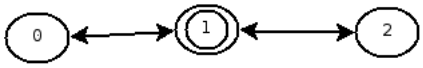
CDS Expansion Algorithm

The CDS Expansion Algorithm is composed of two algorithms. One for constructing the Connected Dominating Set, and other algorithm for performing the expansion, using the information provided by the CDS contruction algorithm.

1 CDS Construction Algorithm

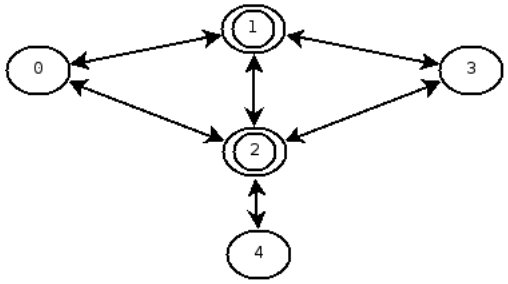
The algorithm used for the construction of the CDS is described in [1]. The algorithm is summarized in Algorithm 1. The algorithm allows to determine as a local decision of each node whether the node should be part of the Connected Dominating Set or not. The following is the description of what happens at each node:

1. The node starts unmarked, this is, the node is not part of the CDS.
2. The node exchanges its list of neighbors with all its neighbors. This involves sending al least two beaconing messages, the fist one to let know everyone around that the node is there, and the second one, just after collecting the first beacon from all the neighbors, to let everyone know node the list nodes seen by the node. At this point each node knows:
   - Its list of neighbors
   - The list of neighbors of each of its neighbors
3. With this information the node decides to mark itself as a CDS node if there is unconnected neighbors. This is, if two of its neighbors don't contain each other as neighbors. As an example take the following topology:

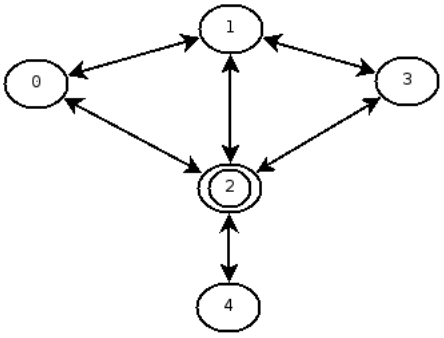
   

   The list of neighbors of 0 is $\{1\}$, the list of neighbors of 1 is $\{0, 2\}$, the list of neighbors of 2 is $\{1\}$. At this point, 0 and 2, don't do anything because they only have one neighbor each. But 1, decides to mark itself as a CDS node, because it has two neighbors, 0 and 2, that don't see each other.
4. Refine the CDS. At step 3 we already have a CDS, but its size can be reduced using the following 2 rules:

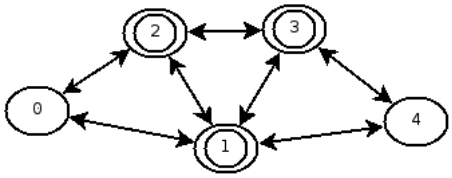
RULE 1: If all the neighbors of a node, including the node itself, are included in the list of neighbors of one of its neighbors that has a higher id and that is a CDS, then the node removes itself from the CDS. For example, take the topology:



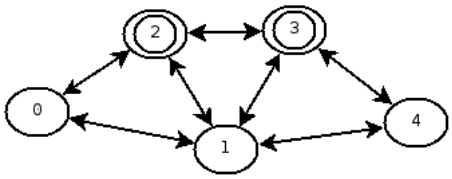
Nodes 1 and 2 are CDS because they noticed that 0 and 3 are disconnected. Now if we apply RULE 1, the list of neighbors of 1, including itself, $\{0, 1, 2, 3\}$, is included in the list of neighbors of 2, including itself, $\{0, 1, 2, 3, 4\}$, and 1 has a lower id than 2, so as a result 1 unmarks itself:



RULE 2: If all the neighbors of a node, not necessarily including the node itself, are included in the union of two or more of its neighbors that have higher ids and that are CDSs, then the node removes itself from the CDS. For example take the topology:



Nodes 1 is CDS because 0 and 4, 2 and 4 and 0 and 3, are not connected, Node 2 is CDS because 0 and 3 are not connected and Node 3 is CDS because 2 and 4 are not connected. Now, if we apply RULE 2, the list of neighbors of 1, $\{0, 2, 3, 4\}$, is included in the union of the list of neighbors of 2 and 3, which are CDS nodes, $\{0, 1, 2, 3, 4\}$, so as a result 1 unmarks itself:



Algorithm 1 CDS algorithm

Input: n {this node's index}
Input: $\mathcal{N}$ {list of neighbors}
Input: $\mathcal{N}_i$ for each $i \in \mathcal{N}$ {neighbor's lists of neighbors}
  $m \leftarrow 0$ {Start as an unmarked node}
  if $\exists i, j \in \mathcal{N} : i \notin \mathcal{N}_j$ **then**
    $m \leftarrow 1$
    if $\exists i \in \mathcal{N} : i \in CDS \bigwedge n < i \bigwedge \mathcal{N} \subseteq \mathcal{N}_i$ **then**
      $m \leftarrow 0$
    else
      if $\exists i, j \in \mathcal{N} : i, j \in CDS \bigwedge n < i \bigwedge n < j \bigwedge \mathcal{N} \subseteq \mathcal{N}_i \bigcup \mathcal{N}_j$ **then**
        $m \leftarrow 0$
      end if
    end if
  end if

2 Expansion Algorithm

The expansion algorithm uses the information provided by the CDS construction mechanism to decide how to expand the network to try to find new nodes and minimize the islanding. The expansion algorithm runs periodically on each node and it has 2 elements. The first element is to determine if the node should expand or contract. This function takes as input the reception power of packets coming from CDS nodes. The second element is the function that determines the direction and the distance that the node should move during the next interval.

1. **Expansion or contraction**

   - Calculate the average of the reception power of packets coming from all CDS nodes around the node.
   - If the average is above a maximum threshold (in the simulations the value selected has 1.2 times the reception threshold for non CDS nodes and 2.5 for CDS nodes), then the node should expand.
   - If the average is below a minimum threshold (in the simulations the value selected was 1.0 times the reception threshold for non CDS nodes and 2.0 for CDS nodes), then the node should contract.

   As a corollary,

   - For expanding, the node should seek to reduce the value of this function.
   - For contracting, the node should seek to increase the value of this function.

2. **Direction and Distance**

   - The angle is an uniform random variable that might take values in certain intervals. The intervals are taken from an array indexed by a counter of the decisions taken, each time the decision is changed, the counter is reseted. The intervals are ordered so an interval in a higher index of the array of intervals is narrower. The following function represents the array used for selecting the angle:

$$\mathcal{A}(x) = \begin{cases} \frac{\pi}{2} & \text{if } x = 0, \\ \frac{\pi}{3} & \text{if } x = 1, \\ \frac{\pi}{4} & \text{if } x = 2, \\ \frac{\pi}{9} & \text{if } x = 3, \\ \frac{\pi}{18} & \text{if } x = 4, \\ \frac{\pi}{36} & \text{if } x = 5, \\ \frac{\pi}{90} & \text{otherwise,} \end{cases} \tag{1}$$

     where x is the counter mentioned above.

   - The distance is taken from an array indexed by a counter of the decisions taken, and each time the decision is changed, the counter is reseted. The distances are ordered so a distance in a higher index of the arrays of distances is longer. The following function represents the array used for selecting the distance to move:

$$\mathcal{D}(x) = \begin{cases} 5 & \text{if } x = 0, \\ 9 & \text{if } x = 1, \\ 12 & \text{if } x = 2, \\ 15 & \text{if } x = 3, \\ 17 & \text{if } x = 4, \\ 19 & \text{if } x = 5, \\ 20 & \text{otherwise.} \end{cases} \tag{2}$$

Algorithm 2 summarizes the expansion algorithm.

Algorithm 2 Expansion algorithm

Input: s_t {current average reception signal}
Input: s_{t-1} {average reception signal during last run of algorithm}
Input: x_t, y_t {current node position}
Input: x_{t-1}, y_{t-1} {node position during last run of algorithm}
Input: T {threshold interval for reception signal $[1.0, 1.2]$ for non CDS nodes and $[2.0, 2.5]$ for CDS nodes}
Input: p {score of moving in a direction}

```
x_{t+1} ← x_t
y_{t+1} ← y_t
if s_t ∉ T then
    α ← arctan((x_t − x_{t−1}) / (y_t − y_{t−1}))
    β ← α + uniform(−A(p), A(p))
    d ← D(p)
    if (s_t < inf(T) · RX_thresh ∧ s_t < s_{t−1}) ∨ (s_t > sup(T) · RX_thresh ∧ s_t > s_{t−1}) then
        β ← β + π
        p ← 0
    else
        p ← p + 1
    end if
    x_{t+1} ← x_t + d · cos(β)
    y_{t+1} ← y_t + d · sin(β)
end if
return (x_{t+1}, y_{t+1})
```

References

[1] Jie Wu and Hailan Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *DIALM '99: Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 7–14, New York, NY, USA, 1999. ACM.