2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

GODDeS: Globally ϵ-Optimal Routing Via Distributed Decision-theoretic Self-organization

Ishanu Chattopadhyay (ixc128@psu.edu), Asok Ray (axr2@psu.edu)
Department of Mechanical Engineering, The Pennsylvania State University

Abstract— **This paper introduces GODDeS: a fully distributed self-organizing decision-theoretic routing algorithm designed to effectively exploit high quality paths in lossy ad-hoc wireless environments, typically with a large number of nodes. The routing problem is modeled as an optimal control problem for a decentralized Markov Decision Process, with links characterized by locally known packet drop probabilities that either remain constant on average or change slowly. The equivalence of this optimization problem to that of performance maximization of an explicitly constructed probabilistic automata allows us to effectively apply the theory of quantitative measures of probabilistic regular languages, and design a distributed highly efficient solution approach that attempts to minimize source-to-sink drop probabilities across the network. Theoretical results provide rigorous guarantees on global performance, showing that the algorithm achieves near-global optimality, in polynomial time. It is also argued that GODDeS is significantly congestion-aware, and exploits multi-path routes optimally. Theoretical development is supported by high-fidelity network simulations.**

I. INTRODUCTION & MOTIVATION

The routing problem has been widely studied in the context of ad-hoc wireless networks, and reported algorithms can be broadly classified as follows. A routing protocol is pro-active (DBF (e.g. Distributed Bellman-Ford) [1] and DSDV (Highly Dynamic Destination-Sequenced Distance Vector routing) [2]), if fresh destination lists and their routes are maintained by periodically distributing routing tables; it is reactive (e.g. AODV (Ad-hoc On-demand Distance Vector) [3] and DSR (Dynamic Source Routing) [4]) if routes are computed if and when necessary by flooding the network with Route Request packets. Pro-active protocols suffer from expensive route maintenance and slow reaction to topology changes, while reactive methods have high latency in discovery and induce congestion due to periodic flooding. Hybrid protocols attempt to combine advantages of both philosophies e.g. HRPLS (Hybrid Routing Protocol for Large Scale Mobile Ad Hoc Networks with Mobile Backbones) [5] and HSLS (Hazy Sighted Link State routing protocol) [6]. Other approaches use geographic, or power information, and in the context of sensor networks, query based routing strategies (e.g. [7]) have been proposed.

Reported ad hoc routing protocols for wireless networks primarily focus on node mobility, rapidly changing topologies, overhead, and scalability; with little attention paid to finding high-quality paths in the face of lossy wireless links. An implicit assumption is that links either work well or dont work at all; which is not reasonable in the wireless case where many links have intermediate loss ratios. This problem has been partially addressed by designing new quality-aware metrics such as the expected transmission count (ETX) [8], where the authors correctly note "*minimizing hop-count maximizes the distance traveled by each hop, which is likely to minimize signal strength and maximize the loss ratio*". Even if the best route is one with minimal hop-count, there may be many routes (particularly in dense networks) of the same minimum length with widely varying qualities; arbitrary choice made by most minimum hop-count metrics is not likely to select the best. The problem is also crucial in multi-rate networks [9], where the routing protocol must select from the set of available links. While in single-rate networks all links are equivalent, in multi-rate networks each available link may operate at a different rate. Thus the routing protocol faces a complex trade-off: *Long distance links take fewer hops, but the links operate slower; short links can operate at high rates, but more hops are required.*

In this paper, we give a theoretical solution to this potentially large-scale decision problem via formulating a probabilistic routing policy that very nearly minimizes the end-to-end packet drop probabilities. In particular, the routing problem is modeled and solved as an optimal control problem for a Decentralized Markov Decision Process (D-MDP). We begin by assuming that the communication links are imperfect, and are being characterized by locally known drop probabilities. The mean or expected values of the link-specific drop probabilities, and the network topology is assumed to be are either constant or changing over a time scale which is significantly slower compared to that of the communication dynamics. We then seek local routing decisions that maximize throughput in the sense of minimizing the source-to-sink probability of packet-drops. The Markov structure emerges, since we assume that the local link-specific drop probabilities are independent of the history of sequential link traversal by individual packets.

The results developed in this paper effectively resolve the issues described above (actually attaining near global optimality). To the best of the author's knowledge, such an approach has not been previously investigated. The reason for this apparent neglect is as follows: Recent investigations [10], [11] into the solution complexity of decentralized Markov decision processes have shown that the problem is exceptionally hard even for two agents; illustrating a fundamental divide between centralized and decentralized control of MDP. In contrast to the centralized approach, the decentralized case provably does not admit polynomial-time algorithms, suffering super-exponential worst case complexity. Such negative results do not preclude the possibility of obtaining *near-optimal* solutions efficiently. This is what we achieve in this paper, in the context of the routing problem. We show that a highly efficient, fully distributed, decision algorithm can be designed that effectively solves the distributed MDP such that the control policy, on convergence, is within an ϵ bound of the global optimal. Furthermore, one can freely choose the error bound ϵ, with the caveat

978-1-4577-0079-8/11/$26.00 ©2011 AACC

that the convergence time increases with decreasing ϵ.

We call this algorithm GODDeS (Globally ϵ-Optimal Routing Via Distributed Decision-theoretic Self-organization). In place of a standard MDP formulation, we use a representation based on Probabilistic Finite State Automata (PFSA), which allows us to set up the decision problem as that of performance maximization of PFSA, and obtain solutions using the recently reported quantitative measures of probabilistic regular languages [12]. This shift of modeling paradigm allows one to achieve near-global optimality in polynomial time. Theoretical results also establish that GODDeS is highly scalable, optimally take advantage of existing multi-path routes, and is expected to be significantly congestion-aware. For simplicity of exposition, a single sink is considered throughout the paper. This is not a serious restriction, since the results carry over to the general case with ease. The resulting algorithm is both pro-active and reactive, but not in the usual sense of reported hybrid protocols. It uses both distance-vector (in a generalized sense via the language-measure construction) and link-state information, and uses local multi-cast to forward messages; optimally taking advantage of multi-path routing.

The rest of the paper is organized in four sections. Section II summarizes the theory of quantitative measures of probabilistic regular languages, and the approach to centralized performance maximization of PFSA. Section III develops the PFSA model of an ad-hoc network, and Section IV presents the key theoretical development for decentralized PFSA optimization. The paper is summarized and concluded in Section V with recommendations for future work.

II. Background: Language Measure Theory

This section summarizes the concept of signed real measure of probabilistic regular languages, and its application in performance optimization of probabilistic finite state automata (PFSA) [12]. A string over an alphabet (i.e. a non-empty finite set) Σ is a finite-length sequence of symbols from Σ [13]. The Kleene closure of Σ, denoted by Σ^*, is the set of all finite-length strings of symbols including the null string ϵ. The string xy is the concatenation of strings x and y, and the null string ϵ is the identity element of the concatenative monoid.

Definition 1 (PFSA): A PFSA G over an alphabet Σ is a sextuple $(Q, \Sigma, \delta, \widetilde{\Pi}, \chi, \mathscr{C})$, where Q is a set of states, $\delta : Q \times \Sigma^\star \to Q$ is the (possibly partial) transition map; $\widetilde{\Pi} : Q \times \Sigma \to [0, 1]$ is an output mapping, known as the probability morph function that specifies the state-specific symbol generation probabilities and satisfies $\forall q_i \in Q, \sigma \in \Sigma, \widetilde{\Pi}(q_i, \sigma) \geqq 0$, and $\sum_{\sigma \in \Sigma} \widetilde{\Pi}(q_i, \sigma) = 1$, the state characteristic function $\chi : Q \to [-1, 1]$ assigns a signed real weight to each state, and $\mathscr{C}$ is the set of controllable transitions that can be disabled (Definition 2).

Definition 2 (Control Philosophy): If $\delta(q_i, \sigma) = q_k$, then the *disabling* of σ at q_i prevents the state transition from q_i to q_k. Thus, disabling a transition σ at a state q replaces the original transition with a self-loop with identical occurrence probability, i.e. we now have $\delta(q_i, \sigma) = q_i$. Transitions that can be so disabled are *controllable*, and belong to the set $\mathscr{C}$.

Definition 3: The language $L(q_i)$ generated by a PFSA G initialized at the state $q_i \in Q$ is defined as: $L(q_i) = \{s \in \Sigma^* \mid \delta(q_i, s) \in Q\}$ Similarly, for every $q_j \in Q$, $L(q_i, q_j)$ denotes the set of all strings that, starting from the state q_i, terminate at the state q_j, i.e., $L(q_i, q_j) = \{s \in \Sigma^* \mid \delta(q_i, s) = q_j \in Q\}$

Definition 4 (State Transition Matrix): The state transition probability matrix $\Pi \in [0, 1]^{\textsc{Card}(Q) \times \textsc{Card}(Q)}$, for a given PFSA is defined as: $\forall q_i, q_j \in Q, \Pi_{ij} = \sum_{\sigma \in \Sigma \text{ s.t. } \delta(q_i, \sigma) = q_j} \widetilde{\Pi}(\sigma, q_i)$ Note that Π is a square non-negative stochastic matrix [14], where Π_{ij} is the probability of transitioning from q_i to q_j.

Notation 1: We use matrix notations interchangeably for the morph function $\widetilde{\Pi}$. In particular, $\widetilde{\Pi}_{ij} = \widetilde{\Pi}(q_i, \sigma_j)$ with $q_i \in Q, \sigma_j \in \Sigma$. Note that $\widetilde{\Pi} \in [0, 1]^{\textsc{Card}(Q) \times \textsc{Card}(\Sigma)}$ is not necessarily square, but each row sums up to unity.

A signed real measure [15] $\nu^i : 2^{L(q_i)} \to \mathbb{R} \equiv (-\infty, +\infty)$ is constructed on the σ-algebra $2^{L(q_i)}$ [12], implying that every singleton string set $\{s \in L(q_i)\}$ is a measurable set.

Definition 5 (Language Measure): Let $\omega \in L(q_i, q_j) \subseteq 2^{L(q_i)}$. The signed real measure ν^i_θ of every singleton string set $\{\omega\}$ is defined as: $\nu^i_\theta(\{\omega\}) \triangleq \theta(1 - \theta)^{|\omega|} \widetilde{\Pi}(q_i, \omega) \chi(q_j)$. For every choice of the parameter $\theta \in (0, 1)$, the signed real measure of a sublanguage $L(q_i, q_j) \subseteq L(q_i)$ is defined as: $\nu^i_\theta(L(q_i, q_j)) \triangleq \sum_{\omega \in L(q_i, q_j)} \theta(1 - \theta)^{|\omega|} \widetilde{\Pi}(q_i, \omega) \chi_j$. Similarly, the measure of $L(q_i)$, is defined as $\nu^i_\theta(L(q_i)) \triangleq \sum_{q_j \in Q} \nu^i_\theta(L_{i,j})$.

Notation 2: For a given PFSA, we interpret the set of measures $\nu^i_\theta(L(q_i))$ as a real-valued vector of length $\textsc{Card}(Q)$ and denote $\nu^i_\theta(L(q_i))$ as $\nu_\theta|_i$.

The language measure can be expressed vectorially:

$$\nu_\theta = \theta \left[\mathbb{I} - (1 - \theta) \Pi \right]^{-1} \chi \tag{1}$$

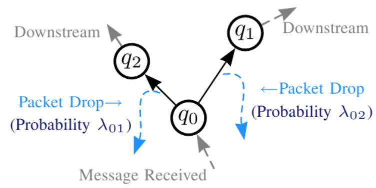
The inverse exists for $\theta \in (0, 1]$ [12].

Remark 1 (Physical Interpretation): In the limit of $\theta \to 0^+$, the language measure of singleton strings can be interpreted to be product of the conditional generation probability of the string, and the characteristic weight on the terminating state. Hence, smaller the characteristic, or smaller the probability of generating the string, smaller is its measure. Thus, if the characteristic values represent the control specification, with more positive weights given to desirable states, then the measure represents how *good* the particular string is w.r.t the given specification, and the given model. The limiting language measure $\nu_0|_i = \lim_{\theta \to 0^+} \theta \left[\mathbb{I} - (1 - \theta) \Pi \right]^{-1} \chi \big|_i$ sums up the limiting measures of each string starting from q_i, and thus captures how *good* q_i is, based on not only its own characteristic, but on how *good* are the strings generated in future from q_i. It is thus a quantification of the impact of q_i on future dynamics [12].

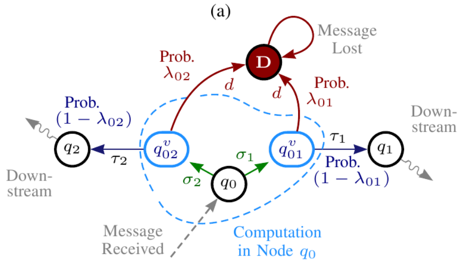
Definition 6 (Supervisor): A supervisor disables a subset of the set $\mathscr{C}$ of controllable transitions and hence there is a bijection between the set of all possible supervision policies and the power set $2^{\mathscr{C}}$.

Language measure allows a quantitative comparison of supervision policies.

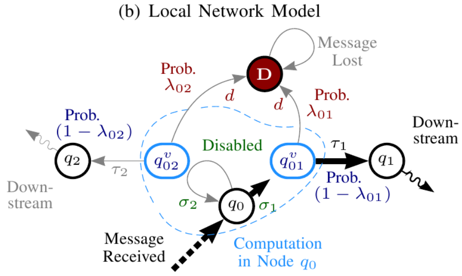
Definition 7 (Optimal Supervision Problem): Given a PFSA $G = (Q, \Sigma, \delta, \widetilde{\Pi}, \chi, \mathscr{C})$, compute a supervisor disabling $\mathscr{D}^\star \subseteq \mathscr{C}$, s.t. $\nu^\star_0 \geqq_{\texttt{(Elementwise)}} \nu^\dagger_0 \quad \forall \mathscr{D}^\dagger \subseteq \mathscr{C}$ where $\nu^\star_0, \nu^\dagger_0$ are the limiting measure vectors of supervised plants $G^\star$, $G^\dagger$ under $\mathscr{D}^\star$, $\mathscr{D}^\dagger$ respectively.



(a)



(b) Local Network Model



(c) Controlled Local Model

Fig. 1: Node centric decision for packet forwarding with non-zero drop probability for all choices

Remark 2: The solution to the optimal supervision problem is obtained in [12] by designing an optimal policy using ν_θ with $\theta \in (0,1)$. To ensure that the computed optimal policy coincides with the one for $\theta \to 0^+$, the authors choose a *small, but non-zero* value for θ in each iteration step of the design algorithm. To address numerical issues, algorithms reported in [12] computes how small a θ is actually required, i.e., computes the critical lower bound $\theta_\star$. Moreover the solution obtained is optimal, unique, efficiently computable, and maximally permissive among policies with maximal performance.

Language-measure-theoretic optimization is *not a search based approach*. It is an iterative sequence of combinatorial manipulations, that monotonically improves the measures, leading to element-wise maximization of ν_θ (See [12]). It is shown in [12]:

$$\lim_{\theta \to 0^+} \theta \big[\mathbb{I} - (1-\theta)\Pi\big]^{-1} \chi = \mathscr{P}\chi \qquad (2)$$

where the i^{th} row of $\mathscr{P}$ (denoted as $\wp^i$) is the stationary probability vector for the PFSA initialized at state q_i. In other words, $\mathscr{P}$ is the Cesaro limit of the stochastic matrix Π, satisfying $\mathscr{P} = \lim_{k\to\infty} \sum_{j=0}^{k} \Pi^k$ [14].

Proposition 1 (See [12]): Since the optimization maximizes the language measure element-wise for $\theta \to 0^+$, it follows that for the optimally supervised plant, the standard inner product $\langle \wp^i, \chi \rangle$ is maximized, irrespective of the starting state $q_i \in Q$.

Notation 3: The optimal θ-dependent measure for a PFSA is denoted as $\nu^\star_\theta$ and the limiting measure as $\nu^\star$.

III. Modeling Ad-hoc Networks as PFSA

We consider an ad-hoc network of communicating nodes endowed with limited computational resources. For simplicity of exposition, we develop the theoretical results under the assumption of a single sink. This is not a serious restriction and can be easily relaxed. The location and identity of the sink is not known a priori to the individual nodes. Inter-node communication links are assumed to be imperfect, with the possibility of packet drop in each transmission attempt. We assume nodes can efficiently gather the following information:

1) *(Set of Neighboring Nodes:)* Number and unique id. of nodes to which it can send data via a 1-hop link.
2) *(Local Link Properties:)* Link-specific probability of packet drop for one-way communication to a specific neighbor.

We further assume that the link-specific packet drop probabilities are either constant, or change slowly enough, making it possible to treat them locally as time-invariant constants for route optimization. Note that this does not imply that the network topology is assumed to be static; we only require that the packet-drop probability for communication from any given node q_i to a particular neighbor q_j be more or less constant, say 0.7. Thus q_i may choose not to send data to q_j all the time, but when it does, then, on the average, 70% of the packets get dropped. In practice, the packet drop probabilities may vary with current network condition, e.g. congestion leading to buffer overflow at specific nodes or (in the context of sensor networks) high-traffic nodes running out of power. We do not consider these effects in detail; however we briefly describe strategies to handle such effects via simple modifications of the basic principles laid out under the assumption of constant drop probabilities. Specific applications, such as wireless sensor networks, require routing schemes that in addition to throughput, are aware of energy and power issues. Also, data-priority need to be respected to enable context-aware routing.

First we formalize the modeling of an ad-hoc network as a probabilistic finite state automata.

Definition 8 (Neighbor Map): If Q is the set of all nodes in the network, then the neighbor map $\mathcal{N} : Q \to 2^Q$ specifies, for each node $q_i \in Q$, the set of nodes $\mathcal{N}(q_i) \subset Q$ (excluding q_i) to which q_i can communicate via a single hop direct link.

Definition 9 (Packet Drop Probability): The link specific packet drop probability $\lambda_{ij} \in [0,1]$ is defined to be the limiting ratio of the number of packets dropped to the total number of packets sent, in communicating from node q_i to node q_j.

Note that the drop probabilities are not constrained to be symmetric in general, i.e., $\lambda_{ij} \neq \lambda_{ji}$. Also, note that we assume the node-based estimation of these ratios to converge fast enough. We visualize the local network around a node q_0 in a manner illustrated in Figure 1(a) (shown for two neighbors q_1 and q_2). In particular, any packet transmitted from q_0 for q_1 has a drop probability λ_{01}, and the ones transmitted to q_2 have a drop probability λ_{02}. To correctly represent this information, we require the notion of *virtual nodes* (q^v_{01}, q^v_{02} in Figure 1(b)).

Definition 10 (Virtual Node): Given a node q_i, and a neighbor $q_j \in \mathcal{N}(q_i)$ with a specified drop probability λ_{ij}, any transmitted data-packet from q_i for q_j is assumed to be first delivered to a virtual node q^v_{ij}, upon which there is either an automatic (i.e. uncontrollable) forwarding to q_j with probability $1 - \lambda_{ij}$, or a drop with probability λ_{ij}. The set of all virtual nodes in a network of Q nodes is denoted by Q^v in the sequel.

Hence, the total number of virtual nodes is $\textsf{CARD}(Q^v) = \sum_{i:q_i \in Q} \mathcal{N}(q_i)$, and satisfies: $0 \leqq \textsf{CARD}(Q^v) \leqq \textsf{CARD}(Q)^2 - \textsf{CARD}(Q)$. We are ready to model an ad-hoc network as a PFSA.

Definition 11 (PFSA Model of Network): For a given set of nodes Q, the function $\mathcal{N} : Q \rightarrow 2^Q$, the link specific drop probabilities λ_{ij} for any node q_i and a neighbor $q_j \in \mathcal{N}(q_i)$, and a specified sink $q_{\textsf{SINK}} \in Q$, the PFSA $\mathbb{G}_N = (Q^N, \Sigma, \delta, \widetilde{\Pi}, \chi, \mathscr{C})$ is defined to be a model of the network, where (denoting $\textsf{CARD}(\mathcal{N}(q_i)) = m$):

∘ STATES: $Q^N = Q \bigcup Q^v \bigcup \{q_D\}$

where Q^v is the set of virtual nodes, and q_D is a dump state which models packet loss. For the alphabet Σ:

∘ ALPHABET: $$\Sigma = \bigcup_{i:q_i \in Q} \left(\bigcup_{j:q_j \in \mathcal{N}(q_i)} \sigma_{ij} \right) \bigcup \{\sigma_D\}$$

σ_{ij} denotes transmission (attempted or actual) from q_i to q_j, and σ_D denotes transmission to q_D (packet loss).

∘ TRANSITION MAP: $$\delta(q, \sigma) = \begin{cases} q^v_{ij} & \text{if } q = q_i, \sigma = \sigma_{ij} \\ q_j & \text{if } q = q^v_{ij}, \sigma = \sigma_{ij} \\ q_D & \text{if } q = q^v_{ij}, \sigma = \sigma_D \\ q_D & \text{if } q = q_D, \sigma = \sigma_D \\ - & \text{undefined otherwise} \end{cases}$$

∘ PROBABILITY MORPH MATRIX: $$\widetilde{\Pi}(q, \sigma) = \begin{cases} \frac{1}{m} & \text{if } q = q_i, \sigma = \sigma_{ij} \\ 1 - \lambda_{ij} & \text{if } q = q^v_{ij}, \sigma = \sigma_{ij} \\ \lambda_{ij} & \text{if } q = q^v_{ij}, \sigma = \sigma_D \\ 1 & \text{if } q = q_D, \sigma = \sigma_D \\ 0 & \text{otherwise} \end{cases}$$

∘ CHARACTERISTIC WEIGHTS: $$\chi_i = \begin{cases} 1 & \text{if } q_i = q_{\textsf{SINK}} \\ 0 & \text{otherwise} \end{cases}$$

∘ CONTROLLABLE TRANSITIONS: $\forall q_i \in Q, q_j \in \mathcal{N}(q_i), q_i \xrightarrow{\sigma_{ij}} q^v_{ij} \in \mathscr{C}$

We note that for a network of Q nodes, the PFSA model almost always has a significantly larger number of states. This state-explosion will not be a problem for the distributed approach developed in the sequel, since we use the complete model $\mathbb{G}_N$ only for the purpose of deriving theoretical guarantees. Note, that Definition 11 generates a PFSA model which can be optimized in a straightforward manner using the language-measure-theoretic technique described in Section II (See [12]) for details). This would yield the optimal routing policy in terms of the disabling decisions at each node that minimize source-to-sink drop probabilities (from every node in the network). To see this explicitly, note that the measure-theoretic approach elementwise maximizes $\lim_{\theta \rightarrow 0^+} \theta \left[\mathbb{I} - (1-\theta)\Pi\right]^{-1} \chi = \mathscr{P}\chi$, where the i^{th} row of $\mathscr{P}$ (denoted as $\wp^i$) is the stationary probability vector for the PFSA initialized at state q_i (See Proposition 1). Since, the dump state has characteristic -1, the sink has characteristic 1, and all other nodes have characteristic 0, it follows that this optimization maximizes the quantity $\wp^i_{\textsf{SINK}} - \wp^i_{\text{DUMP}}$, for every source state or node q_i in the network. Note that $\wp^i_{\textsf{SINK}}, \wp^i_{\text{DUMP}}$ are the stationary probabilities of reaching the sink and incurring a packet loss to dump respectively, from a given source q_i. Thus, maximizing $\wp^i_{\textsf{SINK}} - \wp^i_{\text{DUMP}}$ for every $q_i \in Q$ guarantees that the computed routing policy is indeed optimal in the stated sense. However, the procedure in [12] requires centralized computations, which is precisely what we wish to avoid. The key technical contribution in this paper is to develop a distributed approach to language-measure-theoretic PFSA optimization. In effect, the theoretical development in the next section allows us to carry out the language-measure-theoretic optimization of a given PFSA, in situations where we do not have access to the complete Π matrix, or the χ vector at any particular node (i.e. each node has a limited local view of the network), and are restricted to communicate only with immediate neighbors. We are interested in not just computing the measure vector in a distributed manner, but optimizing the PFSA via selected disabling of controllable transitions (See Section II). This is accomplished by Algorithm 1.

The PFSA based modeling framework is somewhat different from the standard MDP architecture. For example, in contrast to the latter, our actions are "controllable" transitions, and have probabilities associated with them. Rewards and penalties are not associated with individual actions, but with state visitations (and modeled via the characteristic weights). We maximize the long term or expected reward by maximizing the probability of reaching the sink, while simultaneously minimizing the probability of reaching the dump state, i.e., a drop, from any node in the network. (See [16] for more details).

IV. DECENTRALIZED PFSA OPTIMIZATION

Notation 4: In the sequel, the current measure value, for a given θ, at node $q_i \in Q$ is denoted as $\widehat{\nu}_\theta|_i$, and the measure of the virtual node $q^v_{ij} \in Q^N$ is denoted as $\widehat{\nu}_\theta|_{(q^v_{ij})}$. The parenthesized entry (q^V_{ij}) denotes the index of the virtual node q^v_{ij} in the state set Q^N. Similarly, the transition probability from q_i to q^v_{ij} is denoted as $\Pi_{i(q^v_{ij})}$. The subscript entry $i(q^V_{ij})$ denotes the ik^{th} element of Π, where $k = (q^V_{ij})$.

Algorithm 1 establishes a distributed, asynchronous update procedure which achieves the following:

$$\forall q_i \in Q, \widehat{\nu}_\theta|_i \xrightarrow[\text{convergence}]{\text{global}} \nu^\star_\theta|_i \tag{4}$$

where $\nu^\star_\theta|_i$ is the optimal measure for $q_i \in Q$ that would be obtained by optimizing the PFSA $\mathbb{G}_N$, for a given θ, in a centralized approach (See Section II). The optimal routing policy can then be obtained by forwarding packets to neighboring nodes which have a better or equal current measure value. If more than a one such neighbor is available, then one chooses the forwarding node randomly.

Algorithm 1 has four distinct parts, marked as (a1), (a2), (a3) and (a4). Part (a1) involves internode communication, to enable a particular node $q_i \in Q$ to ascertain the current measure values of neighboring nodes, and the drop probabilities λ_{ij} on respective links. Recall, that we assume the probabilities λ_{ij} to be more or less constant; nevertheless

input : $\mathbb{G}_N = (Q, \Sigma, \delta, \widetilde{\Pi}, \chi, \mathscr{C})$, θ
begin
    Initialize $\forall q_i \in Q, \widehat{\nu}_\theta|_i = 0$
    /* Begin Infinite Asynchronous Loop */
    while true do
        for *each node* $q_i \in Q$ **do**
            if $\mathcal{N}(q_i) \neq \varnothing$ **then**
                $m = \text{CARD}(\mathcal{N}(q_i))$
                for *each node* $q_j \in \mathcal{N}(q_i)$ **do**
                    /* (a1) Internode Communication */
                    Query $\widehat{\nu}_\theta|_j$ & Drop Prob. λ_{ij}
                    /* (a2) Control Adaptation */
                    if $\widehat{\nu}_\theta|_j < \widehat{\nu}_\theta|_i$ **then**
                        $\Pi_{ii} = \Pi_{ii} + \Pi_{i(q^V_{ij})}$
                        $\Pi_{i(q^V_{ij})} = 0$; /* Disable */
                    else
                        if $\Pi_{i(q^V_{ij})} == 0$ **then**
                            $\Pi_{i(q^V_{ij})} = \frac{1}{m}$
                            $\Pi_{ii} = \Pi_{ii} - \frac{1}{m}$ /* Enable */
                    /* (a3) Updating Virtual Nodes */
                    $\widehat{\nu}_\theta|_{(q^V_{ij})} = (1-\theta)(1-\lambda_{ij})\widehat{\nu}_\theta|_j$
                /* (a4) Updating Node */
                $$\widehat{\nu}_\theta|_i = \sum_{j:q_j \in \mathcal{N}(q_i)} (1-\theta)\Pi_{i(q^V_{ij})}\widehat{\nu}_\theta|_{(q^V_{ij})} + (1-\theta)\Pi_{ii}\widehat{\nu}_\theta|_i + \theta\chi|_i$$
end

Algorithm 1: Distributed Update of Node Measures

nodes estimate these values to adapt to changing (albeit slowly) network conditions. Part (a2) is the control adaptation, in which the nodes decide, based on local information, the set of forwarding nodes. Part (a3) is the computation of the updated measure values for the virtual nodes q^v_{ij} where $j : q_j \in \mathcal{N}(q_i)$. Finally, part (a4) updates the measure of the node q_i based on the computed current measures of the virtual nodes. We note that Algorithm 1 only uses information that is either available locally, or that which can be queried from neighboring nodes.

Proposition 2 (Convergence): For a network Q modeled as a PFSA $\mathbb{G}_N = (Q^N, \Sigma, \delta, \widetilde{\Pi}, \chi, \mathscr{C})$, the distributed procedure in Algorithm 1 has the following properties:

1) Computed measure values for every node $q_i \in Q$ are non-negative and bounded above by 1, i.e.,
$$\forall q_i \in Q^N, \forall t \in [0, \infty), \ \widehat{\nu}^t_\theta|_i \in [0, 1] \tag{5}$$
2) For constant drop probabilities and constant neighbor map $\mathcal{N} : Q \to 2^Q$, Algorithm 1 converges in the sense:
$$\forall q_i \in Q^N, \lim_{t\to\infty} \widehat{\nu}^t_\theta|_i = \nu^\infty_\theta|_i \in [0, 1] \tag{6}$$
3) Convergent measure values coincide with the optimal values computed by the centralized approach:
$$\forall q_i \in Q^N, \nu^\infty_\theta|_i = \nu^\star_\theta|_i \tag{7}$$

Proof: Due to limited space, the interested reader is referred to the preprint [17]. ■

Proposition 3 (Initialization Independence): For a network Q modeled as a PFSA $\mathbb{G}_N = (Q^N, \Sigma, \delta, \widetilde{\Pi}, \chi, \mathscr{C})$, convergence of Algorithm 1 is independent of the initialization of the measure values, i.e., if $\widehat{\nu}^t_{\theta,\alpha}$ denotes the measure vector at time t with arbitrary initialization $\alpha \in [0, 1]^{\text{CARD}(Q^N)}$, then:
$$\lim_{t\to\infty} \widehat{\nu}^t_{\theta,\alpha} = \lim_{t\to\infty} \widehat{\nu}^t_\theta \tag{8}$$
where $\widehat{\nu}^0_{\theta,\alpha} = \alpha$ and $\widehat{\nu}^0_\theta = [0 \cdots 0]^T$.

Proof: Due to limited space, the interested reader is referred to the preprint [17]. ■

Next we establish guarantees on global performance achieved via local decisions dictated by Algorithm 1. Next, we make rigorous our notion of policy performance, and near-global or ϵ-optimality.

Definition 12 (Policy Performance & ϵ-Optimality): The performance vector ρ^S of a given routing policy S is the vector of node-specific probabilities of power eventually reaching the sink. A policy U has *Utopian performance* if its performance vector (denoted as ρ^U) element-wise dominates the one for any arbitrary policy S, i.e. $\forall q_i \in Q^N, \rho^U_i \geqq \rho^S_i$. A policy P has *$\epsilon$-optimal* performance, if for some given $\epsilon > 0$, we have:
$$||\rho^P - \rho^U||_\infty \leqq \epsilon \tag{9}$$
For a chosen θ, the limiting policy P_θ computed by Algorithm 1 results in element-wise maximization of the measure vector over all possible supervision policies (where supervision is to be understood in the sense of the defined control philosophy). $\widehat{\nu}^\infty_\theta$ is related to the policy performance vector ρ^{P_θ} as follows. Selective disabling of the transitions dictated by the policy P_θ induces a controlled PFSA, which represents the optimally supervised network, for a given θ. Let the transition matrix for this optimized PFSA be $\Pi^\star_\theta$, and its Cesaro limit be $\mathscr{P}^\star_\theta$. (Note: $\Pi^\star_\theta$, $\mathscr{P}^\star_\theta$ are stochastic matrices.) Then:
$$\forall q_i \in Q^N, \mathscr{P}^\star_\theta \chi\big|_{i,(q_{\text{SINK}})} = \rho^{P_\theta}_i \tag{10}$$

Due to limited space, we state the following key results without proof. The interested reader is referred to the preprint [17].

Proposition 4 (Global ϵ-Optimality): Given any $\epsilon > 0$, choosing $\theta = \epsilon / m^2$ where $m = \max_{q \in Q} \text{CARD}(\mathcal{N}(q))$ guarantees that the limiting policy computed by Algorithm 1 is ϵ-optimal in the sense of Definition 12.

Proposition 5 (Asymptotic Runtime Complexity): With no communication delays and assuming synchronized updates, convergence time $\mathbb{T}_c$ to ϵ-optimal operation for a network of N physical nodes and maximum m neighbors, satisfies:
$$\mathbb{T}_c = O\left(\frac{Nm^2}{\epsilon(1-\gamma_\star)}\right)$$
where $\gamma_\star$ is a lower bound on drop probabilities

Remote sensing applications necessitate route updates as nodes die. Each node can regulate incoming traffic by deliberately reporting lower values of its current self-measure to its neighbors:
$$\text{Reported} \longrightarrow r^{[k]}_\theta\big|_i = \zeta(q_i, k)\nu^{[k]}_\theta\big|_i \longleftarrow \text{Computed} \tag{11}$$
where $\forall q_i \in Q, k \in [0, \infty), \zeta(q_i, k) \in [0, 1]$ is a multiplicative factor which is modulated to have decreasing values

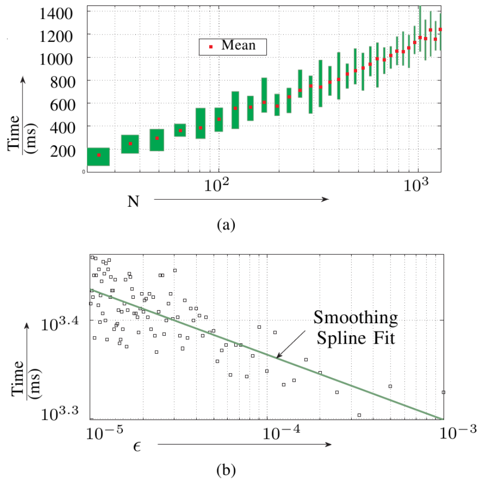


Fig. 2: Convergence complexity: (a) illustrates little dependence of convergence on network size. (b) captures the $O(1/\epsilon)$ dependence

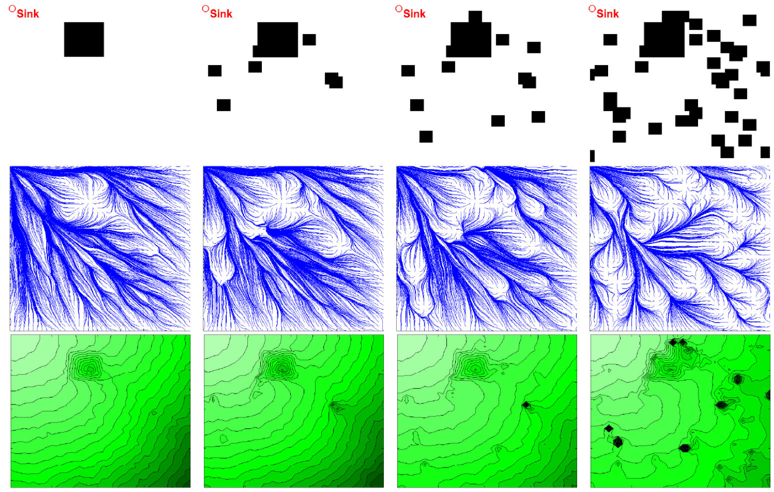


Fig. 3: Progressive node deaths: Top row: failed regions, Middle row: packet path signatures to sink from live nodes, Bottom row: level sets for measure field

as node energy gets depleted, or as local congestion increases. Such modulation forces automatic self-organization to compute alternate routes that tend to avoid the particular node. The dynamics of such context-aware modulation may be non-trivial; while for slowly varying $\zeta(q_i, k)$, the convergence results presented here is expected to hold true, rapid fluctuations in $\zeta(q_i, k)$ may be problematic.

V. Conclusions & Future Work

This paper introduces GODDeS: a new routing algorithm designed to effectively exploit high quality paths in lossy adhoc wireless environments, typically with a large number of nodes. The routing problem is modeled as an optimal control problem for a decentralized Markov Decision Process, with links characterized by locally known packet drop probabilities that either remain constant on average or change slowly. Theoretical results provide rigorous guarantees on global performance, showing that the algorithm achieves near-global optimality, in polynomial time. It is also argued that GODDeS is significantly congestion-aware, and exploits multi-path routes optimally. Theoretical development is supported by network simulation.

Future work will proceed in the following directions:

1) Design explicit strategies for energy and congestion awareness within the GODDeS framework. In particular, investigate the ramifications of various choices of the measure reduction factor described in Eq. (11).
2) Grossly incorrect estimations of the link-specific drop probabilitieswill translate to incorrect routing decisions, and decentralized strategies for robust identification of these parameters need to be investigated.
3) Explicit design of implementation details such as packet headers, node data structures and pertinent neighbor-neighbor communication protocols.

References

[1] D. P. Bertsekas and R. G. Gallager, "Distributed asynchronous bellman-ford algorithm," in *Data Networks*. Prentice Hall, Englewood Cliffs, 1987, ch. 5.2.4, pp. 325–333.

[2] C. E. Perkins and E. M. Royer, "The ad hoc on-demand distance-vector protocol," in *Ad Hoc Networking*, C. E. Perkins, Ed. Addison-Wesley, 2001, ch. 6, pp. 173–219.

[3] ——, "Ad-hoc on-demand distance vector routing," in *2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, USA*. IEEE, February 1999, pp. 90–100.

[4] D. B. Johnson, D. A. Maltz, and J. Broch, "Dsr: The dynamic source routing protocol for multihop wireless ad hoc networks," in *Ad Hoc Networking*, C. Perkins, Ed. Addison-Wesley, 2001, pp. 139–172.

[5] A. Pandey, M. N. Ahmed, N. Kumar, and P. Gupta, "A hybrid routing scheme for mobile ad hoc networks with mobile backbones," in *International Conference on High Performance Computing*, IEEE. IEEE, December 2006, pp. 411–423.

[6] G. Koltsidas, G. Dimitriadis, and F.-N. Pavlidou, "On the performance of the hsls routing protocol for mobile ad hoc networks," *Wirel. Pers. Commun.*, vol. 35, no. 3, pp. 241–253, 2005.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th annual international conference on Mobile*. New York, USA: ACM Press, 2000, pp. 56–67.

[8] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, pp. 419–434, 2005.

[9] B. Awerbuch, D. Holmer, and H. Rubens, "High throughput route selection in multi-rate ad hoc wireless networks," in *WONS*, 2004, pp. 253–270.

[10] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of markov decision processes," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-2000)*, 2000, pp. 32–37.

[11] ——, "The complexity of decentralized control of markov decision processes," *Math. Oper. Res.*, vol. 27, no. 4, pp. 819–840, 2002.

[12] I. Chattopadhyay and A. Ray, "Language-measure-theoretic optimal control of probabilistic finite-state systems," *International Journal of Control*, vol. 80, no. 8, pp. 1271–1290, Aug. 2007.

[13] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation, 2nd ed.* Addison-Wesley, 2001.

[14] R. Bapat and T. Raghavan, *Nonnegative matrices and Applications*. Cambridge University Press, 1997.

[15] W. Rudin, *Real and Complex Analysis, 3rd ed.* McGraw Hill, New York, 1988.

[16] I. Chattopadhyay and A. Ray, "Optimal control of infinite horizon partially observable decision processes modeled as generators of probabilistic regular languages," *International Journal of Control*, vol. 83, no. 3, pp. 457–483, March 2010.

[17] I. Chattopdhyay, "GODDeS: Globally \epsilon-Optimal Routing Via Distributed Decision-theoretic Self-organization," *ArXiv e-prints 1008.1620*, Aug. 2010. [Online]. Available: http://adsabs.harvard.edu/abs/2010arXiv1008.1620C