A Correlation-aware Splitting Algorithm for Opportunistic Selection

Reneeta Sara Isaac, Neelesh B. Mehta, *Senior Member, IEEE*

***Abstract*—Opportunistic selection is a key technique to improve the performance of wireless systems. In it, one among the available users is selected on the basis of their channel gains or local parameters such as battery energy state. Formally, each user possesses a real-valued metric that only it knows, and the goal is to select the best user, which has the highest metric. The splitting algorithm is a popular, fast, and scalable algorithm to implement opportunistic selection; it is distributed and guarantees selection of the best user. We show that this algorithm, which has thus far been designed assuming that the metrics are independent and identically distributed, is no longer scalable when the metrics are correlated. We then propose a novel correlation-aware splitting algorithm (CASA) and show how it can be applied to practically motivated probability distributions and correlation models. We present computationally feasible techniques for pre-computing the thresholds that CASA specifies, thereby ensuring that CASA can be implemented in practice. We benchmark the performance of CASA with the conventional algorithm, and show that it reduces the average selection time significantly as the number of users or the correlation among them increases.**

***Index Terms*—Opportunistic selection, splitting algorithm, correlation, multiple access protocols.**

I. Introduction

Opportunistic selection is a key technique that exploits spatial or multi-user diversity to improve the performance of many different wireless systems [1]–[8]. In it, one user is selected from a set of available users based on their instantaneous channel conditions, measurements, or battery energy states. For example, in opportunistic scheduling in cellular systems, the base station selects or schedules users based on their instantaneous channel conditions for downlink transmission [1, Chap. 14]. A relay-aided cooperative communication system exploits spatial diversity by opportunistically selecting one relay from among a set of geographically separated relays based on their instantaneous channel conditions to forward data from a source to a destination [2], [3]. In ad hoc wireless networks, opportunistic routing schemes select a group of next-hop forwarders dynamically at each hop based on their instantaneous link qualities [4], [5]. In wireless sensor networks (WSNs), selecting the sensor that transmits based on its measurement or battery energy state increases the network lifetime [6]–[10].

In all the above systems, the process of selection can be formally stated as follows. Each user possesses a real-valued metric that quantifies how useful the user would be if selected. The best user is defined as the one with the highest metric, and is the one that is selected. For example, in opportunistic scheduling, the metric of a user is its downlink signal-to-noise ratio (SNR) [1, Chap. 14]. Fairness can also be incorporated in this framework by suitably defining the metric [11]. In amplify-and-forward cooperative relaying, the metric of a relay is the harmonic mean of its source-relay (SR) and relay-destination (RD) channel power gains [2]. In WSNs, the metric of a sensor is its measurement, remaining battery energy, or channel gain to the fusion node [6]–[9].

While opportunistic selection is appealing, a key challenge in it is that the users are geographically separated from each other, and the metric of a user is known only to it. Hence, a selection algorithm to discover the best user, which no user knows a priori, is essential. A simple and popular example of a selection algorithm is polling, in which a controller sequentially receives the metric from each user and then selects the best one. However, polling is not scalable since its selection time increases linearly with the number of users. This also makes the system sensitive to Doppler spread. Distributed selection algorithms, such as the timer algorithm [2], [12] and the splitting algorithm, see [10], [13]–[18] and the references therein, solve this problem.

The splitting algorithm traces its roots to the first-come first-serve (FCFS) medium access control (MAC) protocol [19, Chap. 4]. It is a time-slotted algorithm in which each user locally decides to transmit in a slot to a coordinating user called sink if its metric lies between two thresholds. At the end of each slot, the sink feeds back to all the users whether an idle (no user transmitted), a success (one user transmitted), or a collision (multiple users transmitted) event occurred in that slot. On receiving an idle or a collision feedback, the thresholds are updated at each user and the algorithm continues. A success feedback implies that the best user has been selected; hence, the algorithm terminates. The metrics are assumed to remain constant over the time slots required to select the best user.

Two properties of the splitting algorithm make it stand out. First, it is guaranteed to select the best user, unlike the timer algorithm. Second, it is remarkably fast and scalable as it can select the best user in 2.467 slots, on average, even when the number of users tends to infinity [14]. The average selection time is an important performance measure of a selection algorithm because reducing it implies that more time is available for data transmission and the system is less sensitive to Doppler spread. We focus on this algorithm in this paper.

Manuscript submitted April 24, 2017; revised July 17, 2017; accepted October 31, 2017. The editor coordinating the review of this paper and approving this for publication was Hong-Chuan Yang.

R. S. Isaac and N. B. Mehta are with the Dept. of Electrical Communication Eng. (ECE), Indian Institute of Science (IISc), Bangalore, India (Emails: reneetaisaac@gmail.com, neeleshbmehta@gmail.com).

This work was partially supported by the DST-Swaranajayanti Fellowship award DST/SJF/ETA-01/2014-15 and the Qualcomm Innovation Fellowship, India.

DOI: xxx

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2017.2772266, IEEE Transactions on Communications

The rules used to update the thresholds are key to ensuring the splitting algorithm's excellent performance. In [13], [14], they depend on the phase the algorithm operates in. The algorithm begins with a pre-collision phase and stays in it until a collision occurs, in which case it switches to a post-collision phase. The thresholds are updated at each user depending on the feedback of the previous slot and the phase. In the pre-collision phase, both thresholds are lowered so that one user, on average, transmits in the next slot. Whereas, in the post-collision phase, they are updated so that, on average, half of the users involved in the last collision transmit. The threshold-update rules in the most-informative-first-serve splitting algorithm proposed in [10] for decentralized detection in WSNs exploit, in addition, the statistical information of the sensor data.

A critical and common assumption that has been made in the design of the conventional splitting algorithm in [13], [14], [19] is that the metrics are independent and identically distributed (i.i.d.). We shall, therefore, refer to the above algorithm as the i.i.d. based splitting algorithm (IIDSA). Unfortunately, this i.i.d. assumption is not valid in several practical systems and scenarios of interest, as we discuss below.

Practical scenarios with correlated metrics: In a WSN, the metric of a sensor node can be the observation itself or some function of it. For example, in [7], [8], the metric of a node is the absolute value of its log-likelihood ratio (LLR), which is a function of the observation. The observations recorded by the sensors, such as the temperature or any other physical phenomenon, can be correlated [9], [20].

In a cooperative relay system, the metrics of the relays are their SR or RD SNRs, which are exponential random variables (RVs) due to Rayleigh fading. The SR (and RD) SNRs can be correlated if the relays are relatively close to each other, as can happen in a relay cluster, or due to common scatterers [21]. In [22], [23], the metrics are the instantaneous SNRs of the channels, and are assumed to be equally correlated [24].

Another example is lognormal shadowing in cellular systems, indoor WLANs [25], and other multi-hop networks [26], [27]. Here, the metric of a user is its SNR, which is proportional to the shadowing. The shadowing of the links of different users can be correlated since some obstructions are common across users [28, Chap. 20]. The correlation coefficient between the SNRs can be as high as 0.95 [25].

A. Contributions

We make the following contributions in this paper:

1. *Performance of IIDSA:* In the presence of correlation, we show that IIDSA performs poorly and is no longer as scalable. Specifically, its average selection time increases substantially as the correlation and the number of users increase. Intuitively, the reason behind this is that the average number of users that are involved in a collision increases as the correlation among the users increases. This leads to more slots being required to select the best user in the post-collision phase.

2. *Redesigned splitting algorithm:* We propose a novel correlation-aware splitting algorithm (CASA) for selecting the best user when the metrics are correlated and non-identical. While correlation has been considered previously in the design of MAC protocols [9], to the best of our knowledge, CASA is the first splitting algorithm designed for correlated metrics. We propose a new and common design rule that specifies the thresholds in each slot as solutions of equations involving the joint cumulative distribution function (CDF) of the metrics. This captures their non-identical behavior as well as correlation. As per this rule, the thresholds are chosen in both pre-collision and post-collision phases such that q users, on average, transmit in each slot. Here, q is a parameter that we shall optimize.[1] We shall see that computing the thresholds in CASA requires a more sophisticated approach than for the i.i.d. case. However, the thresholds can be pre-computed just once and stored in the users when the system commences operation.

3. *Instantiation of CASA using computationally efficient approaches:* We show how CASA can be applied to the following two probability distribution models for the metrics: (i) exponential RVs with a generalized correlation structure, in which the correlation coefficient between any two RVs is given by $\rho_{kj} = \lambda_k^2 \lambda_j^2$, for $k \neq j$ [29]. This includes as a special case the constant correlation model [24], in which any two metrics have a correlation coefficient ρ; and (ii) lognormal RVs with an arbitrary covariance matrix. For both models, we discuss computationally efficient methods to evaluate the joint CDFs of the metrics and, therefore, the thresholds. We also propose a novel *hybrid approach* that reduces the computational and storage burden further by leveraging the simplicity of IIDSA once the algorithm has run for sufficiently many slots.

4. *Analysis of selection time and performance benchmarking:* We derive an expression for the CDF of the time taken in slots to select the best user by CASA. From this, performance measures, such as mean and variance of the selection time, can be directly computed. Our benchmarking results show that in the presence of correlation, the average selection time of CASA is much less than that of IIDSA and the polling algorithm. The higher the correlation among the metrics, the more significant is the reduction. We observe that as the number of users increases, the average selection time of CASA increases at a rate smaller than that of IIDSA.

B. Organization and Notations

The paper is organized as follows. Section II describes the general form of the splitting algorithm and the design of CASA. Section III presents two different models to which CASA applies, techniques to efficiently pre-compute its thresholds, and the analysis of its selection time. Section IV presents the simulation results and performance comparisons, and is followed by our conclusions in Section V.

Notations: The probability of an event A is denoted by $\Pr(A)$ and the probability of A conditioned on event B is denoted by $\Pr(A|B)$. The indicator function $1_{\{A\}}$ equals 1 if A occurs and is 0 otherwise. The expectation of an RV X is denoted by $\mathbb{E}[X]$ and the expectation of X conditioned on A is denoted by $\mathbb{E}[X|A]$. The covariance of two

[1] Although a similar approach was followed in the pre-collision phase of IIDSA in [14], our approach is different as we use the same rule in the post-collision phase as well and also because we account for the correlation among the metrics.

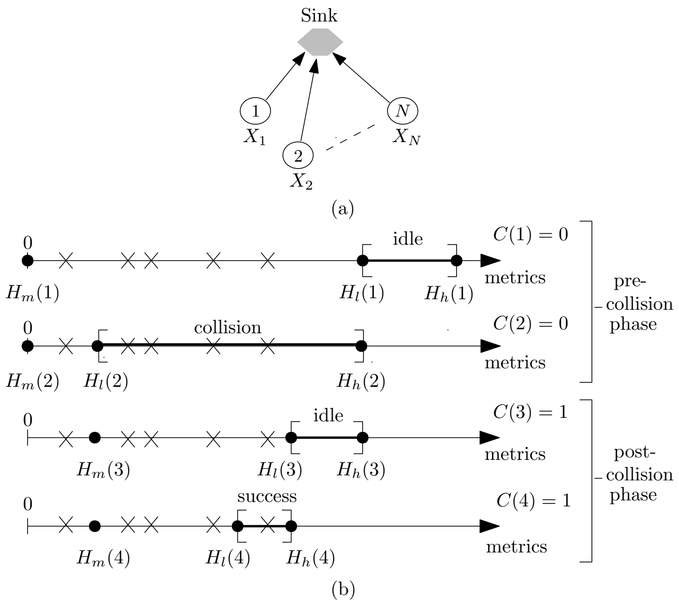

Fig. 1. (a) A system consisting of a sink and N users, in which the metric of user i is X_i; (b) An illustration of how the thresholds are updated in each slot by the splitting algorithm for $N = 5$ users. The metric of each user is indicated by a marker × and the transmission interval in each slot by [—].

RVs X_k and X_j is denoted by $\text{cov}(X_k, X_j)$. The notation $X \sim \mathcal{N}(\mu, \sigma^2)$ implies that X is a Gaussian RV with mean μ and variance σ^2. The magnitude of X is denoted by $|X|$, and $(\cdot)^T$ denotes transpose. The joint CDF of a random vector $\mathbf{X} = (X_1, \ldots, X_N)$ evaluated at $\mathbf{x} = (x_1, \ldots, x_N)$ is denoted by $F_{\mathbf{X}}(\mathbf{x}) = \Pr(X_1 \leq x_1, \ldots, X_N \leq x_N)$.

II. Splitting Algorithm Basics and Design of CASA

We consider a time-slotted system with N users and a sink, as shown in Figure 1a. Each user i possesses a real-valued metric X_i, which is an RV and is known only to it. For a given realization $x_1, \ldots, x_N$ of the metrics, our goal is to select the user with the highest metric, which is $\text{argmax}_{1 \leq i \leq N} \{x_1, \ldots, x_N\}$. The metrics remain constant over the time slots taken by the algorithm to select the best user [13], [14].

A. General Form of Splitting Algorithm and Review of IIDSA

We first formally state the splitting algorithm in a general form. It uses two functions lower$(\cdot)$ and split$(\cdot,\cdot)$ to update the thresholds. We then present the specific forms of lower$(\cdot)$ and split$(\cdot,\cdot)$ that IIDSA uses and explain the rationale behind them. In Section II-B, we use this general form to specify CASA.

Let the metrics be distributed between L and H. In each slot $k \geq 1$, every user maintains the variables $C(k)$, $H_m(k)$, $H_l(k)$, and $H_h(k)$, such that $L \leq H_m(k) < H_l(k) < H_h(k) \leq H$. Here, $H_m(k)$, $H_l(k)$, and $H_h(k)$ are the minimum, lower, and higher thresholds, respectively. The metric of the best user is known to lie in $(H_m(k), H_h(k)]$ at the beginning of slot k. The phase of the algorithm at the beginning of slot k is given by $C(k)$; it is 0 in the pre-collision phase and is 1 in the post-collision phase. A user i will transmit to the sink in the k^{th} slot only if its metric X_i lies in the transmission interval $(H_l(k), H_h(k)]$. At the end of slot k, the sink broadcasts a two-bit feedback $f(k)$ to all the users to communicate one among the following three outcomes: idle (0) if no user transmitted in that slot, success (1) if exactly one user transmitted, or collision (e) if two or more users transmitted and it could not decode any transmission.

Threshold-update rules: The threshold-update rules are given in terms of the functions lower$(\cdot)$ and split$(\cdot,\cdot)$. The variables in the first slot are initialized to $H_m(1) = L$, $H_h(1) = H$, $H_l(1) = \text{lower}(H)$, and $C(1) = 0$. Based on the feedback and the phase of the $(k-1)^{\text{th}}$ slot, every user updates its phase and thresholds for the k^{th} slot as follows:

1) If $f(k-1) = \text{idle}$ and $C(k-1) = 0$, then set $C(k) = 0$, $H_h(k) = H_l(k-1)$, $H_m(k) = H_m(k-1)$, and $H_l(k) = \text{lower}(H_l(k-1))$.
2) If $f(k-1) = \text{collision}$, then set $C(k) = 1$, $H_h(k) = H_h(k-1)$, $H_m(k) = H_l(k-1)$, and $H_l(k) = \text{split}(H_l(k-1), H_h(k-1))$.
3) If $f(k-1) = \text{idle}$ and $C(k-1) = 1$, then set $C(k) = 1$, $H_h(k) = H_l(k-1)$, $H_m(k) = H_m(k-1)$, and $H_l(k) = \text{split}(H_m(k-1), H_l(k-1))$.
4) If $f(k-1) = \text{success}$, then the best user has been selected; hence, the algorithm terminates.

Specific forms of lower$(\cdot)$ and split$(\cdot,\cdot)$ in IIDSA: For i.i.d. metrics, let the complementary CDF of X_i be denoted by $\bar{F}(x) = \Pr(X_i > x)$. Then, in IIDSA, lower$(\cdot)$ and split$(\cdot,\cdot)$ are defined in terms of $\bar{F}(\cdot)$ as follows [13]:

$$\text{lower}(h) = \bar{F}^{-1}\left(\bar{F}(h)\left(1 - \frac{1}{N}\right) + \frac{1}{N}\right), \qquad (1)$$

$$\text{split}(l, h) = \bar{F}^{-1}\left(\frac{\bar{F}(l) + \bar{F}(h)}{2}\right). \qquad (2)$$

To understand the rationale behind these definitions, consider the insightful special case in which the metrics are independent and uniformly distributed between 0 and 1. Then, (1) and (2) simplify to $\text{lower}(h) = h(1 - (1/N))$ and $\text{split}(l, h) = (l + h)/2$.

Since the average number of users that transmit in a slot is $N(H_h(k) - H_l(k))$, lower$(\cdot)$ ensures that one user, on average, transmits in the pre-collision phase. Doing so can be shown to maximize the probability of success in the next slot. On the other hand, in the post-collision phase, split$(\cdot,\cdot)$ ensures that, on average, half of the users involved in the last collision transmit [13], [14].

Brief explanation of the threshold-update rules: The algorithm is based on the classical collision model that is widely used in the MAC literature [19, Chap. 4]. The logic behind the threshold-update rules above is as follows:

1) If the feedback of the $(k-1)^{\text{th}}$ slot is idle and the algorithm is in the pre-collision phase ($C(k-1) = 0$), it implies that the metrics of all the users are less than or equal to $H_l(k-1)$. Hence, the algorithm continues to be in the pre-collision phase in the k^{th} slot, i.e., $C(k) = 0$, and the new higher threshold $H_h(k)$ is updated to $H_l(k-1)$. Since the best user is now known to lie in $(H_m(k-1), H_l(k-1)]$ at the beginning of the k^{th} slot,

$H_m(k)$ is set as $H_m(k-1)$. The new lower threshold $H_l(k)$ is given by lower$(H_l(k-1))$.

2) Regardless of the phase of the algorithm in the $(k-1)^{\text{th}}$ slot, if the feedback of the $(k-1)^{\text{th}}$ slot is collision, then the algorithm will be in the post-collision phase in the k^{th} slot, i.e., $C(k)=1$. A collision feedback implies that the metric of the best user lies in the interval $(H_l(k-1), H_h(k-1)]$ along with at least one other user's metric. Therefore, $H_m(k)$ is updated to $H_l(k-1)$ and $H_h(k)$ is set as $H_h(k-1)$. The interval $(H_l(k-1), H_h(k-1)]$, or equivalently $(H_m(k), H_h(k)]$, is split into two sub-intervals $(H_m(k), H_l(k)]$ and $(H_l(k), H_h(k)]$, and the upper sub-interval $(H_l(k), H_h(k)]$ is chosen as the transmission interval for the k^{th} slot. The value of $H_l(k)$ is given by split$(H_l(k-1), H_h(k-1))$.

3) If the feedback of the $(k-1)^{\text{th}}$ slot is idle and the algorithm is in the post-collision phase ($C(k-1)=1$), it implies that the metric of the best user lies in the interval $(H_m(k-1), H_l(k-1)]$ along with at least one other user's metric. Thus, $H_h(k)$ is updated to $H_l(k-1)$ and $H_m(k)$ is set as $H_m(k-1)$. Similar to the above case, $H_l(k)$ is determined by splitting the interval $(H_m(k-1), H_l(k-1)]$, or equivalently $(H_m(k), H_h(k)]$, and its value is given by split$(H_m(k-1), H_l(k-1))$.

4) If the feedback of the $(k-1)^{\text{th}}$ slot is success, it implies that exactly one user, which is the best user, transmitted and is successfully decoded. Hence, the algorithm terminates.

Figure 1b illustrates how the thresholds are updated by the splitting algorithm for 5 users. Slot 1 results in an idle outcome because there are no users whose metrics lie in $[H_l(1), H_h(1))$. Therefore, both higher and lower thresholds are lowered. In slot 2, four users whose metrics lie in $[H_l(2), H_h(2))$ transmit, resulting in a collision. Hence, the algorithm switches to the post-collision phase and this interval is split. In slot 3, an idle occurs because no user's metric lies in $[H_l(3), H_h(3))$. Now, the interval $[H_m(3), H_l(3))$ is split. In slot 4, only the best user transmits and success occurs.

B. Redesign of lower$(\cdot)$ and split$(\cdot,\cdot)$ in CASA

In order to handle correlated metrics, we propose the following common design rule for both phases to define lower$(\cdot)$ and split$(\cdot,\cdot)$ in CASA.

Design rule: Choose the thresholds such that the expected number of users that transmit in a slot, conditioned on the feedback and the phase of the previous slot, is q.

Here, $q>0$ is a system parameter, which we shall optimize. It is analogous to the *contention-load parameter* discussed in [14]. However, it was used only in the pre-collision phase in [14] and only for i.i.d. metrics. Instead, in CASA, we use it in both phases.

Let $n(k)$ denote the number of users that transmit in the k^{th} slot. This happens if and only if the metrics of these users lie in $(H_l(k), H_h(k)]$. Hence,

$$n(k) = \sum_{i=1}^{N} 1_{\{X_i \in (H_l(k), H_h(k)]\}}. \tag{3}$$

The design rule can then be mathematically stated as

$$\mathbb{E}\left[n(k) \mid f(k-1), C(k-1)\right] = q. \tag{4}$$

Based on the above rule, we now derive expressions for lower$(\cdot)$ and split$(\cdot,\cdot)$ in terms of the joint CDF $F_{\mathbf{X}}(\cdot)$ of the metrics. Let $\mathbf{X}^{(i)} = (X_1,\ldots,X_{i-1},X_{i+1},\ldots,X_N)$ and $F_{X_i,\mathbf{X}^{(i)}}(a,b,\ldots,b) = \Pr(X_i \le a, X_1 \le b, \ldots, X_{i-1} \le b, X_{i+1} \le b, \ldots, X_N \le b)$. Note that $X_1,\ldots,X_N$ need not be identical or independent, unlike the assumptions in IIDSA.

1) lower$(\cdot)$: Using the above design rule, we have the following result for lower$(\cdot)$.

Result *1:* When $f(k-1) =$ idle and $C(k-1)=0$, the lower threshold $H_l(k)$, which is given by lower$(H_l(k-1))$, is the solution of the following equation that is written in terms of the lower threshold of the $(k-1)^{\text{th}}$ slot:

$$\frac{\sum_{i=1}^{N} F_{X_i,\mathbf{X}^{(i)}}(H_l(k), H_l(k-1),\ldots,H_l(k-1))}{F_{\mathbf{X}}(H_l(k-1),\ldots,H_l(k-1))} = N-q. \tag{5}$$

Proof: The proof is relegated to Appendix A. ■

Setting $C(0)=0$, $f(0) =$ idle, and $H_l(0)=H$ ensures that $H_l(1) =$ lower(H).

2) split$(\cdot,\cdot)$: The value of $H_l(k)$ is given by split$(\cdot,\cdot)$ when: (i) the algorithm is in the post-collision phase in the $(k-1)^{\text{th}}$ slot and the feedback of the $(k-1)^{\text{th}}$ slot is idle, in which case it is $H_l(k) =$ split$(H_m(k-1), H_l(k-1))$, or (ii) the feedback of the $(k-1)^{\text{th}}$ slot is collision, in which case it is $H_l(k) =$ split$(H_l(k-1), H_h(k-1))$. In both cases, $H_l(k)$ can be compactly represented as $H_l(k) =$ split$(H_m(k), H_h(k))$ since $H_m(k)$ and $H_h(k)$ are directly specified in terms of the thresholds of the $(k-1)^{\text{th}}$ slot, as we saw in Section II-A. Using the above design rule, we have the following result.

Result *2:* When (i) $f(k-1) =$ idle and $C(k-1)=1$, or (ii) $f(k-1) =$ collision, the lower threshold $H_l(k)$, which is given by split$(H_m(k), H_h(k))$, is the solution of the following equation:

$$\frac{\Phi_1(H_l(k))}{\Phi_2(H_l(k))} = q, \tag{6}$$

where

$$\begin{aligned}\Phi_1(H_l(k)) = {} & N F_{\mathbf{X}}(H_h(k),\ldots,H_h(k)) \\ & - \sum_{i=1}^{N} F_{X_i,\mathbf{X}^{(i)}}(H_l(k), H_h(k),\ldots,H_h(k)) \\ & - \sum_{i=1}^{N} F_{X_i,\mathbf{X}^{(i)}}(H_h(k), H_m(k),\ldots,H_m(k)) \\ & + \sum_{i=1}^{N} F_{X_i,\mathbf{X}^{(i)}}(H_l(k), H_m(k),\ldots,H_m(k)),\end{aligned} \tag{7}$$

$$\begin{aligned}\Phi_2(H_l(k)) = {} & F_{\mathbf{X}}(H_h(k),\ldots,H_h(k)) \\ & - F_{\mathbf{X}}(H_m(k),\ldots,H_m(k)) \\ & - \sum_{i=1}^{N} F_{X_i,\mathbf{X}^{(i)}}(H_h(k), H_m(k),\ldots,H_m(k)) \\ & + N F_{\mathbf{X}}(H_m(k),\ldots,H_m(k)).\end{aligned} \tag{8}$$

Proof: The proof is relegated to Appendix B. ■

The above two results specify $H_l(k)$ as solutions of equations that are written in terms of the joint CDF $F_{\mathbf{X}}(\cdot)$ of the metrics. Note that these thresholds need not be computed on the fly by the users in each slot. Instead they can be pre-computed and broadcast to the users just once when the system commences operation.

III. Implementation and Analysis of CASA

We now consider two practically motivated and theoretically interesting models for the metrics. We discuss computationally efficient methods to compute their joint CDFs and, therefore, the thresholds of CASA. We then propose in Section III-C a hybrid approach, which reduces the computational burden as well as the memory storage requirements for executing CASA.

A. Correlated Exponential RVs

In this model, the metric X_k of user k is given by $X_k = |G_k|^2$, where

$$G_k = \sigma_k\left(\sqrt{1-\lambda_k^2}\,y_k + \lambda_k y_0\right) + \sqrt{-1}\sigma_k\left(\sqrt{1-\lambda_k^2}\,z_k + \lambda_k z_0\right), \tag{9}$$

$\lambda_k \in (-1,1)\setminus\{0\}$ and $y_0,\ldots,y_N,z_0,\ldots,z_N$ are i.i.d. zero-mean Gaussian RVs with variance $1/2$ [29]. Then, G_k, for $1 \leq k \leq N$, is a zero-mean circularly symmetric complex Gaussian RV with variance σ_k^2. It follows that X_k, for $1 \leq k \leq N$, is an exponential RV with mean σ_k^2, and the correlation coefficient ρ_{kj} between X_k and X_j, for $k \neq j$, is given by $\rho_{kj} = \lambda_k^2\lambda_j^2$. This includes as a special case the constant correlation model [24], for which $\lambda_1 = \cdots = \lambda_N = \lambda$.

Furthermore, the joint CDF $F_{\mathbf{X}}(\mathbf{x})$ of $\mathbf{X} = (X_1,\ldots,X_N)$ has the following single-integral form [29]:

$$F_{\mathbf{X}}(\mathbf{x}) = \int_0^\infty e^{-t}\prod_{k=1}^N\left[1 - Q_1\left(\frac{\sqrt{t}\sqrt{\sigma_k^2\lambda_k^2}}{\beta_k}, \frac{\sqrt{x_k}}{\beta_k}\right)\right]dt, \tag{10}$$

where $\beta_k^2 = \sigma_k^2(1-\lambda_k^2)/2$ and $Q_1(\cdot,\cdot)$ is the first order Marcum Q-function [30, Chap. 5]. It can be written in an integral-free form as

$$F_{\mathbf{X}}(\mathbf{x}) \approx \sum_{i=1}^n w_i\prod_{k=1}^N\left[1 - Q_1\left(\frac{\sqrt{t_i}\sqrt{\sigma_k^2\lambda_k^2}}{\beta_k}, \frac{\sqrt{x_k}}{\beta_k}\right)\right], \tag{11}$$

where t_i and w_i are the abscissas and weights, respectively, of the Gauss-Laguerre quadrature method [31, (25.4.45)]. The error in the approximation decreases to zero as n increases. This sum can be easily computed numerically since n is typically small.

We note that a similar approach can be applied even for other Gaussian-class multi-variate distributions, such as Rayleigh, Rician, Weibull, and Nakagami-m, which have the above general correlation structure [29].

B. Correlated Lognormal RVs

In this model, the metric X_k of user k is a lognormal RV. Hence, $X_k = e^{Z_k}$, where $Z_k \sim \mathcal{N}(\mu_k, \sigma_k^2)$. Furthermore, $Z_1,\ldots,Z_N$ are jointly Gaussian RVs with a covariance matrix $\mathbf{\Sigma}$. It can be shown that the mean of X_k is $e^{\mu_k+(1/2)\sigma_k^2}$ and $\text{cov}(X_k,X_j) = e^{\mu_k+\mu_j+(1/2)(\sigma_k^2+\sigma_j^2)}(e^{\Sigma_{kj}}-1)$, where Σ_{kj} denotes the $(k,j)^{\text{th}}$ element of $\mathbf{\Sigma}$. The correlation coefficient ρ_{kj} between X_k and X_j, for $k \neq j$, is given by

$$\rho_{kj} = \frac{e^{\Sigma_{kj}}-1}{\sqrt{(e^{\sigma_k^2}-1)(e^{\sigma_j^2}-1)}}. \tag{12}$$

The joint CDF $F_{\mathbf{X}}(\mathbf{x})$ of $\mathbf{X}$ is given by

$$\begin{aligned} F_{\mathbf{X}}(\mathbf{x}) &= F_{\mathbf{Z}}(\ln(\mathbf{x})) \\ &= \frac{1}{\sqrt{|\mathbf{\Sigma}|(2\pi)^N}}\int_{-\infty}^{b_1}\cdots\int_{-\infty}^{b_N} e^{-\frac{1}{2}\boldsymbol{\theta}^T\mathbf{\Sigma}^{-1}\boldsymbol{\theta}}d\boldsymbol{\theta}, \end{aligned} \tag{13}$$

where $b_k = \ln(x_k) - \mu_k$, for $1 \leq k \leq N$, and $|\cdot|$ is the determinant. To efficiently compute this integral, we use the randomized quasi-Monte Carlo (QMC) method, which we summarize below. We refer the reader to [32, Chap. 4] for a detailed discussion of the steps.

Randomized QMC method: It first performs the following transformations to convert the integral in (13) into an integral over a unit hyper-cube.

1) It starts with the transformation $\mathbf{y} = \mathbf{L}^{-1}\boldsymbol{\theta}$, where $\mathbf{LL}^T$ is the Cholesky decomposition of $\mathbf{\Sigma}$. After this transformation, (13) becomes

$$\begin{aligned} &F_{\mathbf{X}}(\mathbf{x}) \\ &= \frac{1}{\sqrt{(2\pi)^N}}\int_{-\infty}^{b'_1} e^{-\frac{y_1^2}{2}}\int_{-\infty}^{b'_2} e^{-\frac{y_2^2}{2}}\cdots\int_{-\infty}^{b'_N} e^{-\frac{y_N^2}{2}}d\mathbf{y}, \end{aligned} \tag{14}$$

where $b'_k = (b_k - \sum_{j=1}^{k-1} l_{kj}y_j)/l_{kk}$, for $1 \leq k \leq N$.

2) Next, it employs the transformation $w_k = \Phi(y_k)/\Phi(b'_k)$, where $\Phi(y) = (1/\sqrt{2\pi})\int_{-\infty}^{y} e^{-(1/2)\theta^2}d\theta$. After this transformation, we get the following integral over a unit hyper-cube:

$$F_{\mathbf{X}}(\mathbf{x}) = \int_0^1\int_0^1\cdots\int_0^1 f(\mathbf{w})d\mathbf{w}, \tag{15}$$

where $f(\mathbf{w}) = \prod_{k=1}^N e_k$ and $e_k = \Phi((b_k - \sum_{j=1}^{k-1} l_{kj}\Phi^{-1}(w_j e_j))/l_{kk})$, for $1 \leq k \leq N$.

3) To improve convergence, $f(\mathbf{w})$ is replaced with $(f(\mathbf{w}) + f(1-\mathbf{w}))/2$ and $\mathbf{w}$ with $|2\mathbf{w}-1|$.[2]

The N-dimensional integral in (15) is then computed as follows:

$$\begin{aligned} F_{\mathbf{X}}(\mathbf{x}) \approx \frac{1}{M}\sum_{i=1}^M \frac{1}{2P}\sum_{q=1}^P & f(|2\{\mathbf{s}_q+\mathbf{u}_i\}-1|) \\ &+ f(1-|2\{\mathbf{s}_q+\mathbf{u}_i\}-1|), \end{aligned} \tag{16}$$

[2] In order to speed up convergence, the integration variables in (13) are also reordered so that the inner integrals have values close to 1 [32, Chap. 4.1.3].

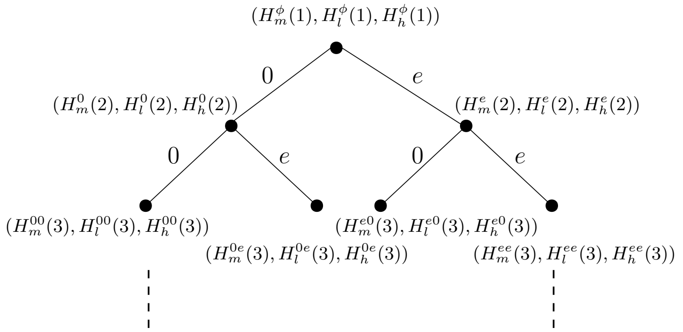
$(H_m^{\phi}(1), H_l^{\phi}(1), H_h^{\phi}(1))$
0
e
$(H_m^{0}(2), H_l^{0}(2), H_h^{0}(2))$
$(H_m^{e}(2), H_l^{e}(2), H_h^{e}(2))$
0
e
0
e
$(H_m^{00}(3), H_l^{00}(3), H_h^{00}(3))$
$(H_m^{e0}(3), H_l^{e0}(3), H_h^{e0}(3))$
$(H_m^{0e}(3), H_l^{0e}(3), H_h^{0e}(3))$
$(H_m^{ee}(3), H_l^{ee}(3), H_h^{ee}(3))$

Fig. 2. Tree-based representation of the splitting algorithm.

where $\{\mathbf{t}\}$ denotes the remainder of each of the elements of the vector $\mathbf{t}$ modulo 1, M is the number of random shifts, and $u_{i1}, \ldots, u_{iN}$ are i.i.d. uniform RVs that are distributed between 0 and 1. The set of vectors $\mathbf{s}_1, \ldots, \mathbf{s}_P$ is called a low-discrepancy sequence. An example of such sequence is the Kronecker sequence. In it, $\mathbf{s}_q$, for $1 \leq q \leq P$, is given by $\mathbf{s}_q = \left\{q\sqrt{\mathbf{p}}\right\}$, where $\mathbf{p} = (2, 3, 5, \ldots)$ consists of the first N prime numbers. M is typically between 8 and 12, and is, thus, small. It is shown in [32, Chap. 4] that the use of the low-discrepancy sequence results in a convergence rate of $\mathcal{O}(1/P)$, which is better than the convergence rate of $\mathcal{O}(1/\sqrt{P})$ of conventional Monte Carlo methods.

We note that the above approach can also be applied to metrics that are jointly Gaussian RVs, which arise, for example, in WSNs [20].

C. A Hybrid Approach to Reduce Computational Complexity and Memory Storage

The number of slots for which the thresholds need to be pre-computed also affects the computational complexity and the memory storage requirements of CASA. We propose a *hybrid approach* that leverages the simplicity of IIDSA to address this issue. To describe it, we use an alternate tree-based representation of the splitting algorithm, which is shown in Figure 2.

Each level of the tree indicates the slot number k. The root node is at level 1, its children are at level 2, and so on. Each node of the tree is associated with a threshold triplet, which denotes the set of the minimum, lower, and higher thresholds at the beginning of slot k. The algorithm begins from the root-node and traverses down the tree depending on the feedback received at the end of each slot. It chooses the left path (0) in case of an idle feedback, and the right path (e) in case of a collision feedback. If it receives a success feedback, then it terminates. Therefore, this is not shown in the figure.

We define a path s to a node to be the sequence of feedback events that led to that node. The threshold triplet associated with a node in level k depends on the path s that led to it, and is denoted as $(H_m^s(k), H_l^s(k), H_h^s(k))$. For the root node, it is denoted as $(H_m^{\phi}(1), H_l^{\phi}(1), H_h^{\phi}(1))$. Let $\mathcal{S}_k$ be the set of all paths that start from the root-node and end at any node in level k; there are 2^{k-1} such paths. For example, $\mathcal{S}_1 = \phi$, $\mathcal{S}_2 = \{0, e\}$, $\mathcal{S}_3 = \{00, 0e, e0, ee\}$, and so on. As a result, the total number of threshold triplets that need to be pre-computed and stored up to level U is $2^U - 1$.

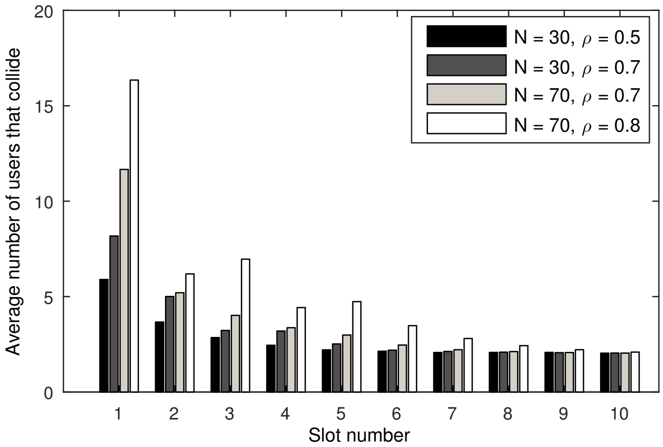

Fig. 3. Average number of users that collide in each slot if a collision occurs in that slot using CASA ($q = 1$).

In the *hybrid approach* that we propose, the threshold triplets at each node of the tree are pre-computed using the rules given in Section II-B only up to a particular slot U, which we shall refer to as the *pre-compute tree depth*. The thresholds for the subsequent slots, which come into play if no user has been selected until then, are determined as per IIDSA, and are given in (1) and (2). For example, if the metrics are exponential RVs as in Section III-A, these simplify to lower$(h) = -\gamma \ln(e^{-h/\gamma}(1 - (1/N)) + (1/N))$ and split$(l, h) = -\gamma \ln((e^{-l/\gamma} + e^{-h/\gamma})/2)$, where γ is the mean of the exponential RVs. These can be computed easily by the users themselves, and, therefore, need not be stored in them a priori.

The above approach is justified due to the following reasons. First, the average number of users that collide in slot k if a collision occurs decreases as k increases. To illustrate this, we consider the case of exponential metrics with a constant correlation coefficient ρ. Figure 3 plots the average number of users that collide in each slot if a collision occurs in that slot using CASA, for different values of ρ and N, when $q = 1$. To obtain this plot, 10^5 realizations of the vector of metrics $\mathbf{X}$ are generated using (9) with $\sigma_k^2 = 2$ and $\lambda_k = \rho^{1/4}$, for all k. We use $n = 8$ terms in (11) to compute $F_{\mathbf{X}}(\mathbf{x})$.

We observe that if a collision has occurred in a slot, then the average number of users that collide decreases with each slot. For example, it decreases from 16.35 in slot 1 to 2.42 in slot 8, for $N = 70$ and $\rho = 0.8$. Consequently, if the correlation is ignored, it makes sense to split the interval of the 8th slot such that, on average, half of the users that collided previously transmit in the 9th slot. This is what the function split$(\cdot, \cdot)$ of IIDSA does in the post-collision phase.

Second, the probability that the algorithm will be in the pre-collision phase in slot k decreases as k increases. For example, for $N = 50$ and $\rho = 0.7$, the above simulations show that the probability that the algorithm is in the pre-collision phase in the 6th, 7th, and 8th slots is 0.10, 0.06, and 0.04, respectively. Consequently, replacing lower$(\cdot)$ with that of IIDSA after slot

8 has a negligible impact. We numerically evaluate the efficacy of this approach and determine U in Section IV-A.

D. Analysis of Selection Time

The number of slots T taken by CASA to select the best user is an RV because it is a function of the realizations of the metrics of the users. We, therefore, derive its CDF below. For this, we utilize the tree-based representation of the splitting algorithm that was described in Section III-C.

Result *3:* The CDF $F_T(t)$ of the selection time T is given by

$$F_T(t) = \Pr(T \leq t) = \sum_{k=1}^{t} \sum_{s \in \mathcal{S}_k} \Pr(T = k, s)\,, \qquad (17)$$

where $\Pr(T = k, s)$ is the probability that the best user is selected in the k^{th} slot after traversing the path $s \in \mathcal{S}_k$. It is given in terms of the joint CDF of the metrics by

$$\begin{aligned}\Pr(T = k, s) = &\sum_{i=1}^{N} F_{X_i, \mathbf{X}^{(i)}}(H_h^s(k), H_l^s(k), \ldots, H_l^s(k)) \\ &- N F_{\mathbf{X}}(H_l^s(k), \ldots, H_l^s(k)) \\ &- \sum_{i=1}^{N} F_{X_i, \mathbf{X}^{(i)}}(H_h^s(k), H_m^s(k), \ldots, H_m^s(k)) \\ &+ \sum_{i=1}^{N} F_{X_i, \mathbf{X}^{(i)}}(H_l^s(k), H_m^s(k), \ldots, H_m^s(k)).\end{aligned} \qquad (18)$$

Proof: The proof is relegated to Appendix C. ■

The expression in (18) can be numerically computed easily using (11) for correlated exponential RVs and the randomized QMC method in Section III-B for correlated lognormal RVs. From the CDF, other performance measures, such as the expectation and variance of the selection time, can be directly computed. For example, the expected selection time is given by $\mathbb{E}\left[T\right] = \sum_{t=1}^{\infty}(1 - F_T(t))$.

IV. Simulation Results and Comparisons

We now present Monte Carlo simulation results that are averaged over 10^5 realizations of the metrics $\mathbf{X}$. We benchmark the performance of CASA against IIDSA and the polling algorithm for the two models considered in Sections III-A and III-B.

A. Correlated Exponential RVs

We first consider the constant correlation model [24], in which the correlation coefficient between any two metrics is equal to ρ. This model helps us study the implications of correlation by varying the parameter ρ. The metrics $X_1, \ldots, X_N$ are generated as follows. We first generate the RVs $G_1, \ldots, G_N$ as per (9) with $\lambda_1 = \cdots = \lambda_N = \rho^{1/4}$ and $\sigma_1^2 = \cdots = \sigma_N^2 = 2$. Then, $X_k = |G_k|^2$, for $1 \leq k \leq N$, is an exponential RV with mean 2 and the correlation coefficient between X_k and X_j, for $k \neq j$, is ρ. We use $n = 8$ in (11).

Determining the pre-compute tree depth U of the hybrid approach: Figure 4 plots the average selection time of CASA as a function of N for different choices of U and ρ, for $q = 1$. For any ρ, $U = 0$ corresponds to the average selection time when IIDSA is used. For $\rho = 0$ (the i.i.d. case), the curves overlap for all values of U. However, for $\rho > 0$, the average selection time decreases as U increases, for all N. This is because the design of CASA takes correlation into account for more slots U. For any ρ and N, the curves are close to each other for $U \geq 8$. We, therefore, set $U = 8$ in our simulations henceforth. For small ρ (e.g., $\rho \leq 0.5$), $U = 5$ is sufficient.

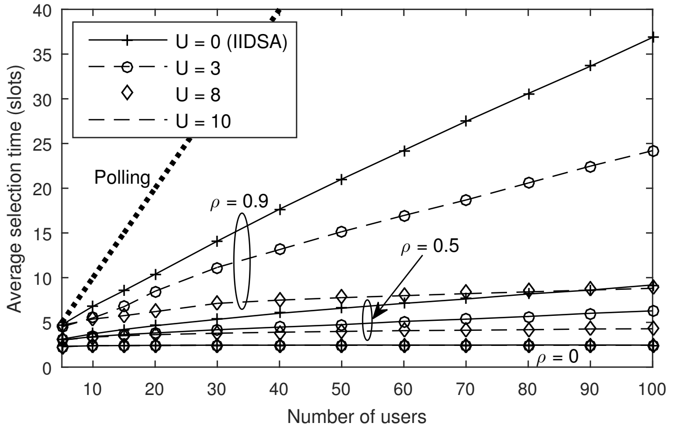

Fig. 4. Average selection time of CASA as a function of the number of users for different choices of U for exponential RVs with a constant correlation coefficient ρ ($q = 1$).

Figure 4 also benchmarks the performance of CASA with IIDSA ($U = 0$) and the polling algorithm. The average selection time of the polling algorithm increases linearly with N for all ρ. It performs much worse than both CASA and IIDSA. When $\rho = 0$, the average selection times of CASA and IIDSA both saturate at 2.47 slots even when N tends to infinity. However, for $\rho > 0$, the average selection time of CASA is less than that of IIDSA. For example, when $N = 50$, the average selection time of CASA is 21.91% to 62.75% less than that of IIDSA as ρ increases from 0.3 to 0.9. Similarly, when $\rho = 0.9$, it is 21.05% to 75.59% less than that of IIDSA as N increases from 10 to 100. While the average selection time of CASA also increases as ρ or N increases, it does so at a smaller rate than for IIDSA. For example, when $\rho = 0.7$, as N increases from 10 to 100, it increases by 3.59 times for IIDSA, but by only 1.46 times for CASA.

CDF of average selection time: An alternate way to visualize and compare the performances of the selection algorithms is to compare the CDF $F_T(t)$ of their selection times. Figure 5 compares $F_T(t)$ of IIDSA and CASA for different ρ with $N = 20$ and $q = 1$. We observe that the CDF of CASA is to the left of IIDSA. This implies that the probability that the best user has been selected by a given slot is greater for CASA. For example, for $N = 20$ and $\rho = 0.7$, $F_T(7)$ is 0.932 for CASA, but only 0.689 for IIDSA. We also observe that the CDF curves of both algorithms shift downward as ρ increases from 0 to 0.7. This implies that the number of slots required to select the best user, on average, increases for both algorithms as ρ increases. The separation between the CDF curves of the two algorithms increases as ρ increases, which is consistent

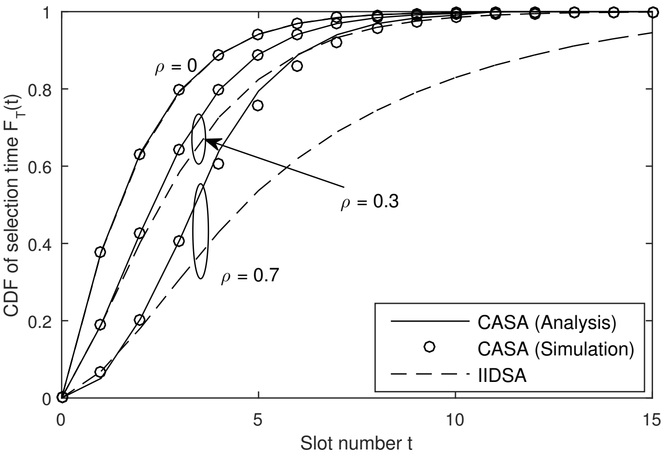

Fig. 5. Comparison of CDFs of selection time of CASA and IIDSA for different ρ ($N = 20$ and $q = 1$).

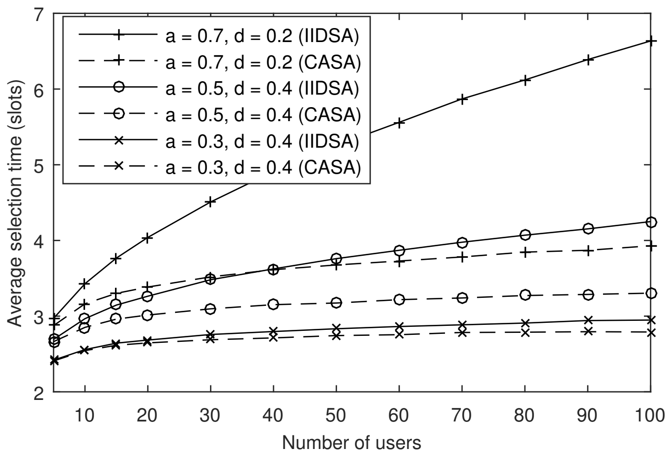

Fig. 7. Comparison of average selection times of CASA and IIDSA as a function of the number of users for exponential RVs with arbitrary correlation ($q = 1$ and $U = 8$).

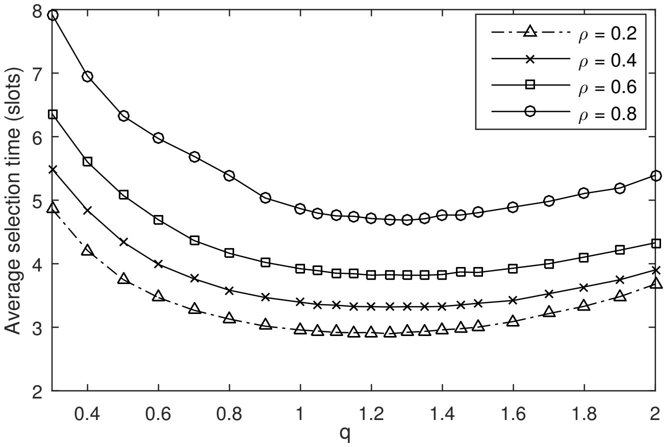

Fig. 6. Average selection time of CASA as a function of q for different values of ρ ($N = 20$).

with the trends observed in Figure 4. Also shown in the figure are the results from the analytical expression in (17). These match the simulation results well.

Optimizing q: Next, we optimize the loading parameter q in order to reduce the average selection time further. Figure 6 plots the average selection time of CASA as a function of q for $N = 20$. The results are generated numerically by varying q from 0.3 to 2.0.

For any ρ and N, we observe that the average selection time initially decreases as q increases. It reaches a minimum value and then starts increasing as q is increased further. This is because when q is small, the algorithm wastes more idle slots in the pre-collision phase. On the other hand, if q is large (e.g., $q \geq 2$), the algorithm wastes more slots resolving collisions in the post-collision phase because many more than two users are likely to transmit in these slots. The optimal value of q, thus, lies between 1 and 2, and is determined numerically from the figure. When $\rho = 0.8$ and $N = 20$, the optimal value of q is 1.3. Compared to $q = 1$, the average selection time is 3.72% lower. Similarly, for $\rho = 0.8$ and $N = 30$, the optimal value of q is 1.45, for which the average selection time is 9.9% lower than for $q = 1$.

Arbitrary correlation model: In order to illustrate the generality of the design of CASA, we now present results for the case when the correlation coefficients are different for different pairs of users. To do this, we generate the RVs $G_1, \ldots, G_N$ as per (9) with $\sigma_1^2 = \cdots = \sigma_N^2 = 2$. The values of λ_k, for $1 \leq k \leq N$, are evenly spaced in the interval $[a - (d/2), a + (d/2)]$ as follows: $\lambda_k = a - (d/2) + d(k-1)/(N-1)$, where $d < 2$ and $-(1-(d/2)) < a < (1-(d/2))$. Intuitively, the larger the ratio d/a, the more spread out are the correlation coefficients. As before, the metric of user k is $X_k = |G_k|^2$.

Figure 7 compares the average selection times of CASA and IIDSA as a function of N for different values of a and d, for $U = 8$ and $q = 1$. We observe that CASA again outperforms IIDSA. For example, when $a = 0.7$ and $d = 0.4$, the average selection time of CASA is 7.75% to 40.78% less than that of IIDSA as N increases from 10 to 100. The average selection time of CASA increases as a increases, which is similar to the behavior with respect to ρ in Figure 4.

B. Correlated Lognormal RVs

We first generate the RVs $Z_1, \ldots, Z_N$, such that $Z_k \sim \mathcal{N}(0, 1)$, for $1 \leq k \leq N$, and the correlation coefficient between Z_k and Z_j, for $k \neq j$, is $\ln(1 + \rho(e-1))$. The metric of user k is $X_k = e^{Z_k}$. Thus, from (12), any two metrics have a constant correlation coefficient of ρ [24]. Here, ρ is a parameter that we vary to study the impact of correlation. To compute the joint CDF $F_{\mathbf{X}}(\mathbf{x})$ using (16), M and P are chosen as 12 and 500, respectively.

Figure 8 compares the average selection times of CASA (dashed line), IIDSA (solid line), and the polling algorithm (dotted line) as a function of N for different values of ρ. As before, for $\rho = 0$, the curves of IIDSA and CASA overlap with each other. However, for $\rho > 0$, the average selection time of CASA is less than that of IIDSA, which mirrors the trends of Figure 4. For example, when $N = 50$, the average selection time of CASA is 33.06% to 58.95% less than that of IIDSA as ρ increases from 0.3 to 0.9. Similarly, for $\rho = 0.7$,

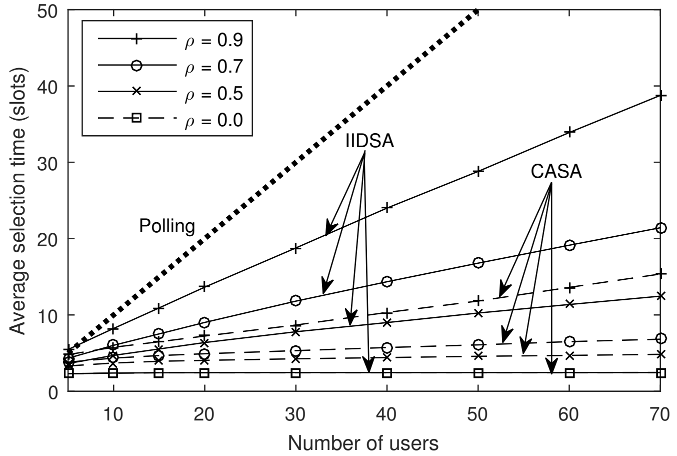

Fig. 8. Comparison of average selection times of CASA, IIDSA, and the polling algorithm as a function of the number of users for lognormal RVs with constant correlation coefficient ρ ($q = 1$ and $U = 8$).

it is 27.81% to 68.16% less than that of IIDSA as N increases from 10 to 70.

C. System-level Throughput Implications

In order to gain a different insight about CASA, we study its efficacy in implementing relay selection in a cooperative system. In it, a source broadcasts a message of B bits to a set of N decode-and-forward (DF) relays for T_s slots, where each slot is of duration t_{slot} seconds. A relay i decodes the source's message only if $B \leq W T_s t_{\text{slot}} \log_2(1 + (P_s h_i)/(N_0 W))$, where W is the transmission bandwidth, P_s is the transmit power, h_i is the SR channel power gain, and N_0 is the power spectral density of noise. This is followed by the relay selection phase of duration T_r slots, in which CASA is employed to select one relay. If relay i is selected, it will forward the message to the destination for T_s slots. The destination will decode this message only if $B \leq W T_s t_{\text{slot}} \log_2(1 + (P_s g_i)/(N_0 W))$, where g_i is the RD channel power gain.

In the relay selection phase, only those relays that decode the source's message and whose RD channel power gains are high enough to enable the destination to decode the relay's transmission participate in the relay selection phase. Hence, the metric of relay i in the selection phase is set as [33]

$$X_i = \begin{cases} g_i, & \text{if } h_i \geq \left(2^{\frac{B}{W T_s t_{\text{slot}}}} - 1\right)\frac{N_0 W}{P_s} \text{ and} \\ & \quad g_i \geq \left(2^{\frac{B}{W T_s t_{\text{slot}}}} - 1\right)\frac{N_0 W}{P_s}, \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

If no relay is selected, an outage occurs. The throughput for this system is given by

$$\eta = \frac{B \Pr(T_r)}{(2T_s + T_r) t_{\text{slot}}}, \tag{20}$$

where $\Pr(T_r)$ is the probability that the destination successfully decodes the source's message, which is the same as the probability that CASA selects a relay within T_r slots. Too small a value for T_r reduces η because $\Pr(T_r)$ becomes small. Too large a value of T_r also reduces η because of its presence in the denominator in (20). Therefore T_r needs to be optimized.

We use the following parameters in our simulations. The SR (and RD) channel power gains are exponential RVs with mean 1 and have a constant correlation coefficient of ρ. We set $B/(W T_s t_{\text{slot}}) = 1$ bits/s/Hz, $T_s = 40$ slots, and $P_s/(N_0 W) = 3$ dB. For $N = 20$ and $\rho = 0.3$, the maximum normalized throughput η/W with CASA turns out to be 0.40 bits/s/Hz and occurs at $T_r = 15$ slots. It is greater than the optimal normalized throughput with IIDSA, which turns out to be 0.38 bits/s/Hz and occurs at $T_r = 20$ slots. When ρ is increased to 0.7, the optimal throughput with CASA decreases to 0.37 bits/s/Hz at $T_r = 17$ slots. It is greater than that with IIDSA, which decreases to 0.33 bits/s/Hz and occurs at $T_r = 29$ slots.

V. Conclusions

We saw that in the presence of correlation, the design of the splitting algorithm is more sophisticated than for the i.i.d. case. We developed a novel splitting-based opportunistic selection algorithm CASA for correlated and non-identical metrics. It used a new design rule, which redefined the functions lower$(\cdot)$ and split$(\cdot,\cdot)$ that were used to update the thresholds. These functions were specified as solutions of equations involving the joint CDF of the metrics. For two practically motivated models, we discussed techniques to compute their joint CDFs with lower complexity, which enabled the thresholds to be pre-computed and broadcast to the users once when the system commenced operation. We also proposed a hybrid approach that further reduced the computational and memory storage burden. We saw that in the presence of correlation, IIDSA was no longer as scalable and that CASA significantly reduced the average selection time.

An interesting avenue for future work is to investigate the impact of imperfect knowledge of correlation statistics on CASA. An in-depth study on the system-level trade-offs associated with opportunistic selection in the presence of correlation is another important problem to be studied.

Appendix

A. Proof of Result 1

Let $\xi(h)$ be the event that all the metrics are less than or equal to h. Since the algorithm is in the pre-collision phase in the $(k-1)^{\text{th}}$ slot and the feedback of the $(k-1)^{\text{th}}$ slot is idle, we know that the event $\xi(H_l(k-1))$ has occurred. From the design rule in (4), then $H_l(k)$ is set such that

$$\mathbb{E}\left[n(k) \mid \xi(H_l(k-1))\right] = q. \tag{21}$$

Substituting the expression for $n(k)$ from (3), and using the linearity of expectation, we get

$$\begin{aligned} &\mathbb{E}\left[n(k) \mid \xi(H_l(k-1))\right] \\ &\quad = \sum_{i=1}^{N} \Pr(X_i \in (H_l(k), H_l(k-1)] \mid \xi(H_l(k-1))) . \end{aligned} \tag{22}$$

Applying Bayes' rule and writing in terms of the joint CDF $F_{\mathbf{X}}(\cdot)$, we get

$$\begin{aligned}&\mathbb{E}\left[n(k) \mid \xi(H_l(k-1))\right] \\ &= N - \frac{\sum_{i=1}^{N} F_{X_i,\mathbf{X}^{(i)}}(H_l(k), H_l(k-1), \ldots, H_l(k-1))}{F_{\mathbf{X}}(H_l(k-1), \ldots, H_l(k-1))}.\end{aligned} \tag{23}$$

Equating (23) to q yields the desired result.

B. Proof of Result 2

Let $A_0(u,v)$, $A_1(u,v)$, and $A_2(u,v)$ denote the events that there are zero, one, and at least two users, respectively, whose metrics lie in the interval $(u,v]$. From the threshold-update rules, it can be observed that whenever split$(\cdot,\cdot)$ is used, the events $\xi(H_h(k))$ and $A_2(H_m(k),H_h(k))$ have both occurred. Thus, as per the design rule, in the post-collision phase, the lower threshold $H_l(k)$ is set such that

$$\mathbb{E}\left[n(k) \mid \xi(H_h(k)), A_2(H_m(k), H_h(k))\right] = q. \tag{24}$$

We first derive a simpler, intermediate expression for $\mathbb{E}\left[n(k) \mid \xi(H_h(k)), A_2(H_m(k), H_h(k))\right]$ in the following lemma.

Lemma *1:* The average number of users that transmit in the k^{th} slot, conditioned on the events $\xi(H_h(k))$ and $A_2(H_m(k),H_h(k))$, is given by

$$\begin{aligned}&\mathbb{E}[n(k) \mid \xi(H_h(k)), A_2(H_m(k), H_h(k))] \\ &\qquad = \frac{\mathbb{E}\left[n(k) \mid \xi(H_h(k))\right] - \mathcal{P}_1}{\mathcal{P}_2},\end{aligned} \tag{25}$$

where

$$\mathcal{P}_1 = \Pr(A_1(H_l(k), H_h(k)), A_1(H_m(k), H_h(k)) \,|\, \xi(H_h(k)))\,, \tag{26}$$

$$\mathcal{P}_2 = \Pr(A_2(H_m(k), H_h(k)) \mid \xi(H_h(k)))\,. \tag{27}$$

Proof: The events $A_0(H_m(k),H_h(k))$, $A_1(H_m(k),H_h(k))$, and $A_2(H_m(k),H_h(k))$ are mutually exclusive and exhaustive. Hence, from the law of total expectation, it follows that

$$\begin{aligned}&\mathbb{E}[n(k) \mid \xi(H_h(k))] \\ &\quad = \sum_{i=0}^{2} \mathbb{E}\left[n(k) \mid \xi(H_h(k)), A_i(H_m(k), H_h(k))\right] \\ &\qquad \times \Pr(A_i(H_m(k), H_h(k)) \mid \xi(H_h(k)))\,.\end{aligned} \tag{28}$$

We consider each term of (28) separately below.

1. *When $i = 0$:* Given that $A_0(H_m(k),H_h(k))$ has occurred and since $H_m(k) < H_l(k) < H_h(k)$, it follows that

$$\mathbb{E}\left[n(k) \mid \xi(H_h(k)), A_0(H_m(k), H_h(k))\right] = 0. \tag{29}$$

2. *When $i = 1$:* Given that $A_1(H_m(k),H_h(k))$ has occurred, it implies that either $A_0(H_l(k),H_h(k))$ or $A_1(H_l(k),H_h(k))$ has occurred. If $A_0(H_l(k),H_h(k))$ has occurred, then $n(k) = 0$. Else, if $A_1(H_l(k),H_h(k))$ has occurred, then $n(k) = 1$. Hence, it follows that

$$\begin{aligned}&\mathbb{E}\left[n(k) \mid \xi(H_h(k)), A_1(H_m(k), H_h(k))\right] \\ &= \Pr(A_1(H_l(k), H_h(k)) \mid \xi(H_h(k)), A_1(H_m(k), H_h(k)))\,.\end{aligned} \tag{30}$$

Substituting (29) and (30) in (28) yields

$$\begin{aligned}&\mathbb{E}\left[n(k) \mid \xi(H_h(k))\right] \\ &= \Pr(A_1(H_l(k), H_h(k)) \mid \xi(H_h(k)), A_1(H_m(k), H_h(k))) \\ &\qquad \times \Pr(A_1(H_m(k), H_h(k)) \mid \xi(H_h(k))) \\ &\quad + \mathbb{E}\left[n(k) \mid \xi(H_h(k)), A_2(H_m(k), H_h(k))\right] \\ &\qquad \times \Pr(A_2(H_m(k), H_h(k)) \mid \xi(H_h(k)))\,.\end{aligned} \tag{31}$$

By Bayes' rule, the first term in (31) reduces to $\Pr(A_1(H_l(k),H_h(k)), A_1(H_m(k),H_h(k)) \mid \xi(H_h(k)))$. Thus, (31) reduces to

$$\begin{aligned}&\mathbb{E}[n(k) \mid \xi(H_h(k))] = \mathcal{P}_1 + \mathcal{P}_2 \\ &\qquad \times \mathbb{E}\left[n(k) \mid \xi(H_h(k)), A_2(H_m(k), H_h(k))\right],\end{aligned} \tag{32}$$

where $\mathcal{P}_1$ and $\mathcal{P}_2$ are as specified in (26) and (27), respectively. Rearranging the terms in (32) yields the desired result. ■

The expression for $\mathbb{E}\left[n(k)|\xi(H_h(k))\right]$ in (25) is already derived in (23). Next, we derive expressions for $\mathcal{P}_1$ and $\mathcal{P}_2$.

1) Expression for $\mathcal{P}_1$: Applying Bayes' rule, (26) can be written as

$$\mathcal{P}_1 = \frac{\Pr(A_1(H_l(k), H_h(k)), A_1(H_m(k), H_h(k)), \xi(H_h(k)))}{\Pr(\xi(H_h(k)))}. \tag{33}$$

The intersection of the events $A_1(H_l(k),H_h(k))$, $A_1(H_m(k),H_h(k))$, and $\xi(H_h(k))$ implies that there is exactly one user whose metric lies in $(H_l(k),H_h(k)]$ and the metrics of the remaining $(N-1)$ users are less than or equal to $H_m(k)$. Thus, (33) can be written in terms of the joint CDF as follows:

$$\begin{aligned}\mathcal{P}_1 = &\frac{\sum_{i=1}^{N} F_{X_i,\mathbf{X}^{(i)}}(H_h(k), H_m(k), \ldots, H_m(k))}{F_{\mathbf{X}}(H_h(k), \ldots, H_h(k))} \\ &- \frac{\sum_{i=1}^{N} F_{X_i,\mathbf{X}^{(i)}}(H_l(k), H_m(k), \ldots, H_m(k))}{F_{\mathbf{X}}(H_h(k), \ldots, H_h(k))}.\end{aligned} \tag{34}$$

2) Expression for $\mathcal{P}_2$: Since $A_0(H_m(k),H_h(k))$, $A_1(H_m(k),H_h(k))$, and $A_2(H_m(k),H_h(k))$ are mutually exclusive and exhaustive, we have

$$\begin{aligned}\mathcal{P}_2 = 1 &- \Pr(A_0(H_m(k), H_h(k)) \mid \xi(H_h(k))) \\ &- \Pr(A_1(H_m(k), H_h(k)) \mid \xi(H_h(k)))\,.\end{aligned} \tag{35}$$

Using Bayes' rule, we get

$$\begin{aligned}&\Pr(A_0(H_m(k), H_h(k)) \mid \xi(H_h(k))) \\ &\qquad = \frac{\Pr(A_0(H_m(k), H_h(k)), \xi(H_h(k)))}{\Pr(\xi(H_h(k)))}.\end{aligned} \tag{36}$$

Since the intersection of the two events $A_0(H_m(k),H_h(k))$ and $\xi(H_h(k))$ implies that the metrics of all the users are less than or equal to $H_m(k)$, we get

$$\begin{aligned}\Pr(A_0(H_m(k), H_h(k)) \mid \xi(H_h(k)))\\ = \frac{F_{\mathbf{X}}(H_m(k), \ldots, H_m(k))}{F_{\mathbf{X}}(H_h(k), \ldots, H_h(k))}. \quad (37)\end{aligned}$$

Similarly, we can show that

$$\begin{aligned}\Pr(A_1(H_m(k),&H_h(k)) \mid \xi(H_h(k)))\\ &= \frac{\sum_{i=1}^{N} F_{X_i,\mathbf{X}^{(i)}}(H_h(k), H_m(k), \ldots, H_m(k))}{F_{\mathbf{X}}(H_h(k), \ldots, H_h(k))}\\ &\quad - N\frac{F_{\mathbf{X}}(H_m(k), \ldots, H_m(k))}{F_{\mathbf{X}}(H_h(k), \ldots, H_h(k))}. \quad (38)\end{aligned}$$

Substituting (23), (34), and (35) in (25), we get

$$\mathbb{E}\left[n(k) \mid \xi(H_h(k)), A_2(H_m(k), H_h(k))\right] = \frac{\Phi_1(H_l(k))}{\Phi_2(H_l(k))}, \quad (39)$$

where $\Phi_1(H_l(k))$ and $\Phi_2(H_l(k))$ are as specified in (7) and (8), respectively. Equating (39) to q yields the desired result.

C. Proof of Result 3

From the law of total probability, the probability that the best user is selected in the k^{th} slot is given by

$$\Pr(T = k) = \sum_{s \in \mathcal{S}_k} \Pr(T = k, s). \quad (40)$$

We derive $\Pr(T = k, s)$ for the following three paths separately: (i) $s = \phi$ path that occurs when the algorithm starts in slot 1, (ii) the all-0 path Ω_k that occurs when the feedbacks of slots $1, \ldots, (k-1)$ are idles, and (iii) all the remaining paths.

1) $s = \phi$: The $s = \phi$ path occurs only when $k = 1$. Hence, $\Pr(T = k, \phi) = 0$, for $k \geq 2$. We have $\Pr(T = 1) = \Pr(T = 1, \phi)$. The probability that a user is selected in slot 1 is the probability that there is exactly one user in $\left(H_l^{\phi}(1), H_h^{\phi}(1)\right]$ and the remaining $(N-1)$ users are less than or equal to $H_l^{\phi}(1)$. Thus, we get

$$\begin{aligned}\Pr(T = 1, \phi) = &\left[\sum_{i=1}^{N} F_{X_i,\mathbf{X}^{(i)}}(H_h^{\phi}(1), H_l^{\phi}(1), \ldots, H_l^{\phi}(1))\right]\\ &- NF_{\mathbf{X}}(H_l^{\phi}(1), \ldots, H_l^{\phi}(1)). \quad (41)\end{aligned}$$

2) All-0 path Ω_k: An all-0 path Ω_k, for $k \geq 2$, is the path in which idles (0) have occurred in slots $1, \ldots, (k-1)$. For example, $\Omega_2 = 0$, $\Omega_3 = 00$, and so on. Hence, the probability that the best user is selected in the k^{th} slot after traversing the path Ω_k is the probability that there is exactly one user whose metric lies in $\left(H_l^{\Omega_k}(k), H_h^{\Omega_k}(k)\right]$ and the metrics of the remaining $(N-1)$ users are less than or equal to $H_l^{\Omega_k}(k)$. Thus, we get

$$\begin{aligned}\Pr(T = k, \Omega_k) = &-NF_{\mathbf{X}}(H_l^{\Omega_k}(k), \ldots, H_l^{\Omega_k}(k))\\ &+ \sum_{i=1}^{N} F_{X_i,\mathbf{X}^{(i)}}(H_h^{\Omega_k}(k), H_l^{\Omega_k}(k), \ldots, H_l^{\Omega_k}(k)). \quad (42)\end{aligned}$$

3) Any other path: For any other path $s \in \mathcal{S}_k \setminus \{\Omega_k\}$ and $k \geq 2$, the events $\xi(H_h^s(k))$ and $A_2(H_m^s(k), H_h^s(k))$, which are defined in Appendices A and B, respectively, have occurred since the path involves at least one collision (e). For the best user to be selected in the k^{th} slot, $A_1(H_l^s(k), H_h^s(k))$ must, therefore, occur. Thus, the probability that the best user is selected in slot k after traversing this path s is the probability of intersection of the events $A_1(H_l^s(k), H_h^s(k))$, $A_2(H_m^s(k), H_h^s(k))$, and $\xi(H_h^s(k))$. Therefore,

$$\begin{aligned}&\Pr(T = k, s)\\ &= \Pr(A_1(H_l^s(k), H_h^s(k)), A_2(H_m^s(k), H_h^s(k)), \xi(H_h^s(k))). \quad (43)\end{aligned}$$

Using Bayes' rule, (43) can be written as

$$\begin{aligned}&\Pr(T = k, s)\\ &= \Pr(A_2(H_m^s(k), H_h^s(k)) \mid A_1(H_l^s(k), H_h^s(k)), \xi(H_h^s(k)))\\ &\quad \times \Pr(A_1(H_l^s(k), H_h^s(k)), \xi(H_h^s(k))). \quad (44)\end{aligned}$$

We now derive expressions for the two terms in (44). First, by simple inspection, we get

$$\begin{aligned}\Pr(A_1(&H_l^s(k), H_h^s(k)), \xi(H_h^s(k)))\\ &= \left[\sum_{i=1}^{N} F_{X_i,\mathbf{X}^{(i)}}(H_h^s(k), H_l^s(k), \ldots, H_l^s(k))\right]\\ &\quad - NF_{\mathbf{X}}(H_l^s(k), \ldots, H_l^s(k)). \quad (45)\end{aligned}$$

Second, we can write the conditional probability term in (44) as follows:

$$\begin{aligned}&\Pr(A_2(H_m^s(k), H_h^s(k)) \mid A_1(H_l^s(k), H_h^s(k)), \xi(H_h^s(k))) =\\ &1 - \Pr(A_0(H_m^s(k), H_h^s(k)) \mid A_1(H_l^s(k), H_h^s(k)), \xi(H_h^s(k)))\\ &\quad - \Pr(A_1(H_m^s(k), H_h^s(k)) \mid A_1(H_l^s(k), H_h^s(k)), \xi(H_h^s(k))). \quad (46)\end{aligned}$$

Since $H_m^s(k) < H_l^s(k) < H_h^s(k)$, given that $A_1(H_l^s(k), H_h^s(k))$ has occurred, we get

$$\Pr(A_0(H_m^s(k), H_h^s(k)) \mid A_1(H_l^s(k), H_h^s(k)), \xi(H_h^s(k))) = 0. \quad (47)$$

By applying Bayes' rule, we also get

$$\begin{aligned}\Pr(A_1(H_m^s(k), H_h^s(k)) \mid A_1(H_l^s(k), H_h^s(k)), \xi(H_h^s(k)))\\ = \frac{\chi_1}{\chi_2}, \quad (48)\end{aligned}$$

where

$$\begin{aligned}\chi_1 = &\sum_{i=1}^{N} F_{X_i,\mathbf{X}^{(i)}}(H_h^s(k), H_m^s(k), \ldots, H_m^s(k))\\ &- \sum_{i=1}^{N} F_{X_i,\mathbf{X}^{(i)}}(H_l^s(k), H_m^s(k), \ldots, H_m^s(k)), \quad (49)\end{aligned}$$

$$\begin{aligned}\chi_2 = &\sum_{i=1}^{N} F_{X_i,\mathbf{X}^{(i)}}(H_h^s(k), H_l^s(k), \ldots, H_l^s(k))\\ &- N\sum_{i=1}^{N} F_{\mathbf{X}}(H_l^s(k), \ldots, H_l^s(k)). \quad (50)\end{aligned}$$

Substituting (45) and (46) in (44), we get the expression in (18) for Pr$(T = k, s)$. Comparing (41) and (42) with (18), we see that the above expression holds true for any $s \in \mathcal{S}_k$ including $s = \phi$ and $s = \Omega_k$. This is because the second term in (18) becomes 0, since $H_m^s(k) = 0$ for $s = \phi$ and Ω_k. Hence, Pr$(T = k, s)$ is given by (18) for all paths.

References

[1] A. J. Goldsmith, *Wireless Communications*, 1st ed. Cambridge Univ. Press, 2005.

[2] A. Bletsas, A. Khisti, D. P. Reed, and A. Lippman, “A simple cooperative diversity method based on network path selection,” *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 659–672, Mar. 2006.

[3] D. S. Michalopoulos and G. K. Karagiannidis, “PHY-layer fairness in amplify and forward cooperative diversity systems,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 3, pp. 1073–1082, Mar. 2008.

[4] S. Biswas and R. Morris, “ExOR: Opportunistic multi-hop routing for wireless networks,” in *Proc. ACM SIGCOMM*, Aug. 2005, pp. 133–144.

[5] P. Jacquet, B. Mans, P. Muhlethaler, and G. Rodolakis, “Opportunistic routing in wireless ad hoc networks: Upper bounds for the packet propagation speed,” *IEEE J. Sel. Areas Commun.*, vol. 27, no. 7, pp. 1192–1202, Sep. 2009.

[6] K. Cohen and A. Leshem, “A time-varying opportunistic approach to lifetime maximization of wireless sensor networks,” *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5307–5319, Oct. 2010.

[7] R. S. Blum and B. M. Sadler, “Energy efficient signal detection in sensor networks using ordered transmissions,” *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3229–3235, Jul. 2008.

[8] K. Cohen and A. Leshem, “Energy-efficient detection in wireless sensor networks using likelihood ratio and channel state information,” *IEEE J. Sel. Areas Commun.*, vol. 29, no. 8, pp. 1671–1683, Sep. 2011.

[9] M. C. Vuran, O. B. Akan, and I. F. Akyildiz, “Spatio-temporal correlation: Theory and applications for wireless sensor networks,” *Comput. Netw. J. (Elsevier)*, vol. 45, no. 3, pp. 245–259, Jun. 2004.

[10] D. Xu and Y. Yao, “Splitting tree algorithm for decentralized detection in sensor networks,” *IEEE Trans. Wireless Commun.*, vol. 12, no. 12, pp. 6024–6033, Dec. 2013.

[11] A. Attar, N. Devroye, H. Li, and V. C. M. Leung, “A unified scheduling framework based on virtual timers for selfish-policy shared spectrum,” in *Proc. ICC*, May 2010, pp. 1–5.

[12] V. Shah, N. B. Mehta, and R. Yim, “Optimal timer based selection schemes,” *IEEE Trans. Commun.*, vol. 58, no. 6, pp. 1814–1823, Jun. 2010.

[13] X. Qin and R. Berry, “Opportunistic splitting algorithms for wireless networks,” in *Proc. INFOCOM*, Mar. 2004, pp. 1662–1672.

[14] V. Shah, N. B. Mehta, and R. Yim, “Splitting algorithms for fast relay selection: Generalizations, analysis, and a unified view,” *IEEE Trans. Wireless Commun.*, vol. 9, no. 4, pp. 1525–1535, Apr. 2010.

[15] D. Y. Kim, H. Nam, and H. Jin, “Pseudo-Bayesian broadcasting algorithm for opportunistic splitting scheduling systems,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5450–5455, Jun. 2017.

[16] H. Nam and M. S. Alouini, “Iterative group splitting algorithm for opportunistic scheduling systems,” *IEEE Trans. Mobile Comput.*, vol. 13, no. 5, pp. 1076–1089, May 2014.

[17] A. D. Gore and A. Karandikar, “Power-controlled FCFS splitting algorithm for wireless networks,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 2, pp. 842–856, Feb 2010.

[18] D. To, T. To, and J. Choi, “Energy efficient distributed beamforming with sensor selection in wireless sensor networks,” in *Proc. VTC (Spring)*, May 2012, pp. 1–5.

[19] D. P. Bertsekas and R. G. Gallager, *Data Networks*, 2nd ed. Prentice Hall, 1992.

[20] M. F. A. Ahmed, T. Y. Al-Naffouri, M. S. Alouini, and G. Turkiyyah, “The effect of correlated observations on the performance of distributed estimation,” *IEEE Trans. Signal Process.*, vol. 61, no. 24, pp. 6264–6275, Dec. 2013.

[21] A. Attarkashani and W. Hamouda, “Throughput maximization using cross-layer design in wireless sensor networks,” in *Proc. ICC*, May 2017, pp. 1–6.

[22] B. V. Nguyen, R. O. Afolabi, and K. Kim, “Dependence of outage probability of cooperative systems with single relay selection on channel correlation,” *IEEE Commun. Lett.*, vol. 17, no. 11, pp. 2060–2063, Nov. 2013.

[23] S. Dang, J. P. Coon, and G. Chen, “An equivalence principle for OFDM-based combined bulk/per-subcarrier relay selection over equally spatially correlated channels,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 122–133, Jan. 2017.

[24] R. K. Mallik, “On multivariate Rayleigh and exponential distributions,” *IEEE Trans. Inf. Theory*, vol. 49, no. 6, pp. 1499–1515, Jun. 2003.

[25] K. S. Butterworth, K. W. Sowerby, and A. G. Williamson, “Base station placement for in-building mobile communication systems to yield high capacity and efficiency,” *IEEE Trans. Commun.*, vol. 48, no. 4, pp. 658–669, Apr. 2000.

[26] Q. H. Chu, J. M. Conrat, and J. C. Cousin, “Experimental characterization and modeling of shadow fading correlation for relaying systems,” in *Proc. VTC (Fall)*, Sep. 2011, pp. 1–5.

[27] J. Zhang and V. Aalo, “Effect of macrodiversity on average-error probabilities in a Rician fading channel with correlated lognormal shadowing,” *IEEE Trans. Commun.*, vol. 49, no. 1, pp. 14–18, Jan. 2001.

[28] S. Sesia, I. Toufik, and M. Baker, *LTE – The UMTS Long Term Evolution, From Theory to Practice*, 2nd ed. John Wiley and Sons, 2011.

[29] N. C. Beaulieu and K. T. Hemachandra, “Novel simple representations for Gaussian class multivariate distributions with generalized correlation,” *IEEE Trans. Inf. Theory*, vol. 57, no. 12, pp. 8072–8083, Dec. 2011.

[30] G. L. Stüber, *Principles of Mobile Communications*, 2nd ed. Kluwer Academic Pub., 2001.

[31] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*, 9th ed., ser. Applied mathematics series. Dover Pub., 1964.

[32] A. Genz and F. Bretz, *Computation of Multivariate Normal and t Probabilities*, 1st ed. Springer Pub. Co., 2009.

[33] V. Shah, N. B. Mehta, and R. Yim, “The relay selection and transmission trade-off in cooperative communication systems,” *IEEE Trans. Wireless Commun.*, vol. 9, no. 8, pp. 2505–2515, Aug. 2010.

Reneeta Sara Isaac received her Bachelor of Technology degree in Electronics and Communication Engineering from Rajiv Gandhi Institute of Technology, Kottayam, Kerala in 2014. She is currently pursuing her Master of Science degree in the Department of Electrical Communication Engineering, Indian Institute of Science (IISc), Bangalore. Her research interests include design and analysis of algorithms for communication networks and multiple access protocols.

Neelesh B. Mehta (S’98-M’01-SM’06) received his Bachelor of Technology degree in Electronics and Communications Engineering from the Indian Institute of Technology (IIT), Madras in 1996, and his M.S. and Ph.D. degrees in Electrical Engineering from the California Institute of Technology, Pasadena, USA in 1997 and 2001, respectively. He is a Professor in the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore. He is a Fellow of the Indian National Academy of Engineering (INAE) and the National Academy of Sciences India (NASI). He is a recipient of the Shanti Swarup Bhatnagar Award 2017 and the Swarnjayanti Fellowship. He serves as the Chair of the Executive Editorial Committee of the IEEE Transactions on Wireless Communications, and as an Editor of the IEEE Transactions on Communications. He served on the Board of Governors of IEEE ComSoc from 2012 to 2015.