



**HAL**  
open science

## Statistical Admission Control in Multi-Hop Cognitive Radio Networks

Guillaume Artero Gallardo, Gentian Jakllari, Lucile Canourgues, André-Luc Beylot

► **To cite this version:**

Guillaume Artero Gallardo, Gentian Jakllari, Lucile Canourgues, André-Luc Beylot. Statistical Admission Control in Multi-Hop Cognitive Radio Networks. *IEEE/ACM Transactions on Networking*, 2018, 26 (3), pp.1390–1403. 10.1109/TNET.2018.2830122 . hal-03044233

**HAL Id: hal-03044233**

**<https://hal.science/hal-03044233>**

Submitted on 11 Dec 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Statistical Admission Control in Multi-Hop Cognitive Radio Networks

Guillaume Artero Gallardo, Gentian Jakllari<sup>1</sup>, Lucile Canourgues, and André-Luc Beylot

**Abstract**—We address the problem of online admission control in multi-hop, multi-transceiver cognitive radio networks where the channel access is regulated by a bare-bones time-division multiple access protocol and the primary user activity is modeled as an ON/OFF process. We show that the problem of computing the available end-to-end bandwidth—necessary for admission control—is NP-Complete. Rather than working on an approximation algorithm and analyzing its worst-case performance, we relax the problem of online admission control by using a randomized scheduling algorithm and analyzing its average performance. Randomized scheduling is widely used because of its simplicity and efficiency. However, computing the resulting average throughput is challenging and remains an open problem. We solve this problem analytically and use the solution as vehicle for BRAND—a centralized heuristic for computing the average bandwidth available with randomized scheduling between a source destination pair in cognitive radio networks. Driven by practical considerations, we introduce a distributed version of BRAND and prove its correctness. An extensive numerical analysis demonstrates the accuracy of BRAND and its enabling value in performing admission control.

**Index Terms**—Multi-hop cognitive radio networks, statistical admission control, TDMA, randomized scheduling.

## I. INTRODUCTION

**D**RIVEN by unprecedented demand for wireless capacity and increasing evidence that a lot of the licensed spectrum is underutilized [2], policymakers and technologists have joined voices in calling for a shift from an exclusive and static mode of allocating wireless spectrum to one that is more adaptive to user traffic [3], [4]. This shift from the regulating authorities, coupled with the emergence of the cognitive radio network concept [5] as the enabling technology, has ignited tremendous interest in cognitive radio networks capable of intelligently exploiting the wireless spectrum [2]. Nevertheless, considering the architectural changes required, many technical and policy challenges need to be ironed out before this vision can become reality [4]. Towards this, a lot of

effort has been put in developing solutions for cognitive radio networks at the physical, link and the routing layers [6]–[9]. However, we are far from having all the answers.

One question that has not been addressed is that of online admission control in multi-hop, multi-transceiver cognitive radio networks where channel access is regulated by a TDMA MAC. Given a newly arrived traffic session, before admitting it, we would like to know whether there is enough bandwidth for servicing its traffic demand. The question can be answered by first computing the bandwidth *currently* available on the path from the source to the destination node. Computing the end-to-end bandwidth in wireless networks is not a new problem [10], [11]. However, there are two reasons for revisiting it. First, most literature on legacy<sup>1</sup> TDMA wireless networks is focused on establishing worst-case bounds at the expense of practicality. For example, some of the classic works on the subject [10], [11] cannot be deployed in practice. Reference [10] assumes a wireless node is always capable of receiving from a one-hop neighbor while a second one-hop neighbor is transmitting to someone else. Reference [11] proposes a centralized solution as benchmarking tool for quantifying the performance of distributed solutions. Solutions landing themselves to practical implementations are rare [12]. Second, the cognitive radio architecture makes the problem non-trivially different. In its landmark ruling [3], permitting the use of unlicensed devices in the UHF spectrum, FCC required that unlicensed devices - the secondary users - do not interfere with incumbents - the primary users. Thus, the end-to-end bandwidth will depend not only on the interference among peers, as it is the case in legacy wireless networks, but also on a different kind of interference. Primary users, as their prerogative, will access the channel without making any effort, such as following an agreed upon protocol, to avoid interfering with secondary users.

In this work, we carefully consider the different constraints and tradeoffs and address the problem of online admission control in multi-hop cognitive radio networks in four steps. First, we construct the problem of computing the available end-to-end bandwidth in multi-hop cognitive radio networks. This is not straightforward as one has to account for the primary user interference. It may be tempting to think of primary user interference as no different than secondary interference. However, note that secondary users are required to implement mechanisms, such as sensing [4], to avoid interfering with primary user traffic. These mechanisms are

This paper was presented at the Proceedings of ACM PM<sup>2</sup>HW<sup>2</sup>N 2013, Barcelona, Spain [1].

(Corresponding author: Gentian Jakllari.)

G. A. Gallardo was with the École Normale Supérieure de Lyon, 69007 Lyon, France. He is now with the Sysoco Group, 69150 Lyon, France (e-mail: guillaume.artero-gallardo@ens-lyon.fr).

G. Jakllari and A.-L. Beylot are with the ENSEEIHT, University of Toulouse, 31000 Toulouse, France (e-mail: jakllari@enseeiht.fr; beylot@enseeiht.fr).

L. Canourgues is with Rockwell Collins, 94320 Thiais, France (e-mail: lucile.canourgues@rockwellcollins.com).

<sup>1</sup>We refer to non-cognitive architectures as legacy.

inevitably imperfect, leading to false positives and negatives whose impact on the bandwidth needs to be carefully modeled and calculated. Second, we analyze the problem and show that it is NP-Complete. In light of this result, the traditional approach would be to design an approximation scheduling algorithm with a bounded worst-case performance [10], [11]. Instead, we relax the problem of online admission control by using a randomized scheduling algorithm and accepting an *average* end-to-end bandwidth. This is motivated by the observation that a distributed solution to the relaxed version of the problem is more useful to admission control than a centralized solution that in theory could calculate a better and more accurate bandwidth but that cannot be implemented. Third, we introduce the Clique Sliding approach and use it to analytically solve the problem of computing the end-to-end bandwidth with randomized scheduling. Randomized scheduling is widely used in practice because of its simplicity and efficiency [13]. However, computing the end-to-end throughput resulting from its application on a multi-hop path is a challenging and, to the best of our knowledge, open problem. We use this solution as vehicle for BRAND, a centralized algorithm for estimating the available Bandwidth with RANDomized scheduling. Finally, we introduce a distributed implementation of BRAND and using the Clock Condition [14] prove its correctness. Our contributions may be summarized as follows:

- We analyze the sources of interference in cognitive radio networks, including how primary users impact the bandwidth (Section III).
- We formally define the problem of computing the available bandwidth between a source destination pair in a TDMA based multi-hop cognitive radio network and show that it is NP-Complete (Section IV-A).
- We introduce a linear time algorithm that can compute the average end-to-end throughput resulting from using randomized slot scheduling for satisfying a given traffic demand on every link of the path connecting a given source destination pair (Section IV-C).
- We introduce BRAND, a heuristic for estimating the available end-to-end bandwidth. BRAND leverages our new algorithm to compute the average end-to-end throughput for all possible traffic demands. The maximum over all the computed throughput values is returned as the available end-to-end bandwidth (Section IV-B).
- We introduce a distributed version of BRAND, prove its correctness and show that it can be implemented by most routing protocol approaches (Section V).
- We perform an extensive numerical analysis of BRAND in general and the throughput computation algorithm in particular. Our analysis demonstrates the accuracy of our algorithm and BRAND's capability to provide correct information for performing online admission control in cognitive radio networks (Section VII).

## II. RELATED WORK

In the following we present related works focused on cognitive radio networks as well as works in the area of

bandwidth estimation and/or allocation for traditional wireless networks.

### *Cognitive Radio Networks*

A main goal of any cognitive radio architecture is to protect the primary users from interference [9]. At the routing layer, a node is required to adapt its path computations according to the primary user activity. To this end, it can either route around the primary user, thus potentially increasing the path length, or, switch its transmission channel on the affected links [15]. Obviously, both strategies will increase the end-to-end delay. An optimal routing metric for multi-hop cognitive radio networks is proposed in [16]. The authors analytically demonstrate its optimality and accuracy for the cases of mobile and static networks. While the works presented so far are shown to handle well the primary users, none of them addresses the problem of admission control for quality of service. Works closer to the problem considered in our work can be found in [17] and [18] where algorithms for joint routing, link scheduling and spectrum assignment have been studied. Nevertheless, the problem of computing the end-to-end bandwidth of a multi-hop path is not addressed in any of these works.

### *Non-Cognitive (Legacy) Networks*

The problem of QoS in non-cognitive wireless multi-hop architectures, with a single or multiple radios, has been subject to significant research efforts and an exhaustive survey is beyond the scope of this paper. Here we simply summarize a subset of the published results closest to the work presented in this paper. The problem of admission control for QoS routing in multi-hop networks is studied by numerous works, including [19] and the references therein. In [19], it is shown that for a TDMA architecture, the problem of computing the residual end-to-end bandwidth for a multi-hop path is NP-Complete. A greedy heuristic is proposed and incorporated in the AODV routing protocol. However, this heuristic was designed for a single radio, non-cognitive radio architecture and, as we discuss in Section VII, cannot be readily applied to a cognitive radio architecture. In [20], Alicherry *et al.* study the joint routing and channel assignment problem for the case of wireless mesh networks with multiple radios. They propose a constant approximation algorithm to the NP-hard problem of maximizing the overall network throughput subjected to fairness constraints. Similarly, [21] introduces a distributed, online and provably efficient algorithm for joint routing, channel assignment and scheduling in multi-hop multi-radio ad hoc networks. In [10] Kodialam and Nandagopal consider the problem of joint routing and link scheduling and propose a solution which consists of forming a set of constraints and solving a linear programming problem. A set of necessary conditions for a rate vector to be achieved is available in [11]. In [12] the capacity region of multi-radio multi-channel wireless networks has been studied by introducing a multi-dimensional conflict graph characterizing the interference between adjacent (radio, link, channel)-tuples. An admission control scheme is introduced by deriving a

set of local sufficient conditions for flow feasibility in such networks. The proposed solution can identify the feasible flows. However, among the feasible flows it can only provide a positive answer for admission for the ones that satisfy the sufficient conditions; there is no answer for the feasible flows that do not satisfy the sufficient conditions. What is more, in this work as well as in [10] and [11], the focus has been on the offline version of the admission control problem. That is, given a network with no prior allocations, the problem considered is that of computing the maximum rate that can be admitted between a source and a destination. In our work, we focus on the online version of the admission control problem: Given a live network, where capacity is allocated as traffic sessions arrive, the problem we tackle is that of computing the bandwidth available between a source and a destination *at the time* a new traffic session arrives.

### III. MODELING INTERFERENCE IN COGNITIVE RADIO NETWORKS

In the following, we describe in detail how we model a multi-hop, multi-transceiver cognitive radio network.

#### A. Network Model

We model a multi-hop cognitive radio network as a graph  $G = (V, E)$ , where  $V$  is the set of nodes and  $E$  the links. We assume that the network comprises only symmetric links. Every cognitive radio node is equipped with a constant number of half-duplex transceivers, each capable of sensing and transmitting on  $B$  predefined orthogonal wireless channels. The transceivers can support several bit-rates on all the channels. An additional transceiver could be used for control signaling. We assume the channel assignment is performed by a spectrum allocation protocol [22] and focus on estimating the available end-to-end bandwidth once such assignment is completed. The only assumption we make about the frequency assignment algorithm is that only one frequency channel is assigned between a particular pair of neighboring nodes.

#### B. Cognitive Channel Access

As a prerequisite for using licensed spectrum, cognitive radios are not allowed to use the channel when it is in use by the respective licensed user. In literature, this is referred to as the secondary-primary<sup>2</sup> hierarchy, with the primary (licensed) user having strict priority over the secondary user when accessing the channel. In this setting, the key novel challenge when designing a channel access protocol is maximizing the realized capacity of the cognitive radio without adversely affecting the Primary User, despite not knowing the latter's communication pattern [23], [24]. In response to this challenge, several new MAC protocols for cognitive radio networks have been proposed [6]. For all the diversity of the proposed solutions, one thing underlying all the protocols is the need for a sensing module for identifying when the cognitive radio may be interfering with the primary user. In its

<sup>2</sup>We will use the terms primary user/secondary user, primary/secondary and PU/SU interchangeably.

most basic form, this module relies on physically sensing the channel [4] periodically to look for primary user activity. When possible, the physical sensing can be complemented by a database of well-known primary users [25].

Given the functionality of the sensing module, a MAC protocol for cognitive radio networks needs to provide periods of network silence dedicated to sensing for primary user activity. What is more, since the activity of the primary user may be completely unknown, the sensing periods need to be periodic. This means that, at given time intervals, all cognitive radios in the network will stop from generating any traffic, and instead, focus on sensing. This is a requirement that can be easily accommodated by a TDMA protocol. Indeed, most MAC protocols proposed for cognitive radio networks [26], [27], including the IEEE 802.22 MAC, are based on TDMA. Nevertheless, some solutions based on random access have also been proposed [7], [8], [28], [29].

While there is no clear winner yet among the MAC protocols proposed, we believe a deterministic medium access protocol will better serve an architecture where multiple technologies share the same spectrum and synchronization is required for sensing. Therefore, we adopt a system in which a TDMA MAC with frame size  $S$  is implemented on every assigned channel. Every time-slot starts with a sensing period. When a node needs to transmit data to a neighboring node, it can get access to the medium by reserving time-slots on the frequency channel assigned to this particular link. For ease of presentation, we refer to the pair (*channel, timeslot*) simply as, a slot.

#### C. Interference in Cognitive Radio Networks

There are two kinds of interference sources in a cognitive radio network. One, the interference caused by other cognitive radios in the same interference domain, usually referred to as secondary-to-secondary interference. And two, the interference caused by the Primary User. The first is not unlike the interference legacy wireless networks have to cope with: Nodes running the same protocol contend for access to the same channel. The primary user interference, however, is different: As its prerogative, the primary user can choose to access the channel at any given time with the expectation of no interference from any potential secondary users. In the following, we discuss how these two sources of interference are modeled in this work.

1) *Secondary-to-Secondary Interference:* For the secondary-to-secondary interference we use the model usually employed in TDMA systems on half-duplex wireless transceivers. Specifically, when a particular node needs to transmit data it reserves a new time-slot on the channel it has been assigned. For the time-slot to be reserved, it needs to satisfy the following requirements:

- 1) It is not used on this channel by the node itself for transmitting,
- 2) It is not used on this channel by a one-hop neighbor for transmitting,
- 3) It is not used on this channel by a two-hop neighbor for transmitting.

We assume every node knows the slot allocations in its two-hop neighborhood and thus can check the satisfiability of the above constraints.

2) *Quantifying the Primary User Interference*: Once the sensing module identifies a primary user, the cognitive radios should stop all communications. Therefore, it is required that a cognitive radio spends part of the time sensing for primary users and part of the time actually transmitting data. To accommodate this requirement, in our model an amount of time in every slot is dedicated to sensing. The optimal ratio between sensing and channel access depends on several factors, including the Primary User activity, the radio environment, etc, and its computation is orthogonal to our work. We refer the reader to [30] and references therein for an authoritative treatment of the subject. Note that, the correctness of our scheme does not depend on the exact values of sensing and access time.

If during the sensing period a primary is identified, no communication will take place in the access part of the slot. Otherwise, the cognitive radio is free to access the channel.

Note, however, that sensing is not perfect. It can very well happen that, while no primary user is identified during the sensing period, a primary user does become active for the whole or part of the access time. When this happens, the exact consequences on whatever secondary user transmissions going on will vary depending on the location and the power strength of the primary user. We follow a somehow pessimistic assumption: A primary, when active, will interfere destructively with any secondary communication taking place in its range.<sup>3</sup>

If we denote with  $\eta$  the part of the slot access time that will be available to the secondary user, based on the reasoning so far, we have:

$$\eta = \text{P[sensing the channel idle]} \\ \times (\text{Fraction of Access Time Free of PU})$$

Denoting with  $u_l$  the probability of a primary user becoming active on link  $l$  during a particular slot, the *fraction of the slot access time available* to the secondary on link  $l$  can be computed as follows:

$$\eta_l = (1 - u_l)^2 \quad (1)$$

*Proof*: The PU's channel state can be modeled by an alternative ON/OFF process [23], [24], [31]. The lengths of the ON and OFF states related to link  $l$ , respectively  $T_{on}$  and  $T_{off}$  can be assumed to be exponentially distributed as follows:

$$f_{on}(t) = \lambda_{on} e^{-\lambda_{on} t} \\ f_{off}(t) = \lambda_{off} e^{-\lambda_{off} t}$$

with parameters  $\lambda_{on}$  and  $\lambda_{off}$  estimated with maximum likelihood methods. Jiang *et al.* [24], [31] have focused on computing the quantity of PU-SU interference accumulated during the access time for evaluating the impact of SU's communications on PU's Quality of Service and appropriately tuning the access time. Defining  $I_0(T_{access})$  as the expected length of all ON states within time  $T_{access}$  given that  $T_{access}$

<sup>3</sup>We assume that if a secondary can sense a primary user then the particular secondary is in the interference range of the primary user.

begins from the OFF state and  $I_1(T_A)$  the same given that  $T_{access}$  begins from the ON state, they demonstrated that  $I_0(T_{access})$  and  $I_1(T_{access})$  satisfy two renewal equations and derived their closed-form expressions. Regarding the communication link  $l$ , as  $\text{P}[\text{occurrence of the OFF state}] = 1 - u_l$ , we get:

$$I_0^l(T_{access}) = u_l T_{access} - u_l^2 \times \frac{1}{\lambda_{off}} \times \left[ 1 - e^{-T_{access} \lambda_{off} / u_l} \right] \quad (2)$$

and

$$I_1^l(T_{access}) = u_l T_{access} + (1 - u_l)^2 \\ \times \frac{1}{\lambda_{on}} \times \left[ 1 - e^{-T_{access} \lambda_{on} / u_l} \right] \quad (3)$$

Although the sensing may declare the primary channel idle, it is still possible that PU's communications begin just before the sensing period ends. This situation can occur if the computed energy does not exceed the selected threshold. Therefore, we define the quantity of interference relative to a slot access time as:

$$Q_l = \frac{(1 - u_l) \times I_0^l(T_{access}) + u_l \times I_1^l(T_{access})}{T_{access}} \quad (4)$$

where  $T_{access}$  corresponds to the slot access time. Substituting Equations 2 and 3 in Equation 4 leads to  $Q_l = u_l$  and  $\eta_l = (1 - u_l)^2$ . ■

Taking into account the sensing time, the *fraction of slot duration*,  $f_l$ , available for secondary-to-secondary communication is:

$$f_l = \eta_l \times \frac{T_{access}}{T_{sensing} + T_{access}} \quad (5)$$

Eq. (1) quantifies the effect of two things on the capacity for the secondary. First, the interference from the primary, who as the owner of the frequency is bounded by no protocol to try to avoid interference with an ongoing secondary communication. And second, the mechanism put in place, i.e. sensing, for satisfying the requirement of doing no harm to the primary.

#### IV. COMPUTING THE AVAILABLE END-TO-END BANDWIDTH FOR ADMISSION CONTROL

##### A. Problem Definition

Let  $d$ , expressed in bits per second, be the amount of bandwidth demanded by an application. Before admitting to route this demand, we would first like to know whether this demand can be satisfied end-to-end. This question can be answered by simply computing the *currently* available end-to-end bandwidth of the path to the destination.

**Definition 1: The available end-to-end bandwidth of a path** is the maximum amount of data, in bits per second, that can be currently transported over the path.

*Remark 1: Unlike the maximum end-to-end bandwidth, the available bandwidth is time sensitive and depends on the current conditions and allocations in the network. If there is no other ongoing traffic in the network and there is no*

TABLE I  
THE MAIN NOTATIONS USED IN THE PAPER

Notation	Description
PSAT	Slots availability table of a path
$N_H$	Number of hops on the path
$S$	TDMA frame size
$\phi$	Physical bit rate
$u$	Probability of a primary user becoming active
$f$	Fraction of slot available to secondary users
$\mathcal{A}$	Set of available slots
$A$	Cardinality of $\mathcal{A}$
$d$	Flow demand (bits/sec)
$r$	Slots required to satisfy $d$
$a$	Slots actually allocated to satisfy $d$
$p$	Proportion ( $a/A$ ) of slots allocated to satisfy $d$
$d_i$	Data forwarded by node $i$ (bits/sec)
$E$	Available slots set
$C$	Cardinality of $E$

primary user activity, the available bandwidth is equivalent to the maximum path bandwidth.

**Constructing the formal problem definition:** A path is modeled as a directed chain  $n_1 \rightarrow n_2 \cdots \rightarrow n_{N_H+1}$  comprising  $N_H$  hops. For ease of presentation, we denote a link  $n_i \rightarrow n_{i+1}$  as  $l_i$ . The bit-rate of every link is denoted by  $\phi_i$  and, for every link, the maximum TDMA frame size is  $S$  slots. To take into account the effect of self-interference, that is, links on the same path interfering with each other, we use the exponential notation  $(j)$  to specify that the considered quantity is evaluated just before node  $n_j$  on the same path does its allocations. Using this convention, we define  $A_i^{(j)}$  as the number of slots available at node  $n_i$  for communication on the link  $l_i$  just before  $n_j$  does its own allocations.

Let us analyze the network behavior when admitting a new flow with demand  $d$ . The first node on the path,  $n_1$ , converts the flow demand,  $d$ , to the required number of slots,  $r_1$ , to be allocated on the first path link,  $l_1$ . The number of required slots will depend on the demand, the link bit-rate,  $\phi_1$ , the TDMA frame size,  $S$ , as well as the primary user interference (quantified in Section III-C.2, Eq. (5)):

$$r_1 = \left\lceil \frac{d}{\phi_1 \times f_1} \times S \right\rceil$$

Let  $a_i$  denote the number of slots allocated on every hop  $i \in \{1, \dots, N_H\}$  for serving this particular flow. For every hop this number will depend on both the demand and how many slots are actually available for new allocations. Thus, for the first hop we have  $a_1 = \min(r_1, A_1^{(1)})$ . If  $a_1 < r_1$  the demand on the second link will be lower than the original demand,  $d$ . To distinguish the two, we denote the demand on the second link, which depends on the allocation on the first link as,  $d_1$ . Table I lists the notations used in the paper.

Rigorously speaking,  $d_1 = \min\left(d, a_1 \times \left(\frac{\phi_1 \times f_1}{S}\right)\right)$ , where the quantity by which  $a_1$  is multiplied is the capacity of a single slot on the first link. We can generalize these results for any hop,  $i > 1$ , as follows:

$$r_i = \left\lceil \frac{d_{i-1}}{\phi_i \times f_i} \times S \right\rceil \quad (6)$$

$$a_i = \min(r_i, A_i^{(i)}) \quad (7)$$

$$d_i = \min\left(d_{i-1}, a_i \times \left(\frac{\phi_i \times f_i}{S}\right)\right) \quad (8)$$

Thus, for a specific demand  $d$ , the **realized end-to-end throughput** is  $\min(d_1, d_2, \dots, d_{N_H}) = d_{N_H}$ , since  $d_i \geq d_{i+1}$ . This analysis gives us a way for tackling the main problem: computing the available end-to-end bandwidth.

*Problem 1: Computing the available end-to-end bandwidth of a path is equivalent to solving the following optimization problem:*

$$\max_{d \in I_d} d_{N_H}(d) \quad (9)$$

where  $I_d = [0, \min(\phi_1, \phi_2, \dots, \phi_{N_H})]$ .

The optimization problem thus defined leads to two observations:

- 1) The realized end-to-end throughput,  $d_{N_H}(d)$ , given a demand,  $d$ , obviously depends on  $d$ .
- 2)  $d_{N_H}(d)$  depends on how the slots are allocated on every hop.

*Theorem 1: Computing the available end-to-end bandwidth of a path in a TDMA-based multi-hop cognitive radio networks with multiple transceivers is NP-complete.*

*Proof:* The proof is straightforward so we provide a sketch. We show that our problem is NP-Complete by reducing the problem of computing the maximum path bandwidth in a single-channel TDMA-based multi-hop network, therein referred to as  $P_2$ , to our problem, therein referred to as  $P_1$ . To this end, we consider the instance of  $P_1$  where a same channel with a constant data rate is assigned on every link along the path and the probability of primary activity on all links is zero. Solving  $P_2$  actually consists of solving one instance of  $P_1$ . Since  $P_2$  has been shown to be NP-complete [19], that concludes the proof. ■

## B. BRAND: An Approach for Estimating the Available End-to-End Bandwidth of a Path

With the problem of computing the available end-to-end bandwidth being NP-Complete, the overwhelming approach in literature has been to design a scheduling heuristic. We follow a different approach: We select a specific slot-scheduling algorithm and focus on computing the available end-to-end bandwidth resulting from applying this particular algorithm. As scheduling algorithm we select the randomized scheduling [13]: when a node needs to assign a certain number of slots, it will select at random among those available. Randomized scheduling is widely used because of its simplicity and efficiency even though the worst case performance can be poor.

Algorithm 1 provides a high-level description of BRAND, our solution for estimating the available end-to-end Bandwidth with RANDom scheduling. It works as follows: For every possible demand  $d$ , the necessary slots are allocated at random among those available on every link and the resulting end-to-end throughput is computed (Section IV-C). By Eq. (9), the available end-to-end bandwidth is simply the maximum end-to-end throughput realized over all possible demands  $d$ .

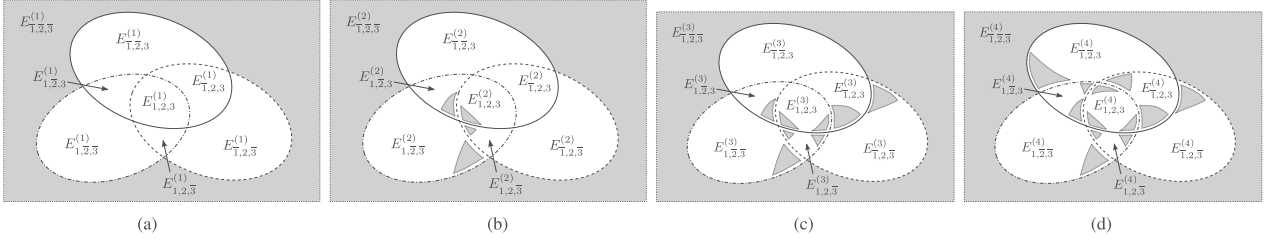


Fig. 1. On a 3-hop path, a given slot can belong to one of the 8 possible slot sets:  $E_{1,2,3}$  (set of slots available for allocation on all 3 links),  $E_{\overline{1,2,3}}$  (set of slots available to none of the links),  $E_{1,2,\overline{3}}$ ,  $E_{1,\overline{2},3}$ ,  $E_{\overline{1},2,3}$ ,  $E_{1,2,\overline{3}}$ ,  $E_{1,\overline{2},3}$ ,  $E_{\overline{1},2,3}$ . When an allocation takes place on  $l_1$ , the proportion of the slots allocated,  $p_1$ , (area painted in gray) is transferred from the sets  $E_{1,2,\overline{3}}$ ,  $E_{1,2,3}$ ,  $E_{1,\overline{2},3}$ ,  $E_{\overline{1},2,3}$  to  $E_{\overline{1,2,3}}$  to account for the interference. The same is repeated for the allocations on  $l_2$  and  $l_3$ . The challenge is estimating the size of the “gray” (or, equivalently, “white”) areas in every set after each allocation. (a) Initial state of the available slots partition. (b) After allocation on the first link. (c) After allocation on the second link. (d) After allocation on the third link.

---

**Algorithm 1 BRAND** (Available Bandwidth With Random Scheduling)

---

**Output** : The available end-to-end bandwidth  
1: **begin**  
2:   **for** every possible demand  $d$  **do**  
3:     //Use randomized scheduling.  
4:      $avThput \leftarrow$  Compute-averageThput ( $d$ );  
5:     **if**  $avThput > AvailBW$  **then**  
6:        $AvailBW \leftarrow avThput$ ;  
7:   **Return**  $AvailBW$ ;

---

The value of the demand,  $d$ , is upper-bounded by the lowest radio bit-rate in the network and lower-bounded by 0. Since this is a slotted system, the values of  $d$  have to be multiples of the smallest slot capacity in the network. Thus, the possible values of  $d$  that need to be considered are bounded by a constant. The non-trivial step of BRAND, line 3 in Algorithm 1, is computing the average end-to-end throughput when the required slots for satisfying a particular demand,  $d$ , are assigned at random. Note that, because the slots are allocated at random, we can only compute the average and not the exact value of the resulting end-to-end throughput. In Section IV-C, we give a centralized approach that, given a demand  $d$ , computes the average end-to-end throughput realized. In Section V we propose a distributed approach.

*C. Computing the Average End-to-End Throughput With Random Scheduling*

We propose an analytical framework for computing the average end-to-end throughput that could be realized if a new demand  $d$  is admitted and randomized scheduling is used on every link. This calculation constitutes line 3 in Algorithm 1.

**(1) The Case of a 3-hop Path**

At first we solve the problem of computing the average throughput on a 3-hop path. This helps introduce our approach along with the notations we use (see Table I) before we show how it is extended to the general case of a  $n$ -hop path.

Let PSAT (path’s slots availability table) be a 2-dimensional array of  $B \times S$  columns and  $N_H$  lines showing the available slots across the  $N_H$  hops constituting the particular path and the  $B$  available frequencies at any given time – the total path capacity.

• **3-hop available slots partition – not all slots are created equal:** Let  $\mathcal{A}_1$  be the set of slots (represented by their respective indices  $\{1, \dots, S.B\}$  in the PSAT table) available for reservation on link  $l_1$  at the beginning of the computation process. These slots are not all equal – some are available on  $l_1$ ,  $l_2$  and  $l_3$ , some only on  $l_1$ , some on  $l_1$  and  $l_2$  but not  $l_3$ , and some are available on  $l_1$  and  $l_3$  but not  $l_2$ . Allocating slots available on all three links will reduce the slots available on  $l_2$  and  $l_3$  and by extension the end-to-end throughput; allocating slots available only on  $l_1$  will have no bearing on  $l_2$  and  $l_3$ . Clearly, for the same number of slots allocated on  $l_1$ , depending on which specific slots are actually allocated, the resulting end-to-end throughput can be different. Therefore, to compute the end-to-end throughput, we divide the available slots into sets of slots whose allocation impacts the bandwidth the same way – the *3-hop available slots partition*. Fig. 1(a) illustrates the concept.

• **Computing the average throughput of a 3-hop path given a demand  $d$ :** Fig. 1 gives a high level picture of how the computation process works. Fig. 1(a) – we carry a 3-hop available slots partition that produces 8 different sets:  $E_{1,2,3}^{(1)}$  (set of slots available for allocation on all 3 links),  $E_{\overline{1,2,3}}^{(1)}$  (set of slots available to none of the links),  $E_{1,2,\overline{3}}^{(1)}$ ,  $E_{1,\overline{2},3}^{(1)}$ ,  $E_{\overline{1},2,3}^{(1)}$ ,  $E_{1,2,\overline{3}}^{(1)}$ ,  $E_{1,\overline{2},3}^{(1)}$ ,  $E_{\overline{1},2,3}^{(1)}$ . The exponent, 1, indicates that the particular membership of each set is valid until node  $n_1$  performs its allocations. Fig. 1(b) – node  $n_1$  tries to allocate enough slots to satisfy the demand  $d$ . As the slots are allocated uniformly at random, it is possible that a portion of slots is allocated (“painted gray”) from each of the 4 possible sets:  $E_{1,2,\overline{3}}^{(1)}$ ,  $E_{1,2,3}^{(1)}$ ,  $E_{1,\overline{2},3}^{(1)}$ ,  $E_{\overline{1},2,3}^{(1)}$ . Fig. 1(c) – node  $n_2$  tries to allocate enough slots to satisfy the demand generated by  $n_1$ . It is conditioned by the initial state of the system as well as the allocations of  $n_1$ . Its own allocations can affect each of the sets  $E_{1,2,\overline{3}}^{(2)}$ ,  $E_{1,2,3}^{(2)}$ ,  $E_{1,\overline{2},3}^{(2)}$ ,  $E_{\overline{1},2,3}^{(2)}$ .

Fig. 1(d) – finally, node  $n_3$  tries to allocate enough slots to satisfy the demand generated by  $n_2$ . It is conditioned by the initial state, the slots allocated by  $n_1$  and those allocate by  $n_2$ . The slots that  $n_3$  is able to allocate define the end-to-end throughput given the demand  $d$  at  $n_1$ . Computing it requires calculating the impact of each allocation on the 8 available slot sets – or, using the illustration, computing the size of the “gray” areas.

Algorithm 2 shows the solution. To make the problem tractable, we relax it by recasting Equations 6-8 (Section IV-A) to use average values while maintaining the same notations. In addition, although mathematically speaking  $E[a_3] \neq \min(E[r_3], E[A_3^{(3)}])$ , we do such an approximation to reduce the calculation complexity. As shown in Section VII, this approximation does not degrade the performance of the overall estimation process.

---

**Algorithm 2** Computing the Average End-to-End Throughput on a 3-Hop Path Using the Available Slots Partition

---

```

input :  $d, S, \phi_1, \phi_2, \phi_3, f_1, f_2, f_3, A_1^{(1)}, A_2^{(1)}, A_3^{(1)}$ 
output :  $a_1, a_2, a_3, d_1, d_2, d_3$ 
1: begin
    //The cardinalities of the 8 sets
    //resulting from the available slots
    //partition.
2:  $C_{1,2,3}^{(1)}, C_{1,2,\bar{3}}^{(1)}, C_{1,\bar{2},3}^{(1)}, C_{1,\bar{2},\bar{3}}^{(1)}, C_{\bar{1},2,3}^{(1)}, C_{\bar{1},2,\bar{3}}^{(1)}, C_{\bar{1},\bar{2},3}^{(1)}, C_{\bar{1},\bar{2},\bar{3}}^{(1)}$ ;
    //Allocation on  $l_1$ 
3:  $r_1 \leftarrow \lceil \frac{d}{\phi_1 \times f_1} \times S \rceil$ ;
4:  $a_1 \leftarrow \min(r_1, A_1^{(1)})$ ;
5:  $d_1 \leftarrow \min(d, a_1 \times \frac{\phi_1 \times f_1}{S})$ ;
6:  $p_1 \leftarrow a_1 / A_1^{(1)}$ ;
7:  $A_2^{(2)} \leftarrow A_2^{(1)} - p_1 \cdot (C_{1,2,3}^{(1)} + C_{1,2,\bar{3}}^{(1)})$ ;
8:  $A_3^{(2)} \leftarrow A_3^{(1)} - p_1 \cdot (C_{1,\bar{2},3}^{(1)} + C_{1,2,3}^{(1)})$ ;
    //Allocation on  $l_2$ 
9:  $r_2 \leftarrow \lceil \frac{d_1}{\phi_2 \times f_2} \times S \rceil$ ;
10:  $a_2 \leftarrow \min(r_2, A_2^{(2)})$ ;
11:  $d_2 \leftarrow \min(d_1, a_2 \times \frac{\phi_2 \times f_2}{S})$ ;
12:  $p_2 \leftarrow a_2 / A_2^{(2)}$ ;
13:  $A_3^{(3)} \leftarrow A_3^{(2)} - p_2 \cdot [C_{1,\bar{2},3}^{(1)} + (1 - p_1) \cdot C_{1,2,3}^{(1)}]$ ;
    //Allocation on  $l_3$ 
14:  $r_3 \leftarrow \lceil \frac{d_2}{\phi_3 \times f_3} \times S \rceil$ ;
15:  $a_3 \leftarrow \min(r_3, A_3^{(3)})$ ;
16:  $d_3 \leftarrow \min(d_2, a_3 \times \frac{\phi_3 \times f_3}{S})$ ;
17: return  $d_3$ ;

```

---

• **Domino Effect:** Algorithm 2 leverages the random nature of the scheduling algorithm to compute the average number of slots impacted in every available slots partition set. For example, a proportion  $a_1/A_1^{(1)}$ , line 7 in Algorithm 2, of the available slots on  $l_1$  is likely to be reserved by  $n_1$ . Therefore, in lines 8 and 9, we can calculate the impact of this allocation on the available slots partition sets and, by extension, the slots left available for allocation on links  $l_2$  and  $l_3$ . This reveals why this problem is fundamentally hard: an allocation on  $l_1$  can impact the allocations on all the following links, creating what we refer to as a domino effect.

While quantifying the domino effect on all allocations is manageable for 3 hops, it becomes intractable for the general case of a  $n$ -hop path. For  $k \in \{0, 1, \dots, n\}$ , the number of sets containing the slots available to  $k$  hops but not the other  $n - k$  equals  $\binom{n}{k}$ . Thus, the number of sets resulting from the  $n$ -hop available slots partition would be:  $\sum_{k=0}^n \binom{n}{k} = 2^n$ . In the following, we present the Interference-Clique sliding approach, capable of breaking the domino effect, thus, making

the problem of computing the average end-to-end throughput tractable.

### (2) The General Case of a $n$ -Hop Path

The interference-clique sliding approach breaks the domino effect by processing the end-to-end bandwidth estimation clique by clique while using only a constant number of variables. We define an interference-clique, or simply clique,<sup>4</sup> as any set of three consecutive links on the path. For instance, a 4-hop path comprises two cliques:  $c_1 = \{l_1, l_2, l_3\}$  and  $c_2 = \{l_2, l_3, l_4\}$ .

When processing a clique  $i$ , the goal is to calculate  $a_i$ , the number of slots that will be allocated on link  $l_i$  given a demand  $d$  at the source. As shown by Eq.(7) in Section IV-A,  $a_i$  depends on the demand arriving from link  $l_1$  and the slots available when the allocation takes place,  $A_i^{(i)}$ . Given the 3-hop available slots partition of clique  $i$ , we have:

$$A_i^{(i)} = C_{i,\bar{i}+1,\bar{i}+2}^{(i)} + C_{i,i+1,\bar{i}+2}^{(i)} + C_{i,\bar{i}+1,i+2}^{(i)} + C_{i,i+1,i+2}^{(i)} \quad (10)$$

The challenge in solving Eq.(10) is computing the cardinalities of the available slots set based on the initial state of the system, i.e. before any allocations for serving the demand  $d$  taking place. Generally speaking, when processing the  $i^{\text{th}}$  clique, the impact of the allocations on the two previous links can be correctly estimated by updating the cardinalities of the sets resulting from its 3-hop available slots partition as follows:

$$\mathbf{C}_i^{(i)} = \mathbf{C}_i^{(1)} - \mathbf{p}_i \times \mathbf{I}_i + \mathbf{u}_i + \mathbf{v}_i \quad (11)$$

where

$$\mathbf{C}_i^{(j)} = (C_{i,\bar{i}+1,\bar{i}+2}^{(j)} \quad C_{i,i+1,\bar{i}+2}^{(j)} \quad \dots \quad C_{i,i+1,i+2}^{(j)})$$

$$\mathbf{p}_i = (p_{i-2} \quad p_{i-1} \quad [p_{i-2} + p_{i-1} \cdot (1 - p_{i-2})])$$

and

$$\mathbf{I}_i = \begin{pmatrix} C_{i-2,\bar{i}-1,\bar{i},\bar{i}+1,\bar{i}+2}^{(i-2)} & \dots & C_{i-2,\bar{i}-1,i,i+1,i+2}^{(i-2)} \\ C_{i-2,i-1,\bar{i},\bar{i}+1,\bar{i}+2}^{(i-2)} & \dots & C_{i-2,i-1,i,i+1,i+2}^{(i-2)} \\ C_{i-2,i-1,\bar{i},\bar{i}+1,\bar{i}+2}^{(i-2)} & \dots & C_{i-2,i-1,i,i+1,i+2}^{(i-2)} \end{pmatrix}$$

The vector  $\mathbf{u}_i$  serves to compensate a set that is not affected by allocations on all the links. The vector  $\mathbf{v}_i$  is then used to update the sets receiving slots that, due to the allocation process, become unavailable on other links. The values of vectors  $\mathbf{u}_i$  and  $\mathbf{v}_i$  depend on some variables used in  $\mathbf{p}_i$  and  $\mathbf{I}_i$  and are given in the Appendix of the companion technical report [32].

In the following we solve Eq.(11) for all the possible cases.

• **Interference-Clique 1:** Clique 1 is the easiest to process as it does not depend on any previous allocations. Exactly  $a_1 = \min(r_1, A_1^{(1)})$  slots are reserved for communication on link  $l_1$ . The remaining available slots on  $l_1$  are not considered anymore and the calculation is passed on to clique 2.

• **Interference-Clique 2:** To measure correctly the impact on clique 2 by the reservations on  $l_1$ , the 3-hop available slots

<sup>4</sup>We use the terms clique and interference-clique interchangeably.



partition related to clique 2 is extended to the 4-hop available slots partition so as to include  $l_1$ . Thus, we have:

$$C_{2,3,4}^{(1)} = \underbrace{C_{1,2,3,4}^{(1)}}_{\text{impacted by allocations on } l_1} + \underbrace{C_{1,2,3,4}^{(1)}}_{\text{not impacted}} \quad (12)$$

Therefore, an average proportion  $p_1 = a_1/A_1^{(1)}$  of slots in  $E_{1,2,3,4}^{(1)}$  is likely to become unavailable for reservation on link  $l_2$  after node  $n_1$  performs its allocations.

Assuming there is no interference between  $l_1$  and  $l_4$ , the clique 2 sets can be updated as follows:

$$\begin{aligned} C_{2,3,4}^{(2)} &= C_{2,3,4}^{(1)} - p_1 \cdot C_{1,2,3,4}^{(1)} \\ C_{2,3,4}^{(2)} &= C_{2,3,4}^{(1)} - p_1 \cdot C_{1,2,3,4}^{(1)} \\ C_{2,3,4}^{(2)} &= C_{2,3,4}^{(1)} - p_1 \cdot C_{1,2,3,4}^{(1)} \\ C_{2,3,4}^{(2)} &= C_{2,3,4}^{(1)} - p_1 \cdot C_{1,2,3,4}^{(1)} \\ C_{2,3,4}^{(2)} &= C_{2,3,4}^{(1)} - p_1 \cdot C_{1,2,3,4}^{(1)} \\ C_{2,3,4}^{(2)} &= C_{2,3,4}^{(1)} - p_1 \cdot C_{1,2,3,4}^{(1)} \\ C_{2,3,4}^{(2)} &= C_{2,3,4}^{(1)} + p_1 \cdot \left( C_{1,2,3,4}^{(1)} + C_{1,2,3,4}^{(1)} + C_{1,2,3,4}^{(1)} \right) \\ C_{2,3,4}^{(2)} &= C_{2,3,4}^{(1)} + p_1 \cdot \left( C_{1,2,3,4}^{(1)} + C_{1,2,3,4}^{(1)} + C_{1,2,3,4}^{(1)} \right) \end{aligned} \quad (13)$$

• **Interference-Clique 3:** This clique is impacted by the allocations on the two previous links,  $l_1$  and  $l_2$ . Furthermore, we need to keep in mind that  $l_3$  is impacted by the allocation on  $l_1$  and  $l_2$  while  $l_4$  only by the allocation on  $l_2$ . Therefore, we have:

$$\begin{aligned} C_{3,4,5}^{(1)} &= \underbrace{C_{1,2,3,4,5}^{(1)}}_{\text{impacted by allocations on } l_1 \text{ and then } l_2} + \underbrace{C_{1,2,3,4,5}^{(1)}}_{\text{impacted by allocations on } l_1 \text{ but not } l_2} \\ &\quad + \underbrace{C_{1,2,3,4,5}^{(1)}}_{\text{impacted by allocations on } l_2 \text{ but not } l_1} + \underbrace{C_{1,2,3,4,5}^{(1)}}_{\text{not impacted}} \end{aligned} \quad (14)$$

which leads to:

$$\begin{aligned} C_{3,4,5}^{(3)} &= C_{3,4,5}^{(1)} - [p_1 + p_2(1 - p_1)] \cdot C_{1,2,3,4,5}^{(1)} \\ &\quad - p_1 \cdot C_{1,2,3,4,5}^{(1)} - p_2 \cdot C_{1,2,3,4,5}^{(1)} \end{aligned} \quad (15)$$

Measuring the impact of allocations on  $l_1$  and  $l_2$  is equivalent to transferring slots from one set to another. Indeed, from the previous equations, we can conclude that, on average,  $p_1 \cdot [C_{1,2,3,4,5}^{(1)} + C_{1,2,3,4,5}^{(1)}]$  slots and  $[p_2(1 - p_1) \cdot C_{1,2,3,4,5}^{(1)} + p_2 \cdot C_{1,2,3,4,5}^{(1)}]$  slots in the set  $E_{3,4,5}^{(1)}$  are allocated by links  $l_1$  and  $l_2$ , respectively. Due to the nature of the interference, when updating the sets resulting from the 3-hop available slots partition related to clique 3, on average  $p_1 \cdot [C_{1,2,3,4,5}^{(1)} + C_{1,2,3,4,5}^{(1)}]$  slots from the set  $E_{3,4,5}^{(1)}$  are transferred to  $E_{3,4,5}^{(1)}$  and  $[p_2(1 - p_1) \cdot C_{1,2,3,4,5}^{(1)} + p_2 \cdot C_{1,2,3,4,5}^{(1)}]$  to the set  $E_{3,4,5}^{(1)}$ .

• **Interference-Clique 4 and beyond:** When processing the third clique, the entries of matrix  $\mathbf{I}_i$  referred strictly to sets that had not changed from the beginning of the estimation

process. However, that is not the case when processing the fourth clique. The corresponding sets are likely to have been impacted by allocations on the previous links. For example,  $E_{2,3,4,5,6}^{(2)}$  is probably impacted by the allocations on  $l_1$  and thus needs to be updated. A straightforward solution would identify the sets resulting from the 6-hop available slots partition and estimate how each set is impacted. This method is correct but the number of resulting sets grows exponentially with the number of links and thus is intractable.

Fortunately, the random nature of the scheduling algorithm can simplify the analysis and bound by a constant the number of variables to deal with at every clique. Say we want to characterize a set used in the clique  $i$ , e.g.  $E_{i-2, i-1, i, i+1, i+2}^{(i-2)}$  as a function of its initial state. We start by doing the 2-hop available slots partition related to  $l_{i-2}$  and  $l_{i-1}$ . This partition leads to four disjoint sets:  $E_{i-2, i-1}^{(j)}$ ,  $E_{i-2, i-1}^{(j)}$ ,  $E_{i-2, i-1}^{(j)}$  and  $E_{i-2, i-1}^{(j)}$ . Let us first work on  $E_{i-2, i-1}^{(j)}$ . This set can also be partitioned into eight subsets resulting from the 3-hop available sets partition of clique  $i$ . In this partition the slot space equals  $E_{i-2, i-1}^{(j)}$  rather than  $\{1, 2, \dots, S.B\}$ , leading to subsets of the following form  $E_{i-2, i-1, i, i+1, i+2}^{(j)}$ . A property of the set  $E_{i-2, i-1}^{(j)}$  is that along the estimation process, it can only transfer slots to the set  $E_{i-2, i-1}^{(j)}$  and cannot receive slots from another. Therefore, the number of slots that initially belonged to the set  $E_{i-2, i-1}^{(j)}$  but became unavailable just

before the allocation on link  $l_{i-2}$  equals  $C_{i-2, i-1}^{(1)} - C_{i-2, i-1}^{(i-2)}$ . These slots became unavailable due to allocations on  $l_{i-4}$  and  $l_{i-3}$ . Because of the random nature of the slot reservation process, these slots were taken uniformly at random among the subsets partitioning  $E_{i-2, i-1}^{(j)}$ . We can thus represent the number of slots that became unavailable in  $E_{i-2, i-1, i, i+1, i+2}^{(j)}$  with the discrete random variable  $X_{i-2, i-1, i, i+1, i+2}$  taking its values in the set  $\{0, \dots, C_{i-2, i-1}^{(1)} - C_{i-2, i-1}^{(i-2)}\}$  and following a hypergeometric distribution with parameters  $(C_{i-2, i-1}^{(1)}, C_{i-2, i-1, i, i+1, i+2}^{(1)}, C_{i-2, i-1}^{(1)} - C_{i-2, i-1}^{(i-2)})$ . From this identification we can deduce that, for  $C_{i-2, i-1}^{(1)}$  strictly positive, the average value of this random variable equals  $[(C_{i-2, i-1}^{(1)} - C_{i-2, i-1}^{(i-2)})/C_{i-2, i-1}^{(1)}] \times C_{i-2, i-1, i, i+1, i+2}^{(1)}$ .

More generally, just before node  $n_{i-2}$  does its allocations,  $C_{i-2, i-1}^{(i-2)}/C_{i-2, i-1}^{(1)}$  of the initially available slots remain available in every subset partitioning  $E_{i-2, i-1}^{(j)}$ . The quantity  $C_{i-2, i-1, i, i+1, i+2}^{(i-2)}$  can be computed as follows:

$$C_{i-2, i-1, i, i+1, i+2}^{(i-2)} = C_{i-2, i-1, i, i+1, i+2}^{(1)} \times \alpha_{i-2, i-1} \quad (16)$$

where the reduction factor of the set  $E_{i-2, i-1}^{(j)}$  equals

$$\alpha_{i-2, i-1} = \begin{cases} 0, & \text{if } C_{i-2, i-1}^{(1)} = 0 \\ \frac{C_{i-2, i-1}^{(i-2)}}{C_{i-2, i-1}^{(1)}} = \frac{C_{i-2, i-1, i, i+1, i+2}^{(i-2)}}{C_{i-2, i-1, i, i+1, i+2}^{(1)} + C_{i-2, i-1, i, i+1, i+2}^{(i-2)}}, & \\ \text{otherwise} \end{cases} \quad (17)$$

and is related to clique  $(i - 2)$  as it can be computed at the beginning of the process of this clique. The same analysis can be applied to the other two sets of interest,  $E_{i-2,i-1}^{(j)}$  and  $E_{i-2,i-1}^{(j)}$ . However, it needs modification for the case of  $E_{i-2,i-1}^{(j)}$  when  $C_{i-2,i-1}^{(1)} = 0$  as this set can receive slots from  $E_{i-2,i-1}^{(j)}$  due to the allocations on the previous links. To correctly update the resulting subsets, we compute the proportion of slots transferred from  $E_{i-2,i-1}^{(j)}$  to  $E_{i-2,i-1}^{(j)}$ . The quantity  $C_{i-2,i-1,i,i+1,i+2}^{(i-2)}$  can be computed as follows:

$$C_{i-2,i-1,i,i+1,i+2}^{(i-2)} = C_{i-2,i-1,i,i+1,i+2}^{(1)} \times \tau_{i-2} \quad (18)$$

with

$$\tau_{i-2} = \frac{C_{i-2,i-1}^{(i-2)}}{C_{i-2,i-1}^{(1)}} = \frac{C_{i-2,i-1,i}^{(i-2)} + C_{i-2,i-1,i}^{(i-2)}}{C_{i-2,i-1,i}^{(1)} + C_{i-2,i-1,i}^{(1)}} \quad (19)$$

if  $C_{i-2,i-1}^{(1)}$  is strictly positive, and zero otherwise.

Note that for all the apparent complexity, the clique-sliding approach does a single pass on the path, with a constant number of computations on every clique. Therefore, the running time is *linear* on the length of the path.

## V. DISTRIBUTED IMPLEMENTATION OF BRAND

The centralized version of BRAND can be easily used with source routing protocols, like DSR [33], where the path computation is centralized at the source. However, this is not the case for non-source routing protocols, like the popular OLSR [34], where the route computation is performed at every node in distributed fashion. To address this limitation we present a simple mechanism that enables the distributed execution of BRAND.

### Algorithm 3 Distributed BRAND

---

```

Input : Bandwidth Request Packet BP
1: begin
2:   if (I'm the source node) then
3:     Initialize (BP);
4:     Transmit (BP, next_hop);
5:     Return;
6:   //Look up the routing table.
7:   next_hop ← get_next_hop(BP.destID);
8:   if (get_hopCount (BP) == 1) then
9:     Transmit (BP, next_hop);
10:  else
11:    Compute_Bandwidth_Clique(BP, avail_BW);
12:    if (next_hop == BP.destID) then
13:      Transmit (avail_BW, BP.sourceID);
14:    else
15:      Update (BP);
16:      Transmit (BP, next_hop);

```

---

BRAND's interference-clique sliding approach (Section IV-C) is what makes it amenable to a distributed computation, for two reasons. The computation at every step requires only knowledge of the interference clique at hand and the three reduction factors,  $\alpha$ , resulting from the computation on the

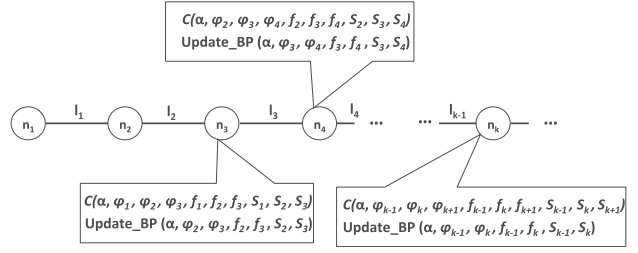


Fig. 2. The BRAND computation is executed clique by clique – first clique being  $l_1, l_2, l_3$ , second  $l_2, l_3, l_4$ , and so forth. Node  $n_3$  knows the source,  $n_1$ , the previous hop,  $n_2$ , and next hop,  $n_4$ , allowing it to bootstrap the BRAND computation.

previous clique. Second, the *sliding* is strictly linear, needing a single pass from the source to the destination.

A straightforward solution would work as follows. The first node on the path would execute the computation for the first clique, comprising links  $l_1, l_2, l_3$ . Once the computation was done, the node would insert the results of the computation as well as the reduction factors in a control packet, call it the bandwidth packet (BP) and would transmit it to the next hop on the path asking it to perform the same for the second clique, comprising links  $l_2, l_3, l_4$ . The process would be repeated till the destination, which would complete the *sliding* and send the results back to the source node. Assuming the BP packet is received by the destination, the result would be identical to centralized BRAND.

The problem with this solution is that, with non-source routing, a particular node does not know which direction a given packet is going beyond the next hop. That means the source node only knows the next hop for a given path, when doing the computation for the first clique of that path requires knowledge of the next three hops. To overcome this challenge, we propose a simple trick: Simply shift the interference-clique calculation by two hops down the path. As illustrated in Figure 2, the BRAND computation is bootstrapped by node  $n_3$ , which regardless of the routing protocol will know  $n_1$ , the source,  $n_2$ , the previous hop and  $n_4$ , the next hop. Equipped with the information necessary,<sup>5</sup>  $n_3$ , performs the BRAND computation for the first clique. Once the computation is done, it sends the BP control packet to  $n_4$  and the rest proceeds as in the straightforward solution. Algorithm 3 gives the specification of the distributed algorithm in pseudo-code for the case of non-trivial (longer than two hops) paths.

Finally, in the following we show the correctness of distributed BRAND.

*Theorem 2: Assuming the bandwidth packet (BP) is eventually received by all nodes along the path and the communications follow the protocol model [35], distributed BRAND results in the same computation as centralized BRAND.*

*Proof:* For distributed BRAND to give the same result as centralized BRAND it suffices to show that clique computations take place in order. That is, computation in clique  $i - 1$  always happens-before computation in clique  $i$ . That is equiva-

<sup>5</sup>A simple heart-beat protocol can give every node the quantities  $\phi, f, p, A$  for all links in the two-hop neighborhood.

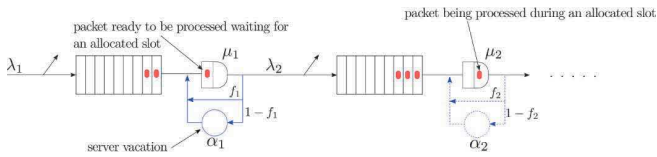


Fig. 3. The model for estimating the end-to-end delay.

lent to showing that if  $C(E_k)$  is the time at which Algorithm 3 is executed at node  $n_k$  along the path,  $C(E_{k-1}) < C(E_k) \forall k$  [14]. This is straightforward given that Algorithm 3 is executed only upon reception of a BP packet, and based on the protocol model assumption a BP packet for a given flow on a given path can be received by node  $k$  only once transmitted by node  $k - 1$ . ■

## VI. R-BRAND: ADMISSION CONTROL FOR TIME-SENSITIVE APPLICATIONS

In this section, we show how to modify BRAND so as to provide admission control to time-sensitive traffic flows. At the high level, R-BRAND, the real-time tailored version of BRAND, works by repeating the following four steps as long as there are flows waiting admission:

- 1) Compute the available end-to-end bandwidth with BRAND,
- 2) Determine the amount of RTP flows that can be admitted end-to-end, given the bandwidth required by every single flow,
- 3) Compute the *average*<sup>6</sup> end-to-end delay for that set of newly admitted flows,
- 4) Decide whether admission can be granted given the end-to-end QoS constraints.

To estimate<sup>7</sup> the average end-to-end delay, we adopt a queuing system approach and model a path as a set of  $N_h$  queues with finite buffer sizes disposed in tandem, as illustrated in Fig. 3. A service corresponds to a transmission occurring over a link  $i$  at rate  $\mu_i$  packets per second. Given that  $a_i$  slots would be allocated on the  $i$ -th link (this quantity is evaluated during the execution of step 1), the probability of server  $i$  being active at an arbitrary time is  $f_i = r_i/S$ . The  $i$ -th queue blocking probability,  $B_i$ , depends on the values of  $\mu_i$  as well the traffic intensity entering the queue,  $\lambda_i$ . Given its value, the traffic intensity entering the following queue equals  $\lambda_{i+1} = (1 - B_i)\lambda_i$ . Denoting with  $Q_i$  and  $R_i$  the average number of packets stored in queue  $i$  and the average delay of packets traversing that queue, respectively, the Little's law gives:  $R_i = Q_i/((1 - B_i)\lambda_i)$ ,

We compute R, the average end-to-end delay, as the sum of the values  $R_i$  for all the queues composing the path. Due to space constraints, however, the rest of the analysis can be found in the Appendix C of the companion technical report [32].

<sup>6</sup>Tail delay would be a better metric for admission control of real-time traffic. However, considering the difficulty of computing it, we relax the problem and work with average delay.

<sup>7</sup>We use the notations from Table I.

## VII. PERFORMANCE EVALUATION OF BRAND

In this section, we evaluate BRAND numerically using MATLAB and compare it with the work of Zhu and Corson [19] and Kodialam and Nandagopal [11].

### A. Simulation Parameters

Nodes implement a TDMA MAC with a 50 ms frame comprising 40 time slots. The frame is divided into a control part, for slot scheduling, and data transmission part. During the control period, a node broadcasts beacons containing information regarding its own allocations and that of its one-hop neighbors, allowing all nodes to eventually acquire two-hop information. 20% of every data transmission slot is dedicated to sensing and the rest to data transmissions. As the specific spectrum assignment process is beyond the scope of this paper, we simply use the following probabilistic model to select the assigned channel for each communication link:  $P[\text{channel1}] = 0.80$ ,  $P[\text{channel2}] = 0.10$ ,  $P[\text{channel3}] = 0.05$  and  $P[\text{channel4}] = 0.05$ .

In practice, link rates depend on the local environment and may fluctuate with time. To approximate this behavior, we sample every link rate  $\phi_i$  according to a normal distribution with mean  $\mu_\phi$  and standard deviation  $\sigma_\phi$  where  $\mu_\phi$  is the mean link transmission rate on the corresponding assigned channel and  $\sigma_\phi$  is taken proportional to  $\mu_\phi$ . In all the simulation results presented here  $\sigma_\phi = \mu_\phi \times 0.10$ . For the channels considered, we choose:  $\mu_\phi(\text{channel1}) = 2\text{Mbps}$ ,  $\mu_\phi(\text{channel2}) = 1.5\text{Mbps}$ ,  $\mu_\phi(\text{channel3}) = 800\text{kbps}$  and  $\mu_\phi(\text{channel4}) = 250\text{kbps}$ . According to the corresponding cumulative distribution function, the generated transmission rates for  $\mu_\phi = 2000\text{kbps}$  and  $\sigma_\phi = 400\text{kbps}$  oscillate between 1000kbps and 3000kbps.

*BRAND Parameters:* BRAND calculates the available end-to-end bandwidth by computing the average end-to-end throughput realized over all possible demands and returning the highest value. In this evaluation, the demand values are taken in the range  $I_d = [0, \min(\phi_1, \phi_2, \dots, \phi_{N_H})]$  (kbps) with step  $\Delta_\phi = 10\text{kbps}$ .

*Basis for Comparison:* To the best of our knowledge, there is no other work that tackles the problem of computing the available end-to-end bandwidth for a cognitive radio network. The closest to our work is that by Zhu and Corson [19], which computes the available bandwidth for legacy TDMA multi-hop network. Furthermore, we use the algorithm proposed by Kodialam and Nandagopal [11] and the optimal solution obtained by solving the integer linear program with *lpsolve* [36] as benchmarking tools.

### B. End-to-End Bandwidth With BRAND

We perform the following experiment for comparing BRAND to KODIALAM and the optimal. We choose the values of no secondary interference,  $p_a$ , for every link uniformly at random in  $(0,1)$  and generate PSAT tables for various path sizes. We run BRAND and KODIALAM in MATLAB using these PSAT tables and compute the available end-to-end bandwidth for different values of primary

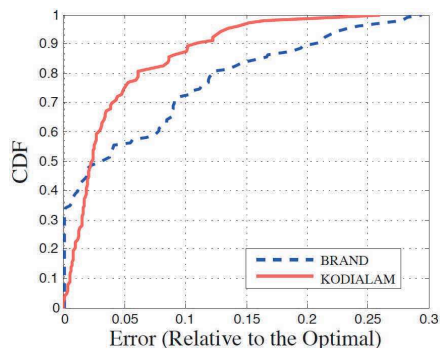


Fig. 4. Available end-to-end bandwidth values computed by BRAND and KODIALAM as function of the optimal values, obtained by using *lpsolve* to solve the integer linear program.

user interference. As computing the optimal slot scheduling is NP-Complete, we formulate the problem as an integer linear program, see the Appendix of the companion technical report [32], and use *lpsolve* [36] to generate optimal scheduling assignments for a given PSAT table of a given path. Figure 4 shows that BRAND is very competitive when compared to KODIALAM, an algorithm proposed as benchmarking tool by its inventors mainly because it cannot be computed in distributed fashion, as well as the optimal scheduler.

### C. End-to-End Throughput With Randomized Scheduling

The most novel and challenging part of BRAND is its algorithm for computing the average throughput of randomized scheduling, introduced in Section IV-C. Given the involved analysis of the algorithm, here we perform a simple experiment for verifying its correctness. For a specific value of hop-count and probability of no secondary interference,  $p_a$ , we generate a PSAT table using MATLAB. The algorithm is applied using the PSAT table and the average end-to-end throughput is computed for all demands described in Section VII-A. With the same PSAT and traffic demands, we run simulations in MATLAB in which the slots are selected at random on every hop. The simulation is run multiple times and the seed for the random generator is changed every time. The throughput values measured at the end of each simulation are averaged over all runs. In Figure 5, for every demand, the computed and the measured throughput values for  $p_a = 33\%$  and  $p_a = 50\%$  are plotted on the  $x$  and  $y$ -axis, respectively. The probability of PU interference is set to 10%. The data shown are for a 10-hop path, a very large value. Even for this challenging scenario, after 5 runs the measurements are already converging to the computed values.

### D. Admission Control Performance

BRAND is designed for enabling admission control. As such, we expect any flow with demand less than or equal to the available bandwidth computed by BRAND to be admitted end-to-end. To verify that this is the case, we perform the following two-step experiment. In the first step, we apply BRAND on a 4-hop path and compute the available end-to-end

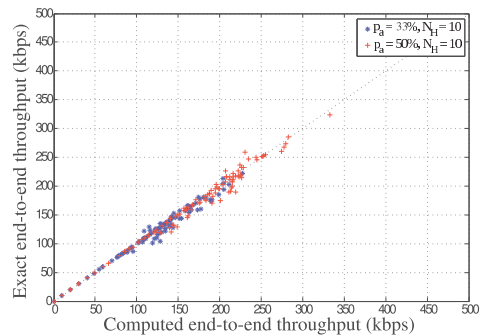


Fig. 5. Numerical verification of the correctness of the algorithm for computing the average throughput introduced in Section IV-C. The  $y$ -axis represents the values obtained numerically, and for these graphs are averages over 5 runs. The probability of PU interference,  $u$ , is set to 10%.

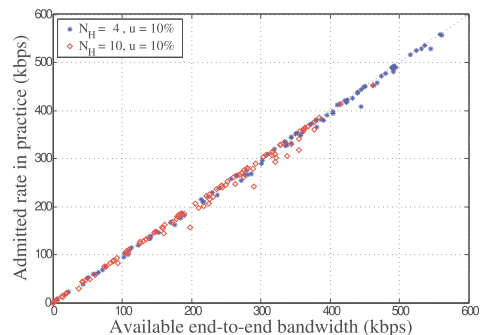


Fig. 6. The  $x$ -axis represents the available end-to-end bandwidth computed by BRAND for various probabilities of no secondary interference. The  $y$ -axis represents the end-to-end throughput measured in simulations – utilizing the same PSAT as the computation – when the traffic demand is equal to the respective end-to-end bandwidth computed by BRAND. The data shown here is for paths of four ( $N_H = 4$ ) and ten hops ( $N_H = 10$ ). Nearly 100% admission is achieved.

bandwidth for various values of  $p_a$ . Specifically, we perform the computation on PSAT tables generated by taking the probabilities of no secondary interference,  $p_a$ , in  $(0, 1)$  while the primary user interference is set to 10%. One PSAT table is generated per value of  $p_a$  and one bandwidth value per PSAT is computed by BRAND.

The available bandwidth values computed by BRAND in the first step are used as input in the second step of the experiment. Specifically, for every value of  $p_a$  and respective PSAT used in the first step, we run a simulation during which a single session with traffic demand equal to the available bandwidth computed by BRAND for this value of  $p_a$  is initiated end-to-end. We measure the end-to-end throughput realized during the simulation and plot it as function of the computed demand. To provide more data about the behavior of BRAND we repeat the same experiment for a 10-hop path, as long a path as one can expect to encounter in deployed multi-hop cognitive radio networks. As shown in Figure 6, the measured throughput is practically identical to the computed values of the available bandwidth. This demonstrates that BRAND delivers the bandwidth it promises and provides nearly 100% admission.

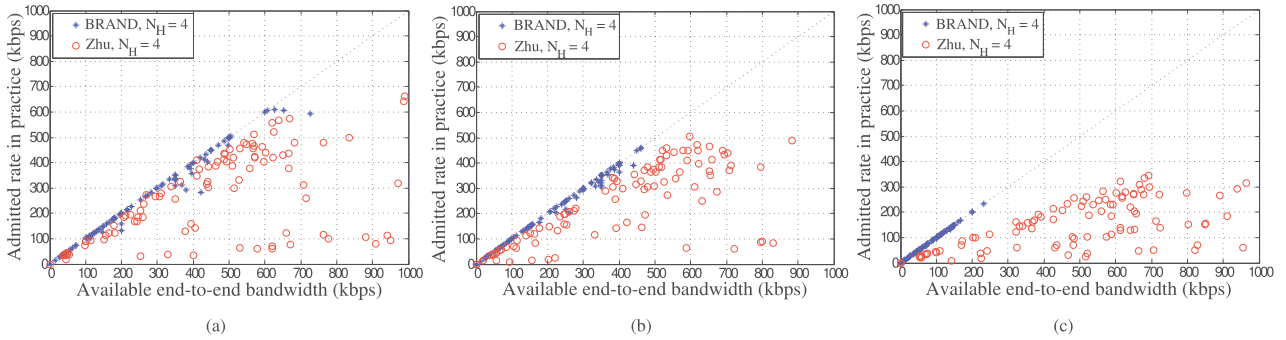


Fig. 7. The presence of multi-rate links and primary users does not affect BRAND’s accuracy. Ignoring them, however, leads to significant errors, as high as 900%, when calculating the available end-to-end bandwidth. (a) Multiple-Transceivers, Multiple-Rates,  $u=10\%$ . (b) Multiple-Transceivers, Multiple-Rates,  $u=20\%$ . (c) Multiple-Transceivers, Multiple-Rates,  $u=50\%$ .

### E. Cognitive Effect: Primary Users and Multi-Rate Links

We now evaluate the performance of BRAND for a cognitive network architecture in which nodes access the spectrum as secondary users. In addition to BRAND, we also use the Zhu heuristic in this part of the evaluation. The latter was designed for a single-rate, single-transceiver legacy architecture so clearly it would be unfair to expect it to perform as well as BRAND. Instead, the reason for which we include it in this evaluation is for quantifying the consequences of ignoring the primary user and multi-rate links when computing the available bandwidth.

To evaluate the “cognitive” effect, we repeat the experiment of the previous section using multi-transceivers, multiple-rates and varying probabilities of the Primary User occurrence. As shown in Figure 7 BRAND almost always estimates the correct value for the available capacity. On the other hand, ignoring the operating specifics of the cognitive radio networks leads to significant errors, as high as 900%, when calculating the available end-to-end bandwidth. A careful analysis of the data, especially that of Figure 7(c), does show that, at times, Zhu realizes a higher available bandwidth in simulations. As predicted by Equation 1 in Section III-C.2, protecting the Primary User comes at a cost in terms of bandwidth, explaining the smaller bandwidth values computed by BRAND at times.

### F. Distributed BRAND: Mobility, Channel Fading, Sensing Errors and Multiple Flows

In this experiment, we consider a more realistic scenario that includes mobility, channel fading and PU sensing errors for evaluating the performance of distributed BRAND. We set up a multi-hop cognitive radio network of 30 nodes placed at random on a 1000mx1000m area. We select at random four source-destination pairs and start generating unicast sessions. Whenever a new traffic session arrives we use the distributed implementation of BRAND for performing admission control. We carry out two sets of experiments, one with no mobility for isolating the effect of sensing errors and a second where we evaluate the effect of fading and node mobility.

Figure 8 shows the performance of distributed BRAND for various PU sensing error levels. As the data shows, BRAND is

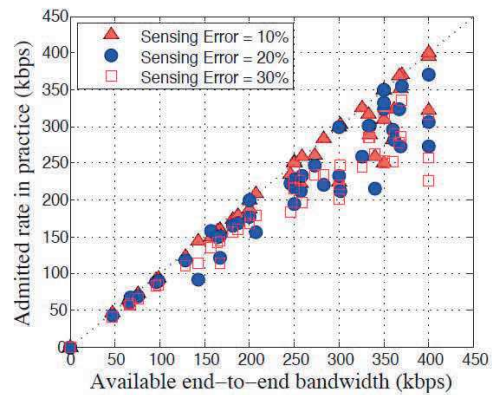


Fig. 8. Sensing error of  $x\%$  means the sensing module underestimates the PU activity by  $x\%$ . Every point is an average across all sessions in the network. For the most part, distributed BRAND deals well with sensing errors, even when those errors reach 30%.

resilient to sensing errors and that, even when the PU activity is underestimating by 30%, the estimation errors are modest in most cases.

Finally, we evaluate the effect of fading and mobility on distributed BRAND. For this set of experiments we use the random waypoint mobility model and consider two levels of maximum speed: 5m/s and 15 m/s, corresponding to pedestrian and vehicle movement, respectively. The routing tables are updated once a minute, as is the case for popular routing protocols like OLSR. In addition we add Rayleigh fading. As the data in Figure 9 shows, for static topologies, where the only source of error is channel fading, BRAND is accurate. When the mobility is introduced the accuracy starts to drop although the drop is limited at pedestrian speeds. As is to be expected, at vehicular speed the errors start becoming significant and we do not recommend using BRAND in this context.

### G. R-BRAND: Admission Control for Time-Sensitive Applications

Finally, in the following experiment we evaluate the performance of R-BRAND.

*Method:* The analysis is performed on 64kbps voice call flows with RTP packets sent once every 20ms. We carried

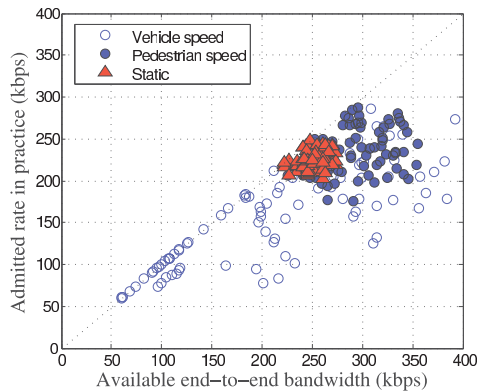


Fig. 9. Performance of distributed BRAND with mobility, channel fading and multiple concurrent flows.

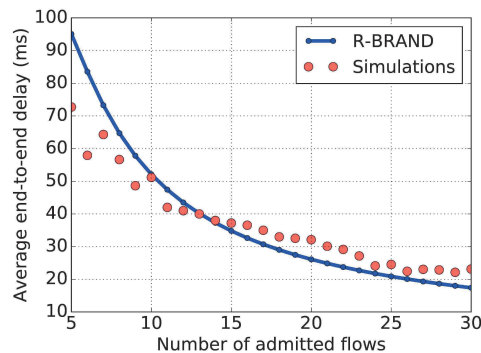


Fig. 10. End-to-end delay as a function of the number of admitted flows. A larger number of admitted flows means there was more available capacity initially, resulting in smaller end-to-end delays.

out the experiments on 4-hop paths with TDMA frame size  $S=32$  slots and buffers of size  $K=16$ . The duration of every time-slot was set to 0,625ms. All the links operated on data rates almost equaling 2Mbps. We compare the estimation for the the end-to-end delay given by R-BRAND with the average of what is observed over 5 simulation runs of 20-second duration.

*Results:* Figure 10 depicts the average end-to-end delay achieved by the RTP packets as a function of the number of flows admitted end-to-end by R-BRAND (packet delivery ratio  $> 95\%$ ). The data shows that R-BRAND is capable of performing delay-based admission control – its computation of end-to-end delay closely resembles what was observed during the simulations.

## VIII. CONCLUSION

In this paper, we revisited the problem of admission control for the cognitive radio context. With the problem NP-Complete, we relaxed the requirement for exact end-to-end bandwidth and replaced it with a quest for an average end-to-end bandwidth using randomized scheduling. We solved analytically the problem of estimating the average throughput with randomized scheduling by introducing the concepts of l-link available slots partition and clique sliding, reducing the calculation complexity from exponential to linear. The solution

was used as vehicle for BRAND, an algorithm for computing the average available bandwidth of a given source, destination pair. A thorough numerical analysis demonstrated the correctness of BRAND as well as its capability to accurately take into account the cognitive radio context, in particular the Primary User and multi-rate links. The distributed version of BRAND was shown to deliver almost 100% admission rate in a variety of conditions, thus, validating our choice to work with the relaxed version of the admission control problem. Finally, we introduced a version of BRAND tailored for time-sensitive applications and showed that it can perform delay-based admission control.

## REFERENCES

- [1] G. A. Gallardo, G. Jakllari, L. Canourgues, and A.-L. Beylot, "On estimating the end-to-end bandwidth in multi-transceiver multi-hop cognitive radio networks," in *Proc. 8th ACM Workshop Perform. Monitoring Meas. Heterogeneous Wireless Wired Netw.*, 2013, pp. 117–124.
- [2] B. Wang and K. J. R. Liu, "Advances in cognitive radio networks: A survey," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 5–23, Feb. 2011.
- [3] FCC Press. (Nov. 2008). *Evaluation of the Performance of Prototype TV-Band White Space Devices*. [Online]. Available: [http://hraunfoss.fcc.gov/edocs\\_public/attachmatch/FCC-08-260A6.pdf](http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-08-260A6.pdf)
- [4] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh, "White space networking with Wi-Fi like connectivity," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2009, pp. 27–38.
- [5] J. Mitola and G. Q. Maguire, Jr., "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, Apr. 1999.
- [6] L. Gavrilovska, D. Denkovski, V. Rakovic, and M. Angelichinoski, "Medium access control protocols in cognitive radio networks: Overview and general classification," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 2092–2124, 4th Quart., 2014.
- [7] Y. Zhao, M. Song, and C. Xin, "FMAC: A fair MAC protocol for coexisting cognitive radio networks," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 1474–1482.
- [8] X. Zhang and H. Su, "CREAM-MAC: Cognitive radio-enabled multi-channel MAC protocol over dynamic spectrum access networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 110–123, Feb. 2011.
- [9] K. R. Chowdhury and I. F. Akyildiz, "CRP: A routing protocol for cognitive radio ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 794–804, Apr. 2011.
- [10] M. Kodialam and T. Nandagopal, "Characterizing achievable rates in multi-hop wireless networks: The joint routing and scheduling problem," in *Proc. 9th Annu. Int. Conf. Mobile Comput. Netw.*, 2003, pp. 42–54.
- [11] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *Proc. 11th Annu. Int. Conf. Mobile Comput. Netw.*, 2005, pp. 73–87.
- [12] H. Li, Y. Cheng, P.-J. Wan, and J. Cao, "Local sufficient rate constraints for guaranteed capacity region in multi-radio multi-channel wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 990–998.
- [13] I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAND: Distributed randomized TDMA scheduling for wireless ad-hoc networks," in *Proc. 7th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2006, pp. 190–201.
- [14] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Commun. ACM*, vol. 21, no. 7, pp. 558–565, Jul. 1978.
- [15] H. Khalife, N. Malouch, and S. Fdida, "Multihop cognitive radio networks: To route or not to route," *IEEE Netw.*, vol. 23, no. 4, pp. 20–25, Jul./Aug. 2009.
- [16] M. Caleffi, I. F. Akyildiz, and L. Paura, "OPERA: Optimal routing metric for cognitive radio ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 8, pp. 2884–2894, Aug. 2012.
- [17] M. Pan, C. Zhang, P. Li, and Y. Fang, "Joint routing and link scheduling for cognitive radio networks under uncertain spectrum supply," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 2237–2245.
- [18] W. Li *et al.*, "Spectrum assignment and sharing for delay minimization in multi-hop multi-flow CRNs," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 11, pp. 2483–2493, Nov. 2013.

- [19] C. Zhu and M. Corson, "QoS routing for mobile ad hoc networks," in *Proc. IEEE INFOCOM*, vol. 2, Jun. 2002, pp. 958–967.
- [20] M. Alicherry, R. Bhatia, and L. E. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *Proc. 11th Annu. Int. Conf. Mobile Comput. Netw.*, 2005, pp. 58–72.
- [21] X. Lin and S. B. Rasool, "Distributed and provably efficient algorithms for joint channel-assignment, scheduling, and routing in multichannel ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 6, pp. 1874–1887, Dec. 2009.
- [22] E. Ahmed, A. Gani, S. Abolfazli, L. J. Yao, and S. U. Khan, "Channel assignment algorithms in cognitive radio networks: Taxonomy, open issues, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 795–823, 1st Quart., 2016.
- [23] H. Kim and K. G. Shin, "Efficient discovery of spectrum opportunities with MAC-layer sensing in cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 5, pp. 533–545, May 2008.
- [24] C. Jiang, Y. Chen, K. J. R. Liu, and Y. Ren, "Renewal-theoretical dynamic spectrum access in cognitive radio network with unknown primary behavior," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 3, pp. 406–416, Mar. 2013.
- [25] R. Murty, R. Chandra, T. Moscibroda, and P. Bahl, "SenseLess: A database-driven white spaces network," *IEEE Trans. Mobile Comput.*, vol. 11, no. 2, pp. 189–203, Feb. 2012.
- [26] S. Debroy, S. De, and M. Chatterjee, "Contention based multichannel MAC protocol for distributed cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 12, pp. 2749–2762, Dec. 2014.
- [27] B. Hamdaoui and K. G. Shin, "OS-MAC: An efficient MAC protocol for spectrum-agile wireless networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 8, pp. 915–930, Aug. 2008.
- [28] P. Ren, Y. Wang, and Q. Du, "CAD-MAC: A channel-aggregation diversity based MAC protocol for spectrum and energy efficient cognitive ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 2, pp. 237–250, Feb. 2014.
- [29] S. Deb, P. Gupta, K. Nagaraj, and V. Srinivasan, "An agile and efficient MAC for wireless access over TV whitespaces," *IEEE Trans. Mobile Comput.*, vol. 14, no. 1, pp. 42–57, Jan. 2015.
- [30] F. Marino, L. Paura, and R. Savoia, "On spectrum sensing optimal design in spatial-temporal domain for cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 8496–8510, Oct. 2016.
- [31] C. Jiang, Y. Chen, Y. Gao, and K. J. R. Liu, "Joint spectrum sensing and access evolutionary game in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 5, pp. 2470–2483, May 2013.
- [32] G. A. Gallardo, G. Jakllari, L. Canourgues, and A.-L. Beylot, "Statistical admission control in multi-hop cognitive radio networks," Dept. Telecommun. Netw., Univ. Toulouse, Toulouse, France, Tech. Rep., Jan. 2017. [Online]. Available: [http://jakllari.perso.enseiht.fr/Publications\\_files/BRANDTechReport.pdf](http://jakllari.perso.enseiht.fr/Publications_files/BRANDTechReport.pdf)
- [33] D. A. Maltz, J. Broch, J. Jetcheva, and D. B. Johnson, "The effects of on-demand behavior in routing protocols for multihop wireless ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1439–1453, Aug. 1999.
- [34] P. Jacquet *et al.*, "Optimized link state routing protocol for ad hoc networks," in *Proc. IEEE INMIC*, Dec. 2001, pp. 62–68.
- [35] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.
- [36] M. Berkelaar, K. Eikland, and P. Notebaert. (May 2004). *LP Solve 5.5, Open Source (Mixed-Integer) Linear Programming System*. [Online]. Available: <http://lpsolve.sourceforge.net/5.5/>



**Guillaume Artero Gallardo** received the Engineering and Ph.D. degrees in telecommunications and networks from INP Toulouse, France, in 2011 and 2015, respectively, and the M.Eng. degree from Dublin City University. He was a Systems Engineer with Rockwell Collins, France, from 2011 to 2014, where he conducted research in software defined radio networks. He was a member of the IRIT and LIP research laboratories, France, from 2015 to 2016. He is currently a Research Engineer with Sysoco Group, France. His current research interests concern networking technologies with an emphasis on the performance evaluation and design of multimedia wireless communications systems, including cognitive radio and mission-critical LTE.



design for gigabit wireless, smartphone-based localization and navigation, IoT, and energy efficiency.

**Gentian Jakllari** received the Ph.D. degree in computer science from the University of California at Riverside, Riverside, USA, in 2007. From 2007 to 2011, he was a Network Scientist with the Internetworking Research Department, BBN Technologies, Cambridge, MA, USA. In 2011 he joined INP-ENSEEIH, University of Toulouse, France, where he is currently an Associate Professor. His research interests are in computer networks with emphasis on wireless networking and mobile systems, including cross-layer and cross-technology



**Lucile Canourgues** received the engineering and Ph.D. degrees in telecommunications and networks from INP Toulouse, France, in 2004 and 2008, respectively. She was a member of the IRIT Research Laboratory, France, from 2005 to 2008. She is currently a Principal Program Manager with Rockwell Collins, France. Her current research interests include networking technologies with an emphasis on the performance evaluation and design of multimedia wireless communications systems, including cognitive radio and mission-critical LTE.



His research interests focus on performance and design of wireless networks.

**André-Luc Beylot** received the Ph.D. degree in computer science from the University of Paris 6 in 1993, and the Habilitation Diriger des Recherches from the University of Versailles in 2000. Since 2000, he has been a Professor with the Telecommunication and Network Department, ENSEEIH. From 2008 to 2011, he has been leading the IRT Team, Institut de Recherche en Informatique de Toulouse (IRIT) Laboratory. Since 2011, he has been the Head of the ENSEEIH site, IRIT. Since 2015, he has been the Head of GDR CNRS - RSD.