The Spinning problem

Bogdan Munteanu, Richard Borie
Department of Computer Science
University of Alabama
Tuscaloosa, AL

Grzegorz Malewicz*
Department of Engineering
Google Inc.
Mountain View, CA

***Abstract*— Neighbor discovery in wireless networks with directional antennas is of crucial importance to many applications. In this paper we propose a variation of the classic neighbor discovery problem which we named the Spinning Problem. Here we are given an arbitrary number of devices on a plane. Each antenna starts spinning at a given rate and transmitting its location. The initial location and orientation are unknown. The goal is to find the rates that minimize the time for each device to find the location of every other device. We analyze a few particular cases of the problem. Specifically, we describe a polynomial time algorithm for 2 devices, and an exponential algorithm for n devices. It remains unknown whether there exists a polynomial time algorithm for an arbitrary number of antennas.**

I. Introduction

Static wireless ad-hoc networks and sensor networks have received an increased interest in the past years, especially due to their applicability. Field operations, rescue operations, habitat monitoring and surveillance are just a few of the numerous applications ad-hoc and sensor networks can be used for. In most applications, after deployment, nodes must first independently discover their neighbors. After the localization phase, these nodes can start communicating among themselves or perform whatever task they were deployed for. The problem of discovering the location of other nodes is known as localization problem or neighbor discovery and it is a very important first step in the establishment of a wireless network. Of course, neighbor discovery should be fast and energy efficient, in order to allow subsequent actions to take place in the network.

Omni-directional antennas and directional antennas may seem closely related, but in practice they are quite different. We focus on directional antennas rather than omni-directional antennas because the former have a stronger signal, a greater range, increased performance and reduced interference from unwanted sources.

We want to place an arbitrary number of battery-powered devices (nodes) on a plane. For example we want to drop these devices from an airplane over a field. We do not have control over how the devices get placed, so we assume their location is arbitrary and unknown. The nodes are equipped with directional antennas. All antennas have the same beamwidth, transmission power, frequency channel and modulation technique. Once deployed, the antennas will start spinning at a predefined speed transmitting and receiving signals at the same time. We do not know and we cannot set the initial orientation of the antennas, since the devise are dropped and not carefully deployed. The goal is for each device to determine the location of every other device so that later they can communicate.

We assume that all devices are in each other's range. Every device has a unique ID. Once a node receives a successful transmission from a neighbor, it will record the identity and the location of that node. This can be done by using Angle-Of-Arrival information of the received signal, or by including direction information in the sent packet.

Before dropping the devices, we can set the rotation speed of the antennas. We want to minimize the energy consumption of the devices but also minimize the time until every device discovers all his neighbors. If the beamwidth is wider and the antennas rotate faster then the energy consumption is higher but also the meeting time may be lower.

Our first question is: Is it possible to set the speeds such that the devices will discover one another? If yes, how can we optimize the total meeting time? Since in real life the speed is quite limited (by current technology), can we find a solution in the case we have an upper bound for the speed we may set? In the current paper we answer some of these questions.

Substantial work has been done in the area of discovery problems with directional antennas. In this section we present some of the work which is closest to the Spinning Problem.

In [1] the authors present several probabilistic algorithms for neighbor discovery in wireless networks. These algorithms are classified in two groups, Direct-Discovery Algorithms in which nodes discover their neighbors only upon receiving a transmission from them and Gossip-Based Algorithms in which nodes gossip about their neighbor's location information to enable faster discovery. Time is divided in time slots and in every time slot each node transmits in a random direction. The authors' goal is to maximize the probability of a node discovering its neighbors within a given amount of time.

In [2, 3] a distributed algorithm for creating a multihop wireless network with a higher lifetime is presented. The lifetime depends on the battery power of the network and on the power consumption for communication. After creating the network using this algorithm, the power consumption will be close to the optimal. The basic idea of the algorithm is that a node u transmits with minimum power p required to ensure that in every cone of degree α around u, there is some node that u

* The research of G. Malewicz was supported in part by NSF Grant ITR-800864

WIOPT 2008, 1st – 3rd Apr 2008, Berlin, Germany.
 ISBN 978-963-9799-18-9
DOI 10.4108/ICST.WIOPT2008.3216

can reach with power p. In these papers, energy consumption depends on the range of transmission. The authors are not concerned with the time required to build the network but with the power consumption for the communication when the network is in place.

The idea in [4] is to equip only a small fraction of the nodes of the network with location determination hardware. These nodes, called "anchor nodes", will act as reference points for location information. The rest of the nodes, called "target nodes", can use the information from the anchor nodes to estimate their location. The sensor nodes, which are equipped with four directional antennas, will determine their own location by measuring the distance from each anchor node.

The current paper uses new assumptions about the model which result in a totally different approach and we focus on minimizing the time until all nodes have discovered their neighbors.

The assumptions and the model of our problem are presented next. In section 3 we present some particular cases which we find quite interesting. Section 4 consists of the formal model and analysis of the problem, leaving Section 5 and 6 for the algorithm and conclusions.

II. MODEL

Next, we present the model for the problem which we named the "Spinning Problem".

1. We have a set of n static nodes arbitrary located on a plane and equipped with directional antennas;
2. Every node has a unique ID from 1 to n.
3. Each antenna has an unknown starting orientation (starting positioning angle);
4. All antennas have the same beamwidth α, $0 \leq \alpha \leq 2\pi$;
5. Two nodes meet (i.e. discover each other) if both emit in the other's direction simultaneously;
6. a) Every node is equipped with a device capable of providing AOA (Angle of Arrival);

   OR

   b) The direction information is included in the transmitted signal;
7. All nodes are in each other's range (the graph forms a clique);
8. Every antenna j, $j=1..n$, rotates clockwise at speed v_j, $Z \leq v_1 \leq v_2 \leq \ldots \leq v_n \leq V$, where v_j is expressed in the number of rotations per unit of time and Z, V are constants;
9. All antennas have the same transmission power, frequency channel and modulation technique.

We want to find a set $\{v_1, \ldots, v_n\}$, such that all n nodes will meet (i.e. each node will discover every other node).

Definition 1: We call a *solution to the spinning problem with given n, α and V*, a pair $(\{v_1, \ldots, v_n\}, t)$ where $\{v_1, \ldots, v_n\}$ is a set of speeds such that the n nodes will meet in at most t time, for any distribution of the nodes on the plane and any starting positioning of the antennas.

Definition 2: An *optimal solution to the spinning problem with given n and α,* is a pair $(\{v_1, \ldots, v_n\}, t)$ such that there exists no other solution $(\{v'_1, \ldots, v'_n\}, t')$ such that $t' < t$.

We assume that the system of coordinates is defined prior to the deployment.

Definition 3: The *starting angle* of an antenna is the angle formed by the beam and the Ox axis in the initial deployment of the device.

Problem: How to find the optimal solution to the spinning problem? In case the problem is proven to be NP-complete, how to find a solution that is close to the optimal solution?

We can see that if $\alpha \geq \pi$ then any 2 beams meet during a full spin of the slower beam. And that no matter what starting positions and what speeds they have (even if they have the same speed). The optimal solution in this case would be to set maximum speed (V) for all beams such that the full spin occurs as fast as possible. Hence, this case does not interest us. From now on we assume that $\alpha < \pi$.

Also, it is not possible that $V = v_1$ because that would mean all speeds must be equal, so for $\alpha < \pi$ it is possible to place the antennas such that the beams never meet (consider the case of 2 antennas rotating at the same speed).

Why is the problem interesting? Naturally, we assume that spinning requires energy and the energy consumption rises proportionally with the speed. Hence, if we spin longer or faster the energy consumption grows. Also, small α is better because we consume less energy when sending signals and we have lower chances of interference.

We want to minimize the energy consumption by finding a trade-off between the beamwidth, speed and meeting time. This leads to the upper and lower bounds Z and V on speed. If Z or V are exceeded then the energy consumption would be too high and not worth considering. Now we want to minimize the meeting time for antennas rotating at speeds between Z and V. Even more, we want to pick the speeds such that the antennas will eventually meet.

III. PARTICULAR CASES

If we do not choose the speeds carefully some antennas might never meet. For example, consider the case $n = 2$, $\alpha < \pi/2$ and a starting position like in the following figure where the second beam is very close to the meeting point:

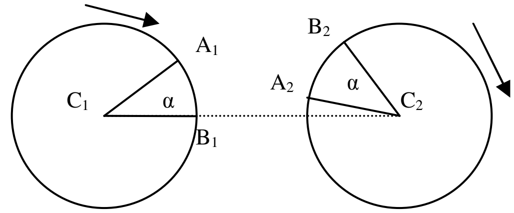

Figure 1.The two beams rotating clockwise are represented as sections of a circle

For ease of representation, we consider C_1 and C_2 to be the circles described by the rotations. The sections (A_1, B_1) and (A_2, B_2) of the circles represent the beams.

Consider the segment C_1C_2 which connects the locations of the 2 beams. We say that beam 1 *is available for meeting* if points A_1 and B_1 are on opposite sides of the segment C_1C_2. In the Fig. 1, beam 1 is available, but beam 2 is not.

Next, we prove a trivial claim in order to introduce the reader to our notations.

Claim 4: It is possible that the 2 beams will never meet.

Proof:

Let us pick $v_2 = 2v_1$ and $\alpha = \pi/6$. By the time beam 1 ends being available (so it rotates α distance), beam 2 rotates 2α. No meeting occurs in this time.

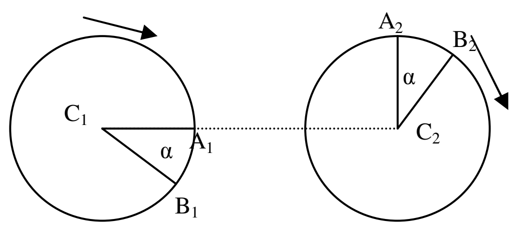

Figure 2. The two beams after beam 1 ends being available

After $t_{1/2}$ time (where t_1 is the time it takes the first beam to make a full rotation) we have a placement where beam 2 has passed over the meeting point and beam one has moved half the distance:

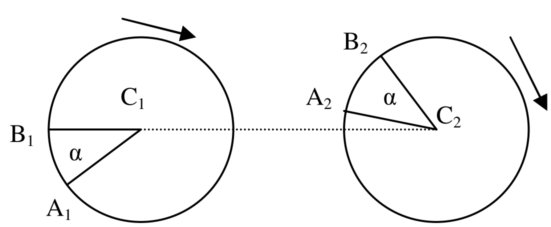

Figure 3. The two beams after t1/2 time

After t_1 time (since the initial deployment) the two beams will be in the starting positions. Beam 2 has completed two full circles while beam 1 has completed one circle. Now the scenario repeats, so the 2 beams will never meet.

So if we pick $v_2 = 2v_1$ the beams might never meet. In fact, if we choose $v_2 = k \cdot v_1$ the 2 beams might never meet ($v_2 = k \cdot v_1$ must be smaller than the upper bound V). Of course, this also depends on α. If α is bigger then the second beam might reach the meeting point before beam 1 ends being available, so they meet. The conclusion is that there is a chance (which depends on α, speeds and the starting positions) that the two beams will never meet.

q.e.d.

Now let us consider the particular case in which $\alpha = 0$. This would correspond to having laser rays (which are straight lines with no angle) instead of antenna beams. We cannot guarantee that the antennas will always meet in this case; it depends on the starting positions. The following Lemma states this formally for $n = 2$:

Lemma 5: If $\alpha = 0$ then for $\forall v_1, v_2 \; \exists \beta_1, \beta_2$ starting angles such that the spinning problem has no solution.

Proof:

We show that there is a starting position that will guarantee the 2 beams will never meet. The proof is non-constructive.

Let $\beta_1 = 0$, $\beta_2 = \pi$ (the antennas start in the meeting position) like in Fig. 4:

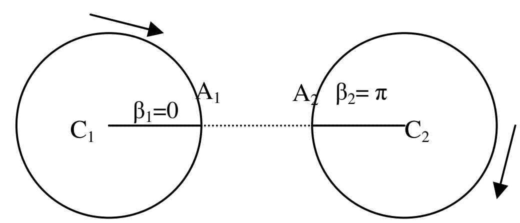

Figure 4. The starting position of 2 rays

Now, there are two possibilities: the antennas will meet again, or they will never meet again.

Case 1: The antennas will never meet again.

In this case, we can pick the positioning of the 2 antennas at time t_1 (after one full rotation of the first beam) like in Fig 5.

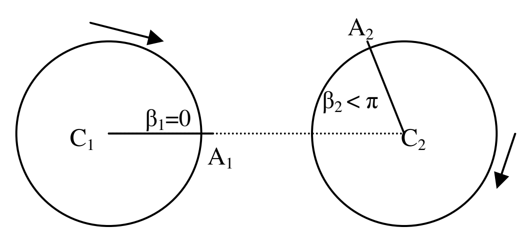

Figure 5. The starting position after one full rotation of C_1

Since beams never meet after t_1 by assumption, the new β_2 will be smaller than π. So if we let the starting positions be these β_1, β_2, according to our assumption for this case, the 2 beams will never meet.

Case 2: The antennas will meet again after k rotations of beam 1, $k \in N^+$ but not earlier (A_1 will be at the starting point only after a number of full rotations).

If the antennas started at the meeting position and they will meet again, it means we have a periodic pattern. That is, they will meet again after $2k$ rotations, $3k$ rotations, and so on.

Now, after $(k - 1)$ rotations we will have a positioning like in Fig. 6. We will consider δ_1 the angle between the second beam and the meeting point after 1 full rotation of the first beam, δ_2 the angle after 2 rotations … δ_{k-1} after k-1 rotations. Let δ be the smallest of them, which occurs after rotation j, where $1 \le j \le k - 1$.

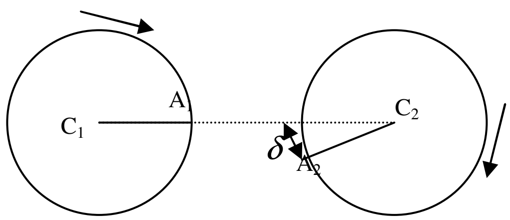

Figure 6. The 2 beams after k – 1 rotations

Now, if we choose other starting positions, with $\beta_1 = 0$ and $\beta_2 = \delta/2$ it means that after j rotations beam one will be at the starting position, beam two will be $\delta/2$ "behind". After k rotations, the second beam will be $\delta/2$ "ahead".

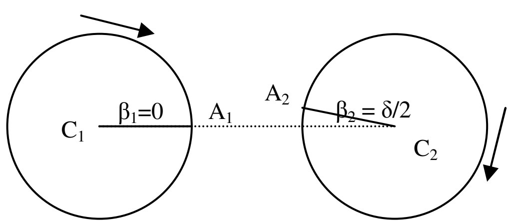

Figure 7. The 2 beams after k rotations

As we stated earlier, we have a pattern. They will not meet after $i \cdot j$ rotations, nor after $i \cdot k$ rotations, with $i \in N$. But they will never be as close to meeting position as after $i \cdot j$ rotations or $i \cdot k$. That is because in the rest of the time the distance between beam 2 and the meeting point will always be greater than $\delta/2$. So the two beams will never meet.

Hence, in either case, we can construct a starting position such that the 2 beams will never meet.

q.e.d.

IV. Preliminaries

In this section we make a few assumptions about the Spinning Problem which will enable us to formally represent all the parameters of the problem. That is because in practice there will always be diffractions at the edges of the beams thus making the signal at the edges weaker than closer to the center. Hence we can "approximate" the continuous model with a discrete one.

First we assume that α is a rational number, i.e. $\alpha = a/b$, $a, b \in Z$, $b \neq 0$. This enables us to divide the circle formed by the rotation of the beam into m equal sections, where m is a multiple of b. Furthermore, the starting angles will also be a rational number of the form c/m. Note that it is possible to pick $m = b$.

These assumptions allow us to reduce the starting position and the angle α to a natural number, representing the number of sections it covers. So an angle α of k means α covers k sections. Also we represent time as time units, without the concern of what a time unit really corresponds to in real world. The above assumptions enable us to formalize the problem and solve it more rigorously.

Now we consider that the first beam rotates k_1 sections in one time unit and the second beam k_2, with k_1 and k_2 natural numbers. $S_1, S_2 \geq 0$ are the starting positions of the two beams. We assume that the first beam starts in the meeting position, which will be true within some finite time after the initial deployment. S_2 varies, and we consider $S_2 = 0 = m$ to be the meeting position.

Next we present the particular case when $\alpha = 0$. The theory presented here is an introduction for the general case. However, it cannot be applied in the real world. We have already proved that for $\alpha = 0$, we cannot guarantee the 2 beams will ever meet. Now we have a few additional assumptions about our model which allow us to analyze this particular case: α is a rational number and the circle has m sections. We can see that as m tends to infinity, our discrete model allows more flexibility in the selection of parameters, and it "approaches" the continuous model. However, in order for meeting to occur, beam 2 must start at one of the m points on the circle, and not somewhere in-between. These points are imaginary, but they are not flexible: one point on each circle must be placed such that when the beams are positioned on them, they are aligned (i.e. we have meeting).

But the devices are dropped arbitrarily, and we cannot know in advance if the beams will start on one of the points. This may not matter when $\alpha > 0$, but when $\alpha = 0$, it is crucial the beams start on the points. So we just assume this happens, in order to provide the idea behind the proofs for the general case.

Given these assumptions, we can now present the case $\alpha = 0$.

Lemma 6: For $\alpha = 0$ and for given S_2, k_1, k_2 and m, we have meeting iff $\exists\, i, j \in N$ such that $i \cdot m \cdot k_2/k_1 = j \cdot m - S_2$

Proof:

Note that if $S_2 = 0$ then we have meeting already, so i and j will be 0. The first beam will rotate a full circle in m/k_1 time.

In m/k_1 time, the second beam will rotate $m \cdot k_2/k_1$. For $\alpha = 0$, the two beams will meet iff the second beam will be at the meeting point at the same time with the first beam. The first beam will be at the meeting point after every m/k_1 time units. But every m/k_1 time units the 2nd beam will rotate $m \cdot k_2/k_1$ distance.

In order to be at the meeting point after i rotations of the first beam, the 2nd beam will have to rotate a total distance of $j \cdot m - S_2$ (j full rotations minus the starting shift). Therefore, for given S_2, k_1, k_2 and m, if $\exists i, j \in N$ such that $i \cdot m \cdot k_2/k_1 = j \cdot m - S_2$, then we have meeting.

q.e.d.

We are interested in the smallest natural i which satisfies this equation.

We can write the previous equation as $i \cdot m \cdot k_2 = j \cdot m \cdot k_1 - S_2 \cdot k_1$. And since S_2, k_1, k_2 and m are known, we can rewrite the equation as $j \cdot a - i \cdot b = c$ with $a, b, c \in N$. This is a linear Diophantine equation with the form $a \cdot x + b \cdot y = c$. There exists a polynomial time algorithm for solving Diophantine equations which is presented in this paper but the reader can find it at [5].

Corollary 7: For $\alpha = 0$ if beams meet then $k_1 = p \cdot m$, $p \in N^+$.

Proof:

We have solution iff $\exists i, j \in N$ such that $j \cdot m \cdot k_1 - i \cdot m \cdot k_2 = S_2 \cdot k_1$

We know from the Diophantine Equation algorithm that the $gcd(a,b)$ must divide c, in other words $m \cdot gcd(k_1,k_2)$ must divide $S_2 \cdot k_1$ for $\forall S_2 = 1..m$. In particular, for $S_2 = 1$, m must divide k_1, which means k_1 can be written as $k_1 = p \cdot m$, $p \in N^+$.

q.e.d.

Theorem 8: For $\alpha = 0$ the two beams meet after at most $m-1$ spins of the first beam in the worst case, or they never meet.

Proof:

After each rotation of the 1st beam, the second beam will visit (be positioned on) one of the m points of the circle. Let p_i be the point the second beam is positioned after i spins. After a number of rotations (let us say r rotations) the beam will be again positioned at point $p_0 = S_2$. After this, all positions become repetitive, meaning that $p_0 = p_r$, $p_1 = p_{r+1}$… and so on. If $r < m$ then there exist some points on the circle that have not been visited. Hence, there exists a starting position for which the meeting point will not be visited, therefore there is no meeting. So in order to have meeting, the second beam must visit all m points on the circle. In the worst case, the meeting point is last visited; therefore the two beams will meet after at most $m-1$ spins in the worst case.

q.e.d.

We now extend our analysis to the case $\alpha > 0$.

Theorem 9: Given k_1, k_2, S_2, m, if $\alpha > 0$ then we have meeting iff $\exists i, j \in N$ such that

$$0 \geq i \cdot m \cdot k_2/k_1 + S_2 - j \cdot m \geq -\alpha \cdot k_2/k_1 - \alpha$$

Proof:

In the case $\alpha > 0$ we consider S_2 to be the distance between the meeting point and the rear margin (A_2) of the second beam, like in the Fig. 8:

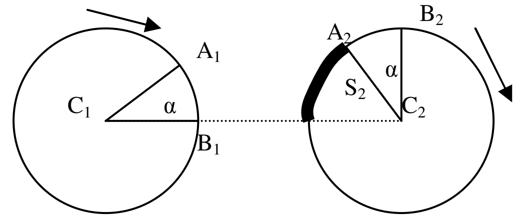

Figure 8. Starting position S_2

Every $m/k1$ time, the first beam will be in the same position as the initial position and it will be available for α/k_1 time. Where should the second beam be positioned on the circle in order to have meeting? If after m/k_1 time the rear margin (A_2) of the second beam is positioned on the meeting point, then we have meeting (Fig. 9).

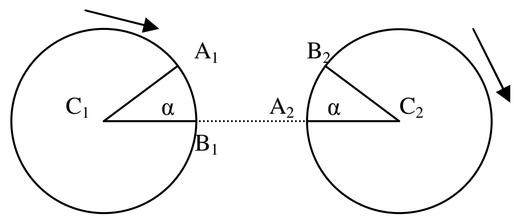

Figure 9. Meeting after m/k_1 time

Or, if the front margin (B_2) is $\alpha \cdot k_2/k_1$ away from the meeting point we still have meeting (Fig. 10). That is because the first beam will be available for α/k_1 time, and in this time the second beam will rotate a distance of $\alpha \cdot k_2/k_1$. In this case, the rear margin is $\alpha + \alpha \cdot k_2/k_1$ away from the meeting point.

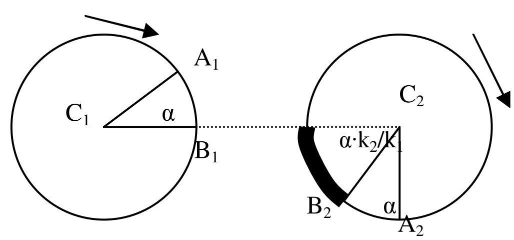

Figure 10. Meeting if B_2 is $a \cdot k_2/k_1$ away

These are the 2 extreme cases. If the 2nd beam is anywhere between these points, we will have meeting. So, if the rear margin is between 0 (the meeting point) and $0 - (\alpha + \alpha \cdot k_2/k_1)$ (the distance beam 2 will rotate in α/k_1 time plus the distance α to the rear margin) then we have meeting.

The total distance the rear margin of the 2nd beam must rotate in $i \cdot m/k_1$ time should be smaller than $j \cdot m - S_2$ and bigger than $j \cdot m - S_2 - (\alpha + \alpha \cdot k_2/k_1)$.

Hence,

$$i \cdot m \cdot k_2/k_1 \leq j \cdot m - S_2 \text{ and } i \cdot m \cdot k_2/k_1 \geq j \cdot m - S_2 - \alpha \cdot k_2/k_1 - \alpha$$

Or,

$i \cdot m \cdot k_2/k_1 - j \cdot m + S_2 \leq 0$

and

$i \cdot m \cdot k_2/k_1 - j \cdot m + S_2 \geq - \alpha \cdot k_2/k_1 - \alpha$.

Therefore, in the case $\alpha > 0$ we have meeting iff $\exists\, i, j \in N$ such that

$0 \geq i \cdot m \cdot k_2/k_1 + S_2 - j \cdot m \geq - \alpha \cdot k_2/k_1 - \alpha$

q.e.d.

V. AN ALGORITHM FOR COMPUTING THE MEETING TIME

5.1. Description of the algorithm

The inequality of Theorem 9 can be also written as:

$- S_2 \cdot k_1 \geq i \cdot (m \cdot k_2) - j \cdot (m \cdot k_1) \geq - S_2 \cdot k_1 - \alpha \cdot (k_1 + k_2)$

We need to solve all Diophantine Equations

$i \cdot (m \cdot k_2) - j \cdot (m \cdot k_1) = c$ where $c \in Z$

and

$- S_2 \cdot k_1 \geq c \geq - S_2 \cdot k_1 - \alpha \cdot (k_1 + k_2)$.

Then, for every solution i, we must compute the exact meeting time. Because at this point we only know the two beams will meet after i full rotations of the first beam, but we do not know exactly when this meeting occurs. First we compute the position of the second beam after $i \cdot m/k_1$ time. Let p_i be the position of the rear margin of the second beam after $i \cdot m/k_1$ time.

$p_i = (i \cdot m \cdot k_2/k_1) mod\ m$

If $m - p_i \leq \alpha$ then we have the situation from Fig 11:

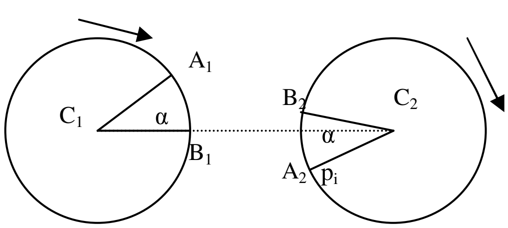

Figure 11. When m-$p_i \leq \alpha$

Hence, after exactly $i \cdot m/k_1$ time, the 2 beams meet. But if $m - p_i > \alpha$, then after $i \cdot m/k_1$ time the two beams are positioned like in Fig 12.:

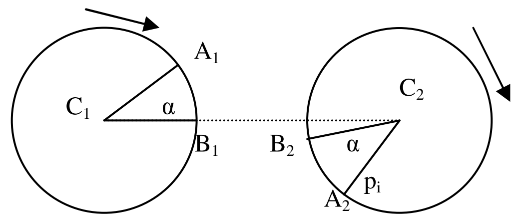

Figure 12. When m-$p_i > \alpha$

Hence, it will take a while until the second beam will be in the meeting position. The second beam will have to rotate the distance between B_2 (the front margin) and the meeting point. That distance is $m - p_i - \alpha$. Hence, the exact meeting time is $i \cdot m/k_1 + (m - p_i - \alpha)/k_2$. After solving all the Diophantine Equations, we can pick the smallest solution as the best meeting time for this choice of k_1, k_2, S_2, m.

Remember, that this is the meeting time if we consider the starting position of the first beam like in Fig. 13:

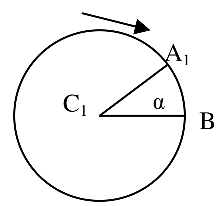

Figure 13. Starting position of beam 1

Now we need to compute the meeting time for any starting positions.

Lemma 10: Assume t is the worst case meeting time of the 2 beams if $S_1 = 0$ and $0 \leq S_2 \leq m - 1$ (computed by our algorithm). Then the worst case meeting time for any S_1, S_2 is smaller than $t + m/k_1$.

Proof: Obvious: For any S_1 the first beam will have to rotate less than a full spin to get into position $S_1 = 0$ (at most $m - 1/k_1$ time). So it will take less than m/k_1 to reach $S_1 = 0$ and once it is there, it will take t time to meet.

q.e.d.

Theorem 11: Given k_1, k_2, S_1, S_2, m and $\alpha > 0$ we have meeting iff $\exists\, i, j \in N$ such that

$0 \geq i \cdot m \cdot k_2/k_1 + S_2 + (m - S_1) \cdot k_2/k_1 - j \cdot m \geq - \alpha \cdot k_2/k_1 - \alpha$

or

$(\alpha - S_1) \cdot k_2/k_1 + S_2 \geq m - \alpha$ and $S_1 < \alpha$.

Proof:

In the Lemma 6 we proved the same result but for $S_1 = 0$. If $S_1 > 0$ we have two possibilities: either the two beams meet during the first full rotation of the first beam or the beams meet later.

The first case is a particular case, and may occur only if $S_1 < \alpha$ (so the first beam is already available) and the second beam rotates so fast, that it will reach the meeting point before beam one ends being available. That means, in $\alpha - S_1/k_1$ time, beam two will rotate more than $m - \alpha - S_2$ distance.

Hence

$$(\alpha - S_1) \cdot k_2/k_1 + S_2 \geq m - \alpha.$$

The second case is trivial, considering the Theorem which was already proven at the beginning of section 4. For $S_1 = 0$, the equation was

$$0 \geq i \cdot m \cdot k_2/k_1 + S_2 - j \cdot m \geq -\alpha \cdot k_2/k_1 - \alpha$$

Since the beams do not meet during the first rotation of the first beam, then we can consider the problem with $S_1 = 0$ but with a different (later) S_2. S_2 will be S_2 plus the distance beam two rotates in $(m - S_1)/k_1$ time (the time it takes until beam one reaches position $S_1 = 0$). Therefore, the new S_2 will be $S_2 + (m - S_1) \cdot k_2/k_1$. Replacing S_2 in the equation we get

$$0 \geq i \cdot m \cdot k_2/k_1 + S_2 + (m - S_1) \cdot k_2/k_1 - j \cdot m \geq -\alpha \cdot k_2/k_1 - \alpha$$

q.e.d.

How to compute the exact meeting time for any initial shift of both beams?

First we compute the meeting time t when $S_1 = 0$, using the previous algorithm. Then we try to find the exact snapshot of the meeting: exactly what point of the first beam will overlap the meeting point? In order to do this, we first compute the distance d between the front margin B_1 and the meeting point after t time.

$$d = (k_1 \cdot t) \bmod m$$

For any $S_1 < \alpha - d$ the meeting will occur at the same time as for $S1 = 0$ because after t time the distance between the front margin and the meeting point will be smaller than $\alpha - d + d = \alpha$. Hence, the meeting point will be between the two margins, therefore we have meeting.

For any $S_1 \geq \alpha - d$ the meeting will not occur at the same time as for $S_1 = 0$ (same justification). Therefore, it will occur later. When? The beam will first have to get in position $S_1 = 0$. Once it is there, it will take t time to meet. Hence, the first beam will have to rotate $m - \alpha + d$ distance.

In this case, the worst case meeting time is

$$t + (m - \alpha + d)/k_1.$$

5.2. Algorithm for two devices

In this section we present the algorithm for finding the best choices of speed for 2 devices. The following algorithm outputs a matrix where every value at row x and column y represents the meeting time if $k_1 = x$ and $k_2 = y$. To find the best meeting time we need to compute the minimum meeting time inside the generated matrix (which can easily be done in $O(m^2)$) additional time.

```
MatrixGeneration{
  for k1=Z to V-1 do
    for k2=k1+1 to V do{
        a = m*k2;
        b = -m*k1;
        g = gcd(a,b);
        max = 0; // max is the longest time
               // until we guarantee meeting
        for S=1 to m do{
          min = MAXINT; // min is the soonest
                       // meeting time
          for c=-S*k1-alpha*(k1+k2)to -S*k1
            if c%g = 0 then {
               a2 = a/g;
               b2 = b/g;
               c2 = c/g;
               solve Diophantine Eq a2*i+b2*j=c2;

        // return a solution (i,j)
        find minimum strictly positive solution (i,j)
               if (min > i)
               min = i;
            }
            if (max < min)
              max = min;
         }
         Matrix[k1][k2] = max + m/k1;
        //we add one more rotation to the final
        //meeting time
    }
}
```

For a better understanding of the MatrixGeneration algorithm, the reader should be familiarized with the algorithm for solving linear Diophantine Equations ([5]).

It is obvious that the matrix is generated in polynomial time. Also, finding the best meeting time for 2 devices can be done in polynomial time. But in order to determine the best time for n devices, we would have to check all possible solutions, because a good meeting time for devices 1 and 2 might result in a very bad meeting time between devices 2 and 3, etc. Hence, we have an exponential running time solution for the Spinning Problem.

5.3 Algorithm for n devices

Next, we present the algorithm for computing the best meeting time for n devices using the matrix generated by the previous algorithm. This algorithm is recursive and it has exponential running time. For every possible solution A (where A is an array of n speeds) we compute the worst case meeting time (compute_time function) then we find the minimum of all these meeting times.

```
spin()
{
  A[0]=Z-1;
  compute(1);
  min= ;
  for all solutions A
    if min>time  then
      min=time ;
        best_solution = A;
  //the best worst case meeting time is min
  // and the best solution (choice of speeds) is A
}
compute(int i)
{
  if i<n then
    for A[i]=A[i-1]+1 to V-n+i
```

```
      compute(i+1);
  else
    for A[i]=A[i-1]+1 to V-n+i
      time  = compute_time(A);
}
int compute_time(array A)
{
  max=0;
  for j=1 to n-1
    for k=j+1 to n
      if max<Matrix[A[j]][A[k]] then
        max=Matrix[A[j]][A[k]]
  return max;
}
```

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented and analyzed a model for neighbor discovery in static wireless ad-hoc networks using directional antennas. We defined our variant of the problem and made some assumptions about our model after which we provided theoretical descriptions of the solutions and a simple algorithm.

The solution for two devices is intuitive and quite simple. Even though we have presented a method to determine the optimal solution to the Spinning Problem for n devices, this method requires exponential time. The next step in solving this problem is proving that it is NP-Hard, or finding a polynomial time algorithm.

Another approach could be to use random rotation speeds and compute the discovery probability or the expected meeting time.

REFERENCES

[1] S. Vasudevan, J. Kurose, D. Towsley "On Neighbor Discovery inWireless Networks With Directional Antennas". 24th Annual Joint Conference of the IEEE Computer and Communication Societies, Proceedings IEEE. Infocom 2005.

[2] Wattenhofer R., Li Li, Bahl P., Wang Y. "Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks". Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Anchorage, Alaska, April 2001.

[3] Li Li, Halpern J.Y., Bahl P., Wang Y., Wattenhofer R. "Analysis of a Cone-Based Distributed Topology Control Algorithm for Wireless Multi-hop Networks". Twentieth ACM Symposium on Principles of Distributed Computing (PODC), Newport, Rhode Island, August 2001.

[4] N.Malhotra, M. Krasniewski, C. Yang, S. Bagchi, W. Chappell "Location Estimation in Ad-Hoc Networks with Directional Antennas". The 25th International Conference on Distributed Computing Systems (ICDCS 2005), Columbus, Ohio, June 2005.

[5] MathWorld at Wolfram Research, http://mathworld.wolfram.com