

Supplemental Material of

Merged Path: Data Dissemination for Large Scale Multiple Mobile Sinks WSN

Ammar Hawbani, Xingfu Wang, Omar Busaileh, Ammar Qarariyah, Liang Zhao, Mohammed A. A. Alqaness, and S. H. Alsamhi

This document is a supporting content for the manuscript (Merged Path: Data Dissemination for Large Scale Multiple Mobile Sinks WSN), given the limited number of pages allowed by the publisher. We provided more details about the algorithms and the simulation results.

Index Terms: Wireless sensor networks, multiple mobile sinks, data routing, merged path.



1. CLUSTERING AND BIFURCATING

1.1 Section formula:

The bifurcation point is exactly where the merged path should be branched to several sub-branches, and explicitly where the clustering process is computed. We used section formula (Point that divides a line in given ratio). The code for this formula is as follows:

```
public static Point GetPointMtoNratio(Point p1, Point p2, double d1, double d2)
{
    double x = ((d2 * p1.X) + (d1 * p2.X)) / (d1 + d2);
    double y = ((d2 * p1.Y) + (d1 * p2.Y)) / (d1 + d2);
    return new Point(x, y);
}
```

1.2 Clustering Algorithm:

The goal of this algorithm is to divide the sinks into groups and to find the branches where the packet to should be traveled. Thus, this algorithm returns a list of branches. The branch object contains Cluster, Start Point and End point of the cluster. Cluster contains a list of sinks of the branch. Start Point is the start point of the branch while End point is the end of the branch. Before explaining how to find the branches, we will first explain how the clustering algorithm runs. The high-tier node sends back the locations of the sinks at the network to the source node that has data to send. Thus, the clustering algorithm is performed at current source node (the current bifurcation point). Thus the inputs for the clustering algorithm are **bk_sinks** (the locations sinks in the field which responded by the high-tier node), the **b_k** (current bifurcation point), and the **clusteringThreshold**. Clustering algorithm runs in three steps as follows:

Finding the Angles

```
private void FindAngles()
{
    int id = 0;
    for (int i = 0; i < sinks.Count; i++)
    {
        Sensor isink = sinks[i];
        for (int j = i + 1; j < sinks.Count; j++)
        {
            if (i != j)
            {
                id++;
                Sensor jsink = sinks[j];
                double ang = new Angle().GetAngle(isink.Position, jsink.Position, sourcePointCenterLocation);
                AngleSimlirityEdge edge = new AngleSimlirityEdge() { Angle = ang, StartVertex = isink, EndVertex = jsink, ID = id };
                isink.SimlirityEdges.Add(edge);
                jsink.SimlirityEdges.Add(edge);
            }
        }
    }
}
```

