds to the relative rate at which $V(r)$ is updated. By Lemma 1, $\underline{V}^*(\xi_{out}, \xi_{relay})$ is a globally asymptotically stable equilibrium for the system of o.d.e-s (12). Hence, Assumption 4(ii) is satisfied.

Consider the functions $\frac{f_r(c\underline{V})}{c}, c \geq 1$ for all r. Clearly, $\lim_{c\to\infty} \frac{f_r(c\underline{V})}{c} = \min\{0, -V(1) + V(r+1)\} - V(r)$ for $r \neq B$, and $\lim_{c\to\infty} \frac{f_B(c\underline{V})}{c} = -V(B)$. Note that $\frac{f_r(c\underline{V})}{c}$ for all r and $\lim_{c\to\infty} \frac{f_r(c\underline{V})}{c}$ all are continuous in $\underline{V}$, and $\frac{f_r(c\underline{V})}{c}$ is decreasing in c. Hence, by Theorem 7.13 of [43], convergence of $\frac{f_r(c\underline{V})}{c}$ over compacts is uniform. Hence, Assumption 4(iii) is satisfied.

Consider the ODE: $\dot{V}_t(r) = \kappa_t(r)(\min\{0, -V_t(1) + V_t(r+1)\} - V_t(r))$ for $r \neq B$ and $\dot{V}_t(B) = \kappa_B(t)(-V_t(B))$. Clearly, by the second part of Lemma 1, there is a unique globally asymptotically stable equilibrium $\underline{V} = \underline{0}$. Hence, Assumption 4(iv) is satisfied.

Checking Assumption 5 of [36]: It is easy to see that, $\{\underline{M}_k\}_{k\geq 1}$ is a Martingale difference sequence adapted to $\mathcal{F}_k$. Hence, Assumption 5(i) is satisfied.

Now,

$$|M_{k+1}(r)| \leq 2\left|\left(\min\{P_M + \xi_{out} + \xi_{relay}, -V^{(k)}(1) + V^{(k)}(r+1)\} - V^{(k)}(r)\right)\right|$$

and

$$|M_{k+1}(B)| \leq \left|\left(P_M + \xi_{out} + \xi_{relay} - V^{(k)}(B)\right)\right|$$

Hence, $||M_{k+1}|| \leq C_0(1 + ||\underline{V}^{(k)}||)$ for some $C_0 > 0$. Hence, Assumption 5(ii) is satisfied. Now, by [36, Theorem 3], $\underline{V}^{(k)} \to \underline{V}^*$. □

Now, by Renewal-Reward Theorem,

$$\lambda^*(\xi_{out}, \xi_{relay}) = \frac{\overline{\Gamma}^*(\xi_{out}, \xi_{relay}) + \xi_{out}\overline{Q}^*_{out}(\xi_{out}, \xi_{relay}) + \xi_{relay}}{\overline{U}^*(\xi_{out}, \xi_{relay})}$$

Since $\lambda^*(\xi_{out}, \xi_{relay})$ is continuous in (ξ_{out}, ξ_{relay}) (by Theorem 4), we conclude that $\overline{Q}^*_{out}(\xi_{out}, \xi_{relay})$ is continuous in ξ_{out} and ξ_{relay}. Hence, $\frac{\overline{\Gamma}^*(\xi_{out}, \xi_{relay})}{\overline{U}^*(\xi_{out}, \xi_{relay})}$, $\frac{\overline{Q}^*_{out}(\xi_{out}, \xi_{relay})}{\overline{U}^*(\xi_{out}, \xi_{relay})}$ and $\frac{1}{\overline{U}^*(\xi_{out}, \xi_{relay})}$ are continuous in (ξ_{out}, ξ_{relay}). □

Now, the proof of the theorem is completed by the following lemma.

Lemma 2: Under Assumption 2, $g(r, \gamma)$ is continuous in (ξ_{out}, ξ_{relay}).

Proof: We will first prove the result for $r \leq B - 1$. Let us fix an $r \in \{1, \cdots, B-1\}$ and any $\gamma \in \mathcal{S}$. We will only show that $g(r, \gamma)$ is continuous in ξ_{out}; the proof for continuity of $g(r, \gamma)$ w.r.t. ξ_{relay} will be similar.

Let us consider a sequence $\{\xi_n\}_{n\geq 1}$ such that $\xi_n \to \xi_{out}$. Let us denote the joint probability distribution of (U_k, Γ_k) by $g_n(r, \gamma)$, if Algorithm 1 is used with ξ_n as the cost for unit outage. We will show that $\lim_{n\to\infty} g_n(r, \gamma) \to g(r, \gamma)$.

Define the sets $\mathcal{E}_{r,\gamma'} = \left\{w_r : \gamma + \xi_{out}Q_{out}(r, \gamma, w_r) < \gamma' + \xi_{out}Q_{out}(r, \gamma', w_r)\right\}$ and $\mathcal{E}_u = \left\{w_u : \min_{\gamma\in\mathcal{S}}(\gamma + \xi_{out}Q_{out}(u, \gamma, w_u)) > -\xi_{relay} - V^*(1) + V^*(u+1)\right\}$ for all $1 \leq u \leq r$.

Let us define $\mathcal{E} = \cap_{\gamma'\neq\gamma}\mathcal{E}_{r,\gamma'} \cap_{u\leq r-1} \mathcal{E}_u \cap \overline{\mathcal{E}_r}$, where $\overline{\mathcal{E}_r}$ is the set complement of $\mathcal{E}_r$.

Now, $g(r, \gamma) = \mathbb{P}(\mathcal{E}) = \mathbb{E}(\mathbb{I}_{\mathcal{E}})$, where $\mathbb{I}$ denotes the indicator function. The expectation is over the joint distribution of $(W_1, W_2, \cdots, W_r)$ (shadowing random variables from r locations).

Now, for any $\gamma' \neq \gamma$, we have $\mathbb{P}\Big(\gamma + \xi_{out}Q_{out}(r, \gamma, W_r) = \gamma' + \xi_{out}Q_{out}(r, \gamma', W_r)\Big) = 0$, and $\mathbb{P}\Big(\min_{\gamma\in\mathcal{S}}(\gamma + \xi_{out}Q_{out}(u, \gamma, W_u)) = -\xi_{relay} - V^*(1) + V^*(u+1)\Big) = 0$ for all $u \leq r$; these two assertions follow from Assumption 2 and from the continuity of $Q_{out}(r, \gamma, w)$ in w. Hence, we can safely assume the following:

- $\overline{\mathcal{E}_{r,\gamma'}}$ has the same expression as $\mathcal{E}_{r,\gamma'}$ except that the $<$ sign is replaced by $>$ sign.
- $\overline{\mathcal{E}_u}$ has the same expression as $\mathcal{E}_u$ except that the $>$ sign is replaced by $<$ sign.

Let $\mathcal{E}^{(n)}_{r,\gamma'}$, $\mathcal{E}^{(n)}_u$ and $\mathcal{E}^{(n)}$ be the sets obtained by replacing ξ_{out} by ξ_n in the expressions of the sets $\mathcal{E}_{r,\gamma'}$, $\mathcal{E}_u$ and $\mathcal{E}$ respectively (also $\underline{V}^*$ has to be replaced by the corresponding optimal $\underline{V}^{(n,*)}$). Clearly, we can make similar claims for $\mathcal{E}^{(n)}_{r,\gamma'}$, $\mathcal{E}^{(n)}_u$.

Now, if we can show that $\mathbb{E}(\mathbb{I}_{\mathcal{E}^{(n)}}) \to \mathbb{E}(\mathbb{I}_{\mathcal{E}})$, the lemma will be proved, because $g(r, \gamma) = \mathbb{P}(\mathcal{E}) = \mathbb{E}(\mathbb{I}_{\mathcal{E}})$.

APPENDIX C
OPTASYOUGOADAPTIVELEARNING WITH CONSTRAINTS ON OUTAGE PROBABILITY AND RELAY PLACEMENT RATE

C.1 Proof of Theorem 8

Let us denote by $g(r, \gamma), r \in \{1, 2, \cdots, B\}, \gamma \in \mathcal{S}$ the joint distribution of (U_k, Γ_k) under Algorithm 2. For the time being, let us assume that $g(r, \gamma)$ is continuous in (ξ_{out}, ξ_{relay}). Then, the mean placement distance $\overline{U}^*(\xi_{out}, \xi_{relay}) = \sum_{r=1}^{B}\sum_{\gamma\in\mathcal{S}} rg(r, \gamma)$, and the mean power per link $\overline{\Gamma}^*(\xi_{out}, \xi_{relay}) = \sum_{r=1}^{B}\sum_{\gamma\in\mathcal{S}} \gamma g(r, \gamma)$ are both continuous in (ξ_{out}, ξ_{relay}).

Claim 1: $\lim_{n\to\infty}\mathbb{I}_{\mathcal{E}_u^{(n)}} \to \mathbb{I}_{\mathcal{E}_u}$, and $\lim_{n\to\infty}\mathbb{I}_{\mathcal{E}_{r,\gamma'}^{(n)}} \to \mathbb{I}_{\mathcal{E}_{r,\gamma'}}$ almost surely, for $\gamma' \neq \gamma$.

Proof: Suppose that, for some value of w_u, $\mathbb{I}_{\mathcal{E}_u}(w_u) = 1$, i.e., $\min_{\gamma\in\mathcal{S}}(\gamma+\xi_{out}Q_{out}(u,\gamma,w_u)) > -\xi_{relay} - V^*(1) + V^*(u+1)$. Now, $V^*(1)$ and $V^*(u+1)$ are continuous in (ξ_{out},ξ_{relay}) for all $1 \leq u \leq r$ (see Theorem 5). Hence, there exists an integer n_0 large enough, such that for all $n > n_0$, we have $\min_{\gamma\in\mathcal{S}}(\gamma+\xi_n Q_{out}(u,\gamma,w_u)) > -\xi_{relay} - \left(V^{(n,*)}(1) + V^{(n,*)}(u+1)\right)\Big|_{\xi_{out}=\xi_n}$, i.e., $\mathbb{I}_{\mathcal{E}_u^{(n)}}(w_u) = 1$ for all $n > n_0$. Hence, $\mathbb{I}_{\mathcal{E}_u^{(n)}}(w_u) \to \mathbb{I}_{\mathcal{E}_u}(w_u)$ if $\mathbb{I}_{\mathcal{E}_u}(w_u) = 1$. For the case $\mathbb{I}_{\mathcal{E}_u}(w_u) = 0$, we can have similar arguments. This proves the first part of the claim, and second part can be proved by similar arguments. □

Now, $\mathbb{I}_{\mathcal{E}^{(n)}} = \prod_{\gamma'\neq\gamma}\mathbb{I}_{\mathcal{E}_{r,\gamma'}^{(n)}}\prod_{u\leq r-1}\mathbb{I}_{\mathcal{E}_u^{(n)}} \times \mathbb{I}_{\overline{\mathcal{E}_r^{(n)}}}$. By Claim 1, $\mathbb{I}_{\mathcal{E}^{(n)}} \to \mathbb{I}_{\mathcal{E}}$ almost surely as $n \to \infty$. Hence, by Dominated Convergence Theorem, we have $\mathbb{E}(\mathbb{I}_{\mathcal{E}^{(n)}}) \to \mathbb{E}(\mathbb{I}_{\mathcal{E}})$.

We can prove the same statement for $r = B$ in a similar method; but we need to define $\mathcal{E} = \cap_{\gamma'\neq\gamma}\mathcal{E}_{B,\gamma'}\cap_{u\leq B-1}\mathcal{E}_u$.

Hence, the lemma is proved. □

C.2 Proof of Theorem 9

We denote the shadowing in the link between the potential locations located at distances $i\delta$ and $j\delta$ from the sink node, by the random variable $W_{i,j}$. The sample space Ω is defined to be the collection of all ω such that each ω corresponds to a fixed realization $\{w_{i,j} : i \geq 0, j \geq 0, i > j, 1 \leq i-j \leq B\}$ of shadowing that could be encountered in the deployment process over infinite horizon. Let $\mathcal{F}$ be the Borel σ-algebra on Ω. We also define a sequence of sub-σ fields $\mathcal{F}_k := \sigma\Big(W_{i,j} : i \geq 0, j \geq 0, k \geq i > j, 1 \leq i-j \leq B\Big)$; $\mathcal{F}_k$ is increasing in k, and captures the history of the deployment process up to $k\delta$ distance.

Let us recall the outline of the proof of Theorem 9 in Section 5.2.

C.2.1 *The Faster Time-Scale Iteration of* $\underline{V}^{(k)}$

Let us denote by $\underline{V}^*(\xi_{out},\xi_{relay})$ the value of $\underline{V}^*$, for given ξ_{out} and ξ_{relay}. Let us also define $\overline{a}(k) := \max_{r\in\mathcal{I}_k} a(\nu(r,k))$.

Using the first order Taylor series expansion of the function $\Lambda_{[0,A_1]}(\cdot)$, and using the fact that $\Lambda_{[0,A_1]}(\xi_{out}^{(k-1)}) = \xi_{out}^{(k-1)}$ (since $\xi_{out}^{(k-1)} \in [0,A_1]$), we rewrite the update equation (10) as (13). Now, for the update equation for ξ_{relay} in (13), we can write:

$$\begin{aligned}
&\lim_{\beta\downarrow 0}\frac{\Lambda_{[0,A_2]}\Big(\xi_{relay}^{(k-1)} + \beta(1-\overline{N}U_{N_k})\Big) - \xi_{relay}^{(k-1)}}{\beta}\\
=\ & (1-\overline{N}U_{N_k})\mathbb{I}\{0 < \xi_{relay}^{(k-1)} < A_2\}\\
+\ & (1-\overline{N}U_{N_k})^+\mathbb{I}\{\xi_{relay}^{(k-1)} = 0\}\\
-\ & (1-\overline{N}U_{N_k})^-\mathbb{I}\{\xi_{relay}^{(k-1)} = A_2\}
\end{aligned}$$

where $y^+ = \max\{y,0\}$ and $y^- = -\min\{y,0\}$. We can write similar expression for the $\xi_{out}^{(k)}$ update. Since outage probabilities and placement distances are bounded quantities, and since $N_k \geq \lfloor\frac{k}{B}\rfloor$ and $\lim_{k\to 0}\frac{b(\lfloor\frac{k}{B}\rfloor)}{\overline{a}(k)} = 0$, we have:

$$\lim_{k\to\infty}\left(\frac{b(N_k)}{\overline{a}(k)}\left(\lim_{\beta\downarrow 0}\left(\Lambda_{[0,A_1]}\left(\xi_{out}^{(k-1)} + \beta(Q_{out}^{(N_k,N_{k-1})} - \overline{q}U_{N_k})\right) - \xi_{out}^{(k-1)}\right)/\beta + \frac{o(b(N_k))}{b(N_k)}\right)\right) = 0$$

Similar claim can be made for ξ_{relay} update.

Lemma 3: Under Algorithm 3, the faster timescale iterates $\{\underline{V}^{(k)}\}_{k\geq 1}$ are almost surely bounded.

Proof: Note that, (13) combines the faster and slower timescale iterations in a single timescale where the step size is $\overline{a}(n)$. We will now use the theory from [44, Section 3] to prove this lemma.

Note that, the R.H.S. of the faster timescale iteration in (13) is Lipschitz continuous in both faster and slower timescale iterates. Hence, the first part of [44, Assumption 2.1] is satisfied.

[44, Assumption 2.2] can be checked, using similar arguments as in checking [36, Assumption $5(ii)$] in the proof of Theorem 7.

Also, $\sum_{n=1}^{\infty}\overline{a}(n) \geq \sum_{n=1}^{\infty}a(n) = \infty$ and $\sum_{n=1}^{\infty}\overline{a}^2(n) \leq \sum_{n=1}^{\infty}a^2(\lfloor\frac{n}{B}\rfloor) < \infty$, which satisfies [44, Assumption 2.3].

Checking [44, Assumption 2.4]: Let us consider the following set of o.d.e. (similar to what we considered in the proof of Theorem 7): $\dot{V}_t(r) = \kappa_t(r)f_r(\underline{V}_t,\xi_{out}(t),\xi_{relay}(t))$ for $r \in \{1,2,\cdots,B\}$, $\dot{\xi}_{out}(t) = 0$ and $\dot{\xi}_{relay}(t) = 0$ (recall the interpretation of $\kappa_t(r)$ from Appendix B). Note that, $\lim_{c\to\infty}\frac{f_r(c\underline{V},c\xi_{out},c\xi_{relay})}{c} = \mathbb{E}_W\min\{\xi_{out}Q_{out}(r,\gamma,W) + \xi_{relay}, -V(1) + V(r+1)\} - V(r)$ for $r \neq B$, and $\lim_{c\to\infty}\frac{f_B(c\underline{V},c\xi_{out},c\xi_{relay})}{c} = \xi_{out}\mathbb{E}_W Q_{out}(B,\gamma,W) + \xi_{relay} - V(B)$. Note that $\frac{f_r(c\underline{V})}{c}$ for all r and $\lim_{c\to\infty}\frac{f_r(c\underline{V},c\xi_{out},c\xi_{relay})}{c}$ all are continuous in $(\underline{V},\xi_{out},\xi_{relay})$, and $\frac{f_r(c\underline{V},c\xi_{out},c\xi_{relay})}{c}$ is decreasing in c. Hence, by Theorem 7.13 of [43], convergence of $\frac{f_r(c\underline{V},c\xi_{out},c\xi_{relay})}{c}$ over compacts is uniform. Hence, one part of [44, Assumption 2.4] is proved. Next, by similar analysis done while checking [36, Assumption 4] in the proof of Theorem 7 (using Lemma 1), we can verify the second part of [44, Assumption 2.4].

Hence, using similar analysis as in [44, Section 3, Theorem 11] (adapted to the case of asynchronous stochastic approximation), we can claim that $||\underline{V}^{(k)}|| \leq C^*(1 + \xi_{out}^{(k)} + \xi_{relay}^{(k)})$ for all $k \geq 1$, for some $C^* > 0$. Now, since the slower timescale iterates are bounded in our problem, the faster timescale iterates are also bounded. This completes the proof of Lemma 3. □

Lemma 4: For Algorithm 3, we have $(\underline{V}^{(k)},\xi_{out}^{(k)},\xi_{relay}^{(k)}) \to \{(\underline{V}^*(\xi_{out},\xi_{relay}),\xi_{out},\xi_{relay}) : (\xi_{out},\xi_{relay}) \in [0,A_1]\times[0,A_2]\}$ almost surely, i.e., $\lim_{k\to\infty}||\underline{V}^{(k)} - \underline{V}^*(\xi_{out}^{(k)},\xi_{relay}^{(k)})|| = 0$ almost surely.

Proof: Note that, the functions $f_r(\underline{V},\xi_{out},\xi_{relay}) =$

$$
\begin{aligned}
V^{(k)}(r) &= V^{(k-1)}(r) + \overline{a}(k)\frac{a(\nu(r,k))}{\overline{a}(k)}\mathbb{I}\{r \in \mathcal{I}_k\}\Big[\min\Big\{\min_{\gamma}(\gamma + \xi_{out}^{(k-1)}Q_{out}(r,\gamma,w_r)) + \xi_{relay}^{(k-1)}, -V^{(k-1)}(1) + V^{(k-1)}(r+1)\Big\} - V^{(k-1)}(r)\Big], \\
&\quad \forall 1 \leq r \leq B-1 \\
V^{(k)}(B) &= V^{(k-1)}(B) + \overline{a}(k)\frac{a(\nu(r,B))}{\overline{a}(k)}\mathbb{I}\{B \in \mathcal{I}_k\}\Big[\min_{\gamma}(\gamma + \xi_{out}^{(k-1)}Q_{out}(B,\gamma,w_B)) + \xi_{relay}^{(k-1)} - V^{(k-1)}(B)\Big] \\
\xi_{out}^{(k)} &= \xi_{out}^{(k-1)} + \mathbb{I}\{N_k = N_{k-1}+1\}\Big(b(N_k)\lim_{\beta\downarrow 0}\frac{\Lambda_{[0,A_1]}\Big(\xi_{out}^{(k-1)} + \beta(Q_{out}^{(N_k,N_{k-1})} - \overline{q}U_{N_k})\Big) - \xi_{out}^{(k-1)}}{\beta} + o(b(N_k))\Big) \\
&= \xi_{out}^{(k-1)} + \mathbb{I}\{N_k = N_{k-1}+1\}\overline{a}(k)\frac{b(N_k)}{\overline{a}(k)}\Big(\lim_{\beta\downarrow 0}\frac{\Lambda_{[0,A_1]}\Big(\xi_{out}^{(k-1)} + \beta(Q_{out}^{(N_k,N_{k-1})} - \overline{q}U_{N_k})\Big) - \xi_{out}^{(k-1)}}{\beta} + \frac{o(b(N_k))}{b(N_k)}\Big) \\
\xi_{relay}^{(k)} &= \xi_{relay}^{(k-1)} + \mathbb{I}\{N_k = N_{k-1}+1\}\Big(b(N_k)\lim_{\beta\downarrow 0}\frac{\Lambda_{[0,A_2]}\Big(\xi_{relay}^{(k-1)} + \beta(1 - \overline{N}U_{N_k})\Big) - \xi_{relay}^{(k-1)}}{\beta} + o(b(N_k))\Big) \\
&= \xi_{relay}^{(k-1)} + \mathbb{I}\{N_k = N_{k-1}+1\}\overline{a}(k)\frac{b(N_k)}{\overline{a}(k)}\Big(\lim_{\beta\downarrow 0}\frac{\Lambda_{[0,A_2]}\Big(\xi_{relay}^{(k-1)} + \beta(1 - \overline{N}U_{N_k})\Big) - \xi_{relay}^{(k-1)}}{\beta} + \frac{o(b(N_k))}{b(N_k)}\Big) \qquad (13)
\end{aligned}
$$

$\mathbb{E}_W\Big[\min\Big\{\min_\gamma(\gamma + \xi_{out}Q_{out}(r,\gamma,W)) + \xi_{relay}, -V(1) + V(r+1)\Big\} - V(r)\Big]$ and $f_B(\underline{V}, \xi_{out}, \xi_{relay}) = \mathbb{E}_W\Big[\min_\gamma(\gamma + \xi_{out}Q_{out}(B,\gamma,W)) + \xi_{relay} - V(B)\Big]$ are Lipschitz continuous in all arguments (by Theorem 5), and the collection of o.d.e. $\dot{\underline{V}}_r(t) = \kappa_t(r) f_r(\underline{V}(t), \xi_{out}, \xi_{relay})$ for all $r \in \{1, 2, \cdots, B\}$ (see [37, Theorem 2, Chapter 7] and the proof of Theorem 7 for an interpretation of $\kappa_t(r)$) has a unique globally asymptotically stable equilibrium $\underline{V}^*(\xi_{out}, \xi_{relay})$ for any $\xi_{out} \geq 0$, $\xi_{relay} \geq 0$ (see Lemma 1 in the proof of Theorem 7). Also, by Theorem 5, $\underline{V}^*(\xi_{out}, \xi_{relay})$ is Lipschitz continuous in ξ_{out} and ξ_{relay}. On the other hand, by Lemma 3 and the projection in the slower timescale, the iterates are almost surely bounded.

Hence, by a similar argument as in the proof [37, Lemma 1, Chapter 6], and by Theorem 7, $(\underline{V}^{(k)}, \xi_{out}^{(k)}, \xi_{relay}^{(k)})$ converges to the internally chain transitive invariant sets of the collection of o.d.e. given by $\dot{V}_r(t) = \kappa_t(r) f_r(\underline{V}(t), \xi_{out}, \xi_{relay})$ for all $r \in \{1, 2, \cdots, B\}$, $\dot{\xi}_{out}(t) = 0$, $\dot{\xi}_{relay}(t) = 0$ (where $\underline{V}(t) := \{V_1(t), V_2(t), \cdots, V_B(t)\}$). Hence, $(\underline{V}^{(k)}, \xi_{out}^{(k)}, \xi_{relay}^{(k)}) \rightarrow \{(\underline{V}^*(\xi_{out}, \xi_{relay}), \xi_{out}, \xi_{relay}) : (\xi_{out}, \xi_{relay}) \in [0, A_1] \times [0, A_2]\}$ and $\lim_{k\to\infty} ||\underline{V}^{(k)} - \underline{V}^*(\xi_{out}^{(k)}, \xi_{relay}^{(k)})|| = 0$. □

Remark: Lemma 4 does not guarantee the convergence of the slower timescale iterates.

C.2.2 *The slower timescale iteration*

We will pose the slower timescale update as a projected stochastic approximation (see [39, Equation 5.3.1]). **In order to do that and to avoid complicated notation, for the rest of this appendix we will denote by $\underline{V}^{(k)}$, $\xi_{out}^{(k)}$ and $\xi_{relay}^{(k)}$ the values of the corresponding variable after placing the k-th relay and performing the update (earlier they were defined to be the iterates after a decision is made at the k-th step).** Let us also recall the definition of the functions $\overline{Q}_{out}(\cdot,\cdot,\cdot)$, $\overline{Q}^*_{out}(\cdot,\cdot,\cdot)$, $\overline{U}(\cdot,\cdot,\cdot)$, $\overline{U}^*(\cdot,\cdot,\cdot)$. Let us define the functions $\overline{Q}'_{out}(\underline{V}^{(k-1)}, \xi_{out}^{(k-1)}, \xi_{relay}^{(k-1)})$ and $\overline{U}'(\underline{V}^{(k-1)}, \xi_{out}^{(k-1)}, \xi_{relay}^{(k-1)})$ to be the mean link outage and mean length of the k-th link that is created by Algorithm 3 (using the two-timescale update) starting with $\underline{V}^{(k-1)}$, $\xi_{out}^{(k-1)}$ and $\xi_{relay}^{(k-1)}$ (which are obtained by the algorithm after placing the $(k-1)$-st relay and and doing the learning/update operation; note that, these quantities are obtained after placing $(k-1)$ nodes and not at the $(k-1)$-th step).

The difference between $\overline{U}'(\underline{V}^{(k-1)}, \xi_{out}^{(k-1)}, \xi_{relay}^{(k-1)})$ and $\overline{U}(\underline{V}^{(k-1)}, \xi_{out}^{(k-1)}, \xi_{relay}^{(k-1)})$ can be explained as follows. $\overline{U}(\underline{V}^{(k-1)}, \xi_{out}^{(k-1)}, \xi_{relay}^{(k-1)})$ is the mean length of the k-th link where no quantity is updated in the process of measurements made to create the k-th link; hence, $\overline{U}(\underline{V}^{(k-1)}, \xi_{out}^{(k-1)}, \xi_{relay}^{(k-1)})$ is the mean placement distance of a stationary *policy which is similar to Algorithm 1 except that ξ_{out}, ξ_{relay} and $\underline{V}^*$ are replaced by $\xi_{out}^{(k-1)}$, $\xi_{relay}^{(k-1)}$ and $\underline{V}^{(k-1)}$ respectively. On the other hand, $\overline{U}'(\underline{V}^{(k-1)}, \xi_{out}^{(k-1)}, \xi_{relay}^{(k-1)})$ is the mean length of the k-th link created under Algorithm 3 (with $(\underline{V}^{(k-1)}, \xi_{out}^{(k-1)}, \xi_{relay}^{(k-1)})$ as starting parameters), where the iterates are updated at each step between placement of the $(k-1)$-th node and the k-th node.*

Let us denote by $\mathcal{G}$ the set $[0, A_1] \times [0, A_2]$, defined by the following constraints:

$$-\xi_{out} \leq 0, \xi_{out} \leq A_1, -\xi_{relay} \leq 0, \xi_{relay} \leq A_2 \qquad (15)$$

Clearly, projection onto the set $\mathcal{G}$ is nothing but coordinate wise projection.

We rewrite the slower timescale iteration in (10) as (14) (note the definitions of the functions $f_1(\xi_{out}, \xi_{relay})$, $f_2(\xi_{out}, \xi_{relay})$, $g_1(\underline{V}, \xi_{out}, \xi_{relay})$, $g_2(\underline{V}, \xi_{out}, \xi_{relay})$, $l_1(\underline{V}, \xi_{out}, \xi_{relay})$ and $l_2(\underline{V}, \xi_{out}, \xi_{relay})$ in (14)). The random variables $M_1^{(k)}$ and $M_2^{(k)}$ are two zero mean Martingale difference noise sequences w.r.t. $\mathcal{F}_{k-1}$ (information available up to the $(k-1)$-st placement instant); this happens due to i.i.d. shadowing across links.

(14) has the form of a projected stochastic approximation (see [39, Equation 5.3.1]). In order to show the desired conver-

$$
\begin{aligned}
\xi_{out}^{(k)} &= \Lambda_{\mathcal{G}}\Big(\xi_{out}^{(k-1)} + b(k)\Big(Q_{out}(U_k,\Gamma_k,W_{U_k}) - \overline{q}U_k\Big)\Big) \\
&= \Lambda_{\mathcal{G}}\Big(\xi_{out}^{(k-1)} + b(k)\Big(\underbrace{\overline{Q}^*_{out}(\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) - \overline{q}\overline{U}^*(\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)})}_{:=f_1(\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)})} \\
&\quad + \underbrace{\overline{Q}_{out}(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) - \overline{q}\overline{U}(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) - f_1(\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)})}_{:=g_1(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)})} \\
&\quad + \underbrace{\overline{Q}'_{out}(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) - \overline{q}\overline{U}'(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) - \Big(\overline{Q}_{out}(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) - \overline{q}\overline{U}(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)})\Big)}_{:=l_1(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)})} \\
&\quad + \underbrace{Q_{out}(U_k,\Gamma_k,W_{U_k}) - \overline{q}U_k - \Big(\overline{Q}'_{out}(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) - \overline{q}\overline{U}'(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)})\Big)}_{:=M_1^{(k)}}\Big)\Big) \\
&= \Lambda_{\mathcal{G}}\Big(\xi_{out}^{(k-1)} + b(k)\Big(f_1(\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) + g_1(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) + l_1(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) + M_1^{(k)}\Big)\Big) \\
\xi_{relay}^{(k)} &= \Lambda_{\mathcal{G}}\Big(\xi_{out}^{(k-1)} + b(k)\Big(1 - \overline{N}U_k\Big)\Big) \\
&= \Lambda_{\mathcal{G}}\Big(\xi_{relay}^{(k-1)} + b(k)\Big(\underbrace{1 - \overline{NU}^*(\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)})}_{:=f_2(\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)})} \\
&\quad + \underbrace{1 - \overline{NU}(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) - f_2(\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)})}_{:=g_2(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)})} \\
&\quad + \underbrace{1 - \overline{NU}'(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) - \Big(1 - \overline{NU}(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)})\Big)}_{:=l_2(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)})} \\
&\quad + \underbrace{1 - \overline{N}U_k - \Big(1 - \overline{NU}'(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)})\Big)}_{:=M_2^{(k)}}\Big)\Big) \\
&= \Lambda_{\mathcal{G}}\Big(\xi_{relay}^{(k-1)} + b(k)\Big(f_2(\xi_{relay}^{(k-1)},\xi_{relay}^{(k-1)}) + g_2(\underline{V}^{(k-1)},\xi_{relay}^{(k-1)},\xi_{relay}^{(k-1)}) + l_2(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) + M_2^{(k)}\Big)\Big) \qquad (14)
\end{aligned}
$$

gence of the iterates in (14), we will use [39, Theorem 5.3.1*]; this requires us to check five conditions from [39], which is done in the next subsection.* □

C.2.3 Checking the five conditions from [39]

We will first present a lemma that will be useful for checking one condition.

Lemma 5: Under Assumption 2, the quantities $\overline{\Gamma}(\underline{V},\xi_{out},\xi_{relay})$, $\overline{Q}_{out}(\underline{V},\xi_{out},\xi_{relay})$ and $\overline{U}(\underline{V},\xi_{out},\xi_{relay})$ are continuous in $\underline{V}$, ξ_{out} and ξ_{relay}.

Proof: The proof is similar to that of Theorem 8. □

Now, we will check conditions A5.1.3, A5.1.4, A5.1.5, A5.3.1. and A5.3.2 from [39].

*Checking Condition A*5.1.3*:* We need $f_1(\cdot,\cdot)$ and $f_2(\cdot,\cdot)$ to be continuous functions; this holds by Theorem 8. □

*Checking Condition A*5.1.4*:* This condition is satisfied by the choice of the sequence $\{b(k)\}_{k\geq 1}$. □

*Checking Condition A*5.1.5*:* This condition requires that $\lim_{k\to\infty} g_1(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) = 0$, $\lim_{k\to\infty} g_2(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) = 0$, $\lim_{k\to\infty} l_1(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) = 0$ and $\lim_{k\to\infty} l_2(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) = 0$ almost surely.

We can find a probability 1 subset of the sample space Ω, such that for any sample path in this subset the conclusions of Lemma 3 and Lemma 4 hold. Take one such sample path ω. By Lemma 3, for this sample path ω, we can find a compact subset $\mathcal{C} \subset \mathbb{R}^B$ such that $(\underline{V}^{(k)},\xi_{out}^{(k)},\xi_{relay}^{(k)})$ lies inside the compact set $\mathcal{C}\times[0,A_1]\times[0,A_2]$ for all $k \geq 1$ along this sample path.

By Lemma 5 and the fact that continuous functions are uniformly continuous over compact sets, we can say that $\overline{Q}_{out}(\underline{V},\xi_{out},\xi_{relay})$, $\overline{\Gamma}(\underline{V},\xi_{out},\xi_{relay})$ and $\overline{U}(\underline{V},\xi_{out},\xi_{relay})$ are uniformly continuous over the compact set $\mathcal{C}\times[0,A_1]\times[0,A_2]$. Now, the Euclidean distance between $(\underline{V}^{(k)},\xi_{out}^{(k)},\xi_{relay}^{(k)})$ and $(\underline{V}^*(\xi_{out}^{(k)},\xi_{relay}^{(k)}),\xi_{out}^{(k)},\xi_{relay}^{(k)})$ converges to 0 along the sample path ω. Hence, by uniform continuity, we can say that $\lim_{k\to\infty}|\overline{Q}_{out}(\underline{V}^{(k)},\xi_{out}^{(k)},\xi_{relay}^{(k)}) - \overline{Q}_{out}(\underline{V}^*(\xi_{out}^{(k)},\xi_{relay}^{(k)}),\xi_{out}^{(k)},\xi_{relay}^{(k)})| = 0$ and $\lim_{k\to\infty}|\overline{U}(\underline{V}^{(k)},\xi_{out}^{(k)},\xi_{relay}^{(k)}) -$

$\overline{U}(\underline{V}^*(\xi_{out}^{(k)},\xi_{relay}^{(k)}),\xi_{out}^{(k)},\xi_{relay}^{(k)})| = 0$ along this sample path ω. Hence, $\lim_{k\to\infty} g_1(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) = 0$ and $\lim_{k\to\infty} g_2(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) = 0$ almost surely.

On the other hand, since $\mathcal{C}$ is bounded, we can say that $\{\underline{V}^{(k)}\}_{k\geq 1}$ is bounded for the chosen ω. In a similar way as in the proof of Theorem 8, in case of Lemma 5 we can show that $g(r,\gamma)$ is continuous in $\overline{V}$, ξ_{out} and ξ_{relay}. Now, between the placement of the $(k-1)$-st relay and k-th relay, at each step, $g(r,\gamma)$ for all $r \in \{1,2,\cdots,B\}, \gamma \in \mathcal{S}$ can change at most by an amount $K^*a(k-1-B)$ (for a suitable constant $K^* > 0$), and hence we can claim that $\lim_{k\to\infty} |\overline{U}'(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) - \overline{U}(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)})| = 0$, $\lim_{k\to\infty} |\overline{Q}'_{out}(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) - \overline{Q}_{out}(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)})| = 0$. Hence, we obtain that $\lim_{k\to\infty} l_1(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) = 0$ and $\lim_{k\to\infty} l_2(\underline{V}^{(k-1)},\xi_{out}^{(k-1)},\xi_{relay}^{(k-1)}) = 0$.

Also, $g_1(\underline{V}^{(k)},\xi_{out}^{(k)},\xi_{relay}^{(k)})$, $g_2(\underline{V}^{(k)},\xi_{out}^{(k)},\xi_{relay}^{(k)})$, $l_1(\underline{V}^{(k)},\xi_{out}^{(k)},\xi_{relay}^{(k)})$ and $l_2(\underline{V}^{(k)},\xi_{out}^{(k)},\xi_{relay}^{(k)})$ are uniformly bounded across $k \geq 1$, since the outage probabilities and placement distances are bounded quantities.

Hence, this condition is satisfied.

*Checking Condition A*5.3.1*:* This condition is easy to check, and done in [1, Appendix E, Section C4]. □

*Checking Condition A*5.3.2*:* This condition is easy to check, and done in [1, Appendix E, Section C4]. □

C.2.4 Finishing the Proof of Theorem 9

Consider the function $h(\xi_{out},\xi_{relay}) := \left(\frac{f_1(\xi_{out},\xi_{relay})}{\overline{U}^*(\xi_{out},\xi_{relay})}, \frac{f_2(\xi_{out},\xi_{relay})}{\overline{U}^*(\xi_{out},\xi_{relay})}\right) = \left(\frac{\overline{Q}^*_{out}(\xi_{out},\xi_{relay})}{\overline{U}^*(\xi_{out},\xi_{relay})} - \overline{q}, \frac{1}{\overline{U}^*(\xi_{out},\xi_{relay})} - \overline{N}\right)$ and the map:

$$\overline{\Lambda}_{\mathcal{G}}(h(\xi_{out},\xi_{relay})) = \lim_{0<\beta\to 0} \frac{\Lambda_{\mathcal{G}}\Big((\xi_{out},\xi_{relay}) + \beta h(\xi_{out},\xi_{relay}))\Big) - (\xi_{out},\xi_{relay})}{\beta} \tag{16}$$

Lemma 6: If $(\xi_{out},\xi_{relay}) \in [0,A_1] \times [0,A_2]$ is a zero of $\overline{\Lambda}_{\mathcal{G}}\left(\frac{f_1(\xi_{out},\xi_{relay})}{\overline{U}^*(\xi_{out},\xi_{relay})}, \frac{f_2(\xi_{out},\xi_{relay})}{\overline{U}^*(\xi_{out},\xi_{relay})}\right)$, then $(\underline{V}^*(\xi_{out},\xi_{relay}),\xi_{out},\xi_{relay}) \in \mathcal{K}(\overline{q},\overline{N})$, provided that A_1 and A_2 are chosen using the procedure described in Section 5.

Proof: The proof is similar to the proof of [1, Lemma 9, Appendix E, Section C5]. □

Now, by using similar arguments as in [1, Appendix E, Section C5] and using [39, Theorem 5.3.1], We can show that the iterates $(\xi_{out}^{(k)},\xi_{relay}^{(k)})$ will converge almost surely to the set of stationary points of the o.d.e. $(\dot{\xi}_{out}(t),\dot{\xi}_{relay}(t)) = \overline{\Lambda}_{\mathcal{G}}\left(\frac{f_1(\xi_{out}(t),\xi_{relay}(t))}{\overline{U}^*(\xi_{out}(t),\xi_{relay}(t))}, \frac{f_2(\xi_{out}(t),\xi_{relay}(t))}{\overline{U}^*(\xi_{out}(t),\xi_{relay}(t))}\right)$.

Using this result and using Lemma 4 and Lemma 6, we obtain that $(\underline{V}^{(k)},\xi_{out}^{(k)},\xi_{relay}^{(k)}) \to \mathcal{K}(\overline{q},\overline{N})$ almost surely, where $k\delta$ can be the distance from the sink or k can be the index of a placed relay node (the result holds for both interpretations of k). This completes the proof of Theorem 9. □