Online Shortest Path Routing: 
The Value of Information

Zhenhua Zou, Alexandre Proutiere and Mikael Johansson
KTH-Royal Institute of Technology, Sweden

Abstract—This paper studies online shortest path routing over dynamic multi-hop networks. Link costs or delays are time-varying and modelled by independent and identically distributed random processes, whose parameters are initially unknown. The parameters, and hence the optimal path, can only be estimated by routing packets through the network and observing the realized delays. Our aim is to find a routing policy that minimizes the regret (the cumulative delay difference) between the path chosen by the policy and the unknown optimal path. We formulate the problem as a combinatorial bandit optimization problem and consider several scenarios that differ in where routing decisions are made and in the information available when making the decision. For each scenario, we derive the tight asymptotic lower bound on the regret that has to be satisfied by any online routing policy. These bounds help us to understand the performance improvements we can expect when (i) taking routing decisions at each hop rather than at the source only, and (ii) observing per-link costs rather than aggregate path costs. In particular, we show that (i) is of no use while (ii) can have a spectacular impact. Efficient algorithms are proposed and evaluated against the state-of-the art.

I. INTRODUCTION

Most real-world networks are dynamic and evolve over time. Packet losses in wireless sensor networks occur randomly and the average loss rates on links may change over time. Nodes in mobile ad-hoc networks are constantly moving which affects the inter-node distances and thus the link parameters (e.g. the success transmission probability or average delay). The delays in overlay networks used in peer-to-peer applications change unpredictably as the load in the underlay network fluctuates. In many cases, the link parameters are initially unknown and must be estimated by transmitting packets and observing the outcome. This leads to a challenging trade-off between exploration and exploitation. On the one hand, it is important to route packets on new or poorly known links to explore the network and ensure that the optimal path is eventually found. On the other hand, the accumulated knowledge must be exploited so that paths with low expected delays are preferred. Of course, when the link parameters evolve over time, it becomes crucial to design algorithms that quickly learn link parameters so as to efficiently track the optimal path.

The design of such algorithms is often referred to as an online shortest-path routing problem in the literature, and actually corresponds to a combinatorial bandit optimization problem [1]. In this paper, we study the stochastic version of this problem, where the cost or delay experienced on each link is an i.i.d. process with unknown average (for example, we can assume that the success and failure of packet trans-missions over a link are i.i.d. with unknown average success rate). We address two fundamental questions: (i) what is the benefit of allowing routing decisions at every node, rather than only at the source; and (ii) what is the added value of feeding back the observed delay for every link that a packet has traversed compared to only observing the end-to-end delay. To this end, we consider several scenarios that differ in where routing decisions are made and what information is available to the decision-maker. The performance of a routing policy is assessed using its regret up to packet $N$ (where $N$ will be typically large), defined as the difference of the total expected end-to-end delays for the $N$ first packets under this policy and under an oracle policy that always sends packets on the best path.

We make the following contributions:

- For each scenario, we derive a tight asymptotic lower bound on the regret. No online routing policy (in the given class) can obtain a lower regret than this and there exists an algorithm that attains the bound.
- Using the regret bounds, we demonstrate that the added flexibility of hop-by-hop routing does not affect the achievable asymptotic regret, while the impact of per-link feedback can be spectacular.
- We propose simple and efficient online routing algorithms based on per-link feedback and evaluate their finite-time regret in simulations. The numerical experiments indicate that our routing policies performs significantly better than state-of-the art algorithms.

A. Related Work

The online shortest path problem has been extensively studied in an adversarial setting where link delays can change arbitrarily [1], [2]. However, only a handful recent papers address the problem with stochastic random delays [3]–[5]. Note that the problem differs from the classical stochastic multi-armed bandit problem solved by Lai and Robbins [6], because the delays observed on two paths with common links are inherently correlated. Gai et.al [3] consider source routing with feedback on the delay of each link of the path along which the packet is forwarded. After each successful end-to-end transmission, a UCB-like [7] index is assigned to each link and the next packet is routed on the shortest path with respect to these indexes. Liu and Zhao [4] consider the case where only the aggregated end-to-end delay of the path is available and apply the idea of barycentric spanner from [8]. A barycentric spanner is a set of paths such that the delay of all other paths can be written as a linear combination
of the delays in the barycentric spanner. The algorithm is composed of an exploration phase where barycentric spanner paths are chosen in a round-robin fashion and an exploitation phase where the empirical minimum delay path is used. Both schemes are shown to have a regret of order \( O(|E|^4 \log n) \) where \(|E|\) is the total number of links on the network. However, neither work provides an asymptotic lower bound on the regret. The first attempt to address this issue appears in He et al. [5], which gives an asymmetric lower bound on the regret (for policies with detailed feedback) scaling as \( O(|E| \log n) \). We show that this bound is not tight, and we obtain (for the first time) a tight regret lower bound. Lastly, all work cited above consider source routing. To the best of our knowledge, this is the first work to consider hop-by-hop routing decisions.

### B. Organization

The paper is organized as follows. In Section II we introduce the network model and formulate our online routing problem. Fundamental performance limits (regret lower bounds) are derived in Section III. Finally, we propose two online routing algorithms and evaluate their performance numerically in Section IV. The detailed proofs are presented in the Appendix.

### II. Online Shortest Path Routing Problems

#### A. Network Model

The network is modelled as a directed graph \( G = (V, E) \) where \( V \) is the set of nodes and \( E \) is the set of links. Each link \( i \in E \) may, for example, represent an unreliable wireless link. Without loss of generality, we assume that time is slotted and that one slot corresponds to the time to send a packet over a single link. At time \( t \), \( X_i(t) \) is a random binary variable indicating whether a transmission on link \( i \) at time \( t \) is (or would be) successful. \( \{X_i(t), t \geq 1\} \) is a sequence of i.i.d. Bernoulli variables with initially unknown mean \( \theta_i \). We assume \( \theta_i > 0 \). Let \( \theta = (\theta_i, i \in E) \) be the vector representing the packet successful transmission probabilities on the various links. We consider a single source-destination pair \((s, d) \in V^2\), and denote by \( \mathcal{P} \) the set of loop-free paths from \( s \) to \( d \) in \( G \).

#### B. Online Routing Policies and Feedback

The source is fully backlogged, and initially, the parameter \( \theta \) is unknown. Packets are sent successively from \( s \) to \( d \) over various paths to estimate \( \theta \), and in turn to learn the path \( p^* \) with the minimum average delay: \( p^* = \arg \min_{p \in \mathcal{P}} \sum_{i \in p} \frac{1}{\theta_i} \). After a packet is sent, we assume that the source gathers some feedback from the network (essentially per-link or end-to-end delays) before sending the next packet. Our objective is to design and analyze online routing strategies, i.e., strategies that take routing decisions based on the feedback received for the packets previously sent.

We consider and compare three different types of online routing policies, depending (i) on where routing decisions are taken (at the source or at each node), and (ii) on the received feedback (per-link or end-to-end path delay).

- **Source routing with aggregate feedback.** The path used by a packet is determined at the source based on the observed end-to-end delays for previous packets. More precisely, for the \( n \)-th packet, let \( p^n(n) \) be the path selected under policy \( \pi \), and let \( D^\pi(n) \) denote the corresponding end-to-end delay. Then \( p^n(k) \) depends on \( p^n(1), \ldots, p^n(n-1), D^\pi(1), \ldots, D^\pi(n-1) \). We denote by \( \Pi_1 \) the set of such policies.

- **Source routing with detailed feedback.** The path used by a packet is determined at the source based on the observed per-link delays for previous packets. In other words, under policy \( \pi \), \( p^n(i) \) depends on \( p^n(1), \ldots, p^n(n-1) \). \( (d_i^\pi(1), i \in p^n(1)) \), \ldots, \( (d_i^\pi(n-1), i \in p^n(n-1)) \), where \( d_i^\pi(k) \) is the experienced delay on link \( i \) for the \( k \)-th packet (if this packet uses link \( i \) at all). We denote by \( \Pi_2 \) the set of such policies.

- **Hop-by-Hop routing.** Routing decisions are taken at each node in an adaptive manner. At a given time \( t \), the packet is sent over a link selected depending on all successes and failures observed on the various links before time \( t \). Let \( \Pi_3 \) denote the set of hop-by-hop routing policies.

In the case of source-routing policies (in \( \Pi_1 \cup \Pi_2 \)), if a transmission on a given link fails, the packet is retransmitted on the same link until it is successfully received (per-link delays are geometric random variables). On the contrary, in the case of hop-by-hop routing policies (in \( \Pi_3 \)), the routing decisions at a given node can be adapted to the observed failures on a given link. For example, if transmission attempts on a given link failed, one may well decide to switch link, and select a different next hop node (and so, a different path).

#### C. Performance Metrics and Objectives

1) **Regret:** Under any reasonably smart routing policy, after sending a large number of packets, the parameter \( \theta \) will be estimated accurately, and the minimum delay path will be discovered with high probability. Hence, to quantify the performance of a routing policy, we examine its transient behavior. More precisely, we use the notion of regret, a performance metric often used in online stochastic optimization, and in multi-armed bandit literature [6]. The regret \( R^\pi_N \) of policy \( \pi \) up to the \( N \)-th packet is the expected difference of delays for the \( N \) first packets under \( \pi \) and under the policy that always select the best path \( p^* \) for transmission:

\[
R^\pi_N := \mathbb{E}_\pi \left\{ \sum_{n=1}^{N} D^\pi(n) \right\} - ND_\theta(p^*),
\]

where \( D_\theta(p) = \sum_{i \in p} \frac{1}{\theta_i} \) is the average packet delay through path \( p \) given link success rates \( \theta \), and \( D^\pi(n) \) denotes the end-to-end delay of the \( n \)-th packet under \( \pi \). The regret quantifies the performance loss due to the need to explore sub-optimal paths to learn the minimum delay path.

2) **Objectives:** The goal is to design online routing policies in \( \Pi_1 \), \( \Pi_2 \), and \( \Pi_3 \) that minimize regret over the \( N \) first packets. This can be formulated as an online stochastic
optimization problem (see [9] for an introduction). In case of source routing, this problem is often referred to as a combinatorial bandit problem [1], [2].

As it turns out, there are policies in any $\Pi_j$, $j = 1, 2, 3$, whose regrets scale as $O(\log(N))$ when $N$ grows large. Moreover we establish that no policy can have a regret scaling as $o(\log(N))$. Our objective is then to identify, for each $j = 1, 2, 3$, the best policy in $\Pi_j$ and its asymptotic regret $c_j(\theta) \log(N)$. By comparing $c_1(\theta)$, $c_2(\theta)$, and $c_3(\theta)$, we can quantify the potential performance improvements taking routing decisions at each hop rather than at the source only, and observing per-link delays rather than aggregate path delays.

III. FUNDAMENTAL PERFORMANCE LIMITS

In this section, we provide fundamental performance limits satisfied by any online routing policy in $\Pi_1$, $\Pi_2$, or $\Pi_3$. Specifically, we derive asymptotic (when $N$ grows large) regret lower bounds for our three types of policies. These bounds are obtained exploiting some results and techniques used in the control of Markov chains [10], and they are tight in the sense that there exist algorithms achieving these performance limits.

A. Regret Lower Bounds

We restrict our attention to so-called uniformly good policies, under which the number of times sub-optimal paths are selected until the transmission of the $n$-th packet is $o(n^\alpha)$ when $n \to \infty$ for any $\alpha > 0$. We know from [10] that such policies exist.

1) Source-Routing with Aggregate Feedback: Denote by $\psi^p(d)$ the probability that the delay of a packet sent on path $p$ is $d$ slots, and by $h(p)$ the length (or number of links) of path $p$. The end-to-end delay is the sum of several independent random geometric variables. For example, if we assume that $\theta_i \neq \theta_j$ for $i \neq j$, we have [11], for all $d \geq h(p)$,

$$\psi^p(d) = \sum_{i \in p} \theta_i (1 - \theta_i)^{d-1} \prod_{j \in p, j \neq i} \left( \frac{\theta_j}{\theta_j - \theta_i} \right).$$

The next theorem provides the fundamental performance limit of online routing policies in $\Pi_1$.

Theorem 3.1: For any uniformly good policy $\pi \in \Pi_1$,

$$\liminf_{N \to \infty} R^\pi_d(N)/\log N \geq c_1(\theta),$$

where $c_1(\theta)$ is the optimal value of the following optimization problem:

$$\inf_{c_p, p \in \mathcal{P}} \sum_{p \neq \pi^*} c_p(D_\theta(p) - D_\theta(p^*))$$

s.t. $c_p \geq 0$, $\forall p \in \mathcal{P}$,

$$\inf_{\lambda \in B_1(\theta)} \sum_{p \neq \pi^*} c_p \sum_{d = h(p)}^{\infty} \log \frac{\psi^p(d)}{\psi^p(d)} \geq 1. \tag{2}$$

and

$$B_1(\theta) := \left\{ \lambda : \{\lambda_i, i \in p^*\} = \{\theta_i, i \in p^*\}, \min_{p \in \mathcal{P}}\{D_\lambda(p)\} < D_\lambda(p^*) \right\}.$$

All proofs are presented in the Appendix. The variables $c_p$’s solving (2) have the following interpretation: for $p \neq p^*$, $c_p \log(N)$ is the asymptotic number of packets that need to be sent (up to the $N$-th packet) on sub-optimal path $p$ under optimal routing strategies in $\Pi_1$. So $c_1$ determines the optimal rate of exploration of sub-optimal path $p$. $B_1(\theta)$ is the set of bad network parameters: if $\lambda \in B_1(\theta)$, then the end-to-end delay distribution along the optimal path $p^*$ is the same under $\theta$ or $\lambda$ (hence by observing the end-to-end delay on path $p^*$, we cannot distinguish $\lambda$ or $\theta$), and $p^*$ is not optimal under network parameter $\lambda$. Please refer to [10] for a more detailed discussion. It is important to observe that in the definition of $B_1(\theta)$, the equality $\{\lambda_i, i \in p^*\} = \{\theta_i, i \in p^*\}$ is a set equality, i.e., order does not matter (e.g., if $p^* = \{1, 2\}$, the equality means that either $\lambda_1 = \theta_1, \lambda_2 = \theta_2$ or $\lambda_1 = \theta_2, \lambda_2 = \theta_1$).

2) Source-Routing with Detailed Feedback: We now consider routing policies in $\Pi_2$ that make decisions at the source, but have information on the individual link delays. Let $\text{KLG}(\theta_i, \theta_j)$ denote the Kullback-Leibler (KL) divergence number between two geometric random variables with parameters $\theta_i$ and $\theta_j$:

$$\text{KLG}(\theta_i, \theta_j) := \sum_{d \geq 1} \theta_i (1 - \theta_i)^{d-1} \log \frac{\theta_i (1 - \theta_i)^{d-1}}{\theta_j (1 - \theta_j)^{d-1}}.$$

Theorem 3.2: For any uniformly good policy $\pi \in \Pi_2$,

$$\liminf_{N \to \infty} R^\pi_d(N)/\log N \geq c_2(\theta), \tag{3}$$

where $c_2(\theta)$ is the optimal value of the following optimization problem:

$$\inf_{c_p, p \in \mathcal{P}} \sum_{p \neq p^*} c_p(D_\theta(p) - D_\theta(p^*))$$

s.t. $c_p \geq 0$, $\forall p \in \mathcal{P}$,

$$\inf_{\lambda \in B_2(\theta)} \sum_{p \neq p^*} c_p \sum_{d = h(p)}^{\infty} \text{KLG}(\theta_i, \lambda_i) \geq 1. \tag{4}$$

and

$$B_2(\theta) := \left\{ \lambda : \lambda_i = \theta_i, \forall i \in p^*, \min_{p \in \mathcal{P}}\{D_\lambda(p)\} < D_\lambda(p^*) \right\}.$$
that in [5], and moreover, our bound is tight (it cannot be improved further).

3) Hop-by-Hop Routing: Finally, consider routing policies in \( \Pi_3 \). These policies are more involved to analyze as the routing choices may change at any intermediate node in the network, and they are also more complex to implement. The next theorem states that surprisingly, the regret lower bound for hop-by-hop routing policies is the same as that derived for strategies in \( \Pi_2 \) (source routing with detailed feedback). In other words, we cannot improve the performance by taking routing decisions at each hop.

**Theorem 3.3:** For any uniformly good rule \( \pi \in \Pi_3 \),

\[
\liminf_{N \to \infty} R_\theta^N(N)/\log N \geq c_3(\theta) = c_4(\theta).
\]

Recall that as shown in [10], the regret lower bounds derived in Theorems 3.1-3.2-3.3 are tight in the sense that one can design actual routing policies achieving these regret bounds (although these policies might well be extremely complex and unpractical). Hence from the fact that \( c_1(\theta) \geq c_2(\theta) = c_3(\theta) \), we conclude that:

1) The best source routing policy with detailed feedback asymptotically achieves a lower regret than the best source routing policy with aggregate feedback;
2) The best hop-by-hop routing policy asymptotically achieves the same regret as the best source routing policy with detailed feedback.

**B. Numerical Example**

There are examples of network topologies where the above asymptotic lower bounds on regret can be explicitly computed. This is the case for line networks, see e.g. Figure 1.

![Fig. 1. A line network. Node 1 is the source, node 5 is the destination.](image)

In line networks, the optimal routing policy consists in selecting the best link on each hop. For any link \( i \), we denote by \( \xi(i) \) the optimal link on the same path as link \( i \).

**Proposition 3.4:** For any line network of length \( h \) hops,

\[
c_1(\theta) = \sum_{i: 1 \neq \xi(i)} \frac{1}{\hat{\theta}_i} - \frac{1}{\hat{\theta}_{\xi(i)}} \frac{1}{\sum_{d=1}^{\infty} \log \psi_\theta^d(d)},
\]

\[
c_2(\theta) = c_3(\theta) = \sum_{i: 1 \neq \xi(i)} \frac{1}{\hat{\theta}_i} - \frac{1}{\hat{\theta}_{\xi(i)}} \frac{1}{KLG(\theta_i, \theta_{\xi(i)})},
\]

where \( p^i \) is the path obtained from the optimal path \( p^* \) by replacing link \( \xi(i) \) by link \( i \).

It is easy to see that the difference between \( c_1(\theta) \) and \( c_2(\theta) \) increases with the network size \( h \). In Figure 2, we plot the ratio \( c_1(\theta)/c_2(\theta) \) averaged over various values of \( \theta \) (we randomly generated 10\(^6\) link parameters \( \theta \)) as a function of the network size \( h \). These results suggest that collecting detailed feedback (delays per link) can significantly improve the performance of routing policies, compared to just recording end-to-end delays. The gain is important even for fairly small networks – the regret is reduced by a factor 1500 on average in 6-hop networks when collecting per-link delays!

**IV. EFFICIENT ROUTING POLICIES**

In this section, we present two online routing policies, which are simple to implement, and yet approach the performance limits identified in the previous section. The first policy, referred to as KL-SR (Kullback-Leibler-Source Routing), belongs to \( \Pi_2 \) (routing decisions are taken at the source based on detailed feedback). The second policy, referred to as KL-HHR (Kullback-Leibler-Hop-by-Hop Routing), belongs to \( \Pi_3 \) (routing decisions are taken at each hop). These algorithms are simple index policies: each path is attached an index, and packets are sent on the path with the current minimal index. The index of a given path is further defined through the indexes of its constituting links. The latter indexes are the same as those used in the KL-UCB algorithm [12], an algorithm known to be asymptotically optimal for classical multi-armed bandit problems. We investigate the regret of the KL-SR and KL-HHR algorithms analytically and numerically; we show that they exhibit similar performance, and that they outperform existing algorithms. We also establish the asymptotical optimality of KL-SR in networks of specific topologies.

**A. The KL-SR Algorithm**

We first define the index for each link \( i \in E \). When the \( n \)-th packet is about to be sent, the index of link \( i \) is given by:

\[
I_i(n) = \min \left\{ \frac{1}{q} \in (1, \infty) : \frac{n_i \cdot \text{KLG}(\hat{\theta}_i(n), q)}{\log n + 3 \log(\log n)} \right\},
\]

(5)

where \( \hat{\theta}_i(n) \) is the empirical success probability on link \( i \) estimated over the transmissions of the \((n-1)\) first packets (i.e., \( \hat{\theta}_i(n) \) is the ratio of the number of successful transmissions on link \( i \) and of the total number of transmission attempts on link \( i \) before the \( n \)-th packet is sent); \( n_i \) is the number of packets routed through link \( i \) before the \( n \)-th packet is
sent. Now the index of path $p$ before the $n$-th packet is sent is given by $J_p(n) = \sum_{i \in p} I_i(n)$. For the $n$-th packet, the KL-SR algorithm selects path $p(n)$ with minimal index, i.e., $p(n) \in \arg \min_{p \in \mathcal{P}} J_p(n)$ (ties are broken arbitrarily). The algorithm requires an initialization phase so that link indexes are well defined, and its pseudo-code is provided below.

**Algorithm 1 KL-SR**

1. **// Initialization:**
2. Select a sequence of paths spanning all links to initialize $I_i(0)$ for all $i$ and $J_p(0)$ for all $p$
3. **// Main Loop:**
4. while $n \geq 0$ do
5. Select path $p(n) \in \arg \min_{p \in \mathcal{P}} J_p(n)$
6. Collect feedbacks on links $i \in p(n)$
7. Update the indexes of links and paths

Next we derive an upper bound of the regret achieved under KL-SR. To this aim, we introduce the following optimization problem with variables $x_i$ for $i \notin p^*$ and $Z$,

$$
\begin{align*}
\text{maximize} & \quad Z \\
\text{subject to} & \quad D_0(p^*) \leq \frac{1}{x_i}, \quad \forall p \neq p^*, \\
& \quad Z \leq \text{KL}(\theta_i, x_i), \quad \forall i \notin p^*, \\
& \quad x_i > \theta_i, \forall i \notin p^*, x_i = \theta_i, \forall i \in p^*.
\end{align*}
$$

Let $\chi_i$ for all $i \notin p^*$ be the solution of the above problem, let $\Delta_{\max} = \max_{p \notin p^*} \{D_0(p) - D_0(p^*)\}$ and let $H$ be the maximum length of paths in $\mathcal{P}$. We are now ready to derive a regret upper bound satisfied by KL-SR.

**Theorem 4.1:** The regret under $\pi = \text{KL-SR}$ satisfies:

$$
\limsup_{N \to \infty} \frac{R_\pi^*(N)}{\log N} \leq H \Delta_{\max} \sum_{i \notin p^*} \frac{1}{\text{KL}(\theta_i, \chi_i)}.
$$

From the above theorem, we can deduce a rough upper bound on KL-SR regret. Indeed, in (6), selecting $x_i$ such that $\frac{1}{x_i} - \frac{1}{\chi_i} = \frac{\theta_i}{H}$, where $\Delta_i = \min_{p \in \mathcal{P}} \{D_0(p) - D_0(p^*)\}$, is a valid choice, and leads to a worse regret upper bound. Combining this observation with Pinsker’s inequality [12], we get $\text{KL}(\theta_i, x_i) \geq 2\Delta_i^2 \theta_i^3/H^2$. Thus, the regret under KL-SR is $O(H^3|E| \log N)$.

The next theorem states that KL-SR is asymptotically optimal on line networks.

**Theorem 4.2:** On line networks, the regret under $\pi = \text{KL-SR}$ satisfies:

$$
\limsup_{N \to \infty} \frac{R_\pi^*(N)}{\log N} \leq \sum_{i: i \notin \chi(i)} \frac{1}{\theta_i} - \frac{1}{\chi(i)} \text{KL}(\theta_i, \theta_{\chi(i)}) = c_2(\theta).
$$

### B. The KL-HHR Algorithm

KL-HHR resembles KL-SR, but takes routing decisions at each node. Let $r \in V$, and denote by $\mathcal{P}_r$, the set of loop-free paths from node $r$ to the destination. KL-HHR decisions rely on link indexes. At time $t$ and when the $n$-th packet is about to be sent or already in the network, the index of link $i$ is $I_i(t)$ defined as:

$$
I_i(t) = \min \left\{ \frac{1}{q} \in (1, \infty) : t_i \text{KL}(\theta_i(q), t) \leq \log n + 3 \log(\log n) \right\},
$$

where $t_i$ denotes the number of transmissions on link $i$ (including retransmissions) up to time $t$, and $\theta_i(q)$ is the empirical successful transmission probability on link $i$ at time $t$; and KL$(\theta, \lambda)$ is the KL divergence number between two Bernoulli distributions of respective parameters $\theta$ and $\lambda$, $\text{KL}(\theta, \lambda) := \theta \log \frac{\theta}{\lambda} + (1 - \theta) \log \frac{1 - \theta}{1 - \lambda}$.

From the link indexes, we define $J_r(t)$ as the minimum cumulative index from node $r$ to the destination:

$$
J_r(t) = \min_{p \in \mathcal{P}} \sum_{i \in p} I_i(t).
$$

$J_r(t)$ can be computed using Bellman-Ford algorithm with simple message passing among nodes. The idea behind KL-HHR is to mix the dynamic programming principle (used in Markov Decision Process), and bandit algorithms (using indexes as in UCB algorithms [7]). Roughly speaking, when the current packet is at node $r$ at time $t$, it is sent on link $i = (r, q) \in E$ that minimizes $I_i(t) + J_r(t)$ over all possible links $i$ starting at node $r$. The pseudo-code of KL-HHR is given below.

**Algorithm 2 KL-HHR**

1. **// Initialization:**
2. Select a sequence of paths spanning all links to initialize $I_i(0)$ for all $i$ and $J_r(0)$ for all $r \in V$
3. **// Main Loop:**
4. while $t \geq 0$ do
5. Suppose the packet is at node $r$
6. Transmit the packet to node $q^*$ (via link $(r, q^*) \in E$) where $q^* \in \arg \min_{q \in \mathcal{V}: (r, q) \in E} \{I_i(q) + J_r(q)\}$
7. Observe whether the transmission is successful or not
8. Update the indexes of links and nodes depending on the observation

The theoretical evaluation of the performance of KL-HHR is beyond the scope of this paper (it is much more complicated than the analysis of the regret of KL-SR). We only present simulation results to assess KL-SR performance.

### C. Numerical Examples

We compare the performance of KL-SR and KL-HHR to that of the algorithm recently proposed in [3]. This policy, referred to as GJK (the initials of the authors), belongs to $\Pi_2$ (source routing and detailed feedback), and relies on link indexes that resemble UCB indexes: the index of link $i$ before sending the $n$-th packet is $\tilde{D}_i(n) - \frac{(H+1) \log(n)}{n_i}$, where $D_i(n)$ is the empirical mean delay observed on link $i$ before
the \( n \)-th packet is sent, \( n_i \) is the number of packets routed through link \( i \) before the \( n \)-th packet is sent and \( H \) is the maximum length of paths in \( P \).

We consider the network depicted in Figure 3 with node 1 as the source and node 16 as the destination. The success transmission probabilities on the various links are randomly generated in \([0, 1]\) (uniformly at random). Figure 4 presents the regret as a function of the number of the received packets under KL-SR, KL-HHR, and GKJ. Our policies exhibit similar performance, and outperform GKJ. We believe that the reason for the improved performance is that KL-SR and KL-HHR are based on a refined notion of index that exploits the knowledge that delay on links are geometrically distributed. KL-SR and KL-HHR also rely on a link index that resembles that used in KL-UCB, an algorithm known to outperform the classical UCB policy upon which GKJ is based.

Figure 5 provides the regret of the three policies when several randomly chosen links have a success transmission probability close to 0. This type of scenarios is not rare in wireless networks where due to moving objects or strong interference, some links may be in deep fade. We observe that here, hop-by-hop routing outperforms source routing: KL-HHR can change routing decisions dynamically at intermediate nodes, and does not waste transmissions on bad links when they are discovered.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we study online shortest path routing problems with stochastic random delays on links, whose parameters are initially unknown and have to be estimated by actual packet transmissions. Three types of routing policies are analyzed: source routing with detailed feedback, source routing with aggregate feedback, and hop-by-hop routing. We assess the performance of these policies using the notion of regret, a metric that actually captures the time it takes for the policies to identify the best path. We derive tight asymptotic lower bounds for the regret of the three types of policies, and by comparing these three bounds, we observe that routing with detailed feedback significantly improves performance, while hop-by-hop routing does not. We also proposed two efficient algorithms that outperform existing routing policies.

REFERENCES

[1] N. Cesa-Bianchi and G. Lugosi, “Combinatorial bandits,” Journal of Computer and System Sciences, vol. 78, no. 5, pp. 1404–1422, 2012.
[2] A. György, T. Linder, G. Lugosi, and G. Ottucsák, “The on-line shortest path problem under partial monitoring,” Journal of Machine Learning Research, vol. 8, pp. 2369–2403, 2007.
[3] Y. Gai, B. Krishnamachari, and R. Jain, “Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations,” IEEE/ACM Transactions on Networking, vol. 20, no. 5, pp. 1466–1478, 2012.
[4] K. Liu and Q. Zhao, “Adaptive shortest-path routing under unknown and stochastically varying link states,” in WiOpt, 2012, pp. 232–237.
[5] T. He, D. Goeckel, R. Raghavendra, and D. Towsley, “Endhost-based shortest path routing in dynamic networks,” in IEEE INFOCOM, 2013.
[6] T. L. Lai and H. Robbins, “Asymptotically efficient adaptive allocation rules,” Advances in applied mathematics, vol. 6, no. 1, pp. 4–22, 1985.
[7] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” Machine Learning, vol. 47, pp. 235–256, 2002.
[8] R. D. Kleinberg, “Online decision problems with large strategy sets,” Ph.D. dissertation, MIT, 2005.
[9] S. Bubeck, “Introduction to online optimization,” http://www.princeton.edu/~sbubeck/BubeckLectureNotes.pdf, 2011.
[10] T. L. Graves and T. L. Lai, “Asymptotically efficient adaptive choice of control laws in controlled Markov chains,” SIAM Journal on Control and Optimization, vol. 35, no. 3, pp. 715–743, 1997.
[11] A. Sen and N. Balakrishnan, “Convolution of geometrics and a reliability problem,” Statistics & Probability Letters, vol. 43, no. 4, pp. 421–426, Jul. 1999.
[12] A. Garivier and O. Cappé, “The KL-UCB algorithm for bounded stochastic bandits and beyond,” in Proceedings of Conference On Learning Theory (COLT), 2011.
[13] T. Van Erven and P. Harremoës, “Rényi divergence and Kullback-Leibler divergence,” arXiv preprint arXiv:1206.2459, 2012.
APPENDIX I
THROUGHPUT REGRET

We first define another notion of regret corresponding to the achieved throughput (i.e., the number of packets successfully received by the destination per unit time). The throughput regret is introduced in this paper to ease the analysis, e.g., computing the throughput regret for hop-by-hop routing policy is much easier.

Define $\mu_\theta(p)$ as the average throughput on path $p$ given link success rates $\theta$: $\mu_\theta(p) = 1/D_\theta(p)$. The throughput regret $S_\theta^\pi(T)$ of $\pi$ over time horizon $T$ is:

$$S_\theta^\pi(T) := T\mu_\theta(p^\star) - \mathbb{E}_\theta\left\{N^\pi(T)\right\},$$

where $N^\pi(T)$ is the number of packets received up to time $T$ under $\pi$.

The following Lemma 1 states that the throughput regret $S_\theta^\pi(T)$ is somehow asymptotically equivalent to the regret $R_\theta^\pi(N)$ (i.e., the delay regret). In the proof of Theorem 3.3 we will derive throughput regret for hop-by-hop routing and convert to the equivalent delay regret by Lemma 1.

**Lemma 1**: For any $\pi \in \Pi_i$, $i = 1, 2, 3$, we have:

$$\lim_{T \to \infty} \inf \frac{S_\theta^\pi(T)}{\log(T)} = \mu_\theta(p^\star) \lim_{N \to \infty} \inf \frac{R_\theta^\pi(N)}{\log(N)}.$$

**Proof.** The proof relies on the fact that for all $T > 0$ and all $N^\pi(T)$,

$$\sum_{n=1}^{N^\pi(T)} D^\pi(n) \leq T \leq \sum_{n=1}^{N^\pi(T)} D^\pi(n) + D^\pi(N^\pi(T) + 1).$$

Now we have:

$$0 \leq T - \mathbb{E}_\theta\left\{\sum_{n=1}^{N^\pi(T)} D^\pi(n)|N^\pi(T)\right\} \leq D_{\max}$$

and subsequently,

$$0 \leq T - \mathbb{E}_\theta\left\{\sum_{n=1}^{N^\pi(T)} D^\pi(n)\right\} \leq D_{\max}$$

Hence,

$$S_\theta^\pi(T) - \mu_\theta(p^\star) R_\theta^\pi(N^\pi(T))$$

$$= S_\theta^\pi(T) - \mu_\theta(p^\star) \mathbb{E}_\theta\left\{\sum_{n=1}^{N^\pi(T)} D^\pi(n)\right\} + N^\pi(T)$$

$$= \mu_\theta(p^\star) \left(T - \mathbb{E}_\theta\left\{\sum_{n=1}^{N^\pi(T)} D^\pi(n)\right\}\right) + N^\pi(T) - \mathbb{E}_\theta\{N^\pi(T)\}$$

which is bounded almost surely since $N^\pi(T) - \mathbb{E}_\theta\{N^\pi(T)\}$ is bounded almost surely. The result easily follows from the fact that $\lim_{T \to \infty} N^\pi(T) = \infty$ almost surely. □

APPENDIX II
PROOFS OF THEOREMS 3.1, 3.2, AND 3.3

To derive the asymptotic regret lower bounds, we apply the techniques used by Graves and Lai [10] to investigate efficient adaptive decision rules in controlled Markov chains. We recall here their general framework. Consider a controlled Markov chain $(X_t)_{t \geq 0}$ on a countable state space $S$ with a control set $U$. The transition probabilities given control $u \in U$ are parameterized by $\theta$ taking values in a compact metric space $\Theta$: the probability to move from state $x$ to state $y$ given the control $u$ and the parameter $\theta$ is $P(x, y; u, \theta)$. The parameter $\theta$ is not known. The decision maker is provided with a finite set of stationary control laws $G = \{g_1, \ldots, g_K\}$ where each control law $g_j$ is a mapping from $S$ to $U$: when control law $g_j$ is applied in state $x$, the applied control is $u = g_j(x)$. It is assumed that if the decision maker always selects the same control law $g$ the Markov chain is then irreducible with respect to some maximum irreducibility measure and has stationary distribution $\pi^g_\theta$. Now the reward obtained when applying control $u$ in state $x$ is denoted by $r(x, u)$, so that the expected reward achieved under control law $g$ is: $\mu_\theta(g) = \sum_x r(x, g(x))\pi^g_\theta(x)$. There is an optimal control law given $\theta$ whose expected reward is denoted $\mu^*_\theta \in \arg\max_{g \in G} \mu_\theta(g)$. Now the objective of the decision maker is to sequentially apply controls so as to maximize the expected reward up to a given time horizon $T$. The performance of a decision scheme can be quantified through the notion of regret which compares the expected reward to that obtained by always applying the optimal control law.

**A. Source Routing with Aggregate Feedback**

We first prove Theorem 3.1. To this aim, we construct a controlled Markov chain as follows. The state space is $N$, the control set is the set of paths $P$, and the parameter $\theta = (\theta_1, \ldots, \theta_K)$ defines the success rates on the various links. The parameter $\theta$ takes value in the compact space $\Theta = [\epsilon, 1]|E|$ for $\epsilon$ arbitrarily close to zero. The set of control laws are stationary and each of them corresponds to a given path, i.e., $G = P$. A transition in the Markov chain occurs at time epochs where a new packet is sent. The state after a transition records the end-to-end delay of the packet. Hence the transition probabilities are

$$P(k, l; p, \theta) = \psi^\theta_p(l),$$

and do not depend on the starting state. The cost (the opposite of reward) at state $l$ is simply equal to the delay $l$. Let us fix $\theta$, and denote by $p^\star$ the corresponding optimal path. For any two sets of parameters $\theta$ and $\lambda$, we define the KL information number under path (or control law) $p$ as:

$$I^p(\theta, \lambda) = \sum_{d=\text{hit}(p)}^\infty \log \frac{\psi^\theta_p(d)}{\psi^\lambda_p(d)} \psi^\theta_p(d).$$

It can be easily checked that $I^p(\theta, \lambda) = 0$ if and only if the delays over path $p$ under parameters $\theta$ and $\lambda$ have the same distribution, which occurs if and only if the two following
sets are identical: $\{\theta_i, i \in p\}, \{\lambda_i, i \in p\}$. We further define $B_1(\theta)$ as the set of bad parameters $\lambda$ such that under $\lambda$, $p^*$ is not the optimal path, and such that $\theta$ and $\lambda$ are statistically not distinguishable (they lead to the same delay distribution along path $p^*$). Then:

$$B_1(\theta) = \{\lambda : \lambda_i, i \in p^* \} = \{\theta_i, i \in p^*\},$$

$$\min_{p \in P} \{D_\lambda(p)\} < D_\lambda(p^*)\}.$$

By applying Theorem 1 in [10], we conclude that the delay regret scales at least as $c_1(\theta) \log(N)$ where

$$c_1(\theta) = \inf_{c_p} \left\{ \sum_{p \neq p^*} c_p (D_\theta(p) - D_\theta(p^*)) :$$

$$c_p \geq 0 \forall p, \inf_{\lambda \in B_1(\theta)} \left( \sum_{p \neq p^*} \sum_{d} \log \frac{\psi^\theta(d)}{\psi^\theta(d)} \right) \geq 1 \right\}.$$

B. Source Routing with Detailed Feedback

The proof of Theorem 3.2 is similar to that of Theorem 3.1 except that here we have to account for the fact that the source gets feedback on per-link delays. To this aim, we construct a Markov chain that records the delay on each link of a path. The state space is $\mathbb{N}^{|E|}$, where $|E|$ is the number of links in the network. Transitions occur when a new packet is sent from the source, and the corresponding state records the observed delays on each link of the chosen path, and the components of the state corresponding to links not involved in the path are set equal to 0. For example, the state $(0, 1, 4, 0, 7)$ indicates that the path consisting of links 2, 3, and 5 has been used, and that the per-links delays are 1, 4, and 7, respectively. The cost of a given state is equal to the sum of its components (total delay). Now assume that path $p = (1, \ldots, h(p))$ is used to send a packet, then the transition probability to a state whose $i$-th component is equal to $d_i$, $k = 1, \ldots, h(p)$ (the other components are 0) is:

$$\prod_{i=k}^{h(p)} q_\theta(i, d_i),$$

where $q_\theta(i, d) = \theta_i (1 - \theta_i)^{d-1}$ for any link $i$ and any delay $d$.

Now the KL information number of $(\theta, \lambda)$ under path $p$ is defined by $I^p(\theta, \lambda)$

$$I^p(\theta, \lambda) = \sum_{d_1=1}^{\infty} \cdots \sum_{d_{h(p)}=1}^{\infty} \left[ \log \frac{\psi^\theta(i_1, d_1) \cdots \psi^\theta(i_h, d_h)}{\psi^\theta(i_1, d_1) \cdots \psi^\theta(i_h, d_h)} \right]$$

$$+ \sum_{d_{h(p)+1}=1}^{\infty} \left[ \log \frac{\psi^\theta(i_h, d_h)}{\psi^\theta(i_h, d_h)} \right] q_\theta(i_1, d_1) + \cdots$$

$$+ \sum_{d_{h(p)+1}=1}^{\infty} \left[ \log \frac{\psi^\theta(i_h, d_h)}{\psi^\theta(i_h, d_h)} \right] \psi^\theta(i_h, d_h)$$

$$= \sum_{i \in p} \text{KLG}(\theta_i, \lambda_i)$$

where $\text{KLG}(\theta_i, \lambda_i)$ is the KL-divergence between two geometric distributions parameterized by $\theta_i$ and $\lambda_i$. Note that under detailed feedback, we have $I^p(\theta, \lambda) = 0$ if and only if $\theta_i = \lambda_i$ for all $i \in p$. The set $B_2(\theta)$ of bad parameters is now defined as:

$$B_2(\theta) = \{\lambda : \lambda_i = \theta_i, \forall i \in p^*, \min_{p \in P} \{D_\lambda(p)\} < D_\lambda(p^*)\}.$$

Theorem 1 in [10] implies that:

$$c_2(\theta) = \inf_{c_p} \left\{ \sum_{p \neq p^*} c_p (D_\theta(p) - D(p^*)) :$$

$$c_p \geq 0 \forall p, \inf_{\lambda \in B_2(\theta)} \left( \sum_{p \neq p^*} \sum_{i \in p} \text{KLG}(\theta_i, \lambda_i) \geq 1 \right) \right\}.$$

C. Hop-by-Hop Routing

This case is more involved. We focus on the throughput regret since it is much easier than directly computing the delay regret. The equivalent asymptotic bound on the delay regret $R^p(N)$ is implied by Lemma 1.

We let the state of the Markov chain be the packet location. The action is the selected outgoing link. The transitions between two states take one time slot – the time to make a transmission attempt. Hence, the transition probability between state $x$ and $y$ with the action of using link $i$ is denoted as $P^i(x, y)$ (where $y \neq x$).

$$P^i(x, y) = \begin{cases} \theta_i & \text{if link } i \text{ connects node } x \text{ and } y; \\ 0 & \text{otherwise}. \end{cases}$$

On the other hand, the probability of staying at the same state is the transmission failure probability on link $i$ if link $i$ is an outgoing link,

$$P^i(x, x) = \begin{cases} 1 - \theta_i & \text{if link } i \text{ is an outgoing link; } \\ 1 & \text{otherwise}. \end{cases}$$

We assume that the packet is injected at the source immediately after the previous packet is successfully delivered, and we are interested in counting the number of successfully delivered packets. In order not to count the extra time slot we will spend at the destination, we use a single Markov chain to represent both the source and the destination.

We give a reward one whenever the packet is successfully delivered to the destination. Let $r(x, y, i)$ be the immediate reward after the transition from node $x$ to node $y$ under the action $i$, i.e.,

$$r(x, y, i) = \begin{cases} 1 & \text{if } y \text{ is the destination node; } \\ 0 & \text{otherwise}. \end{cases}$$

Hence $r(x, i)$ (i.e., the reward at state $x$ with action $i$) is

$$r(x, i) = \sum_{y} P^i(x, y) r(x, y, i)$$

$$= \begin{cases} \theta_i & \text{if link } i \text{ connects node } x \text{ and the destination; } \\ 0 & \text{otherwise}. \end{cases}$$

The stationary control law prescribes the action at each state, i.e., the outgoing link at each node. A stationary control law of this Markov chain is then a path $p$ on the network, and we assign arbitrary actions to the nodes that are not on
the path \( p \). The stationary distribution of the Markov chains under path \( p \) is then,

\[
\pi_\theta^p(x) = \begin{cases} 
\frac{1}{\sum_{i \in p} \theta_i}, & \text{if node } x \text{ is on the path } p \\
0, & \text{otherwise}
\end{cases}
\]

where \( i \in p \) means that the link \( i \) is used in the path \( p \), and \( p(x) \) denotes the link we choose at node \( x \). The long-run average reward of the Markov chain under control law \( p \) is

\[
\sum_x \pi_\theta^p(x)r(x, p(x)) = \frac{1}{\sum_{i \in p} \theta_i} = \mu_\theta(p).
\]

The optimal control law is then \( p^* \) with long run average reward \( \mu_\theta(p^*) \).

Fig. 6. A Markov chain example under a control law \( p \) where the values in the parenthesis denote the transition probability and the reward respectively.

The throughputs regret of a policy \( \pi \in \Pi_3 \) for this controlled Markov chain at time \( T \) is

\[
S_\theta^*(T) = T\mu_\theta(p^*) - \mathbb{E}_\pi \left( \sum_{t=1}^T r(x_t, \pi(t, x_t)) \right),
\]

where \( x_t \) is the state at time \( t \) and \( \pi(t, x_t) \) is the corresponding action for state \( x_t \) at time \( t \).

We have constructed a controlled Markov chain that corresponds to the hop-by-hop routing on the network. Now define \( \mathcal{P}^p(\theta, \lambda) \) as the KL information number for a control law \( p \), and \( \mathcal{I}_p^p(\theta, \lambda) \)

\[
\mathcal{I}_p^p(\theta, \lambda) = \sum_{x} \pi_\theta^p(x) \log \frac{P^p(x)}{P_p^p(x)}\end{equation}

where \( KL(\theta, \lambda) := \log(\frac{\lambda}{\lambda_i})\theta_i + \log(\frac{1-\lambda}{1-\lambda_i})(1 - \theta_i) \) is the KL-divergence between two Bernoulli random variables \( \theta_i \) and \( \lambda_i \) and the last step is implied by Lemma 3 in the Appendix VII. Note that \( \mathcal{P}^p(\theta, \lambda) = 0 \) if and only if \( \theta_i = \lambda_i \) for all \( i \in p \). Hence the set \( B_2(\theta) \) of bad parameters is:

\[
B_2(\theta) = \{ \lambda : \lambda_i = \theta_i \forall i \in p^*, \max_{p \in \mathcal{P}} \{ \mu_\lambda(p) \} > \mu_\lambda(p^*) \} = \{ \lambda : \lambda_i = \theta_i \forall i \in p^*, \min_{p \in \mathcal{P}} \{ D_\lambda(p) \} < D_\lambda(p^*) \}.
\]

Applying Theorem 1 in [10], we get

\[
\lim_{T \to \infty} S_\theta^*(T) / \log T \geq c_3^p(\theta)
\]

where

\[
c_3^p(\theta) = \inf_{c_p} \left\{ \sum_{p \neq p^*} c_p(\mu_\lambda(p^*) - \mu_\lambda(p)) : c_p \geq 0 \forall p, \inf_{\lambda \in B_2(\theta)} \sum_{p \neq p^*} c_p \mu_\lambda(p) \sum_{i \in p} KL(\theta_i, \lambda_i) \geq 1 \right\}.
\]

Hence by Lemma 1

\[
\lim \inf_{N \to \infty} R_\theta^N(N) / \log N \geq c_3(\theta)
\]

where \( c_3(\theta) = c_3^p(\theta) / \mu_\lambda(p*) \).

Lastly, observe the relation that

\[
\mu_\lambda(p^*) - \mu_\lambda(p) = \mu_\lambda(p^*) \mu_\lambda(p)(D_\lambda(p) - D_\lambda(p^*)).
\]

It follows \( c_3^p(\theta) / \mu_\lambda(p^*) = c_2(\theta) \) and hence \( c_3(\theta) = c_2(\theta) \).

APPENDIX III

PROOF OF PROPOSITION 3.4

We will derive the analytic expression of \( c_2(\theta) \) and \( c_1(\theta) \) for the line network.

A. Source Routing with Detailed Feedback \( c_2(\theta) \)

Let us first decompose the set \( B_2(\theta) \). Note the best path on a line network consists of the best link on each hop. In order to have \( \min_{p \in \mathcal{P}} \{ D_\lambda(p) \} < D_\lambda(p^*) \), at least one sub-optimal link \( i \) should have a higher success probability than the link \( \xi(i) \) under the parameter \( \lambda \). We let \( B_1(\theta) \) be the set where link \( i \) is better than the link \( \xi(i) \) under parameter \( \lambda \),

\[
B_1(\theta) := \{ \lambda : \lambda_i = \theta_j \forall j \in p^* \},
\]

\( \lambda_\theta(\xi(i)) \)

Therefore, we have that \( B_2(\theta) = \bigcup_{i \neq \xi(i)} B_1(\theta) \).

Note \( KL(\theta_\xi, \lambda_\xi) = 0 \) if and only if \( \theta_\xi = \lambda_\xi \) and it is monotone increasing in \( \lambda_\xi \) in the range \( \lambda_\xi > \theta_\xi \). Thus, for any \( \lambda \in B_1(\theta) \), the infimum is obtained when \( \lambda_i = \theta_\xi(i) \) and \( \lambda_j = \theta_\xi(j) \forall j \neq i \), i.e.,

\[
\inf_{\lambda \in B_1(\theta)} \sum_{p \neq p^*} c_p \sum_{i \in p} KL(\theta_i, \lambda_i) \geq 1
\]

is equivalent to

\[
KL(\theta_i, \lambda_\xi(i)) \sum_{p \in \mathcal{P}} c_p \geq 1.
\]

Moreover, we have that

\[
\sum_{p \neq p^*} c_p (D_\lambda(p) - D_\lambda(p^*)) = \sum_{p \neq p^*} c_p \sum_{i \in p} \left( \frac{1}{\theta_i} - \frac{1}{\theta_\xi(i)} \right)
\]

Define \( c^i := \sum_{p \neq p^*} c_p \). Hence, the optimization problem in Eq. 4 that computes \( c_2(\theta) \) is

\[
\min \sum_{i \neq \xi(i)} \left( \frac{1}{\theta_i} - \frac{1}{\theta_\xi(i)} \right) c^i
\]

subject to

\[
KL(\theta_i, \lambda_\xi(i)) c^i \geq 1 \quad \forall i \neq \xi(i), \quad c_p \geq 0 \forall p.
\]
Recall that \( p' \) is obtained from the optimal path \( p^* \) by replacing link \( \xi(i) \) by link \( i \). It can be verified that the minimum of the above linear program can be obtained with \( c_{p'} = \frac{1}{\text{KLG}(\theta_i, \theta_{\xi(i)})} \), and \( c_p = 0 \) for all \( p \neq p' \). Therefore, we have that

\[
c_2(\theta) = \sum_{i:i \neq \xi(i)} \left( \frac{1}{\theta_i} - \frac{1}{\theta_{\xi(i)}} \right) \frac{1}{\text{KLG}(\theta_i, \theta_{\xi(i)})}.
\]

**B. Source Routing with Aggregate Feedback \( c_1(\theta) \)**

Similarly, we decompose \( B_1(\theta) \) into parts where a link \( i \) is better than the link \( \xi(i) \) under the parameter \( \lambda \). With some abuse of notations, we also define

\[
B_1(\theta) := \{ \lambda : \{ \lambda_j : j \in p^* \} = \{ \theta_j : j \in p^* \}, \lambda_i > \lambda_{\xi(i)} = \theta_{\xi(i)} \}.
\]

Thus, \( B_1(\theta) = \bigcup_{i \neq \xi(i)} B_1(\theta) \). Define \( \tilde{B}_1(\theta) \) as the set where all links except link \( i \) keep the same parameter under \( \lambda \), i.e.,

\[
\tilde{B}_1(\theta) := \{ \lambda : \lambda_j = \theta_j \forall j \neq i, \lambda_i > \lambda_{\xi(i)} = \theta_{\xi(i)} \} \subset B_1(\theta)
\]

and \( \tilde{B}_1(\theta) := \bigcup_{i \neq \xi(i)} \tilde{B}_1(\theta) \). Note \( \tilde{B}_1(\theta) \) is a subset of \( B_1(\theta) \).

In what follows, we solve Eq. (2) under the set \( \tilde{B}_1(\theta) \) instead of \( B_1(\theta) \). Since this procedure is equivalent to solving Eq. (2) under a smaller number of constraints, we can conclude the proof by showing that the optimal \( c_p \) values from this procedure also satisfy the constraints \( B_1(\theta), \forall i \neq \xi(i) \).

Recall \( I^p(\theta, \lambda) \) is the KL information number under path \( p \),

\[
I^p(\theta, \lambda) = \sum_{d=\text{Inf}(p)}^{\infty} \log \frac{\psi_\theta^p(d)}{\psi_\lambda^p(d)} \psi_\lambda^p(d).
\]

For a fixed parameter \( \theta \), \( I^p(\theta, \lambda) \) becomes larger when \( \lambda \) is further away from \( \theta \) by the property that KL-divergence measures the difference between two distributions. Hence, the infimum of \( \sum_{p \neq p^*} c_p I^p(\theta, \lambda) \) for \( \lambda \in \tilde{B}_1(\theta) \) is obtained at \( \lambda_i = \theta_{\xi(i)} \), i.e.,

\[
\inf_{\lambda \in \tilde{B}_1(\theta)} \{ \sum_{p \neq p^*} c_p I^p(\theta, \lambda) \} \geq 1 \iff \sum_{p \neq p^*} c_p I^p(\theta, \vartheta^i) \geq 1,
\]

where \( \vartheta^i \) is the parameter with \( \vartheta^i_i = \theta_{\xi(i)} \) and all others the same as \( \theta \).

Hence, with the subset \( \tilde{B}_1(\theta) \), Eq. (2) is minimize

\[
\sum_{i:i \neq \xi(i)} \left( \frac{1}{\theta_i} - \frac{1}{\theta_{\xi(i)}} \right) \sum_{p \neq p^*} c_p
\]

subject to \( c_p \geq 0 \forall p, \sum_{p \neq p^*} c_p I^p(\theta, \vartheta^i) \geq 1 \forall i \neq \xi(i) \).

Now, we will derive the analytic solution to this optimization problem. For a given path \( p' \) where \( i \in p' \) and \( p' \neq p^* \), let \( \bar{\theta} \) be the new parameter such that \( \bar{\theta}_j = \theta_{\xi(j)} \) for any \( j \neq i, j \in p' \) and \( \bar{\theta}_j = \theta_j \) for all other \( j \). Let \( \tilde{\vartheta}^j \) defined similarly from \( \vartheta^j \). Thus, \( \bar{\psi}_\theta^p(d) = \psi_\theta^p(d) \) and \( \bar{\psi}_\vartheta^j(d) = \psi_\vartheta^j(d) \) for all \( d \), which implies that \( I^p(\bar{\theta}, \tilde{\vartheta}) = I^p(\theta, \vartheta^j) \).

Moreover, with respect to path \( p', \theta \) and \( \vartheta^j \) differ only on link \( i \) and do so \( \bar{\theta} \) and \( \tilde{\vartheta^j} \). We also have \( \bar{\theta}_i = \vartheta^j_i = \theta_{\xi(i)} \) and \( \bar{\theta}_j = \theta_j \). By Lemma \textsuperscript{3-C} in the Appendix III-C

\[
I^p(\bar{\theta}, \tilde{\vartheta}) \leq I^p(\bar{\theta}, \vartheta^j) = I^p(\theta, \vartheta^j).
\]

Hence, the minimum of the above optimization problem is obtained at \( c_{p'} = \frac{1}{I^p(\theta, \vartheta^j)} \) for all \( p \neq p' \) and \( c_p = 0 \) for all \( p \neq p' \).

Now we will show that the derived optimal \( c_p \) values also satisfy the constraints for \( \lambda \in B_1(\theta) \). Note \( p' \) is the path composed only by the link \( i \) and the links from the optimal path \( p^* \). We have \( \{ \lambda_j : j \in p', j \neq i \} = \{ \theta_j : j \in p', j \neq i \} \) by the definition of \( B_1(\theta) \) and \( \vartheta^j_i = \theta_{\xi(i)} \) for all \( j \in p', j \neq i \) by the definition of \( \tilde{\vartheta}^j \). Hence, \( \{ \lambda_j : j \in p', j \neq i \} = \{ \vartheta^j : j \in p', j \neq i \} \). Moreover, \( \lambda_i > \theta_{\xi(i)} = \vartheta^j_i \). Hence, for any \( \lambda \in B_1(\theta) \), it is further away from \( \theta \) than \( \vartheta^j \), and so

\[
\inf_{\lambda \in B_1(\theta)} \sum_{p \neq p^*} c_p I^p(\theta, \lambda) \geq \inf_{\lambda \in B_1(\theta)} \sum_{p \neq p^*} c_p I^p(\theta, \vartheta^j) \geq 1.
\]

Hence, these \( c_p \) values obtain the minimum in Eq. (2), and

\[
c_1(\theta) = \sum_{i: i \neq \xi(i)} \left( \frac{1}{\theta_i} - \frac{1}{\theta_{\xi(i)}} \right) \frac{1}{\text{KLG}(\theta_i, \theta_{\xi(i)})}.
\]

**C. Supporting Lemmas**

Lemma 2: For parameters \( \lambda^1, \lambda^2, \theta^1 \) and \( \theta^2 \). If there exists \( i \in p \) such that

- \( \lambda^2_i = \theta^2_i \geq \lambda^1_j = \theta^1_j \) for all \( j \neq i \) and \( j \in p \),
- \( \lambda^2_i = \lambda^1_i, \theta^2_i = \theta^1_i \) and \( \lambda^2_i - \theta^2_i = \lambda^1_i - \theta^1_i \geq 0 \),

then \( I^p(\theta^1, \lambda^1) \leq I^p(\theta^2, \lambda^2) \).

**Proof:** We mainly rely the proof on the fact that the KL divergence number measures the “distance” between two distributions. By the first condition, the average delay of the links other than \( i \) on the parameter \( \lambda_1 \) and \( \theta_1 \) is larger than that of \( \lambda_2 \) and \( \theta_2 \). The probability distribution of the delay of these links is more “spread-out”. Moreover, the delay distribution of the path is the convolution of the delay distribution of link \( i \) and the delay distribution of the rest links. Hence, the difference in distribution between \( \theta^1_i \) and \( \lambda^1_i \) is more diluted due to the convolution with a more spread-out distribution. Therefore, its KL-divergence number is smaller.

**APPENDIX IV**

**Proof of Theorem 4.1**

For the \( n \)-th packet, suppose path \( p(n) \) is chosen. Let \( C_i(n) \) denote the counter for link \( i \notin p^* \) till the \( n \)-th packet, and it is added by one for only one link \( i \) that satisfies

\[
p(n) \neq p^*, i \in p(n) \text{ and } i \in \arg \min_{j \notin p^*} \{ n, \text{KLG}(\theta_j(n), \chi_j) \}
\]
where $\text{KLG}^+(x, y) = \text{KLG}(x, y) 1_{x < y}$. We choose a random link in the case of more than one link that satisfy the conditions.

Moreover, let $C_p(n)$ be the number of times of a path $p$ is chosen till the $n$-th packet. Since only one $\hat{C}_i(n)$ is increased by one when a non-optimal path is chosen, we have $\sum_{i \notin p^*} \mathbb{E}[\hat{C}_i(N)] = \sum_{p \notin p^*} \mathbb{E}[C_p(N)]$. Note that $R^*_p(N) \leq \Delta_{\text{max}} \sum_{p \notin p^*} \mathbb{E}[C_p(N)]$ and hence the rest of the proof is to derive an upper bound of $\sum_{i \notin p^*} \mathbb{E}[\hat{C}_i(N)]$.

Let $X_i(n) = 1$ denote the event that $\hat{C}_i(n)$ is added by one at the $n$-th packet for the link $i$. Note since $X_i(n) = 1$, we have the condition that $i \in p(n)$.

Let $\tilde{p}(n) := p(n) \setminus \tilde{J}$ and $\tilde{p}^* := p^* \setminus \tilde{J}$ where $\tilde{J} := \{ j : j \in p(n), j \in p^* \}$. Since we choose the path $p(n)$, we have

$$ \sum_{j \in p^*} I_j(n) < \sum_{j \in p} I_j(n) \Rightarrow \sum_{j \in \tilde{p}} I_j(n) < \sum_{j \in \tilde{p}^*} I_j(n). $$

Thus,

\begin{align*}
\mathbb{I}\{ X_i(n) = 1 \} &\leq \mathbb{I}\{ \sum_{j \in p^*} I_j(n) \geq \sum_{j \in \tilde{p}^*} D_j \} \\
&+ \mathbb{I}\{ \sum_{j \in \tilde{p}} I_j(n) < \sum_{j \in \tilde{p}^*} D_j, X_i(n) = 1 \} \\
&\leq \sum_{j \in \tilde{p}} \mathbb{I}\{ I_j(n) \geq D_j \} \\
&+ \sum_{j \in \tilde{p}} \mathbb{I}\{ I_j(n) < \sum_{j \in \tilde{p}^*} D_j, X_i(n) = 1 \} \\
&:= A + B.
\end{align*}

Recall $D_j = \frac{1}{\theta_j}$ is the average delay of any link $j$. Hence,

$$ \mathbb{E}[\hat{C}_i(N)] = \mathbb{E}\left[ \sum_{n=1}^N \mathbb{I}\{ X_i(n) = 1 \} \right] \leq \mathbb{E}\left[ \sum_{n=1}^N (A + B) \right]. $$

Now, let us bound $A$ and $B$ separately. We have that

\begin{align*}
\mathbb{E}\left[ \sum_{n=1}^N A \right] &= \mathbb{E}\left[ \sum_{n=1}^N \mathbb{I}\{ I_j(n) \geq D_j \} \right] \\
&\leq \sum_{n=1}^N \sum_{j \in p^*} \mathbb{P}\left\{ \frac{1}{I_j(n)} \leq \frac{1}{D_j} \right\} \\
&= \sum_{n=1}^N \sum_{j \in p^*} \mathbb{P}\left\{ \frac{1}{I_j(n)} \leq \frac{1}{D_j} \right\} \\
&\leq C_1' \log(\log(N)).
\end{align*}

By setting $\delta = \log(n) + 3\log(\log(n))$ in the Theorem 10 of [12], we have that for any $j \in p^*$,

\begin{align*}
\sum_{n=1}^N \mathbb{P}\left\{ \frac{1}{I_j(n)} \leq \frac{1}{D_j} \right\} \leq C_1' \log(\log(N))
\end{align*}

for some positive constant $C_1'$. Hence,

$$ \mathbb{E}\left[ \sum_{n=1}^N A \right] \leq \sum_{j \in p^*} C_1' \log(\log(N)) = HC_1' \log(\log(N)). $$

Now, let us focus on $\mathbb{E}\left[ \sum_{n=1}^N B \right]$. We will first derive some inequalities that will be used to drive the upper bound. We let $T(n) := \log(n) + 3\log(\log(n))$ and $T(N) := \log(N) + 3\log(\log(N))$ to save space in the proof.

Firstly, from the optimization problem (6) we have that

$$ \sum_{j \in p(n)} D_j \leq \sum_{j \in \tilde{p}} \frac{1}{x_j} \Rightarrow \sum_{j \in p(n)} D_j \leq \sum_{j \in \tilde{p}} \frac{1}{x_j}. $$

Note that $x_j = \theta_j$ for all $j \in p^*$.

Secondly, for any link $j$, we have from the definition of the index Eq. (5) that

$$ \frac{1}{I_j(n)} > \hat{\theta}_j(n) $$

and

$$ n_j \text{KLG}^+ (\hat{\theta}_j(n), \frac{1}{I_j(n)}) \leq T(n). $$

Thus, if $\frac{1}{I_j(n)} > \chi_j$, then

$$ n_j \text{KLG}^+ (\hat{\theta}_j(n), \chi_j) \leq n_j \text{KLG}^+ (\hat{\theta}_j(n), \frac{1}{I_j(n)}) \leq T(n). $$

Thirdly, recall the condition that

$$ i \in \arg \min_{j : j \notin \tilde{p}} \{ n_j \text{KLG}^+ (\hat{\theta}_j(n), \chi_j) \}. $$

Therefore, we have that

$$ \sum_{n=1}^N \mathbb{E}\left[ \sum_{j \in p(n)} I_j(n) \right] \leq \sum_{n=1}^N \mathbb{E}\left[ \sum_{j \in \tilde{p}} I_j(n) \right] \leq \sum_{n=1}^N \mathbb{E}\left[ \sum_{j \in \tilde{p}^*} I_j(n) \right] \leq n \text{KLG}^+ \left( \hat{\theta}_i(n), \chi_i \right) \leq T(n). $$


\[ H \sum_{s=1}^{N} \mathbb{1}\{s \text{KLG}^+(\hat{\theta}_i(s), \chi_i) \leq T(N)\} \sum_{n=s}^{N} \mathbb{1}\{n_i = s\} \]
\[ \leq H \sum_{s=1}^{N} \mathbb{1}\{s \text{KLG}^+(\hat{\theta}_i(s), \chi_i) \leq T(N)\} \]
where \( \sum_{n=s}^{N} \mathbb{1}\{n_i = s\} \leq 1 \) since \( i \in p(n) \).
For any \( \epsilon > 0 \), let
\[ K_n = \left[ \frac{1 + \epsilon}{\text{KLG}^+(\hat{\theta}_i, \chi_i)} T(N) \right] \]
and we have that
\[ \mathbb{E}\left[ \sum_{n=1}^{N} B_n \right] / H \]
\[ \leq \sum_{s=1}^{N} \mathbb{P}\{s \text{KLG}^+(\hat{\theta}_i(s), \chi_i) \leq T(N)\} \]
\[ \leq K_n + \sum_{s=K_n+1}^{\infty} \mathbb{P}\{s \text{KLG}^+(\hat{\theta}_i(s), \chi_i) \leq T(N)\} \]
\[ \leq K_n + \sum_{s=K_n+1}^{\infty} \mathbb{P}\{K_n \text{KLG}^+(\hat{\theta}_i(s), \chi_i) \leq T(N)\} \]
\[ = K_n + \sum_{s=K_n+1}^{\infty} \mathbb{P}\{\text{KLG}^+(\hat{\theta}_i(s), \chi_i) \leq \frac{\text{KLG}^+(\hat{\theta}_i, \chi_i)}{1 + \epsilon}\} \]
\[ \leq 1 + \frac{1}{\text{KLG}^+(\hat{\theta}_i, \chi_i)} T(N) + \frac{C_2'(\epsilon)}{N^{\beta(\epsilon)}} \]
where the last inequality follows from Lemma 8 in [12] and we use the relation \( \text{KLG}^+(\hat{\theta}_i, \chi_i) = \text{KLG}(\hat{\theta}_i, \chi_i) \) since \( \chi_i \geq \hat{\theta}_i \) for all \( i \).

Combining the bounds on \( A \) and \( B \), we have
\[ \mathbb{E}[\tilde{C}_i(N)] \leq H \frac{1 + \epsilon}{\text{KLG}^+(\hat{\theta}_i, \chi_i)} (\log N + 3 \log \log N) \]
\[ + HC_1^1 \log \log N. \]

Finally,
\[ R_g^*(N) = \sum_{p \neq p^*} \mathbb{E}[C_p(N)]\left[ \sum_{\theta_j} \frac{1}{\theta_j} - D_\theta(p^*) \right] \]
\[ \leq \Delta_{\text{max}} \sum_{p \neq p^*} \mathbb{E}[C_p(N)] \]
\[ = \Delta_{\text{max}} \sum_{i \neq p^*} \mathbb{E}[\tilde{C}_i(N)] \]
\[ \leq H \Delta_{\text{max}} (1 + \epsilon) \sum_{i \neq p^*} \frac{\log N}{\text{KLG}^+(\theta_i, \chi_i)} \]
\[ + C_1(\epsilon) \log \log N + C_2(\epsilon). \]

Therefore, we have proved that for any \( \epsilon > 0 \) there exist \( C_1(\epsilon) \) and \( C_2(\epsilon) \) such that
\[ R_g^*(N) \leq H \Delta_{\text{max}} (1 + \epsilon) \sum_{i \neq p^*} \frac{\log N}{\text{KLG}^+(\theta_i, \chi_i)} \]
\[ + C_1(\epsilon) \log \log N + C_2(\epsilon). \]

The theorem follows directly from this statement.

**APPENDIX V**

**PROOF OF THEOREM 4.2**

On the line network, the shortest path routing problem is simplified since the decisions at different hops are decoupled. It suffices to route on the best link on each hop, and the total regret is the summation of the regrets on all hops.

Note the index at link \( i \) for the \( n \)-packet is
\[ I_i(n) = \min \left\{ 1/q \in (1, \infty) : n \text{KLG}(\hat{\theta}_i(n), q) \leq T(n) \right\} \]
where \( T(n) = \log(n) + 3 \log \log(n) \). On the line network, Algorithm [1] chooses the link with the smallest index on each hop. Hence, on each hop, Algorithm [1] is equivalent to the KL-UCB algorithm for a classical multi-armed bandit problem with geometrically distributed rewards.

By [12, Theorem 1 and Lemma 6], we have that the KL-UCB algorithm achieves the asymptotical lower bound to the KL-UCB algorithm for a classical multi-armed bandit problem.

**APPENDIX VI**

**PROOF OF RELATION \( c_1(\theta) \geq c_2(\theta) \)**

Since \( B_2(\theta) \subset B_1(\theta) \), replacing the constraint set \( B_1(\theta) \) with \( B_2(\theta) \) gives a smaller value, i.e.,
\[ c_1(\theta) \geq \inf_{c_p} \left\{ \sum_{p \neq p^*} c_p (D_\theta(p) - D_\theta(p^*)) : \right\} \]
\[ c_p \geq 0, \inf_{\lambda \in B_2(\theta)} \left\{ \sum_{p \neq p^*} c_p \sum_{d=h(p)}^{\infty} \log \frac{\psi_p^\lambda(d)}{\psi_p^\lambda(d)} \geq 1 \right\} \]

Moreover, since we can observe the same aggregate delay with different delays on each link, the difference between distributions of the aggregate delay is smaller. According to the definition of the KL-divergence or more formally by the data processing inequality [13, Theorem 1], we have
\[ \sum_{d=h(p)}^{\infty} \log \frac{\psi_p^\lambda(d)}{\psi_p^\lambda(d)} \leq \sum_{i \in p} \text{KLG}(\theta_i, \lambda_i) \forall \theta, \lambda, p. \]

This concludes the proof that \( c_1(\theta) \geq c_2(\theta) \).
Appendix VII

Lemma 3: $\text{KLG}(\theta, \lambda)$ denote the KL-divergence number between two geometric random variables, and $\text{KL}(\theta, \lambda)$ denote the KL-divergence number between two Bernoulli random variables. We have

$$\text{KLG}(\theta, \lambda) = \frac{\text{KL}(\theta, \lambda)}{\theta}$$

where $\theta, \lambda \in (0, 1]$.

Proof:

$$\begin{align*}
\text{KLG}(\theta, \lambda) &= \sum_{i=1}^{\infty} \left[ \log \frac{\theta (1 - \theta)^{i-1}}{\lambda (1 - \lambda)^{i-1}} \right] \theta (1 - \theta)^{i-1} \\
&= \sum_{i=1}^{\infty} (\log \frac{\theta}{\lambda}) \theta (1 - \theta)^{i-1} \\
&\quad + \sum_{i=1}^{\infty} (i - 1) (\log \frac{1 - \theta}{1 - \lambda}) (1 - \theta)^{i-1} \\
&= \log \frac{\theta}{\lambda} + \left( \log \frac{1 - \theta}{1 - \lambda} \right) \frac{1 - \theta}{\theta} \\
&= \frac{1}{\theta} \left( \theta \log \frac{\theta}{\lambda} + (1 - \theta) \log \frac{1 - \theta}{1 - \lambda} \right) \\
&= \frac{\text{KL}(\theta, \lambda)}{\theta}.
\end{align*}$$