Appendix

```
function varargout =lightweight synchronization scheme(varargin)
gui_Singleton = 1;

gui_State = struct ('gui_Name',       mfilename, ...
                    'gui_Singleton', gui_Singleton, ...
                    'gui_OpeningFcn', @test1_OpeningFcn, ...
                    'gui_OutputFcn', @test1_OutputFcn,...
                    'gui_LayoutFcn',  [] , ...
                    'gui_Callback',   []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
function lightweight synchronization scheme _OpeningFcn(hObject,
eventdata, handles, varargin)
handles.output = hObject;
 guidata(hObject, handles);
 function varargout = test1_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;
 function pushbutton1_Callback(hObject, eventdata, handles)
 function popupmenu1_Callback(hObject, eventdata, handles)
function popupmenu1_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
 function pushbutton1_ButtonDownFcn(hObject, eventdata, handles)
function pushbutton1_CreateFcn(hObject, eventdata, handles)
function pushbutton3_Callback(hObject, eventdata, handles)

**********************************************************
* %code1:creating nodes :                                 *
**********************************************************
function pushbutton5_Callback(hObject, eventdata, handles)
clear;
clf;
clc;
%-------Parameters-------------------
a=get(handles.popupmenu2,'value');
Number_Of_Nodes=a;
Number_Of_Nodes =100;
L = 200;
R = 30; % maximum range;
%-------Creating nodes-------------------
 netXloc = rand(1,Number_Of_Nodes)*L;
 netYloc = rand(1,Number_Of_Nodes)*L;
nPlaces = 1;
for k =1:Number_Of_Nodes
    mew(k) = rand (1,1);
    mew2(k) = roundTo(mew(k),nPlaces);
end
CH=0;
number_of_clusters=0;
```

```
    for i = 1:Number_Of_Nodes
        plot(netXloc(i), netYloc(i), 'o');axis square;
   hold on;
        for j = 1:Number_Of_Nodes
            distance (i,j)= sqrt((netXloc(i) - netXloc(j))^2 +
(netYloc(i) - netYloc(j))^2);
        end
    end
hold on
nPlaces = 1;
for k = 1:Number_Of_Nodes
    mew(k) = rand (1,1);
    mew2(k) = roundTo(mew(k),nPlaces);
end
hold on

function pushbutton4_Callback(hObject, eventdata, handles)
********************************************************* *
%code2: cluster head selection:                            *
*********************************************************
clear;
clf;
clc;
%-------Parameters-------------------
a=get(handles.popupmenu2,'value');
Number_Of_Nodes=a;
Number_Of_Nodes =100;
L = 200;
R = 30; % maximum range;
%-------Creating nodes-------------------
```

```
 netXloc = rand(1,Number_Of_Nodes)*L;
 netYloc = rand(1,Number_Of_Nodes)*L;
nPlaces = 1;
for k =1:Number_Of_Nodes
    mew(k) = rand (1,1);
    mew2(k) = roundTo(mew(k),nPlaces);
end
CH=0;
number_of_clusters=0;

    for i = 1:Number_Of_Nodes
        plot(netXloc(i), netYloc(i), 'o');axis square;
        hold on;
        for j = 1:Number_Of_Nodes
            distance (i,j)= sqrt((netXloc(i) - netXloc(j))^2 +
(netYloc(i) - netYloc(j))^2);
        end
    end
hold on
nPlaces = 1;
for k = 1:Number_Of_Nodes
    mew(k) = rand (1,1);
    mew2(k) = roundTo(mew(k),nPlaces);
end
hold on
%-------Cluster head selection--------------------
counter_for_all_loops=0;
counter_for_node1_comparison=0;
for i = 1:Number_Of_Nodes
    S_time = mew2(i);
    for j = 1:Number_Of_Nodes
        counter_for_all_loops=counter_for_all_loops+1;
```

```
        if (i==1) || (j==1)
            counter_for_node1_comparison=counter_for_node1_compa
rison+1;
        end
        if (distance (i,j)<=R)
            if (mew2(j)<S_time)
                S_time = mew2(j); %smallest time
            end
        end
    end
    if mew2(i)<=S_time
        mew3(i)= 1;
    else
        mew3(i)= 0;
    end
end
%-------Cluster Head counting--------------------
count_CH=0;
for m = 1:Number_Of_Nodes
    if (mew3(m)== 1)
        count_CH = count_CH +1;
        CH(count_CH)=m;
        plot(netXloc(m),netYloc(m),'r^','MarkerFaceColor','b','M
ark',10)
        t = linspace(0,2*pi,1000);
h=netXloc(m);
k=netYloc(m);
r=R;
x = r*cos(t)+h;
y = r*sin(t)+k;

    end
```

```
end

for i=1:count_CH
    for j=1:Number_Of_Nodes
        if distance(CH(i),j)<=R
          line([netXloc(CH(i)) netXloc(j)],
[netYloc(CH(i))netYloc(j)],'LineStyle', ':','Color',[.5 .5 .5]);
        end
    end
end

**********************************************************************
* %code3:clusterformation:                                           *
**********************************************************************
clear;
clf;
clc;
%-------Parameters-------------------
a=get(handles.popupmenu2,'value');
Number_Of_Nodes=a;
Number_Of_Nodes =100;
L = 200;
R = 30; % maximum range;
Hop = 25;

%-------Creating nodes-------------------
 netXloc = rand(1,Number_Of_Nodes)*L;
 netYloc = rand(1,Number_Of_Nodes)*L;
nPlaces = 1;
for k =1:Number_Of_Nodes
    mew(k) = rand (1,1);
    mew2(k) = roundTo(mew(k),nPlaces);
```

```
end
CH=0;
number_of_clusters=0;

    for i = 1:Number_Of_Nodes
        plot(netXloc(i), netYloc(i), 'o');axis square;
        hold on;
        for j = 1:Number_Of_Nodes
            distance (i,j)= sqrt((netXloc(i) - netXloc(j))^2 +
(netYloc(i) - netYloc(j))^2);
        end
    end
hold on
nPlaces = 1;
for k = 1:Number_Of_Nodes
    mew(k) = rand (1,1);
    mew2(k) = roundTo(mew(k),nPlaces);
end
hold on
%-------Cluster head selection--------------------
counter_for_all_loops=0;
counter_for_node1_comparison=0;
for i = 1:Number_Of_Nodes
    S_time = mew2(i);
    for j = 1:Number_Of_Nodes
        counter_for_all_loops=counter_for_all_loops+1;
        if (i==1) || (j==1)
    counter_for_node1_comparison=counter_for_node1_comparison+1;
        end
        if (distance (i,j)<=R)
            if (mew2(j)<S_time)
                S_time = mew2(j); %smallest time
```

```
            end
        end
    end
    if mew2(i)<=S_time
        mew3(i)= 1;
    else
        mew3(i)= 0;
    end
end
%-------Cluster Head counting--------------------
count_CH=0;
for m = 1:Number_Of_Nodes
    if (mew3(m)== 1)
        count_CH = count_CH +1;
        CH(count_CH)=m;
        plot(netXloc(m),netYloc(m),'r^','MarkerFaceColor','b','M
arkerSize',10)
        t = linspace(0,2*pi,1000);
h=netXloc(m);
k=netYloc(m);
r=R;
x = r*cos(t)+h;
y = r*sin(t)+k;
plot(x,y),axis square; %circle
    end
end

for i=1:count_CH
    for j=1:Number_Of_Nodes
        for z=2:count_CH
        if distance(CH(i),j)<=R
```

```
            line([netXloc(CH(i)) netXloc(j)], [netYloc(CH(i))
netYloc(j)],'LineStyle', ':','Color',[.5 .5 .5]);
           if distance(CH(i),CH(z))>=R
       if distance(CH(i),j)<=hop
     line([netXloc(CH(i)) netXloc(j)], [netYloc(CH(i))
netYloc(j)],'LineStyle', ':','Color',[1 .4 .6]);


       end

           end

                end

                    end
end
 count_CH

function pushbutton6_Callback(hObject, eventdata, handles)
**************************************************
* %intra offset:                                  *
**************************************************
clear all;
close all;
a=get(handles.popupmenu2,'value');
n=a;
n=25;
myhops=[10 15 20 30];
Mmx=25;
for conthops=1:4
hops=myhops(conthops)/1000;
```

```
for M=1:Mmx
Db=hops*randn(n,1);
Da=hops*randn(n,1);
Dc=hops*randn(n,1);
Thba=Db+Da;
Thbc=Da;
Thbd=Da+Db+Dc;
Ac(conthops,M)=sum(abs(Thba)+abs(Thbc)+abs(Thbd))/(n*M);
end
end
n=1:Mmx;
Ac=Ac*60/(1000*60*9*1e-6);
plot(n,Ac(1,:),'o-',n,Ac(2,:),'*-',n,Ac(3,:),'v-',n,Ac(4,:),'^')

legend('Number of hops=10 m','Number of hops=15 m','Number of
hops=20 m','Number of hops=30 m');

Title('Intra Offset')
xlabel('Number of Synchronization Processes')
ylabel('Average clock Offset us')


Ac=Ac*60/(1000*60*0.8*1e-6);
%plot(n,Ac2(1,:),'m-*',n,Ac2(2,:),'r-*',n,Ac2(3,:),'b-v',n,Ac2(4,:),'y-^')

%legend('Ms-assisted(Number of hops=10m)','Ms-assisted(Number of
hops=15m)','Ms-assisted(Number of hops=20m)','Ms-assisted(Number of
hops=30m)','Intra-cluster(Number of hops=10m)','Intra-cluster(Number of
hops=15m)','Intra-cluster(Number of hops=20m)','Intra-cluster(Number of
hops=30m)');



function pushbutton7_Callback(hObject, eventdata, handles)
**************************************************
* %intra skew :                                  *
```

```
**************************************************
%for calculating the skew or clock drift
%the num of nodes N must be the num of sync proccess Mmx
%you can take higher value of n e.g. n=1000
%pulse period=11us(micro sec)

clear all;
close all;
n=100;
N=get(handles.popupmenu2,'value');
N=25;
myhops=[10 15 20 30];
Mmx=25;
for conthops=1:4
hops=myhops(conyhops)/1000;
for M=1:Mmx
for node=1:N
Da=hops*randn(n,1);
Thbc=Da;
Ac(conthops,M)=sum(N.*(abs(Thbc)))/(n*M);
end
end
end
 a=1:N;
Ac=Ac*60/(10000*60*9*1e-6);
plot(a,Ac(1,:),'o-',a,Ac(2,:),'*-',a,Ac(3,:),'v-',a,Ac(4,:),'^')
legend('Number of hops=10 m','Number of hops=15 m','Number of
hops=20 m','Number of hops=30 m');

Title('Intra skew')
xlabel('Number of Synchronization Process')
ylabel('Average skew (ppm)'
```

```
%Ac1=Ac*60/(10000*60*3.5*1e-6);
%plot(a,Ac2(1,:),'m-*',a,Ac2(2,:),'r-*',a,Ac2(3,:),'b-
v',a,Ac2(4,:),'y-^')

%legend('Ms-assisted(Number of hops=10m)','Ms-assisted(Number of
hops=15m)','Ms-assisted(Number of hops=20m)','Ms-assisted(Number
of hops=30m)','Intra-cluster(Number of hops=10m)','Intra-
cluster(Number of hops=15m)','Intra-cluster(Number of
hops=20m)','Intra-cluster(Number of hops=30m)');

function pushbutton8_Callback(hObject, eventdata, handles)
**********************************************************
* %inter offset:                                         *
**********************************************************
clear all;
close all;
%n=get(handles.popupmenu2,'value');
n=25;
interh=[10 15 20 30];
Mmx=25;
for conthops=1:4
hops=interh(conthops)/1000;
for M=1:Mmx
Db=hops*randn(n,1);
Da=hops*randn(n,1);
Dc=hops*randn(n,1);
Thba=Db+Da;
Thbc=Da;
Thbd=Da+Db+Dc;
Ac(conthops,M)=sum(abs(Thba)+abs(Thbc)+abs(Thbd))/(n*M);
```

```
end
end
n=1:Mmx;
Ac=Ac*60/(1000*60*1*1e-6);
plot(n,Ac(1,:),'o-',n,Ac(2,:),'*-',n,Ac(3,:),'v-',n,Ac(4,:),'^')
legend('Number of hops=10 m','Number of hops=15 m','Number of
hops=20 m','Number of hops=30 m');
Title('Inter Offset')
xlabel('Number of Synchronization Processes')
ylabel('Average clock Offset us')
Ac=Ac*60/(1000*60*0.8*1e-6);

%plot(n,Ac2(1,:),'m-*',n,Ac2(2,:),'r-h',n,Ac2(3,:),'b-
v',n,Ac2(4,:),'y-^')
%legend('Ms-assisted(Number of hops=10m)','Ms-assisted(Number of
hops=15m)','Ms-assisted(Number of hops=20m)','Ms-assisted(Number
of hops=30m)','Intra-cluster(Number of hops=10m)','Intra-
cluster(Number of hops=15m)','Intra-cluster(Number of
hops=20m)','Intra-cluster(Number of hops=30m)');

function pushbutton9_Callback(hObject, eventdata, handles)
**************************************************
* %inter skew:                                    *
**************************************************
clear all;
close all;
a=get(handles.popupmenu2,'value');
n=a;
n=80;
interh=[10 15 20 30];
Mmx=25;
```

```
for sigi=1:4
sig=interh(sigi)/1000;
for M=1:Mmx
Db=sig*randn(n,1);
Da=sig*randn(n,1);
Dc=sig*randn(n,1);
Dd=sig*randn(n,1);
Thsba=Da;
Thsbc=Dc;
Thsbd=Dd;
Ac(sigi,M)=sum(abs(Thsba)+abs(Thsbc)+abs(Thsbd))/(n*M);
end
end
n=1:Mmx;
Ac=Ac*60/(1000*60*11*1e-6);
plot(n,Ac(1,:),'o-',n,Ac(2,:),'*-',n,Ac(3,:),'v',n,Ac(4,:),'^')
legend('Number of hops=10 m','Number of hops=15 m','Number of
hops=20 m','Number of hops=30 m');
Title('Inter skew')
xlabel('Number of Synchronization Process')
ylabel('Average skew (ppm)')

%plot(a,Ac2(1,:),'m-*',a,Ac2(2,:),'r-*',a,Ac2(3,:),'b-
v',a,Ac2(4,:),'y-^')
%legend('Ms-assisted(Number of hops=10m)','Ms-assisted(Number of
hops=15m)','Ms-assisted(Number of hops=20m)','Ms-assisted(Number
of hops=30m)','Intra-cluster(Number of hops=10m)','Intra-
cluster(Number of hops=15m)','Intra-cluster(Number of
hops=20m)','Intra-cluster(Number of hops=30m)');

function popupmenu2_Callback(hObject, eventdata, handles)
function popupmenu2_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```