Endhost-Based Shortest Path Routing in Dynamic Networks: An Online Learning Approach

Ting He*, Dennis Goeckel†, Ramya Raghavendra*, and Don Towsley‡
*IBM T.J. Watson Research Center, Yorktown, NY, USA. Email: {the,rraghav}@us.ibm.com
†Department of ECE, University of Massachusetts, Amherst, MA, USA. Email: goeckel@ecs.umass.edu
‡Department of CS, University of Massachusetts, Amherst, MA, USA. Email: towsley@cs.umass.edu

***Abstract*—We consider the problem of endhost-based shortest path routing in a network with unknown, time-varying link qualities. Endhost-based routing is needed when internal nodes of the network do not have the scope or capability to provide globally optimal paths to given source-destination pairs, as can be the case in networks consisting of autonomous subnetworks or those with endhost-based routing restrictions. Assuming the source can probe links along selected paths, we formulate the problem as an online learning problem, where an existing solution achieves a performance loss (called regret) that is logarithmic in time with respect to (wrt) an offline algorithm that knows the link qualities. Current solutions assume coupled probing and routing; in contrast, we give a simple algorithm based on decoupled probing and routing, whose regret is only constant in time. We then extend our solution to support multi-path probing and cooperative learning between multiple sources, where we show an inversely proportional decay in regret wrt the probing rate. We also show that without the decoupling, the regret grows at least logarithmically in time, thus establishing decoupling as critical for obtaining constant regret. Although our analysis assumes certain conditions (i.i.d.) on link qualities, our solution applies with straightforward amendments to much broader scenarios where these conditions are relaxed. The efficacy of the proposed solution is verified by trace-driven simulations.**

I. Introduction

Consider the problem of finding the shortest path between a given source-destination pair in a network with known topology but unknown, possibly time-varying link metrics (e.g., per-hop delays). Such a problem arises during the bootstrapping of routing after network discovery. Even if the source can measure links by sending probes (e.g., traceroute packets) along selected paths, a single measurement may not accurately characterize a link due to noise in the measurement process and/or inherent dynamics of link qualities, e.g., due to fading or transient interference. To learn the true quality of a link, the source has to probe it repeatedly to accumulate sufficiently many samples, assuming a probe on a path returns realizations of metrics for all the links on this path (as in the case of traceroute). In contrast to a classic shortest path routing (SPR) problem where the average link metrics are assumed to be known *a priori*, the source faces an online routing problem where it has to learn about the link metrics from *runtime* observations. Although under basic stationarity conditions, any probing strategy that samples each link infinitely often (e.g., round-robin) will eventually learn the true average link metrics and therefore converge to the true shortest path (in the average sense), the probing strategy affects performance over a finite time horizon, as it determines the number of times each link is measured, which in turn determines the quality of link estimation and thus the quality of routing. It is therefore desirable to have an online probing/routing algorithm that not only converges to the optimal route asymptotically, but also minimizes the use of suboptimal routes during the convergence process. A widely-adopted measure of this suboptimality is called *regret*, defined as the gap (e.g., extra delay) between the routes selected by an online algorithm and the optimal route that is the shortest based on the true average link metrics, accumulated over time [1]. The goal of the source is thus to design a probing/routing strategy to minimize its average regret.

In this paper, we focus on *endhost-based routing*. In contrast to hop-by-hop routing where the source relies on intermediate nodes to choose a desirable path toward its destination, endhost-based routing puts the duty of selecting the right path on the endhosts (i.e., source/destination). Endhost-based routing is needed when intermediate nodes do not have the scope or capability to choose a globally optimal path. One example is a wireless ad hoc network of multiple autonomous subnetworks, where the shortest path may interleave between different subnetworks, but nodes in each subnetwork are only able to optimize paths within the same subnetwork. Another example is a military coalition network, where the set of candidate relay nodes may differ for different endhosts based on attributes of the endhosts (e.g., trust level, command chain). Yet another example is a heterogeneous network supporting multiple routing metrics (e.g., delay, loss rate), where different endhosts may prefer to optimize different metrics. Compared with hop-by-hop routing, the lack of knowledge on link qualities imposes a greater challenge to endhost-based routing because learning is only performed by the endhosts and is limited by their capability of observing the network (e.g., one path at a time). Therefore, the regret generally grows not only with time, but also with the size of the network. It is thus desirable to have a learning algorithm with the minimum rate of regret growth wrt both time and network size.

Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-06-3-0001. The views and conclusions are those of the authors and do not represent the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding copyright notation.

A. Related Work

Learning-based routing has been proposed to deal with dynamically changing networks, especially in the wireless context. Most existing solutions focus on flat ad hoc networks where routing is optimized hop by hop and all nodes learn their local network states simultaneously; see [2] and references therein. In this setting, reinforcement learning techniques have been used where nodes select next-hop relays based on the experiences of previous transmissions. Measuring the performance by the regret, [3] gives an order-optimal solution whose regret is $O(\log T)$ in time T.

The problem becomes more challenging for endhost-based routing, where existing solutions can be divided into two classes according to whether they rely on link measurements or path measurements. The problem of routing based on *link measurements* was first studied in [4] in an adversarial setting where link weights are selected by an adversary to yield the worst routing performance. A benign environment as considered in this paper is better modeled by a stochastic setting, where link weights follow unknown but consistent stochastic rules. Under this setting, a previous study in [5] shows that a regret of $O(N^4 \log T)$ can be achieved, where N is the number of links. Compared with the $O(\log T)$-regret for hop-by-hop routing, the regret for endhost-based routing also grows with the network size. Approaches based on *path measurements* only rely on the sum weights along probed paths. If the paths are disjoint, then the problem is reduced to the classic multi-armed bandit (MAB) problem [1] where the weight of each path is learned separately, and the optimal regret is $\Theta(M \log T)$, where M is the number of paths. In general, common links between paths can be exploited to accelerate learning, which improves the regret to $O(N^2 \log^3 T)$ [6] or $O(N^3 \log T)$ [7] (this is an improvement since M can be exponential in N). Our work belongs to endhost-based routing under link measurements.

A subtlety here is whether probing and routing are coupled or not. Reinforcement learning solutions [2], [5], [6] implicitly assume *coupled probing/routing*, i.e., the probed links/paths are also used for routing, which occurs when probes are piggybacked on data packets. Alternatively, *decoupled probing/routing* uses dedicated probes, e.g., traceroute packets, and one is allowed to probe one path but route on another.

B. Summary of Results

We consider the problem of endhost-based bootstrapping of SPR in a dynamic network. Formulated as online learning, the goal is not only to converge to the optimal route asymptotically but also to minimize the routing suboptimality, measured by regret, during the convergence. Our specific contributions are:

Regret lower bound under coupled probing/routing: We prove a first lower bound on the optimal regret under arbitrary network topology, which is of order $\Omega(\log T)$, with a constant factor depending on the topology. This result complements existing achievability result in [5].

Constant-regret solution under decoupled probing/routing: We then show that, by decoupling probing and routing, regret is immediately reduced to a constant (in time), where a simple greedy algorithm achieves a regret of $O(N^4)$. Combined with the above, this establishes the decoupling as a critical condition for obtaining constant regret.

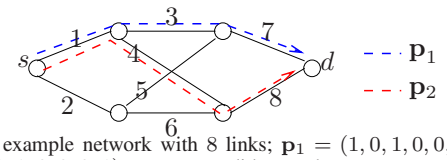


Fig. 1. An example network with 8 links; $\mathbf{p}_1 = (1, 0, 1, 0, 0, 0, 1, 0)$ and $\mathbf{p}_2 = (1, 0, 0, 1, 0, 0, 0, 1)$ are two candidate paths.

Extensions to accelerate learning: We extend our baseline solution to accelerate learning, either by probing multiple paths between the same pair of endhosts or by sharing information with other endhosts, where we show a $1/S$-factor reduction in regret for a probing rate of S.

Evaluation on real traces: We verify our results using simulations on real network traces. The simulations show that the proposed solution significantly reduces the regret of the existing solution (by $60\%+$), and the extensions provide substantial improvement with a controllable measurement overhead.

Practical considerations: Although our analysis is based on certain theoretical assumptions, we show that the main result holds when these assumptions are substantially relaxed, and our algorithm can be easily amended to handle practical challenges such as temporal link correlation, delayed probes, and persistent changes in link qualities.

We point out that our solution still requires basic in-network support to obtain link measurements, e.g., nodes should support Internet Control Message Protocol (ICMP). Moreover, SPR itself has known issues such as congestion under heavy traffic. While acknowledging these potential issues, we will only focus on the learning aspect in this paper, and leave detailed studies of these other issues to future work.

The rest of the paper is organized as follows. Section II formulates the problem. Section III presents solutions for a baseline case, followed by extensions in Section IV. Section V evaluates the proposed solutions. Section VI discusses practical challenges. Then Section VII concludes the paper.

II. Problem Formulation

A. Network Model

We model the topology of the given network as a graph $G(V, E)$, where V is the set of nodes and E is the set of links, with $|E| = N$. For a given source-destination pair s and d, let $\mathcal{P}$ denote the set of all candidate paths from s to d, with $|\mathcal{P}| = M$. We represent each path by a binary vector $\mathbf{p} = (p_i)_{i=1}^N \in \{0, 1\}^N$, where p_i indicates whether link i is on the path or not, as shown in Fig. 1. Let $\mathbf{A} \triangleq [\mathbf{p}_1 \dots \mathbf{p}_M]^T$ be an $M \times N$ *routing matrix* whose rows correspond to the paths in $\mathcal{P}$. Which paths are allowed in $\mathcal{P}$ depend on the endhosts and the underlying routing restrictions, e.g., $\mathcal{P}$ can be limited to simple (i.e., cycle-free) paths consisting of nodes with at least the same trust level and/or in the same command chain as the endhosts. The number of paths M can grow exponentially wrt the number of

links N. Note that $\mathcal{P}$ is only used for formulation; the solution does not have to enumerate the paths explicitly.

Each link $i \in E$ has an unknown *instantaneous* weight $L_i(t)$ at time slot t that varies over time. We do not make any specific assumptions about the distributions of $L_i(t)$, and $L_i(t)$'s may be arbitrarily correlated across links. We only assume that for each link i, $L_i(t)$ has a finite support, and we will formally establish results for the case when $L_i(1), \ldots, L_i(t) \ldots$ are a sequence of i.i.d. random variables. Let $l_i \triangleq \mathbb{E}[L_i(t)]$ denote the *mean* link weight, also unknown. We hasten to note that our results actually hold under a weaker condition, where link weights may be temporally correlated; see Section VI-A. Without loss of generality, we transform the weights such that $L_i(t) \in [0,\ 1]$ for all i and t.

We assume the link weights to be additive, i.e., the aggregate weight along a path $\mathbf{p}$ equals $W_{\mathbf{p}}(t) \triangleq \sum_i p_i L_i(t)$. A typical additive metric is link delay, and there are other metrics that can be written in additive forms (e.g., log of delivery ratio).

B. Main Problem: Online SPR

The problem is to identify, for each given source-destination pair, the minimum average weight path $\mathbf{p}^*$, i.e., $\mathbf{p}^* \triangleq \arg\min_{\mathbf{p}\in\mathcal{P}} \sum_i p_i l_i$. If the mean link weights l_i's are known, this is the classic SPR problem and can be solved efficiently using Dijkstra's algorithm. The challenge here is that the source has to learn about l_i's via runtime observations while trying to optimize the route, which makes it an *online SPR* problem. We emphasize that the goal is to identify the shortest path, not necessarily to use this path, which is motivated by the bootstrapping scenario when sources are discovering optimal paths to potential destinations even if they do not immediately have traffic to send. Accordingly, an implicit assumption is that link weights are independent of the routing decisions.

Consider source routing, where source s periodically measures the network by sending probes along chosen paths and adjusts its route selection. We assume *link-by-link measurements* on the probed paths as in traceroute, i.e., if path $\mathbf{p}(t)$ is probed in slot t, then all link weights on this path (i.e., $L_i(t)$'s with $p_i(t) = 1$) will be observed by the end of slot t. Meanwhile, s selects a (possibly different) routing path $\hat{\mathbf{p}}(t)$ to use in slot t, based on link measurements in slots $1, \ldots, t-1$. Our goal is to jointly select a *probing path* $\mathbf{p}(t)$ and a *routing path* $\hat{\mathbf{p}}(t)$ in each slot t such that $\hat{\mathbf{p}}(t)$ converges to $\mathbf{p}^*$ as fast as possible. Specifically, we want to minimize the total average routing cost $\sum_{t=1}^{T}\sum_{i=1}^{N} \hat{p}_i(t) l_i$ up to time T for each value of T. Although we assume that probes can complete within one slot, our results remain valid in the presence of probing delays; see Section VI-B.

The performance of an online routing algorithm is measured by its *regret*, which denotes the extra cost the algorithm incurs compared with an offline algorithm that knows the mean link weights, i.e., the regret at time T is given by

$$\mathcal{R}_T = \sum_{t=1}^{T}\sum_{i=1}^{N} (\hat{p}_i(t) l_i - p_i^* l_i). \tag{1}$$

Minimizing the routing cost up to T is equivalent to minimizing the regret $\mathcal{R}_T$. Note that $\mathcal{R}_T$ is a random variable because $\hat{\mathbf{p}}(t)$ depends on link measurements, which are random variables. Table I summarizes the main notations used in the paper. All proofs are given in [8].

TABLE I
MAIN NOTATIONS

N/M	number of links/paths
$\mathcal{P}/\mathcal{P}'$	set of all/suboptimal paths
$\mathbf{p}^*$	optimal path (on average link weights)
$\mathbf{p}(t)/\hat{\mathbf{p}}(t)$	probing/routing path at slot t
$l_i/\hat{l}_i(t)$	true/empirical mean weight of link i at t
$m_i(t)$	number of measurements for link i at t
H	maximum number of links per path
$\Delta_{\min}/\Delta_{\max}$	minimum/maximum gap between the weights of optimal and suboptimal paths
S	number of probes per slot

III. Single-Source Learning

Consider the simple case when there is only one source-destination pair probing a single path at a time. This problem has been studied in [5] with an additional constraint that $\hat{\mathbf{p}}(t) = \mathbf{p}(t)$ for all t, i.e., routing and probing are *coupled*. In general, however, routing and probing are *decoupled*, allowing the source to send probes on one path and data packets on another. Intuitively, coupled probing/routing tries to *probe* the shortest path every time, whereas decoupled probing/routing tries to *find* the shortest path. We will show below that this difference leads to not only different probing and routing strategies, but also substantially different regrets.

A. Coupled Probing and Routing

Coupled probing/routing is performed by piggybacking link measurements on data packets. The problem has been studied in terms of algorithm design and achievable performance. We revisit the problem with a focus on establishing bounds on the optimal performance, which is crucial in our later comparison of coupled probing/routing with decoupled probing/routing.

1) Algorithm and Regret Upper Bound: In [5], an algorithm called *Learning with Linear Cost (LLC)* was proposed for path selection under coupled probing/routing. We include the algorithm in Algorithm 1 for completeness. Here $\hat{l}_i(t)$ denotes the empirical mean weight of link i at the end of slot t based on measurements at $s = 1, \ldots, t$, and $m_i(t)$ the number of times link i is measured up to t. Note that the routing decision in slot t is based on the previous observation $(\hat{l}_i(t-1),\ m_i(t-1))_{i=1}^{N}$ (time index is omitted in the algorithm). The main step is line 3, which essentially solves a SPR problem, with link weights estimated by $\hat{l}_i - \sqrt{(H+1)\log t / m_i}$ ($H \triangleq \max_{\mathbf{p}} \sum_i p_i$ is the maximum number of links on a path). Thus, even if the number of paths M can be exponentially large, LLC can be implemented efficiently based on Dijkstra's algorithm.

Algorithm 1 Learning with Linear Cost (LLC)

Require: Candidate paths $\mathcal{P}$.

Ensure: Select a path to route and probe at each time slot and compute estimated mean link weights $(\hat{l}_i)_{i=1}^N$.

1: Initialization: select paths $\mathbf{p}(1), \ldots, \mathbf{p}(N)$ s.t. $p_i(i) = 1$ to measure each link at least once, compute $(\hat{l}_i,\ m_i)_{i=1}^N$ accordingly

2: **for** $t = N+1, \ldots$ **do**

3: Select the path $\mathbf{p}(t)$ such that $\mathbf{p}(t) = \arg\min_{\mathbf{p}\in\mathcal{P}} \sum_i p_i \left(\hat{l}_i - \sqrt{\frac{(H+1)\log t}{m_i}}\right)$

4: Update $(\hat{l}_i,\ m_i)_{i=1}^N$ accordingly

Intuitively, a path is selected because it either shows good average performance in the past, i.e., $\sum_i p_i \hat{l}_i$ is small, or contains under-measured links, i.e., $-\sum_i p_i \cdot \sqrt{(H+1)\log t/m_i}$ is small due to small m_i's (line 3). Rigorously, its average regret is bounded as follows.

Theorem 3.1 ([5]): The average regret under LLC is $O(N^4 \log T)$ and specifically[1],

$$\mathbb{E}[\mathcal{R}_T] \leq \frac{4H^2(H+1)N\Delta_{\max}}{\Delta_{\min}^2}\log T + (1+\frac{\pi^2}{3}H)N\Delta_{\max},$$

where $\Delta_{\min} \triangleq \min_{\mathbf{p}\in\mathcal{P}'} \sum_i (p_i l_i - p_i^* l_i)$, $\Delta_{\max} \triangleq \max_{\mathbf{p}\in\mathcal{P}'} \sum_i (p_i l_i - p_i^* l_i)$ (see Table I). Note that $H = O(N)$.

Note that the $O(N^4 \log T)$ upper bound holds under arbitrary topology, even if the number of paths M is exponential in N; the same applies to all upper bounds in the sequel.

2) Regret Lower Bound: Missing from the solution is the lower bound, i.e., the minimum regret of any online algorithm. We provide below the first regret lower bound for coupled probing and routing that holds under any topology.

Consider the case when link weights are independent across links (and i.i.d. over time). From the theory of multi-armed bandits (MAB) [1], we know that if the paths are disjoint, then the optimal regret is bounded by $\Omega(M \log T)$. On the other hand, we also know that $\Omega(M \log T)$ cannot hold under all topologies because it contradicts the $O(N^4 \log T)$ upper bound since M can be exponential in N (note $M \leq N$ for disjoint paths). Intuitively, this is because when paths share common links, path qualities will be correlated, which allows for more efficient learning as samples from one path may contain information about other paths as well. We show below a regret lower bound for an arbitrary topology that scales *at most linearly* in N. We begin with the following definition.

Definition 3.2: Let $\mathcal{P}' \triangleq \{\mathbf{p} \in \mathcal{P} : \sum_i p_i l_i > \sum_i p_i^* l_i\}$ denote the set of suboptimal paths (between s and d). We say a link $i \in E$ is *competitive* if a suboptimal path $\mathbf{p} \in \mathcal{P}'$ containing link i can be made optimal by modifying the weight of link i, while keeping other link weights unchanged.

[1]The proof in [5] contains a technical glitch, the correction of which makes the constant term proportional to M and thus exponential in N in the worst case; this does not affect the $O(N^4 \log T)$ scaling.

Intuitively, competitive links are critical links where erroneous link weight estimates can lead to suboptimal routing decisions. Since link weights are random variables, we can quantify the error by the Kullback-Liebler (KL) distance. Let E' be the set of competitive links. For each competitive link $i \in E'$, there is a minimum error in the link distribution to make a suboptimal path optimal, defined as

$$D_i \triangleq \min\{D(L_i||L_i') : \exists \mathbf{p} \in \mathcal{P}' \text{ with } p_i = 1 \text{ s.t. } \mathbb{E}[W_{\mathbf{p}}'] \leq \mathbb{E}[W_{\mathbf{p}^*}]\}, \quad (2)$$

where $D(\cdot||\cdot)$ is the KL distance, and $W_{\mathbf{p}}'$ denotes the new weight of path $\mathbf{p}$ under a new link weight distribution L_i'.

The proof is based on a similar idea in [1]: if we modify the link weights so that a competitive link becomes part of the optimal path and thus must be used most of the time by any good algorithm, then this link must be used sufficiently often ($\log T/D_i$ out of T slots) under the original weights as this is the time required to distinguish between the two configurations. Since by definition, competitive links are always on suboptimal paths, this introduces a lower bound on the frequency of using suboptimal paths, and hence on regret. See [8] for a detailed proof.

Theorem 3.3: For mutually independent and i.i.d. link weights, the average regret of any $O(\log T)$-regret algorithm under coupled probing and routing satisfies

$$\mathbb{E}[\mathcal{R}_T] \geq \min_{\Phi} \sum_{\mathbf{p}\in\mathcal{P}'} \Delta_{\mathbf{p}} u_T(\mathbf{p}) \quad (3)$$

for all sufficiently large T, where $\Delta_{\mathbf{p}} \triangleq \mathbb{E}[W_{\mathbf{p}} - W_{\mathbf{p}^*}]$ is the suboptimality of path $\mathbf{p}$, and $u_T(\mathbf{p})$ the average number of times $\mathbf{p}$ is used up to time T, constrained by

$$\Phi \triangleq \{(u_T(\mathbf{p}))_{\mathbf{p}\in\mathcal{P}'} :\ u_T(\mathbf{p}) \geq 0,\ \forall \mathbf{p} \in \mathcal{P}';\ \sum_{\mathbf{p}\in\mathcal{P}'} p_i u_T(\mathbf{p}) \geq \frac{\log T}{D_i},\ \forall i \in E'\}. \quad (4)$$

Basically, the constraint (4) requires the usage of suboptimal paths to satisfy that each competitive link i is used at least $\log T/D_i$ times in T slots; the path usage that minimizes the performance gap wrt the optimal path establishes a lower bound on regret. This bound is topology-dependent since which links are on each path depends on the network topology.

The lower bound in (3) can be computed by solving the following linear programming problem. In vector form, with $\mathbf{\Delta} \triangleq (\Delta_{\mathbf{p}})_{\mathbf{p}\in\mathcal{P}'}$, $\mathbf{u}_T \triangleq (u_T(\mathbf{p}))_{\mathbf{p}\in\mathcal{P}'}$, $\mathbf{d} \triangleq (1/D_i)_{i\in E'}$, and $\mathbf{A}'$ denoting the sub-matrix of the routing matrix $\mathbf{A}$ consisting of paths in $\mathcal{P}'$, we can rewrite (3) as

$$\min \mathbf{\Delta}^T \mathbf{u}_T \quad \text{s.t. } (\mathbf{A}')^T \mathbf{u}_T \geq \mathbf{d}\log T,\ \mathbf{u}_T \geq \mathbf{0}. \quad (5)$$

Moreover, we can bound it in closed form as follows.

Corollary 3.4: The lower bound in (3) is $\Omega(\log T)$ and $O(\min(N,\ M)\log T)$. Specifically, for $D_{\min} \triangleq \min_{i\in E'} D_i$,

$$\frac{\Delta_{\min}}{D_{\min}}\log T \leq \min_{\Phi} \mathbf{\Delta}^T \mathbf{u}_T \leq \frac{\Delta_{\max}}{D_{\min}}\min(M,\ N)\log T.$$

Algorithm 2 Online SPR (OSPR)

Require: Candidate paths $\mathcal{P}$.

Ensure: Select a path to route and a path to probe at each time slot, and compute estimated mean link weights $(\hat{l}_i)_{i=1}^N$.

Initialization: for $t = 1, \ldots, N$,

1: Route over a randomly selected path

2: Probe paths $\mathbf{p}(1), \ldots, \mathbf{p}(N)$ s.t. $p_i(i) = 1$ to measure each link at least once

3: Compute $(\hat{l}_i,\ m_i)_{i=1}^N$ accordingly

4: **for** $t = N+1, \ldots$ **do**

5: Route over path $\hat{\mathbf{p}}(t) = \arg\min_{\mathbf{p}\in\mathcal{P}} \sum_i p_i \hat{l}_i$

6: Probe the path $\mathbf{p}(t)$ containing the least measured link, i.e., link j s.t. $m_j = \min_i m_i$

7: Update $(\hat{l}_i,\ m_i)_{i=1}^N$ accordingly

Remark: Theorem 3.3 provides a consistent lower bound for all topologies. Besides reducing to $\Omega(M \log T)$ for disjoint paths, this bound also applies to paths with arbitrary overlaps, where we have shown that the scaling is at most $O(N \log T)$ even if $M \gg N$. Compared with the upper bound in Theorem 3.1, we see a gap of N^3 to N^4. The lower bound is known to be tight for disjoint paths from the MAB theory [1], i.e., the upper bound is loose by a factor of N^4/M in this case.

B. Decoupled Probing and Routing

When routing and probing are decoupled, we can optimize the two procedures separately. It turns out that the decoupling dramatically impacts the performance, reducing the regret from $\Omega(\log T)$ under coupled probing/routing to a value constant in T, as shown below.

1) Algorithm and Regret Upper Bound: Consider the algorithm shown in Algorithm 2, which always probes the path with the least observed link (line 6) to obtain uniformly good estimates of link qualities. The path with the minimum empirical mean weight is then selected for routing (line 5). As in LLC, the optimization in line 5 can be solved by Dijkstra's algorithm, and thus OSPR can be implemented efficiently even if the number of candidate paths is exponentially large.

Despite its simplicity, OSPR is guaranteed to achieve a regret constant in T and polynomial in N as stated below (the bound holds even if M is exponential in N).

Theorem 3.5: The average regret of OSPR is $O(N^4)$ and specifically,

$$\mathbb{E}[\mathcal{R}_T] \le \Delta_{\max} N \left(\frac{2c}{\Delta_{\min}^2} H^2 N + \frac{4}{\Delta_{\min}^2} H^3 + 1 \right), \qquad (6)$$

where c is a constant satisfying $c \ge \log M/N$, and $\Delta_{\min}$, $\Delta_{\max}$, and H are defined in Theorem 3.1.

We give a sketch of proof below and refer to [8] for details. Consider the event $\mathcal{A}_{\mathbf{p},t}$ of mistakenly routing over a suboptimal path $\mathbf{p}$ in slot t. We show that its probability decays exponentially wrt the minimum link sample size, using known results on the convergence of sample mean to true mean. Since OSPR guarantees that the minimum link sample size will grow linearly with time, $\Pr\{\mathcal{A}_{\mathbf{p},t}\}$ also decays exponentially with t. This result already implies a constant regret wrt T because the average number of times of using a suboptimal path $\mathbf{p}$, given by $\sum_{t=1}^T \Pr\{\mathcal{A}_{\mathbf{p},t}\}$, is finite as $T \to \infty$. This argument alone is not sufficient to show the polynomial scaling wrt N, as the number of suboptimal paths can be exponential in N. To this end, we divide time into a training period and a routing period, where we show that a careful choice of training period length gives $O(N^4)$ overall regret.

Remark: Compared with the $O(N^4 \log T)$ regret in Theorem 3.1, we get a dramatic $\log T$-factor improvement in regret by decoupling probing and routing. Intuitively, this is because once probing and routing are decoupled, we no longer have to sacrifice learning accuracy for routing performance, which allows us to leverage the exponential convergence in link estimation to quickly lock onto the best path and achieve a constant regret (in time). Moreover, the lower bound in Corollary 3.4 shows that the $\log T$ factor is in fact the fundamental gap between coupled and decoupled probing/routing rather than artifacts of the considered algorithms. Note that a constant average regret does not mean that we can learn the optimal path without error in constant time; in fact, any algorithm with a finite learning time will incur $\Theta(T)$ regret.

2) Regret Lower Bound: The lower bound under decoupled probing/routing is a direct implication of *large deviation theory* [9]. Consider the case of mutually independent and temporally i.i.d. link weights. Let $\mathbf{L} = (L_i)_{i=1}^N$ be the random vector denoting the link weight distributions. Define

$$\delta_{\min} \triangleq \min_{\mathbf{p}\in\mathcal{P}'} \min_{\mathbf{L}':\sum_i p_i \mathbb{E}[L_i'] < \sum_i p_i^* \mathbb{E}[L_i']} D(\mathbf{L}'||\mathbf{L}) \qquad (7)$$

as the minimum change in link weights (measured by KL distance) needed to change the optimal path.

Theorem 3.6: For mutually independent and temporally i.i.d. link weights, the average regret of any constant-regret (in T) algorithm under decoupled probing and routing satisfies

$$\mathbb{E}[\mathcal{R}_T] \ge \frac{\Delta_{\min}(1 - e^{-\delta_{\min} T})}{1 - e^{-\delta_{\min}}} \qquad (8)$$

for all sufficiently large T.

Remark: Theorem 3.6 gives an $O(1)$ lower bound for decoupled probing and routing. Compared with the $O(N^4)$ upper bound achieved by OSPR, it shows an N^4-factor gap. This gap, however, is mostly due to the looseness of the upper bound, which has to handle general topologies, and it can be shown that the lower bound is tight in special cases.

IV. Accelerated Learning

We have studied the basic case of a single source with single-path probing. In this section, we extend the proposed solution to more general scenarios involving concurrent probes and cooperative learning between multiple sources, with focus on the impact of these generalizations on the learning performance. We only consider decoupled probing/routing in the sequel because of its efficiency.

A. Multi-Path Probing

Naturally, if a source can probe multiple paths simultaneously, convergence will be faster, and regret will be smaller. We characterize the exact improvement by extending our analysis to this case. Specifically, let S denote the number of simultaneous probes. Assume that link weights measured by different probes within the same time slot also satisfy the assumption in Section II-A. We can still apply Algorithm 2 for probing and routing, except that the probing step (line 6) is repeated S times to select the S probing paths. We allow these paths to contain common links. Note that the initialization phase will also take a shorter time of at most $\lceil N/S \rceil$ slots.

For this modified algorithm, we show that there is a $1/S$-factor decrease in the regret, as proved in [8].

Corollary 4.1: The average regret of OSPR with S-path probing is $O(N^4/S)$ and specifically,

$$\mathbb{E}[\mathcal{R}_T] \leq \frac{\Delta_{\max} N}{S}\left(\frac{2cNH^2}{\Delta_{\min}^2}+\frac{4H^3}{\Delta_{\min}^2}+1\right), \tag{9}$$

assuming $c \geq \log M/N$.

This result shows that as the probing rate S increases, the regret decreases inversely proportionally.

B. Multi-Source Learning

So far we have focused on a single source-destination pair. The presence of multiple source-destination pairs may accelerate learning if sources can share information. Depending on the extent of information sharing, we divide the problem into two cases: *coordinated probing* and *uncoordinated probing*. Both cases assume that sources share link estimates; the difference is in whether the selection of probing paths is centralized (coordinated probing) or distributed (uncoordinated probing). In the rest of this section, assume there are S source-destination pairs probing a total of S paths per slot (our solutions are easily extendable to other probing rates). To distinguish between different source-destination pairs, we will use subscript f ($f = 1, \ldots, S$) to label the parameters for source-destination pair f.

1) Coordinated Probing: Under coordinated probing, probing paths are selected globally from the candidate paths of all the source-destination pairs $\mathcal{P} \triangleq \bigcup_{f=1}^{S} \mathcal{P}_f$, either by a cluster head or by individual sources based on shared link sample sizes, so that each probing path covers at least one globally least measured link. At each slot t, S probing paths are sequentially chosen to satisfy an (average) probing rate of one path per source-destination pair. The link estimates are then shared among all the sources, based on which each source routes over the empirically shortest path among its candidate paths. We refer to this algorithm as *Online SPR under Coordinated Probing (OSPR-CP)*; see Algorithm 3 in [8] for the pseudo code. Note that a source may probe more than one path or no path during a slot.

Coordinated probing basically increases the probing rate from one path to S paths at a time, which is identical to multi-path probing except that the candidate paths are extended from one's own paths to the paths of all the source-destination pairs. Consequently, Corollary 4.1 still applies, i.e., the average regret for source-destination pair f under coordinated probing is $O(N^4/S)$, and specifically,

$$\mathbb{E}[\mathcal{R}_{T,f}] \leq \frac{\Delta_{\max,f} N}{S}\left(\frac{2c_f NH_f^2}{\Delta_{\min,f}^2}+\frac{4H_f^3}{\Delta_{\min,f}^2}+1\right), \tag{10}$$

where $c_f \geq \log M_f/N$.

2) Uncoordinated Probing: Under uncoordinated probing, every source performs probing and routing in a distributed manner using OSPR (Algorithm 2), except that the routing/probing is based on the shared measurement information of all the sources. Specifically, if $\hat{l}_{i,f}$ denotes the empirical mean for link i based only on measurements taken by source f, and $m_{i,f}$ the corresponding sample size. Then after receiving $(\hat{l}_{i,s},\ m_{i,s})_{i=1}^{N}$ from the other sources $s \in \{1,\ldots,S\} \setminus f$, source f can compute the aggregate estimate for link i by $\hat{l}_i = (\sum_{s=1}^{S} \hat{l}_{i,s} m_{i,s})/(\sum_{s=1}^{S} m_{i,s})$, and the total sample size by $m_i = \sum_{s=1}^{S} m_{i,s}$. It then selects the routing path as the shortest path based on $\hat{l}_i$, and the probing path as the path covering the link with the minimum m_i. We refer to this algorithm as *Online SPR under Uncoordinated Probing (OSPR-UP)*; see Algorithm 4 in [8] for the pseudo code.

The difference from the coordinated case is that since sources in uncoordinated probing select probing paths distributedly, links are no longer evenly measured since some links may be covered by more source-destination pairs than others. Let C_{if} ($i = 1, \ldots, N$, $f = 1, \ldots, S$) be the indicator that link i is covered by the paths of source-destination pair f, $E_f \triangleq \{i \in E : C_{if} = 1\}$ the subset of links covered by f, and $N_f \triangleq \sum_{i=1}^{N} C_{if}$ the size of this subset. We show that the regret of OSPR-UP is bounded as follows (see proof in [8]).

Corollary 4.2: The average regret of OSPR-UP for source-destination pair f is bounded as

$$\mathbb{E}[\mathcal{R}_{T,f}] \leq \max_{i \in E_f} \frac{\Delta_{\max,f} N_f \left(c_f N_f + \frac{\Delta_{\min,f}^2}{2H_f^2}\sum_s C_{is}\right)}{\frac{N_f \Delta_{\min,f}^2}{2H_f^2}\sum_s \frac{C_{is}}{N_s}} + \frac{4\Delta_{\max,f} H_f^3}{\Delta_{\min,f}^2 \min_{i \in E_f}\sum_s \frac{C_{is}}{N_s}} \tag{11}$$

for $c_f \geq \log M_f/N_f$.

Corollary 4.2 provides a unified bound for general topologies and source-destination placements. Intuitively, its difference from coordinated probing depends on the overlap between different source-destination pairs. Under complete overlap, i.e., $C_{if} \equiv 1$ and $N_f \equiv N$, the bound in (11) is reduced to $\frac{\Delta_{\max,f} N}{S}\left(\frac{2c_f NH_f^2}{\Delta_{\min,f}^2}+\frac{4H_f^3}{\Delta_{\min,f}^2}+S\right)$, which is only slightly worse than the coordinated case (10). In the absence of any overlap, i.e., $\sum_{f=1}^{S} C_{if} \equiv 1$, (11) reduces to the single source-destination case (6) on the subnetwork occupied by source-destination pair f.

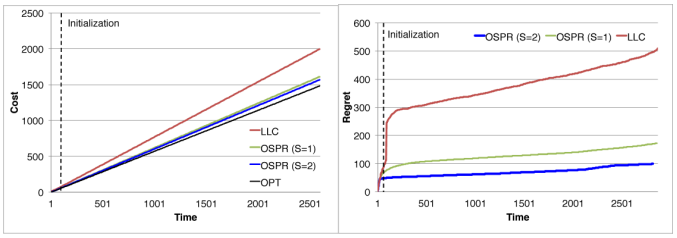
V. Performance Evaluation

A. Evaluation Setup

Network traces: We evaluate the performance of our online shortest path algorithm using network traces from a mesh network deployed on a university campus. The traces contain detailed link quality information collected over several days from the UCSB MeshNet [10]. The UCSB MeshNet is a multi-radio 802.11a/b/g network consisting of 20 PC-nodes deployed indoors on five floors of a typical office building on the UCSB campus. Link quality is measured by the Expected Transmission Time (ETT), which is the estimated time to transmit a packet. ETT is calculated from a link's loss rate and data rate, given by $(\text{packet size})/\ (d_1 \cdot d_2 \cdot b_w)$, where d_1 and d_2 are the delivery ratios in the forward/backward direction, and b_w is the average data rate (bytes/second); packet size is assumed to be 1500 bytes. We use measured ETTs from a 2500-minute period as link weights, with 1-minute slots. Further details about the network topology, link correlations, and long term link behavior can be found in the analysis conducted by Ramachandran et al [11]. Details on the source-destination pairs and number of routes that we consider are specified in each experiment setting below. Note that the traces are only used to simulate realistic link dynamics; the testbed was not routing traffic when the traces were collected, and hence the original routing protocol running on the testbed cannot be used for comparison.

Methodology: We compare the performance of the proposed OSPR algorithm (Algorithm 2) with the existing LLC algorithm (Algorithm 1) for the case of single-source learning discussed in Section III, and then evaluate the extensions of OSPR for the case of multi-source learning discussed in Section IV-B for both coordinated and uncoordinated probing. To remind the reader, both coordinated and uncoordinated cases assume that sources share link estimates; the difference is in whether probing is performed in a centralized (coordinated probing) or distributed (uncoordinated probing) manner.

From the traces, we have the instantaneous link weight (ETT) for each link. The path weight is given by the sum of the link weights of all links on the path. Since we are performing trace-based simulations, the result of selecting a path to "route" is to incur a cost given by the path weight. When we probe a path, we update $\hat{l}_i$ for each link i on the path based on the recorded ETT in the trace and increment m_i. We evaluate the algorithms based on two metrics: *cost* and *regret*. Cost is the absolute cost incurred by the routing algorithm, and is computed as the cumulative sum of the ETTs on routing paths for the duration of the simulation. Regret is the extra cost (in terms of ETT) incurred by an online learning algorithm compared to an oracle that uses the shortest path based on the true average ETTs of the trace, accumulated over time, and is computed by (1). In the evaluation results, cost and regret are measured in the same unit as ETT (second), while simulation time is measured in the same unit as slot (minute).



(a) Routing cost (b) Regret

Fig. 2. Single-source learning: cost and regret of LLC and OSPR.

B. Evaluation Results

Figure 2 shows simulation results of single-source learning algorithm computed from a single source and averaged over all possible destinations. Figure 2(a) shows the accumulated cost (in terms of delay) and Figure 2(b) shows the accumulated regret for the length of the simulation. We evaluate the proposed OSPR algorithm (Algorithm 2) and compare its performance with the existing LLC algorithm (Algorithm 1). We evaluate the impact of single-path probing (OSPR, $S = 1$) and multi-path probing (OSPR, $S = 2$). OPT shows performance of the optimal routing algorithm, i.e. the algorithm that picks the shortest path routes computed using the mean ETTs from the traces. The initialization period, during which time each link is measured at least once, is shown by the dotted line in each graph. We observe that LLC accrues significantly higher cost than OSPR ($> 30\%$) and more than twice as much regret. OSPR quickly discovers near-optimal paths using decoupled probing and routing. As expected, multi-path OSPR works better than the single-path version, although this comes with additional probing overhead.

Figure 3 shows simulation results of multi-source learning algorithms under coordinated probing (OSPR-CP) and uncoordinated probing (OSPR-UP). We first evaluate each algorithm by the average cost/regret per pair for two ($S = 2$) randomly selected source-destination pairs, and then repeat the experiment with four ($S = 4$) randomly selected pairs to test the effect of increasing the number of probing sources. Figure 3(a) shows the average cost. As expected, the average cost decreases as the number of sources increases, since the total probing rate increases. Moreover, as predicted by the analysis in Section IV-B, coordinated probing incurs lower cost than uncoordinated probing. We have verified that both OSPR-CP and OSPR-UP incur lower cost than the single-source, single-path learning algorithm (OSPR ($S = 1$)) shown in Figure 2(a). Figure 3(b) shows the average regret from the same experiment. Among the simulated schemes, we see a modest decrease of 5%-10% in regret when the number of sources is increased from 2 to 4. A much larger improvement is yielded by adopting coordinated probing (OSPR-CP), which achieves up to 25% lower regret than uncoordinated probing (OSPR-UP), although both have the same total probing rate. The real benefit of coordination is shown in Figure 3(c), which compares the gap between the maximum and the minimum regrets of all source-destination pairs from the same experiment. The coordinated probing scheme selects probing paths globally

from the candidate paths of all the source-destination pairs so that each probing path covers at least one globally least measured link. As a result, while the uncoordinated sources exhibit significant variations, the coordinated sources display much less variation (about 4-5 times lower).

VI. Handling Practical Challenges

While we have made several simplifying assumptions in previous analysis, our main result holds under broader scenarios where these assumptions are significantly relaxed.

A. Temporally-Correlated Link Weights

As mentioned in Section II-A, the achievability results actually apply to a broader class of link weight processes than i.i.d. processes. First, the proofs in [8] on the achievability results only require a weaker assumption that each of the link weight processes $(L_i(t))_{t=1}^{\infty}$ has a fixed conditional expectation, i.e., $\mathbb{E}[L_i(t)|L_i(1),\ldots,L_i(t-1)] \equiv l_i$ for all $t \geq 1$. This assumption is a relaxation to the i.i.d. assumption because although it trivially holds for i.i.d. processes, it also allows certain temporal dependency, e.g., Markovian link weights with the same conditional mean for each initial weight.

In general, it is possible to achieve the same regret as before under temporally correlated link weights, as long as the sampling of each link can still provide an estimate $\hat{l}_i$ that converges exponentially fast to l_i. One such example is *K-dependent processes*, where $L_i(t)$ is independent of $L_i(s)$ for $|s-t| > K$, but can be arbitrarily correlated for $|s-t| \leq K$; assume $\mathbb{E}[L_i(t)] \equiv l_i$ for all t. OSPR can still be employed, with a slight modification: after a link is probed, it will be deactivated for K slots (assuming K is known) so that it will not be probed again until the obtained sample becomes independent of the previous samples[2]. We have shown that Theorem 3.5 still holds as long as $K < N$; see Corollary 2.7 in [8].

B. Delayed Probes

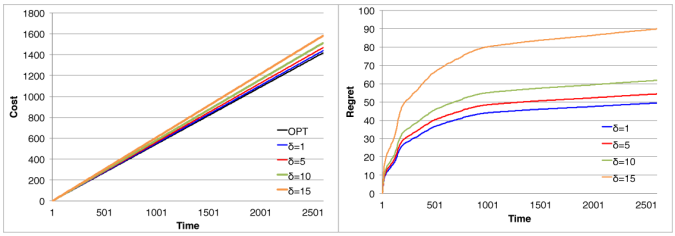
We have assumed that probes return within one slot. In practice, a probe may take multiple slots to complete, and the delay may grow with the network size. However, we will show that such delays do not change the scaling of regret. Specifically, assume probes are delayed by at most $(\beta-1)N$ slots for some constant $\beta \geq 1$, i.e., we only have measurements from slots $1,\ldots,t-(\beta-1)N-1$ at the beginning of slot t. We show that this delay only affects a lower-order term in the regret upper bound (6), and the order of regret remains the same (see proof in [8]).

Corollary 6.1: If probes are delayed by at most $(\beta-1)N$ slots, the average regret of OSPR is $O(N^4)$ and specifically,

$$\mathbb{E}[\mathcal{R}_T] \leq \Delta_{\max} N \left(\frac{2c}{\Delta_{\min}^2} H^2 N + \frac{4}{\Delta_{\min}^2} H^3 + \beta \right), \qquad (12)$$

where the quantities are defined the same as in (6).

Using the traces in Section V, we evaluate the effect of delayed probing on the cost and the regret of OSPR. We show that delayed probes do not result in significant increase in cost and regret, especially when the delays are not very large. In the absence of probing delay, at time slot t, probing responses up to slot $t-1$ are available. A $\delta \triangleq (\beta-1)N$ value of 1 slot implies that at slot t, only probes up to slot $t-2$ are available. We evaluate for δ values of 1, 5, 10 and 15 slots. The evaluation results for a random selected source-destination pair are shown in Figure 4; similar results hold for other source-destination pairs. It can be seen that the cost and hence the regret increase as the delay increases. However, the increases in cost and regret are not linear with the increase in delay. This result is explained by the temporal correlation in the link quality. With smaller delays (e.g., $\delta = 1$), we find that the responses from delayed probes are more strongly correlated with the current link quality, resulting in smaller regret. When the delays are large, the responses have poorer correlation with the current link quality, thus resulting in larger regret.



(a) Cost (b) Regret

Fig. 4. Delayed probing. Cost and Regret for varying $\delta = (\beta-1)N$

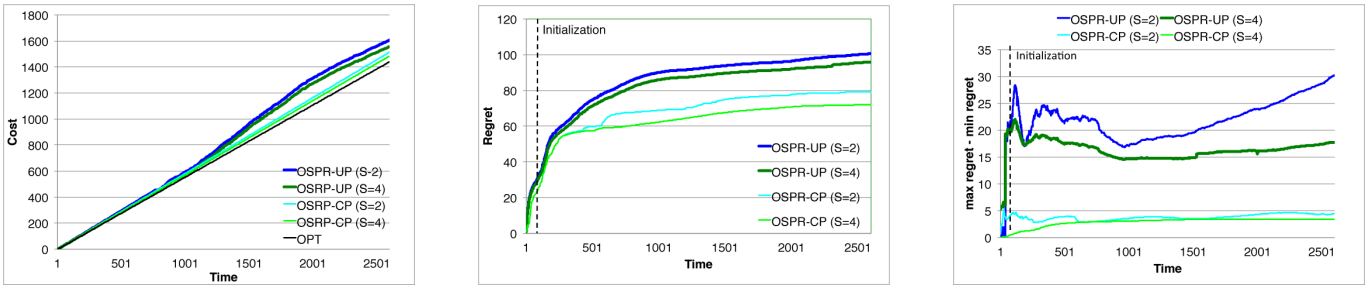
C. Changes in Link Qualities

A major assumption in our analysis is that the link quality, measured by the mean link weight, is fixed. In practice, environmental changes such as the appearance/disappearance of external interferers may cause persistent changes in link qualities. We first note that the proposed solution already adapts to changes to some extent, although its adaptivity is limited by the use of cumulative empirical means as link estimates. To improve adaptivity, one approach is to introduce discounts to past measurements so that the impact of outdated measurements diminishes with time. Another approach that allows more explicit control on the tradeoff between steady-state performance and adaptivity is to use a change detector, such that once a change occurs, the learning algorithm can quickly detect it and restart learning.

As a concrete example, consider a window-based change detector as follows. Define a *control window* as a time interval starting from the last detection of change, and a *sliding window* as a time interval ending at the current time, each containing ω measurements ($\omega \geq 1$). Let $\hat{l}_{i,c}$ (or $\hat{l}_{i,s}$) be the empirical mean weight of link i in the control (or sliding) window. Then given a parameter $\alpha \in (0,\ 1)$, the detector declares change if and only if $|\hat{l}_{i,c} - \hat{l}_{i,s}| \geq \sqrt{\frac{2}{\omega}\log\left(\frac{2}{\alpha}\right)}$ (see Algorithm 5 in [8]). We show that this detector has a false alarm probability bounded by α; see Proposition 2.9 in [8].

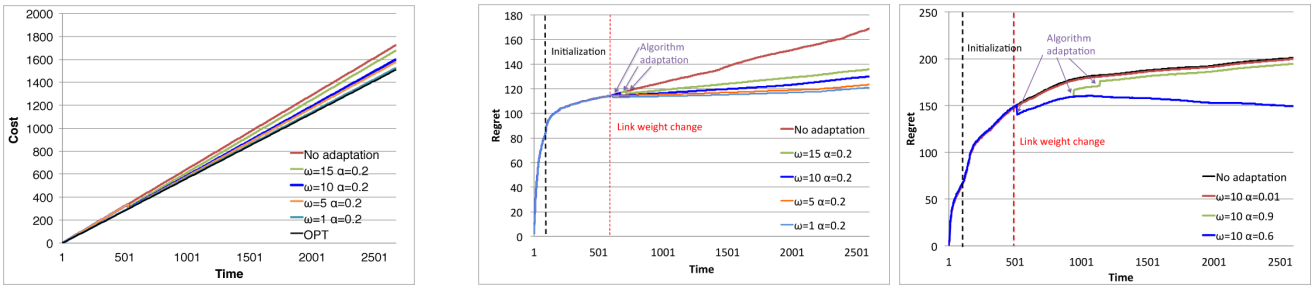
Using the traces in Section V, we evaluate the above detector in combination with OSPR. We show that: (a) the proposed algorithm is capable of adapting to changes in

[2]We allow a deactivated link to be on probing paths, but the measurement for the link will not be used.



(a) Routing cost (b) Regret (c) Variation in regret

Fig. 3. Multi-source learning: regret and its variation for OSPR-UP and OSPR-CP.



(a) Cost with fixed α (b) Regret with fixed α (a) Regret with fixed ω

Fig. 5. Change-adaptive learning.

mean link weights and reducing the overall regret, and (b) successful adaptation relies on proper setting of the parameters ω (window size) and α (false alarm). Figures 5(a) and 5(b) show the cost and regret respectively of varying ω under a fixed α for a selected source-destination pair. We observe that the cost decreases as ω is decreased. As indicated by arrows in Figure 5(b), the algorithm adapts at different times under different ω values, with $\omega = 1$ giving the lowest regret. This is because the selected pair experiences a large jump in link qualities at the red marker, and a smaller ω results in faster adaptation. However, very small window sizes could result in unnecessary adaptations due to short-term fluctuations. Similarly, Figure 5(c) shows the results of varying α under a fixed ω for a (different) source-destination pair with a marginal change in link qualities at the red marker. We see a clear tradeoff in setting α: too small a value ($\alpha = 0.01$) will let the change go unnoticed, reduced to the case of no adaptation; too large a value ($\alpha = 0.9$) will make the algorithm adapt too often, failing to converge to the optimal path. With a proper value ($\alpha = 0.6$), we are able to detect real changes without over-adapting, thus lowering the overall regret.

VII. Conclusion

We have studied the problem of bootstrapping shortest path routing via online learning at communication endhosts. We show that the decoupling of probing and routing is a critical condition for constant-regret learning. Building upon a well-analyzed baseline solution, we construct solutions for a general set of scenarios including multi-path and multi-source probing. Although our analysis is based on certain assumptions on link dynamics, our solution performs well in simulations driven by real traces, and our results remain largely valid when these assumptions are relaxed.

References

[1] V. Anantharam, P. Varaiya, and J. Walrand, "Asymptotically Efficient Allocation Rules for the Multiarmed Bandit Problem with Multiple Plays—Part I: I.I.D. Rewards," *IEEE Trans. Automatic Control*, 1987.

[2] A. Bhorkar, M. Naghshvar, T. Javidi, and B. Rao, "Adaptive Opportunistic Routing for Wireless Ad Hoc Networks," *IEEE Trans. Networking*, 2011.

[3] A. Bhorkar and T. Javidi, "No Regret Routing for Ad-Hoc Wireless Networks," in *Asilomar Conference*, 2010.

[4] A. György, T. Linder, G. Lugosi, and G. Ottucsák, "The On-Line Shortest Path Problem Under Partial Monitoring," *Journal of Machine Learning Research*, 2007.

[5] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial Network Optimization with Unknown Variables: Multi-Armed Bandits with Linear Rewards and Individual Observations," *IEEE/ACM Trans. Networking*, 2012.

[6] V. Dani, T. P. Hayes, and S. M. Kakade, "Stochastic Linear Optimization under Bandit Feedback," in *Annual Conference on Learning Theory*, 2008.

[7] K. Liu and Q. Zhao, "Online Learning for Stochastic Linear Optimization Problems," in *Information Thoery and Applications Workshop*, 2012.

[8] T. He, D. Goeckel, R. Raghavendra, and D. Towsley, "Endhost-Based Shortest Path Routing in Dynamic Networks: Supporting Materials," IBM, Tech. Rep., 2012, http://researcher.ibm.com/files/us-the/rc071201.pdf.

[9] T. Cover and J. Thomas, *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[10] I. Sheriff, K. N. Ramachandran, K. C. Almeroth, and E. Belding, "CRAWDAD trace ucsb/meshnet/neighbortable/20060406 (v. 2007-02-01)," http://crawdad.cs.dartmouth.edu/ucsb/meshnet/neighbortable/20060406.

[11] K. Ramachandran, I. Sheriff, E. Belding, and K. Almeroth, "Routing stability in static wireless mesh networks," in *Proceedings of the 8th international conference on Passive and active network measurement*, ser. PAM'07, 2007.