update(Unit\_type,Unit\_data){

update local\_ets → Unit\_type, Unit\_data

}

hili requset (Sensor\_name,Fire\_name){ / /\_from\_ Sensor

if (first alert to {Sensor\_name,Fire\_name})

if (there is free heli)

Heli=free heli

else

req heli from next server

block receive for answer

\* no heli or timeout → req heli from next server

\* get heli → Heli=received heli

if not find heli in all servers → Heli=none

if(Heli!=none)

add {Sensor\_name,Fire\_name} to DB

Heli.status=working

Heli ! goDist(Sensor\_location,Radios)

}

hili requset (){ ///\_from\_server

if (there is free heli)

return Heli\_name

else

return none

}

heli\_done(Heli\_name){

update local\_ets → {Heli\_name,done}

}

heli\_change\_screen( Heli\_name, Loction,State,State\_data){

remove Heli from local\_ets

Node\_name = f(Loction)

Node\_name ! addHeli( Heli\_name, Loction,State,State\_data)

}

addHeli(Heli\_name, Loction,State,State\_data){

update local\_ets → { Heli\_name, Loction,State,State\_data}

create Heli( Heli\_name, Loction)

start\_heli( Heli\_name,State,State\_data)

}

check\_alerts(Sensor\_name){ //from sensor

FireList= send to all servers one by one(block) ! check\_alerts(Sensor\_loction,Sensor\_radios)

Run trow all fire in local\_ets and add fires in Sensor\_loction to FireList

return FIreList

}

check\_alerts(Sensor\_loction,Sensor\_radios){ //from server

Run trow all fire in local\_ets and add fires in Sensor\_loction to FireList

return FIreList

}

send\_data(){ //from graphical

return local\_ets to list

}