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

一張含有 螢幕擷取畫面 的圖片

自動產生的描述

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++;

}} }