A Flexible Method to Establish Reliable Links in Bluetooth Networks

Dr. Goldthwait, Jenna Wise, Sasha Monroe, Kevin London, Tyler McVicker

MWCDS Algorithm

```
if i = root then
/* Rule for root */
    { (R0) if d_root ≠ 0 ∨ m_root ≠ 0 ∨ s_root ≠ 1
          then { d_root ← 0; m_root ← 0; s_root ← 1; }
else
/* Rule for node i ≠ root */
    { (R1) if d_i ≠ ρ_i ∨ m_i ≠ (enter_i ∨ leave_i)
          then { d_i ← ρ_i;                          [Update_d]
               { m_i ← (enter_i ∨ leave_i);           [Update_m]
      (R2) else if d_i = ρ_i ∧ (enter_i ∨ leave_i) ∧ minM_i = i
          then { m_i ← 0;
               { if enter_i then s_i ← 1;             [Enter]
               { else if leave_i then s_i ← 0;        [Leave]
```

Figure 1: Algorithm MWCDS at Node i, $1 \leq i \leq n$

$$p_i = \min\{d_j | j \in N(i)\} + 1$$

$$minM_i = \min\{j | j \in N[i] \wedge d_j = d_i \wedge m_j = 1\} \text{ where } \min\{\} = null$$

$$\mathbf{enter_i} \stackrel{\text{def}}{\equiv} (s_i = 0) \wedge (\forall j \in N_{<}(i) : s_j = 0)$$

$$\mathbf{leave_i} \stackrel{\text{def}}{\equiv} (s_i = 1) \wedge (\exists j \in N_{<}(i) : s_j = 1)$$

- The algorithm constructs a minimal weakly connected dominating set (MWCDS) in incremental fashion.
- The root (a distinguished node) is initially set to be a member of a set S.
- In the incremental step, each non-root node looks at its neighbors with equal or smaller distance to root.
- If a neighbor is already in the set S, then the non-root node leaves the set S .
- If all of its neighbors are not in the set S, then the non-root node enters into the set S.
- To avoid an infinite loop in the incremental step, we enforce nodes with the same distance to root to change their memberships depending on their ID.
- The algorithm runs on a synchronous daemon.

Algorithm Implementation

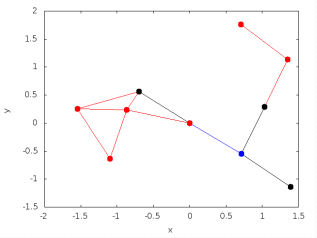

Figure 2: Graph at generation

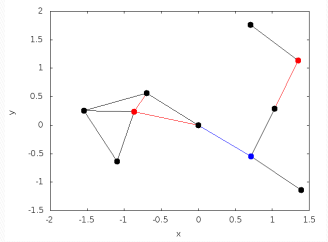

Figure 3: Graph at final step of MWCDS

- The graphs to the left illustrates the MWCDS algorithm.
- The black dots represent nodes that are not in the set.
- Red dots represent nodes that are in the set.
- The blue dot is the root node.
- Figure 2 is the graph generated for the algorithm to run on.
- Figure 3 shows the outcome of the algorithm, a fully converged network. Every slave node is connected to at least one node that is in the set.

Bluetooth Implementation

Bluetooth enables wireless communication via ad-hoc networks. These networks are usually comprised of personal device such as smartphones, headsets, and laptops. The network topologies of Bluetooth include piconets and scatternets. A piconet is a collection of slaves controlled by a master. A scatternet is a group of interconnected piconets that create a multihop network. A hop is one portion between a source and a destination, incremented as information moves from computer to computer through the network.

Every slave node must be connected to a master node. The MWCDS algorithm identifies the master nodes as being in the dominating set. By identifying these nodes, another algorithm can be written to route traffic between the nodes based. The MWCDS algorithm also determines the distance of each node, in hops, in relation to the root. Each master node, or node in the set, can be viewed as a junction.

Future Work

We hope to further develop this algorithm to create a routing algorithm that can be implemented on physical Bluetooth networks and that can adapt to change.

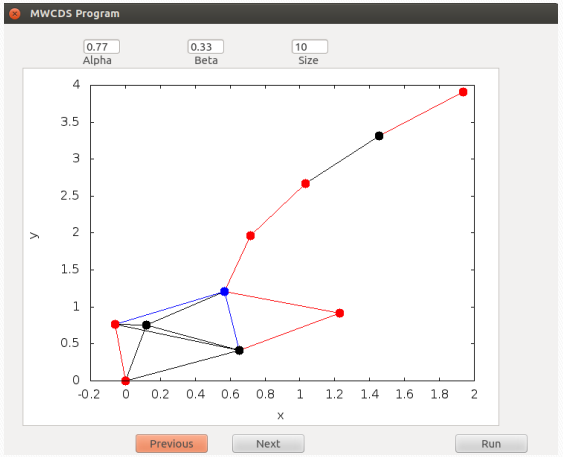

Figure 4: Illustration of our GUI in real time.

The figure above illustrates our GUI in real time.

- Enter alpha, beta, and size parameters
- Press "Run"
- Use the "Previous" and "Next" buttons to step through the algorithm from graph generation to termination.

- **Alpha:** Larger values of alpha result in a more geographically spread out network and
- **Beta:** Larger values of beta result in lower diameter networks.

Selected References:

- Liron Har-Shai, Ronen Kofman, Gil Zussman, and Adrian Segall, *Inter-Piconet Scheduling in Bluetooth Scatternets*, Proc. of the OPNETWORK 2002 Conference, (2002).
- Pradip K. Srimani, Sandeep K.S. Gupta, *Adaptive core selection and migration method for multi-cast routing in mobile ad hoc networks*, IEEE Transactions on Parallel and Distributed Systems 14 (2003), 1.
- Yihua Ding, James Z. Wang, Pradip K. Srimani, *A linear time self-stabilizing algorithm for minimal weakly domination sets*.

Choose Ohio First

Youngstown STATE UNIVERSITY