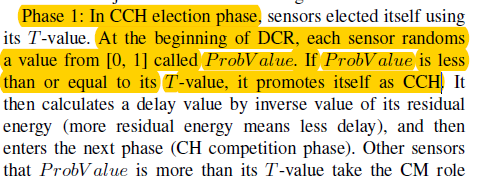
**แบบ dynamic T-values**

* กำหนด กว้างยาว พื้นที่ \*
* กำหนด density เลือก [0.00625, 0.0125 ,0.025 ,0.05] หน่วย node/m^2 \*
* กำหนด ค่า t (fix ไว้สำหรับตอนแรก) \*
* กำหนด R1 = 30 \*
* Packet control = 200 bit \*
* Packet data = 4000 bit \*
* Elec tran = 50 \* (10 \*\* (-9)) 50 Nano J \*
* Elec recive = 50 \* (10 \*\* (-9)) 50 Nano J \*
* fs = 10 \* (10 \*\* (-12)) 10 Pico j \*
* mpf = 0.012 \* (10 \*\* (-12)) 0.012 Pico j \*
* d threshold = 87 \*
* R1 = 30 meter
* R2 = R1\*(2ln10) meter
* จำนวน Base station
* Loop ตามจำนวน bs
* Input ตำแหน่ง bs
* หาจำนวน node by width x height density
* Randomตำแหน่ง x,y ตามจำนวน nodeไว้เก็บ ; ห้ามซ้ำกัน ห้ามซ้ำกับbase

Node[0], Node[1], Node[2], Node[3]

[ตำแหน่งx, ตำแหน่งy, ค่าพลังงาน, ค่า t\_predefine]

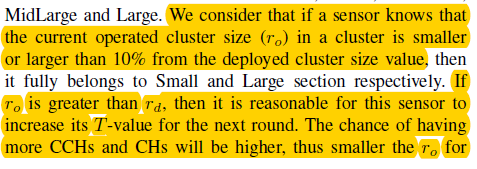
* เก็บใส่ node\_member
* จำนวน candidate มาจาก
* Random ใน node\_member ตามจำนวน candidate เก็บไว้ใน candidate\_member
* ตัว candidate ประกาศ broadcast ให้ candidate ตัวอื่นรู้ ด้วย packet control ที่มี ขนาดรัศมี R1 กับค่าพลังงาน[2]

ตัวส่งเสียพลังงานตามระยะทางตาม d threshold ตัวรับเสียพลังงานตอนรับ

* ถ้าระยะห่าง Candidate น้อยกว่ามากกว่า R1 ให้ตัวที่มีพลังงานมากกว่าได้เป็น cluster head ตัวที่มีพลังงานน้อยกว่ากลับไปเป็น node member เหมือนเดิม
* ใน วงของ cluster หาระยะห่างระหว่าง node ที่มากที่สุด จาก node ทุกตัวในวงต้องส่ง pkt controlที่มีข้อมูลระยะห่าง มาให้ cluster

Node ส่งเสียพลังงาน cluster เสียพลังงานตอนรับ

* ถ้าระนะห่างเกิน R0 ในวงนั้นต้องลดค่า t predefine จาก cluster ประกาศpkt control ทีมีค่า t\_predefine ใหม่ไปให้ node ที่อยู่ภายในวงตัวเอง cluster เสียพลังงานตอนส่ง Node เสียพลังงานตอนรับ



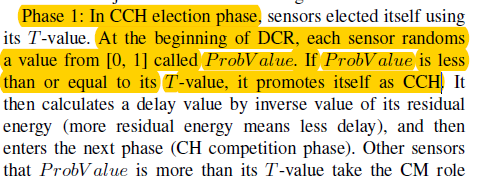
* Node ส่ง packet data ให้กับ cluster head node เสียพลังงานตอนส่งให้ cluster , cluster เสียพลังงานตอนรับ
* Cluster รวบรวมข้อมูลจากที่ได้รับมา หาค่าเฉลี่ยและ ส่งให้กับbase station cluster เสียพลังงานไปตามระยะทาง d threshold
* นับเป็น 1 รอบ
* ทำจนกว่าจะมีตัวใดตัวหนึ่งหมดพลังงานไป
* ทำ t predefine แบบ dynamic มา plot graph

**แบบ fixed T-values**

* กำหนด กว้างยาว พื้นที่ \*
* กำหนด density เลือก [0.00625, 0.0125 ,0.025 ,0.05] หน่วย node/m^2 \*
* กำหนด ค่า t (fix ไว้สำหรับตอนแรก) \*
* กำหนด R1 = 30 \*
* Packet control = 200 bit \*
* Packet data = 4000 bit \*
* Elec tran = 50 \* (10 \*\* (-9)) 50 Nano J \*
* Elec recive = 50 \* (10 \*\* (-9)) 50 Nano J \*
* fs = 10 \* (10 \*\* (-12)) 10 Pico j \*
* mpf = 0.012 \* (10 \*\* (-12)) 0.012 Pico j \*
* d threshold = 87 \*
* R1 = 30 meter \*
* R2 = R1\*(2ln10) meter
* จำนวน Base station\*
* Loop ตามจำนวน bs
* Input ตำแหน่ง bs
* หาจำนวน node by width x height density\*
* Randomตำแหน่ง x,y ตามจำนวน nodeไว้เก็บ\* ; ห้ามซ้ำกัน ห้ามซ้ำกับbase

Node[0], Node[1], Node[2], Node[3]

[ตำแหน่งx, ตำแหน่งy, ค่าพลังงาน, ค่า t\_predefine]

* เก็บใส่ node\_member \*
* จำนวน candidate มาจาก
* Random ใน node\_member ตามจำนวน candidate เก็บไว้ใน candidate\_member
* ตัว candidate ประกาศ broadcast ให้ candidate ตัวอื่นรู้ ด้วย packet control ที่มี ขนาดรัศมี R1 กับค่าพลังงาน[2]

ตัวส่งเสียพลังงานตามระยะทางตาม d threshold ตัวรับเสียพลังงานตอนรับ

* ถ้าระยะห่าง Candidate น้อยกว่ามากกว่า R1 ให้ตัวที่มีพลังงานมากกว่าได้เป็น cluster head ตัวที่มีพลังงานน้อยกว่ากลับไปเป็น node member เหมือนเดิม
* ใน วงของ cluster หาระยะห่างระหว่าง node ที่มากที่สุด จาก node ทุกตัวในวงต้องส่ง pkt controlที่มีข้อมูลระยะห่าง มาให้ cluster

Node ส่งเสียพลังงาน cluster เสียพลังงานตอนรับ

* Node ส่ง packet data ให้กับ cluster head node เสียพลังงานตอนส่งให้ cluster , cluster เสียพลังงานตอนรับ
* Cluster รวบรวมข้อมูลจากที่ได้รับมา หาค่าเฉลี่ยและ ส่งให้กับbase station cluster เสียพลังงานไปตามระยะทาง d threshold
* นับเป็น 1 รอบ
* ทำจนกว่าจะมีตัวใดตัวหนึ่งหมดพลังงานไป
* ทำ fix t predefine ที่ 0.1-0.9 และ แบบ dynamic มา plot graph
* นำ multithreading มาใช้เร่งให้เร็วขึ้น \*\*\*\*
* ลองค่า t หลายๆค่า ดูว่าเข้าใกล้ t = 0.21 (ค่าเฉลียของ dynamic) หรือไม่ \*\*\*\*
* Dead Node เกิดตรงไหนมากที่สุด
* นับจำนวนรอบที่ตายเพื่อเปรียบเทียบกันว่าแบบไหนดีที่สุด \*\*\*\*
* ลอง ใช้ cluster ตัวเดิมรันจนกว่าจะตายแล้วนับจำนวนรอบแล้วค่อยๆลดจำนวนรอบที่ต้องใช้ตัวเดิมเรื่อยๆ จำกว่าจะหาจุดสมดุลได้\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*