Fast information exchange in proximity-based multichannel wireless networks

António Gonga, Themistoklis Charalambous, and Mikael Johansson

Abstract— This paper considers the problem of distributed neighbor discovery in multi-channel wireless networks. We propose a protocol in which nodes randomly select a channel and decide whether to transmit or listen for neighbor discovery beacons. When nodes transmit, they use epidemic information dissemination to spread knowledge about all the nodes they have discovered so far. Theoretical guarantees on discovery times are complemented by extensive simulations and practical implementations. The evaluations show that multi-channel communication effectively reduces the number of collisions between nodes in the network (especially in dense networks) and that epidemic information dissemination yields both significant speed-ups and increased resilience to packet losses. Finally, we also show that our protocol compares favorably to previously proposed solutions in the literature.

Index Terms— Randomized neighbor discovery, wireless networks, multichannel communications, epidemic dissemination.

I. INTRODUCTION

Ad hoc networks are becoming increasingly important due to a number of emerging applications, such as low-power wireless sensing and proximity-based communications. Such networks are usually required to be self-configured, even in conditions at which the network structure changes (because of environmental changes, mobility, etc.). One aspect of self-configuration is neighbor discovery, *i.e.* to gather and maintain information about nodes within direct communication range. This information is then used for medium access control, multi-hop communications, service discovery, etc. It is desirable that neighbor discovery is done rapidly, since (i) it allows other operations to start quickly (*e.g.*, medium access control) and (ii) it decreases energy consumption.

Early work in neighbor discovery can be classified as either *randomized* or *deterministic*. In this paper, we focus on randomized neighbor discovery (*e.g.*, [1]), where each node independently and randomly decides whether to transmit or listen. Performance guarantees are given in terms of expected time to discover all nodes, or the earliest time after which one can guarantee that all nodes have been discovered *almost surely*. Most randomized neighbor discovery protocols use single channel communication. The seminal paper [1] analyzed the single channel neighbor discovery problem as a coupon collector problem. The analysis shows that as the network size increases, the idle slot probability must increase and will eventually grow as large as 36.8%. As

a result, significant energy is wasted on listening in empty time slots; an attempt to use higher transmission probabilities would lead to more collisions, thus increasing the energy consumption even more. In order to alleviate these effects, one has to depart from single channel operation.

In this paper, we present a distributed algorithm for epidemic neighbor discovery in multichannel wireless networks that is shown to discover all its neighboring nodes with low latency. This low-latency neighbor discovery protocol is particularly beneficial for low-power networks, such as wireless sensor networks, where energy conservations is a major concern. Our work is based on the following key ideas:

1) **Multichannel communications:** We exploit multichannel communications to reduce the number of collisions [2]. Existing low-power wireless standards, such as Bluetooth, 802.11 and 802.15.4, all support multichannel communication, and it is becoming increasingly important also in emerging standards such as IEEE 802.15.4e [3]. However, as most low-power radio nodes can only listen to one channel at a time, using multiple channels also reduces the chances that a node will overhear a transmitted beacon.

2) **Epidemic dissemination:** Nodes include a list of already discovered neighbors in their beacons. This epidemic information dissemination allows for nodes to discover neighbors indirectly, which speeds up the overall discovery process and increases the resilience to link losses. Epidemic algorithms have been used in peer-to-peer systems and database replication in wired networks [4], as well as for gossiping in radio networks [5], [6]. An important contribution of this work is to show how to optimally choose the number of channels to match the expected network size.

Based on these ideas, we make the following contributions:

- First, to assess the potential of multichannel epidemic discovery, we analyze a three-phase protocol, showing that it allows for a discovery-time speed up on the order of $\sqrt{N}$ in a clique network with N nodes.

- Then, we present an epidemic neighbor discovery algorithm for multichannel networks that reduces the discovery-times considerably compared to the single-channel protocol proposed in [1].

- Next, we investigate the performance of our algorithm in networks with link failures and we show that our algorithm improves the expected discovery-time; in the case of a single channel only, as the number of nodes increases, epidemics recover the optimal performance of a

A. Gonga, T. Charalambous and M. Johansson are with the Department of Automatic Control, School of Electrical Engineering, Royal Institute of Technology (KTH), Stockholm, Sweden (Emails: {gonga,themisc,mikaelj}@ee.kth.se).

single channel discovery under reliable communications. In dense networks with high loss probabilities, discovery times can be improved by an order of magnitude.

- Finally, we depart from clique networks and consider general multi-hop networks. In this case, we restrict the dissemination of information to a limited number of hops. Simulations show a strong performance of our protocol.

II. SYSTEM MODEL AND PRELIMINARIES

A. System model

Network model: We consider a radio network consisting of N nodes. Nodes are assigned unique identifies in the range $\{1, 2, .., N\}$. Initially, we assume a *clique* network where each node is in communication range with every other node. Later we extend the model to general multi-hop networks.

Radio model: Each node is equipped with a half-duplex radio transceiver (*i.e.*, nodes either transmit or receive but not both simultaneously). There are k orthogonal channels, and each node can transmit or receive packets on any of the k channels. The network is assumed to operate in synchronous mode, where time is divided into discrete time slots of fixed duration. Each time slot allows for the transmission or the reception of a single packet (we do not allow multipacket transmissions per time slot). At any time slot t, each node selects one of the k available channels, uniformly at random.

Collision model: We say a collision occurs when more than one node decides to transmit on the same channel simultaneously. In addition, nodes do not possess any collision detection mechanism and cannot distinguish between packet collisions and idle time slots. Therefore, if a node i does not receive a packet at time slot t, it does not know if no neighbor (idle slot) or multiple neighbors transmitted.

Discovery model: A beacon transmission of node i at time t conveys information about all network nodes known to i at time t. We say that a *direct discovery* of node i occurs when a neighbor successfully receives the beacon sent by i. We say an *indirect* or *epidemic* discovery occurs if a node finds out about node i by a beacon sent by some other node.

Packet loss model: At any time slot t, reception of a packet from a neighboring node may fail independently at random. The probability of a link failure from node i to node j is denoted by γ_{ij}. Without loss of generality and for simplicity of exposition, we assume that the link failure for all links is the same, *i.e.*, $\gamma_{ij} = \gamma$, for all links (i, j) in the network.

B. Preliminaries

1) Randomized discovery in reliable single-channel networks: A randomized protocol for neighbor discovery in single-channel clique networks is proposed in [1], where nodes use Slotted Aloha to decide if they should attempt to broadcast a discovery beacon or if they should listen. In a clique of N nodes, under the assumption of reliable broadcast, the optimal transmit probability is proven to be $1/N$ and the expected number of slots required to discover the full network is

$$\mathbb{E}[T] = \sum_{m=0}^{N-1} \mathbb{E}[T_m] \approx NeH_N = Ne\big(\ln N + \Theta(1)\big), \quad (1)$$

where H_N denotes the Nth harmonic number.

A drawback of single-channel discovery protocols is that the idle probability tends to be high, which results in unnecessary long discovery times. For the protocol considered in [1], the probability of an idle slot $p_{\text{idle}} = \left(1 - \frac{1}{N}\right)^N$, which approaches $e^{-1} \approx 36.8\%$ as N grows large. When the broadcast is unreliable, *i.e.*, when only a subset of the nodes are able to successfully decode a beacon, the discovery times will be even longer. In the remaining parts of this paper, we demonstrate how epidemic information dissemination and multi-channel communication can decrease discovery times and increase robustness to unreliable communication.

C. Assess the potential of multi-channel epidemic discovery

Before we develop, analyze and evaluate our multi-channel discovery protocol, we use the elegant coupon collector analysis from [1] to assess the potential of multi-channel epidemic discovery. We do so by analyzing a theoretical three-phase protocol and show that it allows for a discovery-time speedup on the order of $\sqrt{N}$. We then introduce and analyze a simpler and more easily implementable protocol that retains many of the advantages of the theoretical protocol.

The theoretical protocol proceeds in three phases:

1) in the *first phase*, we run k parallel neighbor discovery processes (one per channel), each with N/k nodes (note that if N/k is not an integer we take integers close to N/k, so that the number of nodes in all channels is N);
2) in the *second phase*, one node from each of the k channels enters an epidemic dissemination process where nodes broadcast information about all nodes that they have discovered in their respective channels;
3) finally, the k nodes return to their original channels and broadcast information about all nodes in the network.

Admittedly, this protocol is not very practical, since the first and second phases have asymptomatic termination. However, it allows for a simple discovery time estimate.

In the first phase, each channel is running a single-channel epidemic neighbor discovery process with N/k nodes, each of which we can expect to complete in

$$\frac{N}{k}\left[\ln\left(\frac{N}{k}\right) + \Theta(1)\right]$$

slots. It will, in expectation, take a little extra time until all k channels have terminated, but we will disregard this fact in our analysis. In the second phase, k nodes participate in a coupon collector process with expected completion time of $k\big(\ln(k) + \Theta(1)\big)$ slots. The final phase is a single slot. Summing these expressions leads to a total expected discovery time of

$$\mathbb{E}(T_d) = \frac{N}{k}\left[\ln\left(\frac{N}{k}\right) + \Theta(1)\right] + k\big(\ln k + \Theta(1)\big) + \Theta(1),$$

therefore, $\mathbb{E}(T_d)$ can be approximated by

$$\mathbb{E}(T_d) \approx \frac{N}{k}\ln\left(\frac{N}{k}\right) + k\ln k.$$

To minimize this expression, we should pick k to balance
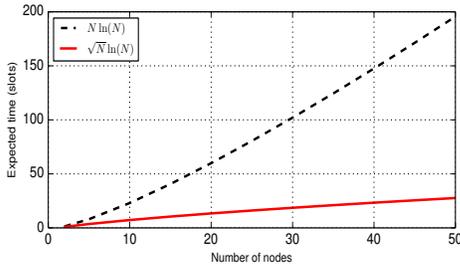
Fig. 1. For a network of N nodes using k channels, each node *epidemically* discovers all its neighbors in $\sqrt{N}\log(N)$ time slots. The speedup factor using epidemic information dissemination is roughly $\sqrt{N}$ compared to single channel neighbor discovery. The speedup factor coincides with the optimal number of channels k^*.

the two terms, *i.e.*, let $k = \sqrt{N}$, which gives

$$\mathbb{E}(T_d) \approx 2\sqrt{N}\ln\left(\sqrt{N}\right) = \sqrt{N}\ln(N).$$

Hence, the protocol is a factor $\sqrt{N}$ faster than the single-channel protocol studied in [1]; see Figure 1.

The discussion above highlights the potential of multi-channel epidemic discovery, but the protocol is not easy to implement. While an approximately equal distribution of nodes to channels could be done simply by hashing based on node IDs, and the leader election (of which node from each channel participates in the second protocol phase) could be done by assigning leadership to the node with, *e.g.*, highest ID in each channel, detection of termination of the neighbor discovery phases can only be done with probabilistic guarantees [1].

III. MULTICHANNEL EPIDEMIC DISCOVERY

The main disadvantage of the previous algorithm is that each phase has asymptotic termination. Realistically, it is very difficult for nodes to know when the next phase should begin. Instead, we propose a neighbor discovery protocol that operates in a single phase, and where nodes are distributed into channels by having each node picking a channel uniformly at random at every time slot. The protocol still exploits epidemic dissemination. Contrary to single channel communications, which is a single collision domain, multichannel communications creates smaller collision domains [3], [7] and allows multiple discovery events to occur in parallel. In addition, epidemic discovery allows for indirect discovery and a faster discovery process.

A. Algorithm operation

To describe the algorithm in detail, we let $M_i(t) \subseteq \{1, \ldots, N\}$ denote the set of nodes known to i at time t. Initially, $M_i(0) = \{i\}$. The discovery algorithm works as follows: In each consecutive time slot, nodes draw a channel uniformly at random. Then, they use an Aloha-like protocol and transmit a discovery beacon with probability p and listen with probability $(1 - p)$. When a node i decides to transmit at a time t, it transmits the complete $M_i(t)$. A node j that successfully decodes the discovery beacon sent by i at time t augments its database of known nodes, $M_j(t + 1) = M_j(t) \cup M_i(t)$.

Detailed description: In what follows we describe how our proposed algorithm works in detail:

(i) At any time slot t nodes select one of the k orthogonal channels $C = \{c_1, c_2, ..., c_k\}$ uniformly at random.

(ii) Nodes compute the optimal transmission probability $p^\star$ and attempt to broadcast their database with probability $p^\star$ and listen to other messages in the selected channel with probability $(1 - p^\star)$ To derive the optimal transmission probability we assume lossless links and the absence of collision detection mechanisms. Hence, nodes listening on channel c_m at time t will receive a message if and only if there is one single other node transmitting on c_m in slot t. In this case, the probability of a node transmitting and at least one neighbor receiving its signal is given by

$$p_s = \frac{p}{k}\left[\sum_{i=1}^{N-1}\binom{N-1}{i}\left(\frac{1-p}{k}\right)^i\left(1-\frac{1}{k}\right)^{N-i-1}\right] \quad (2a)$$

$$= \frac{p}{k}\left(\left(1-\frac{p}{k}\right)^{N-1}-\left(1-\frac{1}{k}\right)^{N-1}\right). \quad (2b)$$

Equation (2a) says that the probability of a successful transmission is given by the probability of one node transmitting on a channel and some nodes listening to this channels, while the rest of the nodes are not in that channel. Equation (2b) is the result of this summation and, in other words, it states that the probability of at least having a node listening on the transmission on that channel is given by the probability that no other node transmits on this channel (otherwise, we would have a collision) minus the probability that no other node is on this channel (at least one should be on the channel).

It can be easily verified that p^* corresponding to $\partial p_s/\partial p = 0$ maximizes p_s, *i.e.*, $p^* = \arg\max_p p_s$. Moreover, p^* should be $p^* \leq \frac{k}{N}$, whereas for $k = 1$, the inequality is satisfied with equality and the optimal transmission probability is identical to the one obtained by the coupon collector analysis, *i.e.*, $p^\star = 1/N$. For $k > 1$, $p^\star$ can be computed numerically.

(iii) A node i that enters transmission mode attempts to broadcast its complete database M_i. The transmission is successful if and only if node i is the only node that attempts a transmission in its selected channel in that time slot.

(iv) A node j in receive mode listens for incoming discovery beacons. If the successful beaconing was performed by node i, node j updates its database to also include the nodes known to i, *i.e.*, $M_j = M_j \cup M_i$. Note that the discovery can be either *direct* (if i was not already in M_j) or *indirect* (when some node $k \in M_i$ is not already in M_j).

B. Adaptive mode for unknown network size

So far, our simulations have been performed with the optimal transmit probability, which can only be computed if all nodes know the clique size N. However, in realistic scenarios, the size of the network to discover is not known a priori. In this case, we can consider a slightly modified algorithm, inspired by [1]: **(i)** the algorithm operation is divided into phases. Each phase $p = 1, 2, \ldots$ lasts for 2^{p+1} time slots; **(ii)** in each phase p, nodes compute and use the optimal transmission probability under the assumption that
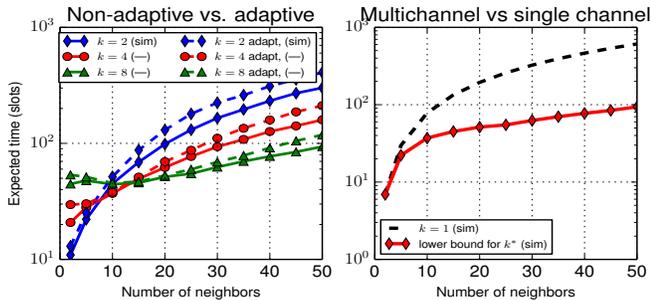
Fig. 2. Single channel vs multichannel epidemic discovery with k channels; (left) expected time to discover all neighbors in *non-adaptive* and *adaptive modes* in multichannel epidemic discovery for $k = \{2, 4, .., 8\}$. Observe that for $N = 50$ and $k = 8$, the speedup factor is $\approx \sqrt{N}$.(right) given the network size N we choose k (lower bound on k^*) that minimizes $E[T]$.

there are $N = 2^p$ nodes in the network.

C. Performance evaluation

We evaluate the multi-channel epidemic discovery protocol in extensive simulations to quantify the speed-ups over single-channel discovery and to understand how the optimal number of channels depends on the clique size N. We vary the number of nodes from 2 to 50. For each value of N and k, we run 1000 iterations and our simulations results are shown with 95% confidence intervals.

Figure 2 shows the average time it takes for all nodes in the clique to discover all other nodes as function of the clique size. The dashed line is the single-channel case, which agrees well with the $N \log(N)$ behavior predicted by the coupon collector analysis. The remaining lines show the discovery times for a varying number of channels. We note that single-channel discovery is only optimal for very small clique sizes, and that the improvement our multi-channel discovery protocol becomes increasingly significant as the clique size increases. For example, when $N = 50$, it is optimal to use $k = 8$ channels and the speedup over the single-channel protocol is around $\sqrt{N}$. Figure 2 also shows the expected discovery time of the adaptive protocol and the multi-channel epidemic discovery protocol assuming global knowledge of N. Note that the two protocols follow each other closely and the slowdown is smaller than 50%.

IV. EPIDEMIC DISCOVERY UNDER LOSSY LINKS

In this section, we will investigate the benefits of epidemic information dissemination in lossy networks. Our analysis will show that not only does epidemic dissemination shorten the discovery times, but it also makes the protocol more robust to link losses, even for the single channel case. In extreme situations with large networks and high loss probabilities, discovery times can be improved by an order of magnitude.

A. Single channel discovery without epidemics

The existence of lossy links possibly require more than one transmission from a node until its message is received successfully by all neighboring nodes. As a result, the expected time for neighbor discovery is increased.
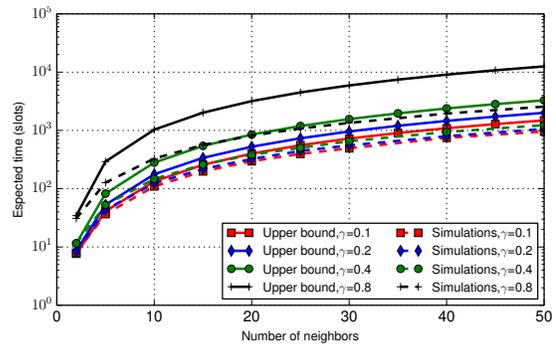

Fig. 3. Single channel discovery without epidemics under lossy links for $\gamma = \{0.1, 0.2, 0.4, 0.8\}$. Simulations considered 1000 network realizations for each network size. A receiving node will require at least one successful reception from each of its neighbors. Since the links fail randomly, this fact will increase the expected time to discover all its neighbors.

Let Y be a random variable that denotes the number of transmission attempts a transmitter makes to deliver a packet to all neighbors. Clearly, $Y = \max\{X_1, X_2, \ldots, X_{N-1}\}$, where X_i is the random variable that denotes the number of transmission attempts a transmitter requires to deliver a packet to receiver i. We assume that the condition of each link is independent of that of the others. Hence,

$$\mathbb{P}[Y \leq m] = \prod_{i=1}^{N-1} \mathbb{P}[X_i \leq m] = (1 - \gamma^m)^{N-1}. \quad (3)$$

Denote by $\mathbb{P}[Y = m]$, the probability that a packet needs exactly k transmissions to be delivered to all neighbors. Then, $\mathbb{P}[Y = m] = (1 - \gamma^m)^{N-1} - (1 - \gamma^{m-1})^{N-1}$. If all the links were reliable, all neighbors would have received the information in 1 slot [8]. Hence, the expected time $\epsilon_l \triangleq \mathbb{E}[Y]$ can be seen as the slowdown factor *for each node* due to lossy links, *i.e.*, how many extra slots are required for a node to disseminate its information to all its neighbors when epidemics are not available and a link fails with probability γ. Hence,

$$\mathbb{E}[Y] = \sum_{m=1}^{\infty} m\mathbb{P}[Y = m] \stackrel{(a)}{=} \sum_{\ell=1}^{N-1} (-1)^{\ell+1} \binom{N-1}{\ell} \frac{1}{1 - \gamma^\ell},$$

where (a) is the result emanating after algebraic manipulations. This result can also be found in the analysis for broadcast delay on erasure channels (see, for example, [9]).

The probability of transmission for each node is independent of the probability of loss on a link. As a result, the event T that all neighbors transmit at least once (with expected time $\mathbb{E}[T]$) is independent of the event Y discussed above. Hence, the expected time of the product of the two events is equal to the product of the expected times; *i.e.*, by letting $Z \triangleq TY$, then $\mathbb{E}[Z] = \mathbb{E}[T]\mathbb{E}[Y] = \epsilon_l(Ne(\ln N + \Theta(1)))$.

In Figure 3, it is shown that the expected time computed is an upper bound to the expected time. Simulations considered 1000 network realizations for each network size.

B. Single channel discovery with epidemics

We will now demonstrate how epidemic dissemination improves discovery performance in networks with lossy links. Intuitively, a failure to receive a message from a neighbor
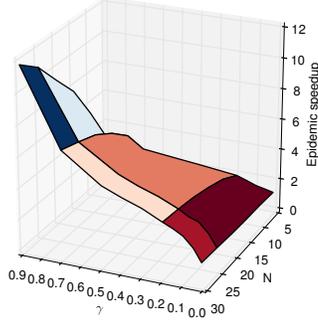
Fig. 4. When links are lossy, epidemic discovery gives significant speed-ups compared to the traditional approach. For large networks and $\gamma = 0.2$, we achieve a speed up factor of 4 in single channel operation.

does not have big impact when epidemic information is used, since the contents of the lost message can be recovered from other neighbors. This is illustrated in Figure 4, where the epidemic speed-up is defined as the ratio of the time taken to discover all the neighbors with and without epidemic information dissemination.

1) Analysis: We consider a clique network and first estimate the expected time needed for a node to convey its information to the rest of the nodes. We formulate the problem as a Markov Chain, as shown in Figure 5.
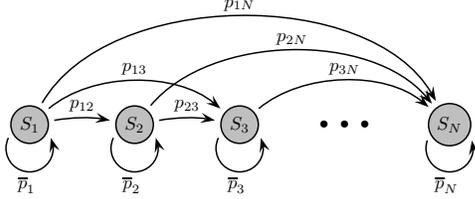

Fig. 5. Markov Chain depicting the evolution of the nodes that find out about a single neighbor.

The nodes that have the information to be spread are called *informed* nodes, whereas the rest are called *non-informed*. The probability of at least one node of the non-informed ones receiving the information from an informed node is given by $p_r^{(1)} = 1 - \gamma^{N-1}$. Staying at state S_1 requires that there was no successful transmission, or, there was a successful transmission but no other node could successfully receive the information. Hence, the probability of staying at state S_1, denoted by $\bar{p}_1$, is given by

$$\bar{p}_1 = 1 - p_s + p_s(1 - p_r^{(1)}) = 1 - p(1-p)^{N-1}(1 - \gamma^{N-1}).$$

The transition probability $p_{1,2}$ is the probability that the message is transmitted successfully by the only informed node and is received by a single node only; it is given by

$$p_{1,2} = p_s \binom{N-1}{1}(1-\gamma)\gamma^{N-2}.$$

Likewise, for $p_{1,k}$ we have

$$p_{1,k} = p_s \binom{N-1}{k-1}(1-\gamma)^{k-1}\gamma^{N-k}.$$

From any state ℓ to any other state k, where $k, \ell \in \mathbb{N}$ and

$k > \ell$, we have

$$p_{\ell,k} = \binom{\ell}{1}p_s \binom{N-\ell}{k-\ell}(1-\gamma)^{k-\ell}\gamma^{N-k}. \tag{4}$$

The probability of staying at state ℓ is the probability that transmission from any of the ℓ informed nodes fails, or, it succeeds but none of the non-informed nodes receives the message, *i.e.*,

$$\bar{p}_\ell = 1 - \binom{\ell}{1}p_s + \binom{\ell}{1}p_s(1 - p_r^{(\ell)}), \tag{5}$$

where $p_r^{(\ell)}$ is the probability that none of the non-informed nodes receives the message; it is given by $p_r^{(\ell)} = 1 - \gamma^{N-\ell}$. Hence, equation (5) is given by

$$\bar{p}_\ell = 1 - \binom{\ell}{1}p(1-p)^{N-1}\left(1 - \gamma^{N-\ell}\right). \tag{6}$$

The transition matrix M of the Markov Chain is column stochastic (note that $\bar{p}_N = 1$) and lower triangular. has the following form: As a result, the eigenvalues of matrix M are its diagonal entries $\bar{p}_i$, $i = 1, 2, \ldots, N$. The transition matrix can be arranged in the following canonical form

$$M = \left(\begin{array}{cccc|c} \bar{p}_1 & 0 & 0 & \ldots & 0 \\ p_{1,2} & \bar{p}_2 & 0 & \ldots & 0 \\ p_{1,3} & p_{2,3} & \bar{p}_3 & & \vdots \\ \vdots & \vdots & \ldots & \ddots & 0 \\ \hline p_{1,N} & p_{2,N} & p_{3,N} & \ldots & \bar{p}_N \end{array}\right). \tag{7}$$

Let Q denote the upper left lower triangular square matrix. Since this is an absorbing Markov Chain, the probability that the process will be absorbed is 1, *i.e.*, $\lim_{m\to\infty} Q^m = 0$. The expected time to go from each of the states of the Markov Chain to the absorbing state is given by $\mathbf{1}^T(I-Q)^{-1}$. Matrix Q is lower triangular and hence, $R \triangleq (I - Q)^{-1}$ is also lower triangular. Let T_j denote the time it takes to reach the absorbing state from state j. The expected time T_1 to reach the absorbing state from state 1 (*i.e.*, if initially one node is informed only) is therefore given by $\mathbb{E}[T_1] \triangleq \mathbf{1}^T R(\bullet, 1)$, where $R(\bullet, 1) \in \mathbb{R}^{N-1}$ is the first column of matrix R. In what follows, we find upper and lower bounds on the expected time needed for node discovery.

Lower bound. The expected time $\mathbb{E}[T_1]$ can be alternatively written as

$$\mathbb{E}[T_1] = \frac{1}{1 - \bar{p}_1} + \sum_{j=2}^{N-1}\left(p_{1,j}\mathbb{E}[T_2]\right).$$

In the case of only 2 nodes, then the expected time $\mathbb{E}[T_1]$ is given exactly by $(1 - \bar{p}_1)^{-1}$, but for more than 2 nodes, $(1 - \bar{p}_1)^{-1}$ is a lower bound to $\mathbb{E}[T_1]$, *i.e.*,

$$\mathbb{E}[T_1] \geq \frac{1}{1 - \bar{p}_1} = \frac{1}{(1 - \gamma^{N-1})}\frac{1}{p(1-p)^{N-1}}. \tag{8}$$

This lower bound on $\mathbb{E}[T_1]$ can be approximated by a Bernoulli process with success probability $q_s = \alpha p(1 - p)^{N-1}$, where $\alpha \triangleq 1 - \gamma^{N-1}$. By taking derivatives of q_s with respect to p, the optimal transmission probability remains $p^* = 1/N$. Hence, following the coupon collector paradigm,
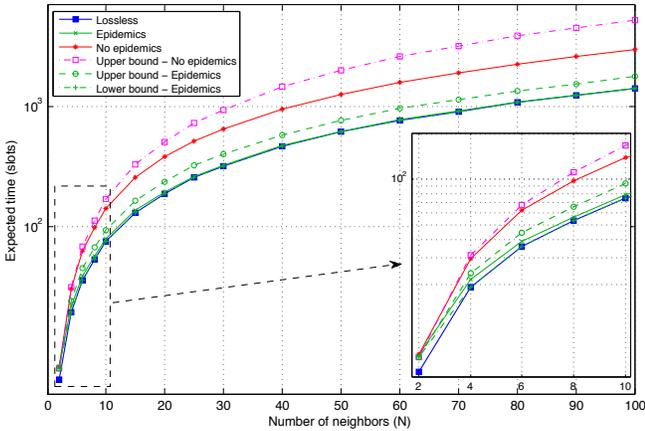
Fig. 6. Single channel epidemic and non-epidemic discovery under lossy links. The probability of a link failure, γ is chosen to be 0.2 for these simulations. Simulations considered 1000 network realizations for each network size.

a *lower bound to* the expected time, denoted by $\mathbb{E}[T_\ell]$, can be modeled as a geometric distribution and can be easily shown to be approximately $\mathbb{E}[T_\ell] = \frac{1}{\alpha} Ne\left(H_N + \Theta(1)\right)$.

Upper bound. The expected time $\mathbb{E}[T_1]$ can be written as

$$\mathbb{E}[T_1] = \mathbf{1}^T R(\bullet, 1) \overset{(a)}{\leq} \sum_{j=1}^{N-1} R(j,j) = \sum_{j=1}^{N-1} \frac{1}{1 - \bar{p}_j} \overset{(b)}{\leq} \frac{N-1}{1 - \bar{p}_{\ell^\dagger}},$$

where $\ell^\dagger \triangleq \arg\max_{\ell \in 1,2,\ldots,N-1} \binom{\ell}{1}\left(1 - p_L^{N-\ell}\right)$. Claim (a) stems from the fact that there is a higher probability of staying on a node longer if you start from that node than the expected time to be on that state if you start from another state, and (b) stems by the definition of $\ell^\dagger$. In both (a) and (b), equality holds if $N = 2$. Hence,

$$\mathbb{E}[T_1] \leq \frac{N-1}{1 - \bar{p}_{\ell^\dagger}} = \frac{N-1}{\ell^\dagger\left(1 - \gamma^{N-\ell^\dagger}\right) p(1-p)^{N-1}}.$$

Let $\beta \triangleq \frac{\ell^\dagger\left(1 - \gamma^{N-\ell^\dagger}\right)}{N-1}$, then $\mathbb{E}[T_1] \leq \frac{1}{\beta} \frac{1}{p(1-p)^{N-1}}$. The upper bound on $\mathbb{E}[T_1]$ can be approximated by a Bernoulli process with success probability $q_s = \beta p(1-p)^{N-1}$.

Again, by following the coupon collector paradigm, an *upper bound* to the expected time, denoted by $\mathbb{E}[T_u]$, can be modeled as a geometric distribution and hence, it can be easily shown to be approximately $\mathbb{E}[T_u] = \frac{1}{\beta} Ne\left(H_N + \Theta(1)\right)$. In case $\ell^\dagger = N - 1$ (when γ is small), then $\beta = 1 - \gamma$.

When $\gamma = 0$, then $\alpha = \beta = 1$ and hence $\mathbb{E}[T_u] = \mathbb{E}[T_\ell]$; as a result, the expected time of the coupon collector problem is recovered. Figure 6 compares the performance of neighbor discovery with and without epidemics and justifies the validity of the the theoretical results. We observe that as the number of nodes increases, the expected time of the epidemic node discovery recovers that of the lossless case, whereas in the case of no epidemics there is a deterioration in the performance due to losses.

C. Multichannel discovery with epidemics

We have already witnessed the superiority of multichannel neighbor discovery with epidemic information dissemination
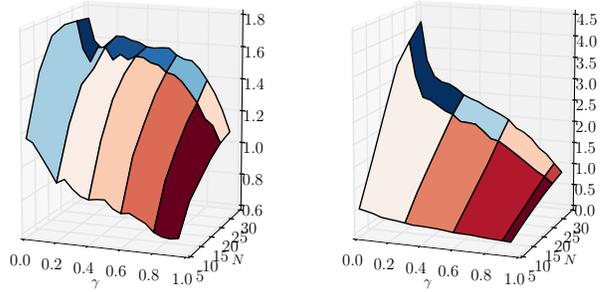

Fig. 7. For larger network sizes and moderate link failures, multichannel epidemic allows to speed up discovery; see the speedup for $k = 2$ (left) and $k = 8$ (right). For smaller networks and in scenarios of heavy link losses, a solution with single channel achieves best results.

in lossless networks. In addition, we have seen that in a single channel, discovery times can be improved even by an order of magnitude in dense networks due to epidemic information dissemination and approach (or achieve) the same performance as in the case for lossless networks. Now we want to investigate how the performance of our multichannel algorithm is affected when we have lossy networks.

For small values of γ (*e.g.*, $\gamma = 0.2$), not many links fail and the networks observed are not very sparse. As a result, multichannel communications is helpful. This is illustrated in Figure 7 where a speed-up is achieved for large networks; for example, for a network of 30 nodes and $\gamma = 0.2$ a speed-up of 1.6 is achieved. However, for larger values of γ and more channels, many link failures exist in channels with few active nodes and hence, the discovery becomes more difficult. It is hence desirable that many nodes stay on the same channel to avoid the situation of an unconnected network. This is again shown in Figure 7, where even for large networks (*e.g.*, $N = 30$) multichannel epidemic neighbor discovery with 8 channels performs worse than the single-channel operation.

What the results suggest is that multichannel epidemic neighbor discovery can be beneficial in dense networks, even when the network is lossy. If, however, there exist multiple channels and the loss probability is high, the performance degrades. This is the same phenomenon that occurs when we have many channels and only a few nodes and makes discovery more difficult.

A theoretical analysis of the expected time to discover all the neighbors in such a setup is of elevated difficulty and remains an open problem. More specifically, the network is switching at every step and the dissemination of information is dependent on the order by which events occur. If one attempts to model the problem as a Markov chain, the number of states grows exponentially with the size of the network and there is no apparent structure that allows further simplification or permits to bound the discovery time.

V. MULTICHANNEL EPIDEMIC DISCOVERY IN MULTIHOP NETWORKS

In our analysis and discussion so far it is assumed that the network is a single clique, where each node directly communicates and disseminates information to every other node in the network. This assumption makes the analysis

tractable and allows us to compare our theoretical result with the single-channel analysis that typically considers cliques. However, in practice, the network is often more complicated than a single clique. In this section, we extend our algorithm to types of networks other than the clique, such as multihop networks. More specifically, we consider the case for which a node is only interested to discover neighboring nodes that are r, $r = \{1, 2, \ldots\}$, hops away.

The proposed algorithm for neighbor discovery in multi-hop networks, which includes the clique scenario as a special case, is given in Algorithm 1.

Algorithm 1: Multichannel epidemic neighbor discovery.

Input: A set of channels $C = \{c_1, c_2, ..., c_k\}$, an estimate N of the number of nodes and a number of hops r required to be discovered.

Data: The set of neighbors each node i has in its database M_i and $M_i^{(r)}$ ($M_i^{(r)} \subseteq M_i$) the set of neighbors that are up to r hops away. At initialization $M_i = M_i^{(r)} = \{i\}$.

1 Based on N and k each node computes the optimal probability of transmission p^*.

2 Each node executes the following algorithm:

3 **for each** time slot $t \leftarrow 1$ **to** ∞ **do**

4   A channel c_m is selected randomly from the k channels; *i.e.*, $m \leftarrow \texttt{random}[\{1, 2, \ldots, k\}]$

5   Whether the node i is in transmission mode $\texttt{Tx}$ or reception mode $\texttt{Rx}$ on channel c_m, it is determined by a Slotted Aloha protocol; *i.e.*, $\texttt{mode} = \texttt{SlottedAloha}(p^*)$

6   **if** *mode*=$\texttt{Tx}$ **then**

7   Node i transmits a packet on channel c_m with *all* neighbors up to r hops away discovered so far, *i.e.*, $\texttt{transmit}(c_m, M_i^{(r)})$

8   **else**

9   **if** *mode*=$\texttt{Rx}$ **then**

10   Node i receives a packet from node j in the same channel c_m, *i.e.*, $M_j = \texttt{receive}(c_m)$

11   Node i updates its database M_i and $M_i^{(r)}$ with the new nodes discovered; *i.e.*, $M_i \leftarrow M_i \cup M_j$, $M_i^{(r)} \leftarrow M_i^{(r)} \cup M_j^{(r-1)}$

12   **else**

13   If node i has not received any packet then either a collision occurred, *or*, no node transmitted on channel c_m.

14   **end**

15   **end**

16 **end**

Result: Every node discovers all other nodes up to r hops away in the network.

VI. IMPLEMENTATION AND TESTBED VALIDATION

A. Implementation details

We implemented our proposed protocol on top of the Contiki OS [10], and we used the TelosB [11] platform. The

---

TABLE I

TESTBED, HARDWARE AND PROTOCOL SPECIFICATIONS

Description	Value
Testbed	Indriya [13]
Number of Nodes	100 (Indriya testbed) and 30 (clique testbed)
Number of hops	5 to 9 (Indriya)
Mote Platform	TelosB
Max. Trans. Power	0 (dBm)
Radio transceiver	TI CC2420, 2.4 GHz (ISM band), 250kbps
MCU	TI MSP430, 8MHz, 10KB RAM, 48KB flash
Experimental time	1 hour (testbed constraint)
Number of RF channels used	$\{1, 2, 4, 8\}$ selected from in $\in [11, 12, .., 26]$
Discovery msg size	2 bytes
Time slot duration	10 ms

Contiki OS is a lightweight open source operating system for wireless sensor nodes. Contiki is built around an event-driven kernel and provides an optional preemptive multithreading functionality that can be applied to individual processes. We also made extensive use of the COOJA simulation [12] for debugging the C code used in the experimental evaluation on real hardware. To allow tight control of synchronization primitives and the radio driver, the protocol was essentially implemented at the RDC (Radio Duty Cycle) layer, and we disabled the default features of the Contiki OS.The TelosB platform has a TI-MSP430 micro-controller clocked at 8MHz and equipped with 10KB of RAM and 48KB of internal flash memory. The TelosB motes use the TI CC2420 radio, which is compliant with the IEEE 802.15.4 standard and operates in the 2.4GHz ISM band. The IEEE 802.15.4 standard splits this frequency band into 16 channels (11 to 26). Table I summarizes the implementation details.

Time synchronization and time slot management: we used Contiki's real-time timer ($rtimer$), which offers micro-second resolution, to implement a basic time synchronization protocol (clique networks only). Nodes are scheduled to wake up at the slot boundaries, and transmissions are delayed with a small guard time relative to the time slot boundary to allow for channel switching and (limited) asynchronism.

Transmission probability: in *non-adaptive* mode, the specific channels (and hence the number k) available to the discovery protocol and the expected clique size N are provided at boot time. The protocol code includes a static table with the optimal transmission probability as a function of N and k, which allows motes to perform a simple lookup to determine $p^\star$. For the *adaptive* mode, only the number k is provided at boot time, while the estimator computes the number of nodes adaptively.

B. Testbed evaluation in clique and multihop networks

For clique networks, we used 30 TelosB motes in our laboratory. We varied the clique size from 2 to 30 nodes. For each case pair (N, k), we run 50 rounds to get a statistically relevant value of the average expected neighbor discovery time. For the experimental evaluation in multi-hop networks, we used the large-scale testbed Indriya [13].

1) Clique networks: Figure 8 (left) shows the testbed results and simulated discovery times for the *non-adaptive* protocol under different clique sizes and for $k = 2, 4$ and 8 channels. We also evaluated the adaptive mode that runs without knowledge of the network size. The deviation between real and simulated results are larger than for the non-adaptive case (Figure 8 (right)), but the results are good and
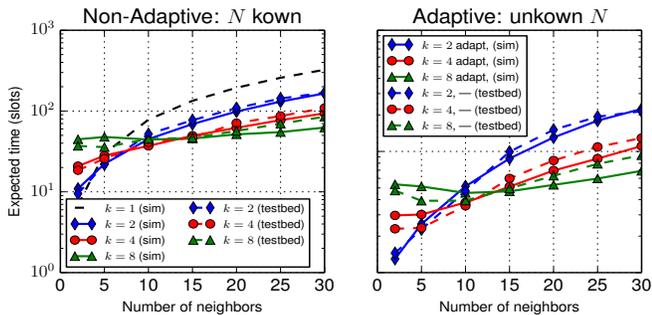
Fig. 8. Testbed evaluation in clique networks for *non-adaptive* mode (top) and *adaptive* mode (bottom): comparison between simulation and experimental results in multichannel epidemic discovery.
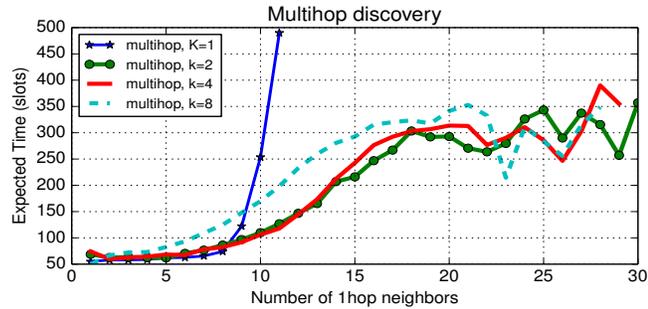

Fig. 9. Testbed evaluation in multihop networks for *adaptive* mode.

the protocol offers substantial benefits over single-channel discovery. The slow-down factor (the discovery time of the adaptive relative to the non-adaptive protocol) is smaller than 1.5 for all evaluated combinations of N and k. We note that the performance of the adaptive mode is particularly good for large networks and large number of channels ($N \geq 15$, and $k = 8$), with a slow-down of less than 10%.

2) Multihop networks: Figures 9 shows the testbed evaluation in multihop networks for *adaptive* mode, where the expected number of slots is improved considerably. The results suggest that the transmission probability should be adjusted accordingly, due to difficulty in achieving large scale synchronization. This preliminary investigation demonstrates the benefits of multi-channel coordination in multihop networks that is achieved via local coordination only, since it involves the degree of the nodes in the network rather than the size of the network itself. The benefits of the new approach to the neighbor discovery problem open a new avenue for research and analysis of multihop networks.

VII. DISCUSSION

A lot of work has been done to develop efficient neighbor discovery algorithms, predominantly in single-channel clique networks. We have combined two ideas key ideas to improve discovery times and resilience to losses: *multichannel communication* and *epidemic information dissemination*.

First, we proposed a theoretical three-phase protocol for a clique network of size N. This analytical approach serves as a way to assess the benefits of multichannel epidemic neighbor discovery and as a benchmark for evaluating the performance of our practical algorithm for epidemic information exchange in multichannel networks and possibly other algorithms that will be proposed in the future. Our

algorithm is shown via simulations and experimental results to outperform the neighbor discovery-time compared to the single channel model proposed in [1]. More specifically, we showed that a discovery-time speeds up on the order of $\sqrt{N}$.

We then investigated the performance of our algorithm in networks with link failures and showed that our algorithm improves the expected discovery-time. For the case of a single channel only, we were able to derive upper and lower bounds on the expected time of clique networks with lossy links. Simulations show that in this case, as the number of nodes increases, epidemics recover the performance of a single channel without any link failures. Overall, in dense networks with high loss probabilities, discovery times can be improved by an order of magnitude due to epidemic information dissemination.

Finally, we departed from clique networks and considered more general networks, since in practice, the network is often more complicated than a single clique. In this case, we restricted the dissemination of information to a limited number of hops. This restriction is performed by hop-counting filtering where each node propagates only information about nodes appearing at most r hops away. We demonstrated via simulations that the performance degradation (in terms of convergence time) is very small, while at the same time the algorithm with restricted information dissemination can be successfully applied to networks other than the clique.

REFERENCES

[1] S. Vasudevan, D. Towsley, D. Goeckel, and R. Khalili, "Neighbor discovery in wireless networks and the coupon collector's problem," in *Proc. of ACM Mobicom*, 2009, pp. 181–192.
[2] K. Pister and L. Doherty, "TSMP: Time synchronized mesh protocol," *IASTED Distributed Sensor Networks*, pp. 391–398, 2008.
[3] T. Watteyne, A. Mehta, and K. Pister, "Reliability through frequency diversity: why channel hopping makes sense," in *Proc. of ACM PE-WASUN*, 2009, pp. 116–123.
[4] P. T. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, and A.-M. Kermarrec, "Lightweight probabilistic broadcast," *ACM Transactions on Computer Systems*, vol. 21, no. 4, pp. 341–374, 2003.
[5] M. Chrobak, L. Gasieniec, and W. Rytter, "A randomized algorithm for gossiping in radio networks," *Computing and Combinatorics*, pp. 483–492, 2001.
[6] A. E. Clementi, A. Monti, and R. Silvestri, "Distributed broadcast in radio networks of unknown topology," *Theoretical Computer Science*, vol. 302, no. 1, pp. 337–364, 2003.
[7] Y. Wu, J. Stankovic, T. He, S. Lin *et al.*, "Realistic and efficient multi-channel communications in wireless sensor networks," in *Proc. of IEEE INFOCOM*, 2008.
[8] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis, "The β-factor: measuring wireless link burstiness," in *Proc. of ACM Sensys*, 2008, pp. 29–42.
[9] N. Xie and S. Weber, "Network coding broadcast delay on erasure channels," in *Proc. of IEEE ITA Workshop*, 2013, pp. 1–8.
[10] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *Proc. of IEEE LCN*, 2004, pp. 455–462.
[11] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Proc. of ACM/IEEE IPSN*, 2005, pp. 364–369.
[12] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with Cooja," in *Proc. of IEEE LCN*, 2006, pp. 641–648.
[13] M. Doddavenkatappa, M. C. Chan, and A. L. Ananda, "Indriya: A low-cost, 3d wireless sensor network testbed," in *Testbeds and Research Infrastructure. Development of Networks and Communities.* Springer, 2012, pp. 302–316.