Performance Evaluation 68 (2011) 859–875

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

End-to-end protocols for Cognitive Radio Ad Hoc Networks: An evaluation study

Marco Di Felice [a,*], Kaushik Roy Chowdhury [b], Wooseong Kim [c], Andreas Kassler [d], Luciano Bononi [a]

[a] Department of Computer Science, University of Bologna, Italy

[b] Department of Electrical and Computer Engineering, Northeastern University, Boston, USA

[c] Department of Computer Science, University of California, Los Angeles, USA

[d] Department of Computer Science, Karlstad University, Sweden

ARTICLE INFO

Article history:
Available online 25 November 2010

Keywords:
Cognitive Radio Networks
Routing layer protocols
Transport layer protocols
Modeling and simulation

ABSTRACT

Cognitive radio ad hoc networks (CRAHNs) constitute a viable solution to solve the current problems of inefficiency in the spectrum allocation, and to deploy highly reconfigurable and self-organizing wireless networks. Cognitive radio (CR) devices are envisaged to utilize the spectrum in an opportunistic way by dynamically accessing different licensed portions of the spectrum. To this aim, most of the recent research has mainly focused on devising spectrum sensing and sharing algorithms at the link layer, so that CR devices can operate without interfering with the transmissions of other licensed users, also called primary users (PUs). However, it is also important to consider the impact of such schemes on the higher layers of the protocol stack, in order to provide efficient end-to-end data delivery. At present, routing and transport layer protocols constitute an important yet not deeply investigated area of research over CRAHNs. This paper provides three main contributions on the modeling and performance evaluation of end-to-end protocols (e.g. routing and transport layer protocols) for CRAHNs. First, we describe NS2-CRAHN, an extension of the NS-2 simulator, which is designed to support realistic simulation of CRAHNs. NS2-CRAHN contains an accurate yet flexible modeling of the activities of PUs and of the cognitive cycle implemented by each CR user. Second, we analyze the impact of CRAHNs characteristics over the route formation process, by considering different routing metrics and route discovery algorithms. Finally, we study TCP performance over CRAHNs, by considering the impact of three factors on different TCP variants: (i) spectrum sensing cycle, (ii) interference from PUs and (iii) channel heterogeneity. Simulation results highlight the differences of CRAHNs with traditional ad hoc networks and provide useful directions for the design of novel end-to-end protocols for CRAHNs.

1. Introduction

The increasing use of wireless access points, cordless telephones, television and computer peripherals in the 2.4 GHz industrial, scientific and medical (ISM) unlicensed band has resulted in growing congestion, and consequently spectrum scarcity. The need for using additional spectrum for unlicensed operation has been pointed early in this decade by the Federal Communications Commission (FCC) [1]. Recently, this effort has gained further impetus by opening up the frequency ranges vacated due to the analog-to-digital television transition for use by unlicensed operators [2]. This allows devices

[*] Corresponding author.
E-mail address: difelice@cs.unibo.it (M. Di Felice).

0166-5316/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.peva.2010.11.005

to sense channels that are not currently used in the UHF and VHF bands, and transmit on them opportunistically. Such devices are enabled with cognitive radio (CR) ability that allows them to *sense* the spectrum utilization, *select* the best available spectrum, and *share* the available spectrum resource among other devices. There has been a great interest in both the academia and industrial organizations in the development of devices and protocols for CR technology. However, the extension of CR techniques in distributed ad hoc networks (CRAHNs) is not straightforward, with several open challenges posed by the limited spectrum knowledge at the individual nodes, and the de-centralized operation [3].

Cognitive radio ad hoc networks (CRAHNs) [3,4] are usually composed of two kind of users: CR users and primary users (PUs). PUs have a license to access the licensed spectrum. CR users access the licensed spectrum as a ''visitor'', by opportunistically transmitting on the spectrum holes. The research community has mainly focused on the development of efficient spectrum sensing and selection techniques, thereby providing valuable insights on how such CR networks may be designed. However, for CRAHNs, many of these assumptions need to be revised in light of the following challenges:

- Common control channels (CCC) must hence be created so that network-wide information broadcasts and policy updates can be quickly disseminated. As each CR node may experience different spectrum availability, identifying these control channels shared by all the wireless nodes is a problem.
- CR nodes may share information about the spectrum availability in the local environment. How to leverage this spectrum data sent by immediate neighbors is an open research area.
- While infrastructure-based networks can establish single-hop connections between the CR nodes and the centralized base stations, the nodes in a CRAHN need to forward packets over multiple hops. The resulting routing and transport layer issues arising out of the end-to-end operation is unique to CRAHNs.

While recent works have proposed solutions toward control channel design [5] and CR network cooperation [6,7], the design of higher layer protocols, such as at the routing and transport layers, is still in a nascent stage. One of the main impeding blocks in CRAHNs is the lack of comprehensive simulators that are capable of handling a large number of dynamically varying spectrum parameters, and integrating in a seamless manner the adaptations undertaken at the different layers. As an example, the choice of transmit power must be undertaken such that it (i) improves the SNR at the physical layer, (ii) limits intra-CR collisions based on the transmission range at the link layer, and (iii) maintains the probability of interference to the licensed or PUs of the spectrum, which is often a policy decision. In another case, the choice of spectrum decides not only the throughput (different spectrum bands experience varying levels of PU activity during which the CR network must cease operation) but also the propagation distance of the transmission packet, and hence the end-to-end path length. These examples clearly demonstrate the high level of inter-dependency that exists among the different protocol stack layers. At present, no general-purpose simulator for CRAHNs has been developed, and existing network simulators (e.g. NS-2, Opnet, Qualnet) exhibit some important restrictions, such as (i) they do not provide any model of the cognitive cycle, i.e. spectrum sensing, decision and mobility algorithms, which are required to model CR users activities, (ii) they do not adequately address interference and signal detection issues at the PHY layer, which are required to model PUs activities, and (iii) they do not allow us to easily implement cross-layer information sharing among protocols at different layers. To this aim, in this paper we propose NS2-CRAHN, an extension of the NS-2 simulator [8], which is designed to support realistic simulation of CRAHNs. NS2-CRAHN allows a researcher to easily change a wide range of parameters, such as power and spectrum, channel bandwidth and frequency, use custom-designed or existing spectrum selection and sharing policies, and model accurately the activity of the PUs and CR users.

We envisage that our proposed simulation platform will be useful in a wide range of protocol design scenarios. Its inherent support for distributed operation makes it suitable for routing and transport protocol design. In this paper, we first investigate the suitability of different route discovery algorithms and routing metrics on end-to-end performance. In a CRAHN, periodic spectrum sensing and spectrum handoff operations may result in localized path disconnections between two CR users. Our simulation results show that sensing- and switching-induced delay has a negative impact on the stability of the path. We compare the performance of single-path [9] and multi-path routing protocols [10], and we find that the latter are able to prolong the end-to-end connection, reducing the number of route discoveries initiated by each source node. Moreover, spectrum selection and next hop determination can no longer be considered as isolated functions, and must be jointly undertaken. Our simulator shows how the integration of route selection metrics such as (ETT [11], ETX [12], hop count) with spectrum availability is achieved for several classical routing techniques by adapting them in a multi-spectrum environment.

Another important contribution of our work is a thorough analysis of existing TCP variants, such as TCP Reno, TCP NewReno [13], TCP SACK [14], TCP Vegas [15] for CRAHNs. To the best of our knowledge, this is the first work that addresses the analysis of TCP over CRAHNs at both macroscopic level, i.e. computing the TCP aggregated throughput, and at microscopic level, i.e. studying the dynamics of the round trip time (RTT) index and congestion window (CW) size, over simulation time. Specifically, we show how the effect of PU activity, periodic spectrum sensing, and bandwidth variations caused by channel switching affects the end-to-end performance. As the TCP source node must infer the presence of congestion based on packet drop or timeout, it is critical to understand how TCP operations are affected by the above factors, which are not present in traditional wireless ad hoc networks. As an example, during spectrum sensing and switching, there is a small delay in which the CR node scans the channels and coordinates the new spectrum with its neighbors, respectively. At these intervals, the path becomes virtually disconnected, with the sensing or switching node unable to forward packets in either direction. This event may, inadvertently, trigger the TCP timeout conditions at the source thereby causing it to mistake the spectrum-related function as network congestion. Simulation results highlight the problems of existing TCP variants and provide the basement for the design of novel transport layer protocols for CRAHNs.

Our proposed simulator design is modular, thereby allowing the addition of newer functionalities easily. Thus, while we provide a default functionality for say, spectrum sensing, the user may easily replace this block with a custom-designed algorithm. Similarly, other functions of choosing the best spectrum and the logic for how best to vacate a re-claimed spectrum are also implemented as default. The underlying parameters that influence higher layer protocol design, such as spectrum, channel bandwidth, and transmission power, are made available to each of the layers of the protocol stack through a cross-layer module. Other channel-specific characteristics, such as fading or noise can also be defined in a separate channel block. Recent measurements have revealed spectrum-specific PU activity patterns, which can be easily used to update the PU activity block provided in our simulator [1]. Thus, we aim to create a fully customizable tool that appeals to researchers at different levels: for the higher layer protocol designer, the underlying CR functions are already implemented, while for a physical layer researcher, each of these parameters can be altered and tested.

The rest of this paper is organized as follows: Section 2 describes the related work for existing performance evaluations of routing and transport protocols over CRAHNs. The system model considered in our work is given in Section 3 and the detailed description of our proposed simulator tool is given in Section 4. The performance evaluation of our simulator is undertaken in Section 5, and finally, Section 6 concludes the paper.

2. Related works

In this section, we review the current proposals in the literature addressing end-to-end solutions for CRAHNs, in particular: (i) routing layer protocols and (ii) transport layer protocols.

2.1. Routing layer protocols for CRAHNs

Routing constitutes an important yet not deeply investigated area of research in CRAHNs [3]. Based on to the relationship between the spectrum management and the routing algorithm, the routing design in CRAHNs is divided into two categories [16]: decoupled design and collaborative design. In the decoupled design, the spectrum decision and the routing selection are carried out independently by the MAC and network layers. In [17], the RACON routing protocol is proposed, by designing a link cost metric which takes into account the spectrum availability over time, for of each link. Three routing metrics for CRAHNs are evaluated in [18]. In the collaborative design, the spectrum decision and the routing selection are integrated into a single task at the network layer. The source node decides the end-to-end route and the spectrum used by each link simultaneously. Although the collaborative design is complex and is sensitive to spectrum fluctuation, it provides better end-to-end performance and it supports quality of service (QoS)-stringent applications [16]. In [4], we proposed the SEARCH protocol which combines geographical routing and spectrum allocation in order to avoid regions characterized by PU activity, and to determine the best path-channel combination which reduces the end-to-end delay. In [19], a joint approach of on-demand routing and spectrum band selection is proposed for CRAHNs, and a delay-based metric is used for evaluating the quality of candidate routes. A spectrum-aware data-adaptive routing algorithm is described in [20]. In this algorithm, the end-to-end route selection depends on the amount of data to be transported. In [21], the routing protocol SAMER builds a forwarding mesh upon a set of candidate routes to the destination, which opportunistically adapts during the forwarding process to the dynamic spectrum conditions. Most of the previous schemes are based on-demand routing scheme (like AODV [9]) and discover a single path between a source and a destination node. Moreover, in most of these papers, the authors do not provide details about the modeling of the cognitive environment and of the link-layer protocol used for the simulation analysis. In this paper, we do not propose a novel routing scheme for CRAHNs, but we discuss the impact of cognitive issues over the routing process, considering an accurate modeling of PU and CR user activities. In particular, we focus on the following issues: (i) impact of sensing-induced delay on route discovery and usage, (ii) impact of PUs on route usage, (iii) comparison between single path vs. multi-path routing approaches and (iv) impact of different routing metrics on end-to-end performance.

2.2. Transport layer protocols for CRAHNs

Since its original proposal in 1974, several versions of TCP have been proposed for wired networks [15,13,14]. All of them provide congestion and source-rate control, by means of a CW which limits the total number of un-acknowledged packets which can be in transit end to end. Modern implementations of TCP operate in four different protocol phases: *slow start, congestion avoidance, fast retransmit* and *fast recovery*. In the slow start phase, the CW size grows exponentially by one segment for each TCP–ACK received. In the congestion avoidance phase, the CW size is increased by one segment per round-trip-time (RTT), till a packet loss occurs. Packet losses are used as an indicator of congestion. In such a case, the rate is decreased by reducing the CW size at the sender side. Different versions of TCP differ in the way they detect and react to packet losses [15,13,14].

Many recent papers have investigated and provided evidence that TCP performs poorly on wireless environments, and on mobile ad hoc networks (MANETs) in particular. Simulation studies for wireless networks have investigated the impact of hidden terminals, wireless channel errors and nodes' mobility on TCP performance[22]. At the same time, many

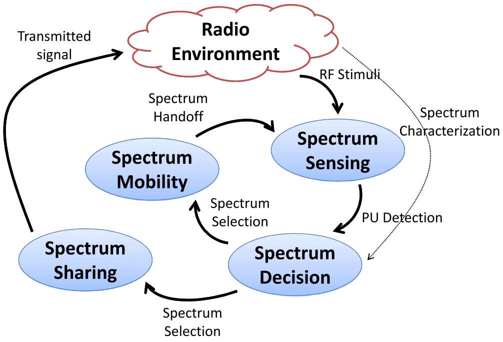

Fig. 1. The cognitive cycle implemented by each CR user.

transport layer protocols for MANETs have been proposed, including (i) completely novel transport protocols [23], (ii) TCP-like schemes [24] and (iii) link-layer and queue management schemes, supporting TCP over wireless networks [25,26].

While TCP over traditional ad hoc networks constitutes a well-investigated research area, there is a lack of papers addressing transport protocols for CRAHNs [27]. In [28], a novel transport protocol (TP-CRAHN) is proposed for CRAHNs. TP-CRAHN retains the window-based approach of TCP, but it also comprises additional protocol phases to adapt the protocol behavior to specific network conditions, such as sensing cycle intervals, spectrum and mobility handoffs, channel bandwidth variations, etc. In [29], an evaluation of TCP over dynamic spectrum access (DSA) links is proposed. Compared to the work presented in [29], we provide the following novelties: (i) we separately analyze the impact of different CRAHNs characteristics such as sensing time, PU activity and channel heterogeneity, on TCP performance (ii) we study the TCP behavior at both macroscopic level e.g. measuring the aggregate throughput, and at microscopic level e.g. analyzing the RTT and CW dynamics over time, and (iii) we address modeling and simulation issues for CRAHNs, which are described in Section 4.

3. System model

In our model, CR users forming the CRAHNs are equipped with M radio transceivers, which can be tuned to any channel in the licensed bands. We assume that N “heterogeneous” channels are present, where different channels may have dissimilar raw channel bandwidth. In a practical implementation, it might be expensive to equip a node with one dedicated radio for each channel since the number of channels may be large. Thus, we assume in this paper that $M < N$, i.e. the number of radios per node is smaller than the number of available channels.

Several multi-radio link-layer solutions have been proposed in the literature to coordinate the presence of multiple channels and multiple transceivers on wireless nodes [30,31]. One widely accepted solution for traditional wireless ad hoc networks is the on-demand channel assignment scheme [32], which divides the available channels into one CCC and multiple data channels. The available interfaces are divided into one control interface and several data interfaces. The control interface is tuned to the common control channel and is used to resolve the contention on the data channels. However, this solution is not feasible for CRAHNs, due to the problem of establishing a CCC in the licensed band [5]. The link-layer solution adopted in this paper is based on the interface assignment strategy proposed in [33], which classifies the available interfaces into *fixed* and *switchable* ones. Here, fixed interfaces stay on a specified *fixed channel* for longer time intervals. Switchable interfaces can switch among the available channels and are used to maintain the network connectivity. A distributed protocol is assumed which assigns a channel to the fixed interface, which can change over time according to e.g. traffic load. In contrast to [33], in our model CR users vacate the fixed channel as soon as PU activity is detected and select a different one. For the sake of simplicity, we assume that $M = 2$, i.e. each CR user is equipped with one fixed interface and one switchable interface.

The PU activity is modeled by the exponential ON–OFF model [34], where each PU has two states: ON and OFF. An ON (busy) state represents the period in which the channel is occupied by a PU, while an OFF (idle) state represents the period in which the channel is idle and can be used by CR users. The switching between the ON/OFF states is regulated by a birth–death Markovian process. Let α_i be the death rate for PU on channel i (e.g. PU_i), then the duration of the ON state follows an exponential distribution with mean $1/\alpha_i$. Similarly, let β_i be the birth rate for PU_i, then the duration of the OFF state follows an exponential distribution with mean $1/\beta_i$.

The CR user activity is modeled by using the cognitive cycle model described in [3]. In Fig. 1, we depict the four components of the cognitive cycle model: the *spectrum sensing* block, the *spectrum mobility* block, the *spectrum decision* block and the *spectrum sharing* block. Each CR alternately senses the channel and transmits data with sensing time equal to t_s and transmission time equal to T_o. If the *spectrum sensing* block detects a PU in the specific portion of the spectrum in use, then the CR user should vacate the spectrum and continue its communication in another portion of the spectrum. The handoff and protocol-reconfiguration are performed by the *spectrum mobility* block. Also, the CR must select a new channel to use based on its QoS requirements. The *spectrum decision* block is responsible for the channel selection. Otherwise, if the current channel is found free from PU activity, then the CR user may transmit on it, by using MAC layer coordination schemes

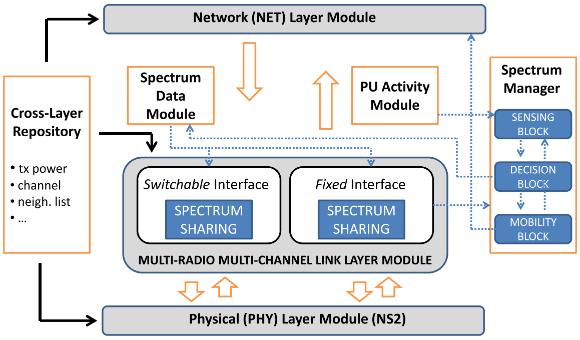

Fig. 2. NS2-CRAHN architecture.

(implemented in the *spectrum sharing* block) which prevent collisions with other CR users using overlapping portion of the spectrum [3].

4. Simulation of CRAHNs using network simulator 2

Based on the system model presented in Section 3, we have extended the NS-2 simulator with various blocks to be able to evaluate the performance of various approaches for CRAHNs. Fig. 2 shows the architectural model of the NS-2 CRAHN simulator. Compared to the traditional NS-2 architecture, we have added the following features, implemented as extendible stand-alone C++ modules:

- *PU Activity module*. It describes the characteristics of active PUs in the current scenario, including operating channel, physical location, and transmitting range. It also contains the description of PU activity in each spectrum band, as a sequence of ON and OFF periods over simulation time. All the information about PUs are contained in a PU-log file, which is generated offline. The format of PU-log file is described in Section 4.1.
- *Spectrum data module*. It describes the PHY characteristics of each channel, such as operating frequency, channel capacity and average bit error rate (BER). The format of channel-log file is described in Section 4.2.
- *Spectrum manager module*. It implements the cognitive cycle for each CR user, as described in Section 3. It is composed of three main blocks: the spectrum sensing block, the spectrum decision block and the spectrum mobility block. The spectrum sensing block is responsible for detecting the activity of PUs on the current channel. To this aim, the spectrum sensing block interacts with the PU activity module. In the case of PU detection, the spectrum decision block chooses the policy to be adopted, i.e. whether to switch to a new channel or to stay on the current channel. If a channel switch is required, the spectrum decision block can choose the next available channel to be used for CR user operation, and the spectrum mobility block manages the spectrum handoff process.
- *Multi-radio multi-channel link layer module*. It implements the multi-radio multi-channel environment for each CR user. Section 4.4 provides the details of the link-layer coordination, among the available radio interfaces. Each radio implements the spectrum sharing block for distributed channel access in wireless networks. Current implementation is based on the CSMA/CA MAC scheme. The spectrum sharing block interacts with the PU activity module to model the interference caused by PUs on current ongoing transmissions of CR users.
- *Network layer module*. Traditional routing protocols for wireless ad hoc networks can be used at network layer, as well as customized network protocols for CRAHNs. We also provide an implementation of the SEARCH routing scheme for CRAHNs [28].
- *Cross-layer repository module*. It enables information sharing among protocols at different layers of the protocol stack. For example, it may contain information collected at the PHY layer (e.g. current transmitting power), MAC layer (e.g. current size of the backoff window) and network layer (e.g. current neighbors' list).

In the following, we provide a detailed description of the PU-log (Section 4.1) and channel-log file (Section 4.2), of the spectrum manager functionalities (Section 4.3) and of the rationale of the CR user model (Section 4.4).

4.1. PU activity log file format

The PU-log file contains information about (i) PUs location and characteristics and (ii) PUs activity over time. The first part of the file is composed of entries with this format:

$$\langle id, x, y, channel, tpower \rangle \tag{1}$$

where *id* is the unique identifier of a PU, *x*, *y* are its location, *channel* and *tpower* are the channel and transmitting power used by the PU, respectively. The second part of the file describes the activity of each PU over time, by means of a list of entries with this format:

$$\langle id, arrival_{time}, departure_{time} \rangle \tag{2}$$

where *id* is the PU identifier, $arrival_{time}$ is the simulation time when the PU enters the ON period and starts transmitting, and $departure_{time}$ is the simulation time when the PU enters the OFF period and ceases transmitting. For each PU (e.g. PU_i), average ON and OFF time are regulated by an exponential distribution with mean $\frac{1}{\alpha_i}$ and $\frac{1}{\beta_i}$, based on the ON/OFF model described in Section 3. We develop a script shell program which takes as input the α and β parameters for each PU and generates the corresponding activity over simulation time.

4.2. Channel-log file format

The channel-log file contains information about (i) physical channel characteristics and (ii) channel quality. The spectrum data module is responsible for loading the information from the file and making them available to the other modules. For each CR channel, the log file contains an entry with this format

$$\langle id, frequency, bandwidth, noise\rangle \tag{3}$$

where *id* is the identifier of the channel (a number between 1 and N), *frequency* is the channel central frequency, *bandwidth* is the raw bandwidth of the channel (e.g. 11 Mb/s), and *noise* is the average value of the noise on that channel. By using *bandwidth* and *noise*, and by knowing the power received at a given location, it is possible to model the average BER experienced by the receiver node, for the QPSK modulation [35].

4.3. Spectrum manager

The spectrum manager module implements the cognitive cycle for each CR user, by using the spectrum sensing block, the spectrum decision block and the spectrum mobility block.

4.3.1. Spectrum sensing block

Channel sensing is modeled as a lookup function on the PU-log file, for the current channel (say channel i), sensing interval t_s and simulation time t. The readings from the file are then adjusted on the basis of the sensing accuracy, so that the final outcome of the sensing process is a binary response whether a PU signal is detected or not on the current channel.

More in details, a CR user (say node C) performing sensing on channel i at time t checks the PU-log file if there is an entry for a PU (say P) satisfying these two conditions (**C**):

- **C1**: P is transmitting on the same channel i, or on an adjacent[1] channel, for the time interval $[t; t + t_s]$;
- **C2**: The amount of power injected on channel i by node P and received by node C (i.e. P_r^C) is higher than the sensitivity threshold P_{th}^C:

$$P_r^C \geq P_{\text{th}}^C. \tag{4}$$

Signal propagation is modeled through a generalized free-space model, i.e. P_r^C is computed by node C as follows:

$$P_r^C = \frac{P_t \cdot C_t}{(d^\alpha)} \cdot k \tag{5}$$

where P_t is the transmitting power of P, the constant C_t captures different transmission properties such as the antenna gains and height, α is the attenuation factor, d is the physical distance between P and C, and k is the overlap factor between channel i and the central channel frequency used by P. We consider the overlap factors as 1, 0.5 and 0.25 for 0, 1 and 2 channel spacings from the PU's central channel frequency, respectively. Overlap factors can be adjusted to accommodate for different systems and spectral overlap mask characteristics (see the discussion on the I-factor from [36]). If both conditions **C1** and **C2** are verified, then the variable *puON* is set as true. However, this is not enough to guarantee a correct PU detection, because node C might misdetect the presence of P. The probability of correct detection P_d is described in [34]. In the current implementation, we use a simplified equation for P_d, as a function of the sensing time interval (t_s) only. A PU signal is detected if *puON* is set as true, with probability P_d. In this case, a notification is sent to the spectrum decision block.

4.3.2. Spectrum decision block

The spectrum decision block is responsible for (i) deciding the spectrum policy and (ii) choosing the next channel to be used by CR users, in case a PU activity detection requires a channel switch. Each time a PU activity is detected on the current channel, a CR user can adopt two different policies:

- *Switch policy*. The CR user immediately vacates the current channel and moves to another portion of the frequency spectrum.
- *Stay and wait policy*. The CR user does not vacate the current channel, but stops transmitting on it, because it must not interfere with the PU activity. In this case, a notification is sent to the sensing block. As soon as the PU activity ceases, the CR user re-starts its operations on the current channel.

[1] We define two channels e.g. i and j to be adjacent if their channel spacings $|i - j|$ is ≤ 2.

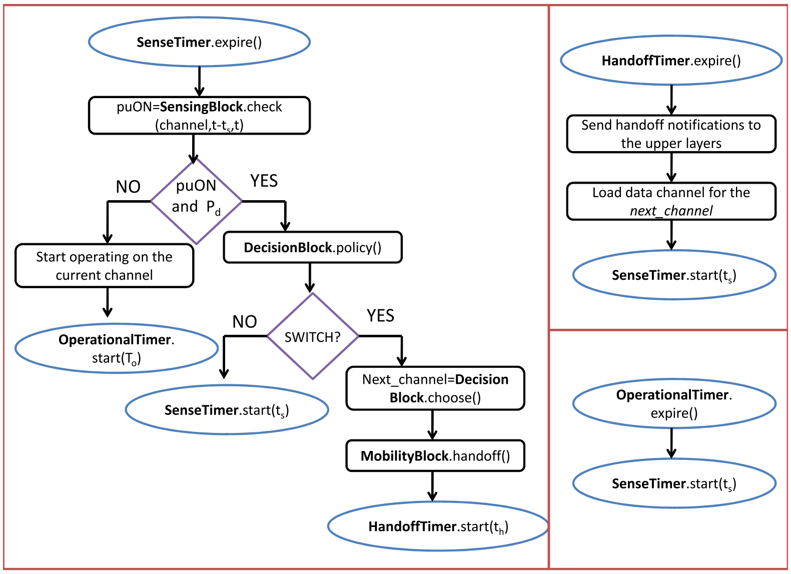
SenseTimer.expire()
puON=SensingBlock.check (channel,t-t_s,t)
puON and P_d
NO
YES
Start operating on the current channel
OperationalTimer. start(T_o)
DecisionBlock.policy()
SWITCH?
NO
YES
SenseTimer.start(t_s)
Next_channel=Decision Block.choose()
MobilityBlock.handoff()
HandoffTimer.start(t_h)
HandoffTimer.expire()
Send handoff notifications to the upper layers
Load data channel for the next_channel
SenseTimer.start(t_s)
OperationalTimer. expire()
SenseTimer.start(t_s)

Fig. 3. The CR model implemented in the NS2-CRAHN simulator.

In the case of channel switch, the CR user must find another portion of the spectrum for its operations. The channel selection can be performed at the link-layer (by the spectrum decision block) or at the routing layer. The current NS2-CRAHN implementation supports both these approaches. In the case of channel selection performed by the spectrum decision block, three channel allocation schemes can be used:

- *Random allocation*. The CR user chooses randomly among the available N channels.
- *Sequential allocation*. The CR user visits all the N channels, by a round-robin algorithm:

$$next\_channel = (current\_channel + 1)\%N. \qquad (6)$$

- *PU-aware allocation*. The CR user chooses the channel providing the highest transmission opportunities, by considering the amount of interference injected on each channel by each PU and by neighbor CR users.

We emphasize here that additional spectrum policies and spectrum allocation algorithms can be easily integrated into the current module, by considering metrics provided by the MAC, physical or routing layer, or any combination among them. This also facilitates mechanisms that allow a CR user to temporarily route around the PU activity on a different frequency portion until the current PU activity stops.

4.3.3. Spectrum mobility block

The spectrum mobility block is invoked when the spectrum decision block decides that the CR user must vacate the current channel. It receives from the spectrum decision block the new channel to switch to (e.g. *next_channel*). The delay induced by the channel switch is modeled by using a timer. A CR user is not allowed to utilize the radio interface for communication during the handoff operation. When the handoff process is completed, the spectrum sensing block is invoked to detect the PU activity on *next_channel*. If *next_channel* is found free of PU activities, then a spectrum handoff notification is sent to the upper layer, and protocol reconfiguration is performed at the network layer.

4.4. CR user model

The CR user model is shown in Fig. 3.

4.4.1. Link layer management

Each CR user at initialization designates one interface as its fixed interface, and the second interface as the switchable interface. Moreover, each CR user periodically informs its neighbors about the channel used by its fixed interface, by broadcasting an HELLO message on all the available channels. When a CR user (e.g. node A) needs to communicate with a neighbor node (e.g. node B), it tunes its switching interface to the channel used by the fixed interface of node B, and starts transmitting.

Each CR user performs a sensing cycle on the fixed radio interface, by periodically switching between two states: a sensing state (for a t_s time interval) and operational state (for a T_o time interval). To this aim, each CR user (say node C) is associated with a sensing timer and an operational timer. When the sensing timer expires, C stops sending/receiving data on the current channel and performs sensing operations to detect the presence of a PU, as described in Section 4.3.1. In the case of channel switching to *next_channel*, a broadcast message is sent by node C to inform its neighborhood about the channel used by its fixed interface.

4.4.2. Spectrum sharing

Each interface implements a spectrum sharing scheme, based on a carrier sensing multiple access with collision avoidance (CSMA-CA) MAC scheme, with acknowledgments (ACK) and frame retransmissions at the MAC layer. Moreover, we extend the MAC scheme to take into account the interference caused by PUs on CR users. Each time a CR user (say node C) starts receiving a packet by another CR user (say C2), it checks if a PU (say P) is currently transmitting on the current or adjacent channel, or if it will start transmitting during the reception time. In such a case, the amount of power injected on the current channel by P is computed at node C, based on the propagation model described by Eq. (5). Based on P_r^C, the actual signal-to-noise ratio (SINR) is computed by node C. If the SINR is lower than a given threshold (which is usually referred in NS-2 as CPTresh), then node C discards the packet sent by node C2.

5. Performance evaluation of end-to-end protocols over CRAHNs

In this section, we provide an evaluation of the end-to-end protocols over CRAHNs, considering routing (Section 5.1) and transport (Section 5.2) layer protocols. We use the NS-2 simulation tool [8], with the extensions described in Section 4. Both Sections 5.1 and 5.2 are structured as follows. First, we introduce the simulation methodology and the metrics considered. Then, we describe the simulation setup, and we show the simulation results for each test. Finally, we discuss the design issues to enhance the performance of routing and transport layer protocols over CRAHNs.

5.1. Routing layer

Routing protocols for wireless ad hoc networks can be classified into reactive and proactive protocols [37]. Reactive protocols build on-demand paths between the source and destination nodes only when needed. In proactive protocols, each node maintains fresh routing tables for all destinations, by means of periodic distribution of information (e.g. link state) throughout the network. Therefore, proactive protocols reduce path acquisition time compared to reactive ones, but at the same time exhibit very slow reaction to network dynamics, such as nodes' mobility [37,38]. In CRAHNs, the dynamic spectrum allocation determines that each wireless link may experience different conditions over time, as a function of the interference of PUs, available bandwidth, and so on. For this reason, we focus our attention on reactive protocols, and we consider two approaches for route formation over CRAHNs:

- *Single-path routing*. We consider the AODV [9] routing protocol, which discovers a single path between a source and a destination node.
- *Multi-path Routing*. We consider the AOMDV [10] routing protocol, which discovers multiple paths between a source and a destination node. The discovered paths can be node disjoint, i.e. they have no nodes in common, or link-disjoint, i.e. they have no links in common. We use AOMDV with node-disjoint configuration in our analysis.

We consider two metrics for the performance analysis:

1. *Route discovery frequency*. This is the average number of route discovery procedures generated by the source node, for each second. A route discovery is initiated when a source node broadcasts a route request message (RREQ), containing the address of the destination node. It provides an indicator of the stability of the route discovered by the routing protocol.
2. *Packet delivery ratio* (PDR). This is the ratio of packets which are received by the destination node, over the number of packets sent by the source. It provides an indicator of end-to-end delivery capabilities of the routing protocol.

All the metrics described so far are measured at CR users since the focus here is to evaluate the impact of PU activity on the performance of routing protocols for secondary systems.

The CRAHN environment is constructed as follows. We assume that 10 PU spectrum bands are present (i.e. $N = 10$) with each band independently occupied by PUs, according to the ON/OFF model described in Section 3. The average duration of ON/OFF state is the same in all the spectrum bands. We let $\frac{1}{\alpha}$ and $\frac{1}{\beta}$ be the average ON and OFF period of a spectrum band, respectively. We consider random multi-hop CRAHN topologies, composed of 50 CR users in a simulated area of 1000×1000 m. Three UDP/constant bit rate (CBR) connections are established inside the CRAHNs, among three random pairs of CR source-destination nodes. Each connection may traverse multiple intermediate nodes before reaching the destination node. Unless stated otherwise, each CBR connection has a data rate of 250 Kbps, and the packet size is 1000 bytes. No mobility effect is considered at this stage.

In Section 5.1.1, we evaluate the impact of the sensing-induced delay on the link failure detection performed by the routing protocols. In Section 5.1.2, we consider the impact of the PU activity on link failures. In Section 5.1.3, we analyze the end-to-end performance when different routing metrics are used during route setup.

5.1.1. Sensing cycle analysis

Simulation setup. In the sensing cycle analysis, we consider the CRAHN environment described in Section 5.1, and we vary the duration of the sensing interval (t_s), while the operation interval (T_o) is fixed to 0.6 s for each CR user. All the channels

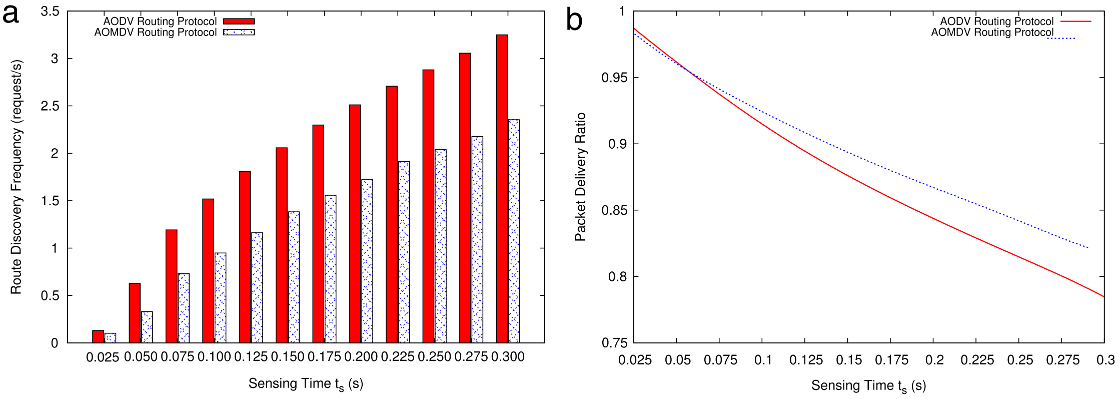

Fig. 4. The route discovery frequency for different configurations of the sensing time (t_s) is shown in Fig. 4(a). The packet delivery ratio in the same configurations is shown in Fig. 4(b).

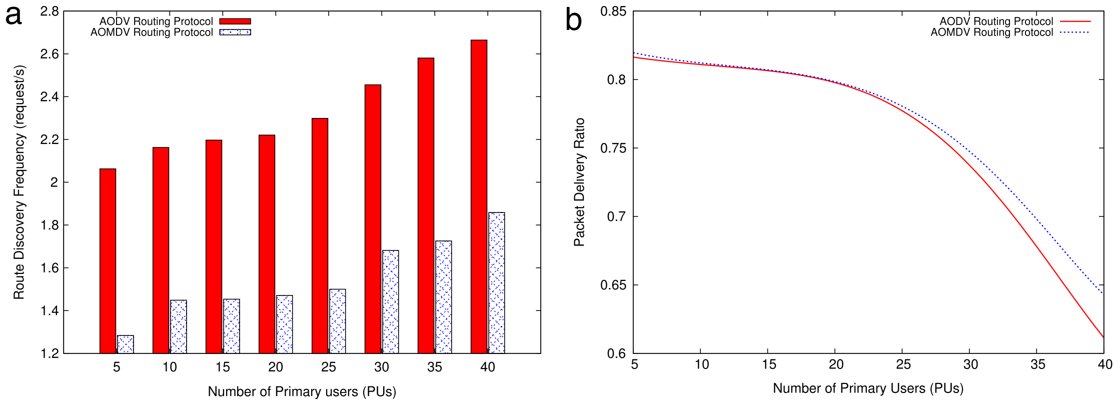

Fig. 5. The route discovery frequency for varying number of PUs is shown in Fig. 5(a). The packet delivery ratio in the same configurations is shown in Fig. 5(b).

have capacity equal to 1 Mb/s. We set the α and β parameters equal to 0.5, so that the average ON and OFF time of a channel is equal to 1 s in both cases. We assume ideal sensing without detection errors, i.e. P_d is equal to 1.

Simulation results. Fig. 4(a) shows the route discovery frequency for the AODV and AOMDV protocol, as a function of the sensing time (t_s). The route discovery frequency increases for both protocols with t_s, as a consequence of link failures caused by the sensing-induced delay. In fact, link failure detection is based on feedbacks from the MAC layer. According to the 802.11 DCF standard [39], a node (say node A) attempting to communicate with a neighbor node (say node B) sends an RTS message and waits to receive the CTS message. After seven RTS retransmissions have failed, the MAC layer notifies the upper layer that the link is down, and an ERROR message is sent back by node A to the source node. Then, a new route discovery procedure is initiated by the source node. When sensing is considered, each CR link is virtually disconnected during the sensing operation of the CR nodes. A false link failure may be triggered when node A attempts to communicate with node B, which is performing sensing and cannot reply with a CTS message. Fig. 4(a) shows that the risk of false link failure increases with longer values of t_s. At the same time, Fig. 4(a) shows that the end-to-end connectivity can be prolonged when multiple paths are used at the source node. In the AODV protocol, a new route request is initiated by the source node each time an ERROR message is received. In the case of AOMDV, a source node keeps multiple paths toward the destination and can switch to an alternate path when the current one is invalidated. As a result, AOMDV reduces significantly the route discovery frequency generated by each node (Fig. 4(a)). This reduction translates in higher stability of the end-to-end connection, and in higher end-to-end performance, as shown in Fig. 4(b), which depicts the PDR values for both the protocols. AOMDV provides higher PDR than the AODV scheme, because of the utilization of backup paths.

5.1.2. PU activity analysis

Simulation setup. In the PU activity analysis, we consider the CRAHN environment described in Section 5.1, and we vary the number of PUs operating in the scenario. The sensing and transmitting time interval are equal to 0.1 s and 0.6 s, respectively. We set the α and β parameters equal to 0.5, for all the PUs.

Simulation results. Fig. 5(a) shows the route discovery frequency for the AODV and AOMDV protocol, as a function of the number of PUs. When we increase the number of PUs, CR users experience higher probability to detect PU activity on

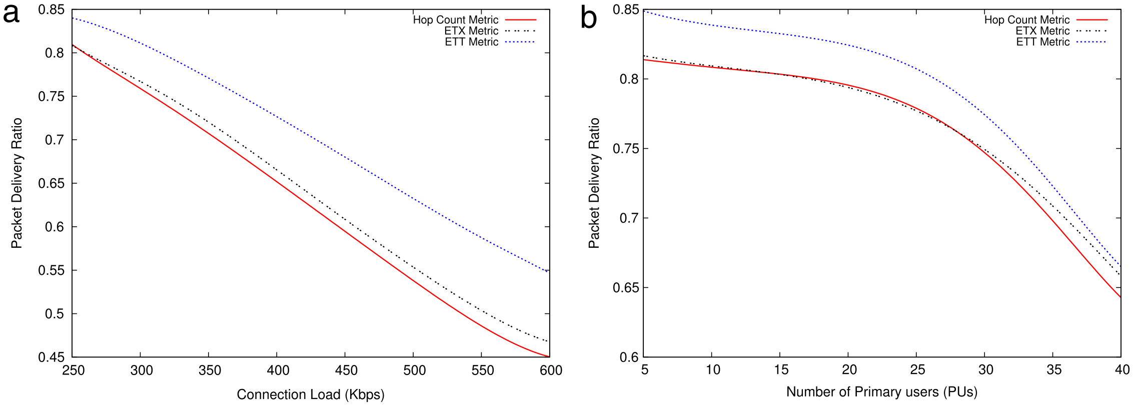

Fig. 6. The packet delivery ratio for varying load produced by each connection of CR users is shown in Fig. 6(a). The packet delivery ratio as a function of the number of active PUs is shown in Fig. 6(b).

the current channel, and to perform a spectrum handoff operation. As for the sensing-delay discussed in Section 5.1.1, the spectrum handoff delay introduces additional link failure events, because a node cannot reply to any RTS message while it is reconfiguring its hardware for a channel switch event. At the same time, Fig. 5(b) shows that the overhead of route discovery can be reduced when multiple paths are used, for the same considerations discussed above. AOMDV overcomes AODV also in terms of PDR, as shown in Fig. 5(b).

5.1.3. Routing metric analysis

Simulation setup. In the routing metric analysis, we consider the CRAHN environment described in Section 5.1 with these parameters: $\alpha = 0.5,\ \beta = 0.5,\ t_s = 0.1$ s, $T_o = 0.6$ s. We remove the assumption of channel heterogeneity, i.e. different channels have different raw bandwidths.

Simulation results. Fig. 6(a) and (b) show the PDR for the AODV protocol, when three different routing metrics are evaluated. The hop-count metric is the default AODV metric, and discovers the shortest path between the CR source and destination nodes. The expected transmission count (ETX) metric was proposed in [12], to model the average number of MAC transmissions required to send a unicast packet over a link. The routing algorithm then chooses the path with the smallest accumulated link ETX values along the path. The expected transmission time (ETT) [11] extends the ETX metric by considering the raw bandwidth of each link. The ETT value for link A-B is defined as $ETX(A, B) \cdot \frac{S}{B}$, where S is the payload of the packet (which is fixed and equal to 1000 bytes in our tests) and B the bandwidth of the link. Fig. 6(a) shows the packet delivery ratio for the studied metrics, as a function of the system load injected by each source node. The PDR decreases for all the metrics, when the system load increases. However, Fig. 6(a) shows that the shortest path routing is likely to produce suboptimal performance, because spectrum considerations are not involved into the route discovery. The ETX metric takes into account that different links may experience different BER, and thus guarantees higher end-to-end delivery than the hop-count metric. The ETT metric provides the highest performance in the CRAHN environment, because it takes into account both the packet losses and the bandwidth of each link. However, the ETT metric does not include considerations about PU activity on each spectrum, and thus about the stability of each link. We do not propose a new metric for CRAHNs at this stage. However, we think that a routing metric for CRAHN should combine (i) the hop-count, (ii) the spectrum quality and (iii) the spectrum availability for CR users, at each link. Fig. 6(b) shows the PDR as a function of the number of PUs operating in the multi-hop scenario. The ETT metric enhances the ETX and hop-count metrics, for the same considerations discussed above.

5.1.4. Considerations

Sensing-induced and spectrum handoff-induced delays have a considerable impact on route stability. Accurate link-failure detection schemes should be designed, in order to isolate the effect of nodes' mobility, from the effect of sensing cycle and spectrum handoff operations. Multi-path routing can be considered a viable approach to enhance the end-to-end performance, because of the geographical separation of the disjoint paths, which may experience different channel conditions. The source node can switch to an alternate path when the current one is invalidated, without initiating a new discovery process. Moreover, our analysis confirms that shortest path routing is likely to produce suboptimal performance over CRAHNs, because it does not consider the spectrum allocation on each link. Spectrum allocation determines the current bandwidth and quality of each link and may dynamically change over time, as a consequence of PU activity. We can conclude that novel routing metrics should be designed for CRAHNs, taking into account the PU activity on each link.

5.2. Transport layer

Since TCP is the de facto transport protocol standard on Internet, it is crucial to estimate its ability in providing stable end-to-end communication over CRAHNs. In this section, we investigate the performance of different TCP variants over CRAHNs,

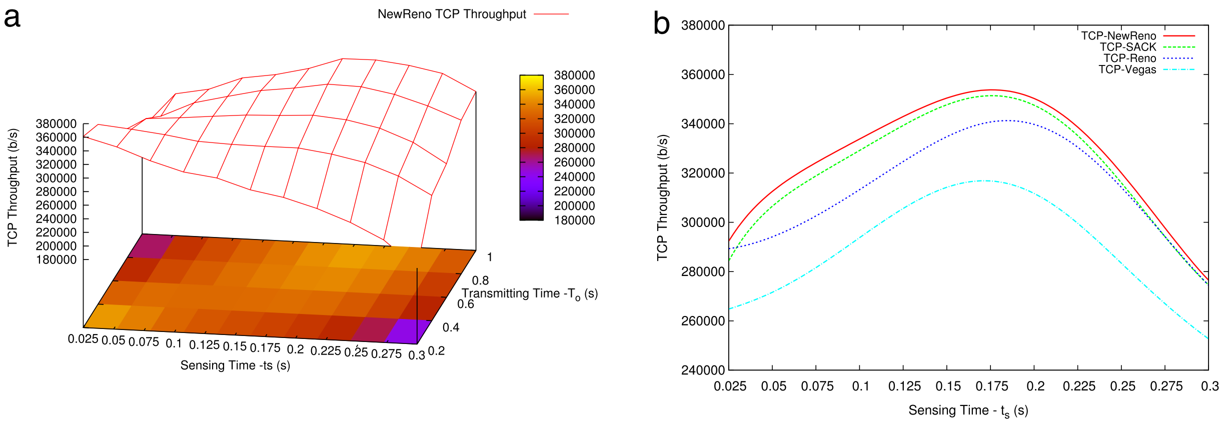

Fig. 7. The throughput of TCP NewReno for different configurations of sensing (t_s) and operational (T_o) time is shown in Fig. 7(a). The throughput for different TCP versions is shown in Fig. 7(b).

i.e. TCP Reno, TCP NewReno, TCP with Selective Acknowledgment (TCP SACK) and TCP Vegas. In this set of experiments, the CRAHN environment is based on five channels (i.e. $N = 5$) and constructed as follows. As in Section 5.1, we assume individual channels are occupied randomly and independently of each other by PUs, according to the ON/OFF model described in Section 3. The average duration of ON/OFF state is the same, in all the channels. We consider two different topologies: a static single-hop single-flow CRAHN scenario and a multi-hop multi-flow mobile CRAHN scenario.

In the first case, the CRAHN is composed of two CR users, and one TCP/FTP connection is established between them. No mobility effect is considered at this stage. We study the performance of TCP under different CRAHNs characteristics, e.g. the sensing time interval of CR users, interference caused by PU activity and bandwidth variation in an heterogeneous channel environment. The choice of the single-hop scenario can be motivated as follows. First, the single-hop scenario is simple enough to understand the impact of CRAHNs characteristics on the dynamics of TCP, while this might be difficult to investigate in multi-hop topologies. Second, the single-hop topology constitutes a basecase, from the point of view of protocol performance. If we discover that a single parameter, e.g. the sensing time interval, has a strong impact on TCP performance, then this effect will be emphasized in a multi-hop environment by the presence of multiple intermediate nodes between the source and the destination CR users. Moreover, although very simple, the single-hop topology constitutes a realistic model for the evaluation of infrastructure-based CR networks, where the mobile CR users are attached to a fixed cognitive base station (BS). Then, we consider the evaluation of TCP over multi-hop random topologies, where all the CRAHN characteristics are considered.

We consider two metrics in the performance analysis:

- *TCP throughput*: this is the end-to-end TCP throughput at the application layer, i.e. the amount of bits for seconds received by the upper layer FTP application at the destination node, without considering out-of-order, duplicated and TCP–ACK packets.
- *TCP efficiency*: this is an estimation of bandwidth resource utilization by TCP. It is defined as follows [29]:

$$\epsilon = \frac{TCP_{THR}(t_1, t_2)}{\int_{t_1}^{t_2} C(t) \cdot dt}, \quad 0 \le \epsilon \le 1 \tag{7}$$

where $TCP_{THR}(t_1, t_2)$ is the average TCP throughput computed over the measurement period from t_1 to t_2, and $C(t)$ is the available channel capacity at time t [29].

All the metrics described so far are measured at CR users since the focus here is to evaluate the impact of PU activity on the performance of the TCP protocol for secondary systems.

In Sections 5.2.1–5.2.3, we show the performance results of TCP in the single-hop topology, when we vary the sensing time interval of CR users (Section 5.2.1), the PU activity (Section 5.2.2) and the difference in capacity between adjacent channels (Section 5.2.3). In Section 5.2.4, we show the simulation results of TCP over multi-hop multi-flow random topologies, in which all the previous factors are considered.

5.2.1. Sensing cycle analysis

Simulation setup. In the sensing cycle analysis, we consider the basic single-hop topology, and we vary the sensing interval (t_s) and operation interval (T_o) of CR users. All the primary bands have capacity equal to 1 Mb/s. We set the α and β parameters equal to 0.5, so that the average ON and OFF time of a spectrum band is equal to 1 s in both cases.

Simulation results. Figs. 7(a) and (b), 8(a) and (b) show the TCP performance for the sensing cycle analysis. Fig. 7(a) shows the TCP NewReno throughput as a function of the sensing time (t_s, on the x-axis) and of the operational time (T_o, on the

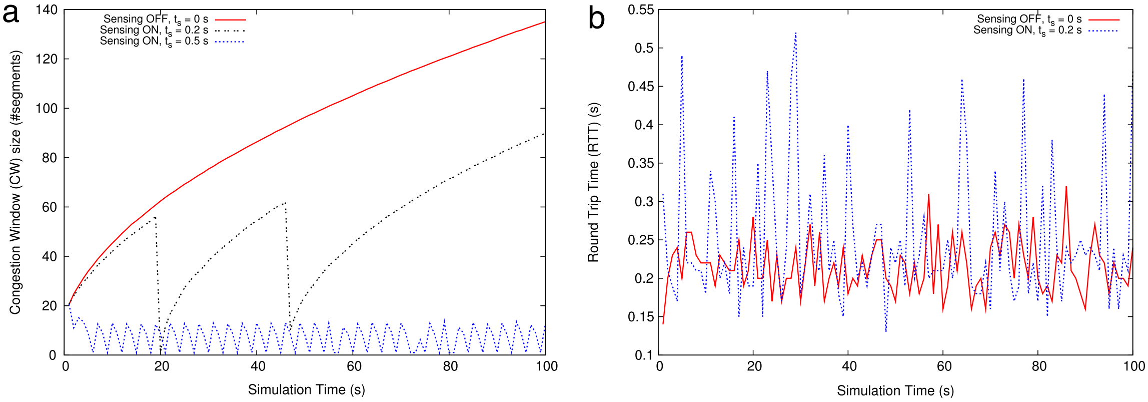

Fig. 8. The CW size over time is shown in Fig. 8(a), for different values of the sensing time (t_s). The RTT over time is shown in Fig. 8(b) for different values of the sensing time (t_s). TCP NewReno is used in both cases.

y-axis). Fig. 7(a) confirms that the selection of $\langle t_s, T_o\rangle$ parameters by CR users has a remarkable impact on the TCP performance. In particular, given $\langle \alpha, \beta\rangle$ parameters which describe the PU activity, there exists an optimal selection of $\langle t_s, T_o\rangle$ parameters which maximizes the TCP performance for CR users. In Fig. 7(a), the maximum throughput is reached when t_s and T_o are equal to 0.2 s and 1 s, respectively. This is in accordance with the theoretical analysis described in [34]. Fig. 7(b) shows the throughput of TCP NewReno in the same scenario, where T_o is fixed to 0.6 s, and t_s varies between 0.025 s and 3 s. In Fig. 7(b), we also show different throughput curves for different TCP versions. From Fig. 7(b), we can see that the throughput increases with t_s till a critical threshold is reached. When $t_s > 0.2$ s, all the protocols experience throughput degradation. This behavior is independent of the TCP version in use and can be explained by the fact that the selection of the sensing interval constitutes a trade-off between (i) accurate PU detection and (ii) efficient channel utilization. In Section 3, we introduced the probability of correct detection (P_d), as a function of t_s [34]. When $t_s < 0.2$ s, a CR user might misdetect the presence of the PU on the current channel. The PU interference causes TCP–DATA and TCP–ACK packet losses for CR users, and thus affects the TCP performance. When $t_s > 0.2$ s, the throughput decreases because of the sensing-induced delay. Retransmission timeout (RTO) events may be triggered at sender side because of TCP–ACK packets which are delayed at the MAC layer, even in the absence of true congestion. RTO events are interpreted as indicators of network congestion and force the TCP sender to reduce the CW size to one segment. This translates in TCP performance degradation when $t_s > 0.2$ s. Moreover, Fig. 7(b) shows that TCP SACK and TCP NewReno overtake TCP Reno and TCP Vegas in terms of throughput. Surprisingly, TCP Vegas provides the lowest performance for all the configurations of t_s. This might be because TCP Vegas tries to estimate the available bandwidth to avoid congestion rather than react to it. It uses the measured RTT to accurately calculate the number of data packets that a source can send. Due to PU activity and sensing time, this available bandwidth estimation might not be the correct one leading to low throughput. As a result TCP Vegas bandwidth estimation mechanism needs to be adjusted to work effectively over CRAHNs.

In Fig. 8(a) and (b), we analyze the impact of sensing-induced delay on TCP dynamics at sender side. No PU activity is considered on the CR channel. In Fig. 8(a), we show the behavior of the CW size under three different configurations of t_s, when T_o is fixed to 0.6 s. When sensing is disabled, the CW keeps increasing till the capacity of the channel is reached. When sensing is enabled, TCP–DATA and TCP–ACK packets experience an extra delay which triggers RTO events at TCP sender side. When t_s is equal to 0.5, the sensing delay is comparable with the maximum RTO timer value, and thus frequent RTO events are triggered. In Fig. 8(b), we show the behavior of the RTT in the same scenario, for two different configurations of t_s. As expected, the configuration with sensing enabled (e.g. t_s equal to 0.2 s) experience higher RTT values than the configuration with sensing disabled (e.g. t_s equal to 0 s), due to the fact that a CR user consumes time in sensing the channel periodically.

5.2.2. PU activity analysis

Simulation setup. In the PU activity analysis, we consider the single-hop topology, and we vary the α and β parameters regulating the average PU activity on each channel. The sensing and transmitting time interval are equal to 0.2 s and 0.6 s, respectively. We consider the TCP throughput at application layer, defined in Section 5.2.

Simulation results. Fig. 9(a) shows the throughput of TCP NewReno as a function of the α (x-axis) and β parameters (y-axis). Based on values of α and β, we can distinguish among four different regions of PU activity:

- $\alpha \le 1, \beta \le 1$, (*Long-term activity region*): in each spectrum band, there are long ON periods followed by long OFF periods.
- $\alpha \le 1, \beta > 1$, (*High activity region*): in each spectrum band, there are long ON periods followed by short OFF periods.
- $\alpha > 1, \beta \le 1$, (*Low activity region*): in each spectrum band, there are short ON periods followed by long OFF periods
- $\alpha > 1, \beta > 1$, (*Intermittent activity region*): in each spectrum band, there are short ON periods followed by short OFF periods.

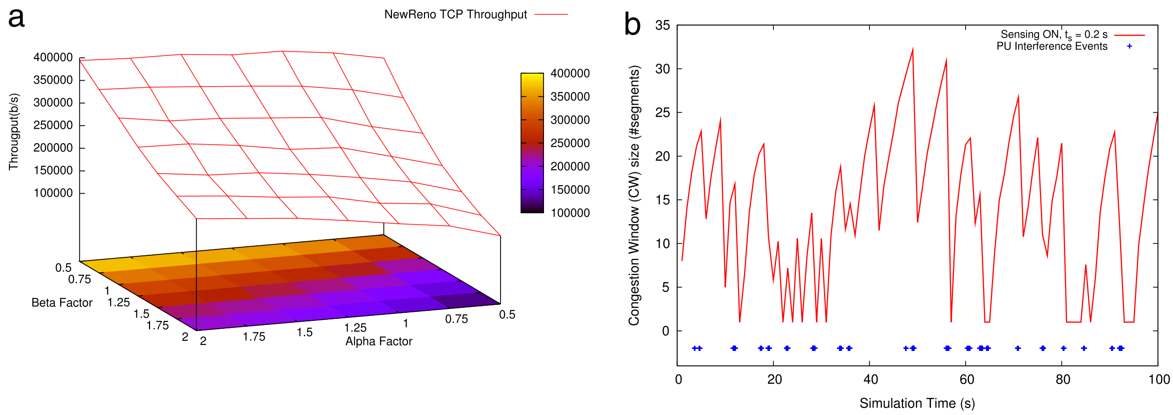

Fig. 9. The throughput of TCP NewReno for different values of PU activity ($\langle\alpha, \beta\rangle$) is shown in Fig. 9(a). The CW size over time and the PU interference events are depicted in Fig. 9(b).

Not surprisingly, TCP throughput is maximized when the CR users have more possibility to access the licensed spectrum without interfering with the PU activity, i.e. in the *low activity region* in Fig. 9(a). When the PU activity is high in each spectrum band, the CR users might need to frequently switch among the available channels, before finding a different vacant portion of the spectrum. The additional delay introduced by the spectrum handoff operations might trigger RTO events and adversely affect the end-to-end performance, for the same reasons described in Section 5.2.1. As a result, TCP experiences the lowest throughput in the *high activity region*. Moreover, Fig. 9(a) shows that TCP performs better when there are long ON periods followed by long OFF periods (i.e. in the *long-term activity region*) rather than when frequent ON/OFF switches occur (i.e. in the *intermittent activity region*), in each primary band. This is due to the fact that when ON/OFF periods are very short a PU might arrive during transmitting period of CR users, even if the channel was found free during sensing. In addition, a long OFF-period allows the CR users to increase the CW significantly which leads to increase in throughput.

Fig. 9(b) shows the CW size over simulation time, for the configuration with $\alpha = 0.5$, $\beta = 0.5$, $t_s = 0.2$ s and $T_o = 0.6$ s. In the same figure, we depict the events of PU interference over CR activity, i.e. the events where a PU starts transmitting on the channel currently used by the CR users. As stated before, PU interference may cause packet losses at both sender and receiver side, so that RTO events are triggered. This is shown in Fig. 9(b) by the fact that CW size is collapsed to one segment in proximity of PU misdetection and interference events.

5.2.3. Channel heterogeneity analysis

Simulation setup. In the channel heterogeneity analysis, we consider the basic single-hop topology, but we remove the assumption that all CR user channels have the same capacity. To evaluate the impact of channel heterogeneity on TCP performance, we model a channel environment where the capacity of channel *i* is defined as follows:

$$C(i) = \begin{cases} \gamma \cdot \left(1 + \dfrac{DF}{2}\right) & \text{if } i\%2 = 0 \\ \gamma \cdot \left(1 - \dfrac{DF}{2}\right) & \text{if } i\%2 = 1 \end{cases} \tag{8}$$

for $0 < i < 5$. From Eq. (8), it is easy to see that a switching operation from channel *i* to channel $i+1$ ($0 < i < 4$) produces an increment/decrement of capacity equal to $\gamma \cdot DF$. In the following, we define *DF* as *channel bandwidth differentiation factor*, because higher values of *DF* correspond to higher difference in raw bandwidth between adjacent channels. We set γ equal to 1 Mb/s. The α and β parameters are both equal to 0.5, while t_s and T_o are equal to 0.025 s and 1.0 s, respectively.

Simulation results. Fig. 10(a) and (b) show the TCP performance for the channel heterogeneity analysis. Fig. 10(a) shows the TCP efficiency as a function of the *DF* factor, for different TCP versions. All TCPs experience the highest efficiency for $DF = 0.2$, but are not able to dynamically adapt to bandwidth variation when *DF* increases. This is in accordance with the results described in [29]. From Fig. 10(a), we can also see that TCP SACK enhances TCP NewReno and TCP Reno, for all the values of *DF*. Fig. 10(b) shows the CW size over simulation time on the *x*–*y* axes for the configuration with $DF = 1.6$, i.e. when large bandwidth variation is experienced by CR users after switching to operations. Moreover, we also show the available bandwidth for CR users over simulation time on the *x*–*y*2 axes. Fig. 10(b) confirms that the CW of classical TCP is unable to correctly track the available bandwidth. This is caused by the additive increase/multiplicative-decrease (AIMD) algorithm, which performs the rate control in TCP. The AIMD algorithm combines linear growth of the CW size during the congestion avoidance phase, with an exponential reduction when a congestion takes place. In the congestion avoidance phase, the CW is increased by one segment every RTT. Such a conservative approach can be inadequate when a brief bandwidth increase is experienced by CR users, because the spectrum opportunity is often lost before the CW has increased to half the segments

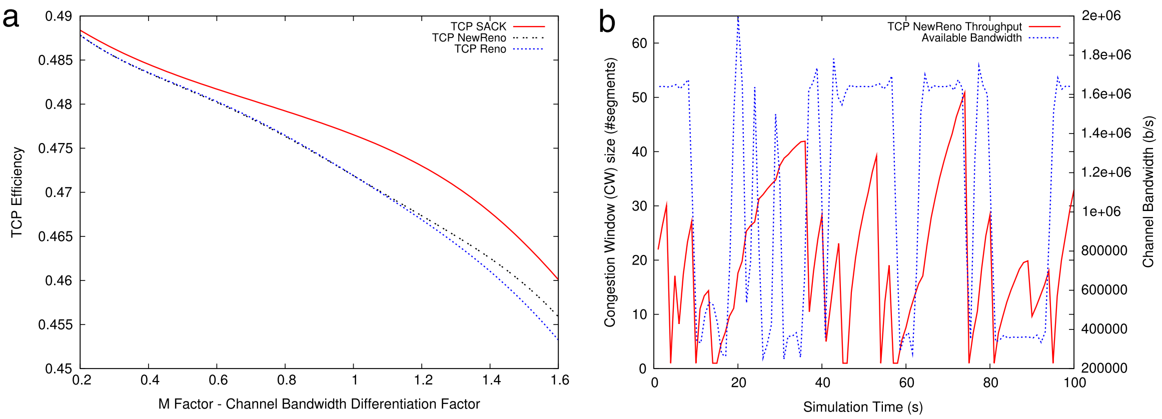

Fig. 10. The TCP efficiency is shown in Fig. 10(a), for different TCP versions. The CW size and the available channel bandwidth over time for TCP NewReno are shown in Fig. 10(b).

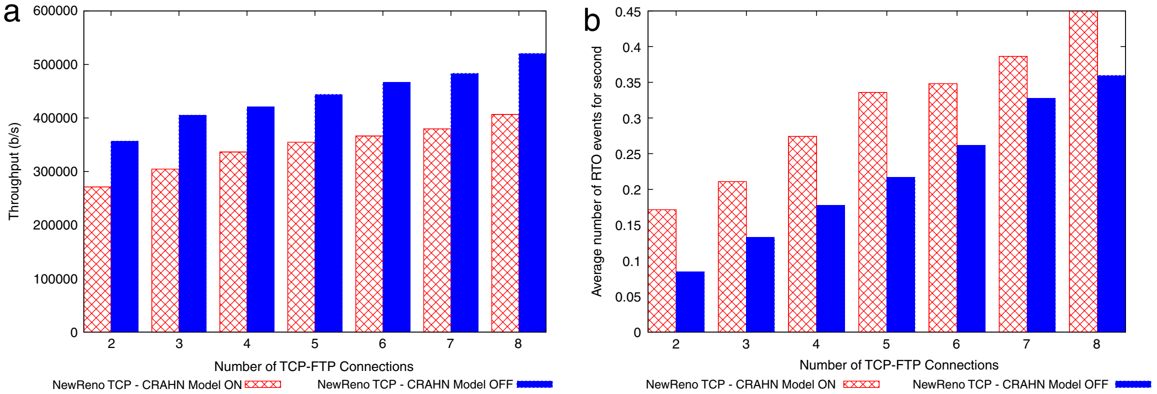

Fig. 11. The throughput of TCP NewReno with CRAHN model ON/OFF is shown in Fig. 11(a), as a function of the number of active connections between the CR users. The average number of RTO events in the network (for second) is shown in Fig. 11(b).

that may be supported on the new channel (Fig. 10(b)). Analogously, the AIMD algorithm does not take into account the events of bandwidth decrease, because it keeps growing the CW till a RTO event occurs. Fig. 10(b) shows that RTO events are triggered in correspondence of bandwidth reduction.

5.2.4. *Multi-hop topology analysis*

In this section, we evaluate the TCP performance over generic random multi-hop topologies, and we show the aggregated results when all the CRAHN characteristics discussed so far are enabled. The goal of the analysis is to highlight how such characteristics might produce different results compared to traditional characteristics of ad hoc networks (e.g. nodes' mobility), and thus to motivate the need for novel transport layer solutions for CRAHNs. We consider random topologies composed of 25 mobile CR users over a simulation area of 1000 × 1000 m^2. All the CR users move with uniform speed v ($2.5\ \text{m/s} < v < 4.0\ \text{m/s}$), and an average pause time of 3 s. We vary the number of active FTP-New Reno TCP connections, from 2 up to 8, with these parameters for the CRAHN environment: $\alpha = 0.5$, $\beta = 0.5$, $t_s = 0.2$ s, $T_o = 1$ s, $DF = 0.2$. In Fig. 11(a) and (b), we evaluate the performance of TCP NewReno, considering two different simulation models:

- *CRAHN model ON*: We use the CRAHN model described in Section 4, with the cognitive parameters stated above. We take into account the impact of sensing time, PU activity and channel bandwidth variation on the end-to-end performance.
- *CRAHN model OFF*: We do not simulate the impact of CRAHN characteristics. At the MAC Layer, we use the traditional 802.11 MAC DCF scheme [39], without the modification described in Section 4. Basically, we evaluate the performance of TCP NewReno in a traditional mobile ad hoc scenario, as addressed in [22].

Fig. 11(a) shows the TCP NewReno throughput as a function of the number of active connections between CR users (on the *x*-axis). As expected, the configuration with *CRAHN model OFF* experiences higher throughput than the configuration with *CRAHN model ON*, and the performance difference increases when more connections are added in the simulation scenario. Fig. 11(b) provides an orthogonal view of system performance, by showing the average number of RTO events generated in the network (for seconds). As expected, RTO events increase when more flows are added into the network, as a consequence of the increasing network congestion. However, additional RTO events are also triggered by link failures and route reconfiguration caused by the nodes' mobility [22]. Some TCP-variants proposed for ad hoc networks exploit network feedback, which allows us to detect packet losses caused by mobility handoff or by network congestion [24]. This approach might not work well over CRAHNs, where additional RTO events might be triggered by the sensing-induced delay, by the handoff delay and by brief variations in channel capacity, as shown by Fig. 11(b).

5.2.5. Considerations

The sensing interval plays a critical role in deciding the optimal end-to-end TCP throughput. A short sensing interval increases the risk of interfering with PU activities, while a long sensing interval increases the PU protection but reduces the transmission opportunities for CR users. The transport layer should balance the tradeoff so that the throughput is maintained at the desired level while the PU interference is minimized. Since the sensing cycle is implemented at the MAC/PHY layers, explicit feedback should be used to inform the transport layer that a channel sensing event is being performed. Notifications of sensing events might be exploited to adapt the flow control mechanism in TCP, by freezing the current TCP state during sensing time intervals, or by reducing the sending rate, as proposed in [28].

When a PU is detected, the CR user should cease the current activity and move to another vacant portion of the spectrum. However, while spectrum sensing is periodic and has a well-defined interval, the time taken to (i) search for a new channel, (ii) coordinate with the next hop neighbor to find a mutually acceptable channel and (iii) notify such information at upper layer is of an uncertain duration, and depends on the parameters describing the PU activity (e.g. $\langle \alpha, \beta \rangle$). Frequent or long spectrum handoff events adversely affect the end-to-end performance.

TCP cannot effectively adapt to brief reductions/increases in channel capacity, due to the drawbacks of the AIMD algorithm. We believe that the CW size in TCP must be scaled appropriately to meet the new channel conditions. To this aim, TCP should be notified of the occurrence of a channel switching event by the link layer. Moreover, link layer metrics should be used to estimate the raw bandwidth of the new channel.

6. Conclusions

In this paper, we have addressed the performance evaluation of end-to-end protocols over cognitive radio ad hoc networks (CRAHNs). To this aim, we have introduced NS2-CRAHN, an extension of the NS-2 simulation tool for the modeling and simulation of CRAHNs. The NS2-CRAHN tool provides accurate modeling of PUs activities and CR users spectrum management functionalities, including spectrum sensing, decision and mobility schemes. Moreover, it provides facilities to support cross-layer information sharing among network protocols at different layers of the protocol stack. By using the NS2-CRAHN, we have evaluated the impact of sensing-induced delay and primary users (PU) interference on the route formation and maintenance process. We have also considered different routing metrics and discovery algorithms. Then, we have studied the impact of specific characteristics of CRAHNs over TCP end-to-end performance, such as the sensing cycle, the interference caused by PUs and the channel heterogeneity. We have evaluated different TCP versions, by varying several network parameters over single-hop and multi-hop CRAHN topologies. Simulation results show that sensing time interval and type of PU activity play a critical role in determining the stability of the route, as well as in deciding the TCP performance. Moreover, simulation results stress that CRAHNs characteristics are not present in traditional wireless ad hoc networks, and that, among others, novel routing metrics and transport protocols should be designed for CRAHNs.

Acknowledgements

This work is partially supported by the University of Bologna and Italian MIUR funds, and by the grant YR2009-7003 from Stiftelsen for *interntionalisering av hogre utbildning och forskning* (STINT).

References

[1] FCC, Spectrum policy task force report, ET Docket No. 02-135, November 2002.
[2] FCC, Spectrum policy task force report—in the matter of unlicensed operation in the TV broadcast bands: second report and order and memorandum opinion and order, ET Docket No. 08-260, November 2008.
[3] I. Akyildiz, W.Y. Lee, K.R. Chowdhury, CRAHNs: Cognitive Radio Ad HoC Networks, Ad Hoc Networks Journal 7 (5) (2009) 810–836.
[4] K.R. Chowdhury, M. Di Felice, Search: a routing protocol for mobile cognitive radio ad-hoc networks, Computer Communication Journal 32 (18) (2009) 1983–1997.
[5] K.R. Chowdhury, I.F. Akyildiz, OFDM based common control channel design for Cognitive Radio Ad HoC Networks, IEEE Transactions on Mobile Computing (2010) (in press).
[6] A. Ghasemi, E.S. Sousa, Spectrum sensing in cognitive radio networks: the cooperation-processing tradeoff, Wireless Communications and Mobile Computing 7 (9) (2007) 1049–1060.
[7] C. Sun, K.B. Letaief, User cooperation in heterogeneous cognitive radio networks with interference reduction, in: Proc. of IEEE ICC 2008, Beijing, China, 2008, pp. 3193–3197.
[8] Network simulator version 2. http://www.isi.edu/nsnam/ns/.
[9] C.E. Perkins, E.M. Belding-Royer, Ad hoc on demand distance vector (AODV) routing, in: Proc. of IEEE WMCSA 1999, New-Orleans, USA, 1999, pp. 90–100.
[10] M.K. Marina, S.R. Das, On-demand multipath distance vector routing in ad hoc networks, in: Proc. of IEEE ICNP 2001, Riverside, USA, 2001, pp. 14–23.

[11] R. Draves, J. Padhyea, Z. Bill, Routing in multi-radio multi-hop wireless mesh networks, in: Proc. of ACM MOBICOM 2004, Philadelphia, USA, 2004, pp. 4–12.

[12] D. De Couto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, in: Proc. of ACM MOBICOM 2003, San Diego, USA, 2003, pp. 134–146.

[13] S. Floyd, T. Henderson, The NewReno modification to TCP's fast recovery algorithm, Internet Engineering Task Force, Request for Comments, Experimental 2582, April 1999.

[14] M. Mathis, J. Mahadavi, S. Floyd, A. Romanow, TCP selective acknowledgment options, Internet Engineering Task Force, Request for Comments, Proposed Standard, 2018, October 2006.

[15] L. Brakmo, L. Peterson, TCP vegas: end to end congestion avoidance on a global Internet, IEEE Journal on Selected Areas in Communications 13 (8) (1995) 1465–1480.

[16] Q. Wang, H. Zheng, Route and specrum selection in dynamic spectrum networks, in: Proc. of IEEE CCNC 2006, Las Vegas, USA, 2006, pp. 625–629.

[17] A.C. Talay, D.T. Altilar, RACON: a routing protocol for mobile cognitive radio networks, in: Proc. of ACM CORONET 2009, Bejing, China, 2009, pp. 73–78.

[18] G. Lei, W. Wang, T. Peng, W. Wang, Routing metrics in cognitive radio networks, in: Proc. of IEEE ICCSC 2008, Shanghai, China, 2008, pp. 265–269.

[19] G. Chen, W. Liu, Y. Li, W. Cheng, Spectrum aware on-demand routing in coginitive radio networks, in: Proc. of IEEE DySPAN 2007, Dublin, Ireland, 2007, pp. 571–574.

[20] R. Pal, Efficient routing algorithms for multi-channel dynamic spectrum access networks, in: Proc. of IEEE DySPAN 2007, Dublin, Ireland, 2007, pp. 288–291.

[21] I. Pefkianakis, S.H.Y. Wong, S. Lu, SAMER: spectrum aware mesh routing in cognitive radio networks, in: Proc. of IEEE DySPAN 2008, October 2008, pp. 1–5.

[22] G. Holland, N.H. Vaidya, Analysis of TCP performance over mobile ad hoc networks, in: Proc. of ACM MOBICOM 1999, Seattle, USA, 2009, pp. 219–230.

[23] K. Sundaresan, V. Anantharaman, H.-Y. Hsieh, R. Sivakumar, ATP: a reliable transport protocol for ad hoc networks, IEEE Transactions on Mobile Computing 4 (6) (2005) 588–603.

[24] J. Liu, S. Singh, ATCP: TCP for mobile ad hoc networks, IEEE Journal on Selected Areas in Communications 19 (7) (2002) 1300–1315.

[25] S.E. ElRakabawy, A. Klemm, C. Lindemann, TCP with adaptive pacing for multihop wireless networks, in: Proc. of ACM MOBIHOC 2005, Urbana-Champaign, USA, 2005, pp. 288–299.

[26] K. Xu, M. Gerla, L. Qi, Y. Shu, Enhancing TCP fairness in ad hoc wireless networks using neighborhood red, in: Proc. of ACM MOBICOM 2003, San Diego, USA, 2003, pp. 16–28.

[27] M. Di Felice, K.R. Chowdhury, L. Bononi, Modeling and performance evaluation of transmission control protocol over Cognitive Radio Ad HoC Networks, in: Proc. of ACM MSWIM 2009, Tenerife, Spain, 2009, pp. 4–12.

[28] K.R. Chowdhury, M. Di Felice, I. Akyildiz, TP-CRAHN: a transport protocol for mobile cognitive radio ad hoc networks, in: Proc. of IEEE INFOCOM 2009, Rio de Jainero, Brazil, 2009, pp. 2482–2490.

[29] A.M.R. Slingerland, P. Pawelczak, R.V. Prasad, A. Lo, R. Hekmat, Performance of transport control protocol over dynamic spectrum access links, in: Proc. of IEEE DYSPAN 2007, Dublin, Ireland, 2007, pp. 486–495.

[30] M. Gong, S. Mao, S. Midkiff, B. Hart, Medium access control in wireless mesh networks, Wireless Mesh Networking 1 (1) (2007) 147–182.

[31] J. Mo, H.S.W. So, J. Walrand, Comparison of multi-channel MAC protocols, in: Proc. of ACM MSWIM 2005, Montreal, Canada, 2005, pp. 209–218.

[32] S.-L. Wu, C.-Y Lin, Y.-C. Tseng, J.-P. Sheu, A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks, in: Proc. of IEEE ISPAN 2000, Dallas, USA, 2000, pp. 232–237.

[33] K. Pradeep, N.H. Vaidya, Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks, ACM SIGMOBILE Mobile Computing and Communications Review 10 (15) (2006) 31–43.

[34] W.Y. Lee, I. Akyildiz, Optimal spectrum sensing framework for cognitive radio networks, IEEE Transactions on Wireless Communications 7 (10) (2008) 3485–3857.

[35] T. Rappaport, Wireless Communications: Principles and Practice, Prentice Hall, 2001.

[36] V. Shrivastava, S. Rayanchu, J. Yoon, S. Banerjee, 802.11n under the microscope, in: Proc. of the ACM/USENIX IMC 2008, Vouliagmeni, Greece, October 2008.

[37] E.M. Belding-Royer, Routing approaches in mobile ad hoc networks, Mobile Ad Hoc Networking 1 (1) (2004) 275–300.

[38] A. Boukerche, Performance evaluation of routing protocols for ad hoc wireless networks, Monet Journal 9 (4) (2004) 333–342.

[39] Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE Std. 802.11, 2007.

Marco Di Felice received the Laurea (summa cum laude) and Ph.D. degrees in Computer Science from the University of Bologna, Italy, in 2004 and 2008, respectively. In 2007, he was a visiting researcher in the Broadband Wireless Networking Laboratory, Georgia Institute of Technology, Atlanta. In 2009, he was a visiting researcher in the Electrical and Computer Engineering Department at Northeastern University, Boston. Since 2008, he has been a post-doctoral researcher from the University of Bologna. His research interests include: modeling and simulation of wireless systems, cognitive radio technology, distributed resources optimization and multi-hop communication in wireless networks.

Kaushik Roy Chowdhury is assistant professor in the Electrical and Computer Engineering Department at Northeastern University, Boston, MA. He graduated with B.E. in Electronics Engineering with distinction from VJTI, Mumbai University, India, in 2003. He received his M.S. in Computer Science from the University of Cincinnati, OH, in 2006, and Ph.D. from the Georgia Institute of Technology, Atlanta, GA in 2009. His M.S. thesis was given the outstanding thesis award jointly by the ECE and CS departments at the University of Cincinnati. He has also won the BWN researcher of the year award during his Ph.D. in 2007, and the best paper award in the Ad Hoc and Sensor Networks symposium at the IEEE ICC conference in 2009. His expertise and research interests lie in wireless Cognitive Radio Ad HoC Networks, energy harvesting, and multimedia communication over sensors networks. He is a member of the IEEE.

Wooseong Kim received his B.S. degree from the Department of Electrical Engineering, University of Seoul, Korea, in 2000 and his M.S. degree from the Department of Computer Science, Korean Advanced Institute of Science and Technology ICC, Korea in 2004. From January 2004 to September 2008, he was a researcher at the Research and Development Center of LG Electronics. Since September 2008, he has been a Ph.D. student of University of California, Los Angeles. His research interests include cognitive radio in ad hoc network, cross-layer architecture and resource allocation in mobile ad hoc networks.

Andreas Kassler is professor of Computer Science at Karlstad University, Karlstad, Sweden, that he joined in 2005. From 2003 to 2004, he was assistant professor at the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests are in the area of Wireless Meshed Networks, Ad-Hoc Networks, Multimedia Networking, Quality of Service, and P2P systems. He has published over 100 conference and journal papers, and several book chapters. He received the Docent title in Computer Science from Karlstad University in 2006, the Ph.D. degree in Computer Science from Universität Ulm, Germany, in 2002 and an M.S. degree from Universität Augsburg, Germany. Prof. Kassler is a Member of the IEEE.

Luciano Bononi received the Laurea (summa cum laude) and Ph.D. degrees in Computer Science from the University of Bologna, Bologna, Italy, in 1997 and 2002, respectively. In 2000, he was a visiting researcher with the Department of Electrical Engineering, University of California, Los Angeles. In 2001, he joined the University of Bologna as a post-doctoral researcher, and since 2002, he has been an assistant professor with the Department of Computer Science at the same university. He is the author of more than 60 refereed conference and journal publications and five book chapters on topics related to on-chip, mobile and wireless network protocols, standards and architectures. Dr. Bononi serves as an associate editor of six international journals. He has served as the Chair and the Technical Program Committee Member of more than 10 and 100 international conferences and workshops, respectively.