

RESEARCH

Open Access

Contention-based node selection with applications to relay communications and load balancing

Günther Brandner^{1*}, Christian Bettstetter^{1,2} and Udo Schilcher¹

Abstract

This article proposes and analyzes a distributed probabilistic selection protocol in which several nodes perform a random access competition on a shared slotted channel. Nodes that are better suited according to some metric are preferred in the random access and are thus selected with high likelihood. Analytical performance studies are made in terms of reliability, message complexity, and delay. The protocol is applied in two case studies: relay communications and load balancing in wireless networks. In relay communications, a node is selected based on its channel state to serve as relay. In load balancing, a node is selected based on its battery charge state to perform a task. Results indicate that selecting nodes based on the observed metric contributes to better performance and longer network lifetime.

1 Introduction

Many network protocols require a distributed solution for selecting a node from a candidate set to perform a certain task. Examples of such scenarios from the area of mobile computing include the selection of a *cluster head* in multihop networks [1], a *relay node* in cooperative wireless communications [2], and a *gathering node* for data processing in sensor networks [3].

This article proposes such a distributed mechanism. The basic concepts are as follows: First, the selection is performed in a contention-based manner with the goal to obtain low signaling overhead. Two random access methods are proposed as alternatives. The first one maximizes the probability of successful selection; the second one reduces the signaling overhead in terms of reply messages sent by candidate nodes. Second, the selection takes into account the specific capability of each node to perform a given task. This capability is quantified in terms of a metric maintained by each node. Nodes with a high metric are well-suited to become selected and are thus preferred in the random access process.

The node selection is evaluated in two case studies. The first case study refers to the selection of a relay node

in a multipath fading environment [2]. Each node maintains a metric describing its current channel quality. The selection process prefers nodes with good channels to be selected as relay. The second case study refers to the selection of nodes performing energy-consuming tasks, and the goal is to balance the load in the network. Using the battery charge state of each node as a selection metric, the proposed selection mechanism contributes toward a longer network lifetime, i.e., it increases the time span until one of the nodes fails due to an empty battery.

Many algorithms for node selection in distributed systems have been proposed in literature. Most of these algorithms employ explicit message exchange between the nodes for selection (see, e.g., [4-9]). Unlike these, the approach presented here is of *probabilistic* nature using a random access competition between the nodes. The intention is to reduce the overhead in terms of signaling and delay at the cost of a nondeterministic success of selection. The message complexity of many selection algorithms significantly depends on the number of nodes n . For example, the algorithm in [8] exhibits a message complexity of order $n \log(n)$, and that of [9], of order n . The message complexity of the selection protocols presented in this article, however, is in principle independent of n and thus suited for dense networks.

*Correspondence: guenther.brandner@uni-klu.ac.at

¹ Mobile Systems Group, Institute of Networked and Embedded Systems, University of Klagenfurt, Klagenfurt 9020, Austria

Full list of author information is available at the end of the article

The remainder of this article is structured as follows: Section 2 presents the selection protocol and mathematically analyzes its reliability, message complexity, and delay. Section 3 first demonstrates the protocol for relay selection in wireless systems and then also for load balancing. Related work addressing selection algorithms, and relay selection and load balancing is discussed in Sections 2 and 3 individually.

Parts of this article, including some equations and results, were previously published in [10] and [11]. Those papers presented some first ideas and results of Section 2. The article at hand gives a more comprehensive analysis and presents two case studies illustrating the use and performance of the proposed approaches.

2 Contention-based node selection

In this chapter we first present the contention-based selection approach in Section 2.1 and performance metrics like reliability, message complexity, and delay in Section 2.2. In Section 2.3 we derive two access strategies applicable for the presented contention-based selection, where the first one (uniform access) maximizes reliability, while the second one (slow-start access) maximizes the probability that only one reply is sent during the selection. In Section 2.4 we analyze the proposed selection schemes theoretically, while in Section 2.5 we introduce the metric-based selection which prefers better-suited nodes to be selected. Finally, in Sections 2.6 and 2.7, we briefly recap the selection procedure and address related work, respectively.

2.1 Basic approach

A query node transmits a *request* message to all its neighboring nodes indicating its need to find a node to perform a certain task. The protocol continues with a contention-based random access scheme using slotted ALOHA during s time slots. These slots are called contention slots in the following. All n nodes receiving the request compete within the contention slots in a way that each node i sends a *reply* in slot j with probability p_{ij} , where $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, s\}$.

The sending probabilities can be written in a matrix

$$\mathbf{P} := \begin{pmatrix} p_{11} & p_{12} & \dots & p_{1s} \\ p_{21} & p_{22} & \dots & p_{2s} \\ \dots & \dots & \ddots & \dots \\ p_{n1} & p_{n2} & \dots & p_{ns} \end{pmatrix}. \quad (1)$$

If we assume these probabilities to be node-independent, they can be written as a vector $\vec{p} := (p_1, \dots, p_s)$, where $p_j := p_{ij} \forall i, j$ is the probability that a node sends in contention slot j . Note that a node may send more than one reply during the selection. The

reply contains a node identifier and, optionally, a metric indicating the node's suitability for selection.

If no node transmits a reply in a given slot, the slot is called empty slot. If more than one node transmit in this slot, the messages will collide in this slot; such slots are called collision slots. Only if exactly one node transmits in a given slot the reply will be properly received by the query node. Such slots are called singleton slots; such messages are called non-colliding messages. The node sending the *first non-colliding message* is the winner and serves as selected node. To avoid misinterpretations, a message is called first non-colliding message if it is a non-colliding message and all previous slots were either empty or collision slots. The winner is informed by the query node in a dedicated control slot.

A main challenge in the protocol design is to use a suitable \mathbf{P} that enables reliable and fast selection with few messages. In the following, we analyze different approaches for choosing \mathbf{P} and compare their performance with respect to (a) *reliability*, the probability that the selection is successful; (b) *message complexity*, the number of replies sent from candidate nodes; and (c) *delay*, the number of contention slots used for one selection attempt.

A node selection is successful once the first non-colliding message is received. Depending on the application, the selection process can be either stopped upon such a message or it can be continued until all s slots are finished. In the following we evaluate the selection algorithms for both cases. We denote the case where the selection is stopped after the first non-colliding message as *early stop*, while we denote the case where the selection is not stopped as *standard protocol*.

In some applications the number of candidate nodes must be known or estimated to apply contention-based node selection. The number of neighboring nodes is often known in certain technologies through the exchange of beacon signals. If not, simple and fast approaches to estimate the number of nodes can be found in [12,13].

2.2 Performance metrics

We evaluate the proposed selection algorithms based on three metrics which are reliability, message complexity, and delay, respectively.

2.2.1 Reliability

The probability that node selection is successful, i.e., at least one singleton slot occurs within s contention slots, is

$$R(n, s, \mathbf{P}) = 1 - \prod_{j=1}^s \left(1 - \sum_{i=1}^n p_{ij} \prod_{k=1, k \neq i}^n (1 - p_{kj}) \right). \quad (2)$$

This term simplifies for node-independent access probabilities to

$$R(n, s, \vec{p}) = 1 - \prod_{j=1}^s (1 - n p_j (1 - p_j)^{n-1}). \quad (3)$$

Furthermore, we define $R(0, s, \cdot) := 0$ and $R(n, 0, \cdot) := 0$.

2.2.2 Message complexity

The number of reply messages required for a selection attempt is called message complexity M . For the standard protocol (i.e., no early stop), the probability mass function (pmf) is

$$\mathbb{P}(M = r) = \sum_{\substack{\vec{x} \in \{0,1,\dots,n\}^s \\ \sum_{i=1}^s x_i = r}} \prod_{j=1}^s \binom{n}{x_j} p_j^{x_j} (1 - p_j)^{n-x_j}, \quad (4)$$

where $0 \leq r \leq ns$ (see Appendix B.1 in Additional file 1). The expected number of messages is $\mathbb{E}(M) = \sum_{j=1}^s n p_j$, and its variance is $\text{Var}(M) = \sum_{j=1}^s n p_j (1 - p_j)$. If each node has the same sending probability in each contention slot, $p := p_j \forall j$, the pmf simplifies to a binomial distribution

$$\mathbb{P}(M = r) = \binom{ns}{r} p^r (1 - p)^{ns-r}. \quad (5)$$

When applying early stop, the pmf of message complexity is given by replacing the upper limit s in (4) with

$$z := \min(\{1 \leq t \leq s | x_t = 1\} \cup \{s\}).$$

The expected message complexity with early stop can be expressed as

$$\mathbb{E}(M) = \sum_{j=1}^s n p_j (1 - R(n, j - 1, \vec{p})). \quad (6)$$

We refer to Appendix B.2 in Additional file 1 for derivations of these expressions.

2.2.3 Delay

For the standard protocol, the delay D equals the number of slots s . When applying early stop, D equals the slot number where the first non-colliding message occurs. Then, the pmf of D is (see Appendix B.2 in Additional file 1)

$$\mathbb{P}(D = d) = \begin{cases} R(n, d, \vec{p}) - R(n, d - 1, \vec{p}), & 1 \leq d \leq s - 1 \\ 1 - R(n, s - 1, \vec{p}), & d = s. \end{cases} \quad (7)$$

The expected delay $\mathbb{E}(D)$ and its variance $\text{Var}(D)$ can be calculated from (7).

2.2.4 Cumulative message complexity and delay

Note that so far $E(M)$ and $E(D)$ express message complexity and delay, respectively, for a single selection independent of whether or not the selection was successful. A simple strategy to handle unsuccessful selections is to

repeat the selection until it is successful. Values for the expected cumulative message complexity and delay are given by $cE(M)$ and $cE(D)$ with a proportionality constant c . As shown in Appendix B.3 in Additional file 1, this constant is $c = 1/R(n, s, p)$. For example, five nodes with ten contention slots yield $c = 1.067$ for slow-start and $c = 1.005$ for uniform access.

2.3 Access strategies

We propose two access strategies, where the first one (uniform access) maximizes reliability while the second one (slow-start access) maximizes the probability that only one reply is sent under early stop, i.e., the message complexity of the selection is reduced.

2.3.1 Uniform access

We assume node-independent access probabilities $\vec{p} := (p_1, \dots, p_s)$ and maximize these with respect to (3) for given n and s . The probabilities maximizing (3) are given as

$$\vec{p}_{\frac{1}{n}} := \left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n} \right). \quad (8)$$

This property is proven in Appendix A.1 in Additional file 1. The approach using (8) as sending probabilities in the random access period of the selection protocol is called *uniform access*.

2.3.2 Slow-start access

The objective of uniform access is to maximize the reliability R . When evaluating its message complexity using equation (6), we see that, e.g., for $n = 5$ nodes and $s = 10$ slots, the expected number of replies is 2.43. This high message complexity demands for a different, non-uniform slot access policy to reduce this complexity. Therefore, we maximize the probability that only one reply is sent when applying early stop, i.e.,

$$\mathbb{P}(M = 1) \xrightarrow{\text{early stop}} \max.$$

The expression for $\mathbb{P}(M = 1)$ is

$$\Phi(n, s, \vec{p}) = n \sum_{i=1}^s \left(\prod_{w=1}^{i-1} (1 - p_w)^n \right) p_i (1 - p_i)^{n-1}, \quad (9)$$

where $\prod_{w=1}^0 (1 - p_w)^n := 1$.

The node-independent access probabilities $\vec{p}_{\Phi} = (p_1, p_2, \dots, p_s)$ that maximize (9) are

$$p_{s-k} = \frac{1}{n} \cdot \frac{(n-1)^k - n \cdot \alpha_k \cdot \beta_k}{(n-1)^k - \alpha_k \cdot \beta_k} \quad (10)$$

for all $k \in \{0, \dots, s - 1\}$ with $\alpha_k := \left(\frac{n-1}{n}\right)^k$ and the recursively defined term

$$\beta_k := \begin{cases} 0 & \text{for } k = 0 \\ ((n-1)^{k-1} - \alpha_{k-1} \cdot \beta_{k-1})^{1-n} & \text{else.} \end{cases}$$

Appendix A.2 in Additional file 1 gives derivations of these expressions. The resulting algorithm is called *slow-start access*.

2.4 Performance evaluation

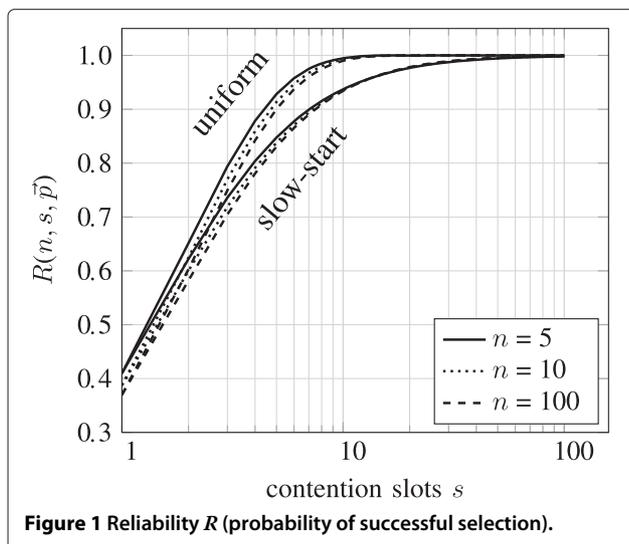
We evaluate the performance of both access strategies in terms of reliability, message complexity, and delay.

2.4.1 Reliability

Figure 1 plots the reliability R over s for n nodes applying uniform access (upper curves). Clearly, the more contention slots and the fewer nodes there are, the more likely the selection process will succeed. It is important to note that the number of nodes has in fact only minor influence on the success probability for reasonable values of n . The reliability is always better than 80% if more than four contention slots are used. Successful selection is achieved with high probability using ten contention slots or more. The reliability of node selection using slow-start access is slightly lower than that of uniform access (Figure 1, lower curves). Nevertheless, selection success is achieved almost always and independently of n if at least $s = 40$ contention slots are used.

2.4.2 Message complexity

Figure 2a shows the pmfs of message complexity for the standard protocol for $n = 5$ and $s = 10$. The pmf of uniform access shows that it is very likely to receive many replies. For slow-start access, however, the pmf is much more pronounced at lower numbers of replies. Table 1 lists



$\mathbb{E}(M)$ and $\text{Var}(M)$ for some typical values of n and s . As can be seen, the expected message complexities using uniform access do not scale nicely with the number of slots, while for slow-start access, the expected message complexity is reduced significantly. Nevertheless, it increases for increasing s , although not as severely as with uniform access. When applying early stop, the pmfs are shown in Figure 2b ($n = 5, s = 10$). In both cases the message complexities are reduced tremendously; however, for slow-start access the probability of having only one reply is even higher than for uniform access. Furthermore, also the probabilities of having three, four, or five replies are significantly lower for slow-start access than for uniform access. Table 1 shows $\mathbb{E}(M)$ and $\text{Var}(M)$ for early stop. The expected message complexities are much lower now and do not increase significantly with s . Even for $n = 100$ and $s = 100$, it is low at 2.7 for uniform access

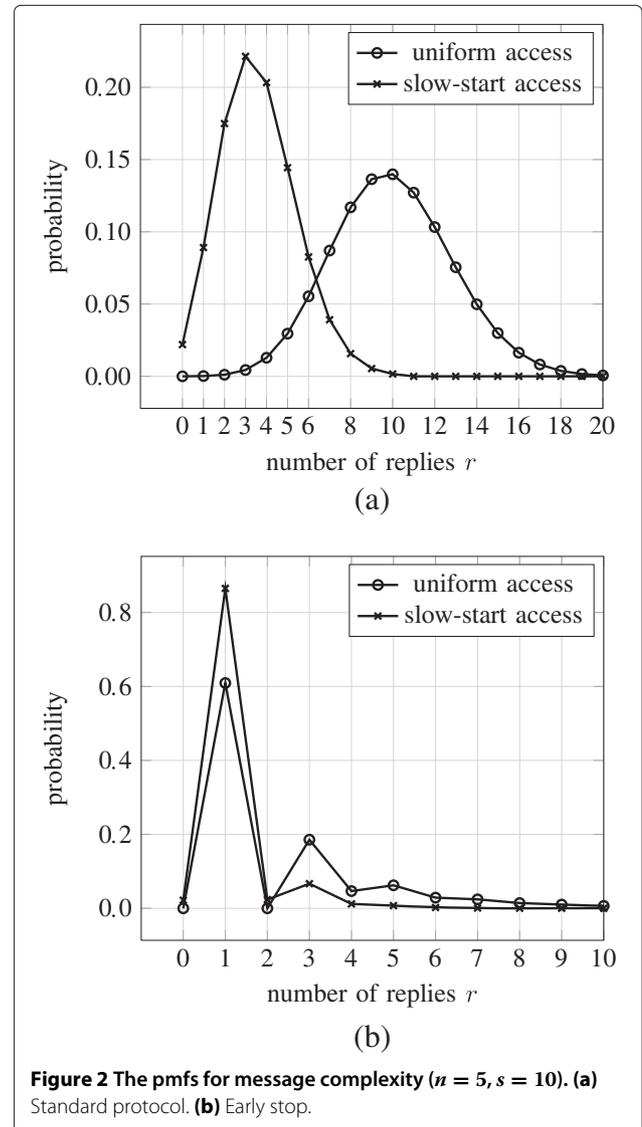
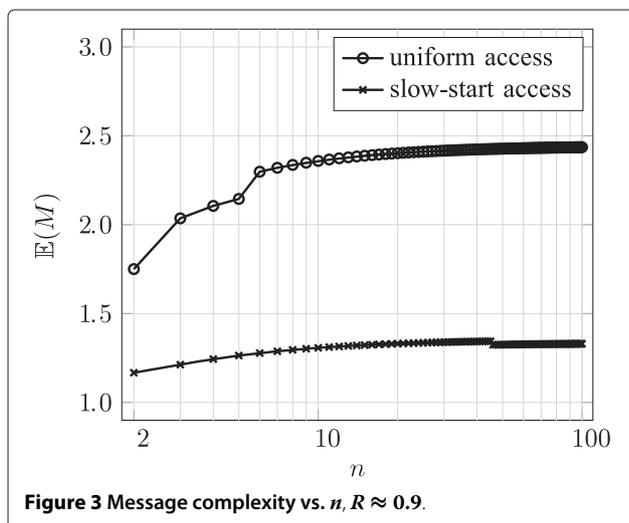


Table 1 Message complexity

Access	n	E(M)			Var(M)		
		s = 10	20	100	s = 10	20	100
Standard protocol							
Uniform	5	10	20	100	8	16	80
	10	10	20	100	9	18	90
	100	10	20	100	9.9	19.8	99
Slow-start	5	3.60	4.63	7.21	3.22	4.23	6.77
	10	3.72	4.82	7.61	3.52	4.61	7.38
	100	3.84	5.03	8.04	3.82	5.01	8.01
Early stop							
Uniform	5	2.43	2.44	2.44	5.21	5.48	5.52
	10	2.56	2.58	2.58	5.96	6.41	6.40
	100	2.68	2.70	2.70	6.86	7.29	7.17
Slow-start	5	1.22	1.13	1.03	0.60	0.35	0.08
	10	1.26	1.16	1.04	0.72	0.42	0.08
	100	1.29	1.18	1.04	0.82	0.48	0.09

and 1.04 for slow-start access. Node selection with slow-start access yields better performance than uniform access independently of the parameters. For example, with five nodes and ten contention slots, the expected number of replies is about 2.4 for uniform access and 1.2 for slow-start access. Furthermore, the expected number of replies approaches one for increasing s using slow-start access, while it approaches values between 2.4 and e using uniform access, depending on n . As shown in Appendix B.2 in Additional file 1, $\lim_{n,s \rightarrow \infty} E(M) = e$ for uniform access.

Figure 3 shows the expected message complexities for fixed reliability $R \approx 0.9$ and early stop. For each $n \in \{2, \dots, 100\}$, the number of slots s is determined such that



$|R(n, s, \vec{p}) - 0.9| \rightarrow \min$ and the corresponding message complexities are shown.

The figure illustrates that, for comparable reliabilities, slow-start access exhibits a message complexity that is approximately halved compared to uniform access. For $R = 0.98$ (not shown), the message complexity of slow-start remains at about one-half compared to uniform access. Note that for access probabilities $p_j \leq \frac{1}{n}$, the message complexities are almost independent of n and always - for both standard protocol and early stop - bounded by the number of slots s .

Proof. $E(M) = \sum_{j=1}^s n p_j \leq \sum_{j=1}^s n \frac{1}{n} = s$ □

As can be seen from Table 1, the expected message complexities are much lower for early stop than for the standard protocol.

2.4.3 Delay

For the standard protocol the delay of one selection is always equal to the number of slots s , independent of the access strategy. Therefore, in the following we only consider the delay when applying early stop.

Figure 4 shows the pmfs of delay for uniform access and slow-start access for increasing s . The pmfs show that for uniform access the probabilities of short delays are much higher than for slow-start access. Table 2 shows that for uniform access, the delay increases initially but then converges to a value between about 2.4 and 2.8 depending on n . For slow-start access, the delay is mainly depending on the number of slots s , i.e., $E(D) \approx 0.4 s$. Thus, there is a trade-off between low message complexity (slow-start access) and short delay (uniform access).

Figure 5 shows the expected delay for fixed reliability $R \approx 0.9$ and early stop. Uniform access exhibits a shorter

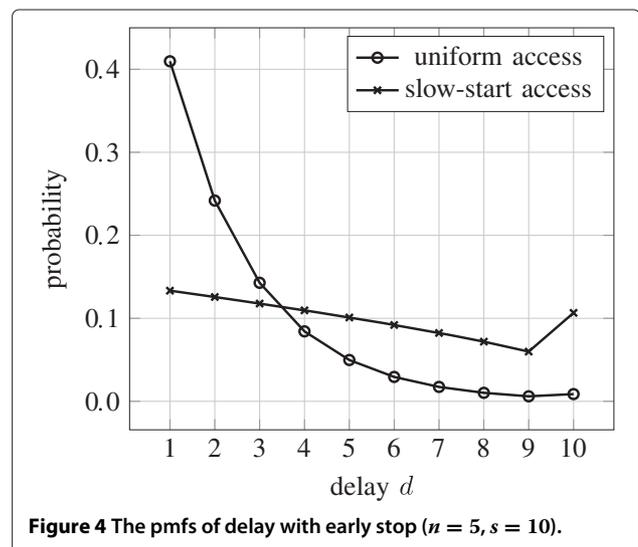


Table 2 Delay with early stop

Access	n	$\mathbb{E}(D)$			Var (D)		
		s = 10	20	100	s = 10	20	100
Uniform	5	2.43	2.44	2.44	3.28	3.52	3.52
	10	2.56	2.58	2.58	3.72	4.08	4.08
	100	2.68	2.70	2.70	4.10	4.60	4.61
Slow-start	5	5.00	8.80	38.90	8.54	30.67	668.80
	10	4.92	8.54	37.06	8.54	30.02	634.54
	100	4.84	8.28	35.27	8.53	29.31	599.72

delay than slow-start access. For the chosen reliability, the delay of uniform access is about half of that of slow-start access. For $R = 0.98$ (not shown), the delay of uniform access is about one-fourth of the delay of slow-start access.

2.5 Metric-based selection

So far, all nodes have been treated as equally suited to be selected. In many scenarios, however, it is beneficial to select a node that is better suited than others with respect to a certain metric (see, e.g., [1]). More suitable nodes should be preferred in the random access competition, i.e., such nodes should have a higher chance of sending a first non-colliding message. A similar concept is also applied, for example, in the context of quality-of-service support in wireless local area networks [14]. To obtain node-dependent slot access probabilities, we take the node-independent access probabilities and modify them by a metric that quantifies a node's capabilities. The metric is defined for each node i at time instant t as a real-valued number $m_i(t) \in [0, 1]$.

To be more specific, the approach is to apply for each node i a modification function $f(m_i(t))$ on the node-independent access probabilities p_j such that

$$p_{ij}(t) := p_j \cdot f(m_i(t)). \quad (11)$$

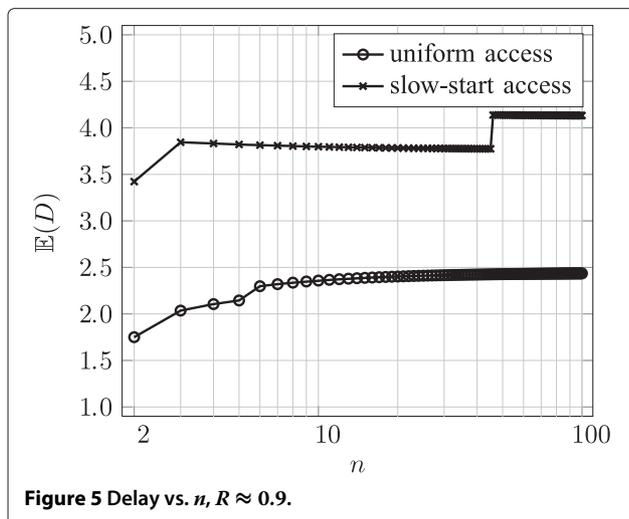


Figure 5 Delay vs. n, R ≈ 0.9.

We impose the following requirements on the modification function:

1. The resulting access probabilities $p_{ij}(t)$ must be properly bounded, i.e., $p_{ij}(t) \in [0, 1]$.
2. The function should allow for an adjustment of the modification intensity that is imposed on the probabilities p_j .
3. For each time instant t , the sum of node-dependent access probabilities $p_{ij}(t)$ in a given contention slot shall be approximately equal to the sum of node-independent sending probabilities, i.e., $\sum_{i=1}^n p_{ij}(t) \approx np_j$. This property ensures that performance measures such as reliability, message complexity, and delay, as shown in Section 2.4, are not significantly altered when applying metric-based selection.

A modification function that fulfills all requirements is given by

$$f(m_i(t), \bar{m}_i(t), \delta) := \frac{g(m_i(t), \bar{m}_i(t), \delta)}{\frac{1}{n} \sum_{k=1}^n g(m_{ik}(t), \bar{m}_i(t), \delta)}, \quad (12)$$

where δ is a parameter for adjusting the intensity of the modification, $\bar{m}_i(t) \in [0, 1]$ is the threshold value of node i at time instant t that determines whether its current metric $m_i(t)$ is 'good' or 'bad', and $m_{ik}(t)$ is node i 's estimate of node k 's metric. Furthermore, we define

$$g(m, \bar{m}, \delta) := n^{h(m, \bar{m}, \delta)}, \quad (13)$$

where $h(m, \bar{m}, \delta) := \text{sgn}(m - \bar{m}) \cdot \left\{ 1 - e^{-[(m - \bar{m}) \cdot \delta]^2} \right\}$ is the double logistic function.

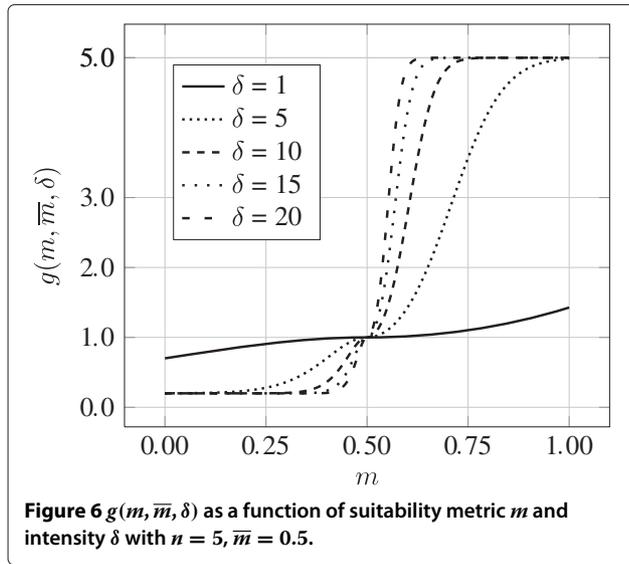
Appendix C in Additional file 1 proves that the node-dependent access probabilities are properly bounded, when applying $f(\cdot)$ on the node-independent probabilities for uniform access (8) and slow-start access (10). It also proves the third property, i.e., $\sum_{i=1}^n p_{ij}(t) \approx np_j$.

Figure 6 plots $g(m, \bar{m}, \delta)$ over m for different δ -values ($n = 5, \bar{m} = 0.5$). The figure visualizes that with increasing δ the modification performed on the probabilities intensifies and that for nodes with 'average' metric values, i.e., $m \approx \bar{m}$, $g(m, \bar{m}, \delta) \approx 1$.

If a node cannot estimate the metrics of the other nodes, the denominator of (12) is set to

$$\frac{1}{n} \cdot \left[g(m_i(t), \bar{m}_i(t), \delta) + (n - 1) \mathbb{E}_{M(t)} (g(M(t), 0, \delta)) \right], \quad (14)$$

where the distribution of the random variable $M(t)$ represents or approximates the distribution of the values $m_k(t) - \bar{m}_k(t)$. The term $\mathbb{E}_{M(\cdot)}$ denotes the expected value with respect to the random variable M .



2.6 Summary of the selection protocol

In summary, the selection protocol works as follows:

- The query node sets the system parameters: number of candidate nodes n and contention slots s .
- The query node broadcasts a request to all n candidate nodes. This message includes the values of n and s .
- Candidate nodes send a reply in each contention slot based on their metric (if we apply metric-based selection) and one of the access strategies. Slow-start access is applied if low message complexity is desired. Uniform access is applied if low delay is desired.
- If the selection is successful, the winner is informed by the query node in a control slot.

2.7 Related work

In general there are selection algorithms for slotted and non-slotted channels. In the work at hand, we focus on slotted channels since they are widely used in practical applications.

Many distributed algorithms for node selection are of deterministic nature (see, e.g., [5,7,8]). The message complexity of these algorithms depends on n . The slow-start algorithms presented in this article are of probabilistic nature and exhibit a message complexity that is in principle almost independent of n (see Table 1). In the following, we thus focus on probabilistic approaches for node selection.

Tay et al. [15] investigate and optimize the probability that the first channel access does not collide. The modeling assumptions defer, however, in the following ways: Tay et al. determine the probability distribution for nodes picking a certain slot. Consequently, each node accesses

exactly one slot, implying that the sum of the probabilities equals one. In the paper at hand, we determine node access probabilities for each slot. In particular this means that the sum of the probabilities is in general not equal to one and that nodes may access several slots or no slot at all. Due to the fact that we do not impose the requirement that each node must reply in exactly one slot, we achieve a higher selection reliability: When applying the probabilities of Tay et al. for, e.g., $n = 16$ and $s = 8$, we get a reliability of $R = 0.89036$, while for slow-start access we get $R = 0.91029$. For other parameters, similar results were obtained.

Tang et al. [16] propose a selection protocol where all nodes with metric values exceeding some threshold participate in the selection. These nodes access the channel with node-independent probabilities, which implies that they are regarded to be equally suited. Although we employ the same basic idea of channel access for performing node selection, we extend the work of Tang et al. [16] in the following ways: We propose, derive, and analyze two access probabilities, where the first one maximizes reliability while the second one reduces message complexity. In contrast to Tang et al., we propose a metric-based selection scheme, where nodes with better metrics are preferred, and analyze this scheme in two case studies.

Shah et al. [17] propose a selection protocol where metrics of nodes are mapped to discrete timers. The better the node's metric value, the shorter it waits until it accesses the shared medium. The assumption of this selection procedure is that metric values are samples of a continuous random distribution and, furthermore, that all metrics are statistically independent. Under these assumptions, the probability that metric values are equal is zero. In applications, however, metrics are often distributed discretely, e.g., received signal strength indicator (RSSI) in radio platforms [18]. In some applications, it might even be the aim that metrics become equal over time. We show such an application in Case study II: load balancing (Section 3.3), where the nodes' battery states become balanced over time. For such applications, the selection algorithm proposed by Shah et al. will suffer from collisions since the likelihood that the metric values of the best and second-best node are very close to each other (or even equal) is clearly nonzero. In contrast to this, the selection algorithm proposed in the paper at hand maintains a high reliability even if metrics are equal. Basically, in the extreme case where all nodes' metric values are equal, the metric-based selection becomes metric-independent, and its reliability is given in Figure 1.

Qin et al. [19] present an opportunistic algorithm allowing a node to send in each slot only if its metric falls within a specified interval. An extension of this algorithm is given by Yim et al. [20]. Nodes send at variable transmit powers. The main difference from the article at hand is that those

protocols start with a well-defined interval and split this interval at each stage into a lower and upper interval until the best node is selected. For the case where, e.g., the metric values of the best and the second-best node are very close to each other or even equal (e.g., if the assumption of statistical independence of the nodes' metrics does not hold), the message complexity of these algorithms will be high. We aim for low message complexity while retaining a high selection reliability even in cases where the best and second-best nodes exhibit the same metric value.

3 Case studies

Let us now demonstrate and analyze the metric-based selection in two case studies. The first study addresses the task of relay selection for cooperative wireless communications (Section 3.2). The second study is concerned with load balancing to increase network lifetime (Section 3.3). Before we address these case studies, however, we first consider related work on relay selection and load balancing in Section 3.1.

3.1 Related work

3.1.1 Relay selection

Bletsas et al. [21] describe a relay selection scheme with random access based on timers and channel estimation. Upon receiving a CTS packet, each relay candidate starts a timer. The value of these timers is based on channel measurements, and the timer of the relay with the best end-to-end channel condition will expire first.

Lo et al. [22] propose a relay selection scheme that favors nodes with better channel gains. All relay candidates compete on a slotted channel, where each one sends a reply with a defined probability in one randomly selected slot. This message contains information about the node's channel condition. The destination then chooses one of the candidates as the relay node. In our work, only nodes possessing a good value of the regarded metric reply, which in turn reduces the message complexity.

Nosratinia et al. [23] present a distributed relay selection protocol, where each node decides on its own whether or not to assist other nodes. Each node maintains a priority list used for cooperation purposes. In contrast to our proposal, the aim of this protocol is to achieve full diversity across the network.

Adam et al. [24] study relay selection with explicit consideration of the energy required to receive data. They propose a relay selection scheme that exhibits benefits with respect to energy efficiency. Nodes compete in the relay selection process by setting timers based on the channel state.

Lima et al. [25] present a contention-based relay selection mechanism. Nodes access the medium based on a common probability. Whenever a collision occurs, the relays split into subgroups. The goal is to get replies from

many relay candidates such that the source can select the most appropriate one.

These papers demonstrate the importance of a selection protocol for cooperative relay communications. The paper at hand performs a much more comprehensive analysis of such a relay selection protocol than previous work in this area.

3.1.2 Load balancing

Several algorithms for load balancing in task assignments have been proposed for mobile and wireless networks. Examples include the following: Tonguz et al. [26] present a mathematical framework for the evaluation of dynamic load balancing strategies in cellular networks. Alanyali et al. [27] focus on dynamic resource allocation and analyze the performance of two call assignment policies, i.e., least loaded routing and least relatively loaded routing. Further load balancing strategies are proposed and evaluated, e.g., in [28] and [29]. In the case study presented in this article, each node determines its suitability to perform a certain task on its own. The design of the random access scheme keeps the overhead at a low level.

3.2 Case study I: relay selection

A current field of application for contention-based selection can be found in cooperative wireless communications. In such systems, a radio link between a source and a destination node is supported by one or more relay nodes [2]. An important task is to select an appropriate relay from the set of all possible relay candidates to forward messages to the destination [24]. We focus on on-demand relaying techniques, where relaying is only performed if requested by the destination [24]. We assume that all relay candidates receive the message from the source. Relay selection is carried out by the destination.

3.2.1 Relay selection algorithm

The destination acts as query node and transmits a request message with transmit power κ . Relay selection should put preference on nodes having good channel conditions to the destination node. Assuming symmetric channels, each node measures the reception power of the selection requests from the destination and averages these reception powers over time.

The signal power received by a node i from the destination at a certain time instant t is called $\wp_i(t)$ (in dBm). The receiver sensitivity of node i - i.e., its minimum power for successful decoding - is denoted by θ_i (in dBm). Upon each reception of a request, a node i computes the metric

$$m_i(t) := 1 - \frac{\wp_i(t)}{\theta_i} . \quad (15)$$

The power values $\wp_i(t_1)$, $\wp_i(t_2)$, ... are stored by the node. The average received power over z selection requests is

$$\bar{\wp}_i(t_z) = 10 \log \left(\frac{1}{z} \sum_{\xi=1}^z 10^{\wp_i(t_\xi)/10} \right). \quad (16)$$

The metric threshold at time t is set to

$$\bar{m}_i(t) := 1 - \frac{\bar{\wp}_i(t)}{\theta_i}. \quad (17)$$

This selection process prefers nodes for which the instantaneous channel conditions are better, i.e., node i is regarded to be better than node k if

$$m_i(t) - \bar{m}_i(t) > m_k(t) - \bar{m}_k(t). \quad (18)$$

Considering a channel with pathloss exponent α and multipath propagation, the received power can be modeled by [30]

$$\wp_i(t) = \kappa - 10\alpha \log d_i + 10 \log H(t), \quad (19)$$

where the distance d_i between destination and node i is normalized to meters, and the random variable H describes multipath fading. In case of Rayleigh fading, H is exponentially distributed with mean 1. Furthermore, we require $H > 0$. For a sufficiently large number of measurements z , the average power becomes

$$\bar{\wp}_i(t_z) \rightarrow \kappa - 10\alpha \log d_i. \quad (20)$$

Combining (15), (17), (19), and (20) yields

$$m - \bar{m} = -\frac{10 \log H}{\theta} =: F \quad (21)$$

for each node i at time instant t . Thus, using (14), each node can compute the denominator of (12) by

$$\frac{1}{n} g(m_i(t), \bar{m}_i(t), \delta) + \frac{n-1}{n} \mathbb{E}_F(g(F, 0, \delta)). \quad (22)$$

Applying the resulting modification function $f(m_i(t), \bar{m}_i(t), \delta)$ in (11), each node computes its access probabilities for this selection and replies in each slot accordingly. For large-enough channel coherence time, the selected relay will exhibit good channel conditions for relay-destination communications.

3.2.2 Performance study

Let us now investigate the performance of this metric-based relay selection algorithm by simulations. As a case study, we employ TelosB nodes, which are programmable wireless platforms. The nodes have a transmit power $\kappa = 0$ dBm and a receiver sensitivity $\theta = -94$ dBm [18]. The pathloss exponent is set to $\alpha = 3$. Furthermore, we assume Rayleigh fading. We randomly place n relay candidates using a uniform distribution in an area of size $A = 1,200 \text{ m} \times 1,200 \text{ m}$. The query node (additional node)

is placed at the center of the area. Random access for relay selection is performed with $s = 10$ contention slots.

For each request sent by the query node, each relay candidate computes its metric and access probabilities. We perform 1,000 simulation runs, each consisting of 100 selection attempts. At the beginning of each simulation run, nodes are randomly (re-)distributed and selection thresholds are reset.

We first consider $n = 5$ relay candidates within the range of the query node. The reliability R of each selection is computed using (2). As the sending probabilities adjust to the current metric, R also changes over time. We visualize these changes of R for the entire simulation using boxplots. The whiskers of the boxplot visualize the 0.1 and 0.9 quantile, respectively. Note the different scalings of the y -axes to improve visibility. As shown in Figure 7, uniform access with intensity $\delta = 20$ yields high reliability, where more than 90% of all values are larger than $R = 98\%$. Using slow-start access, the reliabilities decrease. Nonetheless, 90% of all values are larger than $R = 86\%$. Applying $\delta = 10$ yields 99% and 91%, respectively.

Table 3 shows cumulative message complexities for different parameters. Slow-start access is always superior to uniform access, and in both cases, early stop can reduce the complexities significantly. The cumulative delays (not shown) follow a different behavior: they are lower for uniform access than for slow-start access. With $\delta = 20$, e.g., the average delay is 5.7 contention slots for slow-start access and 2.5 contention slots for uniform access. With $\delta = 10$, we obtain 5.4 and 2.4, respectively.

Besides reliability, message complexity, and delay, two further performance measures are analyzed:

1. The percentage of selections γ_1 where the *best* node is selected.
2. The percentage of selections γ_2 where the *best* or *second-best* node is selected.

Figure 8 shows γ_1 and γ_2 for different values of δ . The best node is selected in about $\gamma_1 = 35\%$ to 40% of all cases when using $\delta = 20$. This value decreases to about 28% when using $\delta = 10$. The best or second-best node is selected in about $\gamma_2 = 62\%$ for $\delta = 20$ or 50% for $\delta = 10$, respectively.

For uniform access, one can also apply more extreme values for δ , without decreasing the reliability too much: When applying, e.g., $\delta = 50$, the best node is selected in $\gamma_1 = 53\%$ of all cases, while the best or the second-best node is selected in $\gamma_2 = 78\%$ of all cases. The reliability remains at high levels, i.e., the median is 99.6% ; 90% of all reliabilities are larger than 96% .

Let us now increase the number of nodes from $n = 5$ to 20 . The resulting reliabilities of node selection are about the same as with low density. For $\delta = 20$, about 90% of all values are larger than $R = 85\%$ for slow-start

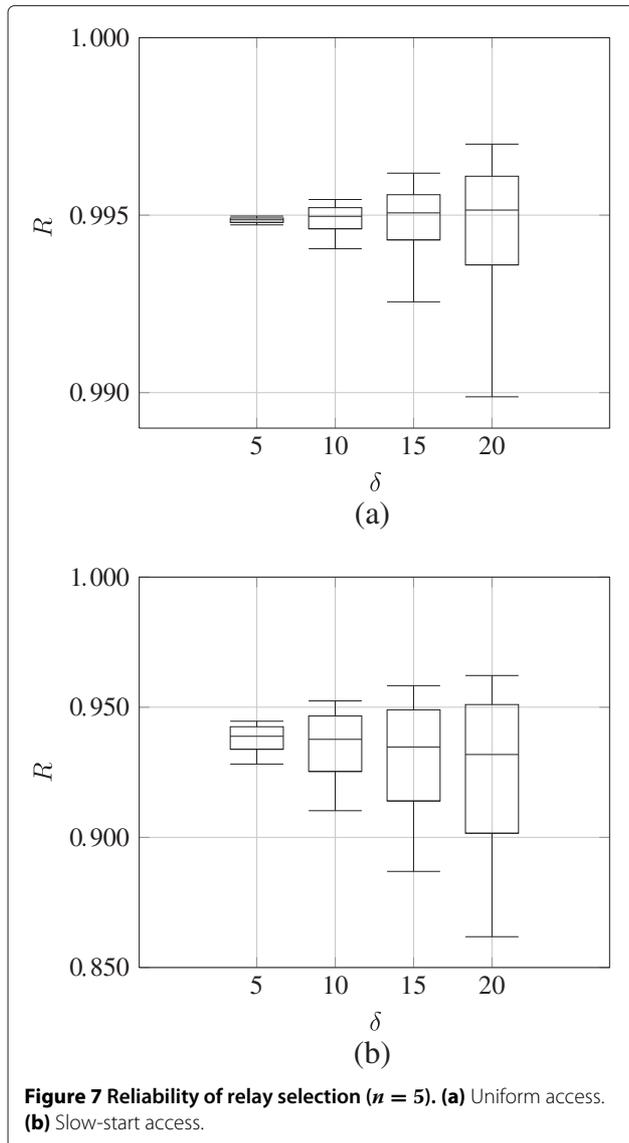


Figure 7 Reliability of relay selection ($n = 5$). (a) Uniform access. (b) Slow-start access.

access and larger than 98% for uniform access. As shown in Table 4, the average cumulative message complexities increase only slightly (cf. Table 3). Also, the average delays (not shown) are only slightly increased.

Last but not the least, the best node is selected for slow-start access in about $\gamma_1 = 26\%$ ($\delta = 20$) or 13% ($\delta = 10$) of all cases. For uniform access, the performance is $\gamma_1 = 28\%$ and 13% , respectively. The best or second-best node is selected at a rate of $\gamma_2 = 42\%$ or 22% applying slow-start access; it is 45% or 23% for uniform access. Roughly speaking, metric-based selection chooses the best node about five times as often as metric-independent selection ($\gamma_1 = 1/n = 5\%$) when using $\delta = 20$.

3.2.3 Impact of discrete RSSI values

In practical applications, the RSSI value provided by radio platforms is discrete. For the CC1100E low-power

Table 3 Message complexity of relay selection ($n = 5$)

Access	δ	Early stop		No early stop	
		Mean	Var	Mean	Var
Slow-start	mi	1.3	0.87	3.85	3.52
	5	1.30	0.87	3.86	3.54
	10	1.29	0.84	3.81	3.63
	15	1.27	0.77	3.71	3.68
	20	1.26	0.74	3.61	3.77
Uniform	mi	2.43	5.41	10.04	8.55
	5	2.46	5.63	10.10	8.86
	10	2.41	5.55	9.92	10.22
	15	2.34	5.36	9.59	11.10
	20	2.25	5.19	9.24	13.64

mi=metric-independent selection.

sub-GHz RF transceiver, for example, the RSSI value can be retrieved (in dBm) with a resolution of 0.5 dB [31]. Simulations showed that the performance of the proposed selection algorithm does not degrade due to this quantization.

3.3 Case study II: load balancing

Another application of node selection is to balance the load among several devices. We use a selection where devices with high battery charge states are preferred.

3.3.1 Load balancing algorithm

Assume that a given node requires another node to perform a certain energy-consuming task. It sends a request to all candidate nodes. Each candidate node i derives a metric m_i from its battery state $b_i \in [0, b_{\max}]$ by

$$m_i := \frac{b_i}{b_{\max}}. \quad (23)$$

We assume that a reply from node j includes its value m_j . Then, each node i updates its threshold \bar{m}_i as

$$\bar{m}_i := \frac{1}{n} \sum_{j=1}^n m_{ij}, \quad (24)$$

where m_{ij} is the value of node j 's metric received by node i from node j 's last reply. Furthermore, each node calculates its sending probabilities p_{ij} using (12) and accesses the contention slots according to these.

The first few selections are made with a metric-independent selection until nodes know (most) of the other nodes' metrics. Once all candidate nodes can well estimate the values m_{ij} , the protocol changes to metric-based selection.

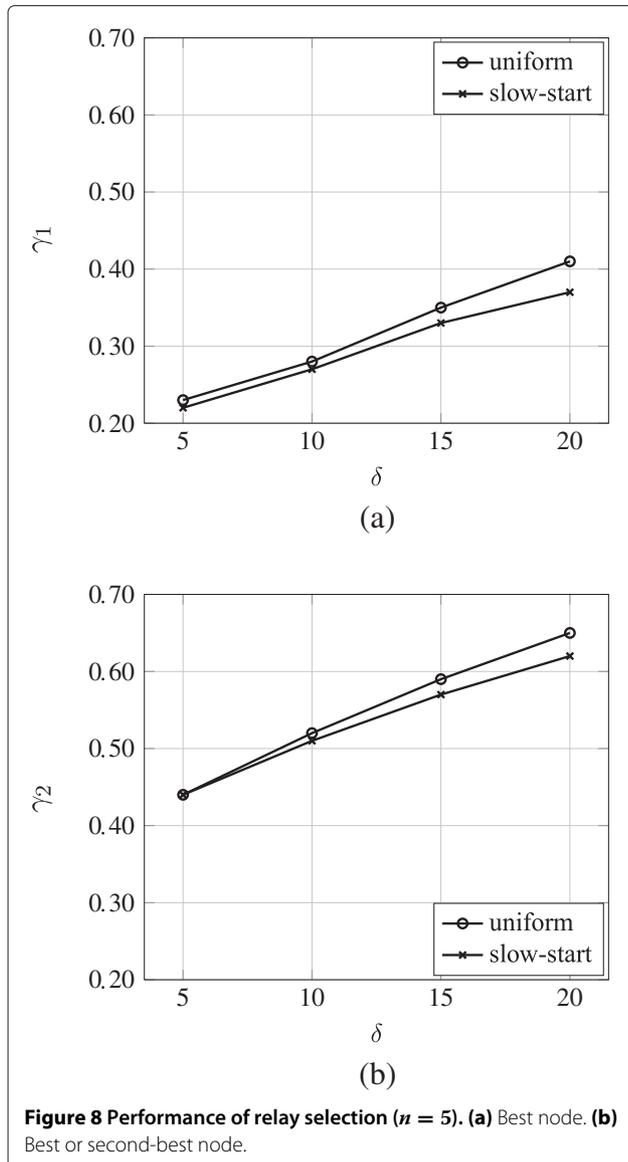


Figure 8 Performance of relay selection ($n = 5$). (a) Best node. (b) Best or second-best node.

3.3.2 Performance study

For performance analysis of contention-based load balancing, let us consider the specifications of the CC1100E low-power sub-GHz RF transceiver [31]. Each node is powered with a battery providing 3,000 mAh. For a data rate of 38.4 kbaud, the current consumption is 16.5 mA in transmission mode (at 0 dBm output power) and 17.7 mA in reception mode; it is only 700 nA in sleep mode. The time required to switch from sleep to transmit or receive mode is $240 \mu\text{s}$ at a current of 9 mA during state transitions. We use binary FSK as a modulation scheme. Each packet has a length of 88 bytes, including preamble/sync word of 4 bytes. We set the duration of each slot such that one packet can be transmitted during a single slot. All nodes listen for the preamble/sync word at the beginning of each slot. If no preamble/sync word is received, a node

Table 4 Message complexity of relay selection ($n = 20$)

Access	δ	Early stop		No early stop	
		Mean	Var	Mean	Var
Slow-start	mi	1.36	1.10	4.06	4.09
	5	1.37	1.14	4.10	4.19
	10	1.35	1.07	4.04	4.21
	15	1.33	1.01	3.94	4.39
	20	1.31	0.91	3.83	4.62
Uniform	mi	2.64	6.86	10.07	10.40
	5	2.69	7.19	10.20	10.98
	10	2.63	7.03	10.01	12.02
	15	2.53	6.80	9.66	14.75
	20	2.45	6.95	9.34	17.34

switches to sleep mode for the remaining slot duration and wakes up right before the next slot starts (cf. Figure 9). Based on these specifications, the energy consumption per slot when no reply is sent is

$$0.84 \cdot 17.7 + 0.24 \cdot 9 + 17.02 \cdot 0.0007 + 0.24 \cdot 9 \approx 0.02 \cdot 10^{-6} \text{ As},$$

while the energy consumption per slot when receiving is

$$18.34 \cdot 17.4 \approx 0.32 \cdot 10^{-6} \text{ As}.$$

Thus, the energy consumption of listening (i.e., no reply is received during the slot) is about 6% of that of receiving.

In the following, we assume that the task to be fulfilled requires 3.6 mJ. A practical task with such an energy consumption would be, e.g., measuring temperature and atmospheric pressure. The barometer specified in [32] requires a current of 5 mA, while the temperature sensor specified in [33] requires 4 mA. We average over a time period of 350 ms to get accurate measurement values. Thus, the energy consumption for performing the measurement is $(1.7 + 5 + 4) \cdot 0.35 \approx 3.6 \text{ mJ}$, where 1.7 mA is the current consumption of the CC1000E in idle mode.

The simulation is performed in the following way: Each node is assigned an initial battery state of charge. The query node sends a request to all candidate nodes containing the task to perform. The candidate nodes access the

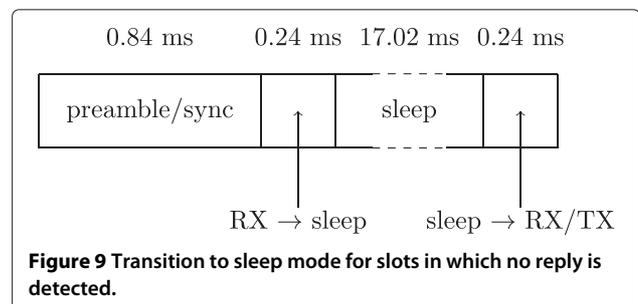
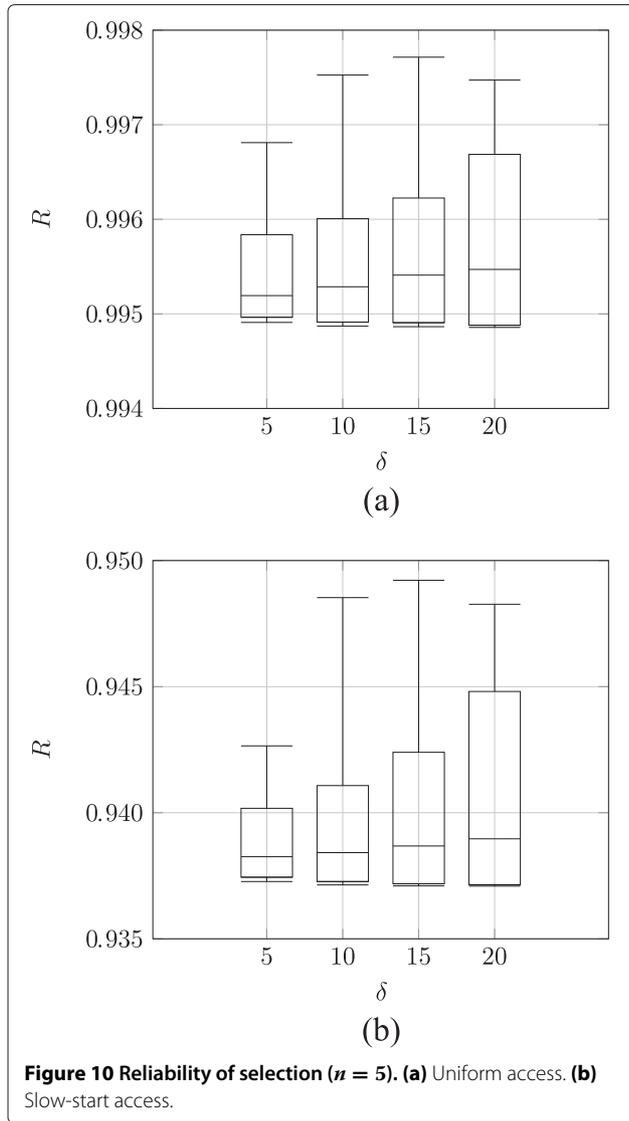


Figure 9 Transition to sleep mode for slots in which no reply is detected.



channel using uniform access or slow-start access, respectively. The node that sends the first non-colliding message will have to perform the task. If the selection is unsuccessful, it will be repeated until it is successful. Since nodes are not aware of other nodes' battery states in the beginning, we perform the first n selections always with the metric-independent strategy and then switch to the metric-based selection. As reference, we also evaluate the performance of metric-independent selection, i.e., the entire simulation is performed applying the metric-independent strategy. If not mentioned otherwise, node selection is performed over $s = 10$ contention slots. We compute the total energy needed for successful and unsuccessful selections and execution of the tasks. We also sum up the number of replies and delays of selections. The simulation runs until there is only one charged node left.

There are five candidate nodes with initial battery states $(b_1, \dots, b_5) = (3,000, 2,700, 2,400, 2,150, \text{ and } 1,520 \text{ mAh})$. The reliability R of each selection is evaluated using (2). As the sending probabilities adjust to the current metric, R also changes over time. Figure 10 shows boxplots of the reliabilities R based on a complete simulation for different values of δ . The whiskers of the boxplot represent the 0.1 and 0.9 quantile, respectively. Note the different scalings of the y-axes to improve visibility. Uniform access yields a very high reliability with a median value of 99.5% almost independent of δ . The reliability of slow-start access is worse, yielding a median of 93.8% and higher variance. A larger δ leads, in general, to a higher variance of the reliability as can be seen in Figure 10, but the influence of the metric on R is rather limited. In particular, we observe that the impact is smaller than in Case study I, which is due to the fact that each candidate node is informed about the other nodes' metrics during the selections, such that each node can adapt its sending strategy (more) appropriately.

Let us now study the network lifetime [34], defined as the time period from the beginning of the simulation until the point in time when the *first* node fails due to an empty battery. Figure 11 shows the battery levels of all nodes over time, comparing metric-independent and metric-based selection and comparing different strategies. The key observation is that metric-based selection extends the network lifetime for both access schemes by balancing the load between the nodes. For example, using slow-start access with early stop (Figure 11a), the first node runs out of battery after approximately five million tasks using metric-independent selection. Applying metric-based selection, however, the network lifetime increases to almost 8 million tasks for $\delta = 20$ or 7.3 million tasks for $\delta = 10$ (not shown). In general, node selection with uniform access slightly reduces the network lifetime, independent of whether or not metrics are employed for selection (cf. Figure 11a,b). This behavior is due to the fact that the energy consumption of listening is low compared to that of receiving. Applying node selection without early stop (Figure 11c,d), significantly reduces the network lifetime. This is because nodes have to listen for (and potentially receive) messages during all $s = 10$ contention slots.

There is again a trade-off between message complexity and delay. As shown in Table 5, slow-start access has almost half the message complexity of uniform access. For example, if $\delta = 20$, an average of about 1.3 replies is received for slow-start access compared to about 2.4 for uniform access. Uniform access, in turn, has about half the delay of slow-start access (no table).

The ratio between the energy consumption of listening and receiving determines as to whether or not slow-start

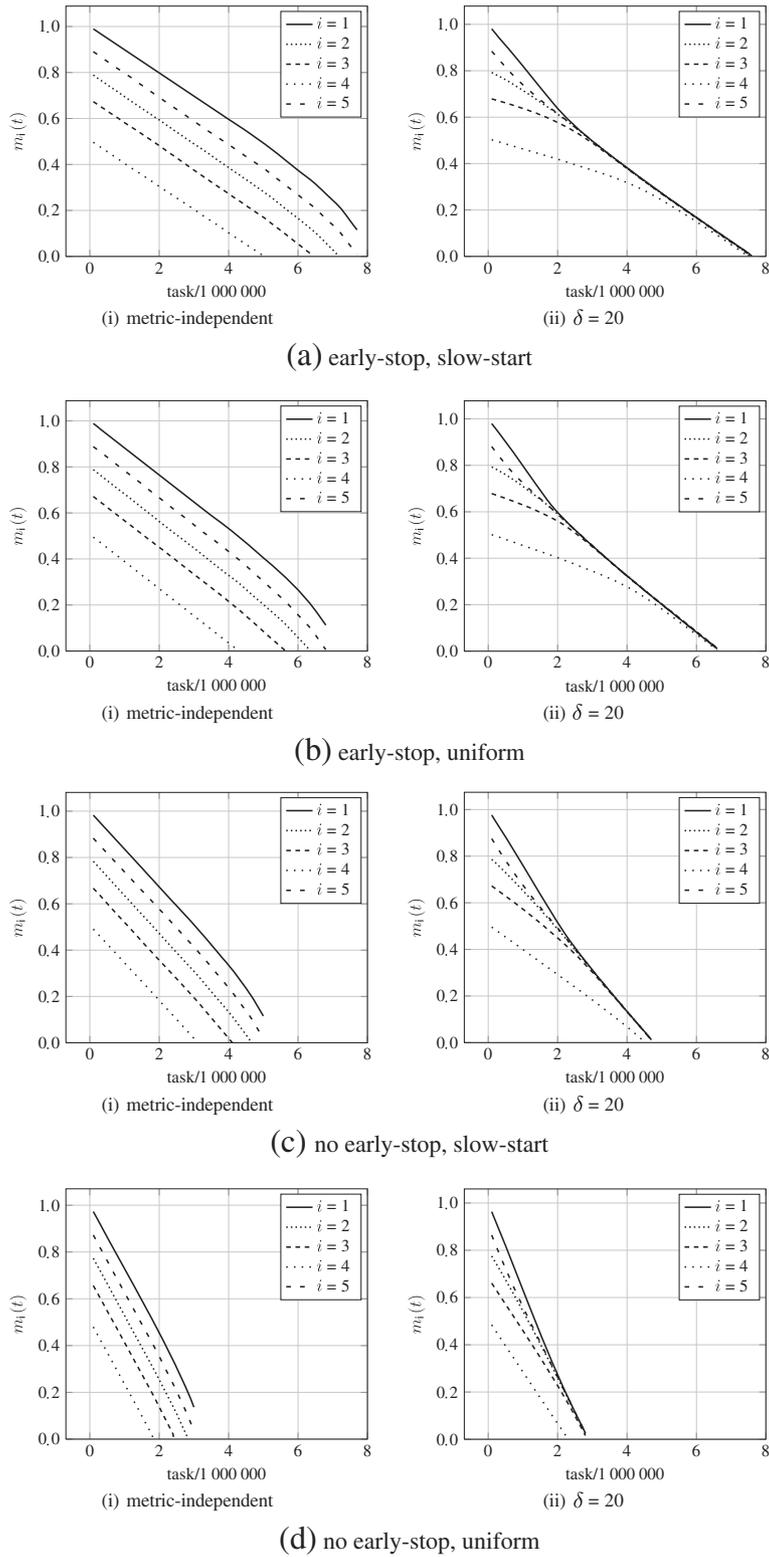


Figure 11 Battery consumption. Slow-start (a) and uniform (b) access with early stop. Slow-start (c) and uniform (d) access with no early stop.

Table 5 Message complexity of load balancing ($n = 5$)

Access	δ	Early stop		No early stop	
		Mean	Var	Mean	Var
Slow-start	mi	1.29	0.84	3.78	3.33
	5	1.29	0.82	3.82	3.35
	10	1.29	0.82	3.83	3.33
	15	1.28	0.78	3.81	3.29
	20	1.28	0.79	3.82	3.29
Uniform	mi	2.40	5.21	10.08	8.10
	5	2.42	5.32	10.03	7.90
	10	2.36	4.99	10.02	7.53
	15	2.37	5.03	10.04	7.40
	20	2.37	5.18	10.02	7.32

access exhibits benefits in terms of energy consumption. To analyze this further, we derive the expected energy consumption per node as the sum of the weighted probabilities that either a node is sending, listening, or receiving:

$$E(n, s, \vec{p}, e_S, e_R, \tau) = \sum_{j=1}^s [p_j e_S + (1 - p_j) (1 - p_j)^{n-1} \tau e_R + (1 - p_j) (1 - (1 - p_j)^{n-1}) e_R] \cdot [1 - R(n, j - 1, \vec{p})], \quad (25)$$

where e_S and e_R are the energy consumptions per slot for sending and receiving, respectively, and $0 \leq \tau \leq 1$ expresses the ratio of e_R needed for listening. To determine the value of τ for which both uniform and slow-start access exhibit equal expected energy consumption, denoted by τ^* , we solve

$$\frac{E(n, s, \vec{p}_{\frac{1}{n}}, e_S, e_R, \tau)}{R(n, s, \vec{p}_{\frac{1}{n}})} = \frac{E(n, s, \vec{p}_{\Phi}, e_S, e_R, \tau)}{R(n, s, \vec{p}_{\Phi})}, \quad (26)$$

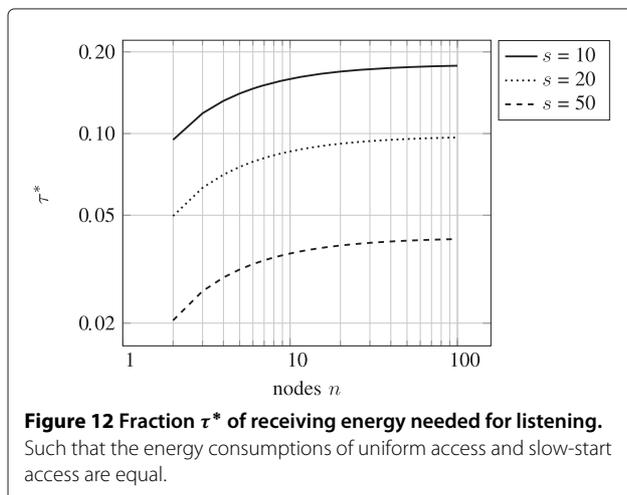


Table 6 Initial battery states ($n = 20$)

i	b_{i+1}	b_{i+2}	b_{i+3}	b_{i+4}	b_{i+5}
0	3,000	2,400	2,050	1,520	2,700
5	2,600	2,000	3,000	2,900	2,400
10	1,800	1,900	2,500	2,800	2,950
15	1,600	1,900	2,130	2,750	1,800

where $e_S = 16.5$ mA and $e_R = 17.7$ mA. Figure 12 shows the values of τ^* for different n and s . The value depends strongly on s and depends somewhat on n . For $s = 10$ and $n \geq 5$, τ must be smaller than about 0.15 to yield a benefit of slow-start access in terms of energy efficiency. For the same s and $n = 100$ nodes, this value is about 0.18.

Let us perform the same analysis using $n = 20$ nodes with initial battery states shown in Table 6. The obtained reliabilities of metric-based selection are very similar to those with low node density. For example, more than 90% of all values are larger than 93% for slow-start access and 99% for uniform access ($\delta = 20$). Table 7 shows the message complexities. Both message complexity and delay increase only very slightly compared to low node density. Results on the network lifetime can be summarized as follows: Slow-start access with early stop enables about 9 million tasks for metric-independent selection and about 12.7 million tasks for metric-based selection with $\delta = 20$. Uniform access with early stop reduces the lifetime to 7.3 million and 9.7 million tasks, respectively.

4 Conclusions

This article presented and analyzed a general method for distributed node selection using a random access competition on a shared slotted channel. Two access strategies were proposed: uniform access maximizes reliability; slow-start access reduces message complexity. Making slot access probabilities dependent on a node metric, both

Table 7 Message complexity of load balancing ($n = 20$)

Access	δ	Early stop		No early stop	
		Mean	Var	Mean	Var
Slow-start	mi	1.35	1.07	4.03	3.99
	5	1.35	1.07	4.02	3.95
	10	1.35	1.06	4.02	3.90
	15	1.35	1.06	4.01	3.86
	20	1.35	1.06	3.96	3.85
Uniform	mi	2.62	6.71	10.06	10.10
	5	2.62	6.67	10.06	9.81
	10	2.59	6.51	9.96	9.46
	15	2.59	6.41	9.84	9.45
	20	2.59	6.52	9.66	9.47

schemes can be applied for node selection. We analyzed the selection protocol theoretically in terms of reliability, message overhead, and delay. Furthermore, we tested the protocol in two case studies, namely relay selection and load balancing.

For the application of relay selection, we applied the channel state as the metric and showed that the metric-based strategy selects one of the better relays with high likelihood compared to metric-independent selection. For the application of load balancing we applied the battery state of a node as the metric. Nodes are selected to perform a certain task. To balance the battery consumption of the nodes, we put preference on nodes that have higher battery states. The metric-based selection increases the network lifetime considerably compared to metric-independent selection.

Additional file

Additional file 1: Appendix. Document contains proofs and derivations.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Mobile Systems Group, Institute of Networked and Embedded Systems, University of Klagenfurt, Klagenfurt 9020, Austria. ²Lakeside Labs GmbH, Klagenfurt 9020, Austria.

Received: 15 October 2012 Accepted: 4 August 2013

Published: 19 August 2013

References

1. S Basagni. Distributed clustering for ad hoc networks, in *Proceedings of the International Symposium on Parallel Architectures, Algorithms, and Networks (ISPAN)* (Perth, 2–25 June 1999)
2. JN Laneman, DNC Tse, GW Wornell, Cooperative diversity in wireless networks: efficient protocols and outage behavior. *IEEE Trans. Inform. Theory*. **50**(12), 3062–3080 (2004)
3. J Chou, D Petrovic, K Ramachandran. A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks, in *Proceedings of the IEEE INFOCOM* (San Francisco, 30 March–3 April 2003)
4. G Tel, *Introduction to, Distributed Algorithms*, 2nd edn. (Cambridge University Press, Cambridge, 2000)
5. H Garcia-Molina, Elections in a distributed computing system. *IEEE Trans. Comput.* **C-31**, 48–59 (1982)
6. R Gallager, P Humblet, P Spira, A distributed algorithm for minimum-weight spanning trees. *ACM Trans. Program. Lang. Sys.* **5**, 66–77 (1983)
7. E Korach, S Kutten, S Moran, A modular technique for the design of efficient distributed leader finding algorithms. *ACM Trans Program. Lang. Sys.* **12**, 84–101 (1990)
8. E Chang, R Roberts, An improved algorithm for decentralized extrema-finding in circular configurations of processes. *Commun. ACM*. **22**, 281–283 (1979)
9. M Ramanathan, R Ferreira, S Jagannathan, A Grama, W Szpankowski, Randomized leader election. *Distributed Comput.* **19**, 403–418 (2007)
10. C Bettstetter, G Brandner, R Vilzmann. On colliding first messages in slotted ALOHA, in *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)* (Cannes, 15–18 Sept 2008)
11. G Brandner, U Schilcher, M Gyarmati, C Bettstetter. Non-colliding first messages: further insights toward a practical solution, in *IEEE Vehicular Technology Conference* (Barcelona, 26–29 April 2008)
12. Adam H, Yanmaz E, Elmenreich W, C Bettstetter. Contention-based neighborhood estimation, in *Proceedings IEEE Vehicular Technology Conference (VTC)* (Taipei, 16–19 May 2010)
13. M Kodialam, T Nandgopal, W Lau. Anonymous tracking using RFID tags, in *Proceedings of the IEEE INFOCOM* (Anchorage, 6–12 May 2007)
14. IEEE Standards Association, *IEEE 802.11e, wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 8: medium access control (MAC) quality of service enhancements*. (IEEE Tech. rep, New York, 2005)
15. Y Tay, K Jamieson, H Balakrishnan, Collision-minimizing CSMA and its applications to wireless sensor networks. *IEEE J. Select. Areas Commun.* **22**(6), 1048–1057 (2004)
16. T Tang, R Heath, Opportunistic feedback for downlink multiuser diversity. *IEEE Commun. Lett.* **9**(10), 948–950 (2005)
17. V Shah, N Mehta, R Yim, Optimal timer based selection schemes. *IEEE Trans. Commun.* **58**(6), 1814–1823 (2010)
18. TelosB datasheet (2009). http://www.willow.co.uk/TelosB_Datasheet.pdf. Accessed 20 June 2013
19. X Qin, R Berry. Exploiting multiuser diversity for medium access control in wireless networks, in *Proceedings of the IEEE INFOCOM* (San Francisco, 30 March–3 April 2003)
20. R Yim, N Mehta, A Molisch. Best node selection through distributed fast variable power multiple access, in *Proceedings of the IEEE International Conference on Communications (ICC)* (Beijing, 19–23 May 2008)
21. A Bletsas, A Khisti, DP Reed, A Lippman, A simple cooperative diversity method based on network path selection. *IEEE J. Select. Areas Commun.* **24**, 659–672 (2006)
22. C Lo, R Heath, S Vishwanath. Opportunistic relay selection with limited feedback, in *IEEE Vehicular Technology Conference* (Dublin, 22–25 April 2007)
23. A Nosratinia, TE Hunter, Grouping and partner selection in cooperative wireless networks. *IEEE J. Select. Areas Commun.* **25**(2), 369–378 (2007)
24. H Adam, C Bettstetter, SM Senouci. Adaptive relay selection in cooperative wireless networks, in *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)* (Cannes, 15–18 Sept 2008)
25. C de Lima, G de Abreu. Analysis of contention-based relay selection mechanisms in autonomous multi-hop networks, in *Proceedings of the IEEE Information Theory Workshop on Network Information Theory* (Volos, 10–12 June 2009)
26. O Tonguz, E Yanmaz, The mathematical theory of dynamic load balancing in cellular networks. *IEEE Trans. Mobile Comput.* **7**(12), 1504–1518 (2008)
27. M Alanyali, B Hajek. On simple algorithms for dynamic load balancing, in *Proceedings of the IEEE INFOCOM* (Boston, 2–6 April 1995)
28. G Gupta, M Younis. Performance evaluation of load-balanced clustering of wireless sensor networks, in *Proceedings of the IEEE International Conference on Communications (ICC)* (Anchorage, 23 Feb–1 March 2003)
29. Y Bejerano, SJ Han, LE Li. Fairness and load balancing in wireless LANs using association control, in *Proceedings of the ACM MobiCom* (New York, 2004)
30. A Goldsmith, *Wireless Communications*. (Cambridge University Press, Cambridge, 2005)
31. Texas Instruments, CC1100E: low-power sub-GHz RF transceiver (2013). <http://www.ti.com/lit/ds/swrs082/swrs082.pdf>. Accessed 20 June 2013
32. Ammonit, Barometric pressure sensor AB 60/AB 100 (2012). http://www.ammonit.com/images/stories/download-pdfs/DataSheets/AirPressure/EN_DS_Barometer_AB60-AB100_40.pdf. Accessed 20 June 2013
33. Global Water, WQ101 temperature sensor (2011). <http://www.globalwater.com/downloads/WQ/WQ101B.pdf>. Accessed 20 June 2013
34. I Kang, R Poovendran. Maximizing static network lifetime of wireless broadcast ad hoc networks, in *Proceedings of the IEEE International Conference on Communications (ICC)* (Anchorage, 11–15 May 2003)

doi:10.1186/1687-1499-2013-211

Cite this article as: Brandner et al.: Contention-based node selection with applications to relay communications and load balancing. *EURASIP Journal on Wireless Communications and Networking* 2013 **2013**:211.