**ÖZET**

**KABLOSUZ SENSÖR AĞLAR**

**YÖNLENDİRME PROTOKOLLERİNİN**

**OMNET ORTAMINDA SİMÜLASYONU**

Kablosuz sensör ağlarda karşılaşılan problemlerden birisi de sensörlerde bulunan kısıtlı enerji kaynaklarıdır. Genelde sensörlerin uzak ya da yerleşim yerlerinin dışında çalışmaları beklendiinden, pillerinin değiştirilmeleri ya da tekrar şarj edilmeleri mümkün olamamaktadır. Bu da enerjinin en verimli şekilde kullanılarak, sensör ağın ömrünün uzatılması gerekliliğini ortaya çıkarmaktadır. Bu amaçla kullanılacak birçok yöntem vardır.

Bu çalışmada, sensörlerin anten menzilleri kısıtlanıp, baza ulaşabilen sensörlerden başlayarak seviyelere ayrılan sensörlerin çalışmaları simüle edilecektir.

**Anahtar Kelimeler:** Kablosuz Sensör Ağlar, Sensör Ömrü, Kapsama Alanı, Seviye Tabanlı

**ABSTRACT**

**WİRELESS SENSOR NETWORK**

**ROUTİNG PROTOCOLS**

**SİMULATED ON OMNET**

One of the major problems of wireless sensor networks is limited energy resources of sensors. Due to the fact that sensors consume their energy and they are generally placed in remote or inhabited places, it is usually impossible to replace the battaries or recharge them. Thus it is crucial to increase the network lifetime by using the energy efficiently. There are several methods used for this purpose.

In this study, antenna was restricted range of the sensor, the sensor allocated to reach the base level starting from the sensors will be simulated.

**Keywords:** Wireless Sensor Network, Lifetime of Sensor, Covarage, Level Base

**İÇİNDEKİLER**

ÖZET.......................................................................................................................1

ABSTRACT............................................................................................................2

TEŞEKKÜR…........................................................................................................3

İÇİNDEKİLER.......................................................................................................4

ŞEKİLLER LİSTESİ..............................................................................................5

1.GİRİŞ...................................................................................................................6

2.YÖNLENDİRME PROTOKOLÜ.......................................................................6

2.1.ÇALIŞMA PRENSİBİ..........................................................................6

2.2.SEVİYELEME AŞAMALARI.............................................................7

3.PERFORMANS ANALİZİ..................................................................................9

KAYNAKLAR........................................................................................................12

EKLER……….........................................................................................................13

EK 1 “baz.cc” Dosyası…………………………………………………….13

EK 2 “sensor.cc” Dosyası………………………………………………….15

**ŞEKİLLER LİSTESİ**

Şekil 2.1. Rastgele Dağılmış Sensörler ve Merkezde Baz İstasyonu………………………….7

Şekil 2.2. Baz yayını ve ilk seviye sensörler…………………………………………………..7

Şekil 2.3. Baza ulaşabilen sensörlerin yayını………………………………………………….8

Şekil 2.4. Seviyeleme devam………………………………………………………………….8

Şekil 2.5. Seviyeleme Sonu……………………………………………………………………9

Şekil 3.1. Performans Grafikleri……………………………………………………………….9

Şekil 3.2. Performans Grafikleri……………………………………………………………...10

Şekil 3.3. Performans Grafikleri……………………………………………………………...10

Şekil 3.4. Performans Grafikleri……………………………………………………………...11

Şekil 3.5. Performans Grafikleri…………………………………………………...…………11

Şekil 3.6. Performans Grafikleri…………………………………………………………...…12

**1.GİRİŞ**

Kablosuz sensör ağlar (KSA) fiziksel bir olayı izleyen çok sayıda sensörden oluşurlar. Sensör düğümleri kısıtlı kaynaklara sahip olup veri işleme,veri toplama ve veri iletme görevlerine sahiptirler. Bulundukları ortamda kablosuz iletişim ile topladıkları verileri baz istasyonuna iletirler. Kablosuz sensör ağların yerleştirildiği ortamlarda herhangi bir altyapı bulunmamaktadır ve bir çok durumda ağda bulunan düğümlerin bakım onarım ve enerjilerini yenileme imkanları bulunmamaktadır. Bu yüzden sahip olunan enerjiyi verimli bir şekilde kullanmak kablosuz sensör ağların işlemesinde önemli bir yer tutar. Kablosuz sensör ağlar tasarlanırken hataya toleranslı, kendi kendine organize olabilen ve enerjisini ağ içinde dengeli bir şekilde kullanabilecek şekilde tasarlanmalıdır.

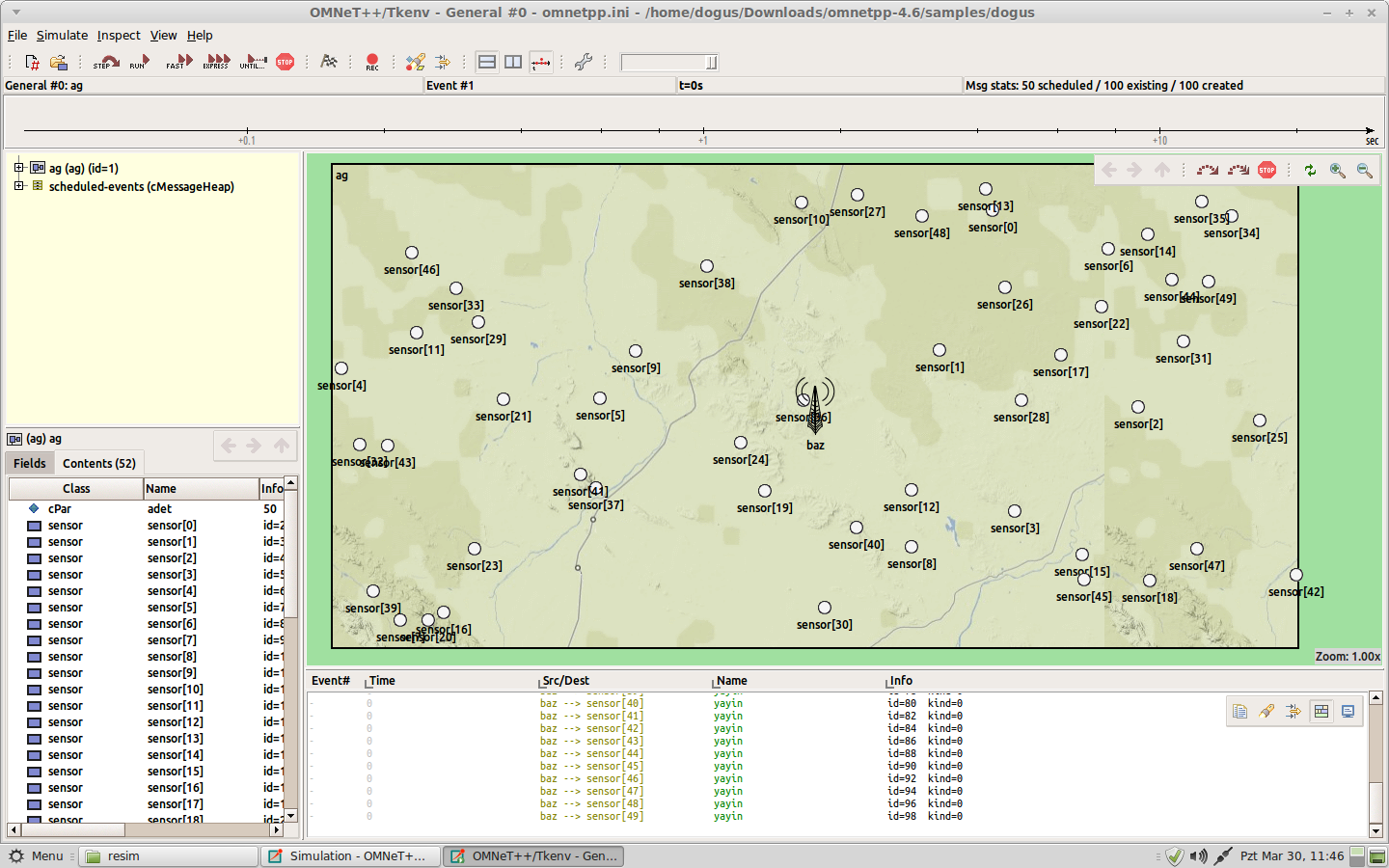
Bu bitirme çalışması kapsamında kablosuz sensör ağların yaşam sürelerinin en iyi seviyelere ulaşması amaçlanmıştır. Sensörlerin enerji tasarrufu söz konusu olduğunda en büyük düşman veri göndermedir. Veri gönderirken harcanan enerjiyi belirleyen faktörlerin başında alıcıyla aradaki uzaklık gelir. Bu nedenle, sensörlerin veri gönderimi direk olarak baza değil de kendi içlerinde çalışan protokol çerçevesinde, komşularının üzerinden verilerini göndermesi sağlanılarak enerji tasarrufu yapmak hedeflenmiştir.

**2. YÖNLENDİRME PROTOKOLÜ**

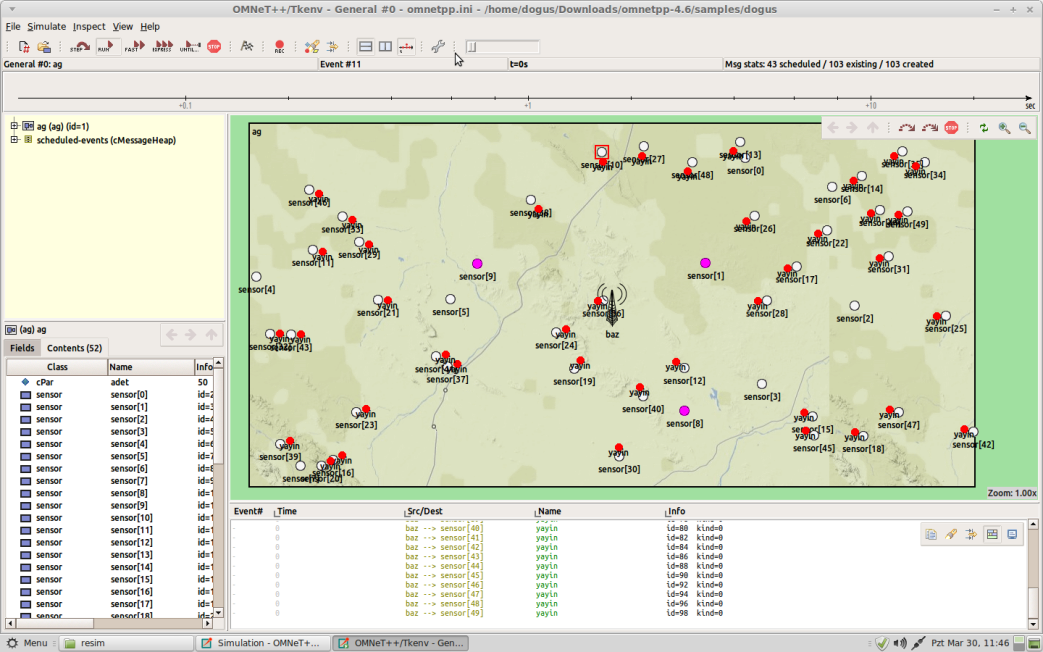
**2.1. Çalışma Prensibi**

İlk olarak baz yayın mesajı yayarak sensörlere kendi konumunu ve ulaşmaları gereken maksimum anten menzilini bildirir. Böylece sensörlerin anten menzilleri kısıtlanır, baza ulaşabilen sensörler kendilerine seviye numarası atayıp, konumunu ve seviyesini bildirmek için yayın mesajı yayarlar. Bu mesajı alan sensörler; seviyesi belli değilse aynı işlemi tekrarlarlar. Seviyeleme işlemi bu şekilde baza en uzak sensörlere kadar ilerler. Seviyeleme işlemi bittikten sonra her sensör sadece ulaşabildiği üst seviye komşusuna veri gönderir. Alt seviyeden veri alan sensörler de aldıkları veriyi üst seviye komşusuna iletir. Böylece sensörlerin uzak mesafelere veri göndermesi engellenir ve yaşam sürelerinde iyileşme sağlanır.

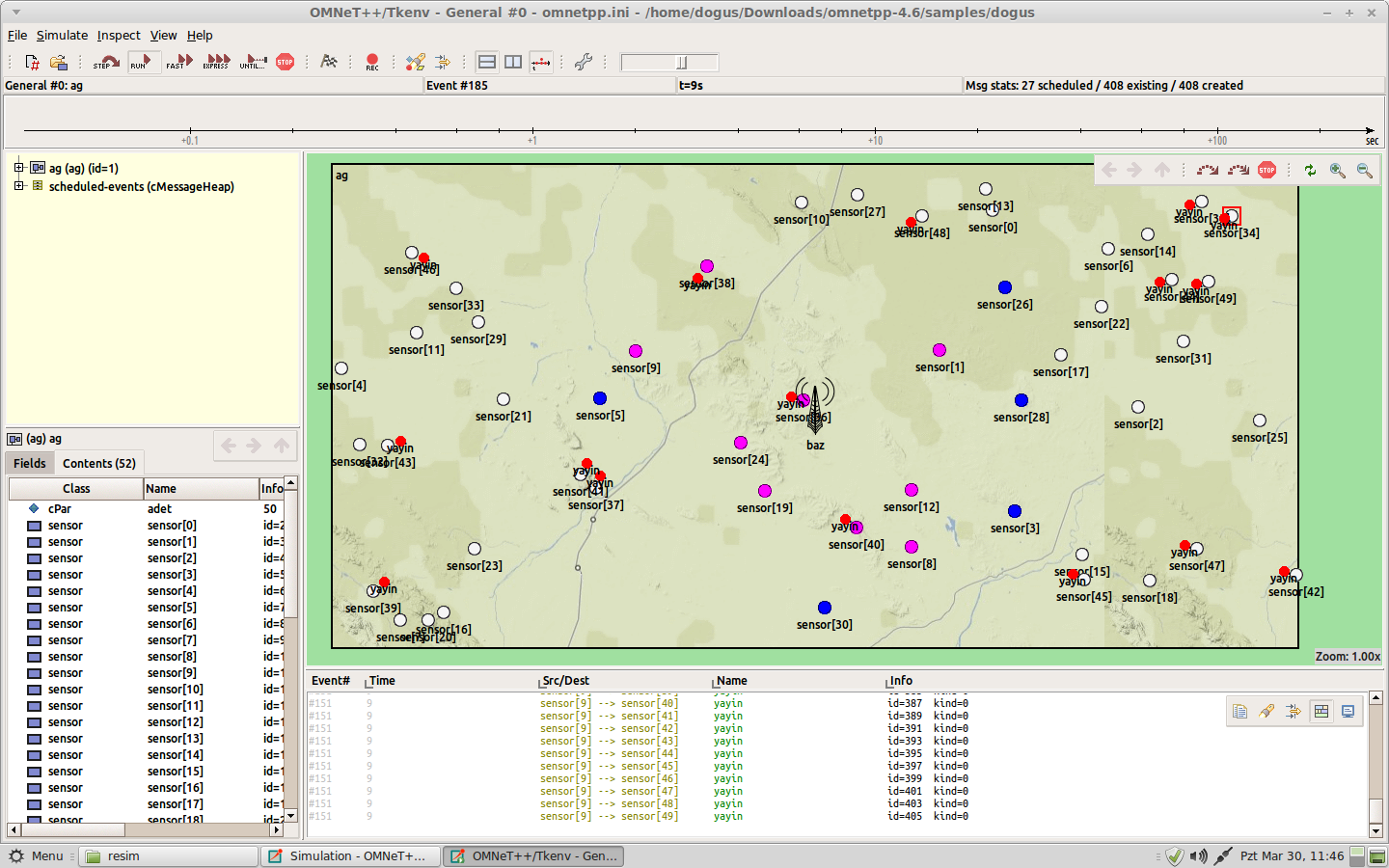
**2.2. Seviyeleme Aşamaları**



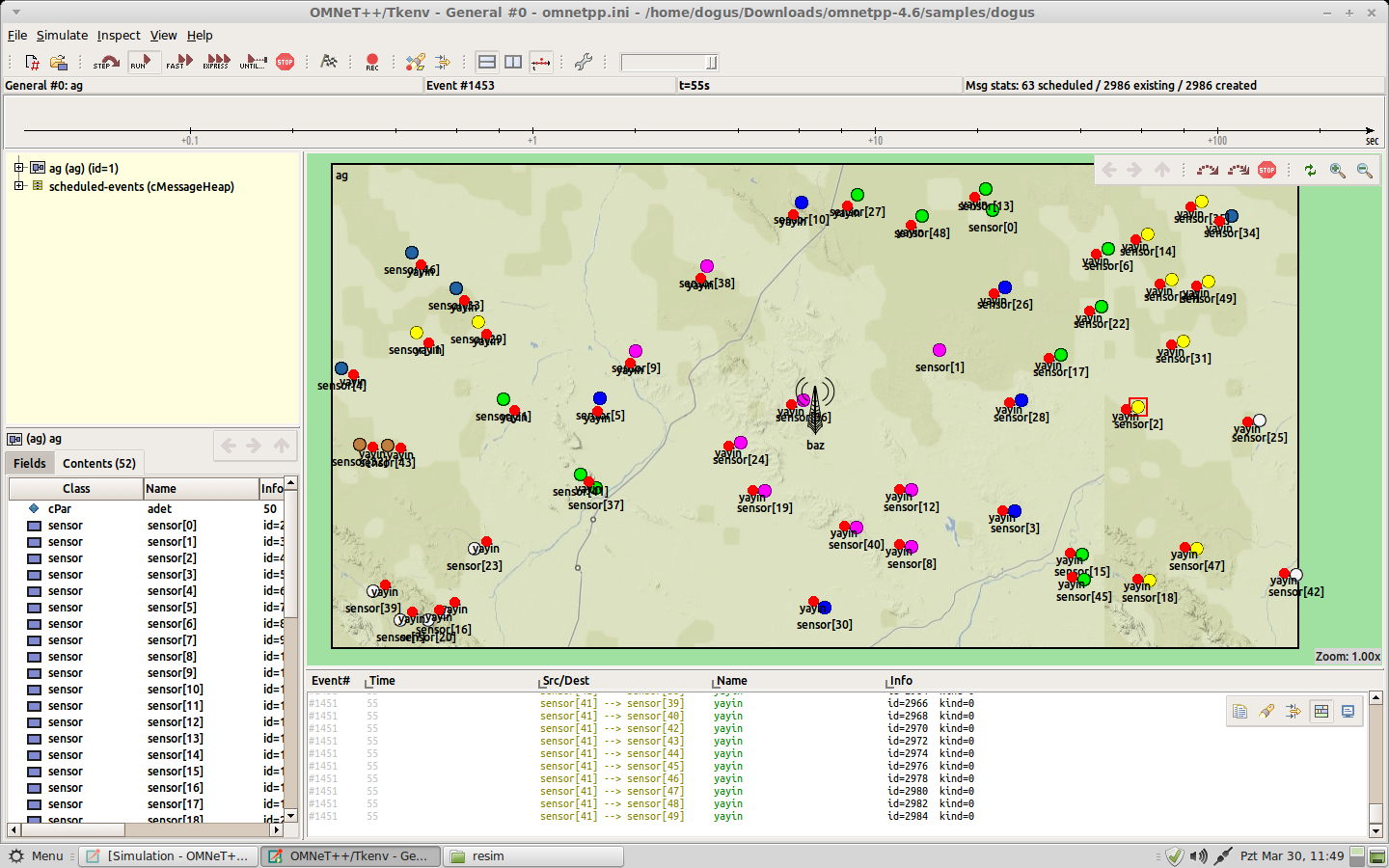
Şekil 2.1 Rastgele Dağılmış Sensörler ve Merkezde Baz İstasyonu



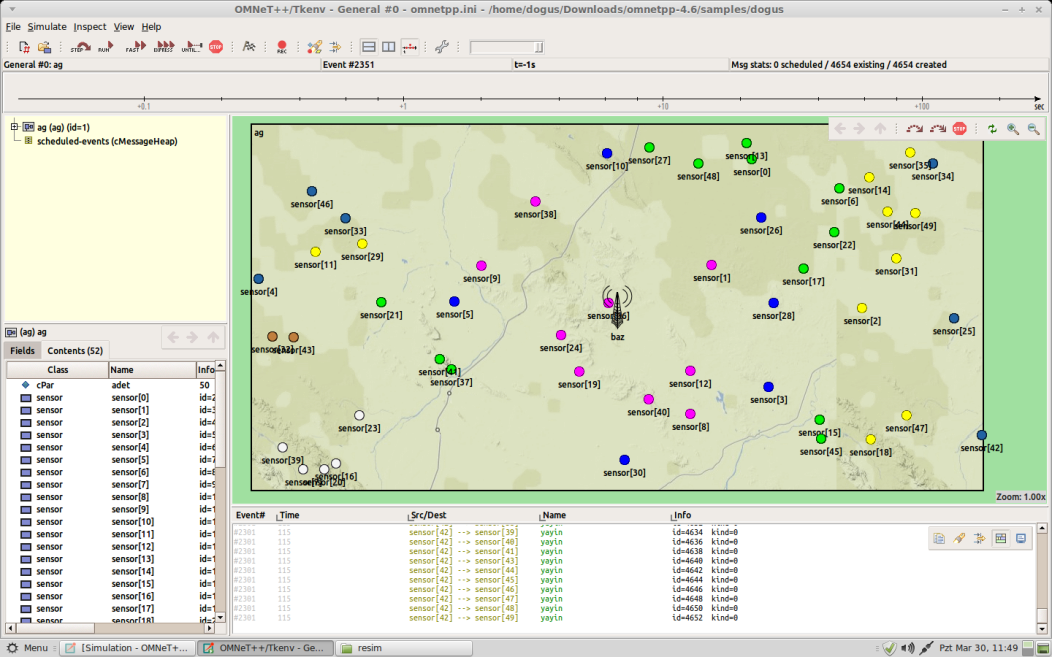
Şekil 2.2 Baz yayını ve ilk seviye sensörler



Şekil 2.3 Baza ulaşabilen sensörlerin yayını



Şekil 2.4 Seviyeleme devam

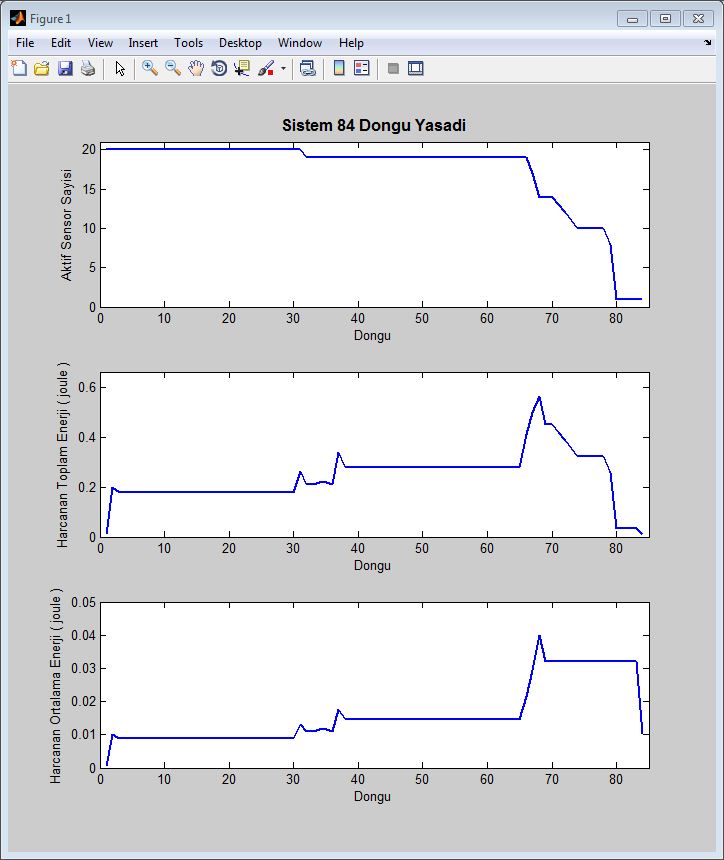


Şekil 2.5 Seviyeleme Sonu

**3. Performans Analizi**

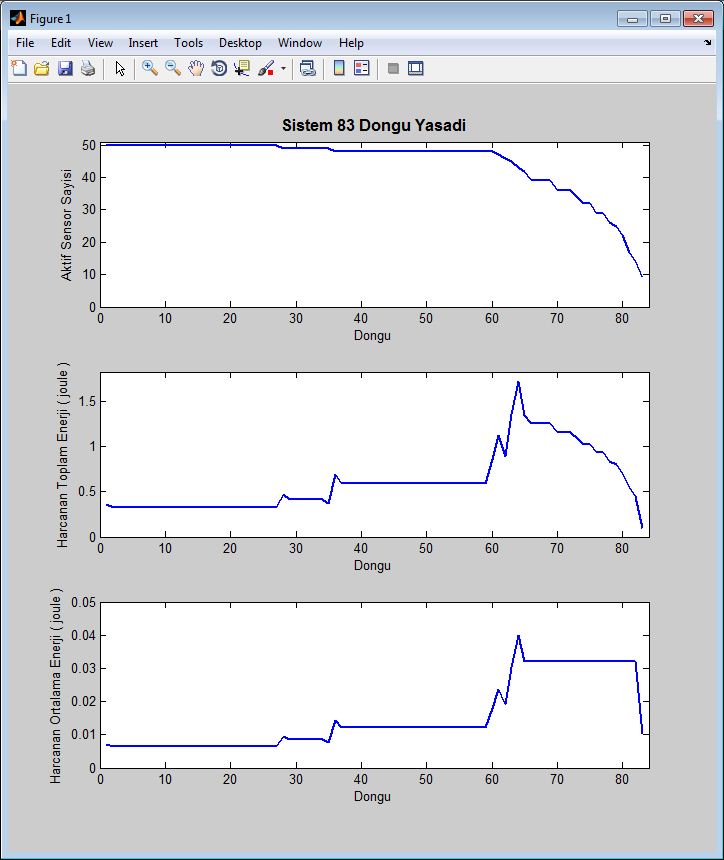
Sensörlerin başlangıç enerjileri 1 joule, veri mesajları 2000 bit, yayın mesajları 500 bit kabul edilerek analiz yapılmıştır.

**3.1. 500 x 500 alanda 20 sensör**



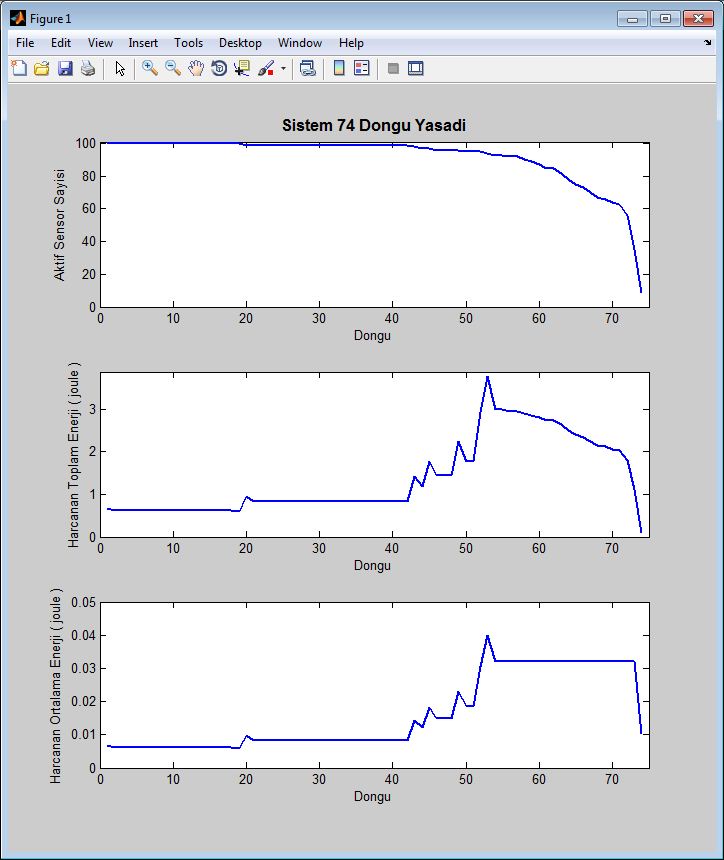
Şekil 3.1 Performans Grafikleri

**3.2. 500 x 500 alanda 50 sensör**



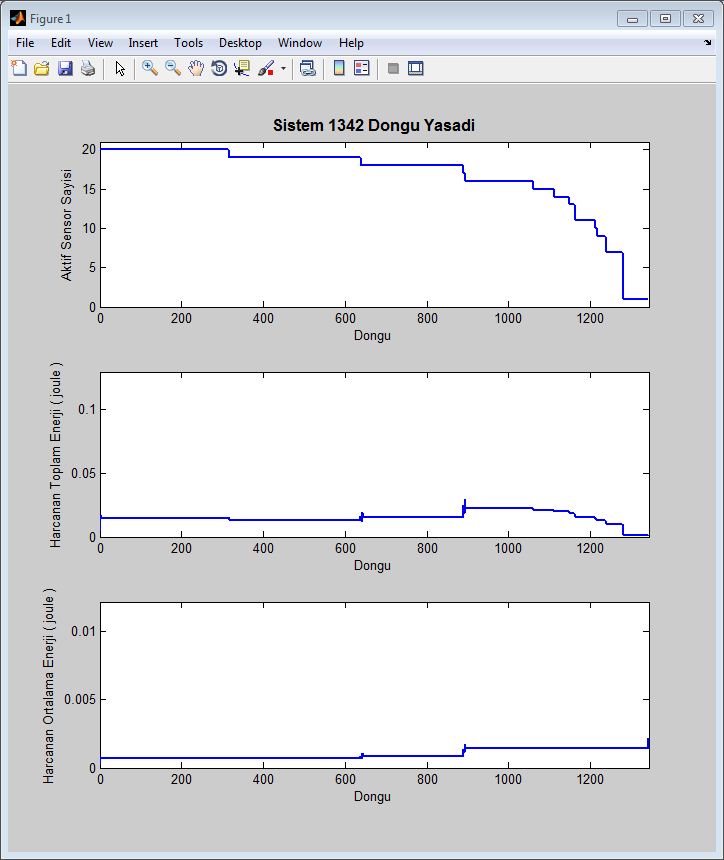
Şekil 3.2 Performans Grafikleri

**3.3. 500 x 500 alanda 100 sensör**



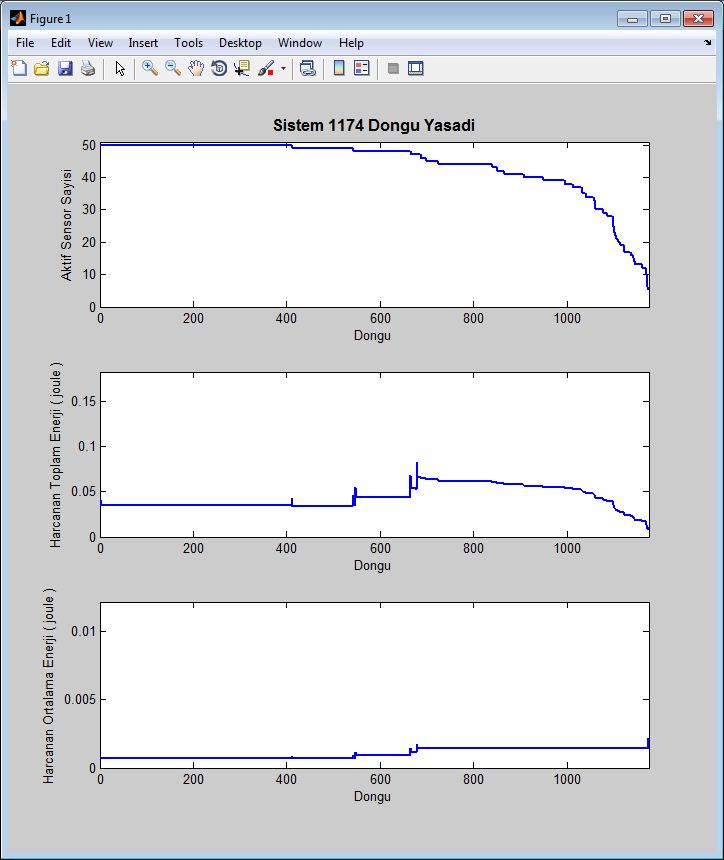
Şekil 3.3 Performans Grafikleri

**3.4. 100 x 100 alanda 20 sensör**



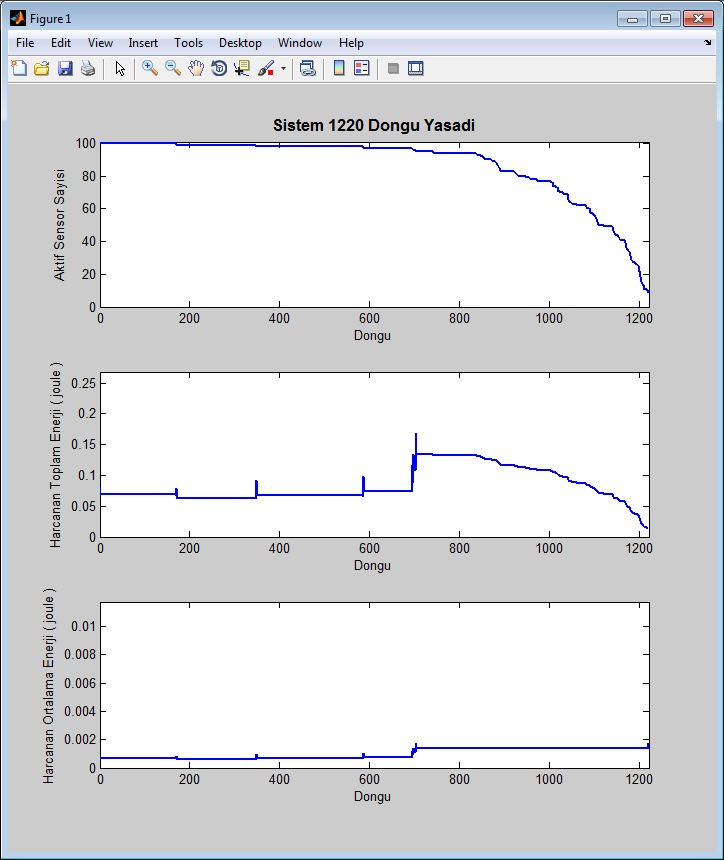
Şekil 3.4 Performans Grafikleri

**3.5. 100 x 100 alanda 50 sensör**



Şekil 3.5 Performans Grafikleri

**3.6. 100 x 100 alanda 100 sensör**



Şekil 3.6 Performans Grafikleri

**KAYNAKLAR**

1. OMNET++ Simulation Library, http://www.omnetpp.org/doc/omnetpp/api/main.html

**EK 1 “baz.cc” Dosyası**

#include <stdio.h>

#include <string.h>

#include <omnetpp.h>

#include <iostream>

#include <fstream>

#include <cmath>

class baz : public cSimpleModule

{

public:

int snsr\_say;//dongu baslangicindaki sensor sayisi

int aktif=0;//dongu sonunda hala aktif sensor sayisi

double anten\_menzil=100;//sensorlere gonderilecek

double anten\_arttir=50;//olum durumunda artirilacak menzil

double uzak;//en uzak sensore olan uzaklik, yani en buyuk menzil degeri

int dongu=1;

virtual void initialize();

virtual void handleMessage(cMessage \*msg);

void yayin\_yap(int a,char\* b);

void kendime\_mesaj(simtime\_t zaman);

int rapor\_aktif=0;

double rapor\_toplam\_enerji=0;

};

Define\_Module(baz);

void baz::initialize()

{

snsr\_say=par("adet").longValue();//ned dosyasindan parametere olarak cektik

uzak=par("boyut").doubleValue()\*1.5;//en uzak sensor, boyut\*kok2 uzaklikta olabilir

yayin\_yap(anten\_menzil,(char \*)"agkur");//ilk dongu

}

void baz::handleMessage(cMessage \*msg)

{

char \*tip=(char \*)msg->par("tip").stringValue();//gelen mesajin tipi

if(strcmp(tip,"veri")==0)//veri geldiyse

{

if(msg->getParList().exist("batarya"))//mesajda batarya bilgisi var mi?

{

double batarya=msg->par("batarya").doubleValue();

if(batarya<=((50e-9\*4500)+(100e-12\*2000\*pow(anten\_menzil,2))))//sensor olmek uzeredir

{

aktif--;

}

}

aktif++;

EV<<"sensor["<<msg->par("id").longValue()<<"] verisi : "<<msg->par("sicaklik").longValue()<<"°C"<<endl;

//ekrana yazdir, sensor[0]=42°C

}

else if(strcmp(tip,"rapor")==0)

{

rapor\_aktif++;

rapor\_toplam\_enerji+=msg->par("harcadigim").doubleValue();

}

else if(strcmp(tip,"kendime")==0)//dongu bitmistir

{

std::ofstream dosya;

dosya.open ("aktif\_sensor\_sayisi.txt",std::ofstream::app);

dosya << rapor\_aktif<<endl;

dosya.close();

dosya.open("harcanan\_enerji.txt",std::ofstream::app);

dosya <<rapor\_toplam\_enerji<<endl;

dosya.close();

rapor\_aktif=0;

rapor\_toplam\_enerji=0;

if(anten\_menzil>uzak && aktif>0)//butun sensorler baza ulasabiliyorsa,yeniden seviyelemeye gerek yok

{

snsr\_say=aktif;

aktif=0;

yayin\_yap(anten\_menzil,(char \*)"devam");//yeniden seviyelenme yapmadan devam edin kocum

}

else if(aktif<snsr\_say)//olmek uzere yada olu varsa

{

snsr\_say=aktif;

aktif=0;

anten\_menzil+=anten\_arttir;

yayin\_yap(anten\_menzil,(char \*)"agkur");//yeniden seviyelenme icin yayin yap

}

else if(aktif>0)//butun sensorler olene kadar

{

snsr\_say=aktif;

aktif=0;

yayin\_yap(anten\_menzil,(char \*)"devam");//yeniden seviyelenme yapmadan devam edin kocum

}

}

}

void baz::yayin\_yap(int a,char\* b)

{

if (strcmp(b,"agkur")==0)

{

for (int i=0;i<(gateCount())/2;i++)

{

bubble("DONGU BASLIYOR");

double x=atoi(getDisplayString().getTagArg("p",0));//similasyon goruntu ayarlarından konum al

double y=atoi(getDisplayString().getTagArg("p",1));

cMessage \*msg=new cMessage("yayin");

msg->addPar("tip");

msg->par("tip")="yayin";

msg->addPar("x");//mesaja konum koy

msg->par("x")=x;

msg->addPar("y");

msg->par("y")=y;

msg->addPar("anten\_menzil");//mesaja anten menzilini koy

msg->par("anten\_menzil")=a;

send(msg,"cikis",i);

}

}

else if(strcmp(b,"devam")==0)

{

for (int i=0;i<(gateCount())/2;i++)

{

bubble("DONGU BASLIYOR");

cMessage \*msg=new cMessage("yayin");

msg->addPar("tip");

msg->par("tip")="devam";

send(msg,"cikis",i);

}

}

kendime\_mesaj(dongu\*1000);//dongu sonunda tetiklenmek icin; her dongu 1000 sn periyod

dongu++;

}

void baz::kendime\_mesaj(simtime\_t zaman)

{

cMessage \*msg=new cMessage("kendime");//tetikleme mesaji yaratip kendime gonderecem

msg->addPar("tip");

msg->par("tip")="kendime";

scheduleAt(zaman,msg);

}

**EK 2 “sensor.cc” Dosyası**

#include <stdio.h>

#include <string.h>

#include <omnetpp.h>

#include <cmath>//matematik

class sensor : public cSimpleModule

{

public:

int seviye=0;

int komsu[100];//mesaj yollayacagim ust seviyelerin tutuldugu dizi

int kmsu\_say;//onceki satirdaki diziye indis olarak kullanilacak

double anten\_menzil;//baz gonderecek

int dongu=-1;

double batarya=1.0;

double konum\_x;

double konum\_y;

bool hayat=true;

double harcadigim=0;

virtual void initialize();

virtual void handleMessage(cMessage \*msg);

double uzaklik\_hesapla(double x1,double y1,double x2,double y2);//iki nokta arasi oklid uzakligi dondurur

void yayin\_yap();

void veri\_yolla();

void kendime\_mesaj(char\* ne\_icin,simtime\_t zaman);

char\* renklendir(int a);//parametre olarak seviye alip renk stringi dondurur

void rapor\_yolla();//ag hakkinda raporlama icin;log dosyasi

};

Define\_Module(sensor);

void sensor::initialize()

{

int boyut=par("boyut").longValue();

int a=intuniform(0,boyut);//0-boyut arasi random integer, konum icin

int b=intuniform(0,boyut);

getDisplayString().setTagArg("p",0,a);//gorunum ayarlarina konumu yaz

getDisplayString().setTagArg("p",1,b);

konum\_x=(double)a;//konumumu kaydettim

konum\_y=(double)b;

EV<<"sensor"<<idx<<" : "<<konum\_x<<","<<konum\_y<<endl;//"sensor[0] : 300,250" seklinde konumumu ekrana yazdim

}

void sensor::handleMessage(cMessage \*msg)

{

if (hayat)//hayatta miyim?

{

char \*tip=(char \*)msg->par("tip").stringValue();//gelen mesajin tipi

if(strcmp(msg->getSenderModule()->getName(),"baz")==0)//mesaj bazdan geldiyse

{

dongu++;

if(strcmp(tip,"yayin")==0)//baz seviyelenme istiyor

{

double baz\_x=msg->par("x").doubleValue();//bazin konumunu aldim

double baz\_y=msg->par("y").doubleValue();

anten\_menzil=msg->par("anten\_menzil").doubleValue();//anten menzilini atadim

kmsu\_say=0;//iki ve sonraki seviyeleme dongulerinde eski komsulari unutmak icin

seviye=0;

if(uzaklik\_hesapla(baz\_x,baz\_y,konum\_x,konum\_y)<=anten\_menzil)//baza ulasabiliyorsam

{

seviye=1;

getDisplayString().setTagArg("i",1,renklendir(seviye));//similasyonda sensoru boyadim

komsu[kmsu\_say]=idx;//idx kapisi baza ulasiyor, bazi komsum olarak ekledim

kmsu\_say++;//indisi arttirdim

kendime\_mesaj((char \*)"yayin\_yapak",0);//yayin yapmak icin tetikleme mesaji yaratiyorum

}

}

kendime\_mesaj((char\*)"veri\_gonderek",800);//veri gondermek icin tetikleme mesaji yaratiyorum

kendime\_mesaj((char\*)"rapor\_gonderek",900);

batarya-=(50e-9\*500);

harcadigim+=(50e-9\*500);

}

else//mesaj bazdan gelmediyse

{

if(strcmp(tip,"yayin")==0)//yayin mesajiysa

{

if(seviye==0 || msg->par("seviye").longValue()+1<seviye)//seviyem belli degil veya seviyemi dusurme ihtimalim varsa

{

double komsu\_x=msg->par("x");//mesaji gonderinin konumunu aldim

double komsu\_y=msg->par("y");

if(uzaklik\_hesapla(komsu\_x,komsu\_y,konum\_x,konum\_y)<=anten\_menzil)//gonderene ulasabiliyormuyum

{

kmsu\_say=0;

komsu[kmsu\_say]=msg->par("id").longValue();//gondereni komsum olarak ekledim

kmsu\_say++;//indisi arttirdim

seviye=msg->par("seviye").longValue();

seviye++;//seviyemi belirledim

getDisplayString().setTagArg("i",1,renklendir(seviye));//similasyonunda sensoru boyadim

kendime\_mesaj((char \*)"yayin\_yapak",(seviye-1)\*100);//yayin yapmak icin tetikleme mesaji yaratiyorum

batarya-=(50e-9\*500);

harcadigim+=(50e-9\*500);

}

}

else if(msg->par("seviye").longValue()<seviye)//mesaj bir alt seviyeden geldiyse

{

double komsu\_x=msg->par("x");

double komsu\_y=msg->par("y");

if(uzaklik\_hesapla(komsu\_x,komsu\_y,konum\_x,konum\_y)<=anten\_menzil)//ulasabiliyorsam

{

komsu[kmsu\_say]=msg->par("id");//gondereni komsum olarak ekledim

kmsu\_say++;

batarya-=(50e-9\*500);

harcadigim+=(50e-9\*500);

}

}

}

else if(strcmp(tip,"kendime")==0)//kendime tetikleme mesaji attiysam

{

if(kmsu\_say>0)//kmsu\_say=0 sa,anten seviyesi kimseye ulasmama yetmiyor

{

if (strcmp(msg->par("napak").stringValue(),"yayin\_yapak")==0)//yayin tetiklemesi

{

batarya-=((50e-9\*500)+(100e-12\*500\*pow(anten\_menzil,2)));

harcadigim+=((50e-9\*500)+(100e-12\*500\*pow(anten\_menzil,2)));

yayin\_yap();

}

else if(strcmp(msg->par("napak").stringValue(),"veri\_gonderek")==0)//veri tetiklemesi

{

batarya-=((50e-9\*2000)+(100e-12\*2000\*pow(anten\_menzil,2)));

harcadigim+=((50e-9\*2000)+(100e-12\*2000\*pow(anten\_menzil,2)));

veri\_yolla();

}

}

if(strcmp(msg->par("napak").stringValue(),"rapor\_gonderek")==0)//rapor tetiklemesi

{

rapor\_yolla();

harcadigim=0;

}

}

else if(strcmp(tip,"veri")==0)//sensor verisi geldiyse

{

batarya-=(50e-9\*2000);//mesaji aldigim icin

batarya-=((50e-9\*2000)+(100e-12\*2000\*pow(anten\_menzil,2)));//mesaji ilettigim icin

harcadigim+=(50e-9\*2000);

harcadigim+=((50e-9\*2000)+(100e-12\*2000\*pow(anten\_menzil,2)));

msg->addPar("batarya");//mesaja bataryami da ekliyeyim

msg->par("batarya")=batarya;

send(msg,"cikis",komsu[rand()%kmsu\_say]);//ust seviyeden rasgele komsuma ilettim

}

}

if(batarya<=0) //batarya bittiyse

{

hayat=false;

bubble("Ben bittim la!");

getDisplayString().setTagArg("i",1,"black");

}

}

else if(harcadigim>0)//olmeden once enerji harcadiysam

{

rapor\_yolla();

harcadigim=0;

}

}

double sensor::uzaklik\_hesapla(double x1,double y1,double x2,double y2)

{

double a=sqrt(pow((x1-x2),2)+pow((y1-y2),2));//oklid uzaklik hesapla

return a;

}

void sensor::yayin\_yap()

{

for(int i=0;i<gateCount()/2;i++)

{

if(i!=idx)//idx kapisi baza gidiyor

{

cMessage \*msg=new cMessage("yayin");

msg->addPar("tip");

msg->par("tip")="yayin";

msg->addPar("seviye");

msg->par("seviye")=seviye;

msg->addPar("id");

msg->par("id")=idx;

msg->addPar("x");

msg->par("x")=konum\_x;

msg->addPar("y");

msg->par("y")=konum\_y;

send(msg,"cikis",i);

}

}

}

void sensor::veri\_yolla()

{

cMessage \*msg=new cMessage("veri");

msg->addPar("tip");

msg->par("tip")="veri";

msg->addPar("id");

msg->par("id")=idx;

msg->addPar("sicaklik");

msg->par("sicaklik")=rand()%50;//rasgele sicaklik verisi olusturdum

if(seviye==1)//seviye 1 se

{

msg->addPar("batarya");//mesaja bataryami da ekliyeyim

msg->par("batarya")=batarya;

}

send(msg,"cikis",komsu[rand()%kmsu\_say]);//alt seviyeden rasgele komsuya yollicam

}

void sensor::kendime\_mesaj(char\* ne\_icin,simtime\_t zaman)

{

cMessage \*msg=new cMessage("kendime");//tetikleme mesaji yaratip kendime gonderecem

msg->addPar("tip");

msg->par("tip")="kendime";

msg->addPar("napak");

msg->par("napak")=ne\_icin;

scheduleAt((dongu\*1000)+zaman+idx,msg);//similasyonda gorsel cakisma olmamasi icin,zamanlama ayarladim

}

char\* sensor::renklendir(int a)

{

switch(a)

{

case 1:return (char\*)"magenta";

case 2:return (char\*)"blue";

case 3:return (char\*)"green";

case 4:return (char\*)"yellow";

case 5:return (char\*)"#123456";

case 6:return (char\*)"#654321";

case 7:return (char\*)"#194c03";

case 8:return (char\*)"#b27007";

case 9:return (char\*)"#9cdb08";

case 10:return (char\*)"#086fc9";

default:return (char\*)"#ffffff";

}

}

void sensor::rapor\_yolla()

{

cMessage \*msg=new cMessage("rapor");//tetikleme mesaji yaratip kendime gonderecem

msg->addPar("tip");

msg->par("tip")="rapor";

msg->addPar("harcadigim");

msg->par("harcadigim")=harcadigim;

send(msg,"cikis",idx);//baza raporu yolladim

}