

無線通訊網路 Project - Handoff Policy
H34044016 劉士宏

程式說明：

使用程式語言: c++, python

操作方式：

g++ HandoffSim.cpp

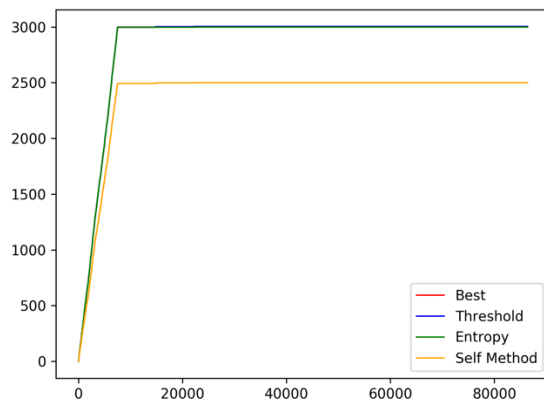
./a.out

python3 plot.py

result.png 即為輸出圖檔

圖表:

handoff times:	Average Power(dBm)
Best=3004	-118.528
Threshold=3004	-118.528
entropy = 2998	-118.528
self = 2499	-118.528



Source code:

一開始先由 power_cal 來計算每一次接收到基地台功率的大小，將最大的值 assign 給 Pcur 並由 hearing 紀錄目前接收的基地台編號

```
void power_cal(base b1, base b2, base b3, base b4) {  
    double dist1 = 0;  
    int xd, yd;  
    xd = (x-b1.x)*(x-b1.x);  
    yd = (y-b1.y)*(y-b1.y);  
    dist1 = sqrt(xd +yd);  
    double chk = 0;  
    if (dist1 > 0 )chk = log10(dist1);  
}
```

```

P1 = -60 -20 * chk;

chk =0;
xd = (x-b2.x)*(x-b2.x);
yd = (y-b2.y)*(y-b2.y);
double dist2 = sqrt(xd +yd);
if (dist2 > 0 )chk = log10(dist2);
P2 = -60 -20 * chk;

chk =0;
xd = (x-b3.x)*(x-b3.x);
yd = (y-b3.y)*(y-b3.y);
double dist3 = sqrt(xd +yd);
if (dist3 > 0 )chk = log10(dist3);
P3 = -60 -20*chk;

chk =0;
xd = (x-b4.x)*(x-b4.x);
yd = (y-b4.y)*(y-b4.y);
double dist4 = sqrt(xd +yd);;
if (dist4 > 0 )chk = log10(dist4);
P4 = -60 -20*chk;
cmp(P1, P2, P3, P4, Pcur, hearing);
cout << "Pcur is " <<Pcur<<endl;
}

```

Best Policy: 當 Pcur 比原先的值 Pold_1 大時且基地台不同時則 handoff

```

void best_policy (int& a) {
    if (Pcur > Pold_1 && hearing!=old_h1){
        cout<<"best count"<<endl;
        cout<<"current best is : " <<a<<endl;
        cout<<"Pold_1 : " <<Pold_1<<endl;
        handoff_best++;
        a++;
        Plisten = Pcur;
        cout<<"Plisten_1 : " <<Plisten<<endl;
        Pold_1 = Plisten;
        Pbest = Plisten;
    }
}

```

```

        old_h1 = hearing;
    }
}

```

Threshold Policy: 當 Pcur 比原先的值 Pold_2 大時, Pold_2 比 threshold 小且基地台不同時則 handoff

```

void threshold (int& b) {
    if ((Pcur > Pold_2) && (Pold_2 < T) && (hearing!=old_h2)){
        cout<<"threshold count"<<endl;
        cout<<"current threshold is : "<<b<<endl;
        cout<<"Pold_2 : "<<Pold_2<<endl;
        handoff_threshold++;
        b++;
        Plisten = Pcur;
        cout<<"Plisten_2 : "<<Plisten<<endl;
        Pold_2 = Plisten;
        Pthresh = Plisten;
        old_h2 = hearing;
    }
}

```

Entropy Policy: 當 Pcur 比原先的值 Pold_3+entropy 大時,且基地台不同時則 handoff

```

void entropy (int&c) {
    if (Pcur > (Pold_3+E)&& (hearing!=old_h3)) {
        cout<<"entropy count"<<endl;
        handoff_entropy++;
        c++;
        Plisten = Pcur;
        cout<<"Plisten_3 : "<<Plisten<<endl;
        Pold_3 = Plisten;
        Pentro = Plisten;
        old_h3 = hearing;
    }
}

```

Self method: 在兩個基地台中間重疊區域不做 handoff

和前三者相比的特點在於，為在重疊處的地方不做考慮 handoff 因此整個 handoff 數量下降非常多，然而平均功率消耗和其他三者之間並無差出太多

```

void self (int&d) {

```

```
if (hearing!=old_h4) {  
    if((x==1500 && y<=1500) || (y==1500 && x<=1500) || (y>1500 &&  
x==1500) || (y==1500 && x>1500)){  
        Pold_4 = Plisten;  
        Pself = Plisten;  
        hearing = old_h4;  
    }  
    else {  
        Plisten = Pcur;  
        Pself = Plisten;  
        old_h4 = hearing;  
        d++;  
    }  
}
```