Optimizing spatial and temporal reuse in wireless networks by decentralized partially observable Markov decision processes

Joni Pajarinen, Ari Hottinen, *Senior Member, IEEE,* and Jaakko Peltonen

Abstract—The performance of medium access control (MAC) depends on both spatial locations and traffic patterns of wireless agents. In contrast to conventional MAC policies, we propose a MAC solution that adapts to the prevailing spatial and temporal opportunities. The proposed solution is based on a decentralized partially observable Markov decision process (DEC-POMDP), which is able to handle wireless network dynamics described by a Markov model. A DEC-POMDP takes both sensor noise and partial observations into account, and yields MAC policies that are optimal for the network dynamics model. The DEC-POMDP MAC policies can be optimized for a freely chosen goal, such as maximal throughput or minimal latency, with the same algorithm. We make approximate optimization efficient by exploiting problem structure: the policies are optimized by a factored DEC-POMDP method, yielding highly compact state machine representations for MAC policies. Experiments show that our approach yields higher throughput and lower latency than CSMA/CA based comparison methods adapted to the current wireless network configuration.

Index Terms—Spatial reuse, wireless network, decentralized POMDP, multi-agent planning, medium access control.

1 Introduction

New applications and proliferation of wireless devices demand increased wireless network performance. In current protocols, devices communicating over congested channels may get long *transmission delays* due to waiting for transmission opportunities inefficiently or losing data in collisions with other transmissions. However, the traffic can contain patterns or structure that can be exploited for efficient communication. Transmission opportunities can arise due to gaps in transmissions of other devices; even when there are no gaps, opportunities can arise due to a spatial arrangement where the other transmissions do not cause much interference. In contrast to current access protocols, more efficient methods should *predict and exploit* both kinds of transmission opportunities.

We propose an efficient solution for planning transmission opportunities in wireless networks, exploiting both spatial structure (interference environment) and probabilities for temporal opportunities occurring in the traffic patterns. Unlike conventional medium access control (MAC) solutions, our random access policies are optimized for the prevailing spatio-temporal communication situation. After optimizing the policies, our solution does not need constant communication (e.g. notification or scheduling messages) between devices or to a central node: the plans are implemented as controllers that run independently in each device for a long time.

Our solution exploits both *spatial reuse* and *opportunities in time*. In spatial reuse [1] several wireless devices can use the same frequency channel simultaneously. Due to signal attenuation, a device interferes only with close-by devices. Two spatially far-away devices can use the same frequency channel simultaneously with little interference. Additionally, *opportunities in time* arise because a wireless device does not always have data to transmit; other devices can exploit such moments. To get low delay and high throughput, devices must *plan in advance* when to transmit. This planning over time is coupled with spatial reuse: how much data a device can currently transmit depends on interference from other devices, which depends on the spatial network configuration.

We study spatial and temporal reuse in a network where each wireless agent gets data into its transmit queue from a possibly bursty traffic source, and transmits data by a transmission scheme whose performance depends on interference from other agents. We consider two interference sensing types: an agent observes interference either at its receiver via a feedback link, or at the transmission point. To optimize a sensing-based MAC policy over both spatial and temporal dimensions, we frame MAC optimization as a *factored decentralized partially observable Markov decision process* (factored DEC-POMDP). In a DEC-POMDP agents do not communicate; they make decisions using local observations about the state of the world and other agents. This DEC-POMDP approach yields MAC protocols for desired objectives like maximal throughput or minimal delay. The optimized protocols are well-performing compact state machines. Modeling the wireless network problem by a DEC-POMDP has further **advantages**: **1.** The op-

- *J.Pajarinen is with the Department of Automation and Systems Technology, Aalto University, Finland. J.Pajarinen carried out the work on the manuscript at the Department of Information and Computer Science, Aalto University, and at Nokia Research Center. Email: Joni.Pajarinen@aalto.fi*
- *A.Hottinen is with Asparrow Ltd, Finland*
- *J.Peltonen is with the Department of Information and Computer Science, Aalto University, Finland and with Helsinki Institute for Information Technology HIIT*

timization goal for the network can be chosen freely, e.g. throughput, delay, or packet loss. **2.** The approach takes into account all network model properties. Policies can be optimized for different spatial network configurations and with individual source traffic models and transmit queue sizes for all agents; thus the approach works well in both homogeneous and very heterogeneous environments. **3.** The model takes into account uncertainty both in environment evolution over time and in observations: e.g. sensor noise can be incorporated as a probability to make wrong interference measurements. **4.** The approach yields compact policies that can be inspected to gain human-understandable information on what MAC policies should take into account in such a network configuration. **5.** Carrier sense multiple access (CSMA), Aloha and other MAC protocols use contention based random channel access while time division multiple access (TDMA) and others use deterministic access. Our optimized policies can include both random access and deterministic behavior.

Section 2 discusses related work. Section 3 outlines a wireless network model. Section 4 tells why a DEC-POMDP yields optimal policies for the network model, how the problem can be framed as a factored DEC-POMDP, and how policies can be efficiently computed. Section 5 experimentally compares the DEC-POMDP approach to CSMA/CA based approaches adapted for the network model and each network configuration. Section 6 concludes with discussion and future directions.

2 Related work

We frame the problem of deciding when to transmit as a multi-agent planning problem and optimize transmission policies over temporal and spatial dimensions. We now discuss related works having some aspects of our solution: MAC protocols, spatial reuse, work on combining temporal and spatial aspects, and previous work on multi-agent planning in wireless network problems.

Medium access control. Kumar et al. [2] survey MAC protocols in wireless ad hoc networks. CSMA protocols transmit only when the channel is sensed idle; they use a carrier sense threshold to detect whether a channel is occupied for a perceived interference level. CSMA protocols like the IEEE 802.11 MAC update an internal state with channel state observations, and decide whether to transmit based on the internal state. Channel state observations, and thus CSMA behavior, can be influenced by tuning the carrier sense threshold or tuning transmit power, which affects perceived interference.

CSMA operation is superficially similar to our method as an agent's transmit decision is updated temporally and decided using current measured interference at its spatial location; decisions are not optimized for the communication situation, they are based on a factory-assigned protocol. Our approach is more advanced: each agent decides whether to transmit based both on current observed interference and predicted future interference (see Section 4 for details), by updating its controller state using channel observations and making decisions based on the controller state; the controller has been optimized using a probabilistic model of the network for a joint objective (maximize throughput, minimize delay, ...). Our optimization sets parameters of all controllers to work well together and take future interference levels, caused by controllers' actions, into account.

The IEEE 802.11 [3] MAC protocol uses CSMA, with binary exponential backoff, with physical or virtual carrier sensing. For simplicity, we consider only physical carrier sensing in this paper. Binary exponential backoff increases the maximum waiting time (contention window) exponentially with the number of consecutive collisions. Many approaches [4], [5], [6] improve efficiency of CSMA-type protocols by adjusting the backoff. We go beyond adjusted backoff mechanisms; our methods predict and avoid collisions in an optimized proactive way rather than only reacting to them by a fixed protocol.

Kaynia et al. [7] analyze outage probability of CSMA and Aloha protocols considering the spatial location of wireless devices. By letting the receiver sense the channel and inform its transmitter whether to initiate transmission, outage probability of conventional continuous-time CSMA is clearly improved. We also consider receiver-side observations but our method is not limited to them.

Some MAC approaches [8], [9] use multiple channels to help wireless network performance. For MAC, standards like WiMAX use centralized scheduling. There is work on using communication among agents [9] and using polling to schedule data transfers [10]. We focus on single channel random access control without additional communication between agents during policy execution.

Spatial reuse. A survey [1] on improving spatial reuse in multihop wireless networks discusses approaches to enhance IEEE 802.11 performance. Previous work exists on spatial approaches that tune the carrier sense threshold [11], optimize transmission power control [12], adapt transmission rate [13], or use directional or MIMO antennas [14], [15]. The perhaps closest previous work on spatial reuse is about tuning the carrier sense threshold and optimizing transmission power, discussed next.

Carrier sense threshold. The physical carrier sense threshold is tuned in [11]; an analytical model for the optimal threshold is shown, given a network topology, reception power and data rate assuming a homogeneous network with identical interference and noise at all nodes. A heuristic distributed algorithm, which assumes all agents have the same local network configuration, is given; it adapts the sense threshold using information also from other agents. Another paper [16] adjusts individual carrier sense threshold levels: each agent tries several levels and for each level counts the number of ACK frames of any agent, corresponding to the number of successful transmissions. The best level is chosen.

Some papers investigate both the **carrier sense threshold and the transmit power.** Fuemmeler et al. [17], [18] analyze the relationship between transmit power

and carrier sense threshold in wireless networks that use CSMA. They conclude the product of the transmit power and carrier sense threshold should be constant. Their analysis uses, for tractability, an approximation that assumes the interference to the transmitter is the same as the interference to the transmitter's receiver.

Kim et al. [12] show network capacity depends only on the ratio of transmit power to the carrier sense threshold. Their analysis assumes the network is densely populated, agents interfere as if they were in a hexagonal grid around the transmitter, and all agents have equal transmit powers and sense thresholds. They show when the set of data rates is limited, tuning transmit power gives more sophisticated rate control than tuning the carrier sense threshold, if there are enough power levels.

Overall, approaches that dynamically tune MAC access parameters for spatial or temporal reuse use a predefined protocol (like CSMA/CA) and adjust parameters like contention window size, backoff, carrier sense threshold, or transmit power. Parameter selection is often based on analyses that, for tractability, use assumpions like saturated source traffic or a homogeneous network configuration. In contrast, our model makes no assumptions on wireless agent independence, network traffic type, or how agents are spatially distributed; we only assume the world is Markovian and that the agents optimize a joint goal. Our model includes bursty network traffic, takes into account how wireless agents influence each other over time, and includes the spatial dimension. Our approach yields finite state controllers (MAC protocols) that are optimized for the network model and take into account the interplay between agents.

We exploit both spatial reuse and temporal opportunities; our experiments compare our DEC-POMDP method to spatial reuse by carrier sense threshold tuning.

Multi-agent planning in wireless networks. We model optimal decision making in a wireless network as a factored DEC-POMDP whose solution is a set of medium access control policies for wireless agents. The paper [19] gives the first method for solving factored infinite-horizon DEC-POMDP problems. One experiment in [19] is a simple wireless network benchmark, where devices arranged in a row transmit slotted fixed size packets. The network model differs clearly from the one in this paper: in [19] the transmit queue size is only a few slots, the collision model is binary, and the transmit queue is discrete. In this paper the collision model and transmit queues are continuous-valued, the discrete transmit queue model is longer, and wireless devices may be in any arrangement. Also the convergence time in [19] is much longer, on a much simpler problem, compared to the method used in this paper.

Shirazi et al. [20] model channel dynamics with a Markov model in a wireless relay network: the goal is to choose which relay nodes to use for re-transmissions. They model the problem as a DEC-POMDP where neighboring relay nodes are allowed to communicate and use a gradient ascent type method to compute policies for relay nodes. The DEC-POMDP method in [20] is restricted to special problems, where an agent's local state is not influenced by other agents. In [20] channels evolve independently of the relays being optimized, but most often in wireless networks, and in also in this paper, an agent's transmissions influence other agents.

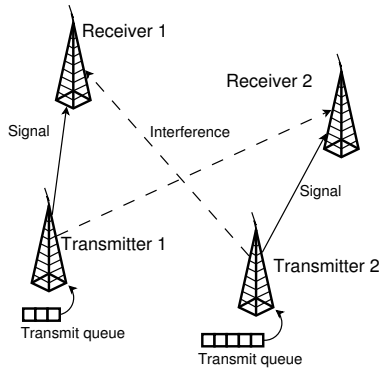

Fig. 1. Wireless network

Multi-agent reinforcement learning is widely used to optimize cognitive radio (CR) behavior [21], [22], [23], [24] in wireless networks with high priority primary users (PUs) and multiple frequency channels. The paper [21] studies convergence of multi-agent Q-learning (MAQL) for channel selection in a CR network, when the number of CRs is less or equal to the number of channels. In [22] MAQL is used to limit CR interference to PUs in an IEEE 802.22 based wireless network where CRs are secondary base stations. Wu et al. [23] use MAQL to reduce interference to PUs by optimizing channel selection and transmit powers of CRs. MAQL is used in [24] for selecting which channels to sense in order to avoid collisions with PUs. To avoid collisions among CRs a special backoff based mechanism is used.

Unlike our approach, the main aim of the above CR approaches is to prevent interference to PUs. They do not use knowledge of source traffic or agents' spatial configuration to optimize policies; they avoid interference by using multiple channels, but in our approach agents avoid interference using opportunities over space and time. The above approaches use special problem structure in Q-learning. To our knowledge there is no Q-learning approach that could be applied directly to the wireless network problem in this paper. For instance, [21], [22], [23], [24] do not consider that an agent's actions affect local states of other agents over time.

3 Network model

Our method is designed for a typical wireless communication situation (overall network model in Fig. 1). The network has N transmitter-receiver pairs: a transmitter (agent) i has an associated receiver i and vice versa. We index transmitter-receiver pairs, transmitters, receivers, and agents interchangeably. Each transmitter has a finite transmit queue with real-valued current length. A source

puts data into the queue; the transmitter removes data from the queue when transmitting. For simplicity we assume transmitters do transmision and sensing actions at discrete time intervals called "time slots".

How much data a transmitter can successfully transmit in a time slot depends on the signal-to-interference ratio (SIR) of the activated link. Consider the ith transmitter-receiver pair (transmitter i and receiver i). The amount of data transmitted by transmitter i at slot t is $D_i(t) = \min(B_i(t), WC_i(t))$, where $B_i(t)$ is the amount of data in the transmit queue,

$$C_i(t) = \log_2(1 + \mathrm{SIR}_i(t)) \tag{1}$$

is the information capacity (bits per channel use) from transmitter i to receiver i where the receiver's SIR at slot t is denoted as $\mathrm{SIR}_i(t)$, and W is the number of available channel uses per slot. The equation of $D_i(t)$ follows because even with optimal transmission schemes (achieving Shannon capacity), one can at most empty the transmit queue.

The SIR depends on all active links; we use the conventional definition

$$\mathrm{SIR}_i(t) = g_{i,i}x_i(t)/(\sum_{j\neq i} g_{j,i}x_j(t) + \sigma^2)\ , \tag{2}$$

where $g_{j,i}$ is the link gain (power) from transmitter j to receiver i, $x_j(t)$ is the transmit power of transmitter j at time t, and σ^2 is the (thermal) noise power at time t; we approximate the momentary σ^2 by the average thermal noise power, assumed equal at all receivers. At each time slot each receiver i informs its associated transmitter i about the current channel capacity via a feedback channel. In optimization, link gains between agents are assumed known or are first estimated by usual path-loss measurements (in experiments estimated by path loss using line-of-sight).

The goal is to optimize policies for the transmitters, with a common objective such as maximizing sum throughput ST or minimizing average queue delay AQD of the transmitters:

$$ST = \frac{\sum_{t=1}^{T_{max}}\sum_{i=1}^{N} D_i(t)}{T_{max}}\ ,\quad AQD = \frac{\sum_{t=1}^{T_{max}} w(t)l(t)}{\sum_{t=1}^{T_{max}} l(t)}\ , \tag{3}$$

where data inserted at time t of size $l(t)$ has to wait $w(t)$ time in the queue before exiting.

In each time slot, a transmitter's policy tells whether to transmit. A transmitter does not know what channel capacity will occur over the time of the transmission, but it can make a prediction using previous observations of channel capacity (see Section 4 for details). Transmission policies are computed centrally, but are *optimized to be executed in a decentralized manner*. There is no need to negotiate transmission slots with peers while executing policies, so communication pairwise between devices or to a central device is not needed; channel capacity is not spent on pairwise or central communication so it is available for communication of actual data between transmitters and receivers.

The source traffic of each agent is modeled by a discrete Markov model; the models are assumed to be known by the optimizer or agents can estimate their models from their network traffic. We use Markov models since they model network traffic compactly and have successfully modeled real-life bursty wireless traffic [25], [26] like Web or VOIP traffic. States of the source traffic model belong to either the "data" or "no-data" class; the number of "data" and "no-data" states is 1 at simplest and can be larger for more detailed models. In a "data" state the model inserts a fixed amount of data into the transmit queue. Counting state transitions is a trivial way to estimate transition probabilities of a two state Markov model; for models with more states probabilities can be estimated e.g. as in [26]. In experiments, the source traffic model (see Fig. 2b) is a two state Markov model, estimated from simulated NS-2 VOIP and HTTP traffic: it has probability 0.9530 to stay in the "no-data" state and 0.9259 to stay in the "data" state; on average the model puts data into the transmit queue 38.81% of the time. For clarity we use the same source traffic model in experiments for all agents; when agents have different source models, our DEC-POMDP approach which adapts to the network model could yield even better results compared to other methods.

4 Medium access control by a DEC-POMDP

We compute policies for wireless devices by a decentralized partially observable Markov decision process (DEC-POMDP). Resulting policies are finite state controllers (FSCs); an FSC is effectively a MAC protocol optimized for a certain network configuration. We compute FSCs using a Markov model of network dynamics and the wireless agents then execute the FSCs independently for a long time. Fig. 2 illustrates the approach. We describe how the MAC for the network model of Section 3 is solved by a DEC-POMDP, and the properties of the DEC-POMDP approach. We then show how to transform the wireless network problem into a factored infinite-horizon DEC-POMDP [19] problem and how good policies can be computed efficiently for several agents.

In the network model, transmitters do not need to exchange information with each other, they just execute a policy given to them. The policy is a conditional plan that tells what action to do upon receipt of an observation. In the network model a traffic source inserts data into a transmit queue and the transmitter transmits data from the queue. Evolution of transmit queues and traffic sources over time is modeled by Markov models; the state of traffic sources and transmit queues of all agents defines the world state. Policies are optimized by grading each policy during optimization by a sum of goodness measures called *rewards*: at each time step a common reward is given for the current state and

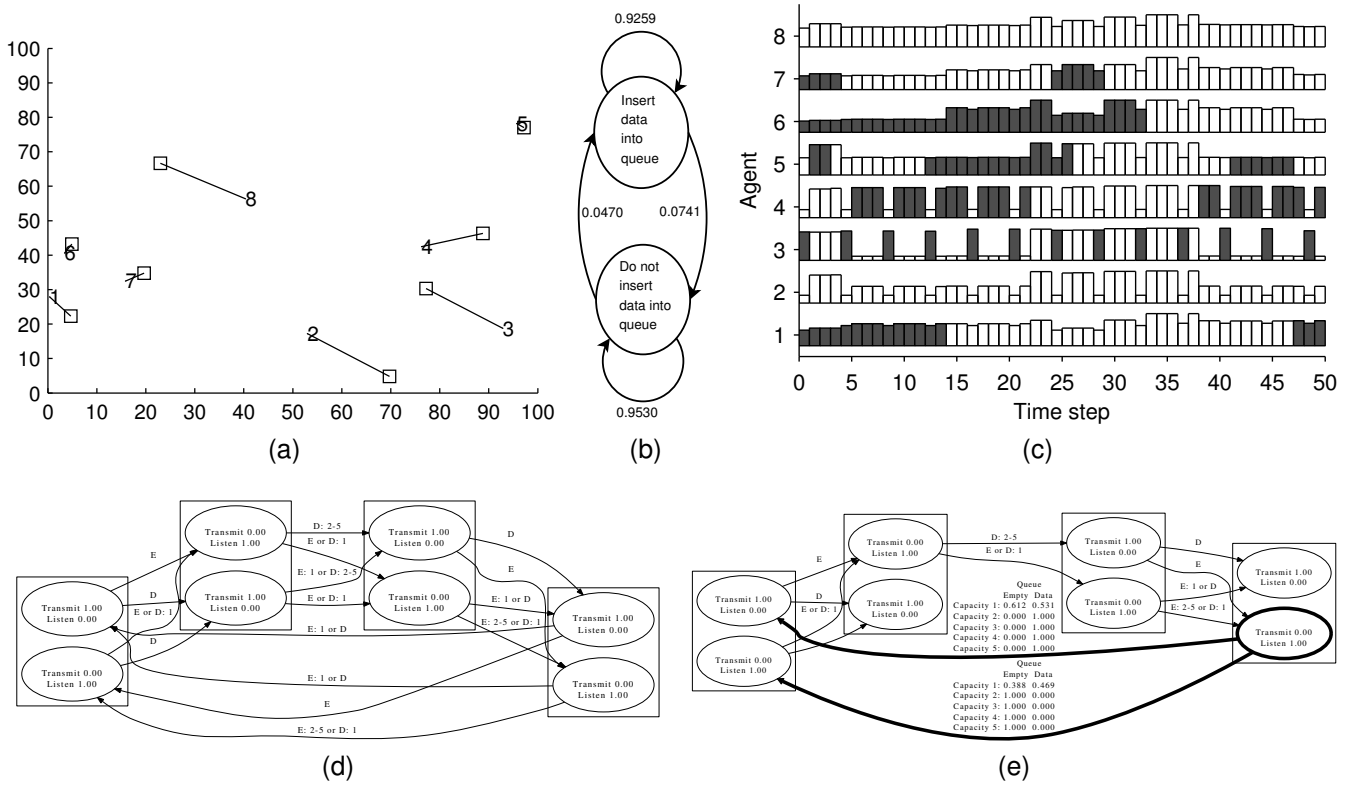

Fig. 2. Illustration of the proposed approach. (a) Wireless network with transmitters (numbers) and their receivers (boxes). The interference at the receiver determines the capacity of a transmitter. When each transmitter transmits it interferes with the receivers of other transmitters. (b) Bursty source traffic is modeled by a Markov model. The model inserts data into a transmitter's queue when in a data-insertion-state and changes state according to the shown probabilities. (c) A 50 time step snapshot of transmissions. Transmitters transmit data from their queues according to the Shannon capacity. Box height denotes the capacity possible if the agent would transmit at this time, relative to the maximum capacity. The potential capacity for an agent is large when only few of the highly-interfering other agents are transmitting. The agents do not know for certain what the potential capacity would be over the next transmission slot, they decide whether to transmit based on past observations of channel capacity and predictions of future capacity. Filled boxes denote actual transmissions and empty boxes denote non-transmissions. Interestingly, the agents seem to use both deterministic and contention based behavior, for example, Agent 3 seems to deterministically avoid collisions with Agent 4. Subfigure (d) shows a complete overview of the controller of agent one. Boxes denote the set of states the agent can be in during a particular time step. Ellipses denote finite state controller states, each of which directs the agent to perform a transmit or listen action (probabilities of both actions are listed inside the ellipse). Note that transmission is stochastic in our model: even when the action of a controller state is deterministic (an action probability of 0.00 or 1.00), the transition between controller states may be stochastic. Arrows denote transitions from one state to another; for clarity only state transitions with a probability greater than 40% are shown. Text above each arrow denotes the condition (observation) when the transition is allowed: "E" denotes empty queue and "D" data, a non-empty queue. Numbers after E or D denote the observed capacity level: for example "E: 1-2 or D" denotes the transition is allowed with either an empty queue and observed capacity levels 1 and 2 or with data in the queue and any observed capacity levels. Subfigure (e) illustrates the detailed transition probabilities from one state for each observation combination.

current actions of the agents, and the goal is to optimize a reward sum into the future. To act optimally an agent must take into account the current world state and how it may evolve into the future. The world state depends on the actions of all agents. An agent does not observe the current state of other agents' transmit queues or source models, but it can observe the interference level (locally or communicated from the receiver) which gives indirect information about the other agents. The agent does not know what observations other agents have made or what they are planning to do. To act optimally an agent must consider all possible observation histories of all other agents and all possible future action, observation sequences of all agents. In this setting optimal policies are given by a DEC-POMDP [27].

To act optimally agents must consider extremely many possible action-observation sequences, hence computational complexity of finite-horizon DEC-POMDPs is

NEXP-complete [28] and infinite-horizon DEC-POMDPs are undecidable even for one agent [28]. Our wireless network problem needs a policy that can run for a long time, i.e. a solution to an infinite-horizon DEC-POMDP. General (approximate) infinite-horizon DEC-POMDP algorithms have been demonstrated only for few agents [29], [30], but our recent (approximate) algorithm for *factored* infinite-horizon DEC-POMDPs in [19] shows good results for several agents.

In our DEC-POMDP approach wireless agents operate independently using their current policies. If a new communication situation (a changed network model) arises, policies can be recomputed to improve performance in the new situation; this may need occasional communication of the changed dynamics model and changed policies to and from agents. Before an agent's policy has been optimized, the agent can use a pre-defined default policy such as simplified CSMA/CA, that operates reasonably well in different wireless environments. In experiments we optimized policies starting from random controllers; our method can also start from hand-crafted policies, which could yield even better results.

In this paper we do not discuss in detail how the network model or policies are communicated, but concentrate on optimizing MAC policies of wireless agents. In the setup used in the experiments, communicating network models or policies is not expected to take much effort: policies of agents are very compact finite state controllers and source traffic models are simple two state Markov models that do not consume much bandwidth. To communicate link gains, one can communicate only link gains of the largest interferers for each receiver and a sum of the other link gains (see Section 4.2.1 on how the interference information is used to compute policies).

We now show how to frame the wireless network problem as a factored infinite-horizon DEC-POMDP [19] and compute good policies efficiently for several agents.

4.1 Wireless network as a factored infinite-horizon DEC-POMDP

We show how to describe the wireless network problem as a factored DEC-POMDP and give our algorithm based on [19] to compute good policies for wireless agents.

Notation. A (discrete-valued) DEC-POMDP involves transitions of variables from one time step to the next: for each variable y we denote its value in the current time step t simply by y and the value in the next time step with y'. We do not denote the time slot t explicitly here. For each variable y we denote conditional probabilities with $P(y|\mathrm{Pa}(y))$, where $Pa(y)$ are the variables that the probability of y depends on (details below). We describe in Section 4.2.1 how to transform real-valued quantities such as capacity into discrete random variables and use the real-valued variable symbols for clarity here.

States. A wireless network problem with N agents consists of N transmitters, and their respective receivers, source traffic models, and transmit queues. When the wireless network problem is described as a factored DEC-POMDP, the current world state s is a combination of state variables so that $s = (m_1, m_2, \ldots, m_N, B_1, B_2, \ldots, B_N)$, where m_i is the state of the source model of agent i, B_i is the size of the transmit queue of agent i. The world state space s has size $|m_1| \times |m_2| \times \cdots \times |m_N| \times |B_1| \times |B_2| \times \cdots \times |B_N|$, that is, the world state space is exponential w.r.t. the number of agents N.

Influence diagram. Fig. 3 shows an influence diagram for the factored DEC-POMDP for a two agent wireless network, when the objective is to minimize average queue delay. In the diagram circles denote variables, diamonds rewards, and arrows dependencies between variables. We describe next the variables in the network.

Actions. At each time step each agent i performs an action a_i which is either "do not transmit" or "transmit"; in our current experiments we use a simple power and rate control model where agents either transmit or do not transmit, but our approach also supports more fine-tuned power and rate control schemes simply by adding more transmit actions with different powers and rates. In the experiments the transmit power is

$$x_i = \begin{cases} 0 & \text{if } a_i \text{ is "do not transmit"} \\ TP_i & \text{if } a_i \text{ is "transmit"} \end{cases},$$

where TP_i denotes the transmit power of agent i. We used the simple scheme in experiments since it already sufficed to demonstrate the benefit of our method.

Capacity and interference variables. After the actions are performed, the channel capacity C_i at each receiver i is determined, and also the interference power I_i visible to agent i. Note that C_i and I_i are intermediate variables, that are used for computing variables part of the formal factored DEC-POMDP definition (controller, observation, action, state, or reward variables). C_i and I_i depend only on active transmitters: $P(C_i|a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_N)$ and $P(I_i|a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_N)$. Equation (1) yields the capacity (transmit power $x_i = 0$, when agent i is not transmitting). C_i can be computed using Equation 1. I_i is identical to C_i for exact receiver-side interference measurements, but for example for transmitter-side interference measurements the interference power visible to the agent can be different from the interference that yields the channel capacity at the receiver.

Transitions. Next each state variable s_i transitions to state s_i' according to the transition probability $P(s_i'|\mathrm{Pa}(s_i'))$, where $\mathrm{Pa}(s_i')$ is the set of variables that $s_i' = (m_i', B_i')$ depends on. The next time step source model state m_i' depends only on the current source model state m_i: $P(m_i'|m_i)$. The next time step transmit queue size B_i' of agent i depends on whether the agent is transmitting, on the channel capacity C_i, whether the source model m_i inserts data into the queue, and on the current size B_i of the transmit queue: $P(B_i'|a_i, C_i, m_i, B_i)$.

Observations. After the state transitions, each agent i

makes observations o_i according to the observation probability $P(o_i|\text{Pa}(o_i))$, where $\text{Pa}(o_i)$ is the set of variables that o_i depends on. Here $\text{Pa}(o_i) = (I_i, B_i)$, where I_i is the interference power visible to the agent and B_i is the size of the agent's transmit queue.

Rewards. At each time step, a real valued reward $R(s, \vec{a})$ is then given that depends on the current state of the world s and the actions $\vec{a}$ of the agents. The reward is a sum of reward functions: $R(s, \vec{a}) = \sum_{k=0}^{K} R_k(S_k)$ where R_k is a reward function operating on a subset S_k of state and action variables. The wireless network problem allows several different measures of network quality to be used as the reward in a straightforward fashion. To minimize the average queue delay AQD of Equation 3 for the network, the corresponding DEC-POMDP reward function is $R(s, \vec{a}) = \sum_{i=1}^{N} R_i(B_i)$ where $R_i(S_i) = -B_i/L_i$ is the negative queue size for agent i divided by the average arrival rate L_i (in the experiments identical for all agents). The reward function follows from Little's law: AQD is in the limit the average queue length divided by the average arrival rate. To maximise the throughput ST of Equation 3 for the network, the corresponding DEC-POMDP reward function is $R(s, \vec{a}) = \sum_{i=1}^{N} R_i(B_i, C_i)$ where $R_i(B_i, C_i) = D_i(B_i, C_i) = \min(B_i, WC_i)$ is the minimum of capacity and transmit queue size for agent i when agent i is transmitting, and $R_i(B_i, C_i) = 0$ when the agent does not transmit.

The DEC-POMDP approach easily allows many other kinds of objective functions, that is, other measures of communication quality over the network. For example, to minize the amount of dropped data, the reward function could be expressed with the queue size, agent action, capacity, and traffic source state variables, where any data that does not fit into the queue incurs a penalty.

Goal of the DEC-POMDP. Having defined the model and the reward function, the goal in an infinite-horizon DEC-POMDP is to compute agent policies π that maximize the expected discounted infinite-horizon reward $E\left[\sum_{t=0}^{\infty} \gamma^t R(s, \vec{a})|\pi\right]$, where γ is the discount factor.

Form of the policy. Here, the policy of an agent is a stochastic finite state controller (FSC) [19]. Fig. 2 shows an example FSC. A FSC is defined by a set of states and action and transition probabilities. When in FSC state q_i, agent i executes action a_i according to the action probability $P(a_i|q_i)$, observes o_i and moves from state q_i to state q_i' with transition probability $P(q_i'|q_i, o_i)$. In the experiments we used periodic FSCs [30], which allow optimization of larger controllers than with regular FSCs.

Optimality and efficiency. When the wireless network problem under investigation can be framed directly as a DEC-POMDP, the optimal solution to the DEC-POMDP yields optimal MAC protocols. However, as discussed earlier, the computational complexity of even discrete DEC-POMDPs makes exact policy computation intractable. Additionally the network model includes real-valued capacities and queue sizes and therefore we describe next approximations that enable efficient policy computation for the described factored DEC-POMDP.

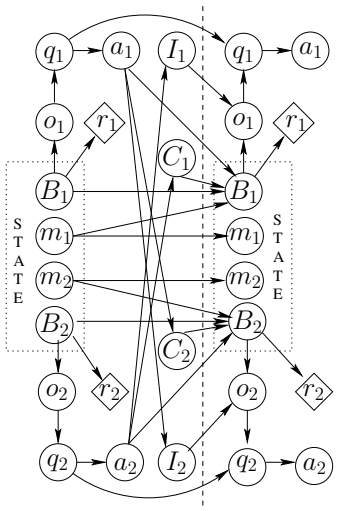

Fig. 3. Influence diagram for the two agent wireless network problem. The vertical dashed line separates the current and next time steps. B_1 and B_2 are the transmit queue state variables of agent 1 and 2, respectively. m_1 and m_2 are the source traffic model state variables. Together m_1, m_2, B_1, B_2 define the current state illustrated with a dotted box. The Shannon capacities at the receivers C_1 and C_2 and the observed interference powers I_1 and I_2 are determined by the actions a_1 and a_2 of the agents. The next state of transmit queue B_1 depends on the Shannon capacity C_1, the state of the source model m_1, the agents action a_1, and the current state of the transmit queue B_1. The agent observations o_1 and o_2 depend on the observed interference power and whether there is data in the transmit queue. Here, the reward (r_1 and r_2) is the negative transmit queue size for minimizing delay. q_1 and q_2 are finite state controller states.

4.2 Computing policies

Our approach for computing policies is based on the factored infinite-horizon DEC-POMDP method that we introduced in [19]. The idea is to use EM, a widely used approach for improving the likelihood of parameters in a probabilistic model, to improve the likelihood to achieve maximum reward (maximum throughput/minimum delay) of finite state controller parameters. The approach operates on a probability distribution, called the belief, over the world state s and FSC states $\vec{q} = (q_1, \ldots, q_N)$. One iteration of the approach shown in Algorithm 1 consists of computing an "average belief" and then scaling FSC parameters by reward probabilities (see [19, Section 3.3] and [19, Section 3.4] for details) accumulated when projecting the "average belief" forward. Because a belief over all state variables has exponential size w.r.t. the number of agents, we keep the belief in factored form: random variables are in possibly overlapping clusters with a limited number of variables in each cluster.

To project the belief from one time step to the next we apply the junction tree algorithm [31] on the two time step dynamic Bayesian network (DBN) [31] of the

Input: FSC parameters $\vec{\theta} = (\theta_1, \ldots, \theta_N)$ for N agents
Output: Optimized FSC parameters $\vec{\theta}$
while *Value of $\vec{\theta}$ increases and time limit not reached* **do**
  Compute "weighted average" belief $\hat{\alpha}(s, \vec{q})$ from projected beliefs. See [19, Section 3.2].
  foreach *Agent i* **do**
    1) Project $\hat{\alpha}(s, \vec{q})$ for different θ_i parameter values. **2)** Scale θ_i by reward probabilities accumulated during belief projection. See [19, Section 3.3].
  end
end

Algorithm 1: Our expectation maximization method for factored DEC-POMDPs (basic form of the pseudocode is based on [19], differences are in the model)

factored DEC-POMDP (Fig. 3 illustrates a two time step DBN for the wireless network problem described in Section 4.1). We project the belief directly into clustered form breaking dependencies between variable clusters. Note that even though some variables are assumed independent inside one time step, they are not assumed independent over multiple time steps (see [19] for details) and thus the actions of any agent can influence any other agent in policy computation, allowing the policy of each agent to take other agents into account as desired. Note also that the approximation error introduced by variable clustering is bounded over time [32], because mixing of the Markov chain attenuates the effect of approximation errors in the past. For practical wireless network problems with N agents the computational complexity of one EM iteration is $\mathcal{O}(N^2)$ [19, Section 3.5].

To compute actual policies using the factored DEC-POMDP algorithm we use approximations and modifications to deal with continuous values and to speed up the algorithm. We will next describe modifications that are specific to the wireless network problem; for completeness generic modifications are described in 4.2.2.

4.2.1 Wireless networking modifications

In order to perform wireless network optimization with the discrete factored DEC-POMDP model we have to solve the following problems: (1) capacities are real valued, (2) capacity and transmit queue observations may be real valued, (3) the transmit queue size is real valued and the range of values may be large, (4) the capacity of a transmitter depends on all other transmitters. We solve problems 1 and 2 by a soft quantization trick: to represent a real value z for a discrete state variable s, we assign probabilities $P(s = \lfloor z \rfloor) = \frac{\lceil z \rceil - z}{\lceil z \rceil - \lfloor z \rfloor}$ and $P(s = \lceil z \rceil) = \frac{z - \lfloor z \rfloor}{\lceil z \rceil - \lfloor z \rfloor}$, where $\lfloor z \rfloor$ denotes rounding z down and $\lceil z \rceil$ denotes rounding z up. Thus the expected value of the discrete variable s is the desired value z. For example, to represent a transmission rate of 2.4 we set a 0.6 probability to transmit with rate 2 and a 0.4 probability to transmit with rate 3. To solve problem 3 we quantize the real valued queue size into ratios of the maximum queue size; in the experiments eight equally spaced ratios were used. Problem 4 is actually not DEC-POMDP specific: there are 2^N different combinations for which of N transmitters are transmitting, and any approach that considers these combinations must take a large N into account. Here we model the largest interferers (eight in the experiments) accurately and use an approximation for the remaining smallest interferers. When we compute the capacity C for a specific combination of transmitting largest interferers, we compute a lower bound capacity C_{lower} corresponding to when all smallest interferers are transmitting and an upper bound capacity C_{upper} corresponding to when all smallest interferers are not transmitting. In more detail, the capacities C_{lower} and C_{upper} for receiver i are computed using the SIR defined in Equation 2, where for each smallest interferer j: $x_j(t) = 0$ for C_{upper} and $x_j(t) = TP_j$ for C_{lower}. The capacity is then written as a combination of the upper and lower bounds: $C = C_{lower} p_{source} + C_{upper}(1 - p_{source})$, where p_{source} is the average probability that a source traffic model is inserting data into a transmit queue. This approximation assumes that all generated traffic of the smallest interferers is transmitted. p_{source} for receiver i is communicated to transmitter i together with the policy of the transmitter. p_{source} can be computed as the (weighted) average probability for the source Markov model of each smallest interferer j for receiver i to be in a "data" state. In the experiments, source models are identical, but for a good general solution, weight can be assigned directly to interferer j based on the capacity difference at receiver i, when j is transmitting vs. idle.

The DEC-POMDP algorithm requires that variables are clustered at each planning time step. We cluster the transmit queue state, source model state, and FSC state of each agent into a single cluster, because these variables are guaranteed to influence each other during one time step: for example in Fig. 3 (q_1, m_1, B_1) and (q_2, m_2, B_2) would form the clusters. In the experiments variable clustering was needed only for more than two agents.

4.2.2 Modifications to the general factored infinite-horizon DEC-POMDP method

For completeness we describe below details on how the DEC-POMDP method is modified to converge faster and then how the starting probability distribution for optimization can be made better.

Greedy nonlinear reward scaling. In order to get a fully probabilistic model for which EM optimization can be applied, the DEC-POMDP method scales the reward function $R(s, \vec{a})$ into a probability: $\hat{R}(r = 1 | s, \vec{a}) = (R(s, \vec{a}) - R_{min})/(R_{max} - R_{min})$, where R_{min} and R_{max} are minimum and maximum immediate rewards. The value of the current DEC-POMDP policies is computed using the reward probabilities. To speed up EM optimization we use a greedy approach by applying a further nonlinear scaling step which emphasizes the highest rewards, making the algorithm concentrate on

the most promising actions and converging faster (see Fig. 4h for an optimization speed comparison).

The greedy reward scaling is done by applying a softmax transformation to the reward probabilities: when optimizing the action (or state transition) for some controller node by EM, we compute reward probabilities for different possible actions (or transitions) i (for details see [19]). We apply the greedy scaling to these probabilities, and set $\tilde{r}_i = e^{\alpha \hat{r}_i} / \sum_j e^{\alpha \hat{r}_j}$, where $\tilde{r}_i$ is the reward probability after the softmax transformation, $\hat{r}_i$ is the original untransformed reward probability for action (or transition) i, and α is a constant that was in the experiments 500. To reduce numerical instability, we apply the same softmax function a second time to $\tilde{r}_i$, yielding the final transformed values $\tilde{r}'_i = e^{\alpha \tilde{r}_i} / \sum_j e^{\alpha \tilde{r}_j}$.

To ensure the greedy weighting based optimization does not decrease the computed policy value, α is halved whenever the value would decrease, and the update is retried with the new α; if α becomes less than 2 the softmax scaling is stopped, and further optimization proceeds as in the original EM of [19]. Thus the optimization never decreases the computed policy value (the computed value is an approximation, because of variable clustering discussed in Section 4.2.1).

Small added noise in parameter updates. In the EM parameter updates, we add a very small amount of noise to updated parameters to help escape local minima.

Improved initialization. The method in [19] assumes that optimization starts from the same probability distribution over world states in each iteration. However, the goodness of the initial probability distribution depends on the current policies of the agents. Therefore, optimization should start from the most likely probability distribution given the current policies and a long running time. Therefore, we ran in each iteration the Markov chain (defined by the agents' policies and DEC-POMDP transition and observation probabilities) until convergence and used the converged probability distribution as an initial distribution for optimization. Because of this modification we can also use a short planning horizon (denoted T in [19]) together with a discount factor of 1 and still get good results: in the experiments a horizon of 4 was used. A larger T would take future events better into account at the cost of more optimization time.

5 Experiments

We compare our DEC-POMDP approach to four other methods. We compare to a CSMA/CA version that uses a carrier sense threshold similar to IEEE 802.11, and to three CSMA/CA versions that tune the carrier sense threshold of each agent to maximize spatial reuse. We optimized the carrier sense thresholds in the spatial reuse CSMA/CA versions using the relationship between transmit power and sense threshold defined in [17], denoted "product CSMA/CA", and using the relationship in [12], denoted "ratio CSMA/CA". Based on [11] we also optimized carrier sense thresholds online, denoted "online CSMA/CA". In all CSMA/CA versions we optimize the maximum consecutive data transfer length. Next we discuss the experimental setup and the implementation of each approach. Then we outline the experiments and show results.

TABLE 1
Summary of experiment parameters

Parameter	Value
Time step length	20μsec
Maximum transmit queue size	10240 bytes
Data insertion into queue per time step	0 or 80 bytes
Transmission from queue per time step	0–80 bytes
Maximum CSMA/CA backoff window	2^{10} time steps (20msec)
Size of wireless agent area	$100m \times 100m$
Background noise power	0.01W

5.1 Setup

We use a discrete time simulation whose parameters are summarized in Table 1. One time step is 20 μsec; an agent can transmit 0 to 80 bytes per time step, giving a 4 MB/sec maximum rate. The traffic source model inserts either 80 bytes or no data at each time step to the transmit queue. The maximum transmit queue size is 10240 bytes; when new data would not fit into the queue old data is discarded. In the wireless network the line-of-sight path loss exponent is 2.3 and background noise power 0.01W. Each experiment consists of 10 random network configurations: each method is optimized for a maximum of one hour (the DEC-POMDP method converged much earlier) with the network configuration and then evaluated ten times for 50000 time steps starting with empty transmit queues. We report in Section 5.2 average values and 95% confidence intervals (computed using bootstrapping).

The network configuration is created as follows: transmitters are randomly placed on a $100m \times 100m$ field. A receiver is randomly placed within a predefined maximum distance from its transmitter, which is varied in the experiments. If the receiver is not within the $100m \times 100m$ field, its position is resampled until it is inside the field.

In our ad-hoc network model interferers may reside very close or very far from transmitting nodes. In the absence of network wide CSI knowledge at each transmitter, our transmitters determine their transmit power using path-loss (channel) estimates between their respective source-destination pairs, and set transmit power so that a given signal-to-noise ratio (as opposed to SIR) is achieved at destination node. Our model would allow alternative power control policies (like fixed transmit power), but for simplicity, the above solution is selected.

We also consider a more difficult scenario where transmitters have no direct measurements of receiver-side interference; the scenario is common, for example IEEE 802.11 uses transmitter-side interference measurements with carrier sensing. In these cases, our DEC-POMDP method can be used with any indirect estimate

of receiver-side interference in place of the direct value. In experiments we tried this scenario by estimating receiver-side SIR by a simple transmitter-side estimate

$$\widehat{\mathrm{SIR}}_i(t) = x_i(t)/I(t) \ , \tag{4}$$

where $I(t)$ is the interference observed at the transmitter at time step t. This corresponds to assuming that interferers are in the immediate vicinity of the transmitter and background noise is negligible. It turns out our method works well with such estimates too: it performs almost equally well as with direct measurements.

5.1.1 DEC-POMDP

We used the factored DEC-POMDP model in Section 4.1 to model the wireless network and the techniques in Section 4.2.1 to frame the problem as a discrete factored DEC-POMDP and optimize policies. In our model each agent has a transmit queue of 512 slots (each slot = 20 bytes). The maximum capacity is 80 bytes = 4 slots.

In the DEC-POMDP approach the policy of each agent is a finite state controller (FSC), that takes as input capacity observations and outputs transmit actions. We utilized in the experiments 5 different capacity observations and two different actions: "transmit"/"do-not-transmit". In the experiments we used four layer periodic FSCs (see [30] for details about periodic FSCs) with two states in each layer, resulting in a total of eight FSC states. Note that while a controller with eight states may seem small, together the individual controllers can produce very complicated behavior that may yield better performance than sharing a pre-defined protocol. We fix in each layer the action of the first state to "transmit" and of the second state to "do-not-transmit" in order to make FSCs easier to inspect by a human (see Fig. 2 for examples of optimized FSCs). We initialize FSC state transition probabilities randomly. Because random FSC state transitions can lead to a state with any action, an initial FSC chooses also actions randomly, but optimized FSCs can perform both deterministic and random access.

A FSC takes as input discrete observations, but during evaluation transmit queues and capacity observations are continuous valued. Therefore, we quantize observations during evaluation and input them into the FSCs. The quantized observations are enough to yield very good performance for our method in the experiments.

5.1.2 CSMA/CA

The CSMA/CA controller is implemented as a simple state machine. It takes as input an observation that tells whether there are interferers or not (physical carrier sensing is used) and outputs an action that tells whether to transmit or not. Next we describe how exponential backoff, post-backoff, and a (tunable) carrier sense threshold are implemented in the CSMA/CA controller.

5.1.2.1 Exponential backoff: Before transmission the controller listens to the channel at least for one time step. If the channel is sensed occupied or a packet collision occurs, the controller uses binary exponential backoff with a maximum waiting time of 2^{10} time steps.

5.1.2.2 Post-backoff: Because our model transmits data without division into packets, we allow the controller to transmit data for a certain limited time "max transmit length" (corresponding to a maximum packet length) before it has to do a two time step post-backoff, during which it does not transmit. Similar to IEEE 802.11 this allows other agents to use the channel. In the experiments the system was simulated for each CSMA/CA variant for different "max transmit length" values ($1, 2, 4, 8, 16, 32, 64, 128, 256$ time slots, corresponding to $20, 40, 80, 160, 320, 640, 1280, 2560, 5120$ μsec) and the value with best performance was chosen for each network configuration. Note also that adjusting the "max transmit length" changes the effective minimum contention window length relative to the pseudo packet length. In [6], [33] Medepalli et al. show that adjusting the minimum contention window length improves performance in IEEE 802.11 wireless LANs.

5.1.2.3 Carrier sense threshold: In wireless networks such as IEEE 802.11 the channel is assumed occupied when the sensed power level exceeds the carrier sense threshold. In our system, observations are Shannon capacities and the capacity sense threshold C_{th} is a ratio of the maximum capacity ($0 \leq C_{th} \leq 1$). If a CSMA/CA agent observes a capacity larger than C_{th} times the maximum capacity, then the channel is assumed to be unoccupied. For receiver-side observations the observed capacity is the actual capacity at the receiver. For transmitter-side interference measurements the estimated capacity is $C_i(t) = \log_2(1+\widehat{SIR}_i(t))$, where $\widehat{SIR}_i(t)$ is given in Equation 4.

Using standard assumptions about free-space path loss for IEEE 802.11, the capacity sensing threshold is 1 for the field size of 100m $\times$ 100m, that is, only one agent can transmit at the same time. This version of CSMA/CA is called here "CSMA/CA". In order to exploit spatial sparsity and allow more agents to transmit simultaneously, the "ratio CSMA/CA", "product CSMA/CA", and "online CSMA/CA" versions find a good carrier sense threshold value for each agent using the state-of-the-art principles in [12], [17], and [11], respectively. The ratio, product, and online versions of CSMA/CA were optimized over all combinations of both "max transmit length" and carrier sense threshold but basic CSMA/CA only over the "max transmit length" parameter.

"Ratio CSMA/CA" finds a single capacity sense threshold common to all agents; the best threshold maximizes the capacity of the network, under the assumptions of the network capacity analysis in [12] (see Section 5.1.4 for details). In the experiments the system was simulated for different capacity threshold values (0.0125, 0.025, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1) and the best value was chosen for each network configuration.

"Product CSMA/CA" finds a different capacity sense threshold for each agent. It uses the major conclusion in [17] that the product of transmit power and carrier

sense threshold, denoted with xs_i, should be constant (see Section 5.1.4 for details). In the experiments we simulated the network for different values of xs_i (0.04, 0.0625, 0.1111, 0.25, 1, 4, 9, 16, 25) and the best value was chosen for each network configuration.

"Online CSMA/CA" finds the same *carrier* sense threshold for all agents. Because online CSMA/CA requires a target SIR (see [11] for details), we simulated the network for different target SIR values $(0.195, 0.391, 0.781, 1.563, 3.125, 6.25, 12.5, 25, 50, 100)$ and chose the best one for each network configuration. The carrier sense threshold was updated every 1000 time steps according to measured SIRs as discussed in [11].

5.1.3 Movement and imperfect sensing

The experiments included also *transmitter movement* where transmitters moved into random directions for a specified distance. When hitting an edge a transmitter changed to a new random direction. For the experiments we computed the capacity and observation probabilities of the DEC-POMDP model as average probabilities over (100) sampled transmitter positions. CSMA/CA parameters were found similar to the other experiments.

For experiments with *imperfect sensing* we assumed that observed interference is Gaussian and thus interference power estimates, from a set of sensor samples, are chi-squared distributed. Similar to the movement experiment we computed the observation probabilities of the DEC-POMDP model as average probabilities over (100) observation error samples. Observation errors were also used in the search for CSMA/CA parameters.

5.1.4 Capacity threshold for product and ratio principles

For completeness, we provide details about the ratio and product CSMA/CA, specifically details on how finding the best single capacity threshold for all agents conforms with the ratio principle in [12] and how individual capacity thresholds are tuned in the experiments to conform with the product principle introduced in [17]. Readers not interested in these details may skip this section.

Kim et al. [12] show that the network has maximum capacity when the ratio $x_i/I_i^{(th)}$ of the transmit power x_i and the carrier sense threshold $I_i^{(th)}$ is constant over all agents i. The analysis of [12] assumes that the agents are in a densely populated network and that they interfere as if they were arranged in a hexagonal grid around the transmitter and that transmit powers and carrier sense thresholds of all agents are equal.

We use a threshold C_{th} for capacity which is the same for all agents, and show that under the assumptions of Kim et al. [12] it corresponds to a ratio of transmit power to carrier sense threshold which is constant over agents. For an agent i, the capacity is defined as

$$C_i = \log_2\left(1 + \frac{g_{i,i}x_i}{\sum_{j\neq i} g_{j,i}x_j + \sigma^2}\right)/C_{max} = \log_2\left(1 + \frac{g_{i,i}x_i}{I_i}\right)/C_{max} \quad (5)$$

where C_{max} is the capacity without interference, x_i is the transmit power, and $I_i = \sum_{j\neq i} g_{j,i}x_j + \sigma^2$ is the interference at agent i. Therefore the capacity threshold $C_i = C_{th}$ is achieved when

$$I_i = (x_i g_{i,i})/(2^{C_{th}C_{max}} - 1) \equiv I_i^{(th)} \quad (6)$$

where the right-hand side is the carrier sense threshold corresponding to the capacity threshold. Therefore the ratio of the power and the carrier sense threshold is

$$x_i/I_i^{(th)} = (2^{C_{th}C_{max}} - 1)/g_{i,i}\ , \quad (7)$$

where the right-hand side is constant since $g_{i,i}$ is constant over agents in [12]. Therefore, the best capacity sense threshold corresponds to the best transmit power and carrier sense threshold ratio.

"product CSMA/CA" finds a different capacity sense threshold for each agent. It uses the main conclusion in [17] that the product of transmit power and carrier sense threshold, denoted with $xs_i = x_i I_i^{(th)}$, should be a constant $xs_i = xs_{th}$. Therefore the corresponding capacity sense threshold for agent i is

$$C_i^{(th)} = \log_2(1+g_{i,i}x_i/I_i^{(th)}) = \log_2(1+g_{i,i}x_i/(xs_{th}/x_i)) = \log_2(1 + x_i/xs_{th}) \quad (8)$$

where the last equality follows in cases when the receive power is $g_{i,i}x_i = 1$ which holds in our experiments.

5.2 Results

We ran experiments for the DEC-POMDP method and the CSMA/CA methods for different maximum link lengths (distance between transmitter and receiver), agent numbers, and optimization objectives.

In the experiments in Fig. 4a,b,e,f,g,h the methods were optimized to minimize delay and evaluated by the resulting average delay. In the experiments in Figs. 4c and 4d all methods were optimized to maximize throughput and evaluated by the resulting sum throughput. In the experimental setup of Figs. 4b and 4d, observations were computed from interference levels that the transmitter observed, and in Figs. 4a,c,e,f,g,h the observations were computed from interference levels that the receiver observed (see Section 3 for details). (Original) transmitter and receiver positions were identical between different figures and methods.

Figs. 4a and 4b show the average delay for eight agents and different maximum link lengths. DEC-POMDP outperformed all the CSMA/CA based methods by a large margin. Because DEC-POMDP policies are optimized

using the network model that contains observation probabilities, hidden and exposed terminals do not degrade DEC-POMDP performance, when using transmitter-side observations instead of receiver-side observations: DEC-POMDP performance is roughly the same in Fig. 4b as in Fig. 4a, but for "ratio" and "product" CSMA/CA methods the performance decreases (delay increases) with transmitter-side observations.

Figs. 4c and 4d show sum throughput for eight agents and different maximum link lengths. When optimizing throughput there is no penalty for having too much data in the transmit queue in contrast to optimizing delay. To obtain momentary maximum throughput it suffices to have enough data for maximum transmissions. With receiver-side observations the DEC-POMDP method is the best for maximum transmitter-receiver distances of 75m, and has otherwise equal performance with product CSMA/CA. With transmitter-side observations the DEC-POMDP approach is the best by a large margin.

Fig. 4e shows the average delay for different numbers of agents and a maximum link length of 75m. For more than four agents the DEC-POMDP approach achieves the smallest delay, outperforming the other methods by a large margin. The similar performance of the DEC-POMDP approach and the best CSMA/CA approaches for two and four agents can be explained by the offered traffic load. Because the source traffic model used in the experiments inserts data into a transmit queue 38.81% of the time, the probability that two agents have data to transmit at the same time is 15.06%, assuming that the agents can always empty their queues. For few agents it is therefore relatively easy to find transmission opportunities, but with more agents it becomes important that the agents take advantage of all the opportunities.

Fig. 4f shows the average delay for eight agents when transmitters move. Although the performance of the DEC-POMDP approach degrades with movement it has the best performance over all movement distances (each distance is a separate experiment). The figure illustrates also that when the topology changes dramatically the DEC-POMDP model should be optimized for the new topology in order to attain maximum performance.

Fig. 4g shows the average delay for eight agents with imperfect sensing. A higher sensing error (fewer samples) decreases the performance of product and ratio CSMA/CA slightly, but not of the DEC-POMDP approach which takes observation uncertainty into account.

Because the basic CSMA/CA approach has effectively no carrier sense threshold in the experiments, it yields the same average delay for all maximum link lengths in Figs. 4a–g. Because the ratio and product based CSMA/CA methods and the DEC-POMDP approach can take advantage of spatial sparsity, they yield lower delay and higher throughput than basic CSMA/CA.

In Figs. 4a–g online CSMA/CA is outperformed by the product and ratio CSMA/CA approaches whose parameters are optimized offline. In online CSMA/CA the transmitter with the worst average SIR determines the threshold for other transmitters even if others have high average SIRs (see [11] for details). Online CSMA/CA outperforms basic CSMA/CA when transmitter-receiver distances are small, because SIR values are more homogeneous. The better performance of product CSMA/CA compared to ratio CSMA/CA can be explained by the capacity sense threshold being different for each device in product CSMA/CA, but the same in ratio CSMA/CA. The DEC-POMDP approach performs the best, because in the DEC-POMDP approach each device has an individual MAC protocol optimized for the spatial and temporal configuration. In the CSMA/CA approaches the underlying MAC protocol is the same for all devices.

Lastly, Fig. 4h illustrates the convergence of the DEC-POMDP approach without ("Naive DEC-POMDP") and with ("DEC-POMDP (our method)") nonlinear scaling (see Section 4.2.2 for details). Because EM iterations did not end exactly at the displayed optimization times, the average delay in an evaluation run was computed by linear weighted averaging over the results of the two EM iterations closest to the displayed optimization time.

6 Discussion and future work

In this paper, we consider medium access control (MAC) in wireless networks with buffered data. The goal is to optimize MAC protocols for different network configurations and different optimization objectives. In order to optimize the behavior of wireless agents over both spatial and temporal dimensions we frame the optimization problem as a factored decentralized partially observable Markov decision process (DEC-POMDP). The policies optimized by the DEC-POMDP approach incorporate both random access and deterministic behavior. In randomly generated wireless network configurations the DEC-POMDP approach outperforms CSMA/CA based comparison methods by a large margin, especially in cases where complex behavior is required.

Our contribution has strong advantages: Unlike previous approaches our solution is theoretically optimal under the assumed probability model for the network, thus our model is a good framework for studying the problem. The remaining concerns are about practical feasibility. We empirically show that the approach is feasible and yields very good results in reasonably sized problems. Our traffic model is also better than most traffic models incorporated in spectrum access methods due to more detailed source traffic and buffer modeling.

Using the flexibility of the proposed DEC-POMDP approach, future work includes utilization of multiple channels, using several transmit power levels or transmit rates, and experiments with different optimization goals.

Acknowledgments

JPe belongs to the CoE in Computational Inference Research. The work was supported by Nokia, TEKES, Academy of Finland decision 252845, and PASCAL2, ICT 216886. This publication reflects authors' views only.

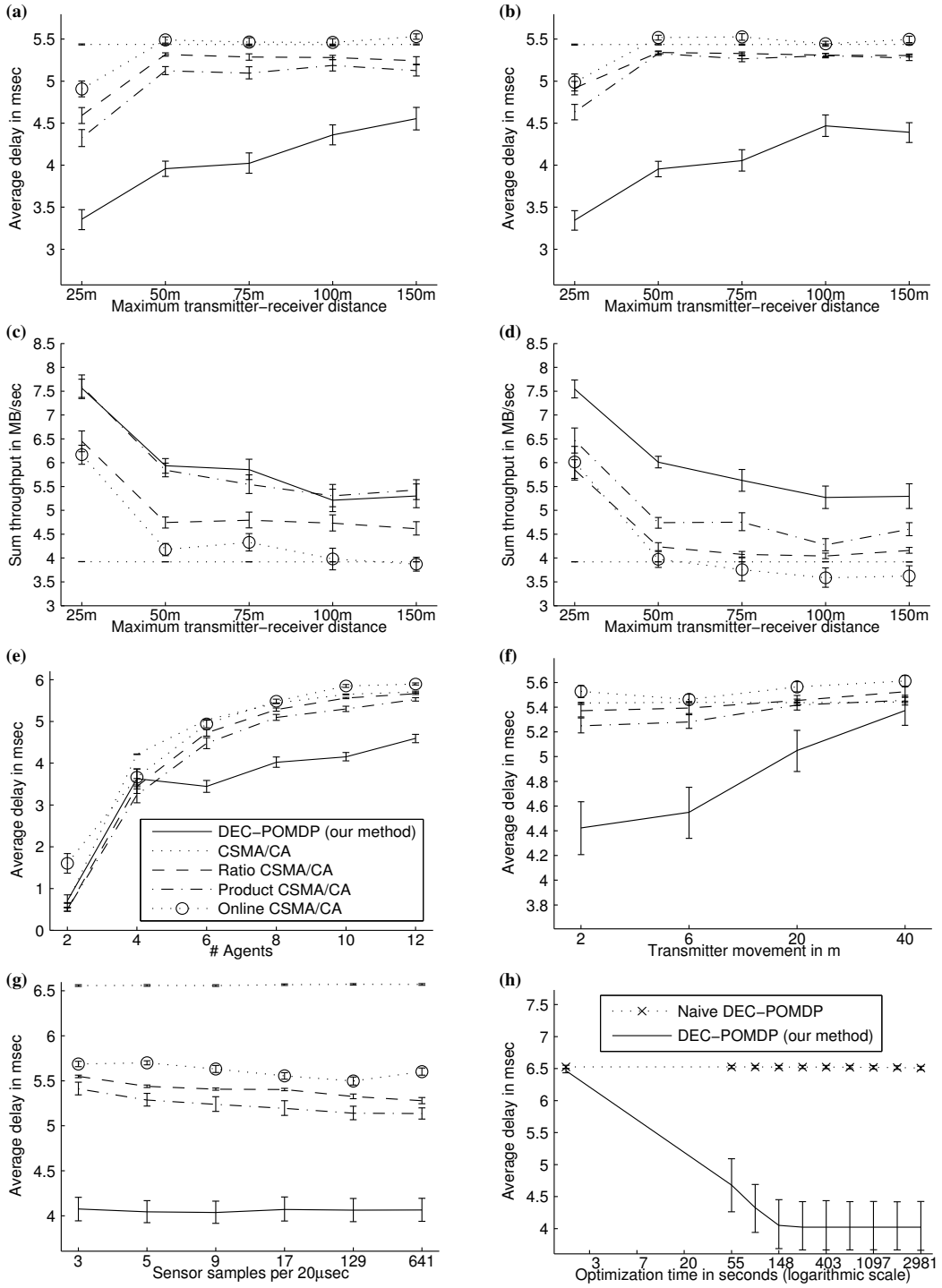

Fig. 4. Experimental results (legend in subfigure (e); separate legend for (h)). All experiments are averages over ten simulations; error bars denote 95% confidence intervals. **Performance as a function of maximum transmitter-receiver distance:** (a) shows average delay (smaller numbers are better) and (c) shows sum throughput (larger numbers are better), when interference is measured at the receiver. (b) and (d) show corresponding performance when interference is measured at the transmitter. DEC-POMDP outperforms other methods. Subfigures (e)-(h) show average delay (smaller numbers are better), with 75m maximum transmitter-receiver distance. **Performance a function of the number of agents** (e): DEC-POMDP outperforms the others for more than 4 agents. **Performance robustness to movement** (f): the horizontal axis shows different experiments where each experiment increases the distance that transmitters move; DEC-POMDP performs well. (g) shows **performance robustness to imperfect sensing**, using different numbers of samples for interference power estimation: DEC-POMDP has stable good performance. (h) shows **convergence** (average delay vs. optimization time) for naive DEC-POMDP ("Naive DEC-POMDP") and our approach ("DEC-POMDP (our method)") which uses nonlinear scaling: nonlinear scaling is crucial for fast convergence.

REFERENCES

[1] B. Alawieh, Y. Zhang, C. Assi, and H. Mouftah, "Improving spatial reuse in multihop wireless networks - A survey," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 71–91, 2009.

[2] S. Kumar, V.S. Raghavan, and J. Deng, "Medium access control protocols for ad hoc wireless networks: A survey," *Ad Hoc Networks*, vol. 4, no. 3, pp. 326–358, 2006.

[3] IEEE 802.11 Working Group et al., "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," 1997.

[4] C. Wang, B. Li, and L. Li, "A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 4, pp. 1235–1246, 2004.

[5] J. Choi, J. Yoo, S. Choi, and C. Kim, "EBA: an enhancement of the IEEE 802.11 DCF via distributed reservation," *IEEE Transactions on Mobile Computing*, vol. 4, no. 4, pp. 378–390, 2005.

[6] K. Medepalli and F.A. Tobagi, "On optimization of CSMA/CA based wireless LANs: Part I - Impact of exponential backoff," in *Proc. of IEEE International Conference on Communications (ICC)*. IEEE, 2006, vol. 5, pp. 2089–2094.

[7] M. Kaynia, N. Jindal, and G.E. Oien, "Improving the performance of wireless ad hoc networks through MAC layer design," *IEEE Transactions on Wireless Communications*, vol. 10, no. 1, pp. 240–252, 2011.

[8] A. Nasipuri, J. Zhuang, and S.R. Das, "A multichannel CSMA MAC protocol for multihop wireless networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*. 1999, pp. 1402–1406, IEEE.

[9] T. Luo, M. Motani, and V. Srinivasan, "Cooperative asynchronous multichannel MAC: design, analysis, and implementation," *IEEE Transactions on Mobile Computing*, vol. 8, no. 3, pp. 338–352, 2009.

[10] P. Nicopolitidis, G.I. Papadimitriou, and A.S. Pomportsis, "Learning automata-based polling protocols for wireless LANs," *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 453–463, 2003.

[11] J. Zhu, X. Guo, L. Lily Yang, W. Steven Conner, S. Roy, and M.M. Hazra, "Adapting physical carrier sensing to maximize spatial reuse in 802.11 mesh networks," *Wireless Communications and Mobile Computing*, vol. 4, no. 8, pp. 933–946, 2004.

[12] T.S. Kim, H. Lim, and J.C. Hou, "Understanding and improving the spatial reuse in multihop wireless networks," *IEEE Transactions on Mobile Computing*, pp. 1200–1212, 2008.

[13] J.C. Bicket, "Bit-rate selection in wireless networks," M.Sc. thesis, Massachusetts Institute of Technology, 2005.

[14] T. Korakis, G. Jakllari, and L. Tassiulas, "A MAC protocol for full exploitation of directional antennas in ad-hoc wireless networks," in *Proc. of ACM international symposium on Mobile ad hoc networking & computing (MobiHoc)*. 2003, pp. 98–107, ACM.

[15] J.S. Park, A. Nandan, M. Gerla, and H. Lee, "SPACE-MAC: Enabling spatial reuse using MIMO channel-aware MAC," in *Proc. of IEEE International Conference on Communications (ICC)*. 2005, vol. 5, pp. 3642–3646, IEEE.

[16] B.J.B. Fonseca, "A distributed procedure for carrier sensing threshold adaptation in CSMA-based mobile ad hoc networks," in *Proc. of IEEE 66th Vehicular Technology Conference: VTC2007-Fall*. 2007, pp. 66–70, IEEE.

[17] J. Fuemmeler, N.H. Vaidya, and V.V. Veeravalli, "Selecting transmit powers and carrier sense thresholds for CSMA protocols," Tech. Rep., University of Illinois at Urbana-Champaign, 2004.

[18] J. Fuemmeler, N.H. Vaidya, and V.V. Veeravalli, "Selecting transmit powers and carrier sense thresholds in CSMA protocols for wireless ad hoc networks," in *Proc. of 2nd Annual International Workshop on Wireless Internet, WICON'06*. 2006, p. 15, ACM.

[19] J. Pajarinen and J. Peltonen, "Efficient planning for factored infinite-horizon DEC-POMDPs," in *Proc. of 22nd International Joint Conference on Artificial Intelligence (IJCAI)*. 2011, pp. 325–331, AAAI Press.

[20] G.N. Shirazi, Peng-Yong Kong, and C.-K. Tham, "A cooperative retransmission scheme in wireless networks with imperfect channel state information," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*. 2009, IEEE.

[21] H. Li, "Multiagent Q-learning for aloha-like spectrum access in cognitive radio systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, pp. 56, 2010.

[22] A. Galindo-Serrano and L. Giupponi, "Distributed Q-learning for aggregated interference control in cognitive radio networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1823–1834, 2010.

[23] C. Wu, K. Chowdhury, M. Di Felice, and W. Meleis, "Spectrum management of cognitive radio using multi-agent reinforcement learning," in *Proc. of 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS): Industry track*. 2010, pp. 1705–1712, IFAAMAS.

[24] M. Bkassiny, S.K. Jayaweera, and K.A. Avery, "Distributed reinforcement learning based MAC protocols for autonomous cognitive secondary users," in *Proc. of Wireless and Optical Communications Conference (WOCC)*. IEEE, 2011, pp. 1–6.

[25] S. Geirhofer, L. Tong, and B. M. Sadler, "Cognitive medium access: constraining interference based on experimental models," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 95–105, 2008.

[26] J. Pajarinen, J. Peltonen, M.A. Uusitalo, and A. Hottinen, "Latent state models of primary user behavior for opportunistic spectrum access," in *Proc. of IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. 2009, pp. 1267–1271, IEEE.

[27] S. Seuken and S. Zilberstein, "Formal models and algorithms for decentralized decision making under uncertainty," *Autonomous Agents and Multi-Agent Systems*, vol. 17, no. 2, pp. 190–250, 2008.

[28] D.S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of Markov decision processes," *Mathematics of Operations Research*, vol. 27, no. 4, pp. 819–840, 2002.

[29] D.S. Bernstein, C. Amato, E.A. Hansen, and S. Zilberstein, "Policy iteration for decentralized control of Markov decision processes," *Journal of Artificial Intelligence Research*, vol. 34, no. 1, pp. 89, 2009.

[30] J. Pajarinen and J. Peltonen, "Periodic finite state controllers for efficient POMDP and DEC-POMDP planning," in *Advances in Neural Information Processing Systems 24 (NIPS)*, J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, Eds., pp. 2636–2644. 2011.

[31] K. Murphy, *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. thesis, University of California, Berkeley, 2002.

[32] X. Boyen and D. Koller, "Tractable Inference for Complex Stochastic Processes," in *Proc. of 14th Conference on Uncertainty in Artificial Intelligence*. 1998, vol. 98, pp. 33–42, Morgan Kaufmann.

[33] K. Medepalli, F.A. Tobagi, D. Famolari, and T. Kodama, "On optimization of CSMA/CA based wireless LANs: Part II - Mitigating efficiency loss," in *Proc. of IEEE International Conference on Communications (ICC)*. IEEE, 2006, vol. 10, pp. 4799–4804.

Joni Pajarinen received the M.Sc. degree from Helsinki University of Technology in 2004. From 2008 to 2012 he was a doctoral student at the Department of Information and Computer Science at Aalto University. He is currently a researcher at the Department of Automation and Systems Technology at Aalto University. His research interests include artificial intelligence, machine learning, robotics, and wireless networking.

Ari Hottinen received the DSc degree from Helsinki University of Technology in 2004. He was with Nokia in 1992-2012, most recently as a Principal Researcher. He is the CEO of Asparrow Ltd.. He is an inventor in over 70 wireless communications patent families. He has co-authored about 90 conference and journal articles including a book *Multi-antenna transceiver techniques for 3G and beyond*, Wiley 2003. He was a guest-editor of IEEE JSAC special issue on MIMO, 2008. He is a Senior Member of IEEE.

Jaakko Peltonen is an academy research fellow and docent at Aalto University, Department of Information and Computer Science. He received the D.Sc. degree from Helsinki University of Technology in 2004. He is an associate editor of Neural Processing Letters and has served in program committees of 16 conferences. He studies machine learning especially for exploratory data analysis, visualization, and multi-source learning.