# Welcome: Low Latency and Energy Efficient Neighbor Discovery for Mobile and IoT Devices

Mariem Harmassi*, Junaid Ahmed Khan*†, Yacine Ghamri-Doudane*, and Cyril Faucher*
*L3i Lab, University of La Rochelle, La Rochelle France
†University of Memphis, Memphis, TN, USA
mariem.harmassi@univ-lr.fr

***Abstract*—Energy efficient neighbor discovery for multiple mobile devices in each others proximity is a challenge along duty cycling where low power devices are inactive for a large fraction of time. Existing schemes allow each device to employ a schedule to become active and send periodic messages or listen to neighboring devices to ensure a neighbor discovery in a bounded delay. However, collisions can occur due to simultaneous transmission of messages from multiple devices resulting in failure of neighbor discovery. We propose to reduce the number of message transmissions in a neighbor discovery process to avoid collisions and in result enhance the number of devices discovered.**

**To do so, in this paper, we propose Welcome, a low latency and energy efficient neighbor discovery scheme. Instead of all nodes transmitting messages, only a single node can become a delegate to discover the nodes in vicinity and provide the neighborhood information to its neighbors. A node first finds its eligibility to become delegate based on its residual energy and association to the neighborhood. It then declares itself a delegate and listens to messages from its neighbors. Finally, it broadcasts the information regarding its neighbors to the devices in its communication range. Moreover, delegates can be rotated among neighbors where a node with high eligibility can content to become delegate. Welcome is compared with seven existing neighbor discovery schemes and it successfully discovers $100\%$ of neighbors with low energy consumption and low latency for a neighborhood size of upto $100$ nodes.**

***Index Terms*—Neighbor Discovery, Energy Efficiency, Wireless Sensor Networks, Internet of Things, Flock Discovery**

## I. Introduction

Internet of Things (IoT) applications for energy efficient data collection require low power sensors in a smart building/home to discover and communicate with nearby sensors and other wireless devices. Similarly, the increasing demand for proximity-based networking applications with the proliferation of smart mobile devices such as tablets, smart phones and smart wearables require wireless devices in each others proximity to interact locally as an ad-hoc network. For example, a tourist guide or a school teacher on excursion can ensure someone is not separated from the group using different proximity-based networking applications on their mobile devices.

The first step for devices to connect is to efficiently discover neighboring devices in their wireless communication range. However, for a low-power device to find all the similarly energy-constrained devices in the vicinity is a challenging task, particularly for battery-powered nodes opting for duty cycling. Duty cycling enables a device to operate between sleep and active state with the aim of conserving energy by staying in sleep state for a large fraction of time. It is also possible to have a clock drift between the active periods of different devices, requiring a careful consideration for a neighbor discovery scheme to be robust with respect to time synchronization between nodes. Another issue is the existence of heterogeneous duty cycles in a neighborhood, i.e. the fraction of time a node is active can be different from its neighboring nodes, thus requiring a neighbor discovery scheme to cope with such diversity.

Energy efficient synchronous and asynchronous, symmetric and asymmetric duty cycled neighbor discovery schemes exists in the literature with a focus on minimizing the worst case latency required for the mutual discovery of a pair of low duty cycled devices. Time is usually divided into regular slots, and each node becomes active on a limited number of slots as the schedule defined based on its respective duty cycle. Neighbor discovery is possible when two or more nodes are simultaneously active in the same slot, hence the objective is to find a maximum number of overlapping slots as an opportunity for mutual discovery between nodes.

However, still a fraction of nodes in a neighborhood fail to find each other using existing neighbor discovery schemes. Such failures in discovery are mainly due to collisions between messages transmitted simultaneously by multiple nodes in the same slot where such phenomenon is never considered previously in the literature.

To address this issue, in this paper, we propose "Welcome", a novel neighbor discovery scheme allowing a single node as a delegate, instead of all nodes in a proximity, to send beacons and listen to neighbors thereby reducing the overall amount of transmitted messages. This caters the issue of collisions between messages sent by concurrently transmitting nodes. We further allow nodes to rotate the role of becoming the delegate where each node computes its eligibility to become the delegate node based on its residual energy and association to the neighborhood. Moreover, Welcome enables nodes to auto-organize where eligible nodes content to become delegate in the absence of an existing delegate node.

We define two metrics to study the relation between *i)* the energy consumption and the fraction of neighbors discovered, and *ii)* the latency of discovery and the fraction of neighbors discovered. The purpose is to analyze Welcome with respect to the energy efficiency and latency trade-off, i.e. reducing the energy consumption by employing low duty cycles can lead to longer latency in discovering neighbors.

Welcome is evaluated theoretically as well as using simulations, analyzing its scalability to discover up to $100$ neighbors in each others communication range for nodes operating on low duty cycles of $1\%$ and $5\%$. Results show that Welcome is a a low latency and energy efficient neighbor discovery scheme allowing nodes to successfully discover $100\%$ of their neighbors unlike existing schemes which resulted in discovering a maximum of $90\%$ nodes with substantially high latency. The contributions of this paper are summarized as follows:

- We present Welcome, a novel neighbor discovery scheme with one delegate node discovering its neighbors and sharing the neighborhood information with the devices in proximity, while reducing excessive messages exchanged among devices.
- We allow nodes to auto-organize and rotate the role to become delegate in a neighborhood to ensure fair energy consumption.
- We define two novel metrics to evaluate the energy consumption and discovery latency with respect to the fraction of discovered neighbors theoretically as well as using extensive simulations.

The remainder of the paper is organized such that the Section II discusses an overview of existing neighbor discovery schemes in the literature identifying their drawback. Section III presents the design and description of the proposed Welcome scheme. In Section IV, we define the evaluation metrics along with a theoretical comparison of Welcome with the state of the art schemes. Section V discusses the performance evaluation and results based on extensive simulations. Finally, Section VI concludes the paper along some insights into future directions.

## II. Overview on Neighbor Discovery

Neighbor discovery schemes can be classified into two categories: *i)* direct, where nodes discover only the neighbors from which they directly receive a message [1] [2], and *ii)* indirect schemes, where nodes can learn the existence of neighbors indirectly from other nodes [3] [4] [5] [6] [7]. Direct schemes are further classified into *i)* quorum-based [8], *ii)* prime number-based [9] [10], *iii)* dynamic listen slot [11] [12], *iv)* fixed listen slot [13] [14], and *v)* stochastic [15]. These schemes can work on one or multiple frequency channels [16] [17] [18], and they all follow a similar principle of dividing time into slots and letting the node to be active in a slot based on a schedule defined by the respective algorithm.

Quorum-based schemes [8] guarantee that two nodes have at least one activity slot in common in a period of $N$ slots by being active in $\sqrt{N}$ slots. These mechanisms result in relatively high duty cycles and only function in homogeneous duty cycle conditions. The cyclic quorum design in heterogeneous duty cycle conditions is known as asymmetric design, and specific solutions were proposed to address this problem. Prime number-based asymmetric discovery schemes require a node to choose a single (e.g. U-Connect [10]) or a pair of prime numbers (e.g. Disco [9]) to derive its duty cycle. The activity slots of a node will be the multiples of the selected prime number(s). This approach can be extended, and differential codes can be built for each pair of nodes starting from relatively prime numbers [19]. Using results from number theory, it can be shown that any two nodes will finally wake-up on the same slot. The discovery latency in this case is the time slot corresponding to the product of the prime numbers used by the two nodes. The different strategies also take different approaches in the activity slots. Disco proposes to send two beacons in each activity slot, one at the beginning and one at the end, and listen for incoming beacons from potential neighbors in the rest of the slot. The slot of U-Connect comprises a single beacon, followed by a listen period.

However, the transmission and listen activities are independent and they can be conducted on different slots. In dynamic listen slot schemes, a large time period is divided into regular sized cycles, where each cycle is further composed of slots. Two types of slots exists, static transmission slots at fixed positions, either at the beginning or end of the cycle, and dynamic listen slots with a regular shift to either the left or right in consecutive cycles, up to the end of the period. Searchlight [12] is an example of such an approach, where a node has a static slot in the beginning of each cycle and an active slot shifted one slot to the right in each consecutive cycle. Similarly, Blinddate [11] uses one static slot in each cycle and two dynamic listen slots, one shifted to the right and one to the left in each consecutive cycle. A fixed schedule can also be used for listen slots. Nihao [14] takes the approach of *talk more listen less*, where more transmissions than listen slots exist in a given period. In the same context, Hello [13] is a highly parameterizable solution, where nodes listen more at the beginning of the period, and periodically wake up for transmissions. This scheme is shown to be a generalization of several other mechanisms, such as Disco, U-Connect and Searchlight. Finally, stochastic schemes such as Birthday [15] allow nodes to transmit beacons, listen for beacons from other nodes or sleep in a slot based on a probability distribution. Energy efficiency is ensured by choosing a lower probability for beacon transmission or for listening. Such schemes perform better on the average case compared with the deterministic approaches above, but they provide no bound on the worst case latency and they can lead to long tails in discovering the last fraction of nodes. The comparative analysis in [20] highlights the performance of direct schemes with respect to energy, latency and the neighbors discovered.

Indirect schemes include group based discovery [5] which exploits existing schemes, such as Disco, and adapts a cooperative approach where nodes broadcast neighborhood table in an active slot. Similarly, Acc [3] improves the discovery phase in Disco by allowing nodes to share information regarding already discovered neighbors in their beacons, to the next encountering node. However, such exchange will result in high overhead, both for the individual node and the network, due to the large size of messages containing neighborhood

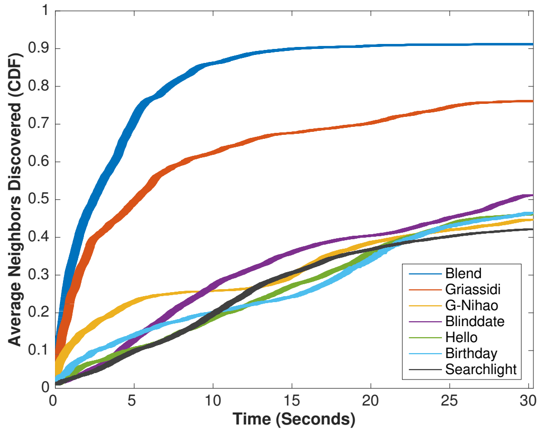



Figure 1: Comparison of neighbor discovery schemes

information, continuously exchanged between relatively low power nodes. EQS is [4] an extension to Quorum based system where nodes commonly active in a slot decide on a rendezvous slot for their next wake up and, thereby, share the information regarding their respective discovered neighbor in the rendezvous slot. However, clock drift, as well as collisions, can lead to a node not waking up in the respective rendezvous slot. Further rendezvous-based indirect schemes are Blend [6] and Griassdi [7] where nodes share their next listen period in the transmitted message and any node receiving it wakes up at the rendezvous time to send additional beacon to ensure bi-directional discovery.

Figure 1 compares the best among the above mentioned schemes where we depict the average neighbors discovered by each scheme with respect to time for a neighborhood size of 100 nodes in each others communication range. Clearly we observe that none of the existing schemes enables the nodes to discover all their neighbors. The recently proposed indirect schemes Blend and Griassdi performed well, although they suffered from the long tail effect. For example, Blend discovered $90\%$ of the neighbors in the first $10$ seconds though exhibits a long tail afterward.

We believe due to multiple nodes transmitting beacons simultaneously, a fraction of nodes fails to discover their neighbors resulting in collisions. The issue exacerbates with the increase in the amount of nodes in the neighborhoods. Therefore, there is a need for a neighbor discovery mechanism where one node discovers and share the neighborhood information to the nodes in its communication range. Additionally, there is a need to reduce the number of messages exchanged over the wireless medium as well as the nodes energy consumption during a neighbor discovery process.

## III. Welcome: Towards a Flock Discovery

### A. System Model

We consider a set of nodes $N = \{n\}$ in the proximity (communication range) of each other, forming a clique like network structure where each node is with degree $k_n = |N| - 1$. A node can be either in an active state where it can transmit a message/listen to incoming messages or in a sleep state and remains idle with minimum energy consumption. Thus, a node operating on low duty cycles alternates between sleep and active state where it stays in sleep state most of the time in order to save energy. It becomes active for a small amount of time $t_b$ to transmit a beacon, or during time $t_l$ to listen to incoming beacons from other devices, in a relatively larger time period $T$, where $t_b < t_l << T$. A node employs a given schedule to send beacons or listen to beacons from another nearby node in the same time period. The goal is to opportunistically find a time when two nodes are simultaneously active to ensure a successful discovery. The energy consumption $E_n$ of the node $n$ to be active as the combination of sending beacon or listening can be represented as $E_n = be_b + le_l$, where, $b$ represents the number of transmitted beacons, $e_b$ is the energy a node takes to transmit a beacon. Similarly, $l$ are the number of listen periods each with energy consumption $e_l$. The latency for the node $n$ to discover its neighbors is $L_n$, characterizing the worst case latency. We define two types of nodes, delegate and member nodes. Delegate nodes responsible for the neighbor discovery process, maintain and share the neighborhood information with nearby nodes. Member nodes are the nodes in a particular neighborhood receiving the information regarding their neighbors from delegate nodes.

We define the following basic message types: (i) *Welcome* message periodically broadcasted by the delegate node in a neighborhood containing neighborhood information and the schedule for the next neighbor discovery process. (ii) A *Discovery* message as the neighbor discovery initialization in case of absence of any delegate node in the neighborhood. (iii) A unicast *Hello* sent by any node to a delegate node in order to provide information regarding its existence in the neighborhood.

### B. Delegate Node Eligibility

Although multiple nodes can be delegates in a fail tolerant neighbor discovery, however, in an energy constraint IoT network the added redundancy increases the energy consumption. We believe it is sufficient to allow a single node to declare itself as a delegate node to represent the neighborhood in a distributed manner and welcome incoming neighbors. Nodes in a neighborhood can auto-organize and rotate the role of being a delegate among spatio-temporally co-located nodes to ensure a fair energy consumption of all nodes.

We present an eligibility function for a node to become a delegate for some period in a neighborhood based on its residual energy level. Besides, energy requirements several other factors can be considered based on the application requirements, though we consider such a node to be reliable in terms of its association to the neighborhood. We need to find for how long the node belongs to the same neighborhood in order to avoid malicious node new to the neighborhood with maximum residual energy level to declare itself as the delegate node and subsequently hijacks the neighborhood. Thus, the longer the node is in the same location (i.e part of the same

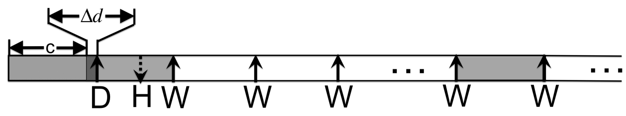



Figure 2: Discovery initialization - no delegate node detected

neighborhood), the more it is eligible to become the delegate node. Moreover, the node frequently becoming a delegate is more likely to exhaust its energy earlier compared to its neighbors. Therefore, such nodes should be less preferred to become delegates.

Each node can compute an eligibility function $f_D$ to measure its ability to become a delegate once it becomes active in a neighborhood. It can compute the following function prior to scanning for existing delegates in order to consider itself as the delegate in case of absence of any delegate in the neighborhood:

$$f_D = \alpha f_R + \beta f_E + \gamma f_H \tag{1}$$

where, $f_R$ is the node reliability function to be a delegate node taking into account its association (i.e. for how long it belongs) to a neighborhood. The function $f_E$ is the residual energy function characterizing the node physical properties where high node residual energy levels yield high eligibility to become a delegate node. Here $f_H$ is the function which considers the node history of being a delegate previously as well as the time staleness with respective to the previous time it was a delegate in the neighborhood where an exponential decay function can be used to represent such time staleness. Thus, the more frequently it is a delegate node since its deployment in the neighborhood, the less it is preferred to be a delegate in order to increase its lifetime.

The function $f_D$ indicates how eligible a node is to become a delegate, in case of absence of a delegate in the neighborhood i.e. no Welcome/Discovery message received for some time, it waits an amount of time inversely proportional to $f_D$ before declaring itself the delegate and sending its Welcome message. This not only impedes low residual energy nodes (lower $f_E$) to become delegates but also reduces the likelihood of a newly joined malicious node (low $f_R$) or a node frequently selected as delegates in the past (high $f_H$) to become delegate.

### C. Model Schedule

The schedule of Welcome can be classified into two categories, the schedule for the delegate node and the member nodes. We discuss below the schedule of the delegate node followed by the member node schedule.

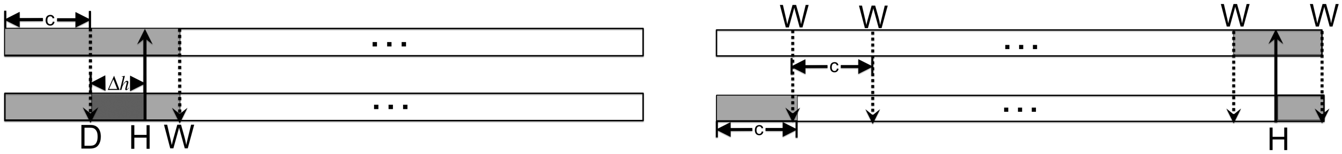
*1) Delegate Node:* Any node becoming active in a neighborhood computes its eligibility function $f_D$ upon wake-up and listen for the existence of a delegate node for a period $c$ as shown in Figure 2. In case it received no message from a delegate, it declares itself as the delegate for the neighborhood by broadcasting a Discovery message after a duration $\Delta d \leq c$ inversely proportional to its eligibility function $f_D$ with a maximum delay of $c$ period. The node still continues listening for the existence of possible delegate during $\Delta d$ and abort transmitting its Discovery message in case it receives a message from an existing delegate node. Subsequent to sending the Discovery message, the node listen for another $c$ duration in order to receive Hello messages from the neighbors that heard its Discovery message. It is to note that a shorter listen period $c$ can lead to collisions since the growth in the neighborhood size results in more Hello messages sent by neighbors. Therefore, $c$ should be flexible enough for a delegate node to accommodate the reception of Hello messages from all its neighbors while avoiding collisions.

At the end of its listen period, the node then broadcast a Welcome message comprising its node ID and the list of neighbors from which it received Hello messages during the listen period. Similarly it contains information regarding the next listen period scheduled by the delegate node based on its duty cycle. The node can then switch to sleep mode and periodically broadcast Welcome message where to ensure a neighboring node detects it, the interval between sending two Welcome messages is less than the defined listen period $c$. The idea is to allow the delegate node to switch itself to sleep mode between two Welcome messages and in result conserve its energy based on its desired duty cycle. Moreover, as shown in the Figure 2, the delegate node repeats its discovery phase (listen to Hello messages from neighbors) for the time allowed by its eligibility function and depending on its duty cycle. Algorithm 1 summarizes the neighbor discovery process for a delegate node.

*2) Member Node:* A member (non-delegate) node upon arriving in the neighborhood becomes active, computes its eligibility function using Equation 1 and listens for a message from a potential delegate node during $c$ period. It can receive either a Discovery message or a Welcome message from an existing delegate node as shown in Figure 3. Figure 3a shows the case when it receives a Discovery message indicating the delegate node already started listening for $c$ duration. The member node responds with a unicast Hello message to the delegate scheduled after a time delay $\Delta h$ randomly chosen between the time the member node receives the Discovery message and the time the delegate node finishes its listen period indicated as the duration $c$. Thus, the size of the interval $c$ is defined as the maximum time to defer a Hello message. The duration of $c$ for a given application should be long enough to accommodate potentially large neighborhood sizes allowing a delegate to be able to listen to Hello messages from a maximum amount of neighbors (member nodes).

The node than continues listening in order to receive the Welcome message containing information regarding all the neighbors detected by the delegate node. A member node can switch itself to sleep mode following the reception of the Welcome message and can schedule a wake-up based on its desired duty cycle for an upcoming discovery phase indicated in the delegate's Welcome message.

The member node upon receiving any of the Welcome messages sent by the delegate node can defer its transmission




(a) Discovery message

(b) Welcome message

Figure 3: Discovery with existing delegate node message detection

**Algorithm 1** Delegate Node

**for** node $n$ **do**
  Compute eligibility $f_d$ upon wake-up
  Listen for $c$ duration
  **if** No message received **then**
    Continue listening for additional $\Delta d$ period
    Broadcast Discovery message
    Listen for Hello messages for $c$ duration
    Broadcast Welcome message
    Sleep and wake-up to broadcast Welcome message after each $c$ period
    Periodically listen for $c$ duration each $c \times t$
  **end if**
**end for**

**Algorithm 2** Member Node

**for** node $n$ **do**
  Compute eligibility $f_d$ upon wake-up
  Listen for $c$ duration
  **if** Discovery message received **then**
    Send Hello with waiting time $\Delta h$
    Listen for Welcome message
  **else if** Welcome message received **then**
    Sleep and schedule a wake-up randomly during delegate's next listen period
    Send Hello message upon wake-up
    Listen for Welcome message
    Sleep
  **else**
    Declare as delegate node
  **end if**
**end for**

of Hello message to the delegate's upcoming discovery phase as shown in the Figure 3b. Thus, upon reception of a Welcome message, the member node can switch to sleep mode and schedule to send its Hello message at a time instant randomly chosen between the beginning and end of the upcoming listen period $c$ indicated in the delegate's Welcome message. It subsequently listens for the Welcome message following the delegate's discovery phase and can return to sleep mode based on its desired duty cycle. The Algorithm 2 shows the neighbor discovery process for a member node in the presence of a delegate node.

### D. Absence of Welcome message - Auto-organization

A node declaring itself as a delegate is responsible for sharing the neighborhood information for a finite duration allowed by its eligibility function, however, over the longer time period, the role of the delegate is rotated to ensure fairness. Delegate node with low residual energy levels can abdicate itself from being the delegate and subsequently switch to become member node and stay in sleep mode for longer period in order to conserve energy.

We consider the possibility for a delegate node to stop broadcasting messages due to its departure from the neighborhood or switching to member node and sleep for energy conservation purposes. Welcome comprises a built-in recovery mechanism allowing nearby nodes to auto-organize and collaboratively preserve neighborhood information in case of disappearance of messages from the existing delegate node for any reason.

The auto-organization process is as follows, Each member node receives and stores the neighborhood information from its delegate node. Once the delegate node stops broadcasting its Welcome message in the neighborhood. Similar to the initial delegate node declaration process, any member node can become the next delegate by broadcasting its respective Discovery message after a wait period $\Delta d$ inversely proportional to its eligibility function $f_D$ in Equation 1.

Since our eligibility function considers the nodes history as its association with respect to the neighborhood, only nodes already in the neighborhood for some time in the past will result in a shorter wait period. Moreover, since a node has already stored the neighborhood information from the previous delegate, it can preserve this neighborhood information along discovering any new nodes joining the neighborhood. Thus, the newly declared delegate node belongs to the same neighborhood while retaining the information regarding the previously known neighbors.

## IV. Analytical Evaluation

### A. Evaluation Metrics

The number of neighbors discovered differ between nodes and can be characterized by the average number of neighbors discovered in a neighborhood as defined below:

*Definition 1:* **Average Discovered Neighbors** The number of neighbors discovered by a node $n$ is defined as the cardinality of $D_n \subset N$, the set of neighbors discovered by $n$. Similarly, the average number of neighbors discovered for the set $N$ is the cardinality of the set $D_N$ represented as $D_N = \frac{1}{|N|} \sum_{n \in N} D_n$, where the unit of both $D_n$ and $D_N$ is a number of nodes between 0 and $|N| - 1$.

The worst case latency in discovering neighbors is an important metric, used by all the works in Section II. However, in the case where only a fraction of neighbors is discovered, we need to consider the joint relation between latency and average neighbor discovery.

*Definition 2:* **Latency vs. Discovery** The latency vs discovery relation for a neighbor discovery process considering a possibility of failure in discovering neighbors for a set of nodes $|N|$ is given as:

$$\theta_N = L_N \cdot \frac{|N| - 1}{D_N}, \tag{2}$$

where $L_N$ can be seen as the theoretical worst case latency, when ignoring collisions, for discovering the $|N| - 1$ neighbors, and $D_N$ is the average number of neighbors actually discovered. The term $\theta_N$ can be measured as a number of time slots with the assumption that the time is divided into slots of smaller granularity, and it has both a general performance meaning (i.e. the average number of time slots needed to discover a certain ratio of nodes) and a relative one (i.e. the average number of neighbors discovered in practice during a time period where all the nodes should have been theoretically discovered).

We also need to consider a measure on the energy efficiency, as the discovery process should be as less energy consuming as possible, while incurring low latency and providing a high discovery ratio. Therefore, we jointly characterize the energy consumption, the average number of neighbors discovered and the average latency, using the relations below.

*Definition 3:* **Energy vs. Discovery** We define, for a set of nodes $N$, the relation between the total fraction of neighbors discovered, the energy consumption and the latency needed for discovery as:

$$\delta_N = E_N \cdot \theta_N, \tag{3}$$

where $E_N$ is the average energy consumption during time period $L_N$. The metric $\delta_N$ is measured in Joule-second and it provides a common benchmark for the different neighbor discovery schemes.

### B. Theoretical Comparison

The key parameters can be derived as follows:

*Duty Cycle:* The listen period $c$ consumes most of the node energy and therefore can be defined for any desired duty cycle using the relation $DC = (c/2 + t)/ct$, where $t$ can be defined by an application as the number of times the listen period $c$ is to be repeated, reflecting the frequency of the re-initialization of the discovery process by the delegate node.

*Worst-case Latency:* For the delegate node, the worst case latency is $L_n = 2c + \Delta d$ since after waiting this amount of time, it broadcasts its Welcome message.

The worst case latency for the member node receiving a Discovery message is $L_n = c$ as it discovers its neighbors from the information in the Welcome message it receives following the Discovery message from the delegate node. The worst case latency for a member node receiving a Welcome message is the discovery cycle of the delegate node, i.e. $L_n = c \times t$. Thus, for a member node, the worst case latency of $L_n = c \times t$ is considered a maximum bound under Welcome protocol.

Table I: Welcome vs Existing Neighbor Discovery Schemes

| Scheme | Parameter(s) | Duty Cycle | Beacons | Listen periods | $L_n$ | $E_n$ |
|---|---|---|---|---|---|---|
| Welcome | $c, t$ | $(c/2+t)/ct$ | $c$ | $c$ | $ct$ | $(2c+\Delta d)e_l + 2ce_b$ |
| Blend [6] | $c$, $t$ | $2/c$ | $2c$ | $2c$ | $c^2/2$ | $2c(e_b + e_l)$ |
| Graissidi [7] | $c$, $t$ | $2/c$ | $2c$ | $2c$ | $c^2/2$ | $2c(e_b + e_l)$ |
| Disco [9] | $p_1$, $p_2$ | $1/p_1 + 1/p_2$ | $2(p_1 + p_2)$ | $p_1 + p_2$ | $p_1 p_2$ | $2(p_1 + p_2)e_b + (p_1 + p_2)e_l$ |
| U-Connect [10] | $p$ | $3/2p$ | $3p/2$ | $3p/2$ | $p^2$ | $3p(e_b + e_l)/2$ |
| Searchlight [12] | $c$, $t$ | $2/c$ | $2c$ | $c$ | $c^2/2$ | $c(2e_b + e_l)$ |
| Blinddate [11] | $c$, $t$, $s$ | $3/5s$ | $2c$ | $c$ | $5s^2/2$ | $c(2e_b + e_l)$ |
| Hello [13] | $c$, $t$ | $(c/2+t)/ct$ | $c$ | $c$ | $c^2/2$ | $c(2e_b + e_l)$ |
| G-Nihao [14] | $c$, $t$ | $2/c$ | $2c$ | $2c$ | $c^2/2$ | $c(e_b + e_l)$ |
| Birthday [15] | $p_b$, $p_l$, $p_s$ | $p_b + p_l$ | $p_b L_n$ | $p_l L_n$ | - | $L_n(p_b e_b + p_l e_l)$ |

*Energy Consumption:* The energy consumption of the delegate node is given as $E_n = (2c + \Delta d)e_l + 2ce_b$ where the number of beacons are $2c$ and the number of listen periods are $2c + \Delta d$. For the member node, the energy consumption is $E_n = ce_l + e_b$, where at worse, it listens during a duration $c$, sends a single Hello message at the delegate's next listen period.

Similar to Welcome, we theoretically derive the key parameters for each of the best start of the art schemes such that all nodes achieve the same desired duty cycle to ensure a fair comparison. Though such schemes can function under heterogeneous duty cycle settings and this step is only required for comparison purposes. Specifically, we are interested in the nodes activity schedule, their energy consumption on transmission and/or listen time slots, and the latency for successful neighbor discovery when operating on a particular duty cycle. Table I summarizes this comparison for seven different neighbor discovery scheme, representative of the approaches discussed in Section II. The first column in the table shows the key parameters used to define a node activity schedule in order to attain a given duty cycle, using the relation shown in the second column. The subsequent columns show the number of beacons, listen periods, worst case latency bound and energy consumption respectively for each scheme.

## V. Simulation-based Evaluation

### A. Simulation Scenario

Simulations are performed by implementing Welcome along state of the art discussed neighbor discovery mechanisms in the NS-3 simulator. A neighborhood is formed by placing a set of IEEE 802.11b/g/n enabled mobile nodes in the communication range (around 150 m) of each other in an ad-hoc network, resulting in a clique type network. We assume nodes stay in the neighborhood for some time. i.e. the topology does not evolve during our analysis however the arrival or departure of mobile nodes can be taken into account by repeating the discovery process after regular intervals. We evaluate each mechanism by considering up to 100 nodes, where the Friis propagation loss model is used to study the impact of fading in the wireless medium. Since energy efficiency can be achieved by allowing a node to operate on low duty cycles, we consider low duty cycles of $1\%$ and $5\%$. We vary the number of co-located nodes from 2 nodes up to a total of 100 nodes simultaneously present in a neighborhood. This encompass not only few mobile nodes discovering each other for applications such as mobile sensing or proximity-based gaming but also allow large number of energy constrained sensors on shipment packages to discover each other autonomously for tracking purposes.

Each member node sends a Hello message of 100 bytes while a delegate node sends a Discovery message and Welcome message of 100 bytes and $1KB$ respectively. Similarly, the node can listen during a duration of $c = 200$ ms large enough to accommodate Hello messages from up to 100 nodes. The parameters $\alpha$, $\beta$ and $\gamma$ in Equation 1 are set to $0.33$ to maintain generality for each function. We consider the possibility of random clock drifts between nodes as an asynchronous discovery since they are unaware of the time lag between each others active periods. Welcome is compared to indirect schemes *i)* Blend and *ii)* Garissidi, *iii)* fixed-slot based Hello, and *iv)* G-Nihao, *v)* dynamic slot based Searchlight and *vi)* Blinddate, and *vii)* stochastic Birthday mechanism. Each node follows an activity schedule using the respective parameters defined by each mechanism in Table I. Since each of the deterministic mechanisms discussed above theoretically ensures a successful discovery if the neighboring nodes are active during $L_n$ period, the simulation duration ensures that each node experiences at least $L_n$ period in common with all its neighbors.

We recall that each of the above mentioned mechanisms tries finding an overlapping active time period between nodes in order to ensure a successful discovery. However, in practice, activating multiple nodes at the same time can lead to collisions, thus resulting in discovery failures. We cater the issue by implementing a CSMA/CA based back-off approach where a node finding the medium busy before transmitting a beacon chooses a wait time randomly between its initial transmission time and a slot size of 10 ms.

We use our proposed evaluation metrics: *i)* The average number of discovered neighbors $D_N$ among $|N|$ nodes in each others communication range. *ii)* Latency vs discovery relation in Equation 2 to find the discovery latency incurred by the nodes when applying the schedule defined by each scheme, and *iii)* Energy vs discovery ($\delta_N$), defined in Equation 3 to find the energy consumption of the node using Welcome and other neighbor discovery schemes. Since each mechanism incurs different worst case discovery bounds, we need some common time bounds in order to fairly compare Welcome using different evaluation metrics.

### B. Simulation Results

We performed simulations on nodes running both on $1\%$ and $5\%$ duty cycles. However for brevity, we present below only the results from simulations using a $5\%$ duty cycle since nodes are more prone to collisions. The results are obtained using 10 simulation runs, where the average values of the results are shown with $95\%$ confidence intervals.

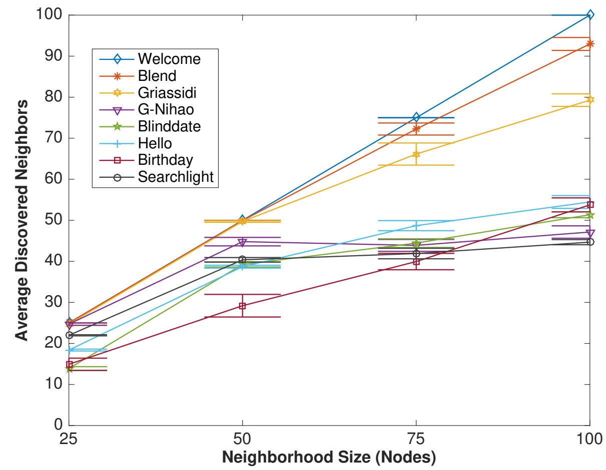



Figure 4: Average discovered neighbors comparison

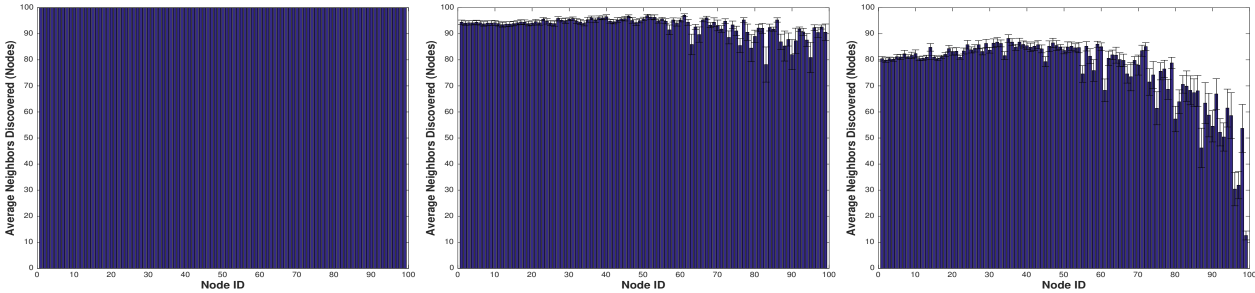
*1) Average discovered neighbors:* The motivation for proposing Welcome is the possibility of collision between beacons transmitting at the same instant, thus resulting in all the neighbors not necessarily discovered by the node. We investigate such behavior by finding the average number of neighbors discovered in different neighborhood sizes. Figure 4 compare the average discovered neighbors using Welcome along different state of the art schemes. It clearly shows that Welcome yields the highest number of discovered nodes, followed by the indirect schemes Blend and Griassdi. On the other hand, direct neighbor discovery schemes resulted in the least number of neighbors discovered on average.

Welcome discovers $100\%$ of the neighbors irrespective of the neighborhood size, and therefore, outperforms other schemes where Blend discovers around $90\%$ of nodes. One possible reason for Welcome high performance is the fact that it avoids collisions at the active overlapping time for discovery allowing a single node to listen to Hello messages from neighbors, which significantly increases the chances to discover neighbors.

We found that among all the compared schemes, Searchlight resulted in the poor performance because the low listen periods reduces its chance of discovering a large fraction of nodes as we can observe that less than $50\%$ of nodes are discovered for the neighborhood size of 100 nodes.

Furthermore, we analyzed the scalability of Welcome by comparing the neighbor discovered by individual nodes in the large neighborhood size of 100 nodes. Figure 5 shows the histogram of the neighbor discovered by each node compared with the better performing indirect schemes Blend and Griassdi. It clearly shows that Welcome in Figure 5a allows all nodes to discover their neighbors where the neighborhood information is shared by the delegate node. On the other hand, for Blend and Griassdi in Figures 5b and 5c, a fraction of nodes failed to discover all the neighbors in the neighborhood size of 100.

Additionally to observe the fraction of neighbors discovered over time, we compute the CDF of the neighbors discovered




(a) Welcome (b) Blend (c) Griassdi

Figure 5: Histogram of average discovered neighbors

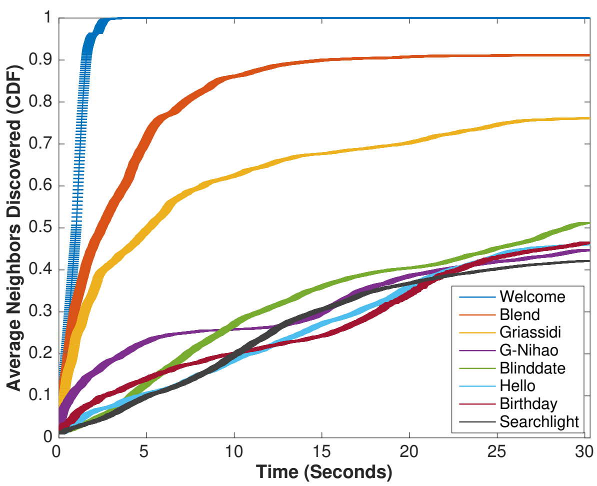



Figure 6: CDF comparison of average discovered neighbors

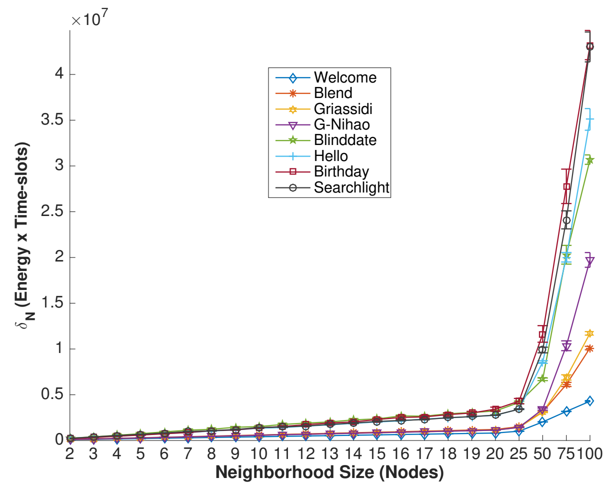



Figure 7: Latency vs discovery

using each scheme for the neighborhood size of 100 nodes. We observe such phenomena in Figure 6 where we compare Welcome with the state of the art schemes shown earlier in the Figure 1 for up to 30 seconds. Welcome discovered all the 100 neighbors with the delay of less than 5 seconds compared to existing neighbor discovery schemes. It is followed by indirect schemes Blend and Griassdi where the direct schemes yielded around similar performance by discovering around 50 neighbors at the delay of around 30 seconds.

Overall the results for average discovered neighbors validate our claims that discovery failures can occur largely due to the collisions of messages transmitted by multiple nodes in a neighborhood. Welcome caters such issues by allowing one delegate node discover neighbors and share this information in a distributed manner. Thus, it results in a low latency, scalable and energy efficient discovery o up to $100\%$ of neighbors for low duty cycle nodes.

*2) Latency vs Discovery:* We analyze the latency achieved by the nodes using Welcome when discovering neighbors compared to other state of the art schemes . Figure 7 shows the latency vs discovery metric for each scheme, running on nodes with $5\%$ duty cycle. A lower value of the latency vs discovery metric reflects a better performance in terms of latency. We notice that Welcome results in the best performance, i.e. quick discovery of a high fraction of neighbors in both the small-scale and large-scale neighborhood size, and thereby validating the scalability of Welcome compared to both, existing direct and indirect neighbor discovery schemes.

For the state of the art schemes, an increase in latency vs discovery is observed with the increase in neighborhood size, particularly when we compare for large scale neighborhoods. This is because, with the increase in the number of nodes, the chances of collisions between nodes beacons increases when multiple nodes try to transmit simultaneously. Welcome overcomes this issue by allowing a single node transmitting in the neighborhood at a time and therefore avoiding collisions between messages.

Overall, the comparison of Welcome with state of the art schemes for the latency vs discovery metric suggests that Welcome can successfully discover large fraction of neighbors with the least latency. We can also infer that Welcome scales better and is relatively stable with respect to the increase in the number of neighbors as well as it remains unaffected by the increase or decrease in the neighborhood size.

*3) Energy vs Discovery:* We finally analyze Welcome for the energy vs discovery metric where a low energy vs discov-

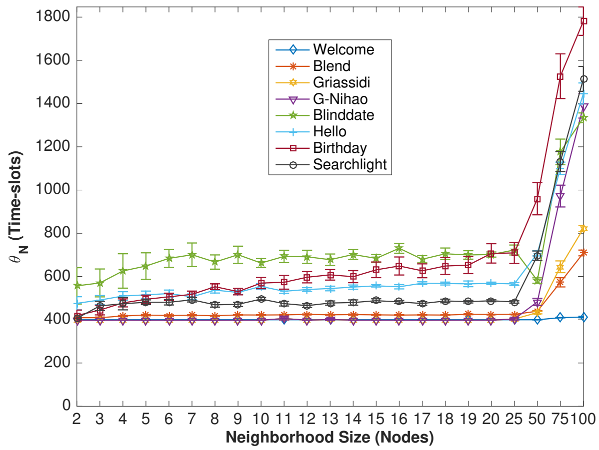



Figure 8: Energy vs discovery

ery value for a scheme correspond to a better performance. Figure 8 shows such comparison with respect to this metric when using each neighbor discovery scheme for nodes operating on $5\%$ duty cycles. It can be clearly shown that Welcome outperforms all existing neighbor discovery schemes with the least energy consumption for the fraction of discovered neighbors in each neighborhood. We also observe that indirect schemes Blend and Griassdi performed better compared to the direct schemes as nodes assist each other to enhance the neighbor discovery process. Still such schemes are consuming significantly high energy compared to our proposed Welcome scheme. Direct neighbor discovery schemes on the other hand resulted in a the poor performance with the probabilistic Birthday consuming the maximum energy.

Thus, the overall energy vs discovery analysis using simulations suggests that Welcome is an efficient and scalable neighbor discovery scheme with relatively less energy consumption and maximum amount of discovered neighbors.

## VI. Conclusions and Future Directions

Neighbor discovery in IoT suffer from large number of collision between messages transmitted by multiple nodes as the neighborhood size grows. To cater the issue, we propose a solution where one node can fully discover its neighbors and inform near-by nodes regarding their mutual neighbors. Therefore, in this paper, we propose Welcome, a new neighbor discovery scheme where one node can declare itself as a delegate for the neighbor discovery process. Welcome allows only nodes eligible with respect to their residual energy and neighborhood association to become delegates. Welcome is evaluated using simulations for neighborhood size of upto 100 nodes and it not only discovered $100\%$ neighbors but also results in low energy consumption, latency and collisions.

Possible extensions are to study the relation between the neighborhood size and the amount of time a delegate listens for messages from neighbors. This will result in further reduction in the delay of discovering neighbors as well as the amount of idle listening period for a delegate node.

## References


[1] L. Chen and K. Bian, "Neighbor discovery in mobile sensing applications: A comprehensive survey," *Ad Hoc Networks*, vol. 48, pp. 38–52, 2016.

[2] R. C. Carrano, D. Passos, L. C. Magalhães, and C. V. Albuquerque, "A comprehensive analysis on the use of schedule-based asynchronous duty cycling in wireless sensor networks," *Ad Hoc Networks*, vol. 16, pp. 142–164, 2014.

[3] D. Zhang, T. He, Y. Liu, Y. Gu, F. Ye, R. K. Ganti, and H. Lei, "Acc: generic on-demand accelerations for neighbor discovery in mobile applications," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 2012, pp. 169–182.

[4] D. Zhang, T. He, F. Ye, R. Ganti, and H. Lei, "Neighbor discovery and rendezvous maintenance with extended quorum systems for mobile applications," *IEEE Transactions on Mobile Computing*, 2016.

[5] L. Chen, Y. Shu, Y. Gu, S. Guo, T. He, F. Zhang, and J. Chen, "Group-based neighbor discovery in low-duty-cycle mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 1996–2009, 2016.

[6] C. Julien, C. Liu, A. L. Murphy, and G. P. Picco, "Blend: practical continuous neighbor discovery for bluetooth low energy." in *IPSN*, 2017, pp. 105–116.

[7] P. Kindt, D. Yunge, G. Reinerth, and S. Chakraborty, "Griassdi: Mutually assisted slotless neighbor discovery protocols," in *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2017.

[8] S. Lai, B. Ravindran, and H. Cho, "Heterogenous quorum-based wake-up scheduling in wireless sensor networks," *IEEE Transactions on Computers*, vol. 59, no. 11, pp. 1562–1575, 2010.

[9] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 2008, pp. 71–84.

[10] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar, "U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol," in *Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks*. ACM, 2010, pp. 350–361.

[11] K. Wang, X. Mao, and Y. Liu, "Blinddate: A neighbor discovery protocol," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 949–959, 2015.

[12] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: Won't you be my neighbor?" in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 185–196.

[13] W. Sun, Z. Yang, K. Wang, and Y. Liu, "Hello: A generic flexible protocol for neighbor discovery," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 540–548.

[14] Y. Qiu, S. Li, X. Xu, and Z. Li, "Talk more listen less: Energy-efficient neighbor discovery in wireless sensor networks," in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*. IEEE, 2016, pp. 1–9.

[15] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*. ACM, 2001, pp. 137–145.

[16] L. Chen, K. Bian, and M. Zheng, "Never live without neighbors: From single-to multi-channel neighbor discovery for mobile sensing applications," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 3148–3161, 2016.

[17] T. Meng, F. Wu, A. Li, G. Chen, and N. H. Vaidya, "On robust neighbor discovery in mobile wireless networks," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*. ACM, 2015, p. 38.

[18] T. Huang, H. Chen, Y. Zhang, and L. Cui, "Easind: Effective neighbor discovery algorithms for asynchronous and asymmetric-duty-cycle multi-channel mobile wsns," *Wireless Personal Communications*, vol. 84, no. 4, pp. 3031–3055, 2015.

[19] T. Meng, F. Wu, and G. Chen, "On designing neighbor discovery protocols: A code-based approach," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 1689–1697.

[20] J. A. Khan, R. Pujol, R. Stanica, and F. Valois, "On the energy efficiency and performance of neighbor discovery schemes for low duty cycle iot devices," in *ACM 14th PE-WASUN'17*, 2017.