

# Optimally Balancing Energy Consumption versus Latency in Sensor Network Routing

WEI LAI and IOANNIS CH. PASCHALIDIS

Boston University

---

We consider wireless sensor networks with nodes switching ON (awake) and OFF (sleeping) to preserve energy, and transmitting data over channels with varying quality. The objective is to determine the best path from each node to a single gateway. Performance metrics of interest are: the expected energy consumption and the probability that the latency exceeds a certain threshold. Under Markovian assumptions on the sleeping schedules and the channel conditions, we obtain the expected energy consumption of transmitting a packet on any path to the gateway. We also provide an upper (Chernoff) bound and a tight large deviations asymptotic for the latency probability on each path. To capture the trade-off between energy consumption and latency probability we formulate the problem of choosing a path to minimize a weighted sum of the expected energy consumption and the exponent of the latency probability. We provide two algorithms to solve this problem: a centralized stochastic global optimization algorithm and a distributed algorithm based on simulated annealing. The proposed methodology can also optimize over the fraction of time sensor nodes remain ON (duty cycle).

Categories and Subject Descriptors: C.2.3 [**Network Operations**]: Network Management; C.4 [**Performance of Systems**]: Modeling Techniques, Performance Attributes; G.3 [**Probability and Statistics**]: Statistical Computing; G.1.6 [**Optimization**]: Constrained Optimization, Nonlinear Programming

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Sensor networks, Energy and resource management, Latency, Routing, Sleeping schedule.

---

## 1. INTRODUCTION

The emergence of wireless sensor networks has enabled numerous novel applications, e.g., indoor location detection [Ray et al. 2006], habitat monitoring [Mainwaring et al. 2002], smart homes and offices. For many of these applications, it is required that data from the sensors be received by the gateway in a timely fashion. In

---

Wei Lai is with the Center for Information and Systems Engineering, Boston University, Email: [wei@bu.edu](mailto:wei@bu.edu).

Ioannis Ch. Paschalidis is with the Center for Information and Systems Engineering, the Department of Manufacturing Engineering, and the Department of Electrical and Computer Engineering, Boston University, 15 St. Mary's St., Brookline, MA 02446, e-mail: [yannisp@bu.edu](mailto:yannisp@bu.edu), url: <http://ionia.bu.edu/>.

Research partially supported by the NSF under grants EFRI-0735974, DMI-0330171, CNS-0435312, ECS-0426453, and by the DOE under grant DOE DE-FG52-06NA27490.

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 20YY ACM 0000-0000/20YY/0000-0001 \$5.00

a surveillance application, for instance, the detection of some (possibly moving) intruder should be communicated to the gateway within a reasonably short time-frame. Yet, as the sensor nodes are typically powered by batteries, and replacing them is expensive, packets have to be transmitted in the most energy efficient way to extend the network lifetime.

In general we cannot achieve both goals at the same time. For instance, in a dense network, when a distant node has a packet for the gateway it can use many short hops (and hence, low power transmissions) to minimize the energy consumption, which causes a large delay. Alternatively, it can use few long hops which incurs a small delay but consumes a lot of energy. Thus, a natural, yet very important, question is: how should those nodes cooperate to balance energy and latency considerations?

Until now there has been an extensive body of research on energy efficient communication protocols from many different perspectives. For example, LEACH [Heinzelman et al. 2002] is a cluster forming approach that uses TDMA schedules to reduce energy consumption. [Ergen and Varaiya 2006] proposes another TDMA scheme that extends to multihop networks assuming there is a high power access point that can reach every sensor in the network. Maximum lifetime routing is considered in [Chang and Tassiulas 2004] where a linear programming problem is formulated to maximize the system lifetime, i.e., the time till the first node dies. Adaptive sleeping policies for sensors are proposed in [Fuemmeler and Veeravalli 2007] to address the tradeoff between energy consumption and tracking error in the framework of partially observable Markov decision processes, and the simulation results show that the proposed sleeping policies perform well relative to the optimal policy. Some of the earlier work has also attempted to address the tradeoff between energy and latency in sensor networks. The minimum energy transmission scheduling problem is considered in [Yu et al. 2004] assuming a given structure of a data aggregation tree, where the data face delivery deadlines. SMAC is a media access control protocol proposed in [Ye et al. 2004] which maintains a synchronized periodic sleeping schedule within a cluster of sensors. Reductions in energy consumption are achieved by decreasing the duty cycle of the common schedule. [Lu et al. 2005] proposes to schedule the activities of the sensor nodes to reduce the latency caused by sleeping nodes, while meeting the duty cycle requirement of each sensor. [Miao and Casandras 2006] focuses on the single hop transmission control problem in which each packet has a different deadline and number of bits, and the objective is to minimize the total energy consumption for transmission while satisfying a latency constraint for each packet.

In this work we consider a more unified and challenging energy-latency trade-off problem by taking into account factors that have not been studied in the same framework before, namely, the effect of varying channel conditions and the use of routing in optimizing a metric that combines energy consumption and latency Quality-of-Service (QoS). To that end, answers to the following two questions are critical in evaluating the performance of a sensor network:

- On average, how much energy is needed to transmit the data from each source to the gateway?
- What percentage of the data generated by each source cannot reach the gateway

before some reasonable deadline?

Motivated by these questions we will work with the expected energy consumption for transmitting the data to the gateway and we will quantify the latency QoS by the probability that delay along a path exceeds a certain amount. For some applications, the expected delay along a path may also be an acceptable latency QoS metric and, indeed, it has been studied in the literature [Ye et al. 2004; Lu et al. 2005]. As we will see later on, our framework can handle this case as well. Of course, working with the latency probability allows far greater generality and can accommodate applications with very stringent QoS requirements.

Modeling the varying channel conditions is critical for sensor network applications. Wireless channel unreliability is a critical phenomenon, especially at the low power levels sensors use to communicate. It affects both the latency, as unsuccessful transmissions are repeated, and energy consumption as nodes are forced to transmit more often and at higher power to successfully forward their packets. To capture the channel characteristics, we utilize finite state Markov models which have been proposed and widely analyzed in the literature [Zhang and Kassam 1999; Wang and Moayeri 1995]. In particular, the Gilbert-Elliot model [Gilbert 1960; Elliot 1963] for burst noise channels can be seen as a special case of the finite state Markov models we use in this paper.

Equally important is to capture nodes turning their radios OFF (going to sleep) to preserve energy, a tactic used by most sensor network platforms. In our model, we assume that the sensors turn their radios ON and OFF independently from each other, and that the transitions between ON and OFF states are randomized, or to be more specific, Markovian. Note that although synchronous and deterministic sleeping schedules have received a lot of attention in the literature, e.g. [Ye et al. 2004; Elson et al. 2002], aside from the purpose of a tractable analysis we choose to consider asynchronous and randomized strategies due to the following reasons:

(1) *Collision/interference avoidance.* With synchronous and deterministic sleeping schedules, nodes tend to wake up at the same time. Therefore, if nodes in a dense network are to exchange information with their neighbors after waking up, e.g., SYNC packets in SMAC [Ye et al. 2004], packet collisions and retransmissions will be plenty. Moreover, nodes could be easily get synchronized with some other periodic wireless interference source in their environment, e.g., some other sensor network, an 802.11 network with periodic traffic, etc. On the other hand, with asynchronous and randomized sleeping strategies we would have significantly less number of nodes awake at the same time when the duty cycle of the sensors is small, which is the typical scenario for practical sensor networks.

(2) *Security.* Synchronous and deterministic sleeping schedules are more vulnerable to Denial of Service (DoS) attacks and eavesdropping. One form of DoS is to selectively jam or interfere with transmissions [Wood and Stankovic 2002]. An adversary can easily achieve this by capturing the sleeping schedule and broadcasting malicious packets during the times nodes wake up. This attack would be difficult to detect without additional security protocols. However, with asynchronous and randomized sleeping strategies the adversary would not be able to predict when nodes are awake and would have to transmit malicious packets most of the time. This, however, is (energetically) expensive for the adversary and the attack is more

easily detectable as the rogue devices are transmitting more often.

We focus on selecting for each node a route to the gateway which minimizes a weighted sum of the expected energy consumed and a characterization of the latency probability (using large deviations ideas). This trade-off problem is related to problems that minimize energy subject to QoS constraints and leads to Pareto optimal points for the two-objective optimization problem involving energy and the latency probability. The problem turns out to be non-convex, thus challenging. We propose two alternatives: a centralized global optimization approach, which generalizes earlier work in [Paschalidis et al. 2007], and a distributed approach based on simulated annealing. A useful conclusion is that the distributed approach comes at a modest performance cost while it is much more appealing from an implementation and energy savings point of view. Essentially, our distributed routing approach orchestrates appropriate coordination between sensor nodes as they look for an optimal path. As we will see, it requires information exchange only between neighboring nodes.

Our framework can also be extended to optimize the sensors' duty cycles to achieve better performance; the simulated annealing algorithm we propose can be easily adapted to solve this joint routing and duty cycle optimization problem.

The rest of the paper is organized as follows. In Sec. 2 we introduce the network model. In Sec. 3 we characterize the expected energy consumption and latency probability on each path. In Sec. 4 we provide a large deviations asymptotic for the latency probability. The trade-off problem is formulated in Sec. 5, where we also introduce the two (centralized and distributed) solution approaches. In Sec. 6 we extend our framework to optimize over duty cycles. Numerical results are in Sec. 7 and conclusions in Sec. 8.

**Notational Conventions:** Throughout the paper all vectors are assumed to be column vectors. We use lower case boldface letters to denote vectors and for economy of space we write  $\mathbf{x} = (x_1, \dots, x_R)$  for the column vector  $\mathbf{x}$ .  $\mathbf{x}'$  denotes the transpose of  $\mathbf{x}$ ,  $\mathbf{0}$  the vector of all zeroes,  $\mathbf{e}$  the vector of all ones, and  $\mathbf{e}_i$  the  $i$ th unit vector. We use upper case boldface letters to denote matrices and write  $\mathbf{I}$  for the identity matrix and  $\mathbf{0}$  for the matrix of all zeroes. We use  $x^{\max}$  and  $x^{\min}$  to denote the largest and the smallest elements of vector  $\mathbf{x}$ , respectively. We denote by  $|\mathcal{A}|$  the cardinality of set  $\mathcal{A}$ . When the matrix  $\mathbf{A}$  is positive semidefinite we write  $\mathbf{A} \succeq \mathbf{0}$ . By  $\text{diag}(x_1, x_2, \dots, x_R)$  we denote the diagonal matrix with the diagonal elements given by  $x_1, x_2, \dots, x_R$ .

## 2. THE MODEL

Consider a Sensor NETWORK (SNET) with  $N$  nodes switching their radios "ON" and "OFF" to save energy. The subsystem controlling the array of sensors attached to a node may remain ON depending on the application. The motivation for switching the radios OFF comes from the fact that communication is energetically expensive. Hereafter, we will simply say that a node is ON or OFF and that would mean that the corresponding radios are ON or OFF, respectively. In the SNET, there is also a data sink (gateway) and all the information collected by the sensors is intended for this data sink. We identify the data sink as the  $N + 1$ st node of the SNET and we model the network as a directed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V} = \{1, \dots, N + 1\}$

is the set of nodes and  $\mathcal{E}$  is the set of all directed links. For any link  $k \in \mathcal{E}$ , we denote the origin and the destination of the link by  $s(k)$  and  $d(k)$ , respectively.

We assume time  $t$  is continuous. Furthermore, the channel state  $X_k$  for each link  $k \in \mathcal{E}$  is random and can take  $U_k + 1$  possible values, i.e.,  $0, \dots, U_k$ . We assume that  $\{X_k^t\}$  is a continuous time Markov process with irreducible transition rate matrix (infinitesimal generator)  $\mathbf{A}_k$ . (An implicit necessary condition is that the channel can be well represented by a stationary process over the time scale of interest; non-stationarities, see e.g., [Konrad et al. 2001], cannot be handled analytically in our framework.) Denote by  $a_k^{uu'}$  the  $(u, u')$  element of the matrix  $\mathbf{A}_k$ . For each  $k \in \mathcal{E}$ , when  $X_k$  is in state  $u$  an attempt to transmit a packet (that is, when both  $s(k)$  and  $d(k)$  are ON and  $s(k)$  has a packet to send to  $d(k)$ ) would be successful with probability  $p_k^{(u)}$  and unsuccessful with probability  $q_k^{(u)} = 1 - p_k^{(u)}$ . In the former case we will say that the channel is “good” and in the latter that the channel is “bad”, and we assume that the channel being “good” or “bad” only depends on the background noise. Therefore, the necessary and sufficient condition for a packet to be successfully transmitted on link  $k$  at time  $t$  is: (a)  $s(k)$  and  $d(k)$  are both ON at  $t$ , and (b) the channel  $k$  is “good” at  $t$ . Note that for any  $k \in \mathcal{E}$ , there exists at least one state  $u \in \{0, \dots, U_k\}$  such that  $p_k^{(u)} > 0$ ; otherwise  $k \notin \mathcal{E}$ . On the other hand, for convenience we assume that if  $d(k) = N + 1$  then there exists at least one state  $u \in \{0, \dots, U_k\}$  such that  $q_k^{(u)} > 0$ , namely, there are times when link  $k$  may have a “bad” channel. We will revisit this assumption in Sec. 3. Let us use  $\pi_k$  to denote the steady-state probability vector of the Markov process  $\{X_k^t\}$ , and thus the probability of  $X_k$  being in state  $u$  and channel  $k$  being “good” is given by  $\pi_k^{(u)} p_k^{(u)}$ ; similarly, the probability of  $X_k$  being in state  $u$  and channel  $k$  being “bad” is  $\pi_k^{(u)} q_k^{(u)}$ ,  $\forall u = 0, \dots, U_k$ . Last, let  $\tilde{\mathbf{A}}_k$  and  $\bar{\mathbf{A}}_k$  be the submatrices of  $\mathbf{A}_k$  corresponding to the states in the sets  $\{u \mid q_k^{(u)} > 0, u = 1, \dots, U_k\}$  and  $\{u \mid p_k^{(u)} > 0, u = 1, \dots, U_k\}$ , respectively. The following assumption remains in effect for the rest of the paper.

### Assumption A

- (i)  $\tilde{\mathbf{A}}_k$  and  $\bar{\mathbf{A}}_k$  are irreducible,  $\forall k \in \mathcal{E}$ .
- (ii)  $X_k^t$  and  $X_{k'}^t$  are independent,  $\forall k \neq k', k, k' \in \mathcal{E}$ .
- (iii)  $\{X_k^t\}$  is time reversible,  $\forall k \in \mathcal{E}$ .

**Remark (i):** Assumption (i) is not as restrictive as it seems to be. This assumption is satisfied when  $\tilde{\mathbf{A}}_k = \bar{\mathbf{A}}_k = \mathbf{A}_k$ , namely, for every state of  $X_k$  there is a positive probability that the packet cannot (or can) go through if both the transmitter and the receiver are ON. Moreover, the Gilbert-Elliot model [Gilbert 1960; Elliot 1963] (i.e., a two-state Markov process which with probability 1 determines the channel is “good” or “bad” at each point of time) can be seen as a special case where  $\tilde{\mathbf{A}}_k$  and  $\bar{\mathbf{A}}_k$  become one-element matrices.

**Remark (ii):** Assumption (ii) is necessary for a tractable analysis in our framework, and it holds in wireless sensor networks where the interference in a certain area does not affect the conditions of multiple links with different destinations. We note, however, that Assumption (ii) may be violated in networks with multi-

ple correlated interference sources, in which case the analysis becomes much more complicated.

**Remark (iii):** Assumption (iii) is a technical condition for the convenience of our analysis. As a special case of our framework, the Gilbert-Elliot model automatically satisfies this assumption. In fact this assumption may be replaced by some less restrictive, though less obvious, assumptions about channel states. We will further clarify this point in Appendix A.

For each link  $k \in \mathcal{E}$ , node  $s(k)$  consumes a certain amount of power while it is trying to establish a connection with node  $d(k)$ . We model this power consumption as a constant  $c_k$  (in Watts) for each  $k \in \mathcal{E}$ . Note that our assumption does not require  $s(k)$  to use a constant transmission power level. Rather,  $c_k$  has the interpretation of the average power consumption during connection establishment and it depends on the (condition of the) link. Once the connection is established and if  $X_k$  is in state  $u$  upon the connection, the energy that node  $s(k)$  uses to transmit data to  $d(k)$  is given by  $g_k^{(u)}$  (in Joules), which is also dependent on the link's condition. Throughout the paper we will adopt the convention that  $g_k^{(u)} = \infty$  if  $p_k^{(u)} = 0$ .

We define the random variable  $Y_i$  for each node  $i = 1, \dots, N$  as the state of node  $i$ , i.e., if node  $i$  is ON at time  $t$  then  $Y_i^t = 1$ ; otherwise  $Y_i^t = 0$ . The data sink, however, is always awake, thus,  $Y_{N+1}^t = 1$  for all  $t$ . For each node  $i = 1, \dots, N$ , we assume that  $Y_i^t$  evolves according to a continuous time Markov process with transition rate matrix  $\mathbf{B}_i$  when it does not receive or transmit any packets, where

$$\mathbf{B}_i = \begin{bmatrix} b_i^{00} & b_i^{01} \\ b_i^{10} & b_i^{11} \end{bmatrix},$$

and  $b_i^{01} > 0$  and  $b_i^{10} > 0$  are the transition rates from OFF to ON, and from ON to OFF, respectively. The steady state distribution of  $Y_i$  is denoted by  $\boldsymbol{\rho}_i = (\rho_i^{(0)}, \rho_i^{(1)})$  and the  $Y_i$ 's are independent for different nodes  $i$  and independent of all channel conditions.

We assume that once a node generates or receives data to forward, it remains in the ON state and keeps trying to transmit until it is successful (i.e., until the downstream node is ON and the channel state is “good”). Then, it resumes its sleeping schedule according to the corresponding Markov process.

We will be making an assumption that the SNET transports “few data generated at a low rate”, which is typical in sensor networks. Thus, data to be sent to the gateway gets generated at a single node at a time, fit in few packets whose transmission time is negligible, and the probability that a transmitting node finds its downstream node with independent data to transmit is also negligible. Moreover, data generation occurs in a much slower time scale than the one at which sleeping and channel state processes make transitions; hence, the corresponding Markov chains are in steady-state at data generation epochs. One implication of our “few data at a low rate” assumption is that the time at which a node acquires data to send is a random incidence on the Markovian sleeping schedule of the downstream node. Moreover, once the channel for some link  $k$  enters a state  $X_k$  we can assume that at most one packet may have to be transmitted before the channel switches to some other state  $X'_k$ .

Let us now focus on a node, say node 1; the analysis for all the other nodes is

identical. Suppose node 1 has data at time 0 which tries to deliver to the data sink, possibly in a multihop fashion. Due to the assumptions made earlier we consider the transmission, propagation, and queueing delays to be negligible. Thus, delays are only introduced by bad channel conditions and nodes that may be sleeping (OFF). Suppose the data is transmitted through a path  $\mathbf{p}$  from node 1 to  $N + 1$ . Then the total latency is given by

$$L_{\mathbf{p}} = \sum_{k \in \mathbf{p}} L_k,$$

where  $L_k$  is the latency the data experiences on link  $k$ . The energy spent on transmitting the data over this link is given by  $T_k = g_k^{(X_k)} + c_k L_k$ , where  $X_k$  indicates the channel state at the time of the connection establishment for link  $k$ . The total energy spent for sending the data over path  $\mathbf{p}$  is given by

$$T_{\mathbf{p}} = \sum_{k \in \mathbf{p}} T_k.$$

Note that for any fixed  $\mathbf{p}$ ,  $T_{\mathbf{p}}$  and  $L_{\mathbf{p}}$  are both random variables. We are interested in characterizing the following two quantities: (1)  $\mathbf{E}(T_{\mathbf{p}})$ : the expected energy consumption on path  $\mathbf{p}$ , and (2)  $\mathbf{P}(L_{\mathbf{p}} \geq d)$ : the probability that the delay of a packet on path  $\mathbf{p}$  exceeds a threshold  $d$ .

### 3. PERFORMANCE ANALYSIS

#### 3.1 Some Preliminary Results

First we present some preliminary results for any path  $\mathbf{p}$  from node 1 to the gateway. Let us first focus on the link  $k$  with  $s(k) = i$  and  $d(k) = j$ . We define the random variable  $O_k$  such that  $O_k = 1$  if the channel  $k$  is “good”; otherwise we let  $O_k = 0$ . Note that  $O_k$  is a Bernoulli random variable conditional on the channel state  $X_k$ . Because of our earlier “few data at a low rate” assumption it suffices to draw  $O_k$  only once every time the channel enters a new state  $X_k$ . Suppose  $\sigma$  is the last link on path  $\mathbf{p}$  and let  $\tilde{\mathbf{p}} = \mathbf{p} \setminus \{\sigma\}$ ; in the following we distinguish between two cases:  $k \in \tilde{\mathbf{p}}$  and  $k = \sigma$ .

**3.1.1 Case 1:  $k \in \tilde{\mathbf{p}}$ .** Consider the process with state  $(X_k, Y_j, O_k)$ . The state space is given by  $\mathcal{S}_k = \mathcal{S}_k^{(0)} \cup \mathcal{S}_k^{(1)} \cup \mathcal{S}_k^{(2)} \cup \mathcal{R}_k$  where

$$\begin{aligned} \mathcal{S}_k^{(0)} &= \{(u, 0, 0) \mid q_k^{(u)} > 0, u = 0, \dots, U_k\}, \\ \mathcal{S}_k^{(1)} &= \{(u, 1, 0) \mid q_k^{(u)} > 0, u = 0, \dots, U_k\}, \\ \mathcal{S}_k^{(2)} &= \{(u, 0, 1) \mid p_k^{(u)} > 0, u = 0, \dots, U_k\}, \\ \mathcal{R}_k &= \{(u, 1, 1) \mid p_k^{(u)} > 0, u = 0, \dots, U_k\}. \end{aligned}$$

Since  $\{X_k^t\}$  and  $\{Y_j^t\}$  are independent Markov processes, and  $O_k^t$  (probabilistically) depends only on  $X_k^\tau$  at any time  $\tau$ , it can be seen that  $\{X_k^t, Y_j^t, O_k^t\}$  is a Markov process as well. The process  $\{X_k^t, Y_j^t, O_k^t\}$  can be interpreted as follows. Consider first the Markov process  $\{X_k^t\}$  and augment the state by  $O_k^t$ . The resulting process is clearly Markovian since  $O_k^t$  depends only on  $X_k^t$ . Specifically, whenever  $\{X_k^t\}$  enters a new state  $u \in \{0, \dots, U_k\}$  we draw the Bernoulli random variable  $O_k^t$ ; depending on the outcome we will say that the  $\{X_k^t, O_k^t\}$  process is in state  $(u, 0)$  or  $(u, 1)$ . Let there be no transitions between the “substates”  $(u, 0)$  and  $(u, 1)$

and note that  $\{X_k^t, O_k^t\}$  changes state only when  $\{X_j^t\}$  does. Then the process  $\{X_k^t, Y_j^t, O_k^t\}$  can be seen as the product of the two independent Markov processes  $\{X_k^t, O_k^t\}$  and  $\{Y_j^t\}$ . We formalize this discussion as the following lemma.

**Lemma 3.1**  $\{X_k^t, Y_j^t, O_k^t\}$  is a Markov process,  $\forall k \neq \sigma$ .

Let  $\tilde{\mathcal{S}}_k = \mathcal{S}_k^{(0)} \cup \mathcal{S}_k^{(1)} \cup \mathcal{S}_k^{(2)}$  and permute the states of  $\{X_k^t, Y_j^t, O_k^t\}$  in the order of  $\mathcal{S}_k^{(0)}$ ,  $\mathcal{S}_k^{(1)}$ ,  $\mathcal{S}_k^{(2)}$  and  $\mathcal{R}_k$ . Then the transition matrix of  $\{X_k^t, Y_j^t, O_k^t\}$  is

$$\mathbf{\Omega}_k = \begin{bmatrix} \mathbf{H}_k & \mathbf{R}_k \\ \mathbf{V}_k & \mathbf{W}_k \end{bmatrix}$$

where  $\mathbf{H}_k \in \mathbb{R}^{|\tilde{\mathcal{S}}_k| \times |\tilde{\mathcal{S}}_k|}$ ,  $\mathbf{R}_k \in \mathbb{R}^{|\tilde{\mathcal{S}}_k| \times |\mathcal{R}_k|}$ ,  $\mathbf{V}_k \in \mathbb{R}^{|\mathcal{R}_k| \times |\tilde{\mathcal{S}}_k|}$ , and  $\mathbf{W}_k \in \mathbb{R}^{|\mathcal{R}_k| \times |\mathcal{R}_k|}$ . More specifically,  $\mathbf{H}_k$  is the submatrix of  $\mathbf{\Omega}_k$  corresponding to the states in  $\tilde{\mathcal{S}}_k$  while  $\mathbf{W}_k$  corresponds to those in  $\mathcal{R}_k$ .  $\mathbf{R}_k$  and  $\mathbf{V}_k$  are the transition rate matrices from  $\tilde{\mathcal{S}}_k$  to  $\mathcal{R}_k$ , and from  $\mathcal{R}_k$  to  $\tilde{\mathcal{S}}_k$ , respectively. As a special case, if  $0 < p_k^{(u)} < 1$  and  $0 < q_k^{(u)} < 1$  for all  $u = 1, \dots, U_k$ , then the matrices are given by:

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{H}_k^{(1)} & \mathbf{H}_k^{(2)} \end{bmatrix}, \quad \mathbf{V}_k = \begin{bmatrix} \mathbf{V}_k^{(1)} & \mathbf{V}_k^{(2)} \end{bmatrix},$$

$$\mathbf{H}_k^{(1)} = \begin{bmatrix} \omega_k^{00} & a_k^{01} q_k^{(1)} & \dots & a_k^{0U_k} q_k^{(U_k)} & b_j^{01} & 0 & \dots & 0 \\ a_k^{10} q_k^{(0)} & \omega_k^{11} & \dots & a_k^{1U_k} q_k^{(U_k)} & 0 & b_j^{01} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_k^{U_k 0} q_k^{(0)} & a_k^{U_k 2} q_k^{(1)} & \dots & \omega_k^{U_k U_k} & 0 & 0 & \dots & b_j^{01} \\ b_j^{10} & 0 & \dots & 0 & \omega_k^{U_k+1, U_k+1} & a_k^{01} q_k^{(1)} & \dots & a_k^{0U_k} q_k^{(U_k)} \\ 0 & b_j^{10} & \dots & 0 & a_k^{10} q_k^{(0)} & \omega_k^{U_k+2, U_k+2} & \dots & a_k^{1U_k} q_k^{(U_k)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & b_j^{10} & a_k^{U_k 0} q_k^{(0)} & a_k^{U_k 1} q_k^{(1)} & \dots & \omega_k^{2U_k+1, 2U_k+1} \\ 0 & a_k^{01} q_k^{(1)} & \dots & a_k^{0U_k} q_k^{(U_k)} & 0 & 0 & \dots & 0 \\ a_k^{10} q_k^{(0)} & 0 & \dots & a_k^{1U_k} q_k^{(U_k)} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_k^{U_k 0} q_k^{(0)} & a_k^{U_k 1} q_k^{(1)} & \dots & 0 & 0 & 0 & \dots & 0 \end{bmatrix},$$

$$\mathbf{H}_k^{(2)} = \begin{bmatrix} 0 & a_k^{01} p_k^{(1)} & \dots & a_k^{0U_k} p_k^{(U_k)} \\ a_k^{10} p_k^{(0)} & 0 & \dots & a_k^{0U_k} p_k^{(U_k)} \\ \vdots & \vdots & \ddots & \vdots \\ a_k^{U_k 0} p_k^{(0)} & a_k^{U_k 1} p_k^{(1)} & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \\ \omega_k^{2U_k+2, 2U_k+2} & a_k^{01} p_k^{(1)} & \dots & a_k^{0U_k} p_k^{(U_k)} \\ a_k^{10} p_k^{(0)} & \omega_k^{2U_k+3, 2U_k+3} & \dots & a_k^{1U_k} p_k^{(U_k)} \\ \vdots & \vdots & \ddots & \vdots \\ a_k^{U_k 0} p_k^{(0)} & a_k^{U_k 1} p_k^{(1)} & \dots & \omega_k^{3U_k+2, 3U_k+2} \end{bmatrix},$$

$$\mathbf{V}_k^{(1)} = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 & a_k^{01} q_k^{(1)} & \dots & a_k^{0U_k} q_k^{(U_k)} \\ 0 & 0 & \dots & 0 & a_k^{10} q_k^{(0)} & 0 & \dots & a_k^{1U_k} q_k^{(U_k)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & a_k^{U_k 0} q_k^{(0)} & a_k^{U_k 1} q_k^{(1)} & \dots & 0 \end{bmatrix},$$

$$\mathbf{V}_k^{(2)} = \begin{bmatrix} b_j^{10} & 0 & \dots & 0 \\ 0 & b_j^{10} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & b_j^{10} \end{bmatrix}, \quad \mathbf{R}_k = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \\ 0 & a_k^{01} p_k^{(1)} & \dots & a_k^{0U_k} p_k^{(U_k)} \\ a_k^{10} p_k^{(0)} & 0 & \dots & a_k^{1U_k} p_k^{(U_k)} \\ \vdots & \vdots & \ddots & \vdots \\ a_k^{U_k 0} p_k^{(0)} & a_k^{U_k 1} p_k^{(1)} & \dots & 0 \\ b_j^{01} & 0 & \dots & 0 \\ 0 & b_j^{01} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & b_j^{01} \end{bmatrix},$$

$$\mathbf{W}_k = \begin{bmatrix} \omega_k^{3U_k+3, 3U_k+3} & a_k^{01} p_k^{(1)} & \dots & a_k^{0U_k} p_k^{(U_k)} \\ a_k^{10} p_k^{(0)} & \omega_k^{3U_k+4, 3U_k+4} & \dots & a_k^{1U_k} p_k^{(U_k)} \\ \vdots & \vdots & \ddots & \vdots \\ a_k^{U_k 0} p_k^{(0)} & a_k^{U_k 1} p_k^{(1)} & \dots & \omega_k^{4U_k+3, 4U_k+3} \end{bmatrix}$$

and  $\omega_k^{ii}$  is such that the sum of each row of  $\mathbf{\Omega}_k$  is zero,  $\forall i = 0, \dots, 4U_k + 3$ .

Let  $\delta_k$  and  $\gamma_k$  be the stationary probability vectors of the states in  $\mathcal{S}_k$  and  $\mathcal{R}_k$ , respectively. As  $X_k$  and  $Y_j$  are independent, it can be seen that

$$\delta_k(u, 0, 0) = \pi_k^{(u)} \rho_j^{(0)} q_k^{(u)}, \quad \forall (u, 0, 0) \in \mathcal{S}_k^{(0)},$$

$$\begin{aligned}
\delta_k(u, 1, 0) &= \pi_k^{(u)} \rho_j^{(1)} q_k^{(u)}, \quad \forall (u, 1, 0) \in \mathcal{S}_k^{(1)}, \\
\delta_k(u, 0, 1) &= \pi_k^{(u)} \rho_j^{(0)} p_k^{(u)}, \quad \forall (u, 0, 1) \in \mathcal{S}_k^{(2)}, \\
\gamma_k(u, 1, 1) &= \pi_k^{(u)} \rho_j^{(1)} p_k^{(u)}, \quad \forall (u, 1, 1) \in \mathcal{R}_k.
\end{aligned}$$

Furthermore,  $\delta_k$  and  $\gamma_k$  are positive vectors, namely,  $\delta_k > \mathbf{0}$  and  $\gamma_k > \mathbf{0}$ . Moreover, by direct computation it can be verified that  $\{X_k^t, Y_j^t, O_k^t\}$  satisfies the detailed balance equations, and thus is a time reversible Markov process. We state this as the following lemma.

**Lemma 3.2** *The Markov process  $\{X_k^t, Y_j^t, O_k^t\}$  is time reversible,  $\forall k \neq \sigma$ .*

3.1.2 *Case 2:  $k = \sigma$ .* In this case as the gateway is always ON, let us consider the stochastic process  $\{X_\sigma^t, O_\sigma^t\}$  with state space  $\mathcal{S}_\sigma = \tilde{\mathcal{S}}_\sigma \cup \mathcal{R}_\sigma$  where

$$\begin{aligned}
\tilde{\mathcal{S}}_\sigma &= \{(u, 1, 0) \mid q_\sigma^{(u)} > 0, u = 1, \dots, U_\sigma\}, \\
\mathcal{R}_\sigma &= \{(u, 1, 1) \mid p_\sigma^{(u)} > 0, u = 1, \dots, U_\sigma\}.
\end{aligned}$$

Let us permute the states of  $\{X_\sigma, O_\sigma\}$  in the order of  $\tilde{\mathcal{S}}_\sigma$  and  $\mathcal{R}_\sigma$ . Following the same reasoning as in Sec. 3.1.1, we can see that the process  $\{X_\sigma^t, O_\sigma^t\}$  is also a Markov process with transition matrix

$$\Omega_\sigma = \begin{bmatrix} \mathbf{H}_\sigma & \mathbf{R}_\sigma \\ \mathbf{V}_\sigma & \mathbf{W}_\sigma \end{bmatrix},$$

where  $\mathbf{H}_\sigma \in \mathbb{R}^{|\tilde{\mathcal{S}}_\sigma| \times |\tilde{\mathcal{S}}_\sigma|}$ ,  $\mathbf{R}_\sigma \in \mathbb{R}^{|\tilde{\mathcal{S}}_\sigma| \times |\mathcal{R}_\sigma|}$ ,  $\mathbf{V}_\sigma \in \mathbb{R}^{|\mathcal{R}_\sigma| \times |\tilde{\mathcal{S}}_\sigma|}$ , and  $\mathbf{W}_\sigma \in \mathbb{R}^{|\mathcal{R}_\sigma| \times |\mathcal{R}_\sigma|}$ . Recall the assumption in Sec. 2 that there exists at least one state  $u \in \{0, \dots, U_\sigma\}$  such that  $q_\sigma^{(u)} > 0$ . Thus,  $|\tilde{\mathcal{S}}_\sigma| > 0$  and  $\mathbf{H}_\sigma$  is well defined. In the special case of  $0 < p_\sigma^{(u)} < 1$  and  $0 < q_\sigma^{(u)} < 1$  for all  $u = 1, \dots, U_\sigma$ , we have

$$\begin{aligned}
\mathbf{H}_\sigma &= \begin{bmatrix} \omega_\sigma^{00} & a_\sigma^{01} q_\sigma^{(1)} & \dots & a_\sigma^{0U_\sigma} q_\sigma^{(U_\sigma)} \\ a_\sigma^{10} q_\sigma^{(0)} & \omega_\sigma^{11} & \dots & a_\sigma^{1U_\sigma} q_\sigma^{(U_\sigma)} \\ \vdots & \vdots & \ddots & \vdots \\ a_\sigma^{U_\sigma 0} q_\sigma^{(0)} & a_\sigma^{U_\sigma 1} q_\sigma^{(1)} & \dots & \omega_\sigma^{U_\sigma U_\sigma} \end{bmatrix}, \\
\mathbf{R}_\sigma &= \begin{bmatrix} 0 & a_\sigma^{01} p_\sigma^{(1)} & \dots & a_\sigma^{0U_\sigma} p_\sigma^{(U_\sigma)} \\ a_\sigma^{10} p_\sigma^{(0)} & 0 & \dots & a_\sigma^{1U_\sigma} p_\sigma^{(U_\sigma)} \\ \vdots & \vdots & \ddots & \vdots \\ a_\sigma^{U_\sigma 0} p_\sigma^{(0)} & a_\sigma^{U_\sigma 1} p_\sigma^{(1)} & \dots & 0 \end{bmatrix}, \\
\mathbf{V}_\sigma &= \begin{bmatrix} 0 & a_\sigma^{01} q_\sigma^{(1)} & \dots & a_\sigma^{0U_\sigma} q_\sigma^{(U_\sigma)} \\ a_\sigma^{10} q_\sigma^{(0)} & 0 & \dots & a_\sigma^{1U_\sigma} q_\sigma^{(U_\sigma)} \\ \vdots & \vdots & \ddots & \vdots \\ a_\sigma^{U_\sigma 0} q_\sigma^{(0)} & a_\sigma^{U_\sigma 1} q_\sigma^{(1)} & \dots & 0 \end{bmatrix},
\end{aligned}$$

$$\mathbf{W}_\sigma = \begin{bmatrix} \omega_\sigma^{U_\sigma+1, U_\sigma+1} & a_\sigma^{01} p_\sigma^{(1)} & \dots & a_\sigma^{0U_\sigma} p_\sigma^{(U_\sigma)} \\ a_\sigma^{10} p_\sigma^{(0)} & \omega_\sigma^{U_\sigma+2, U_\sigma+2} & \dots & a_\sigma^{1U_\sigma} p_\sigma^{(U_\sigma)} \\ \vdots & \vdots & \ddots & \vdots \\ a_\sigma^{U_\sigma 0} p_\sigma^{(0)} & a_\sigma^{U_\sigma 1} p_\sigma^{(1)} & \dots & \omega_\sigma^{2U_\sigma+1, 2U_\sigma+1} \end{bmatrix},$$

and  $\omega_\sigma^{ii}$  is such that the sum of each row of  $\mathbf{\Omega}_\sigma$  is zero,  $\forall i = 0, \dots, 2U_\sigma + 1$ .

Denote by  $\delta_\sigma$  and  $\gamma_\sigma$  the stationary probability vectors of the states in  $\tilde{\mathcal{S}}_\sigma$  and  $\mathcal{R}_\sigma$ , respectively. Then we have

$$\begin{aligned} \delta_\sigma(u, 1, 0) &= \pi_\sigma^{(u)} q_\sigma^{(u)}, \quad \forall (u, 1, 0) \in \tilde{\mathcal{S}}_\sigma, \\ \gamma_\sigma(u, 1, 1) &= \pi_\sigma^{(u)} p_\sigma^{(u)}, \quad \forall (u, 1, 1) \in \mathcal{R}_\sigma, \end{aligned}$$

and as before we have the following lemma.

**Lemma 3.3**  $\{X_\sigma^t, O_\sigma^t\}$  is a time reversible Markov process.

In the rest of the paper we will need the following lemma regarding the eigenvalues of the matrix  $\mathbf{H}_k$ ,  $\forall k \in \mathbf{p}$ . The proof is in Appendix A.

**Lemma 3.4** For all  $k \in \mathbf{p}$ , all the eigenvalues of  $\mathbf{H}_k$  are negative real numbers.

### 3.2 Performance Metrics

Next, we characterize the two performance metrics  $\mathbf{E}(T_{\mathbf{p}})$  and  $\mathbf{P}(L_{\mathbf{p}} \geq d)$  for any path  $\mathbf{p}$  from node 1 to the gateway. Again let us first focus on the link  $k$  with  $s(k) = i$  and  $d(k) = j$ . Suppose node  $s(k)$  receives the data at time  $t$  and tries to deliver them to node  $d(k)$  immediately without any processing delay. Since  $s(k)$  receives data at  $t$ , its state has to be ON, thus, the probability distribution of  $L_k$  is identical with the conditional probability distribution of  $L_k$  given  $Y_i^t = 1$ , i.e.,

$$\mathbf{P}(L_k \leq \ell) = \mathbf{P}(L_k \leq \ell \mid Y_i^t = 1), \quad \forall \ell \geq 0.$$

For convenience we will denote by  $\Phi_k$  the random vector  $(X_k, Y_j, O_k)$  if  $k \in \tilde{\mathbf{p}}$ , or  $(X_k, O_k)$  if  $k = \sigma$  throughout this paper. To proceed we need the following lemma regarding the probability distribution of the first passage times of the Markov process  $\{\Phi_k^t\}$ . The proof is in Appendix B.

**Lemma 3.5** For any  $k \in \mathbf{p}$ , let the elements of vector  $\mathbf{f}_k(t)$  be the probability distribution functions of the first passage times of the Markov process  $\{\Phi_k^t\}$  from the states in  $\tilde{\mathcal{S}}_k$  to the set  $\mathcal{R}_k$ ,  $\forall k \in \mathbf{p}$ . Then  $\mathbf{f}_k(t) = (\mathbf{I} - e^{\mathbf{H}_k t})\mathbf{e}$ .

We are now ready to derive a characterization of the performance metrics.

**Theorem 3.6** For any path  $\mathbf{p}$ ,  $\ell \geq 0$  and  $k \in \mathbf{p}$

$$\mathbf{P}(L_k \leq \ell) = 1 - \delta'_k e^{\mathbf{H}_k \ell} \mathbf{e}, \quad (1)$$

$$\mathbf{E}(T_{\mathbf{p}}) = \sum_{k \in \mathbf{p}} (\tilde{g}_k - c_k \delta'_k \mathbf{H}_k^{-1} \mathbf{e}). \quad (2)$$

where  $\tilde{g}_k = \frac{\sum_{u=0}^{U_k} g_k^{(u)} \pi_k^{(u)} p_k^{(u)}}{\sum_{u=0}^{U_k} \pi_k^{(u)} p_k^{(u)}}$ ,  $\forall k \in \mathbf{p}$ .

PROOF. First note that  $\mathbf{P}(L_k = 0 \mid \Phi_k^t \in \mathcal{R}_k) = 1$  and we have

$$\begin{aligned} \mathbf{P}(L_k \leq \ell) &= \mathbf{P}(L_k \leq \ell \mid \Phi_k^t \in \mathcal{R}_k) \mathbf{P}(\Phi_k^t \in \mathcal{R}_k) + \mathbf{P}(L_k \leq \ell \mid \Phi_k^t \in \tilde{\mathcal{S}}_k) \mathbf{P}(\Phi_k^t \in \tilde{\mathcal{S}}_k) \\ &= \gamma'_k \mathbf{e} + \delta'_k \mathbf{f}_k(\ell) = \gamma'_k \mathbf{e} + \delta'_k (\mathbf{I} - e^{\mathbf{H}_k \ell}) \mathbf{e} = 1 - \delta'_k e^{\mathbf{H}_k \ell} \mathbf{e}, \end{aligned}$$

where the third equality uses the first passage time distribution of the Markov process  $\{\Phi_k^t\}$  to the set  $\mathcal{R}_k$  (cf. Lemma 3.5). This establishes (1).

For (2) let  $(X_k, Y_j, O_k)$  (or  $(X_k, O_k)$  if  $k = \sigma$ ) be the state of the Markov process at the time when the connection is established for link  $k \in \mathbf{p}$ . Note that  $\mathbf{E}(g_k^{(X_k)}) = \mathbf{E}(g_k^{(X_k)} \mid O_k = 1)$  and we have

$$\begin{aligned} \mathbf{E}(g_k^{(X_k)} \mid O_k = 1) &= \sum_{u=0}^{U_k} \mathbf{E}(g_k^{(X_k)} \mid X_k = u, O_k = 1) \mathbf{P}(X_k = u \mid O_k = 1) \\ &= \sum_{u=0}^{U_k} g_k^{(u)} \frac{\pi_k^{(u)} p_k^{(u)}}{\sum_{u=0}^{U_k} \pi_k^{(u)} p_k^{(u)}} = \tilde{g}_k, \end{aligned}$$

and

$$\mathbf{E}(L_k) = \int_0^\infty \mathbf{P}(L_k \geq \ell) d\ell = \int_0^\infty \delta'_k e^{\mathbf{H}_k \ell} \mathbf{e} d\ell = -\delta'_k \mathbf{H}_k^{-1} \mathbf{e}.$$

Last note that  $\mathbf{E}(T_{\mathbf{p}}) = \sum_{k \in \mathbf{p}} \mathbf{E}(T_k)$  and the conclusion follows.  $\square$

**Remark :** In Sec. 2 we assumed that for any link  $k$  with  $d(k) = N + 1$ , there exists some channel state  $u_k$  such that  $q_k^{(u_k)} > 0$ . Now let us suppose this assumption is violated for some link  $\sigma \in \mathcal{E}$ , then

$$\mathbf{P}(L_\sigma = 0) = 1, \quad \mathbf{E}(L_\sigma) = \sum_{u=0}^{U_\sigma} g_\sigma^{(u)} \pi_\sigma^{(u)} \triangleq \tilde{g}_\sigma.$$

Therefore, for any  $d > 0$  and any path  $\mathbf{p}$  such that  $\sigma \in \mathbf{p}$ , we have

$$\mathbf{P}(L_{\mathbf{p}} \geq d) = \mathbf{P}(L_{\tilde{\mathbf{p}}} \geq d), \quad \mathbf{E}(T_{\mathbf{p}}) = \mathbf{E}(T_{\tilde{\mathbf{p}}}) + \tilde{g}_\sigma.$$

As a result we can focus on  $\tilde{\mathbf{p}}$  instead of  $\mathbf{p}$  without loss of generality, and this avoids technical difficulties for Lemma 3.4 (cf. Appendix A).

#### 4. LARGE DEVIATIONS ASYMPTOTICS

With the latency probability on each link given in Thm. 3.6, in principle we can compute  $\mathbf{P}(L_{\mathbf{p}} \geq d)$  for any fixed path  $\mathbf{p}$  and constant  $d$ . However, this is computationally expensive as it involves a convolution of latency probabilities over all links in  $\mathbf{p}$ . In this section, we develop a large deviations asymptotic for  $\mathbf{P}(L_{\mathbf{p}} \geq d)$  which can be calculated with much lower complexity and involves a maximum over all links in  $\mathbf{p}$ . As we will see, such an expression is more amenable to a distributed computation and identifies a ‘‘bottleneck’’ link in  $\mathbf{p}$ .

Define vector the  $\mathbf{r}_k = -\mathbf{H}_k \mathbf{e}$  and note that  $\mathbf{r}_k \geq \mathbf{0}$ ,  $\mathbf{r}_k \neq \mathbf{0}$ . We define the logarithmic moment generating function  $\Lambda_k(\theta)$  for each link  $k \in \mathbf{p}$

$$\begin{aligned} \Lambda_k(\theta) &= \log \mathbf{E}(e^{\theta L_k}) \\ &= \log(\gamma'_k \mathbf{e} - \int_0^\infty e^{\theta \ell} \delta'_k e^{\mathbf{H}_k \ell} \mathbf{H}_k \mathbf{e} d\ell) \\ &= \log(\gamma'_k \mathbf{e} + \int_0^\infty \delta'_k e^{(\theta \mathbf{I} + \mathbf{H}_k) \ell} \mathbf{r}_k d\ell) \end{aligned}$$

for any  $\theta \geq 0$ .

The following lemma establishes the positivity of some matrices that will be used in the rest of the paper. The proof is in Appendix C. This lemma guarantees the applicability of the Perron-Frobenius theorem [Horn and Johnson 1990] regarding positive matrices.

**Lemma 4.1** *For any  $\ell > 0$  and  $\theta \geq 0$ , the matrices  $e^{\mathbf{H}_k \ell}$  and  $e^{(\theta \mathbf{I} + \mathbf{H}_k) \ell}$  are positive for any  $k \in \mathbf{p}$ .*

Let  $\lambda_k$  be the maximum eigenvalue of  $\mathbf{H}_k$  for any  $k \in \mathbf{p}$ . It follows that  $\lambda_k < 0$  as all eigenvalues of  $\mathbf{H}_k$  are negative (cf. Lemma 3.4). The next lemma establishes a useful property of  $\Lambda_k(\theta)$  for all  $k \in \mathbf{p}$ . The proof is in Appendix D.

**Lemma 4.2** *For any  $k \in \mathbf{p}$ , we have  $\Lambda_k(\theta) < \infty$  if  $\theta < -\lambda_k$ , and  $\Lambda_k(\theta) = \infty$  if  $\theta \geq -\lambda_k$ .*

Now we are ready to derive an upper bound on  $\mathbf{P}(L_{\mathbf{p}} \geq d)$ . Note that for any  $\theta \geq 0$  we have

$$\begin{aligned} \mathbf{P}(L_{\mathbf{p}} \geq d) &\leq \mathbf{P}(e^{\theta L_{\mathbf{p}}} \geq e^{\theta d}) \\ &\leq \mathbf{E}(e^{\theta L_{\mathbf{p}}}) e^{-\theta d} \\ &= e^{\sum_{k \in \mathbf{p}} \Lambda_k(\theta) - \theta d}, \end{aligned}$$

where we use the Markov inequality and the independence of  $L_k$ 's for all links  $k$  in  $\mathbf{p}$ . To obtain the tightest such upper bound, we minimize over  $\theta \geq 0$ , namely

$$\mathbf{P}(L_{\mathbf{p}} \geq d) \leq \exp\{\min_{\theta \geq 0} (\sum_{k \in \mathbf{p}} \Lambda_k(\theta) - \theta d)\}. \quad (3)$$

The objective function of the optimization problem in (3) is convex and differentiable. It can be verified that the function  $\Lambda_k(\theta)$  is strictly convex in  $\theta$ , and thus, the function  $\sum_{k \in \mathbf{p}} \Lambda_k(\theta) - \theta d$  is also strictly convex in its domain. Furthermore for any  $k \in \mathbf{p}$ , the derivative of  $\Lambda_k(\theta)$  is given by

$$\begin{aligned} \dot{\Lambda}_k(\theta) &= \frac{\frac{d}{d\theta} \int_0^\infty \delta'_k e^{(\theta \mathbf{I} + \mathbf{H}_k) \ell} \mathbf{r}_k d\ell}{\gamma'_k \mathbf{e} + \int_0^\infty \delta'_k e^{(\theta \mathbf{I} + \mathbf{H}_k) \ell} \mathbf{r}_k d\ell} \\ &= \frac{\int_0^\infty \frac{d}{d\theta} \delta'_k e^{(\theta \mathbf{I} + \mathbf{H}_k) \ell} \mathbf{r}_k d\ell}{\gamma'_k \mathbf{e} + \int_0^\infty \delta'_k e^{(\theta \mathbf{I} + \mathbf{H}_k) \ell} \mathbf{r}_k d\ell} \\ &= \frac{\int_0^\infty \ell \delta'_k e^{(\theta \mathbf{I} + \mathbf{H}_k) \ell} \mathbf{r}_k d\ell}{\gamma'_k \mathbf{e} + \int_0^\infty \delta'_k e^{(\theta \mathbf{I} + \mathbf{H}_k) \ell} \mathbf{r}_k d\ell}, \end{aligned}$$

where the derivative exists for any  $\theta < -\lambda_k$ . Note that  $\dot{\Lambda}_k(\theta)$  is strictly increasing due to the strict convexity of  $\Lambda_k(\theta)$ , for any  $k \in \mathbf{p}$ .

Now let us consider the latency probability  $\mathbf{P}(L_{\mathbf{p}} \geq d)$  and suppose  $d$  is greater than the expected latency along path  $\mathbf{p}$ , namely,

$$\sum_{k \in \mathbf{p}} \mathbf{E}(L_k) - d < 0.$$

Since  $\dot{\Lambda}_k(0) = \mathbf{E}(L_k)$  it follows

$$\sum_{k \in \mathbf{p}} \dot{\Lambda}_k(0) - d < 0,$$

which means that the derivative of the function  $\sum_{k \in \mathbf{p}} \Lambda_k(\theta) - \theta d$  is negative at  $\theta = 0$ . Given the fact that  $\sum_{k \in \mathbf{p}} \Lambda_k(\theta) - \theta d$  approaches infinity as  $\theta \rightarrow -\max_{k \in \mathbf{p}} \lambda_k$  (cf. Lemma 4.2), the necessary and sufficient condition for  $\theta_{\mathbf{p}}$  to be optimal in (3) is

$$\sum_{k \in \mathbf{p}} \dot{\Lambda}_k(\theta_{\mathbf{p}}) = d. \quad (4)$$

**Proposition 4.3 (Upper bound)** *For any path  $\mathbf{p} \in \mathcal{P}$*

$$\limsup_{d \rightarrow \infty} \frac{1}{d} \log \mathbf{P}(L_{\mathbf{p}} \geq d) \leq \max_{k \in \mathbf{p}} \lambda_k.$$

PROOF. Using (3), for any  $\mathbf{p} \in \mathcal{P}$  we obtain

$$\begin{aligned} \limsup_{d \rightarrow \infty} \frac{1}{d} \log \mathbf{P}(L_{\mathbf{p}} \geq d) &\leq \liminf_{d \rightarrow \infty} \frac{1}{d} \min_{\theta \geq 0} \left( \sum_{k \in \mathbf{p}} \Lambda_k(\theta) - \theta d \right) \\ &= \liminf_{d \rightarrow \infty} \frac{\partial}{\partial d} \min_{\theta \geq 0} \left( \sum_{k \in \mathbf{p}} \Lambda_k(\theta) - \theta d \right), \end{aligned}$$

where the equality is due to L'Hospital's rule and the differentiability of the convex dual of  $\sum_{k \in \mathbf{p}} \Lambda_k(\theta)$ . Let  $\theta_{\mathbf{p}}(d)$  be the optimal solution to the optimization problem in (3) with latency parameter  $d$ , and note that the first order optimality condition (4) is in effect. Applying the envelope theorem, we can obtain

$$\liminf_{d \rightarrow \infty} \frac{\partial}{\partial d} \min_{\theta \geq 0} \left( \sum_{k \in \mathbf{p}} \Lambda_k(\theta) - \theta d \right) = -\limsup_{d \rightarrow \infty} \theta_{\mathbf{p}}(d).$$

In the following we proceed to show

$$-\limsup_{d \rightarrow \infty} \theta_{\mathbf{p}}(d) = \max_{k \in \mathbf{p}} \lambda_k. \quad (5)$$

To see this, suppose  $\tau = \operatorname{argmin}_{k \in \mathbf{p}} (-\lambda_k)$  and note

$$\lim_{\theta \rightarrow -\max_{k \in \mathbf{p}} \lambda_k} \sum_{k \in \mathbf{p}} \dot{\Lambda}_k(\theta) \geq \lim_{\theta \rightarrow -\max_{k \in \mathbf{p}} \lambda_k} \dot{\Lambda}_{\tau}(\theta)$$

due to  $\dot{\Lambda}_k(\theta) > 0$  which follows from  $\dot{\Lambda}_k(0) \geq 0$  and the fact that the function  $\Lambda_k(\theta)$  is strictly convex,  $\forall k \in \mathbf{p}$ . Note that  $e^{(\theta + \lambda_{\tau})\ell}$  is the Perron-Frobenius (P-F) eigenvalue of matrix  $e^{(\theta \mathbf{I} + \mathbf{H}_{\tau})\ell}$ , and let  $\mathbf{v} > \mathbf{0}$  be the left P-F eigenvector of  $\mathbf{H}_{\tau}$  corresponding to eigenvalue  $\lambda_{\tau}$ , namely, the left P-F eigenvector of  $e^{(\theta \mathbf{I} + \mathbf{H}_{\tau})\ell}$  corresponding to the eigenvalue  $e^{(\theta + \lambda_{\tau})\ell}$ . Since  $\delta_{\tau}^{\max} \geq \delta_{\tau}^{\min} > 0$  and  $v^{\max} \geq v^{\min} > 0$ , we have

$$\begin{aligned} \delta'_{\tau} e^{(\theta \mathbf{I} + \mathbf{H}_{\tau})\ell} \mathbf{r}_{\tau} &\leq \frac{\delta_{\tau}^{\max}}{v^{\min}} \mathbf{v}' e^{(\theta \mathbf{I} + \mathbf{H}_{\tau})\ell} \mathbf{r}_{\tau} = \frac{\delta_{\tau}^{\max}}{v^{\min}} e^{(\theta + \lambda_{\tau})\ell} \mathbf{v}' \mathbf{r}_{\tau}, \\ \delta'_{\tau} e^{(\theta \mathbf{I} + \mathbf{H}_{\tau})\ell} \mathbf{r}_{\tau} &\geq \frac{\delta_{\tau}^{\min}}{v^{\max}} \mathbf{v}' e^{(\theta \mathbf{I} + \mathbf{H}_{\tau})\ell} \mathbf{r}_{\tau} = \frac{\delta_{\tau}^{\min}}{v^{\max}} e^{(\theta + \lambda_{\tau})\ell} \mathbf{v}' \mathbf{r}_{\tau}, \end{aligned}$$

and therefore

$$\begin{aligned} \lim_{\theta \rightarrow -\max_{k \in \mathbf{p}} \lambda_k} \sum_{k \in \mathbf{p}} \dot{\Lambda}_k(\theta) &\geq \lim_{\theta \rightarrow -\lambda_{\tau}} \frac{\int_0^{\infty} \ell \delta'_{\tau} e^{(\theta \mathbf{I} + \mathbf{H}_{\tau})\ell} \mathbf{r}_{\tau} d\ell}{\gamma_{\tau} \mathbf{e} + \int_0^{\infty} \delta'_{\tau} e^{(\theta \mathbf{I} + \mathbf{H}_{\tau})\ell} \mathbf{r}_{\tau} d\ell} \\ &\geq \lim_{\theta \rightarrow -\lambda_{\tau}} \frac{\int_0^{\infty} \ell \frac{\delta_{\tau}^{\min}}{v^{\max}} e^{(\theta + \lambda_{\tau})\ell} \mathbf{v}' \mathbf{r}_{\tau} d\ell}{\gamma_{\tau} \mathbf{e} + \int_0^{\infty} \frac{\delta_{\tau}^{\max}}{v^{\min}} e^{(\theta + \lambda_{\tau})\ell} \mathbf{v}' \mathbf{r}_{\tau} d\ell} \end{aligned}$$

$$= \lim_{\theta \rightarrow -\lambda_\tau} \frac{\frac{\delta_\tau^{\min}}{v^{\max}} \frac{1}{(\theta + \lambda_\tau)^2} \mathbf{v}' \mathbf{r}_\tau}{\gamma'_\tau e^{-\frac{\delta_\tau^{\max}}{v^{\min}}} \frac{1}{\theta + \lambda_\tau} \mathbf{v}' \mathbf{r}_\tau} = \infty.$$

Notice that  $\sum_{k \in \mathbf{p}} \dot{\Lambda}_k(\theta)$  is strictly increasing due to the strict convexity of  $\sum_{k \in \mathbf{p}} \Lambda_k(\theta)$  in the interval  $[0, -\lambda_\tau)$ . Furthermore, for any fixed  $\theta$  belonging to the interval  $[0, -\lambda_\tau)$ ,  $\sum_{k \in \mathbf{p}} \dot{\Lambda}_k(\theta)$  is finite. Thus, when  $d \rightarrow \infty$ , the only way for the first order optimality condition (4) to hold is to have  $\sum_{k \in \mathbf{p}} \dot{\Lambda}_k(\theta_{\mathbf{p}}) \rightarrow \infty$ , which by monotonicity is the same as to have  $\theta_{\mathbf{p}}$  approaching  $-\lambda_\tau$ , that is,  $\limsup_{d \rightarrow \infty} \theta_{\mathbf{p}}(d) = \min_{k \in \mathbf{p}} (-\lambda_k)$ . Therefore

$$-\limsup_{d \rightarrow \infty} \theta_{\mathbf{p}}(d) = -\min_{k \in \mathbf{p}} (-\lambda_k) = \max_{k \in \mathbf{p}} \lambda_k.$$

We then conclude

$$\begin{aligned} \limsup_{d \rightarrow \infty} \frac{1}{d} \log \mathbf{P}(L_{\mathbf{p}} \geq d) &\leq \liminf_{d \rightarrow \infty} \frac{\partial}{\partial d} \min_{\theta \geq 0} \left( \sum_{k \in \mathbf{p}} \Lambda_k(\theta) - \theta d \right) \\ &= -\limsup_{d \rightarrow \infty} \theta_{\mathbf{p}}(d) = \max_{k \in \mathbf{p}} \lambda_k. \end{aligned}$$

□

The next proposition provides an asymptotic lower bound on the delay probability for any path  $\mathbf{p}$ .

**Proposition 4.4 (Lower bound)** *For any path  $\mathbf{p} \in \mathcal{P}$*

$$\liminf_{d \rightarrow \infty} \frac{1}{d} \log \mathbf{P}(L_{\mathbf{p}} \geq d) \geq \max_{k \in \mathbf{p}} \lambda_k.$$

PROOF. Let  $\tau = \operatorname{argmax}_{k \in \mathbf{p}} \lambda_k$ . Then

$$\begin{aligned} \liminf_{d \rightarrow \infty} \frac{1}{d} \log \mathbf{P}(L_{\mathbf{p}} \geq d) &\geq \liminf_{d \rightarrow \infty} \frac{1}{d} \log \mathbf{P}(L_\tau \geq d) \\ &= \liminf_{d \rightarrow \infty} \frac{1}{d} \log(\delta'_\tau e^{\mathbf{H}_\tau d} \mathbf{e}). \end{aligned}$$

Now let  $\mathbf{v} > \mathbf{0}$  be the left eigenvector of  $\mathbf{H}_\tau$  corresponding to the eigenvalue  $\lambda_\tau$ , namely, the left Perron-Frobenius eigenvector of  $e^{\mathbf{H}_\tau d}$  corresponding to the eigenvalue  $e^{\lambda_\tau d}$ . We have

$$\begin{aligned} \liminf_{d \rightarrow \infty} \frac{1}{d} \log(\delta'_\tau e^{\mathbf{H}_\tau d} \mathbf{e}) &\geq \liminf_{d \rightarrow \infty} \frac{1}{d} \log\left(\frac{\delta_\tau^{\min}}{v^{\max}} \mathbf{v}' e^{\mathbf{H}_\tau d} \mathbf{e}\right) \\ &= \liminf_{d \rightarrow \infty} \frac{1}{d} \log(\mathbf{v}' e^{\mathbf{H}_\tau d} \mathbf{e}) \\ &= \liminf_{d \rightarrow \infty} \frac{1}{d} \log(\mathbf{v}' e^{\lambda_\tau d} \mathbf{e}) \\ &= \lambda_\tau = \max_{k \in \mathbf{p}} \lambda_k. \end{aligned}$$

□

The following theorem is a direct result of Propositions 4.3 and 4.4.

**Theorem 4.5** For any  $\mathbf{p} \in \mathcal{P}$

$$\lim_{d \rightarrow \infty} \frac{1}{d} \log \mathbf{P}(L_{\mathbf{p}} \geq d) = \max_{k \in \mathbf{p}} \lambda_k.$$

**Remark :** Thm. 4.5 establishes that  $\mathbf{P}(L_{\mathbf{p}} \geq d)$  decreases exponentially fast as  $d$  grows with dominant exponent equal to  $\max_{k \in \mathbf{p}} \lambda_k$ , namely,

$$\mathbf{P}(L_{\mathbf{p}} \geq d) \sim e^{d(\max_{k \in \mathbf{p}} \lambda_k)}$$

for large enough  $d$ . We refer to a link  $k^*$  which achieves the maximum above as a “bottleneck” link as its latency probability dominates the latency probability of the path. Furthermore, due to its structure, the exponent can be easily computed and updated in a distributed fashion for a certain path  $\mathbf{p}$ . For a path with  $n$  links its computation involves  $O(n)$  comparisons which can be done in one pass with only local information exchanges between links. Furthermore, slight modifications of the path (adding or removing a few links) result into a modest work for updating the exponent, which can be done in a distributed manner and with small communication overhead. As we will see, this is important in developing energy efficient routing algorithms in SNETs.

## 5. TRADE-OFF BETWEEN LATENCY PROBABILITY AND EXPECTED ENERGY CONSUMPTION

In this section we consider the problem of optimizing the route from a node, say node 1, to the gateway. In doing so we are faced with the trade-off between the latency probability and the expected energy consumption along the route. The following is a possible formulation to capture this trade-off:

$$\begin{aligned} \min_{\mathbf{p} \in \mathcal{P}} \quad & \sum_{k \in \mathbf{p}} \mathbf{E}(T_k) \\ \text{s.t.} \quad & \mathbf{P}(L_{\mathbf{p}} \geq d) \leq \epsilon. \end{aligned} \quad (6)$$

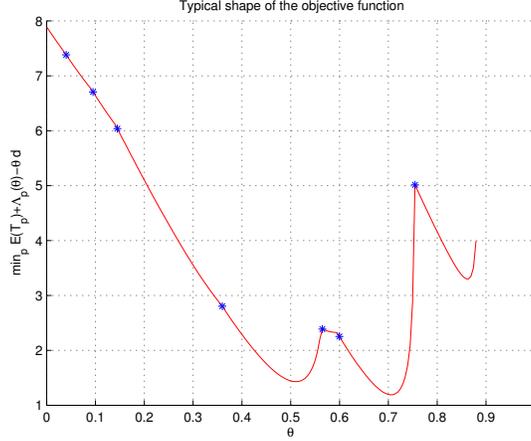
More specifically, we look for a path which minimizes the expected energy consumption while keeping the latency probability below an appropriately small level. We refer to the constraint in (6) as a Quality-of-Service (QoS) constraint. Unfortunately, problem (6) is known to be NP-complete as the *resource constrained shortest path problem* [Garey and Johnson 1979]. To simplify the problem we can relax the QoS constraint by using the Chernoff bound obtained in Sec. 4. Namely,

$$\begin{aligned} \min_{\mathbf{p} \in \mathcal{P}} \quad & \sum_{k \in \mathbf{p}} \mathbf{E}(T_k) \\ \text{s.t.} \quad & \min_{\theta \geq 0} (\Lambda_{\mathbf{p}}(\theta) - \theta d) \leq \log \epsilon, \end{aligned} \quad (7)$$

where  $\Lambda_{\mathbf{p}}(\theta) = \sum_{k \in \mathbf{p}} \Lambda_k(\theta)$ . Problem (7) is still difficult to solve; one approach is to apply Lagrangian relaxation and derive a lower bound that can be used in a branch-and-bound procedure [Bertsekas 1999]. The key step in the Lagrangian relaxation algorithm is to solve the following problem

$$\begin{aligned} \min \quad & \mathbf{E}(T_{\mathbf{p}}) + \beta \min_{\theta \geq 0} (\Lambda_{\mathbf{p}}(\theta) - \theta d) \\ \text{s.t.} \quad & \mathbf{p} \in \mathcal{P}, \end{aligned} \quad (8)$$

for nonnegative  $\beta$ . Motivated by this reasoning, we will focus on solving problem (8). An alternative way to look at problem (8) is to note that for varying

Fig. 1. Typical shape of  $f(\theta)$ .

$\beta$  it generates Pareto optimal points for the two-objective optimization problem with objectives the energy consumption  $\mathbf{E}(T_{\mathbf{p}})$  and the Chernoff upper bound  $\min_{\theta \geq 0} (\Lambda_{\mathbf{p}}(\theta) - \theta d)$  on the latency probability.

Let us therefore consider problem (8) and note that it is a nontrivial task to solve it. To illustrate this, change the order of minimizations in (8) to obtain

$$\min_{\theta \geq 0} \left[ \min_{\mathbf{p} \in \mathcal{P}} (\mathbf{E}(T_{\mathbf{p}}) + \beta \Lambda_{\mathbf{p}}(\theta)) - \theta \beta d \right]. \quad (9)$$

Let  $f(\theta) = \min_{\mathbf{p} \in \mathcal{P}} (\mathbf{E}(T_{\mathbf{p}}) + \beta \Lambda_{\mathbf{p}}(\theta)) - \theta \beta d$  and note that for each network, we only need to concentrate on the interval  $\theta \in [0, -\min_{k \in \mathcal{E}} \lambda_k)$  because  $f(\theta) = \infty$  for all  $\theta \geq -\min_{k \in \mathcal{E}} \lambda_k$ .

In Fig. 1 we plot the typical shape of  $f(\theta)$ . Since  $\Lambda_{\mathbf{p}}(\theta)$  is convex,  $f(\theta)$  is the minimum of a finite number of convex functions, thus, it is a piecewise convex function in  $\theta$ . For any fixed  $\theta$ ,  $f(\theta)$  can be evaluated efficiently (i.e., in polynomial time) by solving a shortest path problem. Whenever  $\theta$  passes a *change point* (marked as stars in the figure), the optimal solution  $\mathbf{p}$  changes to another path. Moreover, the first piece of the curve corresponds to the minimal expected energy path  $\operatorname{argmax}_{\mathbf{p} \in \mathcal{P}} \mathbf{E}(T_{\mathbf{p}})$ . It is evident that problem (9) is a global optimization problem (piecewise convex minimization), and that the minimal energy path is not necessarily optimal.

In the remainder of this section we discuss two different algorithms to solve problem (8). The first one is to use what we call a *Semi-Definite programming-based convex Polynomial Underestimation (SDPU)* technique which is essentially a centralized approach. The other algorithm is based on *simulated annealing* and can be implemented in a distributed fashion.<sup>1</sup>

<sup>1</sup>If the expected delay along a path is chosen as the latency QoS metric, problem (6) becomes a resource constrained shortest path problem with an additive constraint. We can construct an

### 5.1 Semi-Definite programming-based convex Polynomial Underestimation (SDPU)

Convex underestimation as part of a global optimization algorithm has been studied extensively. More recently, [Paschalidis et al. 2007] introduced a Semi-Definite programming-based Underestimation (SDU) technique that produces a general quadratic underestimator of an arbitrary function. SDU was able to handle some particularly challenging functions appearing in protein docking applications (see [Paschalidis et al. 2007]). In this subsection, we combine the SDU method with a higher order polynomial function fitting method from [Magnani et al. 2005]. The motivation is to generate underestimators that are more flexible than a convex quadratic function.

SDPU consists of two main tasks: (i) devising an underestimator of  $f(\theta)$ , and (ii) using the underestimator to guide the search for a global minimum. Towards (i) let us sample the function  $f(\theta)$  and obtain a set of  $M$  local minima  $\theta_1, \dots, \theta_M$  of  $f(\theta)$ . Note that this can be done efficiently by first randomly choosing a starting point  $\hat{\theta}_m$ , obtaining a path  $\mathbf{p}_m$  which solves the inner optimization problem in (9) with parameter  $\hat{\theta}_m$  using any deterministic shortest path algorithm (see e.g., [Bertsekas and Tsitsiklis 1989]), and then applying some standard gradient method to find the optimal  $\theta_m$  which solves the outer optimization problem of (9) for path  $\mathbf{p}_m$ .<sup>2</sup>

Given the set of local minima  $(\theta_1, f(\theta_1)), \dots, (\theta_M, f(\theta_M))$ , we construct the polynomial underestimator

$$u(\theta) = h_0 + h_1\theta + \dots + h_L\theta^L,$$

with  $u(\theta_m) \leq f(\theta_m)$ , for any  $m = 1, \dots, M$  and the degree  $L$  being an even positive number. Define

$$q(\theta, s) = s^2 \nabla^2 u(\theta) = s^2 (\sum_{\ell=2}^L h_\ell \ell(\ell-1) \theta^{\ell-2}).$$

The necessary and sufficient condition for the function  $u(\theta)$  to be convex is that  $q(\theta, s)$  can be expressed as a Sum of Squares (SOS) [Magnani et al. 2005], namely, there exists a vector  $\mathbf{y}$  with each component being a monomial in  $\theta$  and  $s$  of degree less than or equal to  $L/2$ , such that

$$q(\theta, s) = \mathbf{y}' \mathbf{V} \mathbf{y}$$

where  $\mathbf{V}$  is positive semi-definite matrix.

To obtain the tightest such underestimator we solve the following Semi-Definite Programming (SDP) problem

$$\begin{aligned} \min \quad & \sum_{m=1}^M z_m \\ \text{s.t.} \quad & z_m = f(\theta_m) - \sum_{\ell=0}^L h_\ell \theta_m^\ell, \quad m = 1, \dots, M, \\ & s^2 (\sum_{\ell=2}^L h_\ell \ell(\ell-1) \theta^{\ell-2}) = \mathbf{y}' \mathbf{V} \mathbf{y}, \\ & \mathbf{V} \succeq \mathbf{0}, \\ & z_m \geq 0, \quad m = 1, \dots, M, \end{aligned} \tag{10}$$

---

optimization problem very similar to problem (8), and our proposed simulated annealing based approach can be used to solve that problem with minimal modifications.

<sup>2</sup>It is possible that the gradient method converges to a local minimum  $\theta_m$  whose corresponding shortest path is different than  $\mathbf{p}_m$ ; then, we reject it and repeat the procedure with a new starting point  $\hat{\theta}_m$  until we find a local minimum whose corresponding shortest path is the same as the one obtained at  $\hat{\theta}_m$ .

with decisions variables  $z_1, \dots, z_M$ , the elements of  $\mathbf{V}$ , and  $h_0, \dots, h_L$ . In particular, (10) minimizes the L1 norm  $\|(f(\theta_1), \dots, f(\theta_M)) - (u(\theta_1), \dots, u(\theta_M))\|$  and ensures that  $u(\cdot)$  is convex and underestimates  $f(\cdot)$  at  $\theta_1, \dots, \theta_M$ .

Once we obtain the underestimator  $u$ , we use it to bias our search for the global minimum of  $f(\cdot)$ . In particular, we obtain new samples of local minima such that areas close to the underestimator's minimum have higher probability of being sampled while areas far away are removed from the search region. This reflects the reasoning that the underestimator captures the shape of  $f(\cdot)$  in the vicinity of the global minimum. We form a new underestimator based on the new set of local minima and so on and so forth. In the interest of space we omit the details as they closely follow the evolution of the SDU algorithm in [Paschalidis et al. 2007].

One can show that SDPU exhibits a similar behavior with SDU. In particular, it is not guaranteed to converge to the global minimum of  $f(\theta)$ . However, in [Paschalidis et al. 2007] we provide probabilistic guarantees for funnel-like functions. These results apply in the funnel around the global minimum of  $f(\theta)$ . As confirmed with numerical results later on, SDPU is quite effective for the problem of interest in this paper. However, we note that this algorithm is essentially a centralized approach which imposes significant communication overhead in actual implementations. In the following subsection, we present another algorithm that can be implemented in a distributed manner.

## 5.2 A Simulated Annealing Based Approach

We focus on the formulation (8) and propose an algorithm for locally changing path configurations to obtain the optimal solution. To proceed, we define the function

$$v(\mathbf{p}) = \mathbf{E}(T_{\mathbf{p}}) + \beta \min_{\theta \geq 0} (\Lambda_{\mathbf{p}}(\theta) - \theta d) \quad (11)$$

for path  $\mathbf{p}$ . The algorithm is described in Fig. 2, where with a slight abuse of notation we denote by  $T_{ij}$  the energy consumption for transmitting a packet from node  $i$  to  $j$ .

- 
- (1) Initialization: path  $\mathbf{p}$  from node 1 to the gateway with value  $v(\mathbf{p})$ , (high enough) temperature  $\eta > 0$ , constant  $0 < \alpha < 1$ , iteration counter  $m := 0$ , constant  $M > 0$ .
  - (2) Change the path configuration:
    - (a) For each link  $k \in \mathbf{p}$ , select it w.p. 1/2 independently of all other links.
    - (b) For each pair of nodes  $i$  and  $j$  on path  $\mathbf{p}$  which are the end points of a chain of selected links
      - i. w.p. 1/2 connect  $i$  and  $j$  directly;
      - ii. w.p. 1/2 look for a relay node  $r$  (not in  $\mathbf{p}$  and not selected as a relay node so far) to connect  $i$  and  $j$ .  $r$  is chosen w.p. proportional to  $\exp(-\mathbf{E}(T_{ir}) - \mathbf{E}(T_{rj}))$ .
  - (3) Let  $\mathbf{q}$  the modified path formed from  $\mathbf{p}$  at Step 2. Accept  $\mathbf{q}$  w.p.  $\min\{1, \exp(\frac{v(\mathbf{p}) - v(\mathbf{q})}{\eta})\}$ . If  $\mathbf{q}$  is accepted set  $\mathbf{p} := \mathbf{q}$ ,  $\eta = \alpha\eta$ .
  - (4)  $m := m + 1$ . If  $m < M$  goto Step 2; otherwise stop, output  $\mathbf{p}$ .
- 

Fig. 2. The simulated annealing-based distributed algorithm.

A few comments on this algorithm are in order. First, for each  $\mathbf{p}$ ,  $v(\mathbf{p})$  can be evaluated efficiently and in a distributed fashion. To that end, note that  $\Lambda_{\mathbf{p}}(\theta)$  is

strictly convex in  $\theta$ , hence, the corresponding optimization problem can be solved efficiently. Moreover, to calculate  $v(\mathbf{p})$  only the nodes in  $\mathbf{p}$  need to exchange information (for instance, each node can be forwarding the quantities  $\mathbf{E}(T_k)$  and  $\Lambda_k(\theta)$  for its outgoing link to the gateway and the gateway can be iterating over  $\theta$  using a gradient method). Second, to change the path configuration, each link in  $\mathbf{p}$  decides independently if it is chosen according to a simple criterion. Further, all the information needed to modify a path is local, and the conditions to perform each operation are relatively straightforward. Third, in Step 3 we decide to accept or reject the new path  $\mathbf{q}$  according to the *Metropolis criterion* (see [Aarts and Korst 1988]). Moreover, as  $v(\mathbf{p})$  and  $v(\mathbf{q})$  can be obtained in a distributed fashion and the temperature parameter  $\eta$  can be stored and updated only at the source node, this operation can also be done without any global information.

As a further remark, note that the 2nd term in the right hand side of (11) is a surrogate for  $\beta \log \mathbf{P}(L_{\mathbf{p}} \geq d)$ . When  $d$  is large enough we can approximate  $\log \mathbf{P}(L_{\mathbf{p}} \geq d)$  using the result of Thm. 4.5. Towards a more accurate approximation for smaller values of  $d$  we postulate  $\mathbf{P}(L_{\mathbf{p}} \geq d) = \zeta e^{d\lambda_{\mathbf{p}}}$  for some constant  $\zeta$  and  $\lambda_{\mathbf{p}} \triangleq \max\{\lambda_k \mid k \in \mathbf{p}\}$ . Solving  $\int_0^\infty \mathbf{P}(L_{\mathbf{p}} \geq \ell) d\ell = \mathbf{E}(L_{\mathbf{p}})$  for  $\zeta$  yields  $\zeta = -\lambda_{\mathbf{p}} \mathbf{E}(L_{\mathbf{p}})$ . Using this large deviations approximation for  $\log \mathbf{P}(L_{\mathbf{p}} \geq d)$  we can define the following alternative to the scoring function in (11):

$$v(\mathbf{p}) = \mathbf{E}(T_{\mathbf{p}}) + \beta[\log(-\lambda_{\mathbf{p}} \mathbf{E}(L_{\mathbf{p}})) + d\lambda_{\mathbf{p}}]. \quad (12)$$

The above is easier to evaluate than (11) as we do not have to solve an optimization problem for each path  $\mathbf{p}$ . The algorithm in Fig. 2 can still be applied without any modification. The large deviations exponent used in (12) provides an asymptotically tight characterization of the latency probability.

## 6. OPTIMIZATION OVER THE DUTY CYCLE

Till now we have focused on the energy consumption for transmitting a data packet from the source nodes to the gateway. In practice, nodes may wake up and then sleep repeatedly between packet generation epochs, and nodes also consume energy when they are ON even if they are not transmitting. In this section we show that our proposed framework can be extended to accommodate these factors as well.

Suppose the data generation process starts at time zero.<sup>3</sup> Denote by  $Z_m$  the (random) time interval between the  $m - 1$ st and  $m$ th packet generation epochs at node 1 for  $m \geq 1$ . We make the following assumptions:

(1)  $Z_1, Z_2, \dots$  are identical and independently distributed with mean  $1/\mu$ , thus, the process  $\{Z_1, Z_2, \dots\}$  is a renewal process with rate  $\mu$ .

(2) The power consumption for node  $i$  is  $\kappa_i$  when it is ON and not transmitting any packets,  $i = 1, \dots, N$ .

(3) Let  $Z$  be the random variable that has the same distribution as  $Z_m$ ,  $\forall m$ , and consistent with the “few data at a low rate” assumption we made in Sec. 2, we adopt the assumption that  $Z \geq L_{\mathbf{p}}$  with very high probability, i.e.,  $\mathbf{P}(Z \geq L_{\mathbf{p}}) \approx 1$ .

With these assumptions we can see that for each link  $k \in \mathbf{p}$  the expected energy

<sup>3</sup>Note that our result holds regardless of the starting time of the data generation process, and this assumption is just for our convenience.

consumption  $\mathbf{E}(\hat{T}_k)$  during a data generation renewal cycle is given by

$$\begin{aligned}\mathbf{E}(\hat{T}_k) &= \mathbf{E}(T_k) + \mathbf{E}(Z - L_k)\rho_i^{(1)}\kappa_i \\ &= \tilde{g}_k + \frac{\rho_i^{(1)}\kappa_i}{\mu} + (c_k - \rho_i^{(1)}\kappa_i)\mathbf{E}(L_k),\end{aligned}$$

where  $i = s(k)$ , and the total energy consumption on path  $\mathbf{p}$  is given by  $\mathbf{E}(\hat{T}_{\mathbf{p}}) = \sum_{k \in \mathbf{p}} \mathbf{E}(\hat{T}_k)$ .

Denoting by  $\hat{T}_{\mathbf{p}}(t)$  the accumulated energy consumption on path  $\mathbf{p}$  from time 0 to  $t$ , the long term average power consumption  $\alpha_{\mathbf{p}}$  on path  $\mathbf{p}$  is given by

$$\begin{aligned}\alpha_{\mathbf{p}} &= \lim_{t \rightarrow \infty} \frac{\hat{T}_{\mathbf{p}}(t)}{t} = \frac{\mathbf{E}(\hat{T}_{\mathbf{p}})}{\mathbf{E}(Z)} \\ &= \sum_{k \in \mathbf{p}} \left( \mu \tilde{g}_k + \rho_i^{(1)}\kappa_i + \mu(c_k - \rho_i^{(1)}\kappa_i)\mathbf{E}(L_k) \right),\end{aligned}$$

where the second equality holds with probability 1 due to the elementary renewal reward theorem.

Suppose we fix  $b_i^{10}$  and have control on the duty cycle  $\rho_i^{(1)}$  for any node  $i = 1, \dots, N$ . To balance the energy consumption and the latency probability, we can jointly optimize routing and tune the duty cycle of each node. To this end, we propose the following optimization problem

$$\begin{aligned}\min_{\mathbf{p} \in \mathcal{P}} \quad & \alpha_{\mathbf{p}} + \beta[\log(-\lambda_{\mathbf{p}}\mathbf{E}(L_{\mathbf{p}})) + d\lambda_{\mathbf{p}}] \\ \text{s.t.} \quad & 0 \leq \rho_i^{(1)} \leq 1 \quad \forall i = 1, \dots, N,\end{aligned}\tag{13}$$

where the second term in the objective function approximates  $\beta \log \mathbf{P}(L_{\mathbf{p}} \geq d)$ , and  $\lambda_{\mathbf{p}} = \max\{\lambda_k \mid k \in \mathbf{p}\}$  (see Sec. 5).

Problem (13) is difficult to solve due to two sources of complexity: the first one is the combinatorial complexity to find the optimal routing; the second is the nonconvexity of the objective function with respect to  $\rho_i^{(1)}$ ,  $i = 1, \dots, N$ , for any given path  $\mathbf{p}$ . More precisely, the objective function of problem (13) is a continuous, but typically nonconvex and nondifferentiable function of  $\rho_i^{(1)}$ ,  $i = 1, \dots, N$ . Nonetheless, the proposed simulated annealing based algorithm described in Fig. 2 can be extended to solve problem (13). To that end, we can use a new scoring function set as

$$v(\mathbf{p}) = \min_{\{0 \leq \rho_i^{(1)} \leq 1, \forall i\}} \alpha_{\mathbf{p}} + \beta[\log(-\lambda_{\mathbf{p}}\mathbf{E}(L_{\mathbf{p}})) + d\lambda_{\mathbf{p}}].$$

However, it is not guaranteed that the algorithm will find the global minimum of problem (13).

## 7. NUMERICAL RESULTS

Next, we provide numerical results to illustrate the performance of our algorithms. In all the numerical examples we use two-state Gilbert-Elliot model for the channel condition for all  $k \in \mathcal{E}$ . More specifically, when  $X_k = 0$  the channel is “bad” with probability 1; when  $X_k = 1$  the channel is “good” with probability 1. We randomly generate a network of 50 nodes uniformly distributed in the box  $[0, 100m] \times [0, 100m]$  except that node 1 is always at  $(100m, 100m)$ . The gateway is located at  $(0, 0)$ . For each link  $k \in \mathcal{E}$ , we assume that  $c_k$  is 1 mWatt, and  $g_k^{(1)}$  is  $2 \times 10^{-3} \chi_k^2$  Joules,

where  $\chi_k$  is the distance between nodes  $s(k)$  and  $d(k)$ . We randomly generate the transition matrices  $A_k$  and  $B_i$  for all  $k \in \mathcal{E}$ ,  $i \in \mathcal{V}$ , and calculate the stationary distributions accordingly. We set  $\beta = 0.5$  and  $d = 20$  seconds in (8). We are interested in finding a path from node 1 to the gateway that is an optimal solution to problem (8). We generate 300 instances of the network and use both our centralized and distributed approach to solve each instance. We compare the results to a benchmark optimal value  $f_{\min}$  of problem (9) (and equivalently, problem (8)) that is obtained by exhaustive search. All the programs were run on a computer with a CPU running at 3.06 GHz and 3.6 Gbytes of main memory; background processes were minimal.

### 7.1 Semi-Definite programming-based convex Polynomial Underestimation (SDPU)

We use a polynomial underestimator function of degree  $L = 4$ , thus,  $\ddot{u}(\theta) = 2h_2 + 6h_3\theta + 12h_4\theta^2$ . We choose the vector of monomials  $\mathbf{y} = (s, \theta, \theta s)$ . Let  $f^*$  be the objective value obtained by SDPU. For all the 300 instances,  $f^*$  is equal to (or slightly less than, due to discretization) the benchmark value  $f_{\min}$ .

### 7.2 Simulated annealing-based approach

We apply the algorithm in Fig. 2 with the scoring function (11) to solve the same 300 cases as in (1), and obtain the corresponding objective value  $f^*$ . The mean solution quality (computed as  $\frac{f^* - f_{\min}}{|f_{\min}|} \times 100\%$ ) of the distributed algorithm is 2.8% and the standard deviation is 8.8%. Table I provides more detailed statistics, reporting the percentage of the instances we solved within solution quality levels of 5%, 10%, 15% and 20%. It is evident that although the centralized approach

Table I. Quality of the simulated annealing solutions with scoring function (11).

$\frac{f^* - f_{\min}}{ f_{\min} }$	5%	10%	15%	20%
% of cases	87.0%	88.0%	92.7%	95.0%

has better performance in terms of the solution quality, the performance of the simulated annealing-based algorithm is also quite satisfactory. In other words, for a modest performance cost we significantly gain in implementation complexity and save energy by distributing the calculations and minimizing the communication overhead.

### 7.3 Simulated annealing with a large deviations-based scoring function

Next we investigate efficiency gains from using the large deviations-based scoring function in (12). We maintain the network setup and randomly generate 100 instances. For each instance we run the simulated annealing approach of Fig. 2 with two different scoring functions: (i) the scoring function (11), and (ii) the large deviations-based scoring function in (12). Letting  $\mathbf{p}_1$ ,  $\mathbf{p}_2$  be the optimal paths obtained by the two approaches, respectively, we evaluate the quality of  $\mathbf{p}_1$  and  $\mathbf{p}_2$  by computing  $w(\mathbf{p}_1)$  and  $w(\mathbf{p}_2)$  where  $w(\mathbf{p}) \triangleq \mathbf{E}(T_{\mathbf{p}}) + \beta \log \mathbf{P}(L_{\mathbf{p}} \geq d)$ , and the

value of  $\mathbf{P}(L_{\mathbf{p}} \geq d)$  is obtained by convolution.<sup>4</sup> Moreover, for each instance we calculate  $\xi = \frac{|w(\mathbf{p}_1) - w(\mathbf{p}_2)|}{|w(\mathbf{p}_1)|}$ .

For 100 instances we randomly generated, the average running times of the algorithm with scoring functions (11) and (12) are 448.1 seconds and 32.4 seconds, respectively, and the average value of  $\xi$  is 2.7% with standard deviation 3.4%. It follows that scoring function (12) offers a significant efficiency gain (by a factor of 10), due to minimal communication overhead and a much simplified computation, without any loss in performance. Thus, for relatively large  $d$  (equivalently, small latency probabilities) it should be the method of choice.

#### 7.4 Optimization over the duty cycles

To investigate the benefit of optimizing the duty cycles, we set  $\kappa_i = 1$  mWatt and  $b_i^{10} = 1$  for  $i = 1, \dots, 50$ , use the same network setup and randomly generate 100 instances. For each instance, we first run the simulated annealing algorithm with the objective function in (13) with fixed duty cycle  $\rho_i^{(1)} = 0.2, \forall i = 1, \dots, 50$ , and obtain the optimal path  $\mathbf{p}_1$ . Next we run the simulated annealing algorithm with optimization over the duty cycles (for convenience we let every sensor have the same duty cycle), and obtain a path  $\mathbf{p}_2$  and a duty cycle  $\rho$ .

We evaluate the quality of  $\mathbf{p}_1$  and  $\mathbf{p}_2$ , using function  $w(\mathbf{p})$  set as in the previous example, and calculate the improvement  $\xi = \frac{w(\mathbf{p}_1) - w(\mathbf{p}_2)}{|w(\mathbf{p}_1)|}$ . Our numerical results show that in 92 out of 100 instances we have  $w(\mathbf{p}_2) \leq w(\mathbf{p}_1)$ , and the average value of  $\xi$  is 16.5%, which represents a significant performance gain.

## 8. CONCLUSION

We considered routing in SNETs in the presence of varying channel conditions and non-synchronized non-periodic sleeping schedules of sensor nodes. Under Markovian assumptions, we characterized the expected energy consumption and the latency probability for each path employed in delivering data from the sensors to the gateway. The latency probability is hard to compute exactly which led us to develop a large deviations asymptotic. Specifically, we derived an upper bound and a matching lower bound on the latency probability that are asymptotically tight as the delay parameter  $d$  grows large. The result also identifies a “bottleneck” link.

To investigate the trade-off between energy and latency, we focused on solving the problem of minimizing a weighted sum of the expected energy consumption and the exponent of the latency probability (or its approximation). The problem is non-convex, thus, challenging. We proposed two classes of algorithms: a centralized algorithm based on convex polynomial underestimators, and a simulated annealing technique that can be implemented in a distributed fashion. We also illustrated how our framework can be extended for joint routing and duty cycle optimization to achieve better performance.

In our numerical experiments the centralized approach was able to always find the global minimum. The performance of the distributed approach is also quite satisfactory and requires much less communication overhead, which is an attractive

<sup>4</sup>Note that it is not practical to apply the simulated annealing-based algorithm to scoring function  $w(\mathbf{p})$ , because evaluating  $\mathbf{P}(L_{\mathbf{p}} \geq d)$  is computationally expensive when path  $\mathbf{p}$  is long.

feature in actual SNET implementations.

## APPENDIX

### A. PROOF OF LEMMA 3.4

PROOF. We will first give the proof for  $k \neq \sigma$ . Let  $\mathbf{D} = \text{diag}(\boldsymbol{\delta}_k)$  be the matrix with the diagonal elements being the stationary probabilities of  $(X_k, Y_j, O_k)$  at states in  $\tilde{\mathcal{S}}_k$ . Since  $(X_k^t, Y_j^t, O_k^t)$  is time reversible (cf. Lemma 3.2),  $\mathbf{D}\mathbf{H}_k$  is a symmetric matrix. Therefore

$$\mathbf{D}\mathbf{H}_k = (\mathbf{D}\mathbf{H}_k)' = \mathbf{H}'_k\mathbf{D}' = \mathbf{H}'_k\mathbf{D}.$$

Define  $\mathbf{D}^{1/2}$  as the matrix with the diagonal elements being the square roots of the corresponding elements of  $\mathbf{D}$ , and  $\mathbf{D}^{-1/2} = (\mathbf{D}^{1/2})^{-1}$ . Note that  $\mathbf{D}^{1/2}\mathbf{D}^{1/2} = \mathbf{D}$ , and that  $\mathbf{D}$ ,  $\mathbf{D}^{1/2}$  and  $\mathbf{D}^{-1/2}$  are all invertible and diagonal matrices. We have

$$\begin{aligned} \mathbf{D}\mathbf{H}_k\mathbf{D}^{-1/2} &= \mathbf{H}'_k\mathbf{D}\mathbf{D}^{-1/2} = \mathbf{H}'_k\mathbf{D}^{1/2} \\ \Rightarrow \mathbf{D}^{-1/2}\mathbf{D}\mathbf{H}_k\mathbf{D}^{-1/2} &= \mathbf{D}^{-1/2}\mathbf{H}'_k\mathbf{D}^{1/2} \\ \Rightarrow \mathbf{D}^{1/2}\mathbf{H}_k\mathbf{D}^{-1/2} &= \mathbf{D}^{-1/2}\mathbf{H}'_k\mathbf{D}^{1/2} \\ \Rightarrow \mathbf{D}^{1/2}\mathbf{H}_k\mathbf{D}^{-1/2} &= (\mathbf{D}^{1/2}\mathbf{H}_k\mathbf{D}^{-1/2})' \end{aligned}$$

which implies that  $\mathbf{D}^{1/2}\mathbf{H}_k\mathbf{D}^{-1/2}$  is also a symmetric matrix and all its eigenvalues should be real. Since  $\mathbf{D}^{1/2}\mathbf{H}_k\mathbf{D}^{-1/2}$  and  $\mathbf{H}_k$  are similar and have the same set of eigenvalues, all the eigenvalues of  $\mathbf{H}_k$  are also real.<sup>5</sup>

Next we show that the eigenvalues of  $\mathbf{H}_k$  are negative. Since  $\mathbf{A}_k$  is irreducible and  $b_j^{01}, b_j^{10} > 0$ , it can be seen that  $\mathbf{H}_k$  is an irreducible matrix due to Assumption A(*i*). Note that  $\mathbf{H}_k$  is also diagonally dominant. To this end [Horn and Johnson 1990, Theorem 6.2.27] guarantees the negativity of the eigenvalues of  $\mathbf{H}_k$ .

In the case of  $k = \sigma$ , note that we assume  $|\tilde{\mathcal{S}}_k| > 0$  and thus  $\mathbf{H}_\sigma$  is always well defined. The rest of the proof is identical to the case of  $k \neq \sigma$ .  $\square$

### B. PROOF OF LEMMA 3.5

PROOF. Let the matrix  $\tilde{\mathbf{f}}_k(t)$  be defined such that the  $(\phi_0, \phi_1)$  element is given by  $\tilde{f}_{\phi_0\phi_1}(t) = \mathbf{P}(\Phi_k^t = \phi_1 \mid \Phi_k^0 = \phi_0), \forall \phi_0, \phi_1 \in \tilde{\mathcal{S}}_k$ . Thus,

$$\mathbf{f}(t) = \mathbf{e} - \tilde{\mathbf{f}}_k(t)\mathbf{e} = \mathbf{e} - e^{\mathbf{H}_k t}\mathbf{e} = (\mathbf{I} - e^{\mathbf{H}_k t})\mathbf{e},$$

where the second equality is due to the transient analysis of Markov processes with absorbing states [Bhattacharya and Waymire 1990].  $\square$

### C. PROOF OF LEMMA 4.1

PROOF. We will give the proof for the matrix  $e^{(\theta\mathbf{I} + \mathbf{H}_k)\ell}$ , since  $e^{\mathbf{H}_k\ell}$  is just the special case of  $e^{(\theta\mathbf{I} + \mathbf{H}_k)\ell}$  with  $\theta = 0$ .

Note that all the off diagonal elements of  $(\theta\mathbf{I} + \mathbf{H}_k)\ell$  are nonnegative, and that  $\mathbf{H}_k$  is an irreducible matrix (cf. Appendix A). Therefore there exist some nonnegative

<sup>5</sup>Recall the assumption in Sec. 2 that  $\{X_k^t\}$  is time reversible for any  $k \in \mathcal{L}$ . This assumption guarantees that all the eigenvalues of  $\mathbf{H}_k$  are real for any  $k$ , but is not a necessary condition.

and irreducible matrix  $\mathbf{Q}$  and a scalar  $\nu$  such that  $(\theta\mathbf{I} + \mathbf{H}_k)\ell = \nu\mathbf{I} + \mathbf{Q}$ . Since  $\mathbf{Q}$  is irreducible, for any  $\phi_0$  and  $\phi_1$ , there exists some finite  $n$  such that  $(\mathbf{Q}^n)_{\phi_0\phi_1} > 0$ . Thus  $e^{\mathbf{Q}} = \sum_{n=0}^{\infty} \frac{\mathbf{Q}^n}{n!} > \mathbf{0}$ . As a result

$$e^{(\theta\mathbf{I} + \mathbf{H}_k)\ell} = e^{\nu\mathbf{I}}e^{\mathbf{Q}} = e^{\nu}e^{\mathbf{Q}} > \mathbf{0}.$$

□

#### D. PROOF OF LEMMA 4.2

PROOF. Note that  $(\theta + \lambda_k)\ell$  is the maximum eigenvalue of  $(\theta\mathbf{I} + \mathbf{H}_k)\ell$ , and that  $e^{(\theta + \lambda_k)\ell}$  is the Perron-Frobenius (P-F) eigenvalue of the positive matrix  $e^{(\theta\mathbf{I} + \mathbf{H}_k)\ell}$  (cf. Lemma 4.1). Let  $\mathbf{v} > \mathbf{0}$  be the left eigenvector of  $\mathbf{H}_k$  corresponding to the eigenvalue  $\lambda_k$ , and we can see that  $\mathbf{v}$  is also the left P-F eigenvector of  $e^{(\theta\mathbf{I} + \mathbf{H}_k)\ell}$  corresponding to eigenvalue  $e^{(\theta + \lambda_k)\ell}$ , and  $v^{\max} \geq v^{\min} > 0$  by the Perron-Frobenius theorem. Note that  $\delta_k^{\max} \geq \delta_k^{\min} > 0$  due to the fact  $\delta_k > \mathbf{0}$ , thus when  $\theta < -\lambda_k$  we have

$$\begin{aligned} \Lambda_k(\theta) &= \log(\gamma'_k \mathbf{e} + \int_0^{\infty} \delta'_k e^{(\theta\mathbf{I} + \mathbf{H}_k)\ell} \mathbf{r}_k d\ell) \\ &\leq \log(\gamma'_k \mathbf{e} + \int_0^{\infty} \frac{\delta_k^{\max}}{v^{\min}} \mathbf{v}' e^{(\theta\mathbf{I} + \mathbf{H}_k)\ell} \mathbf{r}_k d\ell) \\ &= \log(\gamma'_k \mathbf{e} + \frac{\delta_k^{\max}}{v^{\min}} \int_0^{\infty} \mathbf{v}' e^{(\theta + \lambda_k)\ell} \mathbf{r}_k d\ell) \\ &= \log(\gamma'_k \mathbf{e} - \frac{\delta_k^{\max} \mathbf{v}' \mathbf{r}_k}{v^{\min}(\theta + \lambda_k)}) < \infty. \end{aligned}$$

On the other hand, when  $\theta \geq -\lambda_k$ , we can see that

$$\begin{aligned} \Lambda_k(\theta) &\geq \log(\gamma'_k \mathbf{e} + \int_0^{\infty} \frac{\delta_k^{\min}}{v^{\max}} \mathbf{v}' e^{(\theta\mathbf{I} + \mathbf{H}_k)\ell} \mathbf{r}_k d\ell) \\ &= \log(\gamma'_k \mathbf{e} + \frac{\delta_k^{\min}}{v^{\max}} \int_0^{\infty} \mathbf{v}' e^{(\theta + \lambda_k)\ell} \mathbf{r}_k d\ell) = \infty \end{aligned}$$

since  $\int_0^{\infty} e^{(\theta + \lambda_k)\ell} d\ell$  is unbounded, and  $\mathbf{v}' \mathbf{r}_k > 0$ . □

#### REFERENCES

- AARTS, E. AND KORST, J. 1988. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Wiley-Interscience Series.
- BERTSEKAS, D. 1999. *Nonlinear Programming*, 2nd ed. Athena Scientific, Belmont, MA.
- BERTSEKAS, D. P. AND TSITSIKLIS, J. N. 1989. *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall. republished by Athena Scientific, 1997.
- BHATTACHARYA, R. N. AND WAYMIRE, E. C. 1990. *Stochastic Processes with Applications*. Wiley Series in Probability and Mathematical Statistics.
- CHANG, J.-H. AND TASSIULAS, L. 2004. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Transactions on Networking* 12, 4, 609–619.
- ELLIOT, E. O. 1963. Estimates of error rates for codes on burst-noise channels. *Bell Systems Technology Journal* 42, 1977–1997.
- ELSON, J., GIROD, L., AND ESTRIN, D. 2002. Fine-grained network time synchronization using reference broadcasts. *SIGOPS Oper. Syst. Rev.* 36, SI, 147–163.
- ERGEN, S. C. AND VARAIYA, P. 2006. PEDAMACS: Power efficient and delay aware medium access protocol for sensor networks. *IEEE Transactions on Mobile Computing* 5, 7 (July).
- FUEMMELE, J. AND VEERAVALLI, V. V. 2007. Smart sleeping policies for energy efficient tracking in sensor networks. *IEEE Transactions on Signal Processing*. to appear.
- GAREY, M. R. AND JOHNSON, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman and Company, New York.

- GILBERT, E. N. 1960. Capacity of a burst-noise channel. *Bell Systems Technology Journal* 39, 1253–1266.
- HEINZELMAN, W., CHANDRAKASAN, A., AND BALAKRISHNAN, H. 2002. Energy-efficient communication protocols for wireless microsensor networks. In *Proceedings of the 33rd Hawaiian International Conference on Systems Science (HICSS-33)*. Maui, HI, 3005–3014.
- HORN, R. A. AND JOHNSON, C. R. 1990. *Matrix Analysis*. Cambridge University Press.
- KONRAD, A., ZHAO, B. Y., JOSEPH, A. D., AND LUDWIG, R. 2001. A Markov-based channel model algorithm for wireless networks. In *MSWIM '01: Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*. ACM Press, New York, NY, USA, 28–36.
- LU, G., SADAGOPAN, N., KRISHNAMACHARI, B., AND GOEL, A. 2005. Delay efficient sleep scheduling in wireless sensor networks. In *Proceedings of INFOCOM*. Vol. 14. 2470–2481.
- MAGNANI, A., LALL, S., AND BOYD, S. 2005. Tractable fitting with convex polynomials via sum-of-squares. *Proceedings of 43rd Conference on Decision and Control*, 1672–1677.
- MAINWARING, A., CULLER, D., POLASTRE, J., SZEWCZYK, R., AND ANDERSON, J. 2002. Wireless sensor networks for habitat monitoring. *Proc. 1st ACM Intl. Workshop on Wireless Sensor Networks and Applications (WSNA02)*, 88–97.
- MIAO, L. AND CASSANDRAS, C. G. 2006. Optimal transmission scheduling for energy-efficient wireless networks. In *Proceedings of INFOCOM*.
- PASCHALIDIS, I. C., SHEN, Y., VAKILI, P., AND VAJDA, S. 2007. SDU: A semi-definite programming-based underestimation method for stochastic global optimization in protein docking. *IEEE Trans. Automat. Contr.* 52, 4, 664–676.
- RAY, S., LAI, W., AND PASCHALIDIS, I. C. 2006. Statistical location detection with sensor networks. *Joint special issue IEEE/ACM Trans. Networking and IEEE Trans. Information Theory* 52, 6, 2670–2683.
- WANG, H. S. AND MOAYERI, N. 1995. Finite-state Markov channel—a useful model for radio communication channels. *IEEE Transactions on Vehicular Technology* 44, 163–171.
- WOOD, A. D. AND STANKOVIC, J. A. 2002. Denial of service in sensor networks. *IEEE Computer* 35, 10 (October).
- YE, W., HEIDEMANN, J., AND ESTRIN, D. 2004. Media access control with coordinated adaptive sleeping in wireless sensor networks. *IEEE/ACM Transactions on Networking* 12, 3, 493–506.
- YU, Y., KRISHNAMACHARI, B., AND PRASANNA, V. K. 2004. Energy-latency tradeoff for data gathering in wireless sensor networks. In *Proceedings of INFOCOM*. Vol. 1. p. 255.
- ZHANG, Q. AND KASSAM, K. 1999. Finite-state Markov model for Rayleigh fading channels. *IEEE Trans. on Communication* 47, 11, 1688–1692.