Latent state models of primary user behavior for opportunistic spectrum access

Joni Pajarinen[1], Jaakko Peltonen[1], Mikko A. Uusitalo[2], and Ari Hottinen[2]
[1]Department of Information and Computer Science, Helsinki University of Technology,
P.O. Box 5400, FI-02015 TKK, Finland
[2]Nokia Research Center, P.O.Box 407, FI-00045 NOKIA GROUP, Finland
Email: {joni.pajarinen,jaakko.peltonen}@tkk.fi, {mikko.a.uusitalo,ari.hottinen}@nokia.com

***Abstract*—Opportunistic spectrum access, where cognitive radio devices detect available unused radio channels and exploit them for communication, avoiding collisions with existing users of the channels, is a central topic of research for future wireless communication. When each device has limited resources to sense which channels are available, the task becomes a reinforcement learning problem that has been studied with partially observable Markov decision processes (POMDPs). However, current POMDP solutions are based on simplistic representations where channels are simply on/off (transmitting or idle). We show that more complicated Markov models where on/off states are part of complicated behavior of the channel owner (primary user) yield better POMDPs achieving more successful transmissions and less collisions.**

I. Introduction

Wireless communication is growing: ever more devices communicate over wireless connections, and the data transmitted per device grows due to e.g. increasing wireless transmission of video. Furthermore, public infrastructure will gradually include for instance wireless backhaul over mesh network [1], and sensors for traffic, weather etc. will likely send increasing amounts of traffic over wireless networks.

As there is a limited amount of radio spectrum, more efficient use of the spectrum is important to avoid *congestion*. Congestion is partly due to rigid resource allocation in many wireless systems. *Cognitive radio systems* [2], [3], [4] aim to increase spectrum efficiency by opportunistic spectrum use: they adapt to the radio environment and learn to exploit underutilized radio channels for their own communication while protecting primary users (existing devices on the channels).

Current opportunistic spectrum access methods are often based on a potentially powerful probabilistic approach called partially observable Markov decision processes (POMDPs). However, current methods use only simple POMDPs where each radio channel is modeled with two states: "primary user is transmitting" or "channel idle". Such limited channel models neglect *latent behavior* of the primary user (PU) which is not directly measurable by immediate sensing of the channel: in particular they neglect complicated *dynamics* (evolution through time) of the PU behavior, and also neglect that the PU can *react* to collisions with cognitive radio transmissions.

We introduce a novel POMDP solution to opportunistic spectrum access, based on a more complicated Markov model which can represent dynamic behavior of each channel and can represent the way in which PUs listen and act on the channel. Experiments on data of WLAN channels with Voice over IP (VOIP) and web traffic show that our approach produces better access policies than the two-state approach.

II. Background

A cognitive radio (CR) system is a radio system that adapts to its environment and acts intelligently in it. In an early example of benefits of cognitive radio, local residential connections reused cellular channels opportunistically using interference avoidance [5]. Recently cognitive radio methods have emerged that sense the radio environment and learn to detect or predict when a given channel will be free of traffic [6]; such free "time slots" could then be used by the device for communication. Such *opportunistic spectrum access* would overall result in more efficient usage of the spectrum by several devices.

In opportunistic spectrum access, radio channels can contain existing communicating devices called *primary users* (PUs) that own the right to use that part of the spectrum. A CR device listening to the radio environment, called a *secondary user* (SU), could be allowed to use the remaining spectrum with a lower price if it does not harmfully disturb PU communication. The SU usually cannot spend enough power to sense all potentially idle channels, and must choose at each moment which channels to sense; also, often the SU cannot be certain the PU will not start to transmit at the same time as the SU; then the SU cannot achieve interference-free communication and must simply minimize the risk of interference.

This scenario can be represented as a *reinforcement learning problem*: the SU is rewarded for successful transmissions, penalized for using energy for listening and penalized a lot for interfering with PUs; the SU must find a behavior (action policy) maximizing the overall reward. The policy is optimized based on a probability model telling how channels behave after each action. Often, Markov models are used for the channels; then the interaction of SU actions, channel behavior, and

action rewards is called a partially observable Markov decision process (POMDP; see [6]). We use the discrete-time approach; observations and actions are made at fixed time intervals.

III. The Model

In POMDP learning the SU optimizes its action policy based on a Markov model telling the possible states and transition probabilities. A Markov model only approximates real channel behavior; such models are used as a simplification allowing intuitive and computationally feasible optimization of action policies. The better the model matches real channel behavior, the better the optimized policy performs in reality.

Typical POMDP solutions [6], [7], [8] to opportunistic spectrum access use simple two-state channel models: a channel is idle or has a PU transmitting, and transition probabilities between these two states depend only on the previous state; an action incurs a penalty if the SU transmits at the same time as the PU. We will describe why this simple channel model does not match real channel behavior well.

To get improved POMDP solutions, we now introduce a novel channel model representing more complicated behavior than the usual two-state model. In particular, we model two properties of PU communication on WLAN channels: *channel access patterns are dynamic*, varying even over short time periods, and *PUs react to collisions* where the SU transmits on a channel when the PU was listening or transmitting.

A. Basic POMDP setup

Assume there are N channels potentially available for a secondary user (a cognitive radio device) to transmit on. The channel access of the secondary user (SU) is divided into regular time slots. At the end of each time slot t, the SU chooses an action: what to do during the next time slot $t+1$. The SU can choose to listen to a set of M adjacent channels where $M < N$; during slot $t+1$ the action is carried out, and the SU observes for all those M channels whether a primary user (PU) was transmitting on them during slot $t+1$ or not. At time t the SU can also choose to transmit on exactly one of the M channels being listened to; careless choice of transmission channel can cause a collision with PU transmissions during slot $t+1$. The decisions to listen and transmit on a channel are made at the same time t; thus avoidance of collisions must be based on predicting the next channel states from observations up to time t. We model this task as a POMDP. A discrete POMDP [9] consists of a finite set of states S, actions A, observations O, transition probabilities $P(s'|s,a)$, observation probabilities $P(o|s',a)$, and rewards $R(s,a)$. We describe our states in Sections III-B to III-D, transition/observation probabilities in Section III-E, and rewards in Section III-F.

B. Channel dynamics

We assume a wireless network with multiple WLAN channels and primary users. A primary user (PU) of some channel uses an application (VOIP, web, etc.) which sends and receives data using the computing device network stack, that transmits packets over a wireless channel. Channel access patterns (i.e. the pattern of when a PU communicates or not over the channel) resulting from this sequence can be complicated: user behavior may be complex, network stacks are complicated and the wireless environment affects packet transfers.

We argue that the usual two-state probabilistic models of network traffic (corresponding to exponentially distributed lengths of idle and transmit periods) do not capture crucial characteristics of such traffic well: a large change to network traffic patterns may occur over a small time interval (e.g., a WLAN user may launch a new web application alongside old ones); multiple network protocols with different traffic patterns such as VOIP [10] and HTTP [11] may operate on the same channel; and changes to traffic intensity may change traffic patterns [11]. As a similar situation, in [12] it is shown that simple traffic models are not enough to model ethernet traffic.

We propose the following Markov model solution to channel dynamics: instead of a single "PU transmitting" state, each channel has K different transmit states T_i, $i = 1,..,K$ with different probabilities for continuing the transmission or ending it; this corresponds to modeling packet bursts of K different expected lengths. Similarly, each channel has J different "pause" states P_i where the PU is not transmitting, which model pauses of J different expected lengths.

When the channel leaves a transmit state, it can move to any of the J pause states, and when the channel leaves a pause state, it can move to any of K transmit states.[1] Thus the model can represent longer-term dynamics than a two-state model. The numbers of states K and J could be chosen by cross-validation; in practice they can be kept at a small value to save computational resources (we use $K = J = 3$).

We assume each channel has an independent PU; then the actions of the SU on one channel do not affect other channels and we can model the state of each channel independently.

Note that the SU may sense if a PU is transmitting but not the transmission type: the difference between the J pause states or K transmit states is *latent information* the SU must infer from recent observations. Intuitively, if the PU has kept transmitting for a long time, the transmission likely has a long expected length and the channel will be occupied in the next time step too, so the SU should not attempt to use that channel.

C. Reacting to secondary user actions

SUs are not controlled by the primary system, they are not in synchrony with PUs and SU transmissions cannot be fully coordinated, so collisions occur between PUs and SUs [7]. PUs may react to collisions. The reaction depends on the channel access method of PUs; as an illustrative example we assume PUs use a WLAN protocol such as WLAN 802.11a/b/g. *In these WLAN systems the PUs are required to sense the channel idle before transmitting, in the spirit of carrier sense multiple access [13] and 802.11 specifications.* For the SU we assume synchronous time slotted channel access.

If the SU transmits on a channel at the same time as the PU of the channel (a *transmission collision*), the primary user acts

[1] Before the channel begins actually transmitting it will go through a special "listen" state which we describe in Section III-C.

according to its protocol: it stops transmitting and listens to the channel until the SU has stopped and the channel is idle again; only then the PU can try to retransmit any packets corrupted by the collision and continue further communication. We assume each packet the PU was transmitting during the collision is corrupted and must be retransmitted, even if only part of that packet's transmission occurred during the collision.

If the SU transmits when the PU was listening to determine if the channel is idle (a *listen collision*), the PU notices the SU transmission and considers the channel non-idle, which delays the intended packet transmissions of the PU. Thus both transmit and listen collisions hurt the operation of the PU.

We model the PU listening and collisions situations by special states: before each packet burst (state T_i) the model must go through a corresponding, non-transmitting *listen state* L_i. If the SU starts transmitting just as the model would have moved to L_i, the model enters a *listen-collision state* LC_i (denoting that the PU notices the collision), where it remains as long as the SU transmits. The model enters a normal listen state when the SU stops. From a listen state the model moves to transmission, unless the SU starts transmitting; then the model enters a *transmission-collision* state TC_i; the PU then again tries to listen, entering a listen or listen-collision state. Transmit-collisions may similarly happen later in a packet burst if the SU interferes, with similar state transitions.

Our listen and collision states enable explicit modeling of PU reactions to SU actions. Note that a SU cannot tell the listening behavior of the PU from a pause by direct observation, so the SU must anticipate PU behavior to avoid interference.

D. The resulting state diagram

Figure 1 shows the diagram of states and possible state transitions on a single channel, as described in the previous subsections, when the SU does or does not transmit on that channel. In experiments we use $K = J = 3$ yielding fifteen states per channel. The state transitions of multiple channels occur independently given the SU action.

E. Estimating probabilities

The transition probabilities $P(s'|s, a)$ are probabilities for the system (the set of channels) to move from a state s (current state of all channels) to another state s', when the SU chooses action a (sensing, transmitting). In our model transitions depend only on SU transmission actions, not sensing. Since we assume PUs react deterministically to SU transmission (start listening to the channel if a collision happens), it is enough to learn transition probabilities $P(s'|s, \text{“no action”})$ for a channel under uninterrupted normal operation; other probabilities $P(s'|s, a)$ are deterministically derived from them.

We estimate probabilities $P(s'|s, \text{“no action”})$ from recorded network traffic data (packet start and end timestamps); such data can be captured over the air, taken from packet dumps, or generated by a network simulator as in our experiments. We convert timestamps into time slot data by assuming that if a PU accesses a channel at all during a time slot, the channel is occupied; such conversion does not cause underestimation of interference caused to PUs.

From the time slot data we identify *bursts*, consecutive series of the same slot type (idle/transmit). We arrange transmit-bursts into K clusters by length, so cluster 1 gets the shortest bursts and cluster K the longest, and each cluster (which may contain several bursts of different lengths) has roughly the same total of time slots; we then label all transmit time slots by the cluster label of that slot's parent burst. We next label listen states: if an idle slot comes just before a transmit burst with label T_i, it is labeled as listen state L_i. We lastly group idle-bursts (with listen-slots removed) into J clusters and label idle slots by the label of their cluster, like we did with transmit slots. Now the time slot data has been assigned to latent states. Probabilities are then estimated by standard means: transition probabilities are estimated by counting the number of transitions from state s to s' divided by the number of occurrences of state s, and stationary probabilities of states as relative proportions of counted time slots for each state.

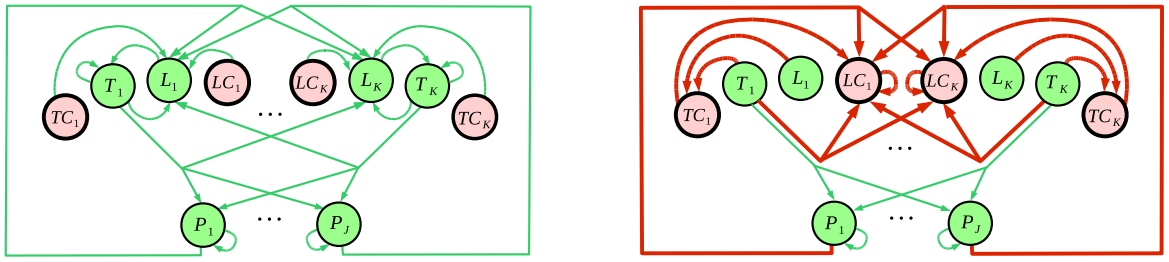
Observation probabilities $P(o|s', a)$ are probabilities for the SU to observe o, when performing action a and moving to state s'. The possible observations for the SU are the sensing results (idle/occupied) for each channel sensed. We simply assume the SU senses accurately for each listened channel, observing "occupied" if the PU is transmitting and "idle" otherwise. More generally, observation probabilities could be estimated from collected or simulated data without changing our model.

F. Setting rewards and learning the POMDP policy

The reward function $R(s, a)$ in the POMDP specifies the immediate value to the SU for performing action a in state s. Regardless of the state, we reward the SU for a transmit action with reward R_t and slightly penalize it for a sensing action (because it drains energy) with negative-valued reward R_s. Lastly, we strongly penalize the SU if s is a listen-collision or transmit-collision state, with negative reward R_c: this indirectly penalizes previous actions that led to the collision state. The R_t, R_s, and R_c could be set by negotiation between companies representing cognitive radio users and PUs.

The reward function, along with the state model and the probabilities, completes the definition of the POMDP. We may then optimize an action policy (which action to take given a sequence of prior actions and observations) for the SU by standard POMDP solvers. As usual, the policy is optimized to maximize an infinite-horizon discounted cost $E\left(\sum_{t=0}^{\infty} \gamma^t R^t\right)$ where γ is a discount factor (we used $\gamma = 0.95$), R^t the reward at t time steps into the future from the action, and the expectation is over the possible current and future states of the model given the known prior actions and observations.

To approximately solve the optimal action policy for our detailed channel models, we use the symbolic Perseus [9] package which has been used in POMDP problems with large state spaces [14]. In our model, independence of channels given the SU action speeds up computation.



(a) Secondary user does not transmit

(b) Secondary user transmits

Fig. 1. State transitions for one channel in our model. Transitions depend on whether the secondary user (SU) chooses to transmit on this channel (b) or not (a). The primary user (PU) can transmit K types of packet bursts and have J types of pauses. In normal operation, each packet burst i starts with a listen state L_i, continues with transmit state T_i for some amount of steps, and then the channel moves to a pause state or to the listen state of another packet burst; similarly, each pause state P_i lasts for some amount of steps and then the channel moves to a listen state of a packet burst. If the SU starts transmitting when the PU is about to listen or transmit, the channel moves to listen collision state LC_i or transmit collision state TC_i respectively. Collision states and transitions to them are shown with red color and thick lines. After collision, the PU listens until the SU does not transmit, then reattempts normal operation.

TABLE I
SUMMARY OF THE EXPERIMENT ENVIRONMENT

4-6 WLAN channels; VOIP+HTTP traffic; dynamic HTTP traffic level
Quantized channel occupancy times ($200\mu s$ slots)
1 PU per channel; 1 SU (can sense 3 channels and transmit on 1)
SU-PU collisions delay PU transmissions
SU is rewarded for transmissions, penalized for collisions and sensing

IV. EXPERIMENTS

We simulated a realistic wireless environment with either 4, 5 or 6 independent primary user (PU) channels with dynamically varying traffic conditions. Voice Over IP (VOIP) and Hypertext Transfer Protocol (HTTP) were used as network traffic. Table I summarizes the experiment environment.

We used the NS2 [15] software package and the VOIP modifications of [10] to generate network traffic. We used a "one-to-one" VOIP scenario with Weibull distributions to model talk spurts and packet delays in VOIP traffic. We used the PackMIME-HTTP [11] traffic generator to simulate HTTP connections between multiple clients and multiple servers. To generate dynamic PU behavior, we varied the connection rate of HTTP traffic in PackMIME-HTTP at three-second intervals, as the absolute value of an AR(1) process with Gaussian noise.

For testing we generated independently for each PU channel $15min$ of 802.11 WLAN traffic data with 54Mbit/s bandwidth. The traffic data (packet timestamps) was transformed to time slot data where each $200\mu s$ time slot is occupied if it contains part of a packet and idle otherwise. We similarly generated $30min$ of training data to estimate transition probabilities as discussed in Section III-E. Training data turned out to have slightly more frequent transmissions than test data, a realistic scenario where training and test data do not exactly match.

We used a reward of $R_c = -10$ for listen and transmit collisions, $R_s = -0.01$ for sensing a single channel, and $R_t = +1$ for a successful transmission by the SU. We ran tests separately using only the first 4 or 5, or all 6 channels. As actions, the SU was allowed to sense any three adjacent channels, to also transmit on one of them, or to do nothing, yielding 9, 13, and 17 possible combinations of the sensing and transmit actions for 4, 5, and 6 channels respectively.

We compared our model to comparable policies learned with two-state channel models. As a baseline we use a simple *listen, then send* random policy: if some channels were previously detected idle, transmit on a random one of them; for sensing, select 3 channels randomly (but so that they contain the transmission channel if the method chose to transmit).

To get *best two-state comparison results*, we ran four methods; each estimated probabilities from the $30min$ training data. We used two representations of collision penalties: R_c weighted by chance of collision (i.e. expected penalty; we ran this with the POMDP solver of [9]; we also used the solver of [16], but its results were usable only for 4 channels), or a third 'collision' state similar to collision states in our model (equivalent to a two-state model where reward is given based on the next state instead of the current state). We also used a hand-coded greedy approximation to an optimal two-state policy: maintain a belief state for the channels using the two state Markov model; at each time step, transmit on the channel with maximal expected one-step reward if it is positive (choose randomly if there is a tie), and sense the set of 3 channels with maximal probability to contain at least one idle channel in two time steps (but so that the transmission channel must be one of the 3 channels; choose randomly if there is a tie). Out of the four methods, we report the *best result* for each evaluation measure: this ensures that we compare our method against the highest-quality results of the two-state approach that we could get.

We learned our model and the comparison models from the training data, and evaluated on the test data. In testing we used realistic PU reactions to SU actions, listening (until the channel is free) in response to listen and transmission collisions and retransmitting all corrupted packets. We evaluate results by the total reward $N_tR_t+N_sR_s+N_cR_C$ where N_t, N_s and N_c are the total amounts of transmissions, sensing actions and collisions. We also list total rewards without counting listen-collisions, to see if the comparison methods perform better when only those collisions they can detect are counted.

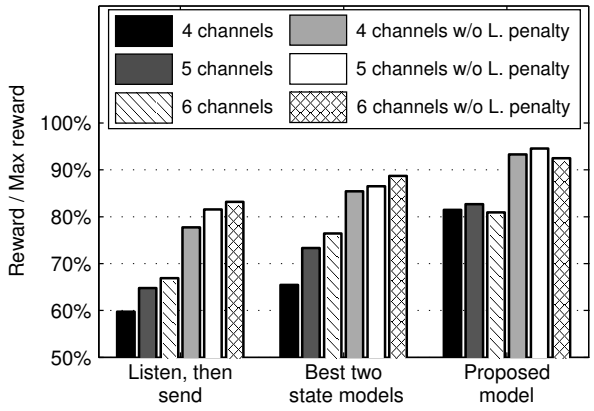


Fig. 2. Tests with dynamic HTTP+VOIP traffic. Total reward and total reward without listen collision penalties, divided by the maximum reward possible (4365000) for our proposed model, the best two state models, and the simple randomized listen-then-send model. Results of the proposed model on 6 channels are initial results from brief training, which already worked well.

TABLE II
TESTS WITH $C = 4, 5, 6$ CHANNELS FOR THE PROPOSED MODEL ("OUR") AND TWO-STATE METHODS: RANDOM LISTEN-THEN-SEND ("RAND."), TWO-STATE POMDP TRAINED WITH EXPECTED PENALTY USING THE SOLVER OF [9] ("2S EP"), AND HANDCODED GREEDY POLICY ("HAND.").

C	Policy	SU transmissions			Reward
		OK	T. Colls.	L. Colls.	(w/o L. pen.)
4	Rand.	4338539	88181	71325	2608479 (3393054)
	2S EP	4312751	56862	120522	2403929 (3729671)
	Hand.	4362320	76090	61019	2856230 (3527439)
	Our	4355080	19596	46875	3556300 (4071925)
5	Rand.	4359017	73088	66639	2826747 (3559776)
	2S EP	4334273	77045	88597	2542853 (3517420)
	Hand.	4393982	53441	52420	3200372 (3776992)
	Our	4420871	20560	47039	3609891 (4127320)
6	Rand.	4367683	66691	64637	2919403 (3630410)
	2S EP	4345373	68763	85738	2665363 (3608481)
	Hand.	4406408	44854	48704	3335828 (3871572)
	Our	4407858	28169	45783	3533343 (4036956)

Fig. 2 summarizes the results. The baseline method is naturally worst. Our proposed model is best on 4, 5, and 6 channel tests, regardless of whether listen-collisions are counted. Table II shows for each method (for the two-state approach we show two of the four methods for brevity) successful transmissions ("OK"), transmit ("T. Colls.") and listen ("L. Colls.") collisions, and total reward with and without ("w/o L. pen.") listen collision penalty. Our method has less transmit and listen collisions than other methods, and better total reward. The majority of collisions in our model are listen-collisions which is natural since they cannot be directly detected. Figure 3 shows our method operating on test data.

V. CONCLUSIONS AND DISCUSSION

More efficient and more dynamic spectrum usage is needed in the future. Opportunistic spectrum access by secondary users in the presence of primary users (PUs) is crucial for this. Compared to literature [6], [7], [8], we extended models

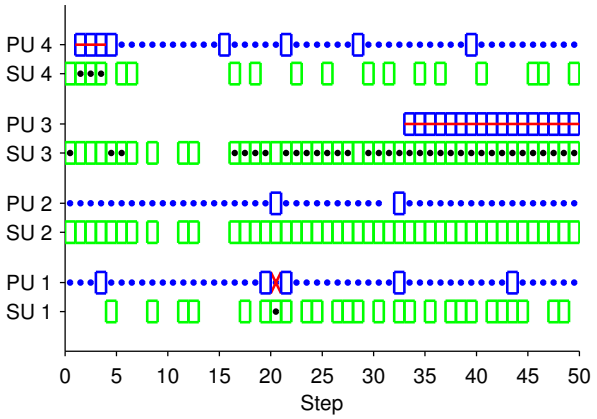


Fig. 3. Example of our proposed model on 4 channels, each with its own primary user (PU). One secondary user (SU) operates over the channels. For the SU and each PU, empty boxes denote sensing; dots denote transmissions; boxes with horizontal lines are listen collisions; dots with crosses are transmit collisions. The shown 50 steps were chosen from a test segment with 20 listen collisions, to show they are harder to avoid than transmit collisions.

of PU behavior from simple transmit/idle state models to better reflect real wireless traffic. Our models yielded more successful transmissions and less collisions, and hence larger communication capacity, than comparison models. Future work includes tests with more channels and real traffic data.

ACKNOWLEDGMENT

This work was supported by TEKES.

REFERENCES

[1] A. Alexiou, K. K. Leung, C. Papadias, A. Valkanas, and G. Paltenghi, "MEMBRANE: Multi-element multihop backhaul reconfigurable antenna network," in *Proc. IST Mobile Summit*, Myconos, Greece, 2006.
[2] J. Mitola III and G. Q. Maguire Jr., "Cognitive radio: making software radios more personal," *IEEE Pers. Commun.*, vol. 6, pp. 13–18, 1999.
[3] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, pp. 201–220, 2005.
[4] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Comp. Netw.*, vol. 50, pp. 2127–2159, 2006.
[5] M. O. Sunay, Z.-C. Honkasalo, A. Hottinen, H. Honkasalo, and L. Ma, "A dynamic channel allocation based TDD DS-CDMA residential indoor system," in *Proc. IEEE 6th Int. Conf. Universal Personal Communications*, San Diego, CA, USA, October 1997, pp. 228–234.
[6] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 3, pp. 589–600, Apr 2007.
[7] S. Geirhofer, L. Tong, and B. M. Sadler, "Cognitive medium access: constraining interference based on experimental models," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 95–105, 2008.
[8] S. Filippi, O. Cappe, F. Clerot, and E. Moulines, "A near optimal policy for channel allocation in cognitive radio," in *EWRL 2008, Revised and Selected Papers*. Springer, 2008, p. 69.
[9] P. Poupart, *Exploiting structure to efficiently solve large scale partially observable Markov decision processes*, Ph.D. thesis, University of Toronto, Toronto, Canada, 2005.
[10] A. Bacioccola, C. Cicconetti, and G. Stea, "User-level performance evaluation of VOIP using NS-2," in *Proc. Int. Conf. Performance evaluation methodologies and tools*, Brussels, Belgium, 2007, ICST.
[11] J. Cao, W. S. Cleveland, Y. Gao, K. Jeffay, F. D. Smith, and M. Weigle, "Stochastic models for generating synthetic HTTP source traffic," in *Proc. IEEE INFOCOM*, 2004, vol. 3.

[12] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Trans. Netw.*, vol. 2, no. 1, pp. 1–15, 1994.
[13] D. P. Bertsekas and R. G. Gallager, *Data Networks*, Prentice Hall, 1992.
[14] J. Hoey, A. Von Bertoldi, P. Poupart, and A. Mihailidis, "Assisting persons with dementia during handwashing using a partially observable Markov decision process," in *Proc. Int. Conf. on Vision Systems*, 2007.
[15] *The Network Simulator - ns-2*, http://www.isi.edu/nsnam/ns/.
[16] A. R. Cassandra, "Tony's POMDP page," 1999, http://www.cs.brown.edu/research/ai/pomdp/.