Statistical Admission Control in Multi-Hop Cognitive Radio Networks

Guillaume Artero Gallardo\textsuperscript{1,2}, Gentian Jakllari\textsuperscript{1}, Lucile Canourgues\textsuperscript{2}, André-Luc Beylot\textsuperscript{1}
\textsuperscript{1}Department of Telecommunications and Networks, ENSEEIHT, University of Toulouse
\textsuperscript{2}Rockwell Collins France
\{guillaume.arterogallardo, jakllari, beylot\}@enseeiht.fr
lcanourg@rockwellcollins.com

Technical Report

Abstract

Cognitive radios promise to revolutionize the performance of wireless networks. Realizing this promise, however, requires revisiting and/or reinventing many of the current network architectures, protocols and policies. In this work we focus on Quality of Service routing and more specifically, admission control. We consider a multi-hop cognitive radio network where every node is equipped with multiple transceivers. In addition, because the research and development of a widely accepted MAC protocol for these networks is still ongoing, we assume a bare-bones TDMA protocol at the link layer. We show that for the network considered the problem of computing the available bandwidth of a given path – necessary for admission control – is NP-Complete. The reason for this result is that computing the available bandwidth of a path requires allocating capacity on every hop of the path such that the realized end-to-end throughput is maximized – and that is an intractable problem. Instead of working on an approximation algorithm we follow a different approach. We use a simple scheduling heuristic: Each node selects the slots on which to transmit by choosing at random among those available. Randomized scheduling is widely used because of its simplicity and efficiency. However, computing the resulting average throughput over a multi-hop path has been an open problem. We solve this problem and use the solution as a vehicle for BRAND, an algorithm for computing the maximum realizable throughput – the available Bandwidth – with RANDomized scheduling of a multi hop path in a cognitive radio network. We show that BRAND can be implemented in a distributed fashion and thus integrated by almost all routing approaches. An extensive numerical analysis demonstrates the accuracy of BRAND and its enabling value in performing admission control.

1 Introduction

Currently, the wireless spectrum is tightly regulated by centralized authorities, such as the FCC in the United States. Operating on wireless spectrum requires either applying for an exclusive license by the respective governmental authority or, requires using select frequencies, like those in the ISM band, that are freely available. However, driven by unprecedented demand for wireless capacity and increasing evidence that a lot of the licensed spectrum is underutilized \[1\] 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 \[2, 3, 4\].

The US President’s council of advisors on science and technology in its 2012 report \[5\] concludes
that, when it comes to the wireless spectrum, ”the norm should be sharing, not exclusivity”,
and estimates that a new architecture and a corresponding shift in practices could multiply the
effective capacity of the spectrum by a factor of 1,000.

This shift from the regulating authorities, coupled with the emergence of the cognitive radio
network concept \[6\] as the enabling technology, has ignited tremendous interest in cognitive ra-
dio networks that are capable of intelligently exploiting the wireless spectrum \[1\]. Nevertheless,
considering the architectural changes required, a lot of technical and policy challenges need to
be ironed out before this vision can turn into reality \[4, 7\]. Towards this, a lot of effort has been
put in developing solutions for cognitive radio networks at the physical, link and the routing
layers \[8, 9, 10, 11, 12, 13, 14, 15, 16\]. The IEEE 802 standards committee has also gotten
involved \[17\]. However, we are far from having all the answers.

One particular question that has not been addressed is that of QoS routing and more
specifically, admission control in multi-hop cognitive radio networks. Provided a newly arriving
traffic session, before admitting it, we would like to know whether there is enough bandwidth
for serving the session’s traffic demand. The question can be answered by first computing the
bandwidth currently available on the path from the source to the destination node. This is
the focus of this work. In particular, we focus our attention on the problem of computing
the available end-to-end bandwidth for a cognitive network where every node is equipped with
multiple transceivers \[1\] and the link layer uses a TDMA protocol \[19, 20, 17, 21, 22\].

Computing the end-to-end bandwidth is not a new problem \[23\]. However, there are two
reasons for revisiting it. First, there is no practical solution widely accepted for legacy \[2\] TDMA
wireless networks \[24\]. Second, the cognitive radio architecture makes the problem non-trivially
different. The main reason for this is the existence of the so called primary-secondary hierarchy \[3\].
In issuing its landmark ruling \[2\], 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 in these networks will be conditioned
not only by the interference between peers, as is the case in legacy wireless networks, but also
by a different kind of interference: Primary users, as their prerogative, will access the channel
without making any effort to avoid interfering with secondary users. In addition to the primary
user interference, another distinguishing feature of these networks is that cognitive radios, in
their pursuit of available spectrum, may access different frequency bands and use different
channel widths \[2\]. This will lead to network links with widely varying capacities. Multi-rate
links can also be seen in legacy networks, such as IEEE 802.11, but in cognitive networks they
are a fundamental feature that has to be taken into account by any bandwidth calculation
algorithm.

In this work, we model and analyze the problem of computing the available end-to-end
bandwidth in multi-hop cognitive radio networks. We provide a solution to the problem that is

\[1\] The cost and size of transceivers is rapidly dropping with spectrum remaining the bottleneck. Accessing
wide parts of spectrum will be best served by multiple transceivers \[13\].

\[2\] We refer to non-cognitive architectures as legacy.

\[3\] In the future, if the vision of a fully dynamic spectrum sharing becomes a reality, we will probably see more
than two classes of users and respective channel access priorities. However, for the moment we focus on the
basic case of a two-class hierarchy.
practical and yet takes into consideration the particularities of cognitive radio networks. Our contributions can be summarized as follows:

- We formally define the problem of computing the available end-to-end bandwidth of a path in TDMA based multi-hop cognitive radio networks and show that, as in the case of legacy networks, the problem is NP-Complete.

- We introduce BRAND, a heuristic for estimating the available end-to-end bandwidth. BRAND works by using a randomized slot scheduling and computing 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.

- As part of BRAND, we introduce a novel, linear time algorithm that can compute the average end-to-end throughput of a multi-hop path resulting from using randomized slot scheduling on every link of the path. Randomized scheduling is widely used because of its simplicity and efficiency [25]. However, computing the end-to-end throughput resulting from its application on a multi-hop path, to the best of our knowledge, was an open problem prior to this work.

- We show that BRAND can be implemented in a distributed fashion and can be used by most routing protocol approaches.

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 in computing the average end-to-end throughput as well as BRAND’s capability in providing correct information for performing admission control in the presence of primary users and multi-rate links.

The rest of the paper is organized as follows. In Section 2, we describe in detail the system model, including how we model the effect of the primary user. In Section 3, we formally define the bandwidth calculation problem and introduce BRAND. In Section 4, we show how BRAND can be implemented in a distributed fashion while in Section 5 we show the performance evaluation. In Section 6, we discuss some related work. Finally, we conclude the paper in Section 7.

2 Model and Preliminaries

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

2.1 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 is composed of only symmetric links, that is, there exists an edge between two vertices $v_i$ and $v_j$ if and only if nodes $n_i$ and $n_j$ are able to correctly communicate with each other. 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 \[18\]. All the channels can offer different data rates. An additional transceiver could be used for control signaling. We assume the channel assignment is performed by a spectrum allocation protocol \[26\] 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.

### 2.2 Cognitive Channel Access

As a prerequisite for using licensed spectrum, cognitive radios are not to use the channel when it is in use by the respective licensed user. In literature, this is referred to as a secondary-primary\[4\] hierarchy, with the primary (licensed) user having strict priority in 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 \[27\] \[28\] \[29\]. In response to this challenge, many new MAC protocols for cognitive radio networks have been proposed \[9\]. For all the diversity in the proposed solutions, one thing underlying all protocols is the need for a sensing module whose responsibility is identifying when the cognitive radio may be interfering with a primary user. In its 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 \[30\].

Given the functionality of the sensing module, a MAC protocol for cognitive radio networks needs to provide periods of network silence to be dedicated to sensing for primary user activity. What is more, since the activity of the primary user may be completely unknown, these dedicated 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. A requirement that can be easily accommodated by a TDMA protocol. Indeed, a majority of the MAC protocols proposed for cognitive radio networks \[19\] \[20\] \[17\] \[21\] \[22\], including the IEEE 802.22 MAC \[17\], are based on TDMA. Nevertheless, some solutions based on random access have also been proposed \[11\] \[31\] \[32\] \[10\].

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 the sensing. Therefore, we adopt a system in which a TDMA MAC with frame size \(S\) is implemented on every assigned channel. Every time-slot is started by a sensing period as illustrated in Figure 1. When a node needs to transmit data to a neighboring node, it can access 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.

### 2.3 Interference in Cognitive Radio Networks

There are two kind of interference sources in a cognitive radio network. There is the interference from other cognitive radios in the same interference domain, usually referred to as secondary-interference.\[4\] For the rest of the paper, we will use the terms primary user/secondary user, primary/secondary and PU/SU interchangeably.
Figure 1: The TDMA frame is composed of $S$ time-slots. During the sensing period, the secondary users remain silent and use energy detection to determine the channel occupancy.

to-secondary interference. And, there is the interference from 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 chose 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 source of interference are modeled in this work.

2.3.1 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, should a particular node need to reserve a new time-slot to transmit data to a neighbor, it does so on the corresponding assigned channel. However, due to the potential interference from other cognitive radios operating on the same channel, for the time-slot to be selected, it needs to satisfy the following requirements:

1. This time-slot is not used on this channel by node itself for transmitting,
2. This time-slot is not used on this channel by any one-hop neighbor for transmitting,
3. This time-slot is not used on this channel by any two-hop neighbor for transmitting.

We assume every node knows the slot allocations in its two-hop neighborhood\textsuperscript{5} and thus can check the satisfiability of the above constraints.

2.3.2 Quantifying the Primary User Interference

Once the sensing module identifies a primary user, the cognitive radios are to 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, as shown in Figure 1, an amount of time in every slot is dedicated to sensing, while the rest for actual channel access\textsuperscript{6}.

\textsuperscript{5}This information can be easily obtained by sending beacons containing bitmaps with slots scheduled for transmission or reception for the node itself and its one-hop neighbors.

\textsuperscript{6}The optimal ratio between sensing and channel access will depend on several factors, including the Primary User activity, the traffic demands for the cognitive radio, etc. A thorough study of these factors for computing
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.\footnote{We assume that if a secondary can sense a primary user then the particular secondary is in the interference range of the primary user.}

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:

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

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:

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

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

$$f_l = \eta_l \times \frac{T_{\text{access}}}{T_{\text{sensing}} + T_{\text{access}}} \quad (2)$$

The simplified formula for Equation 1 can be formally proved by modeling the PU’s activity with an alternative ON/OFF process \cite{28, 29}. The details of the proof can be found in the Appendix.

Equation 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 bound 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. Note that, for a primary activity of 10\%, the secondary user will not realize more than 81\% of the slot access time capacity. At first, one might think that the secondary should instead be able to reach 90\%. The explanation for the 9\% loss is the sensing. A primary could be active during the sensing period but not so during the access time and yet, the secondary will not use the access time, leading to unnecessary loss of capacity.

\footnote{The optimal sensing time is beyond the scope of this paper. However, the correctness of our scheme does not depend on the exact values of sensing and access times.}
3 Computing the available end-to-end bandwidth of a path

3.1 Problem Definition

Let the demand $d$, expressed in bits per second, refer to the amount of end-to-end bandwidth demanded by an application. Before admitting to route this demand, we first would 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 primary user activity, the available bandwidth is equivalent to the maximum path bandwidth.

Remark 2. The admission control problem could alternatively be framed as one of computing the path with the maximum available bandwidth. However, this would require implementing a new routing protocol for this purpose. What is more, given that we want to solve this problem online, as the traffic sessions arrive in real-life, it is not clear that, overall, it would lead to more sessions being admitted. Thus, we opted for an approach that can be added to any available routing protocol that, once computing the routes based on whatever metric it considers crucial, can simply asks us for the available bandwidth on a particular path.

Constructing the formal problem definition: A path is modeled as a directed chain $n_1 \rightarrow n_2 \cdots \rightarrow n_{N_H+1}$ composed of $N_H$ 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 and, how many slots are already allocated on every hop $i \in \{1, \ldots, N_H\}$ for servicing this flow. For every hop this number will depend on both the demand and, how many slots are

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

Let $a_i$ denote the number of slots specifically allocated on every hop $i \in \{1, \ldots, N_H\}$ for servicing this flow.
actually available for new allocations. Thus, for the first hop we have \( a_1 = \min(r_1, A^{(1)}) \). 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, \( d_1 \). 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:

\[
r_i = \left\lceil \frac{d_{i-1}}{\phi_i \times f_i} \times S \right\rceil \quad (3)
\]

\[
a_i = \min \left( r_i, A^{(i)} \right) \quad (4)
\]

and

\[
d_i = \min \left( d_{i-1}, a_i \times \left( \frac{\phi_i \times f_i}{S} \right) \right) \quad (5)
\]

Thus, for a specific demand \( d \), the realized end-to-end throughput is \( \min \left( d_1, d_2, ..., d_{NH} \right) = d_{NH} \), 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:

\[
\max_{d \in I_d} d_{NH}(d) \quad (6)
\]

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

The optimization problem thus defined leads to two observations:

1. The realized end-to-end throughput, \( d_{NH}(d) \), given a demand, \( d \), obviously depends on \( d \).

2. \( d_{NH}(d) \) depends on how the slots are allocated on every hop.

**Why computing the available end-to-end bandwidth is challenging:** To illustrate the implications of the above observations and how challenging the above optimization problem is, let us consider the following toy example. Consider the first three nodes, A, B, C of a multi-hop path and let us assume they all have assigned the same channel, which puts them in the same interference domain. The TDMA frame consists of ten slots. In node A slots 1,2,5,6 are available for transmitting and receiving, in node B, slots 2,5,6 are available for transmitting and receiving while slot 1 is available for receiving only\(^9\), and in node C all slots are available for transmitting and receiving. If the demand \( d \) is such that four slots are required for satisfying it, the first node, A, will allocate slots 1,2,5,6. Since nodes B, C are in the same interference domain, node B will not be able to transmit to node C on any of its slots available for transmitting. This will result in a zero end-to-end throughput. Now let us consider that the demand \( d \) is such that two slots are required for satisfying it. In this case node A has \( \binom{4}{2} \) ways to chose the two slots for allocation out of the four available. While for node A all choices are

\(^9\)It can happen that a node scheduled to receive on slot 1 is in the same interference domain with B but not A.
equivalent, that is not the case for nodes B and C and ultimately, the end-to-end throughput. If A allocates slots 2 and 5, node B will be left with only one slot, slot 6, to use for forwarding traffic to node C. If, instead, A allocates slots 1,2, node B will be left with two slots, slots 5,6, for forwarding traffic.

Clearly, the way slots are allocated on every node will have an impact on the realized end-to-end throughput. What is more, the number of possible schedules can grow exponentially with the number of nodes on the path. This leads to the following result.

**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 $P_1$ actually consists of solving one instance of $P_2$. Since $P_2$ has been shown to be NP-complete[23], that concludes the proof.

3.2 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 [25]: when a node needs to assign a certain number of slots, it will select at random among those available.

Therefore, our high-level algorithm for estimating the available end-to-end Bandwidth with RANDom scheduling, BRAND, 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. By Equation 6, the available end-to-end bandwidth is simply the maximum end-to-end throughput realized over all possible demands $d$.

**Algorithm 1: BRAND** (Available Bandwidth with RANDom Scheduling)

<table>
<thead>
<tr>
<th>Output: The available end-to-end bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 : begin</td>
</tr>
<tr>
<td>2 : for every possible demand $d$ do</td>
</tr>
<tr>
<td>//Use Random Scheduling</td>
</tr>
<tr>
<td>avThput ← Compute-averageThput (d);</td>
</tr>
<tr>
<td>3 : if avThput &gt; AvailBW then</td>
</tr>
<tr>
<td>AvailBW ← avThput;</td>
</tr>
<tr>
<td>4 : Return AvailBW;</td>
</tr>
</tbody>
</table>

9
The value of the demand, $d$, is upper-bounded by the lowest radio bit-rate in the network and lower-bounded by 0. Since it is a slotted system, the values of $d$ have to be multiples of the 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 the following, we give a centralized approach that, given a demand $d$, computes the average end-to-end throughput realized. In Section 4 we propose a distributed approach.

### 3.3 Computing the Average End-to-End Throughput with Random Scheduling

In the following, we propose an analytical framework for computing the average throughput that would be achieved on every link of a path if a new flow with demand $d$ were to be admitted and random scheduling is used on every link. This solution is centralized in that, the source node is assumed to have global knowledge of the network.

#### 3.3.1 The Path’s Slots Availability Table (PSAT)

For the sake of clarity, we define a new data structure indicating, for every link of a path, the slots available for reservation. We call this structure the path’s slots availability table (PSAT). This table is composed of $N_H$ lines and $S \times B$ columns. Each line $i$, composed of $B$ sub-blocks of size $S$, indicates the available slots on link $l_i$. Then, each sub-block refers to the available time-slots on each sensed frequency channel. All the time-slots of each of the $B - 1$ frequency channels not selected for communication on link $l_i$, are considered unavailable and the corresponding entries in the table are set to 0. A simple example of a PSAT table is depicted in Figure 2. Whenever at a node the spectrum decision module carries out channel reassignment, the PSAT is updated accordingly. Therefore, the PSAT summarizes all the necessary information required for computing the average number of slots that would be allocated on every hop of the path if a new flow with demand $d$ were to be admitted.
3.3.2 Fundamental principles of the method

To make the problem tractable, we relax it by working on average values. Even though mathematically speaking $E[a_3] \neq \min \left( E[r_3], E[A_3^{(3)}] \right)$, we do such an approximation of the average number of slots allocated on the third hop to reduce the calculation complexity. As depicted in section 5, our simulation results show that this approximation does not degrade the performance of the overall estimation process. Based on this assumption, a first solution consists of estimating for each hop $i$ the quantity $A_i^{(i)}$ which is now considered an average for the remaining of the paper.

$l$-link available slot set decomposition: From the PSAT, we can calculate for any communication link $l_i$ the set $S_i^{(1)}$ containing the index of slots available for reservation on that link at the beginning of the estimation process. The indexes of slots are now taken in the set $\{1,2,\ldots,S.B\}$ related to the super-frame composing the whole line in the PSAT. It is also crucial to see that, when considering such an indexation, when a slot is allocated on $l_i$, it cannot be allocated anymore on both $l_{i+1}$ and $l_{i+2}$ as this would create interference. Thus, when this slot is only available for reservation on link $l_i$ and neither on links $l_{i+1}$ nor $l_{i+2}$, the sets $S_{i+1}^{(j)}$ and $S_{i+2}^{(j)}$ are not impacted. Inversely, if this slot is also available to one of these links, the corresponding sets are impacted and thus the number of slots that would be allocated on the next hops is likely to decrease. Thus, each slot belongs to a certain category depending on the links it appears to be available for reservation to. Therefore, we propose to divide the set $\{1,2,\ldots,S.B\}$ in non overlapping subsets that cover $\{1,2,\ldots,S.B\}$ and permit to categorise every slot according to the links it is available for reservation to in the PSAT table. To be more precise, we define such a decomposition on a set of $l$ consecutive links $\{i,i+1,\ldots,i+l-1\}$ along the path. For $k \in \{0,1,\ldots,l\}$, the number of subsets characterizing the slots available to a set of $k$ links but not the $l-k$ others is $\binom{l}{k}$. Thus, the total number of subsets in the decomposition is $\sum_{k=0}^{l} \binom{l}{k} = 2^l$. We refer to such a decomposition as a $l$-link available slot set decomposition and use the following notations $E_{a,b,c}^{(j)}$ to denote the set of slots available for reservation on both two links $a$ and $c$ but not $b$ just before node $n_j$ does its allocations. Its cardinality is written as $C_{a,b,c}^{(j)}$.

To elucidate the meaning of these variables, let us consider a 3-hop path with the first channel selected for communication on every hop, $B = 2$, $S = 8$ and $S_1^{(i)} = \{2,3,4,5\}$, $S_2^{(i)} = \{2,6,7\}$ and $S_3^{(i)} = \{1,2,4,5,6,7\}$. This leads to the following eighth sets: $E_{1,2,3}^{(1)} = \{2\}$, $E_{1,2,3}^{(1)} = \emptyset$, $E_{1,2,3}^{(1)} = \{4,5\}$, $E_{1,2,3}^{(1)} = \{6,7\}$, $E_{1,2,3}^{(1)} = \{3\}$, $E_{1,2,3}^{(1)} = \emptyset$, $E_{1,2,3}^{(1)} = \{1\}$ and $E_{1,2,3}^{(1)} = \{8,9,\ldots,16\}$.

Achievable end-to-end throughput calculation: The random nature of the slot allocation process provides good properties to evaluate the average number of slots impacted in every subset as further depicted for the case of a 3-hop path for which a flow with demand $d$ needs to be relayed from the source to the destination. From now on, we work with average values. From the PSAT, we can compute the 3-link available slot set decomposition related to $l_1$, $l_2$ and $l_3$. At the same time, the initial number of available slots on every communication link $A_i^{(i)}$ can be calculated. To forward the new traffic flow to its next hop $n_2$, node $n_1$ reserves exactly $a_1 = \min (r_1, A_1^{(i)})$ additional slots on the first communication link. Among these slots, some might have also been available for reservation on links $l_2$ and $l_3$ but, due to interference,
become unavailable after these allocations.

Let us consider a discrete random variable \( X_i \) taking its values in the set \( \{0, 1, \ldots, a_i\} \) and representing the number of slots initially available for reservation on link \( l_i \) that have been reserved by node \( n_1 \) to relay the new incoming flow on \( l_1 \). \( X_i \) represents the number of slots in the set \( S_i^{(1)} \cap S_i^{(2)} \) reserved by \( n_1 \) for communication on \( l_1 \). Intuitively, as the slots are allocated at random, we see that a proportion \( a_1/A_1^{(1)} \) of these slots are likely to be reserved by \( n_1 \). Mathematically speaking, \( X_i \) follows an hypergeometric distribution with parameters \( (A_1^{(1)}, |S_i^{(1)} \cap S_i^{(2)}|, a_i) \). The expectation value of such a random variable is \( E[X_i] = |S_i^{(1)} \cap S_i^{(2)}| \times a_1/A_1^{(1)} \) and thus, in every set \( S_i^{(1)} \cap S_i^{(2)} \), an average proportion \( p_1 = a_1/A_1^{(1)} \) of slots is reserved by node \( n_1 \). Note that for the case of \( A_1^{(1)} = 0 \) we get \( a_1 = 0 \) and \( p_1 = 0 \). Exactly the same analysis can be carried out on every set resulting from the 3-link available slot set decomposition related to \( l_1, l_2 \) and \( l_3 \). This way, the average values \( A_2^{(2)} \) and \( A_3^{(2)} \) just after \( n_1 \) did its reservations can be computed as detailed in Algorithm 2 and illustrated in Figure 3. Therefore, the average number of slots allocated on the following links can be evaluated and the process repeated until the average number of slots allocated on the last hop is calculated.

![Figure 3: When slot allocations are carried out on \( l_1 \) the proportion of slots allocated, \( p_1 \), is removed from every subset that characterizes slots initially available for reservation on \( l_1 \), \( E_{1,2,3}^{(1)} \). This proportion is represented by plain areas in the above figures. The selected slots become unavailable on all three links and thus are transferred to the set of unavailable slots for those links. The same mechanism is then repeated for the allocations carried out on \( l_2 \). This time, the slots are selected at random among those remaining available for reservations. The proportion of slots selected, \( p_2 \), represented by dotted areas, is thus taken in every subset characterizing slots remaining available for allocation on \( l_2 \), \( E_{1,2,3}^{(2)} \). Finally, the same mechanism is in place for allocations on link \( l_3 \), as depicted by the striped areas.](image)

This approach still applies when increasing the path length. However, the impact of \( n_1 \) allocations is still required to be evaluated when computing the average number of slots that would be allocated on any further communication link \( l_i \). Such an impact can be evaluated by first doing the \( i \)-link available slot set decomposition and then carefully measuring the dependence of each of the previous node allocations. This leads to an exponential number of
Algorithm 2: Computing the average end-to-end throughput on a 3-hop path.

\begin{algorithm}
\begin{algorithmic}
\State \text{input : } d, S, \phi_1, \phi_2, \phi_3, f_1, f_2, f_3, S^{(1)}_1, S^{(1)}_2, S^{(1)}_3
\State \text{output: } a_1, a_2, a_3, d_1, d_2, d_3
\State \begin{algorithmic}
\State //\text{Initialization}
\State \forall i \in \{1, 2, 3\}, A^{(i)}_1 \leftarrow |S^{(i)}_i|;
\State //\text{Available slot set decomposition}
\State C^{(1)}_{1,2,3}, C^{(1)}_{1,2,3}, C^{(1)}_{1,2,3}, C^{(1)}_{1,2,3}, C^{(1)}_{1,2,3}, C^{(1)}_{1,2,3};
\State //\text{Allocations on } l_1
\State r_1 \leftarrow \left\lceil \frac{d}{\phi_1 f_1} \times S \right\rceil;
\State a_1 \leftarrow \text{min}(r_1, A^{(1)}_1);
\State d_1 \leftarrow \text{min}(d, a_1 \times \phi_1 f_1 S);
\State p_1 \leftarrow a_1/A^{(1)}_1;
\State A^{(2)}_2 \leftarrow A^{(1)}_2 - p_1.(C^{(1)}_{1,2,3} + C^{(1)}_{1,2,3});
\State A^{(2)}_3 \leftarrow A^{(1)}_3 - p_1.(C^{(1)}_{1,2,3} + C^{(1)}_{1,2,3});
\State //\text{Allocations on } l_2
\State r_2 \leftarrow \left\lceil \frac{d}{\phi_2 f_2} \times S \right\rceil;
\State a_2 \leftarrow \text{min}(r_2, A^{(2)}_2);
\State d_2 \leftarrow \text{min}(d_1, a_2 \times \phi_2 f_2 S);
\State p_2 \leftarrow a_2/A^{(2)}_2;
\State A^{(3)}_3 \leftarrow A^{(2)}_3 - p_2.(C^{(1)}_{1,2,3} + (1 - p_1).C^{(1)}_{1,2,3});
\State //\text{Allocations on } l_3
\State r_3 \leftarrow \left\lceil \frac{d}{\phi_3 f_3} \times S \right\rceil;
\State a_3 \leftarrow \text{min}(r_3, A^{(3)}_3);
\State d_3 \leftarrow \text{min}(d_2, a_3 \times \phi_3 f_3 S);
\end{algorithmic}
\end{algorithmic}
\end{algorithm}

sets to deal with which makes the solution not tractable. We refer to this phenomenon as the domino effect. To address this issue, in the following we introduce the interference-clique sliding approach that breaks the domino effect and reduces the calculation complexity from exponential to linear.

\subsection{3.3.3 Interference-Clique sliding approach}

The interference-clique\footnote{For the rest of the document, we will use the terms clique and interference-clique interchangeably.} sliding approach breaks the domino effect by processing the end-to-end bandwidth estimation clique by clique while using only a linear number of variables. The basic idea consists of eliminating the dependence on the allocations that took place in the previous nodes. We define an interference-clique, or simply clique, as any set of three consecutive links.
on the path. For instance, a 4-hop path is composed of two cliques: \( c_1 = \{ l_1, l_2, l_3 \} \) and \( c_2 = \{ l_2, l_3, l_4 \} \).

**Initialization:** Given a path of length \( N_H \), we start by computing the available slot sets resulting from the 3-link available slot set decomposition of every clique. This leads to eight corresponding sets for each clique, that is, for the \( i^{th} \): \( E_{i,i+1,i+2}, E_{i,i+1,i+2}, F_{i,i+1,i+2}, E_{i,i+1,i+2}, E_{i,i+1,i+2}, E_{i,i+1,i+2}, E_{i,i+1,i+2} \). The following describes how to extend the bandwidth estimation process when sequentially passing the calculation on to the next cliques.

**Interference-Clique 1:** The clique 1 is the easiest to process as it does not depend on any previous allocations. As described above, exactly \( a_1 = \min(\tau_1, A_1) \) slots are reserved for communication on link \( l_1 \). Then, the slots remaining available for communication on link \( l_1 \) are not considered anymore and the calculation is passed on to clique 2.

**Interference-Clique 2:** To process any clique \( i \), we calculate \( a_i \) by first estimating \( A_i \), the average number of slots remaining available for reservation on link \( l_i \) just before \( n_i \) does its allocations. Given the 3-link available slot set decomposition of clique \( i \), we get:

\[
A_i = C_{1,i+1,i+2}^{(i)} + C_{i,i+1,i+2}^{(i)} + C_{i,i+1,i+2}^{(i)} + C_{i,i+1,i+2}^{(i)}
\]

Indeed, all the resulting sets of the decomposition are disjointed and form a partition of the global slot set \( \{1, \ldots, S, B\} \). Then, to correctly measure the impact caused on clique 2 sets by reservations done on \( l_1 \), we just extend the 3-link available slot set decomposition related to clique 2 to the 4-link decomposition including \( l_1 \). This way, we note that:

\[
C_{2,3,4}^{(1)} = C_{1,2,3,4}^{(1)} + C_{1,2,3,4}^{(1)}
\]

From this equation, we infer that 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 for communication on link \( l_1 \). Using this principle and considering that there is no interference between \( l_1 \) and \( l_4 \), the clique 2 sets can be updated as follows:

\[
C_{2,3,4}^{(2)} = C_{2,3,4}^{(1)} - p_1 C_{1,2,3,4}^{(1)}
\]

As depicted in the previous equations, some sets receive new slots. This phenomenon results from slot allocations on \( l_1 \) having a different impact on slots initially available for reservation.
on links $l_2$, $l_3$ and $l_4$. Indeed, due to the 2-hop nature of the interference, a proportion of slots that were initially available in common for $l_1$, $l_2$, $l_3$ and $l_4$ have become unavailable to $l_2$ and $l_3$ and thus become exclusively available to $l_4$. At this point, it is possible to correctly calculate the average values of $A_2^{(2)}$, $r_2$, $a_2$, $d_2$ and $p_2 = a_2/A_2^{(2)}$.

**Interference-Clique 3:** Exactly the same interference phenomenon occurs when processing the third clique. However, as this clique suffers from interference created by allocations on both previous links $l_1$ and $l_2$, the same approach needs to be followed by extending the available slot set decomposition including these two links. It is even more complex than that since (1) $l_1$ interferes only with $l_3$, not $l_4$ and (2) $l_3$ suffers from interferences created by allocations on both $l_1$ and $l_2$ as:

$$C_{3,4,5}^{(1)} = C_{3,4,5}^{(1)} + C_{3,4,5}^{(1)} \sum_{\text{impacted by allocations on } l_1 \text{ and then } l_2} + C_{3,4,5}^{(1)} \sum_{\text{impacted by allocations on } l_1 \text{ but not } l_2} + C_{3,4,5}^{(1)} \sum_{\text{impacted by allocations on } l_2 \text{ but not } l_1} + C_{3,4,5}^{(1)} \sum_{\text{not impacted}}$$

(10)

that leads to:

$$C_{3,4,5}^{(3)} = C_{3,4,5}^{(1)} - [p_1 + p_2(1 - p_1)] C_{3,4,5}^{(1)} - p_1 C_{3,4,5}^{(1)} - p_2 C_{3,4,5}^{(1)}$$

(11)

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 $p_1 \left[ C_{3,4,5}^{(1)} + C_{3,4,5}^{(1)} \right]$ slots and $[p_2(1 - p_1) C_{3,4,5}^{(1)} + p_2 C_{3,4,5}^{(1)}]$ slots in the set $E^{(1)}_{3,4,5}$ are respectively reserved on links $l_1$ and $l_2$. Due to the 2-hop nature of the interference, when updating the sets resulting from the 3-link decomposition related to clique 3, on average $p_1 \left[ C_{3,4,5}^{(1)} \right]$ slots from the set $E^{(1)}_{3,4,5}$ are transferred to the set $E^{(1)}_{3,4,5}$ and $[p_2(1 - p_1) C_{3,4,5}^{(1)} + p_2 C_{3,4,5}^{(1)}]$ to the set $E^{(1)}_{3,4,5}$. Every set resulting from the 3-link available slot set decomposition related to clique 3 is then similarly updated.

More generally speaking, when processing the $i^{th}$ clique, the influence of allocations on the two previous links can be correctly considered by updating the sets resulting from its 3-link available slot set decomposition as follows:

$$C_i^{(i)} = C_i^{(i)} - p_i \times I_i + u_i + v_i$$

(12)

where

$$C_i^{(j)} = \begin{pmatrix} C_{i,i+1,i+2}^{(j)} & \cdots & C_{i,i+1,i+2}^{(j)} \\ C_{i,i+2,i+1}^{(j)} & \cdots & C_{i,i+2,i+1}^{(j)} \\ \vdots & \ddots & \vdots \\ C_{i,i+1,i+2}^{(j)} & \cdots & C_{i,i+1,i+2}^{(j)} \\ C_{i,i+2,i+1}^{(j)} & \cdots & C_{i,i+2,i+1}^{(j)} \\ C_{i,i+1,i+2}^{(j)} & \cdots & C_{i,i+1,i+2}^{(j)} \end{pmatrix}$$

$$P_i = \begin{pmatrix} p_{i-2} & p_{i-1} & \cdots & p_{i-1} + p_{i-2} - (1 - p_{i-2}) \end{pmatrix}$$

and

$$I_i = \begin{pmatrix} C_{i,i,i-2}^{(i-2)} & \cdots & C_{i,i,i-2}^{(i-2)} \\ C_{i,i-2,i-1}^{(i-2)} & \cdots & C_{i,i-2,i-1}^{(i-2)} \\ \vdots & \ddots & \vdots \\ C_{i,i,i-2}^{(i-2)} & \cdots & C_{i,i,i-2}^{(i-2)} \\ C_{i,i-2,i-1}^{(i-2)} & \cdots & C_{i,i-2,i-1}^{(i-2)} \\ C_{i,i,i-2}^{(i-2)} & \cdots & C_{i,i,i-2}^{(i-2)} \end{pmatrix}$$

(13)
The vector \( \mathbf{u}_i \) serves to compensate a set that does not suffer from all of the interferences. The vector \( \mathbf{v}_i \) is then used to update the sets receiving slots becoming unavailable for reservation on certain 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.

Once the clique sets are updated, the average values \( A^{(i)}_i, r_i, a_i, d_i \) and \( p_i = a_i/A^{(i)}_i \) can be correctly evaluated and the calculation process can be passed on to the following clique.

**Interference-Clique 4 and beyond:** When processing the third clique, the entries of matrix \( \mathbf{I}_i \) were strictly referring to sets that had not varied from the beginning of the estimation process. However, that is not the case when processing the fourth clique. Indeed, the corresponding sets are likely to have been impacted by allocations on previous links. Such a matrix \( \mathbf{I} \) correctly evaluated and the calculation process can be passed on to the following clique.

A straight solution would consist in forming the sets resulting from the 6-link available slot set decomposition and identify the way every set is impacted. This method is correct but leads to the previously mentioned domino effect. Fortunately, the random nature of the slot allocation can simplify the analysis and bound the number of variables to deal with for each clique process. In the following, we illustrate this point when processing any clique \( i \geq 4 \). We now show how to characterize a set used in the clique \( i \) set update equation, say \( E^{(i-2)}_{i-2,\overline{i-1},i,i+1,i+2} \), as a function of its initial state. We start by doing the 2-link available slot set decomposition related to \( l_{i-2} \) and \( l_{i-1} \). This decomposition leads to four disjointed sets: \( E^{(j)}_{i-2,\overline{i-1},i,i+1,i+2} \) and \( E^{(j)}_{i-2,i-1} \). Let us work on the third one, that is \( E^{(j)}_{i-2,\overline{i-1},i,i+1,i+2} \). This set can also be divided in eight disjointed subsets resulting from the 3-link available slot set decomposition of clique \( i \). This time, the slot space equals \( E^{(j)}_{i-2,\overline{i-1}} \) rather than \( \{1,2,\ldots,S,B\} \), leading to subsets of the following form \( E^{(j)}_{i-2,\overline{i-1},i,i+1,i+2} \), taken as an example. A property of the set \( E^{(j)}_{i-2,\overline{i-1}} \) is that along the estimation process, it can only transfer slots to the set \( E^{(j)}_{i-2,\overline{i-1}} \) and cannot receive slots from another. Therefore, the number of slots that initially belonged to the set \( E^{(j)}_{i-2,\overline{i-1}} \) and had become unavailable just before node \( n_{i-2} \) did its reservations for communication on link \( l_{i-2} \) equals \( C^{(1)}_{i-2,\overline{i-1}} - C^{(i-2)}_{i-2,\overline{i-1}} \). These slots had become 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 \( E^{(j)}_{i-2,\overline{i-1},i,i+1,i+2} \). We can thus represent the number of slots that had become unavailable in the set \( E^{(j)}_{i-2,\overline{i-1},i,i+1,i+2} \) with the discrete random variable \( X_{i-2,\overline{i-1},i,i+1,i+2} \) taking its values in the set \( \{0,\ldots,C^{(1)}_{i-2,\overline{i-1}} - C^{(i-2)}_{i-2,\overline{i-1}}\} \) and following an hypergeometric distribution with parameters \( (C^{(1)}_{i-2,\overline{i-1}}, C^{(1)}_{i-2,\overline{i-1}}, i+1, i+2, C^{(1)}_{i-2,\overline{i-1}} - C^{(i-2)}_{i-2,\overline{i-1}}) \). From this identification we can deduce that, for \( C^{(i-2)}_{i-2,\overline{i-1}} \) strictly positive, the average value of this random variable equals \( (C^{(1)}_{i-2,\overline{i-1}} - C^{(i-2)}_{i-2,\overline{i-1}})/C^{(1)}_{i-2,\overline{i-1}} \times C^{(1)}_{i-2,\overline{i-1},i,i+1,i+2} \).

More generally, just before node \( n_{i-2} \) did its allocations, an average proportion \( C^{(i-2)}_{i-2,\overline{i-1}}/C^{(1)}_{i-2,\overline{i-1}} \) of the initially available slots remained available in every subset partitioning \( E^{(j)}_{i-2,\overline{i-1},i,i+1,i+2} \). Hereafter, the quantity \( C^{(i-2)}_{i-2,\overline{i-1},i,i+1,i+2} \) can be correctly evaluated as follows:

\[
C^{(i-2)}_{i-2,\overline{i-1},i,i+1,i+2} = C^{(1)}_{i-2,\overline{i-1},i,i+1,i+2} \times \alpha_{i-2,\overline{i-1}}
\]
where the reduction factor of the set $E_{i-2,i-1}^{(j)}$ equals
\[
\alpha_{i-2,i-1} = \begin{cases} 
0 & \text{if } C_{i-2,i-1}^{(1)} = 0 \\
C_{i-2,i-1}^{(i-2)} & \text{else}
\end{cases}
\]
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 carried out for the two other sets of interest $E_{i-2,i-1}^{(j)}$ and $E_{i-2,i-1}^{(j)}$. However, it differs a little for $E_{i-2,i-1}^{(j)}$ when $C_{i-2,i-1}^{(1)} = 0$ as this set can receive slots from $E_{i-2,i-1}^{(j)}$ due to allocations on previous links. For this case, 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 correctly evaluated as follows:
\[
C_{i-2,i-1,i,i+1,i+2}^{(i-2)} = C_{i-2,i-1,i,i+1,i+2}^{(1)} \times \tau_{i-2}
\]
with
\[
\tau_{i-2} = \frac{C_{i-2,i-1}^{(i-2)}}{C_{i-2,i-1}^{(1)}} = \frac{C_{i-2,i-1,i}^{(i-2)} + C_{i-2,i-1,i}^{(i-2)}}{C_{i-2,i-1,i}^{(1)} + C_{i-2,i-1,i}^{(1)}}
\]
if $C_{i-2,i-1}^{(1)}$ is strictly positive, and zero otherwise.

Then, no additional techniques are required to process the remaining cliques and the calculation can be completed by simply applying the approach given in algorithm 2 when processing the last clique of the path. The main advantage of this approach is that there is no domino effect and the resulting calculation complexity is $O(N_H)$.

### 4 Distributed Implementation of BRAND

In Section 3 we presented a centralized version of BRAND. This version can be easily applied 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 in a distributed fashion by every node. To address this limitation of the centralized version we present a simple mechanism that enables the distributed execution of BRAND. The proposed mechanism makes possible the implementation of BRAND with any routing protocols, regardless of how they perform path computations.

BRAND estimates the available end-to-end bandwidth by estimating, for all possible demands, the average end-to-end throughput of a given path and returning the maximum such value. Next, we describe a distributed implementation of the throughput computation and in Section 4.2 we describe how this process can be repeated for all possible demands.

#### 4.1 Distributed Computation of the Average End-to-End Throughput

As described in Section 3, applying the interference-clique sliding technique for computing the average throughput of a given path given a demand, $d$, requires the knowledge of the interference
clique. In a distributed setting, this requirement means that every node executing the algorithm will need to know the next three hops on the particular path. Unfortunately, with protocols like OLSR [34] or AODV [35], nodes only store the next-hop for a given source-destination pair. To overcome this challenge, we propose a simple trick: Simply shift the interference-clique calculation two hops down the path. Doing so will require, for example, node $n_3$ to do the computation related to the first interference clique, consisting of $l_1, l_2, l_3$, that normally would be done by $n_1$. Regardless of the routing protocol, node $n_3$ will know $n_1, n_2$ and $n_4$ and, therefore, will have the necessary information to execute our algorithm.

The source node starts by sending a bandwidth calculation init packet to its next hop. This packet contains the destination node id, necessary information for performing the calculation for the first link (initial demand $d, \phi_1, f_1$) and a flow id used to identify the session for which the calculation is carried out. Node $n_2$ uses the destination id to identify the next hop, node $n_3$, and transfer the bandwidth calculation init packet to $n_3$ with additional necessary information for doing the calculation for the first clique ($\phi_2, f_2$). Node $n_3$ is then capable of identifying the next hop $n_4$ and does the computation related to the first interference-clique, $l_1, l_2, l_3$. Once this computation is done, $n_3$ sends a bandwidth calculation request packet to its next hop, $n_4$. This packet contains the destination node id, the flow id, $d_1, \phi_2, \phi_3, f_2, f_3, p_1$ and $S_1^{(1)11}$. This way, $n_4$ has sufficient information to do the computation for the second clique, $l_2, l_3, l_4$. Once this computation is done, $n_4$ follows the same procedure as $n_3$ and sends a bandwidth calculation request packet to its next hop. This packet contains the destination node id, the flow id, $d_2, \phi_3, \phi_4, f_3, f_4, p_1, p_2, S_1^{(1)}, S_2^{(1)}$, and the three reduction factors related to clique 2 with possibly $\tau_2$ that will serve in the computation for the fourth clique. With receiving this packet, $n_5$ can do the computation related to the third clique. Then, it sends a bandwidth calculation request packet to its next hop on the path. This packet contains the destination node id, the flow id, $d_3, \phi_4, \phi_5, f_4, f_5, p_2, p_3, S_1^{(1)}, S_2^{(1)}, S_3^{(1)}$ the three reduction factors related to clique 2 with possibly $\tau_2$ and the three reduction factors related to clique 3 with possibly $\tau_3$. Thus, $n_6$ has sufficient information to do the calculation related to clique 4.

This same process is repeated till reaching the last node before the destination, as doing the calculations for the following cliques requires no additional fields in the bandwidth calculation request packet. When the last node before the destination receives a bandwidth request packet, it knows it is processing the calculation for the last clique. Using the information received in the packet it updates the last clique sets’ information and applies the method provided in Algorithm 2 to compute the average end-to-end throughput. Once it finishes the calculation it sends a bandwidth calculation response packet back to the source with the resulting throughput value.

### 4.2 Looping over All Possible Demands

BRAND requires estimating the average end-to-end throughput for all possible demands. This can be done in two ways. One approach is for the source to initiate multiple bandwidth calculation procedures, one for every demand $d$ taken in the range $[0, \min (\phi_1, \phi_2)]$ with step $\Delta_\phi$. The step $\Delta_\phi$ can be selected by the source node and can be as simple as the smallest capacity it can allocate. Another approach is to do the calculation for all such demands at

---

11Every node is assumed to have 2-hop slot availability information.
once, at every hop, and then encapsulate the corresponding data related to each demand in the resulting request packets. In this case the size of the bandwidth calculation request packet is bigger but the end-to-end process is carried out once.

In terms of computation complexity, the number of operations for processing a clique is constant. Therefore, for every demand, the resulting complexity is $O(N_H)$. If the backward path of the response packet from the last node before the destination to the source is similar to the forwarding path of requests packets, the round trip time for the average end-to-end throughput calculation is bounded by $2T \times (N_H - 1)$ with $T$ equal to the TDMA frame duration.

5 Performance evaluation of Brand

In this section, we evaluate BRAND numerically using MATLAB and compare it with the work of Zhu et al. [23]. In summary, we make the following main observations:

- In Section 5.2, we demonstrate that although BRAND uses a very simple scheduling heuristic — randomized scheduling — its achieved end-to-end bandwidth is competitive when compared with the optimal policy.

- In Section 5.3, we demonstrate the correctness of BRAND’s algorithm for computing the average throughput of randomized scheduling.

- In Section 5.4, we demonstrate that BRAND delivers the bandwidth it promises in achieving an almost perfect admission control.

- In Section 5.5, we demonstrate that BRAND deals successfully with the challenges arising from the cognitive radio network architecture. Specifically, BRAND maintains an almost perfect admission control in face of Primary User interference and multi-rate links.

- Also in Section 5.5, in comparing with a heuristic not designed for cognitive radio networks, we demonstrate the significant effect multi-rate links and the Primary User interference can have on the available end-to-end bandwidth and ultimately, the admission control.

5.1 Simulation Parameters

Each link is assigned one orthogonal channel among four sensed ones. The medium is accessed through a TDMA MAC with 40 slots per frame. 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 $P[\text{channel4}] = 0.05$. The probabilities are purposely selected to achieve a high self-interference on the path and thus, to show the worst-case performance of the algorithm. The scheduling algorithm in the simulation is the same with the one used by BRAND, i.e., the necessary allocations of slots are performed at random among those available.

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: $\mu_\phi(\text{channel}1) = 2$Mbps, $\mu_\phi(\text{channel}2) = 1.5$Mbps, $\mu_\phi(\text{channel}3) = 800$kbps and $\mu_\phi(\text{channel}4) = 250$kbps. According to the corresponding cumulative distribution function, the generated transmission rates for $\mu_\phi = 2000$kbps and $\sigma_\phi = 400$kbps oscillate between 1000kbps and 3000kbps.

As described in Section 2.3, there are two sources of interference in a cognitive radio networks: Secondary-to-secondary and primary-to-secondary. To model the interference from other secondary sessions we use a probability, $p_a$, for denoting the chances of a given slot being free of other secondary communications. The primary interference is modeled by the probability $u$ introduced in Section 2.3.2.

**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 $I_d = [0, \min (\phi_1, \phi_2, ..., \phi_{NH})]$ (kbps) with step $\Delta_\phi = 10$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 the work by Zhu et al [23], which computes the available bandwidth for legacy TDMA multi-hop network. We use the Zhu heuristic as basis for parts of our evaluation while taking into account the fact that it is not designed for cognitive radio networks.

### 5.2 End-to-End Bandwidth with BRAND

With BRAND, new demands are allocated by selecting at random among the slots available. This scheduling algorithm is extremely simple and lands itself to practical implementations and easy adaptations to new architectures. However, the question is, how good is the randomized scheduling when compared to the optimal scheduling policy. To answer this question we perform
the following experiment. 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 in MATLAB using these PSAT tables and compute the available end-to-end bandwidth for different values of primary user interference. Given that computing the optimal slot scheduling is NP-Complete, we formulate the problem as an integer linear program, see Appendix D, and use lpsolve [36] to generate optimal scheduling assignments for a given PSAT table of a given path. The optimal schedules computed for the same PSAT tables as the ones used for BRAND are then fed into MATLAB for computing the resulting available end-to-end bandwidth. As evident by the formulation shown in the Appendix D the integer linear program gets highly complicated for high number of hops so we were able to compute optimal values for paths of up to four hops.

Figure 4 shows the optimal values as well as the ones computed by BRAND for the available end-to-end bandwidth of a 4-hop path as a function of primary user occurrence probability. As the data shows, BRAND is within 85% of the optimal policy despite using a very simple scheduling heuristic.

5.3 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 3.3. 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 $p_a$ 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
Figure 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.

5.4 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 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
Figure 7: The presence of multi-link rates 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.

Figure 8: BRAND takes into account the Primary User which leads to almost 100% admission control rate. Ignoring the Primary User leads to significant errors in estimating the available bandwidth. Protecting the primary comes at a cost, as predicted by Equation 1 in Section 2.3.2, which explain the smaller bandwidth values computed by BRAND.

computed demand. To provide more data about the behavior of BRAND we repeat the same experiment for a 10-hop path, which is as long a path as one can be expected to encounter in deployed multi-hop cognitive radio networks. As shown in Figure 8, 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.

5.5 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. We have identified two defining features in this architecture: The primary-secondary hierarchy and multi-rate links. We first evaluate the joint effect of these features and then evaluate them separately to shed light on the relative impact each of them can have.
(a) Multiple Transceivers, Multiple Rates, \(u = 0\%\), (b) Multiple Transceivers, Multiple Rates, \(u = 0\%\), \(N_H = 4\) \(N_H = 10\)

Figure 9: BRAND always computes accurate values for the available end-to-end bandwidth leading to almost 100% admission rate. Ignoring the presence of multi-rate links can lead to significant errors in estimating the available bandwidth.

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 to quantify the consequences of ignoring the primary user and the multi-rate links when computing the available bandwidth.

In this part of the evaluation we carry the same basic experiment as in Section 5.4 but depending on what specifically we evaluate we change the number of transceivers, link-rates and/or the probability of Primary User occurrence.

5.5.1 Combined Effect of Multi-Rate Links and Primary User Activity

To evaluate the overall "cognitive" effect, we repeat the experiment of the previous section using multi-transceivers, multiple-rates and varying probabilities of the Primary User occurrence. As depicted in Figure 7 BRAND almost always estimates the correct value for the available capacity. On the other hand, ignoring the operating specifics of cognitive networks leads to significant errors, as high as 900%, when calculating the available end-to-end bandwidth.

5.5.2 Effect of Primary User Activity

To isolate the effect of the Primary User we repeat the previous experiment but this time we use a single transceiver on every node and a single rate on all links and vary the probability of Primary User occurrence. As seen in Figure 8 ignoring the Primary User can lead to significantly overestimating the available bandwidth. At the same time, as predicted by Equation 1 in Section 2.3.2 protecting the Primary User comes at a cost in terms of bandwidth. This cost, which is higher the higher the probability of Primary User occurrence, \(u\), is, explains the smaller bandwidth values computed by BRAND when compared to a heuristic that completely ignore the Primary User.
5.5.3 Effect of Multi-Rate Links

Finally, for evaluating the impact of the multi-link rates alone, we set the probability of Primary User occurrence to zero and re-run the previous experiment. This time, since the experimental space is smaller we include results for a 10-hop path as well. The data in Figure 9 shows that BRAND almost always delivers 100% admission rate and that ignoring multi-rate link can lead to significant errors when computing the available bandwidth, even more so that when ignoring the Primary User. In addition, the data shows BRAND computing higher available bandwidth values while maintaining almost 100% accuracy. The reason for the ZHU heuristic underperforming BRAND in this experiment is that it allocates a constant number of slots end-to-end, assuming that all links on the path have the same rate as the first one. The assumption can lead it to miscalculate the path’s available bandwidth and, most importantly, makes it very hard for adapting the heuristic to take multi-rate links into account.

The data from this and the previous experiments show the importance of designing a heuristic for computing the available bandwidth around the particularities of the cognitive radio.

6 Related work

Multi-hop cognitive radio networks share similarities with the traditional multi-hop networks, chiefly among them the wireless interference. At the same time, cognitive radio networks present unique challenges, most importantly the presence of the so-called primary users and a far more rich and dynamic selection of channels. This has resulted in new architectures and protocols being proposed for cognitive radio networks. Nevertheless, considering the similarities between the two architectures a lot of the research results for traditional multi-hop wireless networks are relevant and in many cases have been at the basis of the newly proposed protocols and architectures. For reasons of completeness, 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[12]. At the routing layer, a node is required to adapt its path computations according to the primary users 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[37]. Obviously, both strategies will increase the end-to-end delay. In [38], a geographic routing solution is proposed that selects next hops and operating channels so as to avoid regions of primary users activity while minimizing the end-to-end path latency. Joint route selection and spectrum decision is also addressed in [26, 14]. In [26] the authors propose establishing a spectrum tree on each frequency channel and selecting routes according to a newly defined routing metric. While many solutions make use of a common control channel (CCC), an adaptation of the AODV protocol free of a CCC is proposed in [39]. An optimal routing metric for multi-hop cognitive radio networks is proposed in [14]. The authors analytically demonstrates 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 [40, 41, 42] wherein algorithms for joint routing, link
scheduling and spectrum assignment algorithms have been studied. In [40] an opportunistic scheduling that maximizes the overall capacity of secondary users while satisfying a constraint on time average collision rate at the primary users is proposed. In [41] the joint routing and link scheduling problem with uncertain spectrum supply is investigated. The authors in [42] address the problem of minimizing the total transmission latency. Finally, [43] has proposed a distributed algorithm for jointly optimizing routing, scheduling, spectrum allocation and transmit power. Nevertheless, the problem of computing the end-to-end bandwidth of a multihop 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, have been subject of significant research efforts and an exhaustive survey is beyond the scope of this paper. Instead, here we simply summarize a subset of the published works that is 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 [23] and the references therein. In [23], 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. Intuitively speaking, the problem is hard because computing the residual end-to-end bandwidth is coupled with the problem of per-link slot assignments. With the problem being 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 show in Section 5, cannot be readily applied to a cognitive radio architecture. In [44], the authors 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, [45] provides a distributed, online and provably efficient algorithm for joint routing, channel assignment and scheduling in multi-hop multi-radio ad hoc networks. However, unlike these works, we do not propose a new routing scheme and instead focus on solving the problem of admission control once the paths are computed. The advantage of this approach is that it allows for a solution that can be adopted by already established routing protocols.

In [46] the authors 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 [47]. In [24] the capacity region of multi-radio multi-channel wireless networks has been studied by introducing a multi-dimensional conflict graph characterizing the interferences between adjacent (radio, link, channel)-tuples. An admission control scheme is provided by deriving a set of local sufficient conditions for flow feasibility in such networks. Indeed, a simple condition based on the neighborhood constraint is first proposed, before a more accurate one is derived by adapting the scaled clique constraint introduced in [48]. 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 [46, 47], 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.

7 Concluding Remarks

In this paper, we have revisited the problem of admission control for the radio cognitive context. Our solution, BRAND, uses a linear time algorithm for estimating the available end-to-end throughput in TDMA-based multi-hop cognitive radio networks wherein each node is equipped with multiple transceivers. We have addressed the particular case of random slots selection at the MAC layer and provided an admission control scheme for end-to-end flows. Our method is based on the introduced l-link available slot set decomposition and the clique sliding approach, an approximation scheme that reduces the calculation complexity from exponential to linear while still returning accurate and reliable results. We have shown that BRAND can run in a distributed fashion. This, coupled with our choice of computing the available bandwidth of a path, makes BRAND a feasible proposition for any routing approach. Using a thorough numerical analysis we have demonstrated the correctness of BRAND as well as its capability in accurately taking into account the cognitive radio context, in particular the Primary User and multi-rate links, and thus delivering almost 100% admission rate in a variety of conditions.

As future work we intend to explore the problem of admission control in multi-hop cognitive radio networks from a different perspective. That is, instead of being provided the path to the destination and computing its available bandwidth, we want to explore the problem of computing the path that currently has the maximum available bandwidth among all paths to the destination. As we remarked in Section 3, it is not clear that doing so would lead to better overall performance but nevertheless it is a path worth exploring.
A  Update formula parameters

Taking into account the way the available slots in every sets of clique $i$ are impacted by the slot allocation process occurring on the two previous links, leads to the following eight equations:

$$C_{i,i+1,i+2}^{(i)} = C_{i,i+1,i+2}^{(1)} - [p_{i-2} + p_{i-1}(1 - p_{i-2})] \cdot C_{i-2,i-1,i,i+1,i+2}^{(i-2)}$$

- $p_{i-2} \cdot C_{i-2,i-1,i,i+1,i+2}^{(i-2)} - p_{i-1} \cdot C_{i-2,i-1,i,i+1,i+2}^{(i-1)}$

transferred to set $E_{i,i+1,i+2}^{(i-2)}$.

$$C_{i,i+1,i+2}^{(i)} = C_{i,i+1,i+2}^{(1)} - [p_{i-2} + p_{i-1}(1 - p_{i-2})] \cdot C_{i-2,i-1,i,i+1,i+2}^{(i-2)}$$

- $p_{i-2} \cdot C_{i-2,i-1,i,i+1,i+2}^{(i-2)} - p_{i-1} \cdot C_{i-2,i-1,i,i+1,i+2}^{(i-1)}$

$$C_{i,i+1,i+2}^{(i)} = C_{i,i+1,i+2}^{(1)} - [p_{i-2} + p_{i-1}(1 - p_{i-2})] \cdot C_{i-2,i-1,i,i+1,i+2}^{(i-2)}$$

- $p_{i-2} \cdot C_{i-2,i-1,i,i+1,i+2}^{(i-2)} - p_{i-1} \cdot C_{i-2,i-1,i,i+1,i+2}^{(i-1)}$

$$C_{i,i+1,i+2}^{(i)} = C_{i,i+1,i+2}^{(1)} - p_{i-1}(1 - p_{i-2}) \cdot C_{i-2,i-1,i,i+1,i+2}^{(i-2)}$$

- $p_{i-1} \cdot C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + p_{i-2} \cdot [C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + C_{i-2,i-1,i,i+1,i+2}^{(i-1)}]$

$$C_{i,i+1,i+2}^{(i)} = C_{i,i+1,i+2}^{(1)} - p_{i-1}(1 - p_{i-2}) \cdot C_{i-2,i-1,i,i+1,i+2}^{(i-2)}$$

- $p_{i-1} \cdot C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + p_{i-2} \cdot [C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + C_{i-2,i-1,i,i+1,i+2}^{(i-1)}]$

$$C_{i,i+1,i+2}^{(i)} = C_{i,i+1,i+2}^{(1)} + p_{i-1}(1 - p_{i-2}) \cdot [C_{i-2,i-1,i,i+1,i+2}^{(i-2)} +$$

- $C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + C_{i-2,i-1,i,i+1,i+2}^{(i-1)}$

+ $p_{i-2} \cdot [C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + C_{i-2,i-1,i,i+1,i+2}^{(i-1)}]$

+ $p_{i-1} \cdot [C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + C_{i-2,i-1,i,i+1,i+2}^{(i-1)}]$

$$C_{i,i+1,i+2}^{(i)} = C_{i,i+1,i+2}^{(1)} + p_{i-1}(1 - p_{i-2}) \cdot [C_{i-2,i-1,i,i+1,i+2}^{(i-2)} +$$

- $C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + C_{i-2,i-1,i,i+1,i+2}^{(i-1)}$
As shown in Section 3, all the required parameters can be derived before computing these equations. Then, it consists of applying algorithm 2 when processing the last clique of the path to compute the realized end-to-end throughput. Introducing vectors $u_i$ and $v_i$ with non-zero values defined as follows leads to the expression of Eq. 12.

$$u_1 = p_{i-2}.C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + p_{i-1}.C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + [p_{i-2} + p_{i-1}.(1 - p_{i-2})].C_{i-2,i-1,i,i+1,i+2}^{(i-2)}$$

$$u_3 = p_{i-2}.C_{i-2,i-1,i+1,i,i+2}^{(i-2)} + C_{i-2,i-1,i+1,i+2}^{(i-2)}$$

$$u_4 = p_{i-2}.C_{i-2,i-1,i+1,i+2}^{(i-2)} + [p_{i-2} + p_{i-1}.(1 - p_{i-2})].C_{i-2,i-1,i+1,i+2}^{(i-2)}$$

$$u_7 = p_{i-2}.C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + [p_{i-2} + p_{i-1}.(1 - p_{i-2})].C_{i-2,i-1,i,i+1,i+2}^{(i-2)}$$

$$v_1 = p_{i-1}.C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + C_{i-2,i-1,i,i+1,i+2}^{(i-2)}$$

$$v_3 = p_{i-2}.C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + [p_{i-2} + p_{i-1}.(1 - p_{i-2})].C_{i-2,i-1,i,i+1,i+2}^{(i-2)}$$

$$v_4 = p_{i-1}.C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + C_{i-2,i-1,i,i+1,i+2}^{(i-2)}$$

$$v_7 = p_{i-2}.C_{i-2,i-1,i,i+1,i+2}^{(i-2)} + C_{i-2,i-1,i,i+1,i+2}^{(i-2)}$$

### B Equation 1 derivation

The PU’s channel state can be modeled by an alternative ON/OFF process [27, 28, 29]. 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. [28, 29] have focused on computing the quantity of PU-SU interference accumulated during the
Slot access is granted to a secondary user if no primary user is detected during the sensing period. However, sensing does not prevent a primary user to appear during the access period, making the secondary user QoS reduce due to the PU-SU interference.

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 $T_{access}$ 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 $P[occurrence of the OFF state] = 1 - u_l$, we get:

$$I_{l0}(T_{access}) = u_l T_{access} - u_l^2 \times \frac{1}{\lambda_{off}} \times \left[1 - e^{-T_{access}\lambda_{off}/u_l}\right]$$ (18)

and

$$I_{l1}(T_{access}) = u_l T_{access} + (1 - u_l)^2 \times \frac{1}{\lambda_{on}} \times \left[1 - e^{-T_{access}\lambda_{off}/u_l}\right]$$ (19)

Although the sensing may declare the primary channel idle, it is still possible that PU’s communications begin just before the sensing period ends. It is what typically happens during slot 1 in the example illustrated in Figure 10. 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:

$$Q_l = \frac{(1 - u_l) \times I_{l0}(T_{access}) + u_l \times I_{l1}(T_{access})}{T_{access}}$$ (20)

where $T_{access}$ corresponds to the slot access time. Substituting Equations 18 and 19 in Equation 20 leads to $Q_l = u_l$ and $\eta_l = (1 - u_l)^2$.

C  R-BRAND

In the following, we describe in detail the work done and R-BRAND, the new version of BRAND tailored for real-time applications. To take into account the delays constraints of real-time flows, R-BRAND executes a 4-step approach as follows:

![Figure 10: Slot access is granted to a secondary user if no primary user is detected during the sensing period. However, sensing does not prevent a primary user to appear during the access period, making the secondary user QoS reduce due to the PU-SU interference.](image-url)
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 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.

C.1 System description/model

To estimate the end-to-end delay experienced by packets of the newly admitted flows, we adopt a queuing system approach and modeled a path as a set of \(N\) queues with finite buffer sizes disposed in tandem, as illustrated in Fig. 11.

In this figure, the red points represent packets belonging to the RTP flows to be admitted. Each queue comprises a finite size buffer and serves the packets using a first-come-first-served policy (FCFS). A service corresponds to a transmission occurring over a link \(i\) at rate \(\mu_i\) packets per second. As the link transmissions are carried out through a TDMA MAC, the server is either active or in vacation, depending on the availability of allocated slots for the link. Given that \(a_i\) slots would be allocated on the \(i^{th}\) link (this quantity is evaluated during the BRAND execution), the probability of server \(i\) being active at an arbitrary time is \(f_i = a_i/S\), with \(S\) being the TDMA frame size. At the end of an allocated time-slot, the probability that the server start a vacation is \((1 - f_i)\). The average duration of a vacation period, also referred to as \(1/\alpha_i\), is greater that \(1/\mu_i\) and depends on the probability \(f_i\). In particular, we compute its value as follows, given that a vacation period cannot last more than the cumulated durations of the non allocated slots in the TDMA frame:

\[
\frac{1}{\alpha_i} = \frac{1}{\mu_i} + \frac{1 - f_i}{\mu_i} \times \sum_{k=1}^{S(1-f_i)} k f_i (1 - f_i)^{k-1}
\] (21)

The \(i^{th}\) queue blocking probability, denoted as \(B_i\), depends on the values of \(\mu_i\) as well the traffic intensity entering the queue, denoted as \(\lambda_i\). Given its value, the traffic intensity entering the following queue equals \(\lambda_{i+1} = (1 - B_i)\lambda_i\).

Respectively denoting by \(Q_i\) and \(R_i\) the average number of packets stored in queue \(i\) and the average delay of packets traversing that queue, the Little’s law gives:

\[
R_i = \frac{Q_i}{(1 - B_i)\lambda_i}
\] (22)

We eventually compute \(R\), the average end-to-end delay, as the sum of the values \(R_i\) for all the queues composing the path.
C.2 Decomposition in independent continuous time Markov chains

To find out the exact solution of such queuing system, one can resort to techniques based on the analysis of the queues at particular time instant through embedded Markov chains, as done for instance for the study of M/G/1/K queues. These techniques have been extensively used for the analysis of polling systems with finite buffers size, similar to TDMA access modeling. However, they require complex arithmetic operations such as Laplace-Stieltjes transforms that may not scale with the number of queues or with their capacity.

To provide a fast and scalable estimation, we analyze each queue independently from the others and derive its performance by solving the continuous time Markov chain presented in Fig. 12. In this Markov chain, the numerical values written in every state correspond to the number of RTP packets stored in the queue. State \((k, A)\) indicates that the server is active, meaning that the current time slot is allocated for transmission over the link, while there is \(k\) RTP packets waiting for transmission. State \((k, V)\) indicates that the server is in vacation, meaning that the current time slot is not allocated for transmission over the link. In this approach, a maximum of one packet is served per time-slot.

This Markov chain can be solved in time \(O(K)\) by applying the following equations, derived from the frontier equations obtained with simple cuts in the graph:

\[
\rho_i = \frac{\lambda_i}{\mu_i} \tag{23}
\]

\[
\pi_{0,A} = \frac{f_i \rho_i + \alpha_i/\mu_i}{(1-f_i)(1+\rho_i)} \pi_{0,V} \tag{24}
\]

\[
\pi_{k+1,A} = \rho_i [\pi_{k,A} + \pi_{k,V}], \quad \forall k \in [0, N-1] \tag{25}
\]

\[
\pi_{k,V} = \frac{\rho_i}{f_i \rho_i + \alpha_i/\mu_i} \times [\pi_{k-1,V} + (1-f_i)\pi_{k,A}], \quad \forall k \in [1, N-1] \tag{26}
\]

\[
\pi_{K,V} = \frac{\lambda_i}{\alpha_i} \pi_{K-1,V} \tag{27}
\]

After normalizing the queue stationary state probabilities, we can compute the performance parameters of the queue as follows:

\[
Q_i = \sum_{i=0}^{K} k_i (\pi_{k,A} + \pi_{k,V})
\]
\[ B_i = \pi_{K,A} + \pi_{K,V} \]

using the PASTA property of Poisson arrival processes.

Note that to gain scalability, our modeling approach relies on a series of approximations. First, it relaxes the constraint of constant slots duration. Then, it assumes Poisson packet arrival processes at each queue.

### C.3 Performance results

To validate the introduced delay estimation approach, we conducted extensive simulations following the steps described in the R-BRAND algorithm. The analysis is performed on 64kbps voice call flows with RTP packets sent once every 20ms. To this end, we generated numerous RSAT tables with different slot availability probabilities and compared the estimation given by R-BRAND with the end-to-end delay of packets averaged over 5 simulation runs of duration 20 seconds. For the sake of clarity but not loss of generality, we carried 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 channels with data rates almost equaling 2Mbps.

Figure 13 presents 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 \( \geq 95\% \)). We can see that the proposed model globally follows the variations of the delay due to the amount of admitted flows. As we can see on the figure, the average end-to-end delay decreases significantly (\( \geq 15\% \)) when the number of admitted flows grows. The differences between the model and the simulation data are mainly due to the modeling assumptions.

---

**Figure 13**: Achieved end-to-end delay as a function of the number of admitted flows.
D Integer program formulation for the case of 4-hop paths

Setting up the following variables

\[ x = (x_1, x_2, \ldots, x_{23})^T \]
\[ b = \begin{pmatrix}
C_{1,1,2,3,4}^{(1)} + C_{1,1,2,3,4}^{(1)} \\
C_{1,2,3,4}^{(1)} \\
C_{1,1,2,3,4}^{(1)} \\
C_{1,2,3,4}^{(1)} + C_{1,1,2,3,4}^{(1)}
\end{pmatrix} \]
\[ A = \begin{pmatrix}
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
\end{pmatrix} \]

and

\[ P = \begin{pmatrix}
\phi_1 f_1/S & 0 & 0 & 0 \\
0 & \phi_2 f_2/S & 0 & 0 \\
0 & 0 & \phi_3 f_3/S & 0 \\
0 & 0 & 0 & \phi_4 f_4/S \\
\end{pmatrix} \]

we have used lpSolve 5.5 to realize the following objective and evaluate the maximum available bandwidth provided by an optimal slot allocation process

\[
\text{maximize } \min \{ \mathbf{P}, \{ \mathbf{A} \mathbf{x} + \mathbf{b} \} \}
\]
subject to

\[
x_1 + x_2 \leq C_{1,1,2,3,4}^{(1)},
\]
\[
x_3 + x_4 \leq C_{1,1,2,3,4}^{(1)},
\]
\[
x_5 + x_6 \leq C_{1,1,2,3,4}^{(1)},
\]
\[
x_7 + x_8 \leq C_{1,1,2,3,4}^{(1)},
\]
\[
x_9 + x_{10} \leq C_{1,1,2,3,4}^{(1)},
\]
\[
x_{11} + x_{12} + x_{13} \leq C_{1,1,2,3,4}^{(1)},
\]
\[
x_{14} + x_{15} \leq C_{1,1,2,3,4}^{(1)},
\]
\[
x_{16} + x_{17} \leq C_{1,1,2,3,4}^{(1)},
\]
\[
x_{18} + x_{19} + x_{20} \leq C_{1,1,2,3,4}^{(1)},
\]
\[
x_{21} + x_{22} + x_{23} \leq C_{1,1,2,3,4}^{(1)},
\]
\[
x_i \text{ integer and positive.}
\]
References


