Convergence Results for Ant Routing Algorithms via Stochastic Approximation *

Punyaslok Purkayastha
Institute for Systems Research and ECE Department
University of Maryland
College Park, MD, USA
punya@umd.edu

John S. Baras
Institute for Systems Research and ECE Department
University of Maryland
College Park, MD, USA
baras@umd.edu

ABSTRACT

In this paper, we provide convergence results for an Ant Routing (ARA) Algorithm for wireline, packet switched communication networks, that are acyclic. Such algorithms are inspired by the foraging behavior of ants in nature. We consider an ARA algorithm proposed by Bean and Costa [2]. The algorithm has the virtues of being adaptive and distributed, and can provide a multipath routing solution. We consider a scenario where there are multiple incoming data traffic streams that are to be routed to their destinations via the network. Ant packets, which are nothing but probe packets, are used to estimate the path delays in the network. The node routing tables, which consist of routing probabilities for the outgoing links, are updated based on these delay estimates. In contrast to the available analytical studies in the literature, the link delays in our model are stochastic, time-varying, and dependent on the link traffic. The evolution of the delay estimates and the routing probabilities are described by a set of stochastic iterative equations. In doing so, we take into account the distributed and asynchronous nature of the algorithm operation. Using methods from the theory of stochastic approximations, we show that the evolution of the delay estimates can be closely tracked by a deterministic ODE (Ordinary Differential Equation) system, when the step-size of the delay estimation scheme is small. We study the equilibrium behavior of the ODE in order to obtain the equilibrium behavior of the algorithm. We also provide illustrative simulation results.

*Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

HSCC'10, April 12-15, 2010, Stockholm, Sweden

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing Protocols*; G.3 [**Probability Statistics**]: Queuing Theory, Probabilistic algorithms

General Terms

Algorithms, Theory, Performance

Keywords

Ant routing algorithms, stochastic approximations and learning algorithms, queuing networks

1. INTRODUCTION

In this paper we study the convergence and properties of a routing algorithm proposed for communication networks, that belongs to the class of Ant Routing (ARA) Algorithms. It was observed in an experiment conducted by biologists Deneubourg *et. al.* [9], called the double bridge experiment, that under certain conditions, a group of ants when presented with two paths to a source of food, is able to collectively converge to the shorter path. It was found that every ant lays a trail of a chemical substance called *pheromone* as it walks along a path. Subsequent ants follow paths with stronger pheromone trails, and in their turn reinforce the trails. Because ants take lesser time to traverse the shorter path, pheromone concentration increases more rapidly along this path. These "positive reinforcement" effects culminate in all ants following, and thus discovering, the shorter path.

Most of the ARA algorithms proposed in the literature are inspired by the basic idea of creation and reinforcement of a pheromone trail on a path that serves as a measure of the quality of the path. These algorithms employ probe packets called ant packets (analogues of ants) that help create analogues of pheromone trails on paths. In the context of routing, these trails are based on path delay measurements made by the ant packets. Routing tables at the nodes are updated based on the path pheromone trails. The update algorithms help direct data packets along outgoing links that lie on paths with lower delays.

We consider a wireline, packet-switched network, and provide convergence results for an ARA algorithm proposed by Bean and Costa [2]. In an earlier paper [17], we had studied convergence for a simple N parallel links network. In this paper, we provide convergence results for a general, acyclic network[1]. The Bean, Costa algorithm has many attractive features. It is distributed, adaptive, and can provide a multipath routing solution — that is, the incoming traffic at a source is split between the multiple paths available to the destination. This enables efficient utilization of network resources. We now briefly dwell on the literature on ARA algorithms.

Literature. ARA algorithms have been proposed for all kinds of networks — circuit- and packet-switched wireline, as well as packet-switched wireless networks. We briefly discuss the algorithms for packet-switched networks, because they are more relevant; for a more comprehensive survey see Dorigo, Stutzle [10] and Bonabeau, Dorigo, and Theraulaz [5]. Most of the algorithms proposed and studied for such networks — for example, Gabber, Smith [11], Di Caro, Dorigo [7], Subramanian, Druschel, and Chen [19] (all the above are for wireline networks), and Baras, Mehta [1] (for wireless networks) — are variants of the Linear Reinforcement (LR) scheme considered in studies of stochastic learning automata (see Kaelbling, Littman, and Moore [13] and Thathachar and Sastry [20]). In these works, variants of the LR scheme are used to adjust routing probabilities at the nodes based on path pheromone trails. Yoo, La, and Makowski [21] consider the scheme of [19] for a network consisting of two nodes connected by L parallel links. The link delays are deterministic. Ant packets are either routed uniformly at the nodes — called "uniform routing" — or are routed based on the node routing tables — called "regular routing". A rigorous analysis shows that the routing probabilities converge in distribution for the uniform routing case, and almost surely to a shortest path solution for the regular routing case. The LR scheme however, is not designed for applications where the delays are stochastic, time-varying, which is the case of main interest to us. ARA algorithms different from the LR scheme are considered in [5], [2].

Though a large number of ARA algorithms have been proposed, fewer analytical studies are available in the literature. Algorithms similar to those that aim to explain the observations in the double bridge experiment, have been rigorously studied in Makowski [15], Das and Borkar [8]. Makowski considers the case where there are two paths of equal length to a food source, and a model where each ant chooses a path with a probability proportional to a power $\nu \geq 0$, of the number of ants that have previously traversed the path. Using stochastic approximation and martingale techniques the paper provides convergence results, and shows that the asymptotic behavior can be quite complex (in particular, only when $\nu > 1$, all ants eventually choose one path). Das and Borkar consider a scenario with multiple disjoint paths between a source and a destination. There are three algorithms — a pheromone update algorithm that builds a pheromone trail based on the number of ants traversing the path and path length, a utility estimate algorithm based on the path pheromone trail, and a routing probability update algorithm that uses the utility estimates. Using stochastic approximation methods, they show convergence to a shortest path solution if there is an 'initial bias', i.e., if initially there is a higher probability of choosing the shortest path. Gutjahr [12] considers a problem where ant-like agents help solve the combinatorial optimization problem of finding an optimal cycle on a graph, with no nodes being repeated except for the start node. Arc costs are deterministic. Agents sample walks based on routing probabilities, and reinforce pheromone trails on arcs, which in turn, influence the routing probabilities. The paper shows that asymptotically, with probability arbitrarily close to one, an optimal cycle can be found. Another analytical study is the paper [21] discussed above.

Contributions and Related Work. The above set of analytical studies have mostly concentrated on networks with deterministic link delays. In contrast, we provide convergence results when the link delays are stochastic, time-varying, and are dependent on the link traffic. This is a more relevant and interesting case.

Bean and Costa [2] study their scheme using a combination of simulation and analysis. They employ a 'time-scale separation approximation' whereby average network delays are computed 'before' the routing probabilities are updated. Numerical iterations of an analytical model based on this approximation and simulations are shown to agree well. However, the time-scale separation is not justified [2], nor is any formal study of convergence provided.

We consider a stochastic model for the arrival processes and packet lengths of both the ant and the incoming data streams. The ARA scheme consists of a delay estimation algorithm and a routing probability update algorithm, that utilizes the delay estimates. These algorithms run at every node of the network. The delay estimates are formed based on measurements of path delays (these delays are caused by queuing delays on the links). We describe the evolution of these algorithms by a set of discrete stochastic iterations. Our formulation considers the distributed and asynchronous nature of algorithm operation. We show, using methods from the theory of stochastic approximations, that the evolution of the delay estimates can be closely tracked by a deterministic ODE (Ordinary Differential Equation) system, when the step size of the delay estimation scheme is small. We study the equilibrium behavior of the ODE system in order to obtain the equilibrium behavior of the routing algorithm. We provide illustrative simulation results.

Our approach is most closely related to Borkar and Kumar [6], which studies an adaptive algorithm that converges to a Wardrop equilibrium routing solution. Our framework is similar to theirs — there is a delay estimation algorithm and a routing probability update algorithm which utilizes the delay estimates. Their routing probability update scheme moves on a slower "time scale" than the delay estimation scheme. In our case however, the routing probability update scheme is on the same "time scale" as the delay estimation scheme, and our method of analysis is consequently different. This could also be desirable in practice, because the algorithm convergence will be much faster.

The paper is organized as follows. In this paper we separately consider the two cases where ant packets are routed according to uniform and regular routing. There is a parallel development of the discussion related to these two forms of routing. In Section 2 we outline in detail the mechanism of operation of ARA algorithms, and discuss the Bean, Costa algorithm. Section 3 provides a formal discussion of our acyclic network model and assumptions, and a formulation of the routing problem. We analyse the routing algorithm in Section 4, discuss our ODE approximation results and

[1]For a definition of such networks see Section 3.

[2]We shall see, in Section 4, that it holds only when the step-size ϵ in the delay estimation algorithm is small ($\epsilon \downarrow 0$).

related computations, and the equilibrium behavior of the algorithm. In Section 5, we consider an example acyclic network. Related simulation results are provided and discussed. Section 6 provides concluding remarks and discusses a few directions for future research.

2. ANT ROUTING ALGORITHMS: MECHANISM OF OPERATION

We provide in this section a brief description of the mechanism of operation of ant routing for a wireline communication network. Such a network can be represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, with a set of nodes $\mathcal{N}$, and a set of directed links $\mathcal{L}$. Our formal description follows the general framework of Di Caro and Dorigo [7], [10]. Alongside, we describe the Bean, Costa [2] scheme.

Every node i in the network maintains two key data structures — a matrix of routing probabilities, the routing table $\mathcal{R}(i)$, and a matrix of various kinds of statistics used by the routing algorithm, called the network information table $\mathcal{I}(i)$. For a particular node i, let $N(i,k)$ denote the set of neighbors of i (corresponding to the outgoing links (i,j) from i) through which node i routes packets towards destination node k. For the network consisting of $|\mathcal{N}|$ nodes, the matrix $\mathcal{R}(i)$ has $|\mathcal{N}|-1$ columns, corresponding to the $|\mathcal{N}|-1$ destinations towards which node i could route data packets, and $|\mathcal{N}|-1$ rows, corresponding to the maximum number of neighbor nodes of node i. The entries of $\mathcal{R}(i)$ are the probabilities ϕ_{ij}^k. ϕ_{ij}^k denotes the probability of routing an incoming data packet at node i and bound for destination k via the neighbor $j \in N(i,k)$. The matrix $\mathcal{I}(i)$ has the same dimensions as $\mathcal{R}(i)$, and its (j,k)-th entry contains various statistics pertaining to the route from i to k that goes via j (denoted henceforth by $i \to j \to \cdots \to k$). Examples of such statistics could be mean delay and delay variance estimates of the route $i \to j \to \cdots \to k$. These statistics are updated based on the information the ant packets collect about the route. $\mathcal{I}(i)$ thus represents the characteristics of the network that are learned by the nodes through the ant packets. Based on the information collected in $\mathcal{I}(i)$, "local decision-making" — the update of the routing table $\mathcal{R}(i)$ — is done. The iterative algorithms that are used to update $\mathcal{I}(i)$ and $\mathcal{R}(i)$ will be referred to as the *learning algorithms*.

We now describe the mechanism of operation of ARA algorithms. For ease of exposition, we restrict attention to a particular fixed destination, and consider the problem of routing from every other node to this node, which we label as D. The network information tables $\mathcal{I}(i)$ at the nodes contain only estimates of mean delays.

Forward ant generation and routing. At certain intervals, forward ant (FA) packets are launched from a node i towards the destination D to discover low delay paths to it. The FA packets sample walks on the graph $\mathcal{G}$ based either on the current routing probabilities at the nodes as in regular ant routing (regular ARA), or uniformly [3] as in uniform ant routing (uniform ARA). Uniform ant routing might be preferred in certain cases; for instance, when we want the ant packets to explore the network in a completely "unbiased" manner. FA packets share the same queues as data packets and so experience similar delay characteristics as data packets. Every FA packet maintains a stack of data structures containing the IDs of nodes in its path and the per hop delays encountered. The per hop delay measurements can be obtained through time stamping of the FA packets as they pass through the various nodes.

[3]routed with equal probability on each outgoing link

Backward ant generation and routing. Upon arrival of an FA at D, a backward ant (BA) packet is generated. The FA packet transfers its stack to the BA. The BA packet then retraces back to the source i the path traversed by the FA packet. BA packets travel back in high priority queues, so as to quickly get back to the nodes and minimize the effects of outdated or stale measurements. At each node that the BA packet traverses through, it transfers the delay information that was gathered by the FA packet. This information is used to update matrices $\mathcal{I}$ and $\mathcal{R}$ at the respective nodes. Thus the arrivals of BA packets at the nodes trigger the iterative learning algorithms.

We now describe the Bean, Costa [2] learning algorithm. Suppose that an FA packet measures the delay Δ_{ij}^D associated with a walk from i to D via the outgoing link (i,j). This delay is more precisely the following. Let $\widetilde{J}_j^D$ denote a sample sum of the delays in the links, experienced by an FA packet moving from node j to node D (it is thus a sample of the expected 'cost-to-go' from j to D). Let $\widetilde{w}_{ij}$ denote a sample of the delay experienced by an FA packet traversing the link (i,j). Then $\Delta_{ij}^D = \widetilde{w}_{ij} + \widetilde{J}_j^D$. When the corresponding BA packet comes back at node i the delay information is used to update the estimate X_{ij}^D of the mean delay using the simple exponential estimator

$$X_{ij}^D := X_{ij}^D + \epsilon(\Delta_{ij}^D - X_{ij}^D), \tag{1}$$

where $\epsilon > 0$ is a small constant. We refer to X_{ij}^D as *the mean delay estimate for the route* $i \to j \to \cdots \to D$. Estimates X_{ik}^D corresponding to other neighbors $k \in N(i,D)$ are left unchanged.

Simultaneously, the routing probabilities at i are updated using the relation

$$\phi_{ij}^D = \frac{(X_{ij}^D)^{-\beta}}{\sum_{k \in N(i,D)} (X_{ik}^D)^{-\beta}}, \ \forall j \in N(i,D), \tag{2}$$

where β is a constant positive integer. ϕ_{ij}^D is thus inversely proportional to X_{ij}^D. β influences the extent to which outgoing links with lower delay estimates are favored compared to the ones with higher delay estimates.

We can interpret the quantity $(X_{ij}^D)^{-1}$ as analogous to a "pheromone trail or deposit" on outgoing link (i,j). This trail gets dynamically updated by the ant packets. It influences the routing tables through relation (2). Equation (2) shows that the outgoing link (i,j) is more desirable when X_{ij}^D, the delay through j, is smaller; in other words, when the pheromone trail is stronger, relative to the other routes.

3. FORMULATION OF THE PROBLEM. THE ACYCLIC NETWORK MODEL

We consider the problem of routing from the various nodes i of the network to a single destination D. At every node i there exist queues (buffers) Q_{ij} associated with the outgoing links (i,j); we assume these to be FIFO queues of infinite size. The network can be thought of equivalently as a system of inter-connected queues (a queuing network). Every link (i,j) has capacity C_{ij}. We assume that the queuing delays dominate the processing and propagation delays in the links.

We consider acyclic networks and define them following Bertsekas, Gallager [3]. A queue Q_{ij} is said to be downstream with respect to a queue Q_{kl} if some portion of the traffic through the latter queue flows through the former. An acyclic network is one for which it is not possible that simultaneously Q_{ij} is downstream of Q_{kl} and Q_{kl} is downstream of Q_{ij}. The set $N(i) = \{j : (i,j) \in \mathcal{L}\}$ denotes the set of downstream neighbors of i. An example of an acyclic network is given in Figure 2, pp. 8. We shall denote the routing probability entries of $\mathcal{R}(i)$ by ϕ_{ij} (i.e., without explicitly mentioning the destination). The mean delay estimate entries of $\mathcal{I}(i)$ are denoted by X_{ij}.

The general algorithm, as described in Section 2, is asynchronous (and distributed). This is because the nodes launch the FA packets towards the destination in an unco-ordinated way. Moreover, there is a random delay as each FA-BA pair travels through the network. The learning algorithms at the nodes for updating $\mathcal{R}$ and $\mathcal{I}$ are thus triggered at random points of time (when BA packets come back). We consider a more simplified view of the algorithm operation, which is still asynchronous and distributed, retains the main characteristics and the essence of the algorithm, but is more convenient to analyze.

We assume that FA packets are generated according to a Poisson process of rate $\lambda_i^a > 0$ at node i ($\lambda_D^a = 0$). We consider a model with the following assumptions on the algorithm operation.

($M1$) We assume that the BA packets take negligible time to travel back to the source nodes (from which the corresponding FA packets were launched) from destination D. Because BA packets are expected to travel back through high priority queues, the delays might not be significant, except for large-sized networks with significant propagation delays. On the other hand, incorporating the effects of such delays into our model introduces additional complications related to asynchrony.

($M2$) In the general algorithm operation, a BA packet updates the delay estimates at every node that it traverses on its way back to the source, besides the source itself. In what follows, we shall consider the more simplified algorithm operation, whereby only at the source node the delay estimates and the routing probabilities are updated.

We assume that data packets are generated according to a Poisson process of rate $\lambda_i^d \geq 0$ at node i; for some nodes it is possible that no data packets are generated, i.e., the rate is zero (for the destination, $\lambda_D^d = 0$).

Let $\{\alpha(m)\}_{m=1}^{\infty}$ denote the sequence of times at which FA packets are launched from the various nodes of the network. Let $\{\delta(n)\}_{n=1}^{\infty}$ denote the sequence of times at which FA packets arrive at the destination node D (we set $\alpha(0) = 0, \delta(0) = 0$). Because we have assumed that BA packets take negligible time to travel back to the sources, these are also the sequence of times at which BA packets come back to the source nodes. Consequently, these are the sequence of times at which algorithm updates are triggered at the various nodes. At time $\delta(n)$, let $X(n)$ and $\phi(n)$ denote, respectively, the vector of mean delay estimates and the vector of outgoing routing probabilities at the network nodes. The components of $X(n)$ and $\phi(n)$ are $X_{ij}(n), (i,j) \in \mathcal{L}$, and $\phi_{ij}(n), (i,j) \in \mathcal{L}$, respectively.

Thus, by time $\delta(n)$, overall n BA packets will have come back to the network nodes. (At this point it is useful to recall assumption ($M2$)). Let $T(n)$ be the $\mathcal{N}$-valued random variable that indicates which node the n-th BA packet comes back to. Then $\xi_i(n) = \sum_{k=1}^{n} I_{\{T(k)=i\}}$ gives the number of BA packets that have come back at node i by time $\delta(n)$[4]. Let $R_i(.)$ denote the routing decision variable for FA packets originating from node i. We say that the event $\{R_i(k) = j\}$ has occurred if the k-th FA packet that arrives at D *and* that has been launched from i, has been routed via the outgoing link (i,j). Let $\psi_{ij}(n) = \sum_{k=1}^{\xi_i(n)} I_{\{R_i(k)=j\}}$; $\psi_{ij}(n)$ gives the number of FA packets that arrive at node D by time $\delta(n)$, having been launched from node i and routed via (i,j). By the zero delay assumption on the travel time of the BA packets and the assumption ($M2$) on algorithm operation, $\psi_{ij}(n)$ is also the number of BA packets that come back to i via j, by time $\delta(n)$. Let $\{\Delta_{ij}(m)\}$ denote the sequence of delay measurements made by successive FA packets arriving at D, that have been launched from node i and routed via the outgoing link (i,j). This is also the sequence of delay measurements about the route $i \rightarrow j \rightarrow \cdots \rightarrow D$ made available to the source i by the BA packets.

Let's suppose that at time $\delta(n)$ a BA packet comes back to node i. Furthermore, suppose that the corresponding FA packet was routed via the outgoing link (i,j). When this BA packet comes back to node i, the delay estimate X_{ij} is updated using an exponential estimator

$$X_{ij}(n) = X_{ij}(n-1) + \epsilon\Big(\Delta_{ij}(\psi_{ij}(n)) - X_{ij}(n-1)\Big), \quad (3)$$

$\epsilon \in (0,1)$ being a small positive constant. Estimates X_{ik} for the other routes $i \rightarrow k \rightarrow \cdots \rightarrow D$ ($k \in N(i), k \neq j$) are left unchanged

$$X_{ik}(n) = X_{ik}(n-1). \quad (4)$$

Also, the estimates at the other nodes do not change

$$X_{lp}(n) = X_{lp}(n-1), \quad \forall p \in N(l), \forall l \neq i. \quad (5)$$

Also, as soon as the delay estimates are updated at node i, the outgoing routing probabilities are also updated

$$\phi_{ij}(n) = \frac{(X_{ij}(n))^{-\beta}}{\sum_{k\in N(i)} (X_{ik}(n))^{-\beta}}, \quad \forall j \in N(i). \quad (6)$$

The routing probabilities at the other nodes do not change.

In general thus the evolution of the delay estimates at the network nodes can be described by the following set of stochastic iterative equations

$$\begin{aligned} X_{ij}^{\epsilon}(n) \;=\; & X_{ij}^{\epsilon}(n-1) + \epsilon\, I_{\{T^{\epsilon}(n)=i, R_i^{\epsilon}(\xi_i^{\epsilon}(n))=j\}} \times \\ & \Big(\Delta_{ij}^{\epsilon}(\psi_{ij}^{\epsilon}(n)) - X_{ij}^{\epsilon}(n-1)\Big), \forall (i,j) \in \mathcal{L}, n \geq 1, \end{aligned} \quad (7)$$

starting with the initial conditions $X_{ij}^{\epsilon}(0) = x_{ij}, \forall (i,j) \in \mathcal{L}$.

The routing probabilities are updated in the usual way

$$\phi_{ij}^{\epsilon}(n) = \frac{(X_{ij}^{\epsilon}(n))^{-\beta}}{\sum_{k\in N(i)} (X_{ik}^{\epsilon}(n))^{-\beta}}, \quad \forall (i,j) \in \mathcal{L}, n \geq 1, \quad (8)$$

with initial conditions $\phi_{ij}^{\epsilon}(0) = \frac{(x_{ij})^{-\beta}}{\sum_{k\in N(i)} (x_{ik})^{-\beta}}, \forall (i,j) \in \mathcal{L}$. Though not explicitly mentioned, it is understood that there are no algorithm updates being made at D.

The ϵ's in the superscript in equations (7) and (8) above, recognize the dependence of the evolution of the quantities

[4] I_A denotes the indicator random variable for the event A.

involved (for example, the delay estimates X_{ij}) on ϵ. However, for most of the paper[5], we shall not use this notation; this enables the discussion to be less cumbersome. Also, we note that equations (7) and (8) describe the evolution of the delay estimates and the routing probabilities for the regular ARA as well as for the uniform ARA case.

We also introduce the following continuous time processes, $\{x(t), t \geq 0\}$ and $\{f(t), t \geq 0\}$, defined by the equations

$$\begin{aligned} x(t) &= X(n), \quad \text{for } \delta(n) \leq t < \delta(n+1), n = 0, 1, 2, \ldots, \\ f(t) &= \phi(n), \quad \text{for } \delta(n) \leq t < \delta(n+1), n = 0, 1, 2, \ldots. \end{aligned}$$

The components of $x(t)$ and $f(t)$ are denoted by $x_{ij}(t)$ and $f_{ij}(t)$, respectively.

In the case of regular ARA, an FA packet as well as a data packet are routed at an intermediate node based on the current routing probabilities at the node. Thus, in view of the discussion in this section, a packet that arrives at node i at time t, is routed according to the routing probabilities $f_{ij}(t), j \in N(i)$, and joins the corresponding queues. In the case of uniform ARA, a data packet arriving at i at time t, is routed according to the probabilities $f_{ij}(t), j \in N(i)$; an FA packet arriving at t is routed uniformly (see Figure 1).

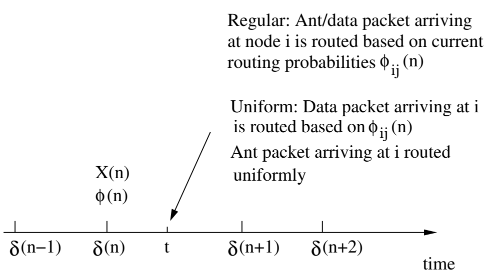


Figure 1: Routing of packet arrivals at a node at time t. Sequence $\{\delta(n)\}$ are the times at which algorithm updates are taking place.

4. ANALYSIS OF THE ROUTING ALGORITHM

We view the routing algorithm, consisting of equations (7) and (8), as a set of discrete stochastic iterations of the type usually considered in the literature on stochastic approximations [14]. We provide below the main convergence result which states that, when ϵ is small enough, the evolution of the vector of delay estimates is closely tracked by an ODE system.

4.1 The ODE approximation

The key observation, which simplifies the analysis, is that there is a time-scale decomposition when ϵ is small enough — the delay estimates X_{ij} then evolve much more slowly compared to the delay processes Δ_{ij}. The probabilities ϕ_{ij} also evolve at the same "time-scale" as the delay estimates (ϕ_{ij} are continuous functions of the delay estimates X_{ij}). Consequently, when ϵ is small enough, with the vector of delay estimates X considered fixed at z (equivalently the vector of routing probabilities fixed at ϕ, the components of ϕ being $\phi_{ij} = \frac{(z_{ij})^{-\beta}}{\sum_{k \in N(i)} (z_{ik})^{-\beta}}$), the delay processes $\{\Delta_{ij}(.)\}$ converge to a stationary distribution, which is dependent on z. Given the $\phi_{ij}, (i,j) \in \mathcal{L}$, and a knowledge of the rates of incoming traffic streams into the queuing network, enable us to determine the total incoming arrival rates into each of the queues Q_{ij}. This can be done by simply solving the flow balance equations; see Bertsekas, Gallager [3], Mitrani [16]. We assume that the total arrival rate is smaller than the service rate of packets in each queue. This assumption (a queue stability assumption) then ensures that, when vector X is considered fixed at z, delay processes $\{\Delta_{ij}(.)\}$ converge to a stationary distribution, which depends on z. We denote the means under stationarity, for each $(i,j) \in \mathcal{L}$, by $D_{ij}(z)$ ($D^U_{ij}(z)$ for the uniform ant case), which is a finite quantity.

[5]except when we are required to be more clear and precise

Also, with delay estimate vector X considered fixed at z, let $\zeta_i(z), i \in \mathcal{N}$, ($\zeta^U_i(z)$ for the uniform ants case) denote, under stationarity, the long-term fraction of FA packets arriving at D that have been launched from i. $\zeta_i(z)$ belongs to the set $(0,1)$ ($\zeta_D(z) = 0, \zeta^U_D(z) = 0$).

Furthermore, when ϵ is small, the evolution of the vector of delay estimates is tracked by an ODE system (an ODE approximation result). This result is not proved in this paper for lack of space, but is available in the technical report [18]. We now introduce some additional notation and state the assumptions under which this result holds. For any fixed $\epsilon \in (0,1)$, and for each (i,j), consider the piecewise constant interpolation of $X^\epsilon_{ij}(n)$ given by: $z^\epsilon_{ij}(t) = X^\epsilon_{ij}(n), \quad n\epsilon \leq t < (n+1)\epsilon, \ n = 0, 1, 2, \ldots$, with the initial value $z^\epsilon_{ij}(0) = X^\epsilon_{ij}(0)$. Consider also the vector-valued piecewise constant process $z^\epsilon(t)$, for all $t \geq 0$, with components $z^\epsilon_{ij}(t)$, $(i,j) \in \mathcal{L}$. Let us consider the increasing sequence of σ-fields $\{\mathcal{F}^\epsilon(n)\}$, where $\mathcal{F}^\epsilon(n)$ encapsulates the entire history of the algorithm for all time $t \leq \delta(n)$. In particular, it contains the σ-field generated by r.v.'s $X^\epsilon(0), X^\epsilon(1), \ldots, X^\epsilon(n)$. It also contains information regarding the arrival times and packet service times, and information regarding the actual routing of packets. The ODE approximation result holds under the following assumptions.

Assumptions:

($A1$) For every $(i,j) \in \mathcal{L}$, and for every $\epsilon \in (0,1)$, the sequence $\{\Delta^\epsilon_{ij}(m)\}$ is uniformly integrable; that is, $\sup_{m \geq 1} E[\Delta^\epsilon_{ij}(m) I_{\{\Delta^\epsilon_{ij}(m) \geq K\}}] \to 0$, as $K \to \infty$.

Regular Ant case.

($A2$) If $X(n)$ is held fixed at z (the sequence $\phi(n)$ is then fixed at a value ϕ; ϕ has components $\phi_{ij} = \frac{(z_{ij})^{-\beta}}{\sum_{k \in N(i)} (z_{ik})^{-\beta}}$) then, for every $l \geq 0$, and for every $(i,j) \in \mathcal{L}$, we have

$$\lim_{r \to \infty} \frac{\sum_{m=l+1}^{l+r} E[I_{\{T(m)=i, R_i(\xi_i(m))=j\}} \Delta_{ij}(\psi_{ij}(m)) | \mathcal{F}(m-1)]}{r} = \zeta_i(z) \phi_{ij} D_{ij}(z), \quad (9)$$

$$\lim_{r \to \infty} \frac{\sum_{m=l+1}^{l+r} E[I_{\{T(m)=i, R_i(\xi_i(m))=j\}} | \mathcal{F}(m-1)]}{r} = \zeta_i(z) \phi_{ij}, \quad (10)$$

the relations above holding almost surely. The quantities $T(n), R_i(n), \Delta_{ij}(n)$, as well as the sequence $\{\mathcal{F}(n)\}$ that appear in the equations above are defined in a similar way as for the case when the delay estimate vector X is time-varying.

($A3$) We assume that the quantities $\zeta_i(z) \phi_{ij} D_{ij}(z)$ and

$\zeta_i(z)\phi_{ij}$ are continuous functions of z.

Uniform Ant case.

$(A2')$ We assume that, if $X(n)$ is held fixed at a value z then, for every $l \geq 0$, and for every $(i,j) \in \mathcal{L}$, we have

$$\lim_{r\to\infty} \frac{\sum_{m=l+1}^{l+r} E[I_{\{T(m)=i, R_i(\xi_i(m))=j\}} \Delta_{ij}(\psi_{ij}(m)) | \mathcal{F}(m-1)]}{r} = \frac{\zeta_i^U(z) D_{ij}^U(z)}{|N(i)|}, \quad (11)$$

$$\lim_{r\to\infty} \frac{\sum_{m=l+1}^{l+r} E[I_{\{T(m)=i, R_i(\xi_i(m))=j\}} | \mathcal{F}(m-1)]}{r} = \frac{\zeta_i^U(z)}{|N(i)|}. \quad (12)$$

$(A3')$ We assume that the quantities $\zeta_i^U(z) D_{ij}^U(z)$ and $\zeta_i^U(z)$ are continuous functions of z.

We have the following convergence theorem which is a central result of this paper. The proof is available in the technical report [18], and follows the approach of Kushner and Yin [14], Chapter 8, Sections 8.1, 8.2.1. We state the result for the regular ARA case. The corresponding result for the uniform ARA case can be similarly stated.

THEOREM 4.1. *Under assumptions* $(A1)$*,* $(A2)$*, and* $(A3)$*, we have the following: there exists a subsequence* $\{\epsilon(k)\}$*, with* $\epsilon(k) \downarrow 0$ *as* $k \to \infty$*, such that the process* $\{z^{\epsilon(k)}(t)\}$ *converges weakly (as* $k \to \infty$*) to a solution* $\{z(t)\}$ *of the ODE system* (13).

For the regular ARA case, $z(t)$, with components $z_{ij}(t)$, $(i,j) \in \mathcal{L}$, is a solution of the ODE system

$$\frac{dz_{ij}(t)}{dt} = \frac{\zeta_i(z(t))(z_{ij}(t))^{-\beta}\Big(D_{ij}(z(t)) - z_{ij}(t)\Big)}{\sum_{k\in N(i)} (z_{ik}(t))^{-\beta}}, \quad \forall (i,j) \in \mathcal{L}, \quad t > 0, \quad (13)$$

with initial conditions $z_{ij}(0) = x_{ij}$, $\forall (i,j) \in \mathcal{L}$.

For the uniform ARA case, $z(t)$, with components $z_{ij}(t)$, $(i,j) \in \mathcal{L}$, is a solution of the ODE system

$$\frac{dz_{ij}(t)}{dt} = \frac{\zeta_i^U(z(t))\Big(D_{ij}^U(z(t)) - z_{ij}(t)\Big)}{|N(i)|}, \quad \forall (i,j) \in \mathcal{L}, \quad t > 0, \quad (14)$$

with initial conditions $z_{ij}(0) = x_{ij}$, $\forall (i,j) \in \mathcal{L}$.

We now briefly discuss the assumptions. A sufficient condition under which $(A1)$ holds is $\sup_{n\geq 1} E\big[\big(\Delta_{ij}^{\epsilon}(n)\big)^{\gamma+1}\big] < \infty$, for some $\gamma > 0$. That is, some moment of the delay higher than the first moment is finite, which we assume. Assumptions $(A2)$ and $(A3)$ can be expected to hold, because they are forms of the strong law of large numbers (they are somewhat weaker because the terms involve conditional expectations). Similar remarks apply for $(A2')$ and $(A3')$.

The dynamic behavior of the algorithm can be studied via the ODE approximation. Numerical solution of the ODE, starting from given initial conditions, requires the computation of the means $D_{ij}(z)$ and the fractions $\zeta_i(z)$ (respectively, $D_{ij}^U(z)$ and $\zeta_i^U(z)$ for the uniform ants case), for given z. In the next subsection, we discuss how to compute these quantities.

4.2 Computations related to the ODE approximation

We assume that, in every queue Q_{ij} the successive service times of both ant (FA) and data packets are i.i.d. exponential with the same mean $\frac{1}{C_{ij}}$ [6]. Furthermore, the service times at each queue are independent of service times at all other queues, and also independent of arrival processes at the nodes. These assumptions are the usual assumptions made for open Jackson networks, and enable us to remain within the domain of solvable models; see, for example, Bertsekas, Gallager [3] and Mitrani [16].

Regular Ant case. In this case, because the ant and data packets are being routed in an identical fashion, we have a single class open Jackson network. Given z, we can compute the routing probabilities $\phi_{ij}, (i,j) \in \mathcal{L}$. The routing probabilities combined with a knowledge of the rates of the incoming streams (ant, data) into the network, enable us to determine the total arrival rate $A_{ij}(z)$ into each queue Q_{ij}. This can be done by simply solving the flow balance equations in the network. For each $(i,j) \in \mathcal{L}$, we assume that $A_{ij}(z) < C_{ij}$ — the arrival rate is smaller than the service rate. Then, under our statistical assumptions, there is a unique joint stationary distribution of the random variables denoting the total number of packets in the queues $Q_{ij}, (i,j) \in \mathcal{L}$. Moreover, this stationary distribution is of a product form. Also, we can compute various quantities of interest to us, like average stationary delays in the queues [3], [16]. Let $w_{ij}(z)$ denote the average stationary delay (sojourn time) in queue Q_{ij}, and let $J_j(z)$ denote the average stationary delay (expected 'cost-to-go') from node j to the destination D, both experienced by an FA packet. $w_{ij}(z)$ is given by the formula, $w_{ij}(z) = \frac{1}{C_{ij} - A_{ij}(z)}$. The quantities $J_i(z), i \in \mathcal{N}$, satisfy the following equations

$$J_i(z) = \sum_{j\in N(i)} \phi_{ij}\Big(w_{ij}(z) + J_j(z)\Big), \quad \forall i \in \mathcal{N}, i \neq D,$$
$$J_D(z) = 0. \quad (15)$$

Once these equations are solved for $J_i(z), i \in \mathcal{N}$, we can compute the quantities $D_{ij}(z), (i,j) \in \mathcal{L}$, using the relations

$$D_{ij}(z) = w_{ij}(z) + J_j(z). \quad (16)$$

Because FAs are generated as a Poisson process with rates λ_i^a at each node i, and because of Assumption $(M2)$, the fraction $\zeta_i(z) = \frac{\lambda_i^a}{\sum_{j\in\mathcal{N}} \lambda_j^a}$ (see Section 7 for a detailed argument).

Uniform Ant case. In this case, the FA packets and the data packets are routed differently. We have an open Jackson network with two classes of traffic, the first class consisting of the FA traffic and the second class consisting of the data traffic. Separate flow balance equations are set up for the two classes of traffic. These flow balance equations enable us to solve for the arrival rates $A_{ij}^a(z)$ and $A_{ij}^d(z)$ of the FA and the data packets into each queue Q_{ij}. The total arrival rate $A_{ij}(z)$ into Q_{ij} is then simply given by the sum $A_{ij}^a(z) + A_{ij}^d(z)$. The average stationary delay $w_{ij}^U(z)$ in queue Q_{ij} is then given by $w_{ij}^U(z) = \frac{1}{C_{ij} - A_{ij}(z)}$. The rest

[6]This amounts to assuming that the average length of a packet (ant or data) is one unit. This is not a restriction, and we can consider the general case by simply multiplying by the average length. However, both ant and data packets must have the same average length.

of the computations which lead to the determination of the quantities $D^U_{ij}(z), (i,j) \in \mathcal{L}$, can be done in a similar manner (with straightforward modifications) as for the regular ants case. Again, because ant packets are generated as a Poisson process at all nodes, and because of Assumption ($M2$), the fraction $\zeta^U_i(z) = \frac{\lambda^a_i}{\sum_{j\in\mathcal{N}} \lambda^a_j}$.

With the knowledge of the quantities $D_{ij}(z), (i,j) \in \mathcal{L}$, and $\zeta_i(z), i \in \mathcal{N}$ (respectively, $D^U_{ij}(z)$ and $\zeta^U_i(z)$ for the uniform ant case), we can numerically solve the ODE (13) ((14) for the uniform ant case), starting from initial condition: $z_{ij}(0), (i,j) \in \mathcal{L}$.

4.3 Equilibrium Behavior of the Routing Algorithm

We now study the equilibrium behavior of the routing algorithm. We denote the equilibrium values of the various quantities by attaching a $*$ to the superscript.

Regular Ant case. Consider the equilibrium points z^* of the ODE system (13). Because the $\zeta_i(z^*)$ are all positive, the points z^* with components z^*_{ij} satisfy the equations

$$\frac{(z^*_{ij})^{-\beta}}{\sum_{k\in N(i)} (z^*_{ik})^{-\beta}}.\Big(D_{ij}(z^*) - z^*_{ij}\Big) = 0, \quad \forall (i,j) \in \mathcal{L}. \quad (17)$$

The interpolated delay estimate vector $z^\epsilon(t)$ approaches the set of equilibrium points z^* asymptotically as $\epsilon \to 0$. More precisely, if E denotes the set of equilibrium points and $N_\delta(E)$ denotes a small enough, δ-neighborhood of E, then asymptotically (as $t \to \infty$), the fraction of time $z^\epsilon(t)$ spends in $N_\delta(E)$ goes to one in probability, as $\epsilon \to 0$ (see Kushner, Yin [14], Section 8.2.1, Theorem 2.1). The vector of routing probabilities $\phi^\epsilon(n)$, being a continuous function of the delay estimate, asymptotically approaches the set of points ϕ^* whose components are given by $\phi^*_{ij} = \frac{(z^*_{ij})^{-\beta}}{\sum_{k\in N(i)} (z^*_{ik})^{-\beta}}$, $(i,j) \in \mathcal{L}$. In what follows, we shall refer to z^*_{ij} as an equilibrium delay estimate, and ϕ^*_{ij} as an equilibrium routing probability, it being understood that the delay estimate $z^\epsilon_{ij}(t)$ and the routing probability $\phi^\epsilon_{ij}(n)$ are asymptotically very close to these quantities with probability close to one, for small enough ϵ.

Because the total arrival rate into every queue is smaller than the packet service rate, the equilibrium delay estimates are finite, and so the equilibrium routing probabilities must be all positive. Consequently, equations (17) reduce to: $D_{ij}(z^*) = z^*_{ij}, \forall (i,j) \in \mathcal{L}$. Now, denoting the functional dependence of the mean stationary delays on the routing probabilities also by $D_{ij}(\phi)$ (a slight abuse of notation), and noting that $\phi^*_{ij} = \frac{(z^*_{ij})^{-\beta}}{\sum_{k\in N(i)} (z^*_{ik})^{-\beta}}$, $\forall (i,j) \in \mathcal{L}$, we find that $\phi^*_{ij}, (i,j) \in \mathcal{L}$, must satisfy the following fixed-point system of equations

$$\phi^*_{ij} = \frac{(D_{ij}(\phi^*))^{-\beta}}{\sum_{k\in N(i)} (D_{ij}(\phi^*))^{-\beta}}, \quad \forall (i,j) \in \mathcal{L}. \quad (18)$$

We check that, for a vector ϕ^*, there is a unique vector with components $D_{ij}(\phi^*), (i,j) \in \mathcal{L}$. To that end, we first notice that, for every $(i,j) \in \mathcal{L}$,

$$D_{ij}(\phi^*) = w_{ij}(\phi^*) + J_j(\phi^*), \quad (19)$$

where $J_j(\phi^*)$ is the expected delay (expected 'cost-to-go') from node j to the destination D experienced by an FA packet when the routing probability vector is ϕ^*; $J_D(\phi^*) = 0$. $w_{ij}(\phi^*)$ is the expected delay along the link (i,j) experienced by an FA packet when the routing probability vector is ϕ^*; we assume that for a given ϕ^*, $w_{ij}(\phi^*)$ is unique [7]. $J_i(\phi^*), i \in \mathcal{N}$, satisfy the following set of equations

$$\begin{aligned} J_i(\phi^*) &= \sum_{j\in N(i)} \phi^*_{ij}\Big(w_{ij}(\phi^*) + J_j(\phi^*)\Big), \ \forall i \in \mathcal{N},\ i \neq D, \\ J_D(\phi^*) &= 0. \end{aligned} \quad (20)$$

Because our equilibrium probabilities ϕ^*_{ij} are all positive, there exists a path from every node i to node D consisting of a sequence of links (i,k_1), …, (k_n, D) for which $\phi^*_{ik_1} > 0$, …, $\phi^*_{k_nD} > 0$. Then, the equations (20) have a unique solution (vector) $J(\phi^*)$, which has components $J_i(\phi^*), i \in \mathcal{N}$ (see Bertsekas and Tsitsiklis [4], Section 4.2, pp. 311-312). Taking note of this and relation (19), we see that for every vector ϕ^*, there is a unique vector of delays $D_{ij}(\phi^*), (i,j) \in \mathcal{L}$.

Also, for any $(i,j) \in \mathcal{L}$, $D_{ij}(\phi^*)$ is a continuous function of the probabilities. (Furthermore, being at least equal to the average service time experienced by an FA packet in the queue Q_{ij}, it is lower bounded by a positive quantity.) Then, by an application of Brouwer's fixed-point theorem, there exists a vector of equilibrium routing probabilities ϕ^* satisfying the fixed-point system (18) (the right-hand side of the fixed-point system maps a compact, convex set — a Cartesian product of probability simplices — to itself).

Uniform Ant case. For the uniform ant case, at equilibrium, the components z^*_{ij} satisfy the following equations

$$\frac{\Big(D^U_{ij}(z^*) - z^*_{ij}\Big)}{|N(i)|} = 0, \quad \forall (i,j) \in \mathcal{L}. \quad (21)$$

We can show in a manner similar to the regular ant case, that the equilibrium routing probabilities must be all positive and must satisfy the fixed-point system of equations

$$\phi^*_{ij} = \frac{(D^U_{ij}(\phi^*))^{-\beta}}{\sum_{k\in N(i)} (D^U_{ij}(\phi^*))^{-\beta}}, \quad \forall (i,j) \in \mathcal{L}. \quad (22)$$

Also, we can show that, for a vector of equilibrium routing probabilities ϕ^* there is a unique vector with components $D^U_{ij}(\phi^*), (i,j) \in \mathcal{L}$. Also there exists a solution to the set of fixed-point equations (22), by a similar application of Brouwer's fixed-point theorem.

5. EXAMPLE: AN ACYCLIC NETWORK

In this section we consider the acyclic network of Figure 2. The numbers beside the links indicate the link capacities (C_{ij} units for link (i,j)). Data packets arrive at nodes 1, 2 and 3 as Poisson processes with rates λ^d_1, λ^d_2, and λ^d_3. Ant packets come in as a Poisson process at node i with rate λ^a_i ($i = 1,\ldots,7$).

We carried out a discrete event simulation of the network and present results for the regular ARA case. The arrival rates of the streams are as follows: $\lambda^a_i = 2, i = 1,\ldots,7$, and

[7]We have a similar abuse of notation for w_{ij} and J_j as we had for D_{ij}. In Section 4.2, we had denoted by $w_{ij}(z)$ and $J_j(z)$ the average stationary delay in queue Q_{ij}, and the average stationary delay (expected 'cost-to-go') from node j to D, both experienced by an FA packet, with the delay estimate vector considered fixed at z.

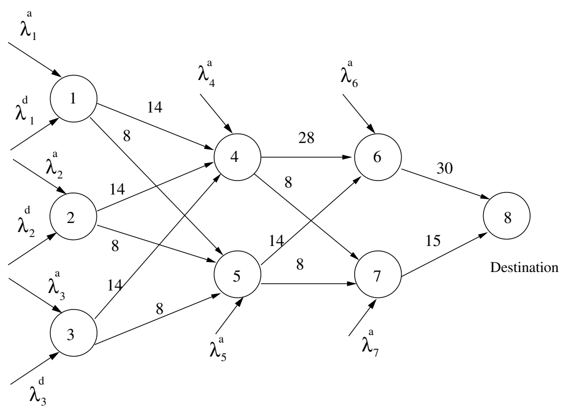


Figure 2: An Acyclic Network

$\lambda_1^d = 6$, $\lambda_2^d = 8$, and $\lambda_3^d = 6$. The parameter $\beta = 1$, and the step size $\epsilon = 0.002$.

The approximating ODE system is given by (13). The ant arrival rates λ_i^a being all equal, $\zeta_i(z) = \frac{1}{7}, i = 1, \ldots, 7$. Delay estimate for route $1 \to 4 \to \cdots \to 8$ is approximated by the component $z_{14}(t)$ which follows the equation

$$\frac{dz_{14}(t)}{dt} = \frac{(z_{14}(t))^{-1}\Big(D_{14}(z(t)) - z_{14}(t)\Big)}{7\Big((z_{14}(t))^{-1} + (z_{15}(t))^{-1}\Big)},\ t > 0.$$

Delay estimates for route $1 \to 5 \to \cdots \to 8$, and for other routes $2 \to j \to \cdots \to 8, j = 4, 5,\ 3 \to j \to \cdots \to 8, j = 4, 5$, are approximated by corresponding components, which follow similar equations. The delay estimate for route $4 \to 6 \to 8$ is approximated by $z_{46}(t)$ which follows the equation

$$\frac{dz_{46}(t)}{dt} = \frac{(z_{46}(t))^{-1}\Big(D_{46}(z(t)) - z_{46}(t)\Big)}{7\Big((z_{46}(t))^{-1} + (z_{47}(t))^{-1}\Big)},\ t > 0.$$

Delay estimates for route $4 \to 7 \to 8$, and for routes $5 \to j \to 8, j = 6, 7$, are approximated by components that follow similar equations. Finally, the delay estimate for link $6 \to 8$ is approximated by $z_{68}(t)$ which follows the equation

$$\frac{dz_{68}(t)}{dt} = \frac{D_{68}(z(t)) - z_{68}(t)}{7},\ t > 0.$$

Delay estimate for link $7 \to 8$ is approximated by $z_{78}(t)$, which follows a similar equation. For each z, computation of the means $D_{ij}(z)$, that appear in the ODE expressions above, can be accomplished following Section 4.2. We solve the above ODE system numerically, starting from certain initial conditions $z_{ij}(0) = x_{ij}, (i, j) \in \mathcal{L}$.

Figures 3a, 3c provide plots of the interpolated delay estimates $z_{14}^\epsilon(t)$, $z_{46}^\epsilon(t)$, and alongside plots of the corresponding components of the ODE system. The ODE approximation tracks the interpolated delay estimates well. Figures 3b and 3d provides plots of the routing probabilities $\phi_{14}^\epsilon(n)$ and $\phi_{46}^\epsilon(n)$, respectively. We note that though we initially start with a routing probability $\phi_{14}^\epsilon(0) < 0.5$, the routing probability $\phi_{14}^\epsilon(n)$ converges to a value which is greater than 0.5. This is to be expected of a routing algorithm; the (equilibrium) routing probability on outgoing links that lie on paths with higher capacity links should be higher.

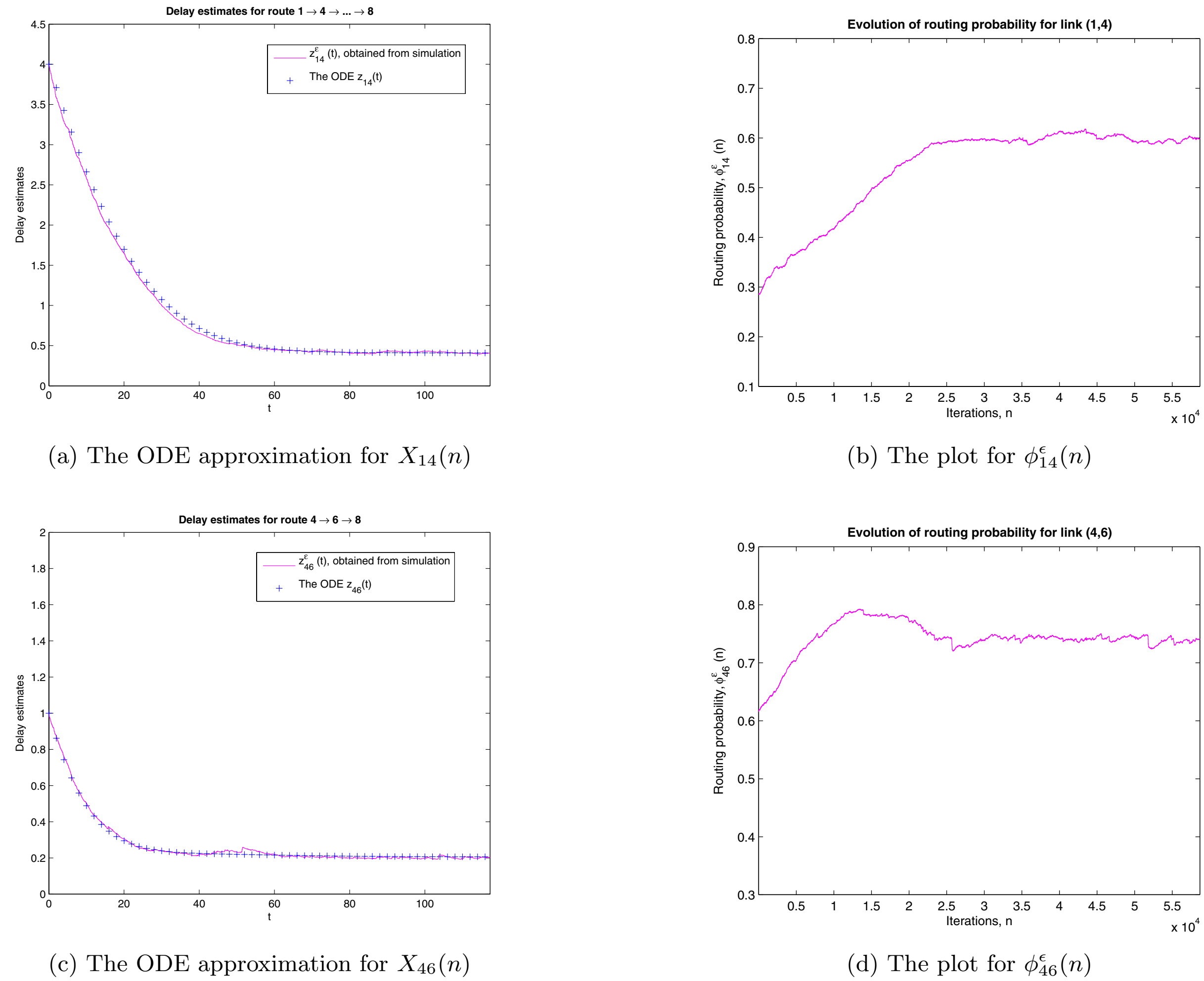
6. CONCLUDING REMARKS

Extensions. We can extend our results to the case when we have an acyclic network, with multiple destinations for the incoming data traffic. As usual, at every node, ant (FA) packets are sent out to explore the delays in the paths towards each destination. The ant packets can be routed using either the regular or the uniform ARA algorithm. Suppose that there are M destinations overall. With assumptions $(M1)$ and $(M2)$ regarding the operation of the algorithm in force, we can write down the stochastic iterative equations, describing the evolution of the delay estimates and the routing probabilities, in a form similar to equations (7) and (8). Let us now consider first the case when the queue Q_{ij} associated with link (i, j), is shared by all ant and data packets that are bound for various destinations. The scheduling discipline is FIFO. In this case it can be checked that, we would again have an ODE approximation similar in form to (13) for the regular ARA case ((14) for the uniform ARA case). There is a set of equations for each of the M destinations, and the equations considered together constitute a system of coupled ODEs. In order to compute the stationary means of the delays — $D_{ij}(z)$, for a given z — related to the ODE approximation, we can employ the same procedure as in Section 4.2, with appropriate modifications. In this regard we note that we again have an open Jackson network, with M classes for the regular ARA case, and with $M+1$ classes for the uniform ARA case (data packets are routed according to the node routing probabilities and ant packets are routed uniformly). Also, the equilibrium behavior of the routing algorithm can be described as in Section 4.3.

The second more general case is a per-destination queuing arrangement, which is more appropriate in a routing context. In this case, for a link (i, j), M separate outgoing queues $Q_{ij}^k, k = 1, \ldots, M$, are maintained. Q_{ij}^k holds FA and data packets that are bound for destination k. The transmission capacity of link (i, j) is then shared between the queues; the manner in which the sharing takes place is known as the link scheduling discipline. In this case, the form of the update algorithms does not change, and we can arrive at an ODE approximation for the system as described above for the first case. However, in this case, it may not be always possible to compute analytically the stationary mean delays. Only for certain *symmetric* link scheduling disciplines like Processor-Sharing, which are analytically tractable (that is, have joint stationary product form distributions for the number of packets in the queues; see [16]), can we compute the stationary mean delays.

Also, in our framework, we can consider a slightly more general dependence of outgoing routing probabilities on the delay estimates: $\phi_{ij} = \frac{g(X_{ij})}{\sum_{k \in N(i)} g(X_{ik})}$, where g is a continuous function, that is positive real-valued, and nonincreasing. The analysis remains the same. An example of g is $g(x) = e^{-\beta x}, x \geq 0$, where β is a positive integer.

Conclusions and Future Directions. In summary, in this paper we have studied the convergence and the equilibrium behavior of an ARA algorithm for wireline, packet-switched networks. We have considered acyclic network models, where there are multiple sources of incoming data traffic whose packets are bound for specified destinations. We have considered stochastic models for the arrival processes and packet lengths for the ant and data streams. We have shown that the evolution of the vector of delay esti-



(a) The ODE approximation for $X_{14}(n)$

(b) The plot for $\phi^{\epsilon}_{14}(n)$

(c) The ODE approximation for $X_{46}(n)$

(d) The plot for $\phi^{\epsilon}_{46}(n)$

Figure 3: ODE approximations and plots of routing probabilities

mates can be tracked by an ODE system when the step-size of the estimation scheme is small. We then study the equilibrium routing behavior. We observe that, at equilibrium, the routing probabilities are higher for outgoing links that lie on paths with higher capacity links.

There are certain advantages of ARA algorithms that are worth pointing out. ARA algorithms do not require explicit knowledge of the incoming traffic rates into the network, or a knowledge of the link capacities. Instead, they rely directly on online information of path delays in the network, that are collected by ant packets. This enables the algorithm to adapt to changes in the incoming traffic rates, and/or changes in the network topology. On the other hand, because there is a learning process to ascertain the path delays (based on which the routing probabilities are updated), the convergence of the algorithm can be slow.

In our work we have considered models where are no cycles in the network. It remains to study convergence and equilibrium behavior of the algorithm when there are cycles. There are two issues that arise. First, cycles in the network affect adversely the process of estimation of path delays by the ant packets. This is because the estimates can grow unbounded if there is a positive probability of an ant packet being routed in a cycle. Second, it might happen that we converge to an equilibrium routing solution which has *loops*. That is, for a given destination k, the equilibrium routing probabilities might be such that, for a sequence of links $(i_1, i_2), \ldots, (i_{n-1}, i_n), (i_n, i_1)$ that forms a cycle, $\phi^k_{i_1 i_2} > 0, \ldots, \phi^k_{i_{n-1} i_n} > 0, \phi^k_{i_n i_1} > 0$. There is no reason to believe that the scheme that we analyse in this paper can lead to a loop-free equilibrium solution. For the case when the network has cycles, we might need to modify the scheme so that it can converge to a loop-free routing solution, which is desirable.

7. APPENDIX: EXPRESSION FOR $\zeta_I(Z)$

We show here that $\zeta_i(z) = \frac{\lambda^a_i}{\sum_{j \in \mathcal{N}} \lambda^a_j}$, for each $i \in \mathcal{N}$, for the regular ARA case. The same argument holds for the uniform ARA case. As discussed in Section 4.2, with the delay estimate vector X considered fixed at z, we have a single class open Jackson network. For each queue Q_{ij} with the arrival rate of packets $A_{ij}(z) < C_{ij}$, the queuing network converges to stationarity. Let $T_{ij}, (i,j) \in \mathcal{L}$, denote the total number of packets in the queues Q_{ij} under stationarity. Let $\{R_n\}$ denote the sequence of times when $T_{ij}, (i,j) \in \mathcal{L}$, returns to the state consisting of all zeros. Thus, $\{B_n\}$, where

$B_n = R_n - R_{n-1}$, constitutes the sequence of successive busy periods for the queuing network. Under our assumptions on the statistics of the arrival processes and the packet lengths of the various streams, $\{B_n\}$ is an i.i.d. sequence, with the mean $E[B_n] < \infty$. $\{R_n\}$ is a sequence of stopping times for the ant Poisson arrival processes at the nodes.

For each $i \in \mathcal{N}$, let $D_i(t) =$ Number of FA packets that arrive at destination D by time t. Then

$$\zeta_i(z) = \lim_{t\to\infty} \frac{D_i(t)}{\sum_{j\in\mathcal{N}} D_j(t)}. \tag{23}$$

Furthermore, we have

$$\lim_{t\to\infty} \frac{D_i(t)}{\sum_{j\in\mathcal{N}} D_j(t)} = \frac{E[D_i(B_n)]}{\sum_{j\in\mathcal{N}} E[D_j(B_n)]}. \tag{24}$$

This is intuitive, and can be established by using the Renewal Reward Theorem, with the inter-renewal times being the sequence $\{B_n\}$.

Now, because $D_i(B_n) =$ Number of ant Poisson arrivals at node i in the interval B_n, the mean $E[D_i(B_n)] = \lambda_i^a E[B_n]$, and so

$$\zeta_i(z) = \frac{\lambda_i^a}{\sum_{j\in\mathcal{N}} \lambda_j^a}. \tag{25}$$

8. REFERENCES

[1] J. S. Baras and H. Mehta. A probabilistic emergent routing algorithm for mobile adhoc networks. In *Proc. WiOpt '03: Modeling and Optimization in Mobile, Adhoc and Wireless Networks*, mar 2003.

[2] N. Bean and A. Costa. An analytic modeling approach for network routing algorithms that use 'ant-like' mobile agents. *Computer Networks*, 49(2):243–268, oct 2005.

[3] D. P. Bertsekas and R. G. Gallager. *Data Networks*. 2nd Edition, Prentice Hall, Englewood Cliffs, NJ, 1992.

[4] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, Englewood Cliffs, NJ, 1989.

[5] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Santa Fe Inst. Studies in the Sciences of Complexity, Oxford University Press, New York, 1999.

[6] V. S. Borkar and P. R. Kumar. Dynamic cesaro-wardrop equilibration in networks. *IEEE Trans. on Automatic Control*, 48(3):382–396, mar 2003.

[7] G. D. Caro and M. Dorigo. Antnet: Distributed stigmergetic control for communication networks. *Jl. of Artificial Intelligence Res.*, 9:317–365, dec 1998.

[8] D. J. Das and V. S. Borkar. A novel aco scheme for optimization via reinforcement and initial bias. *Swarm Intelligence*, 3(1):3–34, mar 2009.

[9] J. L. Deneubourg, S. Aron, S. Goss, and J. M. Pasteels. The self-organizing exploratory pattern of the argentine ant. *Jl. of Insect Behavior*, 3(2):159–168, mar 1990.

[10] M. Dorigo and T. Stutzle. *Ant Colony Optimization*. The MIT Press, Cambridge, MA, 2004.

[11] E. Gabber and M. Smith. Trail blazer: A routing algorithm inspired by ants. In *Proc. of the Intl. Conf. on Networking Protocols (ICNP)*, pages 36–47. IEEE, oct 2004.

[12] W. J. Gutjahr. A generalized convergence result for the graph-based ant system metaheuristic. *Probability in the Engineering and Informational Sciences*, 17(4):545–569, oct 2003.

[13] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Jl. of Artificial Intelligence Res.*, 4:237–285, may 1996.

[14] H. J. Kushner and G. G. Yin. *Stochastic Approximation Algorithms and Applications*. Appl. of Math. Series, Springer Verlag, New York, 1997.

[15] A. M. Makowski. The binary bridge selection problem: Stochastic approximations and the convergence of a learning algorithm. In *Proc. ANTS, Sixth Intl. Conf. on Ant Col. Opt. and Swarm Intelligence, LNCS 5217, M. Dorigo et. al. (eds.), Springer Verlag*, pages 167–178, sep 2008.

[16] I. Mitrani. *Probabilistic Modeling*. Cambridge University Press, London, 1998.

[17] P. Purkayastha and J. S. Baras. Convergence results for ant routing algorithm via stochastic approximation and optimization. In *Proc. IEEE Conf. on Decision and Control*, pages 340–345. IEEE, dec 2007.

[18] P. Purkayastha and J. S. Baras. Convergence results for ant routing algorithms via stochastic approximation. *Institute of Systems Research (ISR) Technical Report*, oct 2009.

[19] D. Subramanian, P. Druschel, and J. Chen. Ants and reinforcement learning: A case study in routing in dynamic networks. In *Proc. of IJCAI(2)1997: The Intl. Jt. Conf. on Artificial Intelligence*, pages 832–839, aug 1997.

[20] M. A. L. Thathachar and P. S. Sastry. *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. Kluwer Academic Publishers, Norwell, MA, 2004.

[21] J. H. Yoo, R. J. La, and A. M. Makowski. Convergence results for ant routing. In *Proc. Conf. on Information Sciences and Systems*, mar 2004.