Splitting Algorithms for Fast Relay Selection: Generalizations, Analysis, and a Unified View

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

Abstract—Relay selection for cooperative communications promises significant performance improvements, and is, therefore, attracting considerable attention. While several criteria have been proposed for selecting one or more relays, distributed mechanisms that perform the selection have received relatively less attention. In this paper, we develop a novel, yet simple, asymptotic analysis of a splitting-based multiple access selection algorithm to find the single best relay. The analysis leads to simpler and alternate expressions for the average number of slots required to find the best user. By introducing a new 'contention load' parameter, the analysis shows that the parameter settings used in the existing literature can be improved upon. New and simple bounds are also derived. Furthermore, we propose a new algorithm that addresses the general problem of selecting the best $Q \geq 1$ relays, and analyze and optimize it. Even for a large number of relays, the scalable algorithm selects the best two relays within 4.406 slots and the best three within 6.491 slots, on average. We also propose a new and simple scheme for the practically relevant case of discrete metrics. Altogether, our results develop a unifying perspective about the general problem of distributed selection in cooperative systems and several other multi-node systems.

Index Terms—Relays, cooperative communications, selection, multiple access, splitting.

I. INTRODUCTION

SELECTION mechanisms arise in many wireless communication schemes that use most suitable candidates from among a set of many candidates. A pertinent example is a cooperative communication system that exploits spatial diversity by selecting the best relay(s) to forward a message from a source to a destination. Selection makes cooperation practical because it simplifies the tight synchronization that is required among many geographically distributed cooperating relays [1]–[11]. Another example is a cellular system that schedules in a proportional fair manner to the best mobile station based on the average data rate and the current state of the channel between the base station and the mobiles [12].

QoS requirements can also be incorporated in the selection metric, as is done, for example, in a wireless local area network (WLAN). In sensor networks, node selection is known to improve network lifetime.

Several relay selection criteria have been proposed and analyzed in the literature. For example, [1] showed that for a decode-and-forward cooperation scheme, best relay selection achieves full diversity. In [3], criteria for selecting multiple relays were proposed to minimize data transmission time. In [10] relay subset selection was considered for rate maximization. In [6], best two relay selection was used to improve the diversity-multiplexing tradeoff of an amplify-and-forward protocol. In [7], multiple relay selection was optimized for cooperative beamforming. Multiple relay selection for wireless network coding was considered in [13].

The design of the mechanism that physically selects – as per the selection or suitability criteria – the best relay or, in general, the Q best relays is, therefore, an important problem. Depending on the transmission scheme, the suitability metric can be a function of both the source-relay and relay-destination channel gains or just the relay-destination or source-relay channel gains. It is desirable that the mechanism be distributed since, typically, the knowledge of the metric is initially available only locally at the relay. For example, a centralized polling mechanism for selection is undesirable as the time to select increases linearly with the number of available relays. To this end, a decentralized back-off timer-based scheme for single best relay selection were proposed in [1]. In it, each node transmits a short message when its timer expires. Making the timer value inversely proportional to the metric ensures that the first node that the sink hears from is the best node. A distributed single relay selection algorithm was also proposed in [14] to minimize the bit error rate. In [15], the source uses handshake messages from relays to track the rate that each candidate relay can support.

An alternate approach considers a time-slotted multiple access contention based algorithm in which each active node locally decides whether or not to transmit in a certain time slot. Recently, variations based on splitting algorithms, which were extensively researched two decades ago for multiple access control [16, Chp. 4], have been proposed for single relay selection [17], [18]. In each step of the splitting-based selection algorithm proposed in [17], only those nodes whose metrics lie between two thresholds transmit. The nodes update the thresholds (independently) in each slot based on the outcome of the previous slot fed back by the sink.[1] It was

[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.

shown in [17] for continuous metrics that the best node can be found, on average, within at most 2.507 slots even for an infinite number of nodes. This result was obtained by deriving an upper bound on the average number of slots when the number of relays tends to infinity. However, the analysis was quite involved and the upper bound was in the form of an infinite series.

While distributed selection mechanisms have, thus, been proposed for single relay selection, several questions remain open. For example, developing a comprehensive analysis of the splitting mechanism is an important open problem. A natural question that such an analysis will answer is how to optimally choose the thresholds to improve the speed of selection. In [17], the thresholds are initially set greedily so to maximize the probability of success per slot. As we show, this is not optimal. Furthermore, efficient mechanisms are yet to be developed for multiple relay selection. The only option known currently is to run the single relay selection algorithm multiple times, which, as we show in this paper, is inefficient. Finally, the mechanisms above assume that the selection metric is continuous, and exploit the fact that, with probability 1, no two relays have the same metric. The mechanism catastrophically breaks down when the metrics are discrete, which can often occur in practice. This occurs, for example, when the estimation inaccuracy renders higher resolution representations unnecessary, or when quantized metrics for feedback or QoS are considered [11], [19].

This paper thoroughly examines splitting-based selection algorithms for both continuous and discrete metrics, and makes the following significant contributions:

- *Analysis of single relay selection:* The paper develops a novel and considerably simpler *exact* asymptotic analysis for a general version of the splitting algorithm. It achieves this by developing a different Poisson process interpretation of the metric distribution, which has not been used before to the best of our knowledge. Furthermore, it also derives a new convex and simple upper bound for the average number of slots required to select the best relay.
- *Optimization of single relay selection:* The paper analytically determines the optimal performance of the splitting algorithm. It also rigorously shows that the greedy parameter choice of [17] is suboptimal, but is still very good.
- *An alternate Markovian analysis:* The Poisson process interpretation also leads to an alternate and novel Markovian analysis, which among other things yields a new alternate asymptotic expression for the average number of slots. As we shall see, while the two new expressions derived in this paper are equivalent, they exhibit different behaviors when truncated for numerical computation purposes.
- *New mechanism for multiple relay selection, including its analysis and optimization:* The paper proposes a novel scalable, fast, and decentralized algorithm for the general problem of selecting not just the single best but the best $Q \geq 1$ relays. To the best of our knowledge, this is the fastest family of Q relay selection algorithms proposed to date. We develop an asymptotic analysis of the general Q relay selection algorithm, and determine its optimal
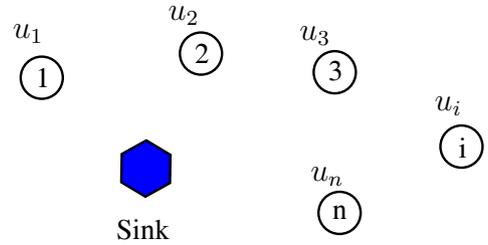
parameters. We show that as Q increases, the greedy parameter choice becomes more suboptimal. In effect, as Q increases, the optimal splitting algorithm prefers that more nodes collide since it is faster to resolve a collision than to avoid one.

- *Unifying perspective on splitting-based selection algorithms:* The paper shows that the optimized best relay selection algorithm, the proposed multiple relay selection mechanism, and Gallager's First Come First Serve (FCFS) multiple access control algorithm [16] are intimately related.
- *A new scalable algorithm for discrete metrics:* Finally, the paper proposes a novel, scalable, and an intuitive distributed scheme called *Proportional Expansion*, which enables the single and multiple relay selection algorithms to be applied to the practical case of discrete metrics.

The rest of the paper is organized as follows. The analysis and results for single best node selection is developed in Sec. II. The new algorithm for $Q \geq 1$ node selection is proposed, analyzed, and simulated in Sec. III. We conclude in Sec. IV. Several mathematical proofs are relegated to the Appendix.

II. SINGLE RELAY SELECTION

A. System Setup

Consider a time-slotted system with n active nodes and a sink, as shown in Fig. 1. Each node i has a suitability metric u_i, which is known only to that specific node. In this section, the goal is to select the node with the highest metric. The metrics are continuous and i.i.d. with complementary CDF (CCDF) denoted by $F_c(u) = \Pr(u_i > u)$. Therefore, the $F_c(.)$ is monotonically decreasing and invertible. (The discrete metric case, where this is not so, is tackled in Sec. III-F.)

B. Splitting Algorithm: Brief Review and Notation

We now formally define the splitting algorithm for single relay selection. To keep the treatment concise, we first define the state variables maintained by the algorithm and their initialization. Thereafter, we describe how the algorithm controls the transmissions of the nodes, how the sink generates feedback based on these transmissions, and how the state variables get autonomously updated based on the feedback.

Definitions: The generalized best relay selection algorithm is specified using three variables $H_L(k)$, $H_H(k)$, and $H_{\min}(k)$; the notation being consistent with that in [17]. $H_L(k)$ and $H_H(k)$ are the lower and upper metric thresholds,

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

respectively, such that a node i transmits at time slot k if and only if its metric u_i satisfies $H_L(k) < u_i < H_H(k)$. $H_{\min}(k)$ tracks the largest value of the metric known up to slot k above which the best metric surely lies.

Initialization: In the first slot ($k = 1$), the parameters are initialized as follows: $H_L(1) = F_c^{-1}(p_e/n)$, $H_H(1) = \infty$, and $H_{\min}(1) = 0$. Here, p_e is a system parameter, and shall henceforth be referred to as the *Contention load parameter.*

Transmission rule: At the beginning of each slot, each node locally decides to transmit. As mentioned, it transmits if and only if its metric lies between $H_L(k)$ and $H_H(k)$.

Feedback generation: At the end of each slot, the sink broadcasts to all nodes a two-bit feedback: (i) 0 if the slot was idle (when no node transmitted), (ii) 1 if the outcome was a success (when exactly one node transmitted), or (iii) e if the outcome was a collision (when at least two nodes transmitted).[2]

Response to feedback: We first define the split function [17] to facilitate the specification of the algorithm[3]: Let $\text{split}(a, b) = F_c^{-1}\left(\frac{F_c(a) + F_c(b)}{2}\right)$. Then, depending on the feedback, the following possibilities occur:

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

Termination: The algorithm terminates when the outcome is a success (1).

We shall call the durations before and after the first non-idle slot as the *idle* and *collision* phases, respectively. Thus, the contention load parameter, p_e, is the average number of users that transmit in a slot in the idle phase. The Qin-Berry algorithm [17] uses $p_e = 1$, which is the value that maximizes the probability of a success outcome in an idle phase slot.

C. Main Analytical Results

The floor and ceil operations are denoted by $\lfloor . \rfloor$ and $\lceil . \rceil$, respectively. $\mathbf{E}[Z]$ will denote the expected value of a random variable Z.

We now develop a new analysis of the average time taken, $m_n(p_e)$, by the splitting algorithm to select the single best relay. The following lemma gives an exact expression for $m_n(p_e)$.

Lemma 1: Let X_k be the number of slots required to resolve a collision among k nodes. Let $q = \left\lceil \frac{n}{p_e} \right\rceil - 1$ denote the idle phase duration in slots. The average number of slots,

$m_n(p_e)$, required to find 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 $\mathbf{E}[X_k]$ follows the recursion given by

$$\mathbf{E}[X_k] = \frac{0.5^k \left(\sum_{l=2}^{k-1} \binom{k}{l} \mathbf{E}[X_l]\right) + 1}{1 - 0.5^{k-1}}, \quad \text{for all } k \geq 2, \quad (2)$$

and $\mathbf{E}[X_1] = 0$.

Proof: The proof is given in Appendix A. ∎

The above expression is complex and does not directly reveal the scalable nature of the algorithm. The theorem below provides two equivalent and new expressions for the asymptotic case ($n \to \infty$).

Theorem 1: The average number of slots required to find the best node as $n \to \infty$ is given by the following two different, yet equivalent, expressions.

1) *Recursive expression:*

$$m_\infty(p_e) = \frac{1}{e^{p_e} - 1} \sum_{k=1}^{\infty} \frac{\mathbf{E}[X_k] p_e^k}{k!} + \frac{1}{1 - e^{-p_e}}. \quad (3)$$

2) *Non-recursive expression:*

$$m_\infty(p_e) = \frac{1}{1 - e^{-p_e}} + \sum_{i=1}^{\infty} p(i), \quad (4)$$

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

Proof: We show the proof for the recursive expression in (3) below as it leads to a powerful new Poisson point process interpretation that will be useful throughout this paper. For example, it will lead to the derivation of the non-recursive expression in (4), whose proof is relegated to Appendix B. The physical meaning of $p(i)$ and P_j will become clear after the proof.

Let node i have metric u_i with CCDF $F_c(u)$. Let $y_i = nF_c(u_i)$. Then, y_i are i.i.d. and are uniformly distributed in $[0, n]$. *Note that selecting the node with the highest u_i is equivalent to selecting the node with the lowest y_i because the CCDF is a monotonically decreasing function.* Sorting $\{y_i\}_{i=1}^{n}$ in ascending order, we get $y_{[1]} \leq y_{[2]} \leq y_{[3]} \cdots \leq y_{[n]}$, where $[i]$ is the index of the relay with the ith largest metric.

Given y_i, we can define a point process [21] $M(t)$ as $M(t) = \max\{k \geq 1 : y_{[k]} \leq t\}$. Thus, $M(t)$ is the number of points that have occurred up to time t. Since $\{y_i\}_{i=1}^{n}$ are i.i.d. and uniformly distributed, $M(t)$ is binomially distributed. As $n \to \infty$, it can be shown that $M(t)$ forms a Poisson process with rate 1 [21]. Now, the probability that the first non-idle slot is the ith slot and $k \geq 1$ nodes transmit is equal to the probability that $y_{[1]}, \ldots, y_{[k]}$ lie between $(i-1)p_e$ and ip_e, and $y_{[j]} > ip_e$, for $k + 1 \leq j \leq n$. It also implies that no

---

[2]The sink can distinguish between these outcomes using, for example, the strength of the total received power [18].

[3]The split function makes sure that, on average, half of the nodes involved in the last collision transmit in the next slot. Splitting can be made faster along the lines of [20]. However, doing so requires each node to numerically calculate thresholds in each slot that are solutions of degree $n-1$ equations. Also, the improvement due to this scheme turns out to be less than 0.5%.

points lie between 0 and $(i-1)p_e$. Therefore,

$$\Pr\left(x_{[1]} > (i-1)\frac{p_e}{n}, (i-1)\frac{p_e}{n} < x_{[k]} < i\frac{p_e}{n}, x_{[k+1]} > i\frac{p_e}{n}\right)$$
$$= \Pr\left(M((i-1)p_e) = 0, \ M(ip_e) = k\right),$$
$$= \Pr\left(M((i-1)p_e) = 0\right)\Pr\left(M(ip_e) = k \mid M((i-1)p_e)=0\right).$$
$$(5)$$

As $n \to \infty$, from the memoryless property of the Poisson process [21], (5) simplifes to

$$\Pr\left(x_{[1]} > (i-1)\frac{p_e}{n}, (i-1)\frac{p_e}{n} < x_{[k]} < i\frac{p_e}{n}, x_{[k+1]} > i\frac{p_e}{n}\right)$$
$$= e^{-(i-1)p_e}\, e^{-p_e}\frac{p_e^k}{k!},$$
$$= e^{-ip_e}\frac{p_e^k}{k!}. \qquad (6)$$

Recall that $\mathbf{E}\left[X_k\right]$ is the expected number of slots required to resolve a collision among k nodes. Thus, if the first non-idle slot is the ith slot and $k \geq 1$ nodes are involved, then $\mathbf{E}\left[X_k\right] + i$ slots are required to find the best node. Hence, we get $m_\infty(p_e) = \sum_{i=1}^{\infty}\sum_{k=1}^{\infty} e^{-ip_e}\frac{p_e^k}{k!}\left(\mathbf{E}\left[X_k\right] + i\right)$. Simplifying further, we get $m_\infty(p_e) = \sum_{k=1}^{\infty}\frac{p_e^k}{k!}\left(\mathbf{E}\left[X_k\right]\sum_{i=1}^{\infty} e^{-ip_e} + \sum_{i=1}^{\infty} i e^{-ip_e}\right)$. The desired result follows with the help of combinatorial identities [22]. ∎

The main theorem readily gives rises to the following upper bound expression that does not involve an infinite series.

Corollary *1:* For any real $k_0 \geq e/2$,

$$m_\infty(p_e) \leq \frac{p_e}{k_0 \log_e(2)} + \log_2\left(\frac{2k_0}{e}\right) + \frac{1}{1 - e^{-p_e}}. \qquad (7)$$

Proof: The proof is given in Appendix C. ∎

Alternatively, since both the expressions derived in Theorem 1 involve only positive terms in the series summation, considering only the first few terms of the infinite series in (3) and (4) results in tight lower bounds near the optimal contention parameter value. These simplified expressions allow system designers to quickly compute the necessary parameters for system optimization. As we shall see, their behavior turns out to be quite different and sheds light on the differences between the two equivalent expressions derived in Theorem 1.

D. Results for Single Relay Selection

Figure 2 plots the average number of slots required to select the best node as a function of p_e for the two expressions and verifies them using Monte Carlo simulations. It can be seen that the asymptotic expression is accurate even when the number of relays is small (e.g., 10). Furthermore, the optimal value of $m_\infty(p_e)$ is 2.467, and occurs at $p_e = 1.088$. As expected, the optimal p_e does not exceed 2. This is because having more than two nodes on average to transmit and collide in a slot is suboptimal. We also observe that $m_\infty(p_e)$ at $p_e = 1$ is quite close to the optimal value.

Figure 3 plots the upper bound using $k_0 = 2$. As expected, it has a unique minimum and follows the behavior of the exact expression well in the region of interest of p_e. The same figure also compares the lower bounds obtained using the first 4 terms of both the expressions of Theorem 1. For higher values of p_e, the lower bound obtained by truncating the recursive
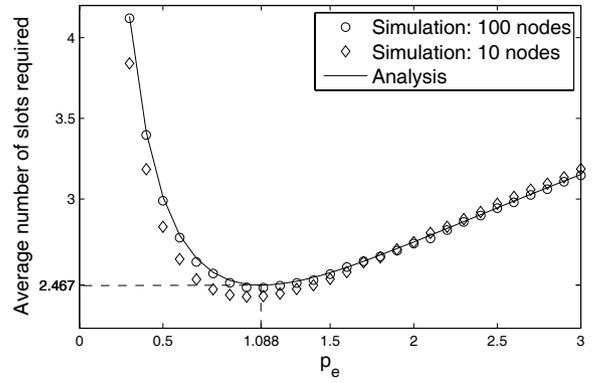

Fig. 2. Average number of slots required to select the best node ($m_\infty(p_e)$) as a function of p_e
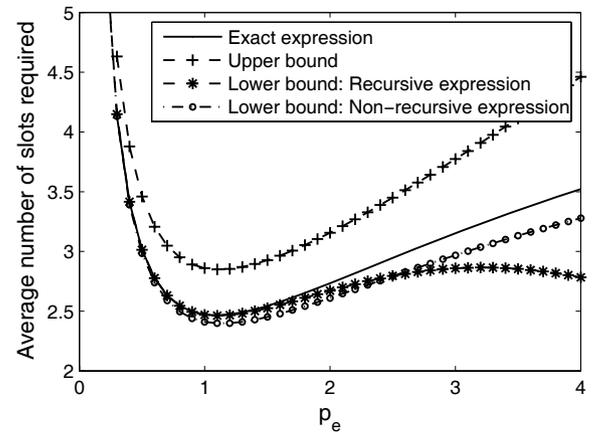

Fig. 3. Upper and lower bounds for the average number of slots required to select the best node.

expression in (3) does not capture the behavior of the exact expression well. This is because of the truncation, on account of which the possibility that a large number of nodes collide in the first non-idle slot is not accounted for. This probability is not negligible for larger p_e. However, the lower bound obtained by truncating the non-recursive expression in (4) does better at larger p_e because the summation in the series is over the number of slots required after the first non-idle slot and not over the number of nodes that collided in the first non-idle slot.

III. Q-RELAY SELECTION ALGORITHM

We now develop a new family of splitting algorithms for selecting the relays with the Q best (highest) metrics, where Q is a pre-specified system parameter. The value of Q depends on the system under consideration. For example, in [13], $M - 1$ relays need to be selected to forward the transmissions by M sources. The choice of Q, which is beyond the scope of this paper, is ultimately governed by the end-to-end system performance and practical constraints such as the synchronization requirements across the selected relays. In some systems, using more relays is beneficial [7], while in some, this might not be the case [13]. Also, selecting multiple relays will also require the system to expend additional resources. The reader

is referred to [3], [7], [10], [23] for a detailed discussion on this aspect.

A. Algorithm Motivation and Definition

When we revisit the asymptotic regime considered in the previous section, we observe the following. The single node selection algorithm, in effect, runs the FCFS algorithm [16] on the Poisson point process $M(t)$ defined in Theorem 1, with t being interpreted as time. However, unlike FCFS, the single relay selection algorithm stops as soon as it finds the first (best) node. In this context, the parameter p_e is analogous to FCFS's *initial contention interval*.

Based on the above insight, we now formally state the new multi-relay selection algorithm given any Q. We then explain the logic behind it and fully analyze it. For this, we adopt the notation used for FCFS in [16], as it turns out to be more convenient.

As in Sec. II, let $y_i = nF_c(u_i)$. The goal of the selection algorithm will be to find the nodes with indices $[1], \ldots, [Q]$, where, as before, $y_{[1]} \leq \cdots \leq y_{[Q]} \leq \cdots \leq y_{[n]}$. The algorithm specifies four state variables $S(k)$, $T(k)$, $\alpha(k)$, and $\sigma(k)$ for each slot k. $S(k)$ is the number of nodes selected before slot k. $(T(k), T(k) + \alpha(k))$ represents the *threshold 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 kth slot interval is the left half or the right half of the previously split interval. During initial slots, when no collision is to be resolved, $\sigma(k) = R$ by convention. Thus, for $k = 1$, we have $S(1) = 0$, $T(1) = 0$, $\alpha(1) = p_e$, and $\sigma(1) = R$.

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 $S(k + 1) = Q$.

B. Brief Explanation

The logic behind the algorithm is as follows: (i) When a collision occurs, the threshold interval for the next slot is the left (L) half of that of the present slot. (ii) When a collision occurs, the threshold interval must have at least 2 nodes. Thus, when a success follows a collision, the threshold interval for the next slot is the right (higher) half (R) of the previous slot, since it is known to have at least one node. (iii) When an idle follows a collision, it implies that all the nodes involved in collision lie in the right half of the previous split interval. Thus, it is further split into two equal halves, and the threshold interval for the next slot is the left half of this split interval.

(iv) When there is no collision to be resolved, the algorithm moves to the adjacent threshold interval (which we call as collision resolution interval) of size p_e.[4] As mentioned above, the algorithm terminates after the Q successes.

Comments: The proposed algorithm is equivalent to the algorithm of Sec. II when $Q = 1$. It is similar to FCFS, except that it stops after the Qth success. There is one subtle difference, however, between the algorithm and FCFS. In FCFS, the contention resolution interval can be smaller than p_e if the difference between the current time and the time of the last resolved interval is small. However, this does not happen in our algorithm (Step 4) because all the nodes know their individual metrics a priori. Notice that the algorithm is greedy in that it does not account for possible interactions between metrics of the relays; the relays do not update the metrics based on the subset of relays that have been selected thus far as $S(k)$ increases from 1 to Q. While greedy approaches have shown promise and are used given their inherent distributability [8], their benefit depends on the specifics of the system that uses such selection.[5]

C. Algorithm Analysis: Best Two Nodes Selection

First, we analyze the algorithm for selecting the best two nodes using the Poisson point approach that came out of Sec. II. This will lead to an analysis for the general $Q > 2$ node selection case. The $Q = 2$ analysis is shown separately as it turns out to be richer.

Let $m_\infty^{(Q)}(p_e)$ represent 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 $m_\infty^{(1)}(p_e)$. The following theorem gives two different but equivalent and exact expressions for $m_\infty^{(2)}(p_e)$.

Theorem 2: 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. As $n \to \infty$, $m_\infty^{(2)}(p_e)$ is given by

$$m_\infty^{(2)}(p_e) = \frac{1}{e^{p_e} - 1} \sum_{k=1}^{\infty} \frac{\mathbf{E}\left[X_k^{(2)}\right] p_e^k}{k!} + \frac{1}{1 - e^{-p_e}}, \quad (8)$$

where

$$\mathbf{E}\left[X_k^{(2)}\right] = \left(2^k - 2\right)^{-1} \left(\left(\sum_{i=2}^{k-1} \binom{k}{i} \mathbf{E}\left[X_i^{(2)}\right]\right.\right.$$
$$\left.\left. + k\left(1 + \mathbf{E}\left[X_{k-1}^{(1)}\right]\right)\right) + 2^k\right), \forall k \geq 3, \quad (9)$$

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

---

[4]We can relax the restriction that each collision resolution interval is of length p_e. However, it can be shown that doing this leads to a negligible improvement.

[5]A more general version of the algorithm would allow for the metrics to be modified on the basis of the relays that have already been selected. Developing such an algorithm is an interesting avenue for future work, and would find several applications, such as in the time-sharing proportional fair solution of [24].

Alternately, $m_\infty^{(2)}(p_e)$ also equals

$$m_\infty^{(2)}(p_e) = \frac{1}{1 - e^{-p_e}} + P_0 m_\infty^{(1)}(p_e)$$
$$+ \sum_{i=1}^{\infty} \left(p(i) + p'(i) + p''(i+1) \right), \quad (10)$$

where

$$p(i) = (1 - P_0) \prod_{j=1}^{i-1} (1 - P_{L,j}), \quad \forall\, i \geq 1, \qquad (11)$$

$$p'(i) = p(i) P_{L,i}, \quad \forall\, i \geq 1, \qquad (12)$$

$$p''(2) = p'(1)(1 - P_{R,1}), \quad \text{and} \qquad (13)$$

$$p''(i) = p'(i-1)(1 - P_{R,i-1}) + p''(i-1)(1 - P_{L,i-1}),$$
$$\forall\, i > 2. \qquad (14)$$

Here, $P_0 = \frac{p_e e^{-p_e}}{1 - e^{-p_e}}$, $P_{L,i} = \frac{2^{-i} p_e e^{-2^{-i} p_e}(1 - e^{-2^{-i} p_e})}{1 - (1 + 2^{-(i-1)} p_e) e^{-2^{-(i-1)} p_e}}$, $\forall\, i \geq 1$, and $P_{R,i} = \frac{2^{-i} p_e e^{-2^{-i} p_e}}{1 - e^{-2^{-i} p_e}}$, $\forall\, i \geq 1$.

Proof: The proof is given in Appendix D. It also gives a physical meaning for $p(i)$, $p'(i)$, $p''(i)$, $P_{L,i}$, and $P_{R,i}$. ∎

D. Algorithm Analysis: Best $Q > 2$ Nodes Selection

We now derive a general expression for $m_\infty^{(Q)}(p_e)$ for any $Q > 2$. This generalizes the first result of Theorem 2.

Theorem 3: As $n \to \infty$, the average number of slots required to select the best $Q > 2$ nodes is

$$m_\infty^{(Q)}(p_e) = \frac{1}{e^{p_e} - 1} \sum_{k=1}^{\infty} \frac{\mathbf{E}\left[X_k^{(Q)}\right] p_e^k}{k!} + \frac{1}{1 - e^{-p_e}}, \quad (15)$$

where

$$\mathbf{E}\left[X_k^{(Q)}\right] = (2^k - 2)^{-1} \left(\left(\sum_{i=2}^{k-1} \binom{k}{i} \mathbf{E}\left[X_i^{(Q)}\right] \right) \right.$$
$$\left. + k \left(1 + \mathbf{E}\left[X_{k-1}^{(Q-1)}\right] \right) \right) + 2^k \right), \quad \forall\, k \geq 3, \quad (16)$$

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

Proof: The proof is given in Appendix E. ∎

An expression for $m_\infty^{(Q)}(p_e)$, for $Q > 2$, along the lines of (4) of Theorem 1 and (10) of Theorem 2 can be also derived. However, the Markov chains become more involved.

E. Results for Q Best Relay Selection

Figure 4 plots $m_\infty^{(2)}(p_e)$ as a function of p_e using Theorem 2 and verifies it using Monte Carlo simulations. It can be seen that the asymptotic expressions are accurate even for a small number of nodes, e.g., $n = 20$. The lowest average number of slots required to select two users is 4.406, which occurs at $p_e = 1.221$. This is 10.7% faster than running the single relay selection algorithm twice, which requires $2 \times 2.467 = 4.934$ slots. The increase in the optimal p_e from 1.088 slots for $Q = 1$ to 1.221 slots for $Q = 2$ occurs because now it is faster to resolve a collision than to avoid it. Specifically, the time taken
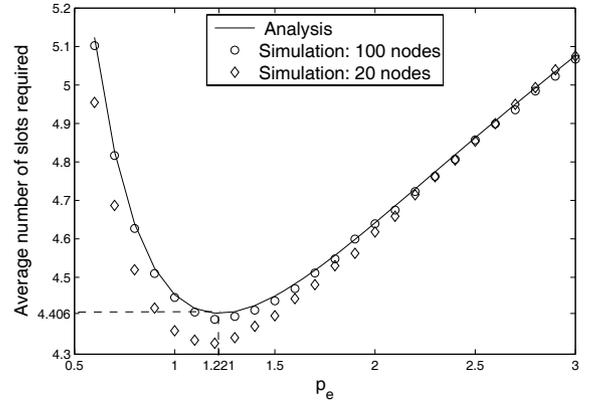

Fig. 4. Average number of slots required to select the best two nodes $(m_\infty^{(2)}(p_e))$ as a function p_e.

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

Q	Optimum p_e	Optimum $m_\infty^{(Q)}(p_e)$ (slots)	Improvement
1	1.088	2.467	-
2	1.221	4.406	10.7%
3	1.214	6.491	12.3%
4	1.231	8.537	13.5%
5	1.236	10.592	14.1%
6	1.241	12.645	14.6%

to select two nodes given that they are involved in a collision is $\mathbf{E}\left[X_2^{(2)}\right] = 3.0$ slots. Where as, the number of slots required to select two nodes, given that the previous slot was idle, is 4.4 slots.

Table I provides 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. We can see that selecting the best three nodes takes 6.491 slots, on average, and is achieved when $p_e = 1.214$.[6] As $Q \to \infty$, the optimum value of p_e increases to 1.266, which is also the optimum value maximizing the throughput of FCFS [16].[7] Also, it can be shown that $\frac{Q}{m_\infty^{(Q)}(1.266)}$, which represents the average number of users selected per slot by the algorithm for $p_e = 1.266$, increases to 0.487 as $Q \to \infty$.

F. Tackling Discrete Metrics Using Proportional Expansion

The thresholding algorithms in Sec. II-A and Sec. III-A exploit the critical fact that with probability one no two metrics are equal. However, as mentioned in the Introduction, when the metric has a discrete probability distribution, the

---

[6]The marginal 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 $\mathbf{E}\left[X_2^{(3)}\right] = 5.48$ 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 5.48. Therefore, the optimum p_e decreases since the selection times after an idle and a collision are not as unequal as for $Q = 2$.

[7]The maximum arrival rate of 0.487 is supported when initial collision interval is capped at 2.6. This implies that there are, on average, $0.487 \times 2.6 = 1.266$ nodes transmitting. The contention parameter p_e is set using the normalized metric CCDF with an 'arrival rate' equal to 1.
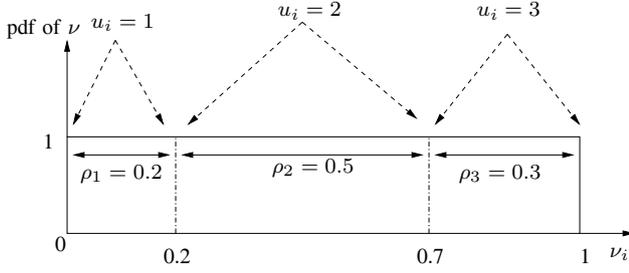
Fig. 5. Illustration of Proportional Expansion for discrete metrics. An example is shown for the case where the metric takes three values 1, 2, and 3 with probabilities 0.2, 0.5, and 0.3, respectively.

algorithms break down because the probability that the metrics of the best two nodes are exactly equal is non-zero. We now provide a simple and novel distributed solution called *Proportional Expansion* to tackle this practical problem.

Proportional Expansion: Let the metric u_i be a realization of an ω-valued discrete random variable that, without loss of generality, takes values $1, 2, \ldots, \omega$ with probability $\rho_1, \rho_2, \ldots, \rho_\omega$, respectively. Each node independently maps its metric u_i into a new metric ν_i as follows: *When $u_i = j$, ν_i is a realization of a uniformly distributed random variable in* $\left(\sum_{\ell=0}^{j-1} \rho_\ell, \sum_{\ell=1}^{j} \rho_\ell \right)$, *where $\rho_0 \triangleq 0$.* In other words, each node chooses a new random metric ν_i that is uniformly distributed over a bin of length *proportional* to the probability mass of its original discrete metric u_i.

The overall distribution of the new metric across all users is then uniformly distributed in $(0, 1)$. Proportional Expansion satisfies two key properties:

- It preserves the sorting order of the metrics: if $u_i > u_j$, then $\nu_i > \nu_j$. Hence, selecting the best Q nodes with the highest ν_is is equivalent to selecting the best Q nodes with the highest u_is.
- The probability that $\nu_i = \nu_j$, for $i \neq j$, is 0 since ν_i is a continuous random variable.

Therefore, the selection algorithm of Sec. III for any $Q \geq 1$ can now be run on ν_i. The following Proposition formally quantifies the performance of Proportional Expansion. It implies that proportional expansion is scalable, *i.e.*, it takes at most 2.47 slots for best relay selection, 4.406 slots for selecting the best 2 relays, and so on, for any number of relays, n.

Proposition 1: The average number of slots required to select the best Q relays by Proportional Expansion for the discrete metrics case is the same as that of the best Q relay threshold based selection algorithm of Sec. III-A that operates on continuous metrics.

Proof: The proof is omitted since it directly follows from the above discussion. ∎

IV. CONCLUSIONS

We developed a new asymptotic analysis for the splitting-based single relay selection algorithm. The analysis is based on a new Poisson point process interpretation of the dynamics of the algorithm. This led to a characterization of the optimal parameters of the algorithm, and enabled a rigorous benchmarking of the greedy parameter setting used in the

literature. We also proposed a new splitting based algorithm for selecting the best Q relays, which is useful for several cooperative protocols proposed in the literature that require the use of multiple relays. The new algorithm is more efficient than running the single relay selection algorithm multiple times. Furthermore, we generalized the analysis to handle multiple relay selection, and derived the exact expressions for the average number of slots required by it. Interestingly, the asymptotic expressions were accurate even for a small number of relays. With the help of proportional expansion, we showed, for the first time, that splitting algorithms can be adapted to work for discrete metrics, as well, without any loss in performance or scalability whatsoever.

The analysis shows that the greedy policy of maximizing the success probability in the next slot is suboptimal. While it works well for single relay selection, it becomes more and more suboptimal as the number of relays to be selected increases. The analysis also shows that the general single relay selection algorithm, the proposed multiple relay selection algorithm, and the FCFS multiple access control algorithm are intimately related. For example, the optimal value of the contention load parameter increases as the number of relays to be selected increases and finally approaches the optimal setting for FCFS. This is despite the fact that selection and multiple access control algorithms serve very different purposes, and are, therefore, evaluated differently. While multiple access control algorithms attempt to serve all nodes and are evaluated, for example, by the maximum traffic they can handle with a finite delay, selection algorithms are evaluated by how fast they can select the best nodes. We hope that this insight will help develop better selection algorithms.

An important property about splitting algorithms is that besides being distributed, they are both extremely fast and scalable. This suggests that selection based protocols will deliver improvements in the overall end-to-end system-level performance even when the time overhead incurred by the selection algorithm is accounted for. The system-level benefits can be further improved if the multiple relay selection algorithm proposed in this paper can be modified to allow the metrics to be updated during the selection process.

APPENDIX

A. Proof of Lemma 1

It can be easily seen that the idle phase consists of at most $q = \left\lceil \frac{n}{p_e} \right\rceil - 1$ slots since at this stage the lower threshold equals the smallest value 0. Given that the first non-idle slot is the ith slot and k nodes are involved, the average number of slots required to find the best node is $\mathbf{E}[X_k] + i$. (The recursive expression for $\mathbf{E}[X_k]$ is given in [17, (6)].) The probability that the first non-idle slot is the ith slot and k nodes transmit in it equals $\binom{n}{k} \left(\frac{p_e}{n} \right)^k \left(1 - \frac{i p_e}{n} \right)^{n-k}$, for $i \leq q$. This constitutes the first term of the right side of (1). The probability that the $(q+1)^{\text{th}}$ slot is the first non-idle slot is $(1 - \frac{q p_e}{n})^n$ since all nodes' metrics must lie in the interval $((q+1) p_e, 1]$. In the event that this happens, all n nodes will transmit and collide, which will take $\mathbf{E}[X_n]$ slots to resolve. Hence, the second term on the right side of (1) follows.

B. Proof of Non-Recursive Expression of Theorem 1

Let the random variable I denote the number of slots required until (and including) the first non-idle slot, and let Y denote the number of slots required after that.

Consider the state transition diagram of Figure 6, in which the state represents the number of slots that have elapsed since the first non-idle slot. The node goes to state $\mathcal{S}$ whenever success occurs, and the algorithm terminates. Otherwise, in case of an idle or collision, the node increments its state by 1. By definition, state 0 is the first non-idle slot itself; thus, an idle outcome cannot occur in it. The following lemma is crucial in analyzing this transition diagram.

Lemma 2: The state transition diagram of Fig. 6 is a Markov chain.

Proof: To prove this, it is sufficient to prove that the transition probability from any state i to $\mathcal{S}$ is dependent only on i. (Having done so, we shall denote this probability by P_i.)

We refer to the interval in $M(t)$ allocated to state i as its threshold interval. Here, P_0 is the probability that in a threshold interval of size p_e only one node transmits given that at least one node transmits in that slot. Let $N(x) = M(t+x) - M(t)$. Then, from the memoryless property of the Poisson process, $\Pr\left(N(x) = i\right)$ is independent of t and is equal to $\frac{x^i e^{-x}}{i!}$. Thus,

$$P_0 = \Pr\left(N(p_e) = 1 \middle| N(p_e) > 1\right) = \frac{p_e e^{-p_e}}{1 - e^{-p_e}}. \qquad (17)$$

P_1 is the probability that the second non-idle slot is a success given that the first non-idle slot (of threshold interval size p_e) is a collision. Due to splitting, the second slot will have a threshold interval size that is half that of the first one. Therefore, P_1 is the probability that conditioned on $N(p_e)$ having at least 2 nodes (*i.e.*, a collision), $N(p_e/2)$ has exactly one node. Thus,

$$P_1 = \Pr\left(N\left(\frac{p_e}{2}\right) = 1 \middle| N(p_e) \geq 2\right), \qquad (18)$$

$$= \frac{\Pr\left(N\left(\frac{p_e}{2}\right) = 1, N(p_e) \geq 2\right)}{Pr(N(p_e) \geq 2)}. \qquad (19)$$

Therefore,

$$P_1 = \frac{\Pr\left(N\left(\frac{p_e}{2}\right) = 1, N(p_e) - N\left(\frac{p_e}{2}\right) \geq 1\right)}{Pr(N(p_e) > 1)}, \qquad (20)$$

$$= \frac{\frac{p_e}{2} e^{-\frac{p_e}{2}}\left(1 - e^{-\frac{p_e}{2}}\right)}{1 - (1 + p_e)e^{-p_e}}. \qquad (21)$$

For P_2, the following two trajectories can occur: State 2 was reached by a collision in state 1 or by an idle in state 1. In case of a collision, the threshold interval of the second non-idle slot (of size $p_e/2$) gets split into two halves. Even in the case of an idle the interval would be split into two halves and nodes from the left half would contend. Thus, P_2 is equal to the probability that conditioned on an interval of size $p_e/2$ having at least two nodes, half the interval (of size $p_e/4$), has
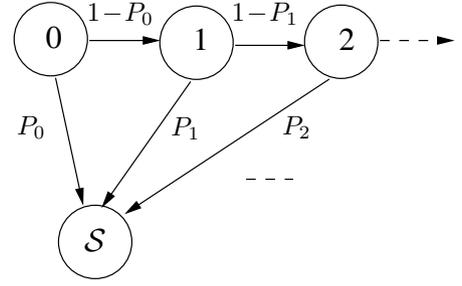

Fig. 6. State transition diagram for the number of slots required to select the best node after the first non-idle slot.

exactly one node. Thus,

$$P_2 = \Pr\left(N\left(\frac{p_e}{4}\right) = 1 \middle| N\left(\frac{p_e}{2}\right) \geq 2\right), \qquad (22)$$

$$= \frac{\Pr\left(N\left(\frac{p_e}{4}\right) = 1, N\left(\frac{p_e}{2}\right) \geq 2\right)}{Pr(N(p_e) \geq 2)}, \qquad (23)$$

$$= \frac{\frac{p_e}{4} e^{-\frac{p_e}{4}}\left(1 - e^{-\frac{p_e}{4}}\right)}{1 - \left(1 + \frac{p_e}{2}\right)e^{-\frac{p_e}{2}}}. \qquad (24)$$

In similar way, we can show that P_i, $\forall\, i \geq 1$, is equal to the probability that conditioned on an interval of size $2^{-(i-1)}p_e$ having at least two nodes, one half of the interval (of size $2^{-i}p_e$) has exactly one node. Thus,

$$P_i = \Pr\left(N\left(\frac{p_e}{2^i}\right) = 1 \middle| N\left(\frac{p_e}{2^{i-1}}\right) > 1\right), \qquad (25)$$

$$= \frac{2^{-i}p_e e^{-2^{-i}p_e}\left(1 - e^{-2^{-i}p_e}\right)}{1 - (1 + 2^{-(i-1)}p_e)e^{-2^{-(i-1)}p_e}}. \qquad (26)$$

∎

Now, $m_\infty(p_e) = \mathbf{E}[I] + \mathbf{E}[Y]$. From the Poisson process interpretation of Theorem 1, we can show that $\Pr\left(I = i\right) = e^{-(i-1)p_e}(1 - e^{-p_e})$. Therefore, $\mathbf{E}[I] = \sum_{i=1}^\infty i e^{-(i-1)p_e}(1 - e^{-p_e}) = \frac{1}{1 - e^{-p_e}}$. The average number of slots required after the first non-idle slot to select the best node, $\mathbf{E}[Y]$, is calculated as follows. First, $\mathbf{E}[Y] = \sum_{i=1}^\infty i \Pr(Y = i)$, can be shown to be identically equal to $\sum_{i=1}^\infty \Pr(Y \geq i)$. Second, since each state in the Markov chain is visited at most once, it follows that $\mathbf{E}[Y] = \sum_{i=1}^\infty p(i)$, where $p(i)$ is the probability that the ith state is visited. From the state transition diagram, it is easy to see that $p(i) = (1 - P_0)\prod_{j=1}^{i-1}(1 - P_j)$. Hence, the desired expression for $m_\infty(p_e)$ follows.

C. Proof of Corollary 1

From [17], we have $\mathbf{E}[X_k] \leq \log_2(k) + 1$, $k \geq 2$, and $\mathbf{E}[X_1] = 0$. 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. Therefore,

$$\log_2(k) \leq \frac{k - k_0}{k_0 \log_e(2)} + \log_2(k_0). \qquad (27)$$

Consequently, $\mathbf{E}[X_k] \leq \frac{k}{k_0 \log_e(2)} + \log_2(2k_0/e)$, $k \geq 2$. Substituting this in (3), we get

$$m_\infty(p_e) \leq \log_2\left(\frac{2k_0}{e}\right) \frac{\sum_{k=2}^\infty \frac{p_e^k}{k!}}{e^{p_e} - 1}$$

$$+ \frac{1}{k_0(e^{p_e} - 1)\log_e(2)} \sum_{k=2}^\infty k \frac{p_e^k}{k!} + \frac{1}{1 - e^{-p_e}}. \qquad (28)$$

For $k_0 \geq e/2$, $\log_2(2k_0/e) \geq 0$. Also, since $p_e > 0$, we have $\sum_{k=2}^{\infty} \frac{p_e^k}{k!} = e^{p_e} - 1 - p_e < e^{p_e} - 1$. Therefore, for $k_0 \geq e/2$, the first term in the right hand side of (28) is less than $\log_2\left(\frac{2k_0}{e}\right)$. Substituting this inequality in (28) and simplifying leads to the desired result in (7).

D. Proof of Theorem 2

Proof of (8): Given that the first non-idle slot is the ith slot and $k \geq 1$ nodes are involved, the average number of slots required to select the best 2 nodes is $\mathbf{E}\left[X_k^{(2)}\right] + i$. The probability that the first non-idle slot is the ith slot and $k \geq 1$ nodes are involved is $e^{-ip_e}p_e^k/k!$. Hence, we get

$$m_\infty^{(2)}(p_e) = \sum_{i=1}^{\infty} \sum_{k=1}^{\infty} e^{-ip_e} \frac{p_e^k}{k!} \left(\mathbf{E}\left[X_k^{(2)}\right] + i\right), \quad (29)$$

simplifying which yields (8).

If only one node transmits in the first non-idle slot, then a success occurs and the node gets selected. Selecting one more node will take $m_\infty^{(1)}(p_e)$ slots, on average. (This follows from the memoryless property of the Poisson process [21].) Thus, $\mathbf{E}\left[X_1^{(1)}\right] = m_\infty^{(1)}(p_e)$. Also, if exactly two nodes transmit in the first non-idle slot, only one node transmits in the slot just after the first success. Thus, $\mathbf{E}\left[X_2^{(2)}\right] = \mathbf{E}\left[X_1^{(2)}\right] + 1 = 3$ slots. When $k > 3$ nodes transmit in the first non-idle slot, the following three cases are possible for the next slot:

1) *Collision among i nodes*: $\mathbf{E}\left[X_i^{(2)}\right]$ more slots would then be required, on average.
2) *Idle*: $\mathbf{E}\left[X_k^{(2)}\right]$ more slots are required, on average.
3) *Success*: The next slot would then surely involve a collision among $k-1$ nodes. $\mathbf{E}\left[X_{k-1}^{(1)}\right]$ slots, on average, would be required *after* that.

The probability that i nodes transmit in the next slot is $\binom{k}{i}/2^k$. Thus, from the law of total expectation,

$$\mathbf{E}\left[X_k^{(2)}\right] = \frac{1}{2^k}\left(\left(\binom{k}{0} + \binom{k}{k}\right)\left(1 + \mathbf{E}\left[X_k^{(2)}\right]\right)\right.$$
$$\left. + \binom{k}{1}\left(1 + \mathbf{E}\left[X_{k-1}^{(1)}\right] + 1\right) + \sum_{i=2}^{k-1}\binom{k}{i}\left(1 + \mathbf{E}\left[X_i^{(2)}\right]\right)\right). \quad (30)$$

Simplifying this further using combinatorial identities [22] results in (9).

Proof of (10): This proof also involves constructing a state transition diagram that will turn out to be a Markov chain. Consider the state transition diagram of Figure 7. It is more involved than that in Figure 6 because we need to also track how many successes have occurred. State i corresponds to the ith split before the first success (which takes i slots), state i' corresponds to the first success occurring at the ith slot, and state i'' corresponds to the first success having already occurred by the ith slot. The state transition diagram can be explained in detail as follows.

State 0 corresponds to the first non-idle slot. If the first non-idle slot is a success, the node moves from state 0 to state $\mathcal{S}_1$. Now, the algorithm starts a new collision resolution to find the

second colliding node. This takes $m_\infty^{(1)}(p_e)$ slots, on average, an expression for which is derived in Theorem 1. If the first non-idle slot is a collision, its threshold interval is split and the node transitions from state 0 to state 1. Each subsequent idle or collision results in one additional split and the node moves from state i to $i+1$. In case of a success, the node moves from state i to state i' as no additional split occurs. A success in state i' results in a transition to state $\mathcal{S}$, at which time the algorithm terminates. In case of a collision in state i', the node moves to state $(i+1)''$, as one more split occurs. In case of a success in state i'', the node moves to state $\mathcal{S}$, and the algorithm terminates. Otherwise, an idle or collision results in a transition from state i'' to $(i+1)''$. Note that in each state (i, i', or i'') the size of threshold interval is $2^{-i}p_e$.

The following Lemma shall prove to be crucial in analyzing this transition diagram.

Lemma *3*: The state transition diagram of Fig. 7 is a Markov chain.

Proof: For this, it is sufficient to prove that transition probabilities for each state depend only on the i and not on the path taken to reach that state.

Let P_0 be the probability of success in state 0 (the first non-idle slot). It is equal to the probability that, in a slot of size p_e, only one node transmits given that at least one node transmits in that slot. Let $N(x) = M(t+x) - M(t)$. Then, from the memoryless property of a Poisson process, $\Pr\left(N(x) = i\right)$ is independent of t and is equal to $\frac{x^i e^{-x}}{i!}$. Thus,

$$P_0 = \Pr\left(N(p_e) = 1 \middle| N(p_e) > 1\right) = \frac{p_e e^{-p_e}}{1 - e^{-p_e}}. \quad (31)$$

Let $P_{L,i}$ be the probability of success in state i. It is equal to the probability that the left half of an interval of size $2^{-i}p_e$ has exactly one node given that the full interval has more than one node. Thus,

$$P_{L,i} = \Pr\left(N\left(\frac{p_e}{2^i}\right) = 1 \middle| N\left(\frac{p_e}{2^{i-1}}\right) > 1\right), \quad (32)$$

$$= \frac{2^{-i}p_e e^{-2^{-i}p_e}(1 - e^{-2^{-i}p_e})}{1 - (1 + 2^{-(i-1)}p_e)e^{-2^{-(i-1)}p_e}}. \quad (33)$$

Let $P_{R,i}$ be the probability of success in state i'. State i' can be entered only after success in state i. Thus, the threshold interval of state i, which is the right half of the interval split during state $i-1$, has *at least* one node. Thus $P_{R,i}$ is equal to the probability that exactly one node transmits in the slot with interval size $2^{-i}p_e$, given that at least one node lies in that interval, which equals

$$P_{R,i} = \Pr\left(N\left(\frac{p_e}{2^i}\right) = 1 \middle| N\left(\frac{p_e}{2^i}\right) > 1\right) = \frac{2^{-i}p_e e^{-2^{-i}p_e}}{1 - e^{-2^{-i}p_e}}. \quad (34)$$

The probability of success in state i'' is again equal to the probability that given that an interval of size $2^{-i}p_e$ has more than one node, its left half has exactly one node. This probability equals $P_{L,i}$. Thus, from (32) and (34), the transition probabilities $P_{L,i}$ and $P_{R,i}$ only depend on i, which proves that Fig. 7 is a Markov chain. ∎

Let the random variable I denote the number of slots required until (and including) the first non-idle slot and Y denote the number of slots required after that. Then,
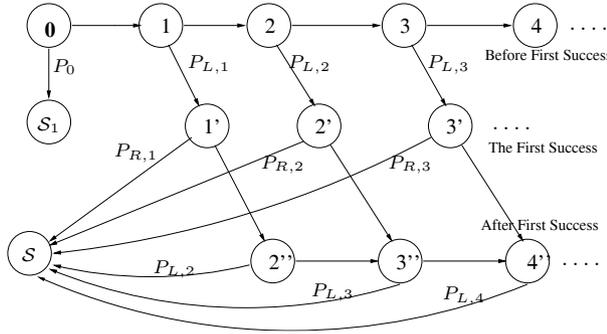
Fig. 7.   State transition diagram for the number of slots required to select the best two nodes after the first non-idle slot.

$m_\infty^{(2)}(p_e) = \mathbf{E}[I] + \mathbf{E}[Y]$. Again, using Poisson point process interpretation, $\Pr(I = i) = e^{-(i-1)p_e}(1 - e^{-p_e})$, which implies,

$$\mathbf{E}[I] = \sum_{i=1}^{\infty} i e^{-(i-1)p_e}(1 - e^{-p_e}) = \frac{1}{1 - e^{-p_e}}. \qquad (35)$$

$\mathbf{E}[Y]$ can be calculated from Lemma 3 as follows. Let $p(i)$, $p'(i)$, and $p''(i)$ be the probability that states i, i', and i'' are visited, respectively. From the state transition diagram, since state i can be reached only from state $i - 1$ and state i' can be reached only from state i, we get $p(i) = (1 - P_0)\prod_{j=1}^{i-1}(1 - P_{L,j})$, $\forall\ i \geq 1$, and $p'(i) = p(i)P_{L,i}$, $\forall\ i \geq 1$. Also, for $i > 2$, state i'' can be reached from $(i-1)'$ and $(i-1)''$. Therefore, we get

$$p''(i) = p'(i-1)(1-P_{R,i-1})+p''(i-1)(1-P_{L,i-1}), \quad \forall\ i > 2, \qquad (36)$$

and

$$p''(2) = p'(1)(1 - P_{R,i-1}). \qquad (37)$$

Now, for state $\mathcal{S}_1$ to be visited, $m_\infty^{(1)}(p_e)$ slots are required, on average. This occurs with probability P_0. Else, the average number of slots is equal to $\sum_{j=1}^{\infty} \Pr(Z \geq j)$, where Z is the total number of states visited excluding state 0. This is so because we are counting the number of slots required *after* the first non-idle slot. Since each state is visited at most once, the average above is equal to $\sum_{i=1}^{\infty}(p(i) + p'(i) + p''(i))$. Thus, the average number of slots required after the first non-idle slot is $\mathbf{E}[Y] = P_0 m_\infty^{(1)}(p_e) + \sum_{i=1}^{\infty}(p(i) + p'(i) + p''(i))$.

E. Proof of Theorem 3

The proof is similar to the proof of (8) in Theorem 2, except for the following differences:

1) When two nodes transmit in the first non-idle slot, $\mathbf{E}\left[X_2^{(2)}\right] = 3$ slots, on average, are required to select both of them. Selecting the remaining best $Q - 2$ nodes takes another $m_\infty^{(Q-2)}(p_e)$ slots, on average. Thus, $\mathbf{E}\left[X_2^{(Q)}\right] = \mathbf{E}\left[X_2^{(2)}\right] + m_\infty^{(Q-2)}(p_e)$.

2) When $k > 3$ nodes transmit in the first non-idle slot,

the average number of slots required thereafter is

$$\mathbf{E}\left[X_k^{(Q)}\right] = \frac{1}{2^k}\left[\left(\binom{k}{0} + \binom{k}{k}\right)\left(1 + \mathbf{E}\left[X_k^{(Q)}\right]\right)\right.$$
$$+ \binom{k}{1}\left(1 + \mathbf{E}\left[X_{k-1}^{(Q-1)}\right] + 1\right)$$
$$\left. + \sum_{i=2}^{k-1}\binom{k}{i}\left(1 + \mathbf{E}\left[X_i^{(Q)}\right]\right)\right]. \qquad (38)$$

REFERENCES

[1] 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, pp. 659-672, Mar. 2006.

[2] Z. Lin and E. Erkip, "Relay search algorithms for coded cooperative systems," in *Proc. Globecom*, 2005.

[3] S. Nam, M. Vu, and V. Tarokh, "Relay selection methods for wireless cooperative communications," in *Proc. Conf. Inform. Sci. Syst. (CISS)*, pp. 859-864, Mar. 2008.

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

[5] 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. VTC (Spring)*, pp. 1489-1493, 2005.

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

[7] 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.

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

[9] 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.

[10] C. K. Lo, J. R. W. Heath, and S. Vishwanath, "Relay subset selection in wireless networks using partial decode-and-forward transmission," *IEEE Trans. Veh. Technol.*, vol. 58, pp. 692-704, Feb. 2009.

[11] C. K. Lo, J. R. W. Heath, and S. Vishwanath, "The impact of channel feedback on opportunistic relay selection for hybrid-ARQ in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 58, pp. 1255-1268, Mar. 2009.

[12] D. Tse and P. Vishwanath, *Fundamentals of Wireless Communications*. Cambridge University Press, 2005.

[13] Z. Ding, K. K. Leung, D. L. Goeckel, and D. Towsley, "On the study of network coding with diversity," *IEEE Trans. Wireless Commun.*, vol. 8, pp. 1247-1259, Mar. 2009.

[14] I. Krikidis and J.-C. Belfiore, "Three scheduling schemes for amplify-and-forward relay environments," *IEEE Commun. Lett.*, vol. 5, pp. 414-416, May 2007.

[15] P. Liu, Z. Tao, Z. Lin, E. Erkip, and S. Panwar, "Cooperative wireless communications: a cross-layer approach," *IEEE Trans. Wireless Commun.*, vol. 13, pp. 84-92, Aug. 2006.

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

[17] X. Qin and R. Berry, "Opportunistic splitting algorithms for wireless networks," in *Proc. INFOCOM*, pp. 1662-1672, Mar. 2004.

[18] R. Yim, N. B. Mehta, and A. F. Molisch, "Fast multiple access selection through variable power transmission," *IEEE Trans. Wireless Commun.*, vol. 8, pp. 1962-1973, Apr. 2009.

[19] K. Choumas, T. Korakis, and L. Tassiulas, "New prioritization schemes for QoS provisioning in 802.11 wireless networks," in *Proc. 16th IEEE Workshop on Local and Metropolitan Area Networks*, pp. 25-30, Sept. 2008.

[20] A. Cohen, M. Kam, and R. Conn, "Partitioning a sample using binary-type questions with ternary feedback," *IEEE Trans. Sys., Man, Cybernetics*, vol. 25, pp. 1405-1408, Oct. 1995.

[21] R. W. Wolff, *Stochastic Modeling and the Theory of Queues*. Prentice Hall, 1989.

[22] L. S. Gradshteyn and L. M. Ryzhik, *Tables of Integrals, Series and Products*. Academic Press, 2000.

[23] V. Shah, N. B. Mehta, and R. Yim, "Relay selection and data transmission throughput tradeoff in cooperative systems," in *Proc. Globecom*, Nov. 2009.

[24] T. J. Oechtering and H. Boche, "Bidirectional regenerative half-duplex relaying using relay selection," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 1879-1888, May 2008.

Virag Shah received his Bachelor of Engineering degree in Electronics from University of Mumbai, India in 2007. He received his Master of Engineering degree in Telecommunications from Indian Institute of Science, Bangalore, India in 2009. Since November 2009, he has been a Research Fellow at Indian Institute of Technology, Bombay, India working in the area of communication networks. His research interests include the design and analysis of algorithms for communication networks, network control, queuing theory, graph theory and information theory.

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, CA in 1997 and 2001, respectively. He was a visiting graduate student researcher at Stanford University in 1999 and 2000. He is now an Assistant Professor at the Dept. of Electrical Communication Engineering, Indian Institute of Science (IISc), Bangalore, India. Until 2002, he was a research scientist in the Wireless Systems Research group in AT&T Laboratories, Middletown, NJ. In 2002-2003, he was a Staff Scientist at Broadcom Corp., Matawan, NJ, and was involved in GPRS/EDGE cellular handset development. From 2003-2007, he was a Principal Member of Technical Staff at the Mitsubishi Electric Research Laboratories, Cambridge, MA, USA.

His research includes work on link adaptation, multiple access protocols, WCDMA downlinks, system-level performance analysis of cellular systems, MIMO and antenna selection, and cooperative communications. He was also actively involved in radio access network physical layer (RAN1) standardization activities in 3GPP. He has served on several TPCs, and was a TPC co-chair for the Transmission technologies track of VTC 2009 (Fall) and the Frontiers of Networking and Communications symposium of Chinacom 2008. He is an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and is an executive committee member of the Bangalore chapter of the IEEE Signal Processing Society.

Raymond Yim (S'00-M'07) received his B.Eng. and M.Eng degrees in electrical and computer engineering from McGill University in Canada, and Ph.D. from Harvard University. In 2006-2007, he was an assistant professor in electrical and computer engineering at F. W. Olin College of Engineering, Needham, MA. Since 2008, he has been a research scientist at the Mitsubishi Electric Research Laboratories, Cambridge, MA, and has been involved in WiMAX development. His research interests include the design and analysis of cross-layered architectures and protocols for wireless communication networks, multiple access protocols, link adaptation, cooperative communications, smart antenna systems, interference management and intelligent transportation systems.