This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

Analysis, Insights and Generalization of a Fast Decentralized Relay Selection Mechanism

Virag Shah, *Student Member, IEEE*, Neelesh B. Mehta, *Senior Member, IEEE*, Raymond Yim, *Member, IEEE*

***Abstract*—Relay selection for cooperative communications has attracted considerable research interest recently. While several criteria have been proposed for selecting one or more relays and analyzed, mechanisms that perform the selection in a distributed manner have received relatively less attention. In this paper, we analyze a splitting algorithm for selecting the single best relay amongst a known number of active nodes in a cooperative network. We develop new and exact asymptotic analysis for computing the average number of slots required to resolve the best relay. We then propose and analyze a new algorithm that addresses the general problem of selecting the best $Q \geq 1$ relays. Regardless of the number of relays, the algorithm selects the best two relays within 4.406 slots and the best three within 6.491 slots, on average. Our analysis also brings out an intimate relationship between multiple access selection and multiple access control algorithms.**

I. INTRODUCTION

A large body of research work in the area of cooperative communication uses dedicated relays to forward the message from the source to the destination [1]. Depending on the cooperation scheme, this is done using either one relay or multiple relays, which are selected from the many available relays according to some suitability criterion or metric. Several criteria for selecting the single best relay or the best Q relays have been proposed and analyzed in the literature, and shown to result in significant performance gains [2]–[11]. For example, [2] showed that for a decode and forward scheme, selecting the best relay achieves full diversity. In [5], criteria for selecting multiple relays were proposed to minimize data transmission time. In [8], two-relay selection was used to improve a diversity-multiplexing trade-off of an amplify-and-forward protocol. In [9], multiple relay selection was optimized for cooperative beamforming.

The design of the mechanism that physically selects the best relay(s) (as per the selection or suitability criteria) is therefore an important problem in cooperative communications. It is desirable that the mechanism be distributed since typically the knowledge of the suitability metric is initially available only locally at the relay. A centralized polling mechanism, for example, is not desirable for selection since it would require resources that increase linearly with the number of available relays.

In fact, the problem applies just as well to cellular systems, in which the scheduler exploits multi-user diversity by choosing which mobiles transmit or receive. In sensor networks, node selection helps improve network lifetime. If the nodes are energy harvesting, selection helps improve the energy neutral operating region [12].

A decentralized back-off timer-based scheme for single (best) relay selection was proposed in [2]. In it, each node transmits a short message when its back-off timer expires. The timer value is set to be inversely proportional to the metric. Therefore, the node that first transmits is the best node. While this scheme is elegant and simple, the timers can expire such that messages from different nodes overlap with each other and collide at the sink. Consequently, the algorithm does not necessarily scale well as the number of relays increases.

Another approach is to consider time-slotted systems, in which each active node locally decides whether or not to transmit in a certain time slot. Here, splitting algorithms are the method of choice. They have been applied in both multiple access control problems [13] and multiple access selection problems [14], [15]. Note that multiple access control and multiple access selection serve different purposes, and are therefore evaluated differently. Multiple access control attempts to serve all nodes and is evaluated, for example, by the maximum traffic it can handle with finite delay. On the other hand, multiple access selection is evaluated by how fast it can select the best node(s) [15], [16].

For multiple access selection, Qin and Berry [14] developed a splitting-based algorithm that could find the best node within 2.507 slots, on average. Remarkably, this holds regardless of the number of nodes in the system. In the algorithm, only those nodes with their metric lying between two thresholds transmit. The nodes update the thresholds (independently) in each slot based on the outcomes of the previous slots.

In this paper, we generalize the Qin-Berry splitting algorithm by introducing a *contention load* parameter p_e (which is explained in detail later). We then derive novel and simple expressions for the asymptotic behavior of this generalized algorithm as the number of relays increases. The involved analysis in [14] was limited to the specialized case of $p_e = 1$ and primarily led to an upper bound expression that was in the form of an infinite series. Our analysis shows that the parameter choice of [14] can be improved upon. Furthermore, the analysis also leads to a much simpler upper bound and an accurate approximation for the average number of slots required to select the best relay.

The analysis also brings out an intimate relationship between the generalized splitting algorithm and the First Come First Serve (FCFS) multiple access control algorithm [13].

V. Shah and N. B. Mehta are with the Electrical Communication Eng. Dept., Indian Institute of Science (IISc), Bangalore, India. R. Yim is with the Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA.

Emails: {virag4u@gmail.com, nbmehta@ece.iisc.ernet.in, yim@merl.com}.

Based on this we derive an alternate non-recursive asymptotic expression. This, in turn, yields good unimodal lower bounds and, more importantly, motivates a novel scalable and fast algorithm for the general problem of selecting $Q \geq 1$ relays.

The paper is organized as follows. The system model and the basic node selection algorithm are described in Sec. II. The analysis for single node selection is developed in Sec. III. The algorithm for selecting best $Q \geq 1$ nodes is developed in Sec. IV. We conclude in Sec. V. Some mathematical details are relegated to the Appendix.

II. System Model

Consider a time-slotted system with n active nodes and a sink, as shown in Fig. 1. We use the generic term 'sink' to refer to the source or access point or base station, as the case may be, that needs to select the best node/relay. We shall use the terms 'node' and 'relay' interchangeably. Each node i has a suitability metric u_i, which is known only to that specific node. The goal is for the sink to select the node with the highest metric. The metrics are assumed to be i.i.d. with complementary CDF (CCDF) $F_c(u) = \Pr(u_i > u)$. We assume that the metric remains constant during the process of selection, which is very reasonable given the short time required by the algorithm to select the best node.

In the Qin-Berry selection algorithm, at the beginning of each slot, each node determines whether to transmit or not in a distributed manner as detailed below. At the end of each slot, the sink broadcasts one of three outcomes to all the nodes: (i) 0 if the slot was idle (when no node transmitted), (ii) 1 if the outcome was a success (when exactly one node transmitted), and (iii) e if the outcome was a collision (when multiple nodes transmitted).[1]

The algorithm runs independently in each of the contending nodes. It basically determines two thresholds, $H_L(k)$ and $H_H(k)$, for each time slot k. Only those nodes whose metric u satisfies $H_L(k) < u < H_H(k)$ transmit in slot k.[2] It also specifies another variable $H_{\min}(k)$, which is the largest value of the metric known up to slot k above which the best metric surely lies. The algorithm terminates when the outcome is a success (1).

Formally, the algorithm can be defined as follows. Let $\text{split}\,(a,b) = F_c^{-1}\left(\frac{F_c(a)+F_c(b)}{2}\right)$. The split function makes sure that on average half of the nodes involved in the last collision transmit in the next slot.[3] In the first slot, the parameters are initialized as follows: $H_L(1) = F_c^{-1}(1/n)$, $H_H(1) = \infty$, and $H_{min}(1) = 0$. Then, the algorithm in the $(k+1)^{\text{th}}$ slot proceeds as follows [14]:

1) If feedback (of the k^{th} slot) is an idle (0) and no collisions has occurred so far, then set $H_H(k+1) = H_L(k)$, $H_L(k+1) = F_c^{-1}(\frac{k+1}{n} + O(\frac{1}{n^2}))$[4] and $H_{\min}(k+1) = H_{\min}(k)$.
2) If feedback is a collision (e), then set $H_H(k+1) = H_H(k)$, $H_{\min}(k+1) = H_L(k)$ and $H_L(k+1) = \text{split}\,(H_L(k), H_H(k))$.
3) If feedback is an idle (0) and a collision has occurred in the past, then set $H_H(k+1) = H_L(k)$, $H_{\min}(k+1) = H_{\min}(k)$ and $H_L(k+1) = \text{split}\,(H_{\min}(k), H_L(k))$.

We will call the duration of the algorithm before the first non-idle slot as the *idle phase*, and after as the *collision phase*.

[1]The sink can distinguish between these outcomes using, for example, the strength of the total received power [15].

[2]The case where two nodes have exactly the same metric occurs with probability 0, and does not need to be accounted for. In practice, the algorithm can be terminated after some time if it fails to select the best node.

[3]In [17], a faster but complex splitting technique was presented, improvement due which can be shown to be less than 0.5%, and hence is not considered here.

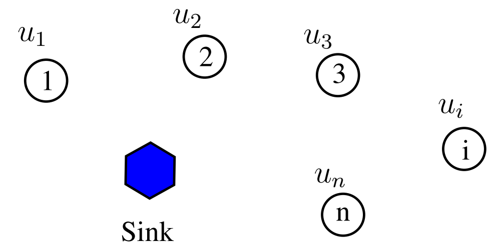

Fig. 1. A relay selection system consisting of a sink and n relays/nodes, with a node i possessing a suitability metric u_i.

III. Single Relay Selection

A. Asymptotic Analysis and Optimal Thresholds

We consider a simple generalization of the Qin-Berry algorithm by introducing the contention-load parameter p_e. During the idle phase, the minimum metric threshold for transmission $H_L(k) = F_c^{-1}(kp_e/n)$ at slot k. It is easy to see that the original algorithm, described in Sec. II, corresponds to $p_e = 1$. Using $p_e = 1$ greedily maximizes probability of success in each slot of the idle phase. However, as we shall see from our asymptotic analysis, this is not optimal.

First we find the expression for the average number of slots required by the algorithm for n nodes and parameter p_e.

Lemma *1:* The average number of slots, $m_n(p_e)$, required to select the best node is given by,

$$m_n(p_e) = \sum_{i=1}^{q}\sum_{k=1}^{n}\binom{n}{k}\left(\frac{p_e}{n}\right)^k\left(1-\frac{ip_e}{n}\right)^{n-k}(\mathbf{E}[X_k]+i) + \left(1-\frac{qp_e}{n}\right)^n(\mathbf{E}[X_n]+q+1), \quad (1)$$

where $q = \left\lceil \frac{n}{p_e} \right\rceil - 1$, $\lceil . \rceil$ is the ceil function, $\mathbf{E}[X_1] = 0$ and $\mathbf{E}[X_k] = \frac{0.5^k\left(\sum_{l=2}^{k-1}\binom{k}{l}\mathbf{E}[X_l]\right)+1}{1-0.5^{k-1}}$; $\forall\ k \geq 2$.

Proof: The proof is given in Appendix A. ■

The above expression is quite complex and does not directly reveal the scalable nature of the algorithm. For $p_e = 1$, it was proved in [14] that $\lim_{n\to\infty} m_n(1) \triangleq m_\infty(1) \leq 2.507$. The theorem below provides an exact and new expression for the asymptotic case for a general p_e.

[4]We shall henceforth ignore the $O(1/n^2)$ term that arises in setting the thresholds as its effect is known to be quite negligible even when the number of nodes is as low as 7.

Theorem *1:*

$$m_\infty(p_e) = \frac{1}{e^{p_e}-1}\sum_{k=1}^{\infty}\frac{\mathbf{E}\left[X_k\right]p_e^k}{k!} + \frac{1}{1-e^{-p_e}}. \quad (2)$$

Proof: Let node i have metric u_i with CCDF $F_c(u)$. Let $x_i = F_c(u_i)$. Then, x_i are i.i.d. and uniformly distributed in $[0,1]$.[5] Further, let $y_i = nx_i$. Clearly, y_i are i.i.d. and are uniformly distributed in $[0,n]$. Sorting the $\{x_i\}_{i=1}^n$ and $\{y_i\}_{i=1}^n$ in ascending order, we get $x_{[1]} \leq x_{[2]} \leq x_{[3]} .... \leq x_{[n]}$ and $y_{[1]} \leq y_{[2]} \leq y_{[3]} .... \leq y_{[n]}$, where $[i]$ is the index of the relay with the i^{th} largest metric.

Given y_i, we can define a point process $M(t)$ as $M(t) = \sup\left\{k \geq 1 : y_{[k]} \leq t\right\}$. Since the $\{y_i\}_{i=1}^n$ are uniformly distributed, $M(t)$ has a binomial distribution with number of trials as n and the probability of success equal to $\frac{t}{n}$. As $n \to \infty$, it can be shown that $M(t)$ forms a Poisson point process with rate 1 [18].

The probability that the first non-idle slot is the i^{th} slot and $k \geq 1$ nodes are involved is therefore equal to the probability that $x_{[1]}, \ldots, x_{[k]}$ lie between $(i-1)p_e/n$ and ip_e/n, and $x_{[j]} > ip_e/n$, for $k+1 \leq j \leq n$. This probability then equals

$$\begin{aligned}
&\Pr\left(x_{[1]} > (i-1)\frac{p_e}{n}, (i-1)\frac{p_e}{n} < x_{[k]} < i\frac{p_e}{n}, x_{[k+1]} > i\frac{p_e}{n}\right),\\
&= \Pr\left(M((i-1)p_e) = 0\right)\Pr\left(M(ip_e) = k \,|M((i-1)p_e) = 0\right),\\
&= e^{-(i-1)p_e}\ e^{-p_e}\frac{p_e^k}{k!} = e^{-ip_e}\frac{p_e^k}{k!}.
\end{aligned}$$

Let $\mathbf{E}\left[X_k\right]$ denote the expected number of slots required to resolve a collision among k nodes. Thus, if the first non-idle slot is the i^{th} slot and $k \geq 1$ nodes are involved, then $\mathbf{E}\left[X_k\right] + i$ slots are required to find the best node. Also, as $n \to \infty$, $qp_e/n \to 1$. Hence, we get

$$\begin{aligned}
m_\infty(p_e) &= \sum_{i=1}^{\infty}\sum_{k=1}^{\infty} e^{-ip_e}\frac{p_e^k}{k!}\left(\mathbf{E}\left[X_k\right] + i\right),\\
&= \sum_{k=1}^{\infty}\frac{p_e^k}{k!}\left(\mathbf{E}\left[X_k\right]\sum_{i=1}^{\infty}e^{-ip_e} + \sum_{i=1}^{\infty} ie^{-ip_e}\right).
\end{aligned}$$

Simplifying the above expression yields the desired result. ■

A key insight of the analysis above is its Poisson point process interpretation. In effect, in the asymptotic regime, what the selection algorithm does is to run the FCFS algorithm [13] on the Poisson process, $M(t)$, where t is interpreted as time. However, unlike FCFS, the selection algorithm stops as soon as it finds the first (best) node. This interpretation enables the use of an approach similar to that in [13, Chp. 4], and leads to an alternate and novel expression for $m_\infty(p_e)$ and also better lower bounds. As we shall see, it also motivates the Q node selection algorithm in the next section. Instead of indexing on number of nodes involved in the first non-idle slot, as done in Theorem 1, the alternate derivation below, in effect, indexes on the number of slots required after the first non-idle slot.

[5] Note that higher value of u_i implies lower value of x_i since the CCDF is a monotonically decreasing function. Thus, selecting the node with the highest u_i is equivalent to selecting the node with the lowest x_i.

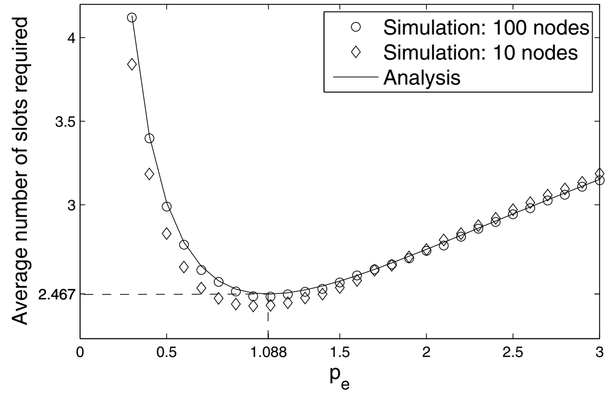

Fig. 2. Average number of slots required to select the best node $(m_\infty(p_e))$ as a function of the contention load parameter p_e.

Theorem *2:*

$$m_\infty(p_e) = \frac{1}{1-e^{-p_e}} + \sum_{i=1}^{\infty} p(i), \quad (3)$$

where $p(i) = (1-P_0)\prod_{j=1}^{i-1}(1-P_j)$, $P_0 = \frac{p_e e^{-p_e}}{1-e^{-p_e}}$, $P_i = \frac{G_i e^{-G_i}(1-e^{-G_i})}{1-(1+G_{i-1})e^{-G_{i-1}}}$, $\forall\ i \geq 1$, and $G_i = 2^{-i}p_e$.

Proof: The proof is given in Appendix B. ■

Figure 2 plots the average number of slots required to select the best node as a function of p_e as obtained from analysis and from Monte Carlo simulations. It can be seen that the asymptotic expression is accurate for even a small number of nodes (e.g., 10). Furthermore, the optimal value of p_e is 1.088, and the minimum value of $m_\infty(p_e)$ is 2.467. This being said, setting p_e as 1, as used in [14], is not a bad choice since it is close to the optimal value.

B. Bounds and Approximation

Both the formulae derived thus far for $m_\infty(p_e)$ involve an infinite series. It is, therefore, desirable to simplify them.

Upper bound: We now derive a simple upper bound and an accurate approximation for $m_\infty(p_e)$, both of which capture the behavior of the exact formula well and do not involve any infinite series. As we shall see, they are also convex, which is desirable since it enables the use of well-developed and computationally efficient numerical algorithms to optimize the system.

Corollary *1:*

$$m_\infty(p_e) \leq \frac{p_e}{k_0\log_e(2)} + \log_2(2k_0/e) + \frac{1}{1-e^{-p_e}}, \quad (4)$$

for any $k_0 > 0$.

Proof: The proof is given in Appendix C. ■

Putting $k_0 = 2$, we get the following upper bound:

$$m_\infty(p_e) \leq \frac{p_e}{2\log_e(2)} + (2 - \log_2(e)) + \frac{1}{1-e^{-p_e}}. \quad (5)$$

Lower bound: Since the expressions derived in Theorems 1 and 2 both involve an infinite series of positive terms, lower

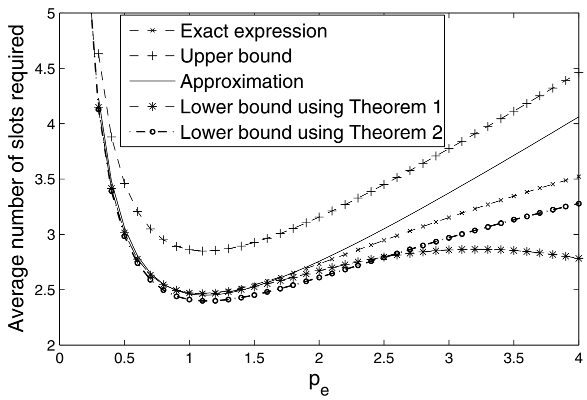

Fig. 3. Upper and lower bounds and an approximation for the average number of slots required to select the best node.

bounds can be obtained by considering only the first few terms of the series. Figure 3 compares the lower bounds obtained using the first 4 terms. It can be seen that Theorem 2 leads to lower bounds that better exhibit the same unimodal property of the exact expression, than Theorem 1.

Approximation: It can be easily verified that $\mathbf{E}[X_k] \approx \log_2(k) + 0.6$ tracks $\mathbf{E}[X_k]$ better than the upper bound $\log_2(k) + 1$. (The figure showing the accuracy of the approximation is not shown here for lack of space.) This leads to the following approximation:

$$m_\infty(p_e) \approx \frac{p_e}{2\log_e(2)} + (1.6 - \log_2(e)) + \frac{1}{1 - e^{-p_e}}. \quad (6)$$

The upper bound (with $k_0 = 2$) and the approximation are plotted in Fig. 3. Being convex, they both have a unique minimum and follow the behavior of the exact expression well in the region of interest of p_e. The approximation is quite accurate for p_e as large as 2.5.

IV. Best Q Relay Selection Algorithm

We now propose a new algorithm for selecting not just the single best relay but $Q \geq 1$ relays in general.

A. Algorithm Definition

We first state the algorithm, and then describe the logic behind its steps. For this, we adopt the notation of FCFS, as it is more convenient.

As in Sec. III-A, let $y_i = nF_c(u_i)$. The algorithm specifies $T(k)$, $\alpha(k)$, and $\sigma(k)$, for each slot k. $(T(k), T(k) + \alpha(k))$ represents the allocation interval for slot k, *i.e.*, all the nodes with $y_i \in (T(k), T(k) + \alpha(k))$ transmit in slot k. (Equivalently, $H_H(k) = F_c^{-1}(T(k)/n)$ and $H_L(k) = F_c^{-1}((T(k) + \alpha(k))/n)$.) $\sigma(k) \in \{L, R\}$ indicates whether the k^{th} slot interval is the left half or the right half of the previously split interval. In addition, $S(k)$ tracks how many nodes have been selected before slot k. For the first slot, the variables are initialized as follows: $T(1) = 0$, $\alpha(1) = p_e$, $S(1) = 0$, and $\sigma(1) = L$. In the $(k+1)^{\text{th}}$ slot ($k \geq 1$):

1) If feedback is a collision (e), then $T(k+1) = T(k)$, $\alpha(k+1) = \alpha(k)/2$, and $\sigma(k+1) = L$.
2) If feedback is a success (1) and $\sigma(k) = L$, then $T(k+1) = T(k) + \alpha(k)$, $\alpha(k+1) = \alpha(k)$, and $\sigma(k+1) = R$.
3) If feedback is an idle (0) and $\sigma(k) = L$, then $T(k+1) = T(k) + \alpha(k)$, $\alpha(k+1) = \alpha(k)/2$, and $\sigma(k+1) = L$.
4) If feedback is an idle (0) or a success (1) and $\sigma(k) = R$, then $T(k+1) = T(k) + \alpha(k)$, $\alpha(k+1) = p_e$, and $\sigma(k+1) = R$.
5) Increment $S(k+1)$ by 1 if feedback is a success (1). Terminate if $S(k+1)$ reaches Q.

The generalized algorithm of Sec. III is a special case of the aforementioned algorithm when $Q = 1$. It is similar to FCFS, except that it stops after the Q^{th} success.[6] Briefly, the logic behind the algorithm is as follows. When a collision occurs, the allocation interval for the next slot is always the left (L) half of that of the present slot. When a success follows a collision, the allocation interval for the next slot is the right half (R) of the previously split interval. When an idle follows a collision, it implies that the previous rightmost interval contains at least two nodes, which is why it is split next into two equal halves. When there is no collision to be resolved, the algorithm moves to the adjacent allocation interval of size p_e. As mentioned above, the algorithm terminates after Q successes.

B. Algorithm Analysis

Let $m_\infty^{(Q)}(p_e)$ be the average number of slots required to select the best Q nodes. (Thus, the symbol $m_\infty(p_e)$, which was used in the previous section on single relay selection, is equivalent to $m_\infty^{(1)}(p_e)$.) Let $\mathbf{E}\left[X_k^{(Q)}\right]$ denote the average number of slots required to select the best Q nodes after k nodes collide. Then, the exact expression for $m_\infty^{(Q)}(p_e)$ is given by the following theorem.

Theorem *3:* As $n \to \infty$, the average number of slots required to select the best $Q > 1$ nodes is

$$m_\infty^{(Q)}(p_e) = \frac{1}{e^{p_e} - 1} \sum_{k=1}^{\infty} \frac{\mathbf{E}\left[X_k^{(Q)}\right] p_e^k}{k!} + \frac{1}{1 - e^{-p_e}}, \quad (7)$$

where,

$$\mathbf{E}\left[X_k^{(Q)}\right] = (1 - 0.5^{k-1})^{-1} \Bigg(0.5^k \Bigg(\sum_{i=2}^{k-1} \binom{k}{i} \mathbf{E}\left[X_i^{(Q)}\right] + k\left(1 + \mathbf{E}\left[X_{k-1}^{(Q-1)}\right]\right) \Bigg) + 1 \Bigg), \ \forall\ k \geq 3, \quad (8)$$

[6]Note that there does exist one difference between FCFS and the proposed algorithm. The initial contention resolution interval duration of FCFS can be smaller if the difference between the current time and the time of the last resolved interval is small. In our algorithm, however, the contention interval in step (4) is always incremented by p_e as nodes know their metrics a priori.

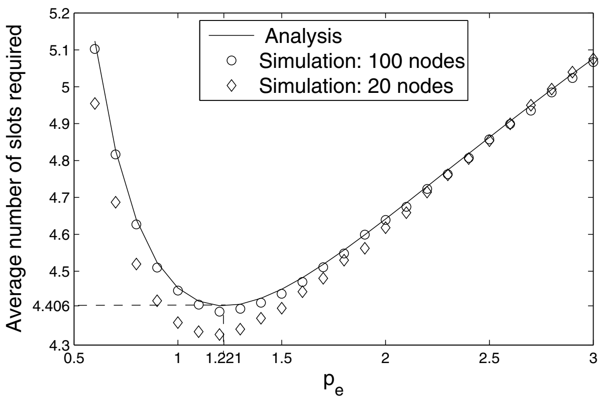

Fig. 4. Average number of slots required to select the best two nodes ($m_{\infty}^{(2)}(p_e)$) as a function p_e.

TABLE I
OPTIMUM p_e AND THE AVERAGE NUMBER OF SLOTS REQUIRED TO SELECT THE BEST Q NODES

Q	Optimum p_e	Optimum $m_{\infty}^{(Q)}(p_e)$ (slots)
1	1.088	2.467
2	1.221	4.406
3	1.214	6.491
4	1.231	8.537
5	1.236	10.592
6	1.241	12.645

$\mathbf{E}\left[X_2^{(Q)}\right] = m_{\infty}^{(Q-2)}(p_e) + 3,\ \forall\ Q > 2$, $\mathbf{E}\left[X_2^{(2)}\right] = 3$, and $\mathbf{E}\left[X_1^{(Q)}\right] = m_{\infty}^{(Q-1)}(p_e)$.

Proof: The proof is given in Appendix D. ■

Figure 4 plots $m_{\infty}^{(2)}(p_e)$ as a function of p_e using the expression of Theorem 3 and verifies it using Monte Carlo simulations (with $n = 100$). The lowest average number of slots required to select two nodes is 4.406, which occurs at $p_e = 1.221$. This is an improvement of 21.4% over the algorithm that applies single relay selection twice to select two nodes and takes $2 \times 2.467 = 4.934$ slots.

Notice that the optimum value of p_e that minimizes the average selection time increase from 1.088, for single relay selection, to 1.221 for two relay selection. This can be understood as follows. The time taken to select two nodes given that they are involved in a collision is $\mathbf{E}\left[X_2^{(2)}\right] = 3.0$ slots. Whereas, the number of slots required to select two nodes, given that the previous slot was idle, increases by 46% to 4.4 slots. Consequently, setting $p_e > 1$ is beneficial since it is faster to resolve a collision than to avoid it.

The optimum values of p_e and the average number of slots as a function of the number of relays that need to be selected is given in Table I. We can see that selecting the three best nodes takes 6.491 slots, on average, and is achieved when $p_e = 1.214$. The decrease in the optimal value of p_e (from 1.221 to 1.214) when Q increases from 2 to 3 can be explained as follows. The time taken to select three nodes after a collision among two nodes is $\mathbf{E}\left[X_2^{(3)}\right] = 5.484$ slots. However, the number of slots required to select three nodes after an idle slot, is 6.49 slots, which is just 17.8% more than 5.484. Therefore, the optimum p_e decreases since the selection times after idle and collision phases are not as unequal as for $Q = 2$. As $Q \to \infty$, the optimum value of p_e increases to 1.266, which is also the optimum value for FCFS [13]. In this case, 0.487 nodes per slot get selected on average.

V. CONCLUSIONS

We presented a generalized splitting algorithm for multiple access selection, and optimized it for single best relay selection. Furthermore, we extended the algorithm for selecting the best $Q \geq 1$ relays. Regardless of the total number of relays in the system, on average, only 2.467 slots were required to select the best relay. And, each additional relay can be selected in even less additional time. In our analysis, a key Poisson point process interpretation of the dynamics of the algorithm was used to derive a simple asymptotic result for computing the average number of slots. Such an asymptotic result can be applied to obtain an accurate approximation even when the number of relays is small. As Q increased, the optimal value of the contention load parameter mostly increased and finally approached 1.266, which is the maximum expected number of packets in the initial contention window in the FCFS multiple access control algorithm. Our analysis thus shows an intimate relationship between multiple access selection and multiple access control (e.g., FCFS) algorithms.

APPENDIX

A. Proof of Lemma 1

It can be easily seen that the idle phase can consist of a maximum of $q = \left\lceil \frac{n}{p_e} \right\rceil - 1$ slots. Let X_k be the number of slots required to resolve a collision among k nodes. Given that the first non-idle slot is the i^{th} slot and k nodes are involved, the average number of slots required to find the best node is $\mathbf{E}[X_k] + i$. (The expression for $\mathbf{E}[X_k]$ is given in [14, (6)].) The probability that the first non-idle slot is the i^{th} slot and k nodes transmit in it equals $\binom{n}{k}\left(\frac{p_e}{n}\right)^k\left(1 - \frac{ip_e}{n}\right)^{n-k}$, for $i \leq q$. For the $(q+1)^{\text{th}}$ slot, the probability that it is the first non-idle slot and k nodes are involved in it is $(1 - \frac{qp_e}{n})^n$, for $k = n$, and is 0 otherwise. Hence, the result follows.

B. Proof of Theorem 2

Let the random variable I denote the first non-idle slot. Then, from the Poisson process interpretation of Theorem 1, we can show that $\Pr(I = i) = e^{-(i-1)p_e}(1-e^{-p_e})$. Therefore, $\mathbf{E}[I] = \sum_{i=1}^{\infty} ie^{-(i-1)p_e}(1 - e^{-p_e}) = \frac{1}{1-e^{-p_e}}$.

Now, we find average number of slots required after the first non-idle slot. Consider the state transition diagram of Figure 5, where the state is the number of slots, Y, required after the first non-idle slot to select the best node. The node goes to state $\mathcal{S}$ whenever success occurs, and the algorithm terminates. It can be shown that the transition probability from state i to

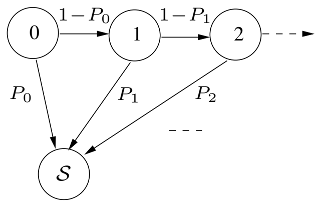

Fig. 5. State transition diagram for the number of slots required to select the best node after the first non-idle slot.

$\mathcal{S}$ is dependent only on i. We henceforth denote it by P_i. Therefore, this is a Markov chain.

The average number of slots required (after the first non-idle slot) to select the best node is $\mathbf{E}[Y] = \sum_{i=1}^{\infty} i \Pr(Y=i)$, which can be shown to be equal to $\sum_{i=1}^{\infty} \Pr(Y \geq i)$. Using the fact that each state is visited at most once, we can show that $\mathbf{E}[Y] = \sum_{i=1}^{\infty} p(i)$, where $p(i)$ is the probability that the i^{th} state is visited. Consequently, from the state transition diagram, we get $p(i) = (1-P_0)\prod_{j=1}^{i-1}(1-P_j)$. Here, P_0 is the probability that the first non-idle slot is a success. It is the probability that in a slot of size p_e only one node transmits given that at least one node transmits in that slot. Hence, it equals $\frac{p_e e^{-p_e}}{1-e^{-p_e}}$. Similarly, the expressions for $P_i,\ \forall\ i \geq 1$, can also be derived [13, Chap. 4, (4.23)]. Finally, the desired result follows from $m_\infty(p_e) = \mathbf{E}[I] + \mathbf{E}[Y]$.

C. Proof of Corollary 1

Since $\log_2(x)$ is concave with respect to x, a tangent to it at any point $(k_0, \log_2(k_0))$ is an upper bound. This leads to the inequality $\log_2(k) \leq \frac{k-k_0}{k_0 \log_e(2)} + \log_2(k_0)$. From [14], we have $\mathbf{E}[X_k] \leq \log_2(k) + 1$ and $\mathbf{E}[X_1] = 0$. Substituting these results in Theorem 1 and using the inequality $e^{p_e} - 1 - p_e \leq e^{p_e} - 1$ leads to the desired result.

D. Proof of Theorem 3

Given that the first non-idle slot is the i^{th} slot and $k \geq 1$ nodes are involved, the average number of slots required to select the best Q nodes is $\mathbf{E}\left[X_k^{(Q)}\right] + i$. The probability that the first non-idle slot is the i^{th} slot and $k \geq 1$ nodes are involved is $e^{-ip_e} p_e^k / k!$. Hence, we get

$$m_\infty^{(Q)}(p_e) = \sum_{i=1}^{\infty}\sum_{k=1}^{\infty} e^{-ip_e}\frac{p_e^k}{k!}\left(\mathbf{E}\left[X_k^{(Q)}\right] + i\right),$$

simplifying which yields equation (7).

If only one node transmits (success) in the first non-idle slot, then selecting the remaining $Q-1$ nodes will take $m_\infty^{(Q-1)}(p_e)$ slots, on average. (This follows from the independent increments property of the Poisson process [18].) Thus, $\mathbf{E}\left[X_1^{(Q)}\right] = m_\infty^{(Q-1)}(p_e)$. Also, if two nodes transmit in the first non-idle slots, $\mathbf{E}\left[X_2^{(2)}\right] = \mathbf{E}\left[X_1^{(2)}\right] + 1 = 3$ slots are required, on average, to select both of them. Selecting the remaining $Q-2$ nodes takes another $m_\infty^{(Q-2)}(p_e)$ slots, on average. Thus, $\mathbf{E}\left[X_2^{(Q)}\right] = \mathbf{E}\left[X_2^{(2)}\right] + m_\infty^{(Q-2)}(p_e)$. When $k > 3$ nodes transmit in the first non-idle slot, the following three cases are possible for the next slot: (i) *Collision among i nodes*: $\mathbf{E}\left[X_i^{(Q)}\right]$ more slots would then be required, on average. (ii) *Idle*: $\mathbf{E}\left[X_k^{(Q)}\right]$ more slots are required, on average. (iii) *Success*: The next slot would surely involve a collision among $k-1$ nodes. $\mathbf{E}\left[X_{k-1}^{(Q-1)}\right]$ slots, on average, would be required *after* that. The probability that i nodes transmit in the next slot is $0.5^k\binom{k}{i}$. Thus,

$$\mathbf{E}\left[X_k^{(Q)}\right] = 0.5^k\Bigg(\left(\binom{k}{0}+\binom{k}{k}\right)\left(1+\mathbf{E}\left[X_k^{(Q)}\right]\right) + \binom{k}{1}\left(1+\mathbf{E}\left[X_{k-1}^{(Q-1)}\right]+1\right) + \sum_{i=2}^{k-1}\binom{k}{i}\left(1+\mathbf{E}\left[X_i^{(Q)}\right]\right)\Bigg).$$

Further simplifications result in (8).

References

[1] A. Nosratinia, T. E. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Commun. Mag.*, vol. 42, pp. 74–80, Oct. 2004.

[2] A. Bletsas, A. Khisti, D. P. Reed, and A. Lippman, "A simple cooperative diversity method based on network path selection," *IEEE J. Select. Areas Commun.*, vol. 24, pp. 659–672, Mar. 2006.

[3] B. Zhao and M. C. Valenti, "Practical relay networks: a generalization of hybrid-ARQ," *IEEE J. Select. Areas Commun.*, vol. 23, pp. 7–18, Jan. 2005.

[4] Z. Lin and E. Erkip, "Relay search algorithms for coded cooperative systems," in *Proc. Globecom*, pp. 1314–1319, 2005.

[5] S. Nam, M. Vu, and V. Tarokh, "Relay selection methods for wireless cooperative communications," in *Proc. Conf. on Inform. Sciences and Systems (CISS)*, pp. 859–864, 2008.

[6] H. Yang, A. P. Petropulu, and T. Camp, "A novel location relay selection scheme for ALLIANCES," *IEEE Trans. Veh. Techol.*, vol. 57, pp. 1272–1284, Mar. 2008.

[7] J. Luo, R. S. Blum, L. J. Cimini, L. J. Greenstein, and A. M. Haimovich, "Link-failure probabilities for practical cooperative relay networks," in *Proc. Veh. Techol. Conf. (Spring)*, pp. 1489–1493, 2005.

[8] S. Yang and J.-C. Belfiore, "A novel two-relay three-slot amplify-and-forward cooperative scheme," in *Proc. Conf. on Inform. Sciences and Systems (CISS)*, pp. 1329–1334, 2005.

[9] R. Madan, N. B. Mehta, A. F. Molisch, and J. Zhang, "Energy-efficient cooperative relaying over fading channels with simple relay selection," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 3013–3025, Aug. 2008.

[10] E. Beres and R. Adve, "Selection cooperation in multi-source cooperative networks," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 118–127, Jan. 2008.

[11] D. S. Michalopoulos and G. K. Karagiannidis, "PHY-layer fairness in amplify and forward cooperative diversity systems," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 1073–1083, Mar. 2008.

[12] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," Technical Report TR-UCLA-NESL-200603-02, Networked and Embedded Systems Laboratory, UCLA, Mar. 2006.

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

[14] X. Qin and R. Berry, "Opportunistic splitting algorithms for wireless networks," in *Proc. INFOCOMM*, pp. 1662–1672, 2004.

[15] R. Yim, N. B. Mehta, and A. F. Molisch, "Best node selection through distributed fast variable power multiple access," in *Proc. ICC*, pp. 5028–5032, 2008.

[16] V. Anantharam and P. Varaiya, "An Optimal Strategy For a Conflict Resolution Problem," in *Proc. 24th Conf. on Decision and Control*, pp. 1113–1114, 1985.

[17] A. Cohen, M. Kam, and R. Conn,"Partitioning a Sample Using Binary-Type Questions with Ternary Feedback," *IEEE Trans. Systems, Man, And Cybernetics*, vol. 25, pp. 1405–1408, Oct. 1995.

[18] R. W. Wolff, *Stochastic modeling and the theory of queues*. Prentice Hall, 1989.