Decentralized Stochastic Control of Delay Tolerant Networks

Eitan Altman, Giovanni Neglia
INRIA
2004 Route des Lucioles, Sophia-Antipolis (France)
email: name.surname@sophia.inria.fr

Francesco De Pellegrini, Daniele Miorandi
CREATE-NET
Via alla Cascata 56/D, Povo, Trento (Italy)
email: name.surname@create-net.org

***Abstract*—We study in this paper optimal stochastic control issues in delay tolerant networks. We first derive the structure of optimal two-hop forwarding policies. In order to be implemented, such policies require the knowledge of some system parameters such as the number of mobiles or the rate of contacts between mobiles, but these could be unknown at system design time or may change over time. To address this problem, we design adaptive policies combining estimation and control that allow to achieve optimal performance in spite of the lack of information. We then study interactions that may occur in the presence of several competing classes of mobiles and formulate this as a cost-coupled stochastic game. We show that this game has a unique Nash equilibrium where each class adopts the optimal forwarding policy determined for the single class problem.**

***Index Terms*—Stochastic Control, Game Theory, Delay Tolerant Networks**

I. Introduction

Delay–Tolerant Networks (DTNs) are sparse and/or highly mobile wireless ad hoc networks where no continuous connectivity guarantee can be assumed [1], [2]. One central problem in DTNs is related to the routing of packets towards the intended destination. Protocols developed in the mobile ad hoc networks field, indeed, fail since a complete route to the destination may not exist most of the time. One common technique for overcoming such problem is to disseminate multiple copies of the message in the network, enhancing the probability that at least one of them will reach, within a suitable time-frame, the destination node [3]. This is referred to as epidemic-style forwarding [4]. Alike the spread of infectious diseases, each time a message-carrying node encounters a new node not having a copy thereof, it may *infect* this new node by passing on a message copy; newly infected nodes, in turn, may behave similarly. The destination receives the message when meets an infected node.

In this paper we consider the zero knowledge scenario [5], [6], where mobile nodes have no *a priori* information on the encounter pattern. Moreover we constrain the analysis to the case when the source of the message can copy it, while the other infected nodes can only forward it to the destination. This is referred to as two-hop forwarding [7]. We investigate the problem of optimal stochastic control of such routing protocol. The control variable is the probability of transmitting a message upon a suitable transmission opportunity (i.e., contact). The goal is to optimize the probability to deliver a message, while satisfying specific energy constraints. The main contributions of our work summarizes as follows:

- We introduce a discrete–time framework to model message diffusion in DTNs; within such framework, we characterize analytically the structure of optimal policies for routing control using sample path techniques. In particular, threshold policies are proved optimal.
- We introduce online estimation algorithms so that nodes can learn online optimal policies in a-priori unkwon network scenarios. These algorithms are based on stochastic approximation theory. Convergence to the optimal control policies, under suitable conditions, is analytically derived.
- We extend the problem of optimal control to the case of several competing classes of mobile terminals. The framework, in this case, is that of cost–coupled stochastic games [8], [9]. We prove that the game has a unique Nash equilibrium where each class adopts the optimal forwarding policy determined for the single class problem.

Simulations confirm analytical results and provide further insights.

The control of forwarding schemes has been addressed in DTNs literature before. In [10], the authors propose an epidemic forwarding protocol based on the Susceptible-Infected-Removed (SIR) model [11] and show that it is possible to increase the message delivery probability by tuning the parameters of the underlying SIR model. In [12] a detailed general framework is proposed in order to capture the relative performances of different adaptive strategies. None of these two papers formalize a specific optimization problem. In [5] and its follow-up [6], the authors assume the presence of a set of special mobile nodes, the ferries, whose mobility can be controlled. Algorithms to design ferry routes are proposed in order to optimize network performance. Works more similar to ours are [13], [14], [15]. In [13] the authors consider buffer constraints and derive, based on some approximations, buffer scheduling policies in order to minimize the delivery time. The optimization goal in [14] can be considered a relaxed version of our problem (e.g., the weighted sum of delivery time and energy consumption), also in this case the optimal policy is a threshold one. Also, under a fluid model approximation, the work in [15] provides a general framework for the optimal control of a broad class of so called monotone relay strategies.

Apart from the differences in the optimization functions, most of the above works do not address the problem of online estimation of optimal policies; an attempt is done in [12], [13] based on some heuristics for the estimation.

Finally, to the best of our knowledge, this is the first formulation of a game with competing nodes in a DTN scenario.

The remainder of the paper is organized as follows. The system model is introduced in Sec. II. The structure of optimal control policies is derived in Sec. III. Methods for optimization in the presence of unknown system's parameters are presented in Sec. IV. The multiclass case is introduced in Sec. V. Numerical results are presented in Sec. VI. Sec. VII concludes the paper.

II. System Model

Consider a network of $N+1$ mobile nodes, each equipped with some form of proximity wireless communications. The network is assumed to be sparse, so that, at any time instant, nodes are isolated with high probability. Communication opportunities arise whenever, due to mobility patterns, two nodes get within mutual communication range. We refer to such events as "contacts".

The time between subsequent contacts of any pair of nodes is assumed to follow an exponential distribution with parameter $\lambda > 0$. The validity of this model for synthetic mobility models (including, e.g., Random Walk, Random Direction, Random Waypoint) has been discussed in [16]. There exist studies based on traces collected from real-life mobility [17] that argue that inter-contact times may follow a power-law distribution. Recently, the authors of [18] have shown that these traces and many others exhibit exponential tails after a cutoff point. For this reason, we choose to stick with the exponential meeting time assumption, which makes our analysis tractable.

There can be multiple source-destination pairs, but we assume that at a given time there is a single message, eventually with many copies, spreading in the network[1]. For simplicity we consider a message originated at time $t = 0$. We assume that the message that is transmitted is relevant during some time τ. This applies, e.g., to environmental information or data referring to events of transient nature (e.g., happenings). The message contains a time stamp reporting its generation time, so that it can be deleted at all nodes when it becomes irrelevant. We do not assume any feedback that allows the source or other mobiles to know whether the message has been successfully delivered to the destination within the time τ.

We focus on a set of relaying strategies that can be defined as probabilistic two-hop routing strategies. At each encounter between the source and a mobile that does not have the message, the message is relayed with some probability taking values in $U = [u_{\min}, u_{\max}]$. If a mobile different from the source has a copy of the message then it transfers it only to the destination.

We adopt a discrete time model, considering a time slot duration Δ. The n–th slot corresponds to interval $[n\Delta, (n+1)\Delta)$ and the number of slots is equal to $K = \lfloor \tau/\Delta \rfloor$. In this discrete time model, we assume that a mobile that receives a copy during a time slot can forward it starting from the following time slot. Moreover the forwarding probability during $[n\Delta, (n+1)\Delta)$ is a constant and it is denoted by u_n.

Let X_n be the number of mobiles, not including the destination, that have a copy of the message at time $n\Delta$ (i.e. at the beginning of the n–th slot), $X_0 = 1$. Under the assumptions above, X_n is a Markov chains with possible states $1, 2, \cdots N$. The transition rates depend on the forwarding probability used by the source in each time slot, so a natural way to optimize performance system is to control such forwarding probabilities.

The problem we address in this paper is to *maximize the probability to deliver the message* by the K-th time slot, *under a constraint on the expected number of infected nodes*. The number of infected nodes is related to the total energy consumption. In particular they are simply proportional if we assume that i) for each transmission a constant amount of energy is consumed, ii) all the other activities require a negligible amount of energy. We want to determine optimal time-dependant forwarding policies the source can implement. More formally we define a forwarding policy (control policy) as a function $\mu : \{0, 1, 2, \cdots K-1\} \to U$.

In what follows a key role will be played by two types of forwarding policies, *static* and *threshold* policies, defined as follows:

Definition 2.1: A policy μ is a *static* policy if μ is a constant function, i.e. $\mu(n) = p \in U$, for $n = 0, 1, 2, K-1$. A policy μ is a *threshold* policy, if there exist $h \in \{0, 1, 2, \cdots K-1\}$ (the threshold) such that

$$\mu(n) = \begin{cases} u_{\max}, & \text{if} \quad n < h \\ u_{\min}, & \text{if} \quad n > h \end{cases} \tag{1}$$

Observe that static and threshold policies are identified by a few parameters: the control p for static policies, and the threshold h and the corresponding value $\mu(h)$ for dynamic policies, which leads to a simple implementation. With static policies, at each communication opportunity, the message is forwarded with a constant probability p. Conversely, with threshold policies, each time a mobile has a forwarding opportunity, it checks the time t elapsed since the message generation time and it forwards the message with some probability $u(t)$, i.e. they require a dynamic approach[2].

It is worth noticing that static and threshold policies are defined based on few parameters only, i.e., the control p for static policies, and the threshold h and the corresponding value $\mu(h)$ for dynamic policies, which leads to a simple implementation.

[1]Results in sections III and IV are valid even for multiple messages at the same time, but we assume that the bandwidth and the buffer are large enough to assure that the different propagation processes are independent.

[2]Incidentally, time t can be traced just summing up the time elapsed at each node with no need for nodes' synchronization.

Symbol	*Meaning*
$N+1$	number of nodes
λ	pairwise intermeeting intensity
τ	timeout value
K	$\lfloor \tau/\Delta \rfloor$
Δ	time slot
X_n	number of nodes having a copy of the message at time $n\Delta$
Ψ	maximum expected number of infected nodes
$F_D(n)$	probability that the message is delivered by time $n\Delta$
$\mu(\cdot)$	control policy
u_n	value taken by the control variable (i.e., forwarding probability) at time $n\Delta$
p	value taken by the control variable under static policy
h	time threshold
θ	$= \sum_{k=0}^{K-1} u_k$
β	θ value for the optimal policy
$\zeta_{n,m}(j)$	indicator that the j–th mobile, among the $N - X_n$ ones that do not have the message at time $n\Delta$, receives it during the next m slots
$Q_{n,m}$	probability that a mobile does not receive the message during time slots $n, n+1, ..., n+m-1$
$\gamma_n(s)$	$= \mathrm{E}[\exp(-s\xi_{0,n}(1)]$
$X_n^*(s)$	Laplace-Stieltjes transform of X_n
$\overline{X}_m$	estimate of $\mathrm{E}[X_K]$ at the m-th round of the stochastic approximation algorithm
$\Pi_H(u)$	projection of the value u on the interval H
$\{\cdot\}^{(i)}$	superscrit indicates that the quantity refers to the i-th class of mobile nodes
$Y_n^{(i)}$	number of class i infected nodes that can transmit to the destination during the n-th time slot
S_n	total number of infected nodes that can transmit to the destination during the n-th time slot
$S_n^{(-i)}$	total number of infected nodes that can transmit to the destination during the n-th time slot but class i ones

TABLE I
NOTATION USED THROUGHOUT THE PAPER

In the following section we characterize optimal static and threshold policies. Then in Sec. IV we show how the source can *learn* online the optimal policy. In Table I the notation used throughout the paper is reported.

III. CHARACTERIZATION OF OPTIMAL POLICIES

We define $F_D(n)$ the probability that a message generated at time 0 is received before $n\Delta$, i.e. $F_D(\cdot)$ is the CDF of the message delay (considering the messages not delivered by τ as delivered at $t = \infty$).

We want to derive policies that maximize $F_D(K)$, while satisfying the following constraint on the expected number of infected nodes: $\mathrm{E}[X_K] \le \Psi$.

We first characterize the evolution of X_n. Let $\zeta_{n,m}(j)$ be the indicator that the j–th mobile among the $N - X_n$ mobiles that do not have the message at time $n\Delta$, receives the message during $(n\Delta, (n+m)\Delta]$. Then we have

$$X_{n+m} = X_n + \sum_{j=1}^{N-X_n} \zeta_{n,m}(j). \tag{2}$$

Variables $\zeta_{n,m}(j)$ are i.i.d. Bernoulli random variables with expected value:

$$\mathrm{E}\left[\zeta_{n,m}(j)\right] = 1 - \exp(-\lambda\Delta \sum_{k=n}^{m-1} u_k) = 1 - Q_{n,m}, \tag{3}$$

where $Q_{n,m}$ is then the probability that a mobile does not receive the message in time slots $n, n+1, ..., n+m-1$. We observe that $\zeta_{n,m}(j)$ are stochastically increasing in the control actions u_k (see [19] for definition and properties of usual stochastic order). More formally, given a policy μ, consider the policy μ' such that $\mu'(n) = \mu(n)$ for $n \neq k$ and $\mu'(k) > \mu(k)$, and denote as $\zeta'_{n,m}(j)$, X'_n and $F'_D()$ respectively the indicator variables, the number of infected nodes and the delivery probability function when policy μ' is applied, then

$$\zeta'_{n,m}(j) >_{st} \zeta_{n,m}(j) \; \forall n < k \text{ and } m > k.$$

Moreover being that the number of infected nodes X_n (X'_n) can be obtained as sum of the indicator variables $\zeta_{0,n}(j)$ ($\zeta'_{0,n}(j)$) (Eq. (2)), it holds

$$X'_n >_{st} X_n, \; \forall n > k.$$

This formalizes the intuition that the higher the forwarding probability the higher the number of infected nodes (the same conclusion can be reached through a simple sample path reasoning).

From the previous equations we can easily derive the expected value of X_n, that will be used in the next section:

$$\mathrm{E}[X_n] = X_0 + (N - X_0)\left(1 - Q_{0,n}\right) \tag{4}$$

Using the Laplace Stieltjes Transform of X_n, $X_n^*(s) := \mathrm{E}[\exp(-sX_n)]$, we can derive the following useful formula for $F_D(n)$:

$$F_D(n) = 1 - \prod_{i=0}^{n-1} X_i^*(\lambda\Delta). \tag{5}$$

In order to prove (5), let us define $G(n) = 1 - F_D(n\Delta)$, then it follows

$$\begin{aligned} G(n+1) &= G(n) \; \mathrm{Pr}\{\text{no delivery in the n-th slot}|X_n\} \\ &= G(n) \; \mathrm{E}\Big[\mathrm{Pr}\big\{\text{no delivery in the n-th slot}|X_n\big\}\Big] \\ &= G(n) \; \mathrm{E}\Big[\exp(-\lambda\Delta X_n)\Big] = G(n) X_n^*(\lambda\Delta) \\ &= \prod_{i=0}^{n} X_i^*(\lambda\Delta) \end{aligned} \tag{6}$$

From Eq. (5), and above considerations on stochastic orderings, it follows that the delivery probability and the final number of infected nodes are increasing in the control actions u_k. Formally,

Proposition 3.1: Given two policies μ and μ', defined as above, it holds: $F_D(K) < F'_D(K)$, $\mathrm{E}[X_K] < \mathrm{E}[X'_K]$.

A consequence of this proposition is the following corollary:

Corollary 3.1: If an optimal policy exists, either it is the static policy μ_{max} with $\mu_{max}(n) = u_{max}, \; \forall n$, or it saturates the constraint, i.e. $\mathrm{E}[X_K] = \Psi$.

The proofs of the above statements are reported in [20].

We observe that the set of admissible policies could be empty. It can be verified that this happens if and only if the static policy $\mu_{min}(n) = u_{min} \forall n$, does not satisfy the constraint.

In what follows we are going to assume that admissible policies exist and we are going to characterize policy optimality. To this purpose it is useful to derive an explicit formula for the Laplace Stieltjes transform. Let us introduce

$$\begin{aligned}\gamma_n(s) &:= \mathrm{E}[\exp(-s\zeta_{0,n}(1))] = (1-Q_{0,n})\exp(-s) + Q_{0,n} \\ &= e^{-s} - (1-e^{-s})\exp\left(-\lambda\Delta\sum_{k=0}^{n} u_k\right) \end{aligned} \quad (7)$$

Then $X_n^*(s)$ can be expressed as a function of $\gamma_n(s)$ as follows:

$$\begin{aligned} X_n^*(s) &= \mathrm{E}[e^{-sX_n}] = \mathrm{E}\left[\exp\left(-s\left(X_0 + \sum_{i=1}^{N-X_0}\zeta_{0,n}(i)\right)\right)\right] \\ &= e^{-sX_0}\left(\mathrm{E}\left[\exp(-\zeta_{0,n}(1))\right]\right)^{N-X_0} = \\ &= e^{-sX_0}\gamma_n(s)^{N-X_0} \end{aligned} \quad (8)$$

We can now introduce the main result of this section.

Theorem 3.1: There exists an optimal threshold policy. A non threshold policy is not optimal.

Proof: The existence of an optimal policy follows from elementary properties of Markov decision processes (see for example [21]). We need simply to prove that a non threshold policy cannot be optimal.

Let us consider a non threshold policy μ that satisfies the constraint ($\mathrm{E}[X_K] \leq \Psi$), then there exists some time $k < K$ and some $\epsilon > 0$ such that $u_k < u_{\max} - \epsilon$ and $u_{k+1} > u_{\min} + \epsilon$.

Let μ' be the policy obtained from μ by setting $u'_k = u_k + \epsilon$ and $u'_{k+1} = u_{k+1} - \epsilon$ (the other components are the same as those of μ). We denote with X'_n, $\gamma'_n(s)$, ${X'}_n^*(s)$ and $F'_D(\cdot)$ the quantities corresponding to μ'.

We notice that $\gamma'_n(s) = \gamma_n(s)$ for $n \neq k$ and $\gamma'_k(s) = \gamma_k(s)\exp(-\lambda\Delta\epsilon) < \gamma_k(s)$. Then from Eq. (8), it follows that ${X'}_n^*(s) = X_n^*(s)$ for $n \neq k$, while ${X'}_k^*(s) < X_k^*(s)$, which in turn brings $F'_D(n\Delta) > F_D(n\Delta)$ for $n \geq k$. Moreover ${X'}_K^*(s) = X_K^*(s)$ implies that $\mathrm{E}[X'_K] = \mathrm{E}[X_K] \leq \Psi$, then the new policy satisfies the constraint and improves the delivery probability. Hence a non threshold policy μ cannot be optimal. ∎

Let us now determine the optimal threshold policy. Due to Corollary 3.1, the optimal policy is μ_{max} if it satisfies the constraint. Otherwise, the constraint has to be saturated and we can obtain the threshold value from Eq. (4), imposing $\mathrm{E}[X_K] = \Psi$:

$$Q_{0,K} = \frac{N-\Psi}{N-X_0}.$$

Hence

$$\sum_{k=0}^{K-1} u_k = -\frac{1}{\lambda\Delta}\log\left(\frac{N-\Psi}{N-X_0}\right) =: \beta \quad (9)$$

This directly yields the threshold h^* of the optimal policy, by considering that $u_n = u_{max}$ for $n < h^*$ and $u_n = u_{min}$ for $n > h^*$ while satisfying Eq. (9). Then $h^* = \max\{h \in \mathbb{N} : v(h) = h \cdot u_{\max} + (K-h)\cdot u_{\min} \leq \beta\}$, and $u_{h^*} = \beta - v(h^*)$. In the particular case of $u_{min} = 0$, this reduces to $h^* = \lceil \beta \rceil$ and $v(h^*) = \beta - \lfloor \beta \rfloor$.

The same reasoning can be applied to determine the best static policy. In particular it is μ_{max}, if μ_{max} satisfies the constraint (and in such case the best static policy is also the optimal one), otherwise Eq. (9) holds, and imposing $u_k = p^*$ for all k, we obtain $p^* = \beta/K$.

IV. Stochastic Approximations for Adaptive Optimization

In this section we introduce methods for achieving the optimal control policies in the case where some parameters (i.e., N and λ) are unknown. We show that simple iterative algorithms may be implemented at each node, allowing them to discover the optimal policy in spite of the lack of information on such parameters[3].

Our approach is based on stochastic approximation theory [22]. This framework generalizes Newton's method to determine the root of a real-valued function when only noisy observations of such function are available.

We consider the two optimization problems:

- Static control: find the constant $p^* \in [u_{\min}, u_{\max}]$ such that the policy $\mu = p^*$ has the best performance among all static policies.
- Dynamic control: find the threshold $h^* \in \{0, 1, \cdots, K-1\}$ and $\mu(h^*)$ characterizing the optimal policy.

We can approach online estimation of static and dynamic control in the same way. Let us denote $\theta = \sum_{k=0}^{K-1} u_k$, the sum of the controls used over the K time slots. θ is univocally determined from the policy μ, but it also identifies univocally a static or a threshold policy. In fact for the static policy is $\mu(n) = p = \theta/K$, while for the threshold policy it is $h = \max\{h \in \mathbb{N} : v(h) = h \cdot u_{\max} + (n-h)\cdot u_{\min} \leq \theta\}$, and $\mu(h) = \theta - v(h)$. Note that if $\theta = \beta$, then the two policies are the best static policy and the optimal (threshold) policy determined in the previous section. Then in both cases our policy estimation problem comes down to estimate β. Again mobiles do not know quantities such as λ, N, etc., so that they can not compute β a priori using Eq. (9). The stochastic approximation algorithm will estimate β looking for the unique solution of a certain function of θ in the interval $H = [\theta_{\min}, \theta_{\max}] = [K \cdot u_{\min}, K \cdot u_{\max}]$.

The algorithm works in rounds. Each round corresponds to the delivery of a set of messages. During a given round, a policy is used. Let us denote by μ_m the policy adopted at round m and $\theta_m = \sum_{k=0}^{K-1} \mu_m(k)$ the corresponding θ value. At the end of each round an estimate of $\mathrm{E}[X_K]$ can be evaluated by averaging the total number of copies made during the round for each different message. Let $\overline{X}_m$ denote such average. $\overline{X}_m$ is used to update θ, according to the following formula:

$$\theta_{m+1} = \Pi_H\Big(\theta_m + a_m(\Psi - \overline{X}_m)\Big), \quad (10)$$

[3]Note that the estimation of N and λ is per se non-trivial in the lack of persistent connectivity.

where

$$\Pi_H(\theta) = \begin{cases} \theta_{\max} & \text{if } \theta \geq \theta_{\max} \\ \theta & \text{if } \theta_{\min} \leq u \leq \theta_{\max} \\ \theta_{\min} & \text{if } \theta \leq \theta_{\min} \end{cases}$$

is the projection of θ on the interval H. As discussed above, the new policy μ_{m+1} is univocally determined from θ_{m+1}. Note that the length of a round affects the variability of the estimates $\overline{X}_m$ and hence of θ_m, but the following convergence results holds independently from round length.

Theorem 4.1: If the sequence $\{a_m\}$ is chosen such that $a_m \geq 0 \ \forall m$, $\sum_{m=0}^{+\infty} a_m = +\infty$ and $\sum_{m=0}^{+\infty} a_m^2 < +\infty$, the sequence of policies μ_m converges to the optimal policy with probability one.

Proof: On the basis of the considerations at the begin of this section we only need to prove that θ_m converges with probability one to β. The proof is divided in two parts. First we prove that the sequence θ_m converges to some limit set of the following Ordinary Differential Equation (ODE)

$$\dot{\theta} = \Psi - E[X_K|\theta]. \tag{11}$$

For this reason the Eq. (10) is said to be the stochastic approximation of Eq. (11). The convergence is a consequence of Theorem 2.1 in [22] (page 127). In [20] we show that all the hypotheses of that theorem hold in our case.

In the second part we show that the solution of such ODE converges to β as time diverges.

We observe that from Eq. (4) and Eq. (3)

$$E[\overline{X}_m|\theta_m] = E[X_K|\theta_m] = N - (N - X_0)e^{-\lambda\Delta\theta_m}$$

so that Eq. (11) can be written as

$$\dot{\theta} = \Psi - N + (N - X_0)\, e^{-\lambda\Delta\theta_m}. \tag{12}$$

We now need to show that the ODE (12) converges as time diverges to an asymptotically global fixed point and this is β.

First, it is easy to check that $\theta^* = \beta$ is an equilibrium point of (12).

Second, as $E[X_K|\theta]$ is strictly monotonic in θ, the equilibrium point is unique. In order to demonstrate the stability of the estimator, we use the Lyapunov function $V(\theta) = (\theta - \theta^*)^2$. Then, we have:

$$\dot{V}(\theta) = 2(\theta - \theta^*) \cdot \dot{\theta} = 2\left[\theta + \frac{1}{\lambda\Delta}\log\left(\frac{N-\Psi}{N-X_0}\right)\right] \cdot \\ \cdot \left[\Psi - N + (N - X_0)\, e^{-\lambda\Delta\theta}\right] < 0 \text{ for } \theta \neq \theta^* \tag{13}$$

Asymptotic global stability follows in both cases from Lyapunov's theorem. ■

Remark 4.1: Roughly speaking, Theorem 2.1 in [22] shows that the θ_m converges to the solution $\theta(t)$ of Eq. (11) for $t \approx t_n$, where $\{t_n\}_{n\geq 0}$ is the sequence defined as follows:

$$t_0 = 0,\ t_m = t_{m-1} + a_m,\ \text{for } m > 0. \tag{14}$$

By comparing Eq. (10) and Eq. (14), we can observe that a trade-off arises, typicale of stochastic approximation algorithms in the form (10). In fact, sequences $\{a_n\}$ vanishing slower guarantee a faster convergence to the ODE trajectory because the series $\sum a_n$ diverges faster and then t_n in Eq. (14) is larger. At the same time the corresponding estimation is noisier since they have weaker filtering capabilities in the iterates equation (10).

Remark 4.2: After some cumbersome derivation, the closed form solution of Eq. (12) is:

$$\theta(t) = \frac{1}{\lambda\Delta}\ln\left\{e^{\lambda\Delta[(\Psi-N)t+\theta(0)]} + \\ + \frac{N - X_0}{N - \Psi}\left[1 - e^{\lambda\Delta(\Psi-N)t}\right]\right\} \tag{15}$$

In Section VI we will provide numerical evidence of the convergence of the "tail" of the iterates to the ODE dynamics.

In the description of the algorithm above we have suggested that the online estimation of the optimal control is obtained by using in Eq. (10) the estimation $\overline{X}_m$ obtained from real message transmission. However, in the case of two-hop routing, we may circumvent this constraint by using a sort of "virtual messages": indeed, the stochastic approximation technique works also if the source simply keeps track of the number of mobiles it *would* infect during the a time window of duration τ if it *had* a message to transmit. Then the source can simply register the contacts and "virtually" apply the policy keeping track of the nodes it would have infected if it had a message. If a real message has to be transmitted, the current policy estimation can be used.

A. Choice of the Sequence $\{a_n\}$

The performance of the stochastic approximation algorithm (10) is known to depend heavily on the choice of the sequence $\{a_n\}$ [23].

A standard choice is $a_n = \frac{C}{n}$; the optimal value of C that guarantees the smallest asymptotic variance is [22] $C = \frac{\partial E[X(\tau)|\theta]}{\partial \theta}\Big|_{\theta=\theta^*}$. In general, however, C is unknown (as it depends on the unknown function $E[X(\tau)|\theta]$) and cannot be set a priori.

Another possible approach to improve the performance of (10) is to use techniques such as Polyak's averages [22], [24]. The idea is to use slower decreasing sequences to let the iterates converge faster, while using averages to smooth actual estimates.

In Polyak's method, we may use a sequence $a_n = O(n^{-1})$, and in particular one that satisfies the condition $a_n/a_{n+1} = 1 + o(a_n)$ and use as estimation of β

$$\Theta_n = \frac{1}{n}\sum_{k=1}^{n}\theta_k. \tag{16}$$

In Section VI we will show that using Polyak's averaging techniques may obtain advantages in terms of convergence time to the optimal control.

B. Constant Step Approximations

In a real DTN implementation, we may be interested in tracking changing conditions. This can be done through by considering constant step approximations, i.e., iterates of the form:

$$\theta^a_{m+1} = \Pi_H\Big(\theta^a_m + a(\Psi - \overline{X}^a_m)\Big). \tag{17}$$

In this way, the system does not “get stuck” at a given θ^* but keeps on modifying its behaviour, in an open–ended fashion. Also for such case, results on convergence can be derived [20]. In particulaf for small enough step size a, the limit process is, with arbitrary high probability, concentrated in an arbitrary small neighbourhood of the optimal control θ^*. This is important in ensuring that the approximation obtained is close to the optimal control policy.

V. The Multiclass Case

In this section we model the decentralized stochastic control problem in the presence of several competing DTNs as a weakly coupled stochastic game, introduced in [8], [9].

A. The model

Consider a network that contains M classes of mobiles. There are N_m mobile nodes in class m. In each class there is a source and a mobile of class i stores and forwards only messages originating from the source of that class. Nodes adopt two-hop routing. All sources generate messages for the same destination. Here we assume that message transmission time is equal to a time slot duration and meetings occur at the begin of a time slot. The transmission technique uses receiver based codes, and an arbitration procedure can avoid collisions among the members of the same class, so that collisions occur if and only if two or more nodes from different classes are trying to deliver their messages to the destination at the same time. We also study the case when the arbitration procedure is coherently applied from all nodes, so that when many nodes have the possibility to transmit a message to the destination, one of them is successful.

We consider two different traffic generation models. In both cases each source has a single relevant message at a given time instant. In the first traffic generation model sources synchronously generate messages with lifetime equal to τ. In the second one, after a message is delivered or time τ has elapsed since its generation, the source can stay idle for a random amount of time after which a new message will be generated. Hence sources operate asynchronously.

As in the previous section, it may not be desirable for a source to transmit a copy of its message at each opportunity it has since this consumes expensive network resources such as energy, hence the source can decide to forward the message with a given probability. Due to interactions among different mobile classes, a problem of non-cooperative control of those probabilities arises.

Our problem falls into a category of stochastic games that was recently introduced in [8], [9], in which each player control an independent Markov chain and knows only the state of that Markov chain. The interaction between the players is due to their utilities or costs which depend on the states and actions of all players. Indeed in our framework each source can infect mobile of its own class independently from the other sources and the only coupling derives from collisions when transmitting to the destination. The possibility of having collisions affects the delivery probability.

A different problem is a classless model where a relay node can be infected by all the available source nodes. In this case the state needs in general to specify which messages are carried by each node. Nevertheless if we consider the synchronous traffic generation model and performance metrics only depending on the delivery of the first message among the competing ones, the problem can be addressed in the same framework [20].

B. A Weakly Coupled Markov Game Formulation

Let $X_n^{(i)}$ be the number of mobiles of class i that are infected at time $n\Delta$. We consider the following discrete time stochastic game.

- **The players:** the M classes of mobiles that act independently.
- **The actions.** If at time $n\Delta$ class-i source encounters a mobile, it attempts transmission with probability $u_n^{(i)}$. $\mu^{(i)}$ is the time-dependant policy of class-i source. In this game theoretical framework we refer to $\mu^{(i)}$ also as the strategy of class-i, while $\mu^{(-i)}$ denotes the set of strategies adopted by the other classes.
- **The performance index.** The utility of each player/class is the probability of successful delivery, $F^{(i)}(K\Delta)$. Each class has also a constraint on the expected number of infected nodes, i.e. $E[X_K^{(i)}] \le \Psi^{(i)}$.
- **Information.** Source i is assumed to know only $X_n^{(i)}$ and not know $X_n^{(j)}$ for $j \ne i$. But it knows its statistics. The precise knowledge of $X_n^{(i)}$ is possible since the source i knows exactly to how many mobiles it transmitted the message for relay. Note that it is not assumed to know if the message was delivered to the destination.

Let us define $Y_n^{(i)}$ as the number of infected nodes of class i that can transmit to the destination during the n- th time slot ($0 \le Y_n^{(i)} \le X_n^{(i)}$), $S_n^{(-i)} = \sum_{j\ne i} Y_n^{(j)}$ and $S_n = \sum_j Y_n^{(j)} = Y_n^{(i)} + S_n^{(-i)}$.

A recurrence law analogous to Eq. (6) can be derived for the CDF of the delivery time of messages of each class. For example when inter-class collisions at destination cannot be avoided, for each class i it holds [20]:

$$G^{(i)}(n+1) = G^{(i)}(n)\left(\prod_j X_n^{(j)*}(\lambda\Delta) + \left(1 - \prod_{j\ne i} X_n^{(j)*}(\lambda\Delta)\right)\right).$$

For the case of a cross-class arbitration procedure, then one needs to take into account the possibility that a node of class i succeeds even in presence of other nodes. In a fair arbitration scheme this will happen with probability $Y_n^{(i)}/(Y_n^{(i)} + Y_n^{(-i)})$.

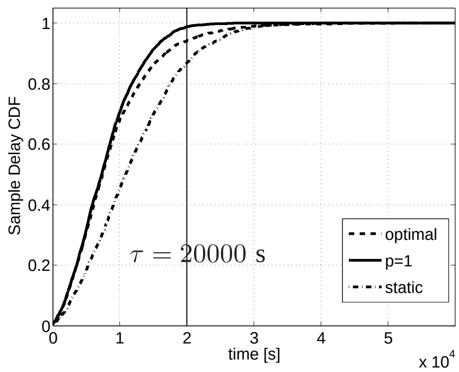

Fig. 1. Delay CDF in the case of a) optimal control policy (dashed line) b) static control (dot-dashed line) and c) $p = 1$ (dotted line).

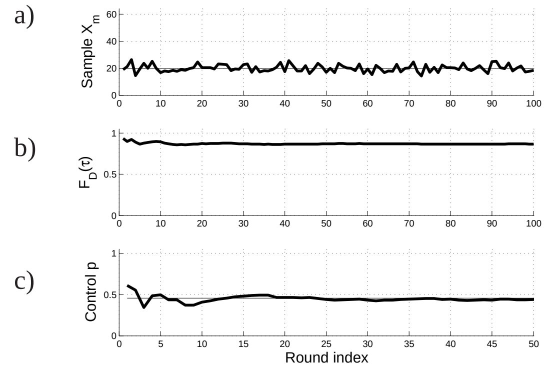

Fig. 2. The dynamics of the stochastic approximation algorithm applied to the static forwarding policies.

We can then derive the following expression for $G^{(i)}(n)$ [20]:

$$G^{(i)}(n+1)=G^{(i)}(n)\left(\Pr\{S_n=0\}+\right.$$
$$\left.(1-\Pr\{S_n=0\})\ \mathrm{E}\left[\frac{S_n^{(-i)}}{S_n}|\ S_n>0\right]\right).$$

We observe that $G^{(i)}(n+1)$ depends on the vectors of control actions $(u_k^{(1)}, u_k^{(2)}, \dots u_k^{(M)})$, for $k \leq n-1$. Before stating our main results we introduce the following observation (the proof is in [20]).

Proposition 5.1: For both the arbitration procedures, $G_{n+1}^{(i)}$ is decreasing in the control action $u_{n-1}^{(i)}$.

Theorem 5.1: If $\forall n\ G^{(i)}(n+1)$ is decreasing in the control action $u_{n-1}^{(i)}$, then the optimal threshold policy for the single-class case is also the best response to all the possible $\mu^{(-i)}$.

Proof: The proof follows the same steps of that of Theorem 3.1: given a non-threshold policy $\mu^{(i)}$, we build in the same way a new policy $\mu'^{(i)}$. In fact equations (3), (4) and (8) hold also for each specific class i and the hypothesis on $G_n^{(i)}$ permits to conclude that $\mu'^{(i)}$ has better performance. ■

Remark 5.1: We observe that the result above applies to both the arbitration schemes and the traffic generation models considered. In fact the different traffic models, for a given class i, only have an effect on the probability distributions of $X_n^{(-i)}$ and $Y_n^{(-i)}$, but they not change the best response strategy for class i.

From the theorem above the following result follows:

Corollary 5.1: The considered game has a unique Nash equilibrium. This Nash equilibrium is obtained when each class adopts its optimal singleclass threshold policy.

Proof: The optimal threshold policies are mutual best responses, so they are a Nash Equilibrium. Moreover whatever a different set of strategies cannot be a Nash equilibrium, because at least one class can improve its performance by adopting the optimal singleclass threshold policy. ■

In some sense this corollary shows how the single class optimal policy is also "robust" towards competition with other classes of nodes. An interesting question is if this Nash Equilibrium is also Pareto optimal, i.e. if it is also a global optimum in the sense that there is no other set of strategies that can guarantee at least the same performance to all the classes and strictly better performance to at least one class. Our simulation results in Sec. VI show that this is not the case.

VI. Numerical Results

Numerical results have been obtained simulating the discrete-time system with Matlab.

The intensity λ of the pairwise meeting process has been selected considering a standard Random Waypoint (RWP) mobility scenario. In fact it is known [16], that for the RWP where nodes move with constant velocity $\lambda = \frac{8wRv}{\pi L2}$, where L is the playground size, R the communication range, $w = 1.3683$ is a constant and v is the scalar speed of nodes. Here, we have chosen $L = 5000$ m, $N = 200$, $R = 15$ m and $v = 5$ m/s. The corresponding value is $\lambda = 1.0453\times10^{-5}\ s^{-1}$. Unless otherwise specified, results have been obtained with $\Delta = 10$s, $\tau = 20000$, $\Psi = 20$, $u_{min} = 0$ and $u_{max} = 1$.

A. Discrete control policies

In the first set of experiments, we simulated the discrete control policies in order to evaluate their relative performances. In Fig. 1 we reported the comparison of the optimal control policy and the best static control policy. For the considered setting, we obtain $h^* = 911$ for the optimal threshold policy, and $p^* = 0.46$ for the static policy. It can be noticed that the static policy attains a much lower success probability, whereas, as expected, the delay CDFs under the optimal control and under the policy $\mu(n) = 1$ coincide at times smaller than $h^*\Delta$.

B. Stochastic Approximation

In the following we consider the stochastic approximation algorithm described in Sec. IV and we show that it is able to learn the optimal control both for static and dynamic controls. The setting is similar to what described above, but in this case several rounds are performed (see Sec. IV). Basically, the source performs for each round a sample measurement of $\overline{X}_m$, based on 30 different estimates of the number of infected

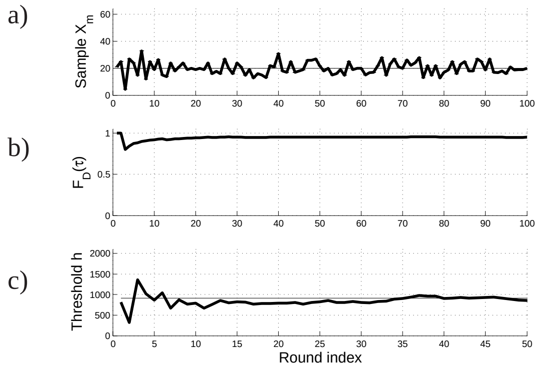

Fig. 3. The dynamics of the stochastic approximation algorithm applied to the optimal forwarding policies.

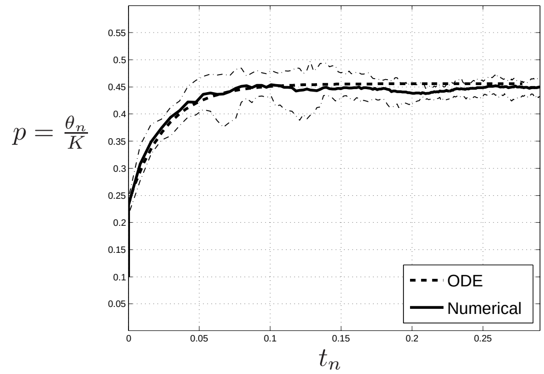

Fig. 4. The convergence of the dynamics of the control variable against the reference ODE; at the time scale t_n and averaged over 10 sample trajectories in the case of static control. Thin dash-dotted lines delimit the maximum and minimum values attained by the estimate trajectories.

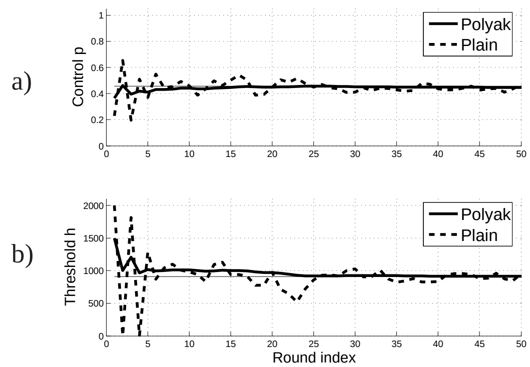

Fig. 5. Algorithm employing Polyak's averages applied to a) static and b) threshold forwarding policies.

nodes at time τ. At the end of the round, a novel policy is generated and is employed in the following run. Unless otherwise specified, results in this section have been obtained with $a_m = 1/(10 \cdot m)$.

Fig 2 illustrates a specific run for the case when the source estimates the parameter p^* of the best static policy. The figure shows that the estimates $\overline{X}_m$ evaluated by the source are noisy, due to the limited number of samples per estimate. Nevertheless, the convergence of the algorithm is evident from the dynamics of the control p, i.e. the static forwarding probability, which stabilizes after about 20 rounds around the optimal value p^* (the horizontal line). For the sake of completeness, we also reported the time evolution of $F_D(K)$, obtained during the run of the algorithm (Fig 2b)).

We repeated the same experiment in the case of the optimal threshold policies. In this case, the source tries to estimate the optimal threshold h^*, and the dynamics of the estimated parameter is depicted in Fig 3c). We observe that the convergence time is similar to that measured in the case of the static policies. This is due to the fact that in both cases the stochastic approximation algorithm estimates the same parameter β and even if the distribution of $\overline{X}_m$ (but not its expected value) is different for static and threshold policies, the sequence of estimates converges with probability one to the solution of the same ODE, as mentioned in Sec. IV.

Concerning this issue, Fig. 4 shows the dynamics of a properly rescaled version (according to considerations in Remark 4.1) of the controlled variable for the static control case against the solutions of the ODE (Eq. (15)). We averaged the trajectory over 10 runs of the algorithm. It can be observed that, after an initial transient phase, the trajectory of the control mimics the original ODE; we superimposed the maximum and minimum values of the trajectories for the sake of completeness. This pictorial representation confirms that the convergence speed of the algorithm is basically dictated by the dynamics of the related ODE solutions.

1) Polyak's averages: As mentioned in Sec. IV, a slowly decaying a_n obtains a fast convergence to the ODE asymptotic value, i.e. in our case to β. The price to pay is a lower rejection to noise, with larger oscillations. Here, we show the benefit of the Polyak-like averaging technique, as we choose a larger sequence, $a_n = 1/(10 \cdot n^{2/3})$, from which we expect faster convergence but a more noisy estimate.

Again, in Fig. 5 we reported the results of the stochastic approximation procedure: we superimposed the plain stochastic estimation of θ_n, based on the chosen a_n coefficients, and the output, obtained using the control from (16). We note the smoothing performed by the Polyak averaging over the estimated optimal control values, both in the case of static policies and in the case of threshold policies. Although this is a particular case, it shows, as anticipated in Sec. IV that it is possible to increase the speed of convergence of the algorithm by means of faster sequences, i.e. approaching faster the tail of the ODE dynamics, while reducing at the same time the estimation noise by averaging.

2) Nash Equilibrium: In the game theoretical framework, the result on the existence of a Nash equilibrium poses the question whether such equilibrium is Pareto optimal. The answer is not straightforward since the success probability depends on the number of nodes involved, on the number of classes and on the underlying encounter process.

For such reason, we resorted to numerical simulations in order to get better insight. In particular, we considered a two-

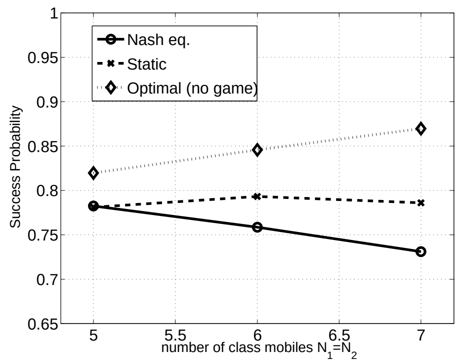

Fig. 6. Performance at the Nash equilibrium compared to the ideal case and to a static strategy; $\tau = 200$ s, $\Psi = N - 1$.

player game where each DTN has $N_1 = N_2 = 5, 6, 7$ nodes, and we rescaled the reference playground side to $L = 100$ m. Also, $\tau = 200$ s in this experiments. We repeated game rounds in order to measure the impact of the different strategies under the collision model. As depicted in Fig. 6, at the Nash equilibrium, the success probability is smaller than the one experienced in isolation by single players using the optimal threshold policy. This was expected, due to the effect of collisions. But, as shown in Fig. 6, if each class adopts the best static policy, the social outcome can be improved. We observe that this is not an equilibrium, because a class would find more convenient to switch to its optimal threshold policy, but it provides numerical evidence that the Nash equilibrium is not Pareto optimal.

VII. Conclusions

In this paper we introduced a discrete time model for the control of mobile ad hoc DTNs. We provided closed form expressions for static and dynamic policies for two-hop routing. Based on such results, we provided an algorithm, based on the theory of stochastic approximations, that enables all nodes in the DTN to tune independently and optimally the parameters of static and dynamic optimal forwarding policies, adapting to the current operating conditions of the system. This algorithm does not require message exchanges to operate and, more important, it guarantees convergence to optimal policies without the need to estimate global parameters of the DTN, such as the number of nodes or the intermeeting intensities. We believe that these features are very appealing for DTNs scenarios, where the estimation of global parameters is extremely challenging due to the absence of persistent connectivity.

Finally, the discrete model has been applied to the case of competing DTNs: we studied a class of cost coupled Markov games where players are different groups of mobiles, and the coupling occurs because of interference at a common destination node. We have shown that the single class optimal policy supports the only possible Nash Equilibrium of this game.

References

[1] S. Burleigh, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: an approach to interplanetary internet," *IEEE Comm. Mag.*, vol. 41, no. 6, pp. 128–136, Jun. 2003.

[2] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, vol. 44, no. 11, pp. 134–141, November 2006.

[3] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multi-copy case," *ACM/IEEE Trans. on Networking*, vol. 16, pp. 77–90, Feb. 2008.

[4] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, Tech. Rep. CS-2000-06, 2000.

[5] W. Zhao, M. Ammar, and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *Proc. of IEEE INFOCOM*, Miami USA, March 13–17 2005.

[6] M. M. B. Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," in *Proc. of ACM MobiHoc*, Florence, Italy, May 22–25, 2006, pp. 37–48.

[7] R. Groenevelt, P. Nain, and G. Koole, "The message delay in mobile ad hoc networks," *Performance Evaluation*, vol. 62, no. 1-4, pp. 210–228, October 2005.

[8] E. Altman, K. Avrachenkov, N. Bonneau, M. Debbah, R. El-Azouzi, and D. S. Menasche, "Constrained cost-coupled stochastic games with independent state processes," *Operations Research Letters*, vol. 36, pp. 160–164, 2008.

[9] ——, "Constrained stochastic games in wireless networks," in *IEEE Globecom General Symposium*, Washington D.C., 2007.

[10] M. Musolesi and C. Mascolo, "Controlled Epidemic-style Dissemination Middleware for Mobile Ad Hoc Networks," in *Proc. of ACM Mobiquitous*, July 2006.

[11] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Comput. Netw.*, vol. 51, no. 10, pp. 2867–2891, 2007.

[12] A. E. Fawal, J.-Y. L. Boudec, and K. Salamatian, "Performance analysis of self limiting epidemic forwarding," EPFL, Tech. Rep. LCA-REPORT-2006-127, 2006.

[13] A. Krifa, C. Barakat, and T. Spyropoulos, "Optimal buffer management policies for delay tolerant networks," in *Proc. of IEEE SECON*, 2008.

[14] G. Neglia and X. Zhang, "Optimal delay-power tradeoff in sparse delay tolerant networks: a preliminary study," in *Proc. of ACM SIGCOMM CHANTS 2006*, 2006, pp. 237–244.

[15] E. Altman, T. Başar, and F. D. Pellegrini, "Optimal monotone forwarding policies in delay tolerant mobile ad-hoc networks," *Proc. of ACM InterPerf*, October 24 2008.

[16] R. Groenevelt and P. Nain, "Message delay in MANETs," in *Proc. of ACM SIGMETRICS*, Banff, Canada, June 6, 2005, pp. 412–413.

[17] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Trans. on Mobile Computing*, vol. 6, no. 6, pp. 606–620, 2007.

[18] T. Karagiannis, J.-Y. L. Boudec, and M. Vojnović, "Power law and exponential decay of inter contact times between mobile devices," in *Proc. of MobiCom '07*. ACM, 2007, pp. 183–194.

[19] M. Shaked and J. G. Shantikumar, *Stochastic Orders and Their Applications*. New York: Academic Press, 1994.

[20] E. Altman, G. Neglia, F. De Pellegrini, and D. Miorandi, "Decentralized stochastic control of delay tolerant networks, INRIA Research Report 2008-Number Pending," August 2008. [Online]. Available: http://www-sop.inria.fr/maestro/personnel/Giovanni.Neglia/publications.htm

[21] E. Altman, *Constrained MArkov Decision processes*. Chapman and Hall/CRC, 1999.

[22] H. J. Kushnern and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2nd Edition, 2003.

[23] J. Maryak, "Some guidelines for using iterate averaging in stochastic approximation," *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, vol. 3, pp. 2287–2290 vol.3, Dec 1997.

[24] B. T. Polyak and A. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM Journal of Control and Optimization*, vol. 30, pp. 838–855, 1992.