MATLAB INTERFACE FOR RPL DODAG VISUALIZATION

Software Recommended: NetSim Standard v12.0 (32bit/ 64bit), Visual Studio 2015/2017/2019, MATLAB (32bit/ 64bit)

Note: This project works only with MATLAB v2015b or higher versions.

Follow the instructions specified in the following link to clone/download the project folder from GitHub using Visual Studio:
https://tetcos.freshdesk.com/support/solutions/articles/14000099351-how-to-clone-netsim-file-exchange-project-repositories-from-github-

Other tools such as GitHub Desktop, SVN Client, Sourcetree, Git from the command line, or any client you like to clone the Git repository.

Note: It is recommended not to download the project as an archive (compressed zip) to avoid incompatibility while importing workspaces into NetSim.

Secure URL for the GitHub repository:

https://github.com/NetSim-TETCOS/RPL_DODAG_Formation_Visualization_in_IOT_Networks_v12.0.git

RPL (Routing Protocol for Low-Power and Lossy Networks) is a routing protocol for wireless networks with low power consumption and generally susceptible to packet loss. It is a proactive protocol based on distance vectors and operates on IEEE 802.15.4, optimized for multi-hop and many-to-one communication, but also supports one-to-one messages.

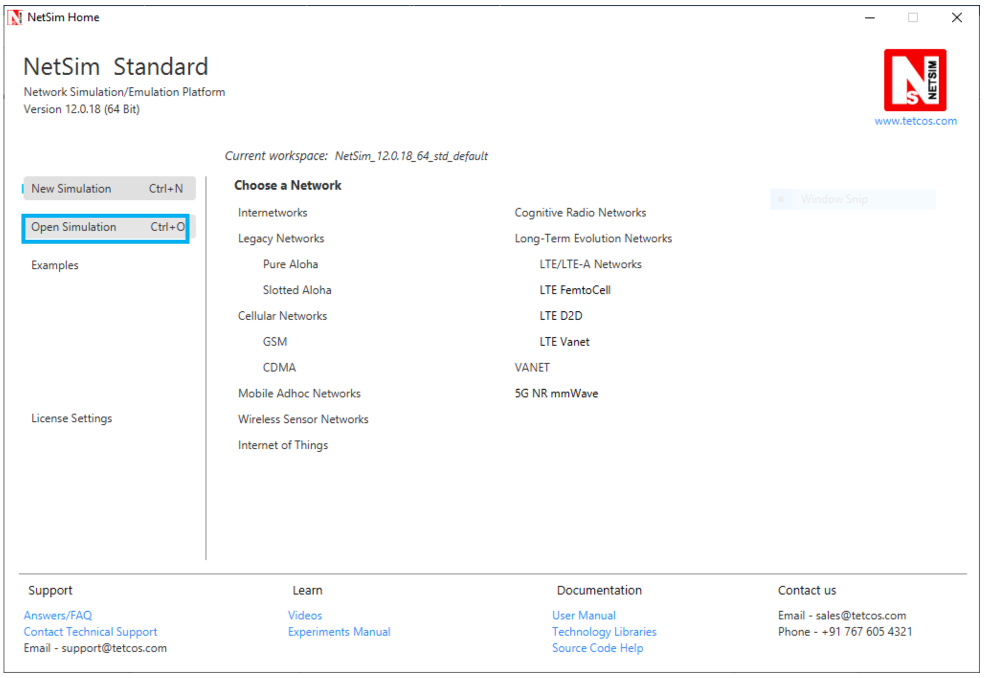
This protocol is specified in RFC 6550 with special applications in RFCs 5867, 5826, 5673 and 5548. RPL can support a wide variety of link layers, including those with limitations, with potential losses or that are used in devices with limited resources. This protocol can quickly create network routes, share routing knowledge and adapt the topology in an efficient way.

RPL creates a topology similar to a tree (DAG or directed acyclic graph). Each node within the network has an assigned rank (Rank), which increases as the teams move away from the root node (DODAG). The nodes resend packets using the lowest range as the route selection criteria.

The DODAG formed as part of RPL protocol in NetSim can be visualized by interfacing NetSim with MATLAB.

Steps to run MATLAB interface

1. After you unzip the downloaded project folder, Open NetSim Home Page click on **Open Simulation** option,


NetSim Home

NetSim Standard
Network Simulation/Emulation Platform
Version 12.0.18 (64 Bit)

www.tetcos.com

Current workspace: NetSim_12.0.18_64_std_default

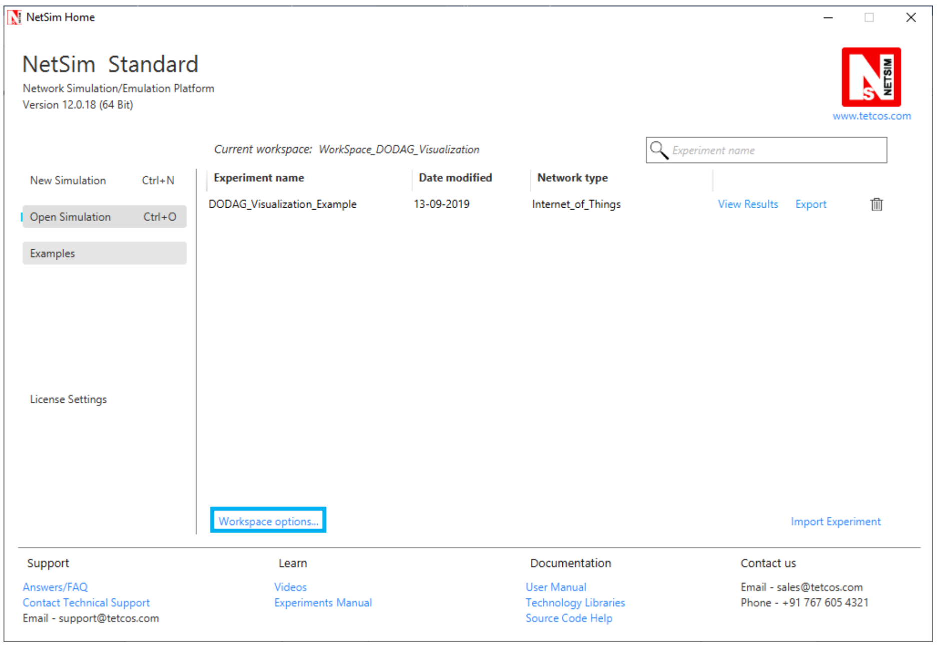
New Simulation Ctrl+N
Open Simulation Ctrl+O
Examples
License Settings

Choose a Network

Internetworks	Cognitive Radio Networks
Legacy Networks	Long-Term Evolution Networks
Pure Aloha	LTE/LTE-A Networks
Slotted Aloha	LTE FemtoCell
Cellular Networks	LTE D2D
GSM	LTE Vanet
CDMA	VANET
Mobile Adhoc Networks	5G NR mmWave
Wireless Sensor Networks	
Internet of Things	

Support	Learn	Documentation	Contact us
Answers/FAQ	Videos	User Manual	Email - sales@tetcos.com
Contact Technical Support	Experiments Manual	Technology Libraries	Phone - +91 767 605 4321
Email - support@tetcos.com		Source Code Help	

2. Click on **Workspace options**


NetSim Home

NetSim Standard
Network Simulation/Emulation Platform
Version 12.0.18 (64 Bit)

www.tetcos.com

Current workspace: WorkSpace_DODAG_Visualization

Experiment name

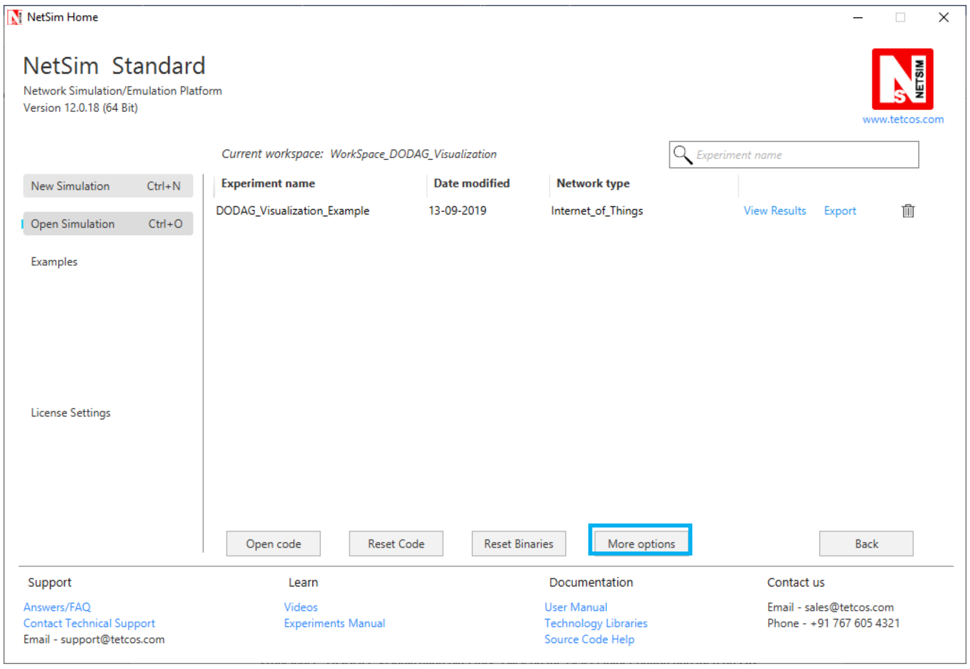
New Simulation Ctrl+N
Open Simulation Ctrl+O
Examples
License Settings

Experiment name	Date modified	Network type		
DODAG_Visualization_Example	13-09-2019	Internet_of_Things	View Results	Export

Workspace options...
Import Experiment

Support	Learn	Documentation	Contact us
Answers/FAQ	Videos	User Manual	Email - sales@tetcos.com
Contact Technical Support	Experiments Manual	Technology Libraries	Phone - +91 767 605 4321
Email - support@tetcos.com		Source Code Help	

3. Click on **More Options,**


NetSim Home

NetSim Standard
Network Simulation/Emulation Platform
Version 12.0.18 (64 Bit)

www.tetcos.com

Current workspace: WorkSpace_DODAG_Visualization

Experiment name	Date modified	Network type
DODAG_Visualization_Example	13-09-2019	Internet_of_Things

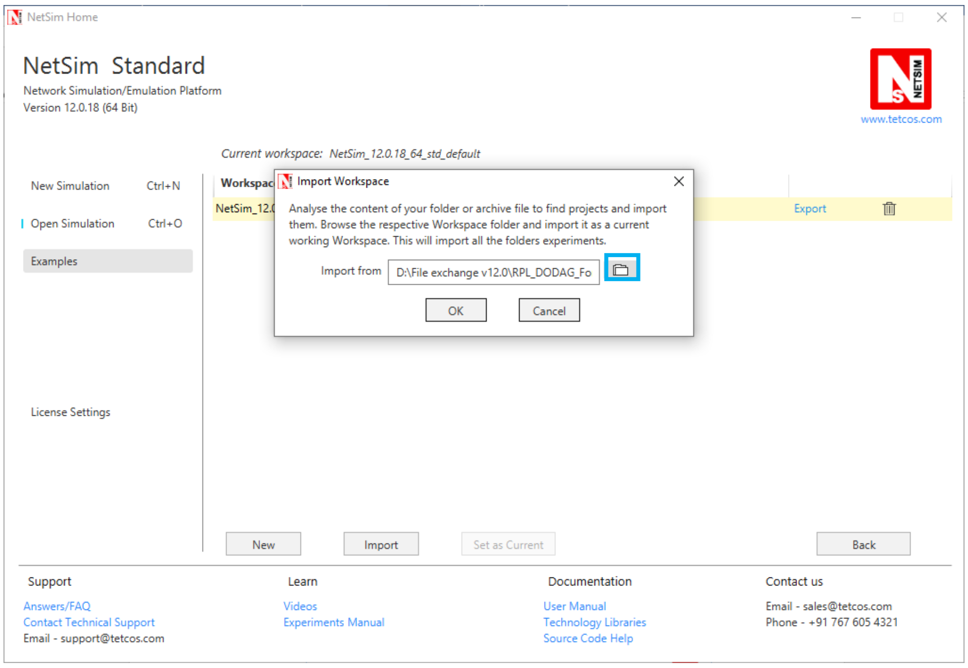
Support
Answers/FAQ
Contact Technical Support
Email - support@tetcos.com

Learn
Videos
Experiments Manual

Documentation
User Manual
Technology Libraries
Source Code Help

Contact us
Email - sales@tetcos.com
Phone - +91 767 605 4321

4. Click on **Import**, browse the extracted folder path and go into the Workspace_DODAG_Visualization directory. Click on the Select folder button and then on **OK.**


NetSim Home

NetSim Standard
Network Simulation/Emulation Platform
Version 12.0.18 (64 Bit)

www.tetcos.com

Current workspace: NetSim_12.0.18_64_std_default

Import Workspace

Analyse the content of your folder or archive file to find projects and import them. Browse the respective Workspace folder and import it as a current working Workspace. This will import all the folders experiments.

Import from D:\File exchange v12.0\RPL_DODAG_Fo

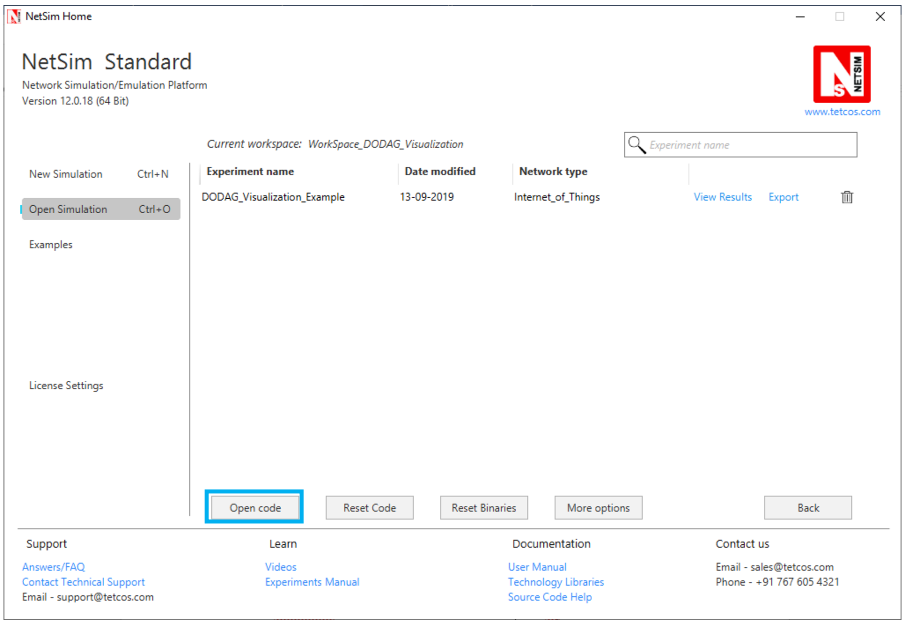
Support
Answers/FAQ
Contact Technical Support
Email - support@tetcos.com

Learn
Videos
Experiments Manual

Documentation
User Manual
Technology Libraries
Source Code Help

Contact us
Email - sales@tetcos.com
Phone - +91 767 605 4321

5. Go to home page, Click on **Open Simulation → Workspace options → Open code**



Current workspace: WorkSpace_DODAG_Visualization

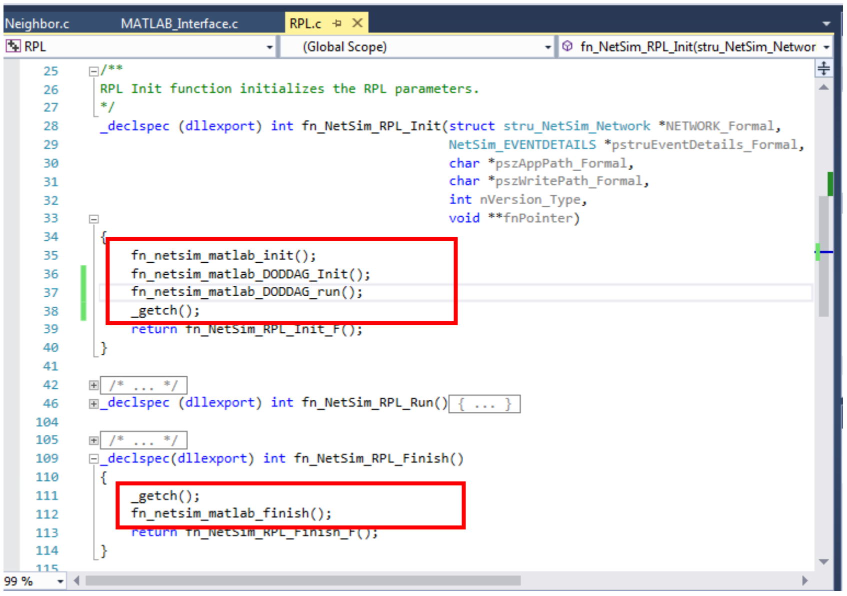
Experiment name	Date modified	Network type
DODAG_Visualization_Example	13-09-2019	Internet_of_Things

Email - support@tetcos.com

Email - sales@tetcos.com
Phone - +91 767 605 4321

6. Place **PlotDAG.m** file inside the MATLAB root directory. For Eg: **“<MATLAB installed path>\MATLAB\R2015b”, (Note: PlotDAG.m** is provided inside the MATLAB_Code directory)

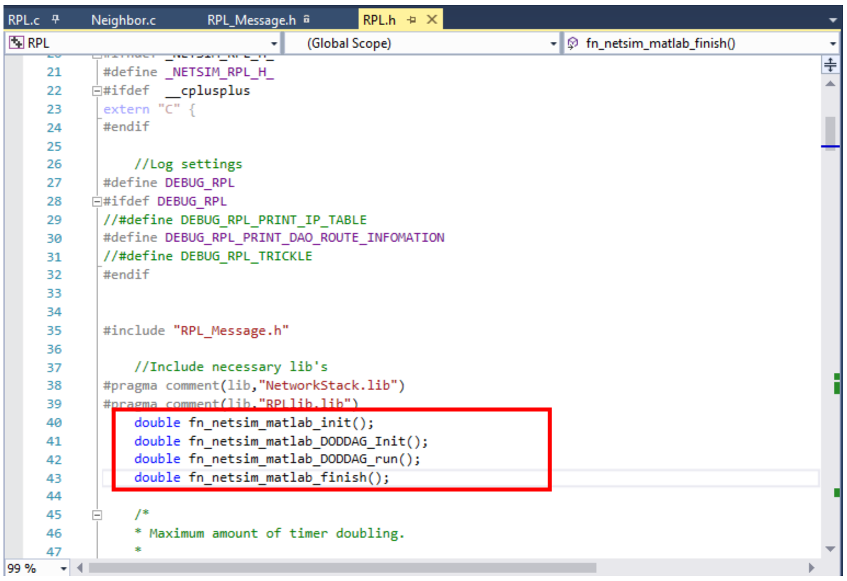
7. Following modifications were done to the RPL project for this implementation:
   a. Open RPL.c file and add **fn_netsim_matlab_init()**, **fn_netsim_matlab_DODDAG_run()** and **fn_netsim_matlab_DODDAG_Init()** inside **fn_NetSim_RPL_Init()** and **fn_netsim_matlab_finish()** inside **fn_NetSim_RPL_Finish ()**.



```
/**
RPL Init function initializes the RPL parameters.
*/
_declspec (dllexport) int fn_NetSim_RPL_Init(struct stru_NetSim_Network *NETWORK_Formal,
                                             NetSim_EVENTDETAILS *pstruEventDetails_Formal,
                                             char *pszAppPath_Formal,
                                             char *pszWritePath_Formal,
                                             int nVersion_Type,
                                             void **fnPointer)
{
    fn_netsim_matlab_init();
    fn_netsim_matlab_DODDAG_Init();
    fn_netsim_matlab_DODDAG_run();
    _getch();
    return fn_NetSim_RPL_Init_F();
}

/* ... */
_declspec (dllexport) int fn_NetSim_RPL_Run(){ ... }

/* ... */
_declspec(dllexport) int fn_NetSim_RPL_Finish()
{
    _getch();
    fn_netsim_matlab_finish();
    return fn_NetSim_RPL_Finish_F();
}
```

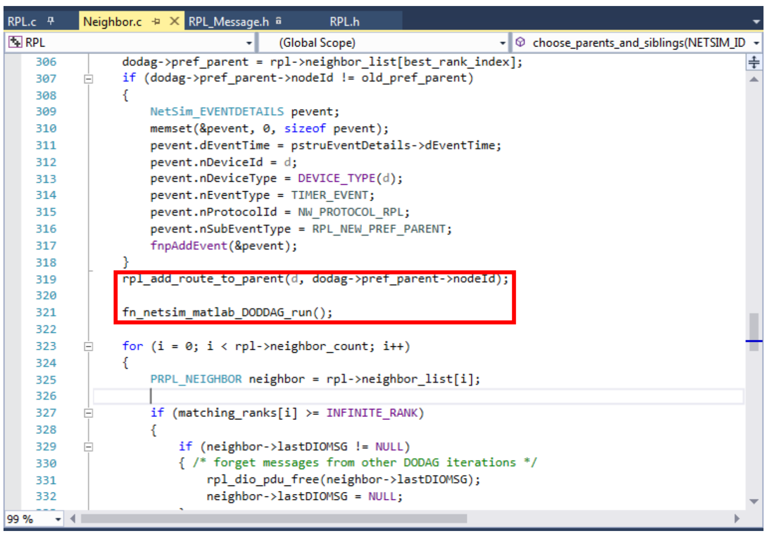
b. Add definitions of the following functions inside **RPL.h** file

```
    a. double fn_netsim_matlab_init();
    b. double fn_netsim_matlab_DODDAG_Init();
    c. double fn_netsim_matlab_DODDAG_run();
c.     double fn_netsim_matlab_finish();
```



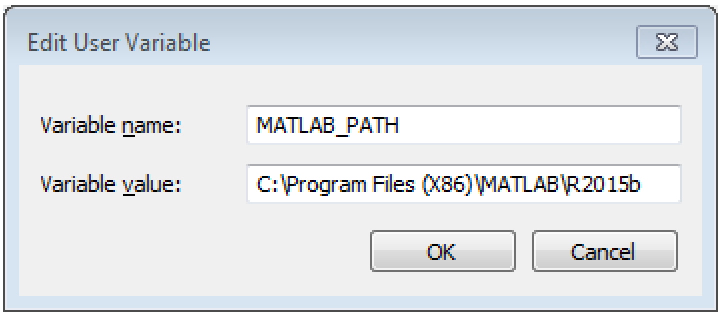
```
#define _NETSIM_RPL_H_
#ifdef __cplusplus
extern "C" {
#endif

    //Log settings
#define DEBUG_RPL
#ifdef DEBUG_RPL
//#define DEBUG_RPL_PRINT_IP_TABLE
#define DEBUG_RPL_PRINT_DAO_ROUTE_INFOMATION
//#define DEBUG_RPL_TRICKLE
#endif


#include "RPL_Message.h"

    //Include necessary lib's
#pragma comment(lib,"NetworkStack.lib")
#pragma comment(lib,"RPLlib.lib")
    double fn_netsim_matlab_init();
    double fn_netsim_matlab_DODDAG_Init();
    double fn_netsim_matlab_DODDAG_run();
    double fn_netsim_matlab_finish();

    /*
    * Maximum amount of timer doubling.
    *
```

d. Go to the **Neighbor.c** file. Inside Function **void choose_parents_and_siblings(NETSIM_ID d)** add **fn_netsim_matlab_DODDAG_run()** below **rpl_add_route_to_parent()**



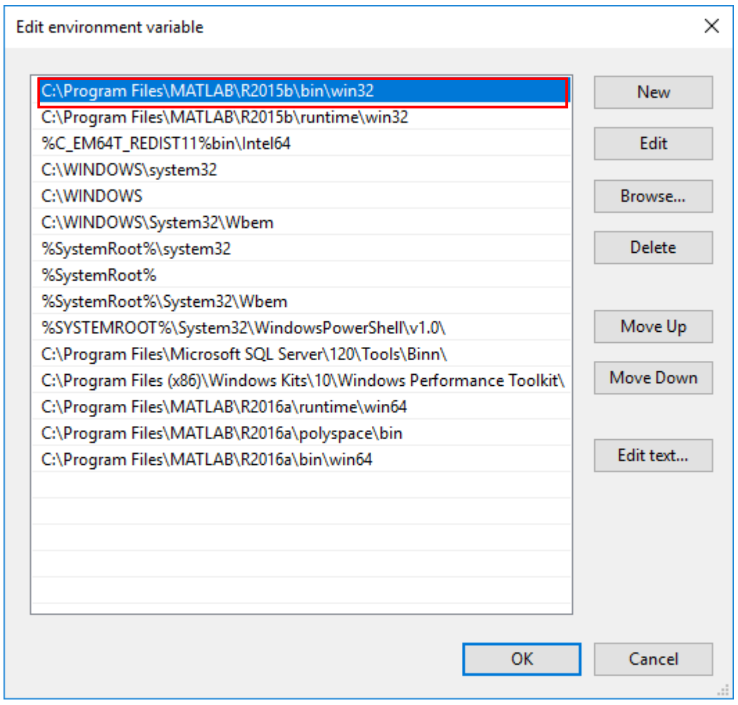
```
    dodag->pref_parent = rpl->neighbor_list[best_rank_index];
    if (dodag->pref_parent->nodeId != old_pref_parent)
    {
        NetSim_EVENTDETAILS pevent;
        memset(&pevent, 0, sizeof pevent);
        pevent.dEventTime = pstruEventDetails->dEventTime;
        pevent.nDeviceId = d;
        pevent.nDeviceType = DEVICE_TYPE(d);
        pevent.nEventType = TIMER_EVENT;
        pevent.nProtocolId = NW_PROTOCOL_RPL;
        pevent.nSubEventType = RPL_NEW_PREF_PARENT;
        fnpAddEvent(&pevent);
    }
    rpl_add_route_to_parent(d, dodag->pref_parent->nodeId);

    fn_netsim_matlab_DODDAG_run();

    for (i = 0; i < rpl->neighbor_count; i++)
    {
        PRPL_NEIGHBOR neighbor = rpl->neighbor_list[i];

        if (matching_ranks[i] >= INFINITE_RANK)
        {
            if (neighbor->lastDIOMSG != NULL)
            { /* forget messages from other DODAG iterations */
                rpl_dio_pdu_free(neighbor->lastDIOMSG);
                neighbor->lastDIOMSG = NULL;
```

8. Create a user variable with the name of MATLAB_PATH and provide the path of the installation directory of user’s respective MATLAB version.



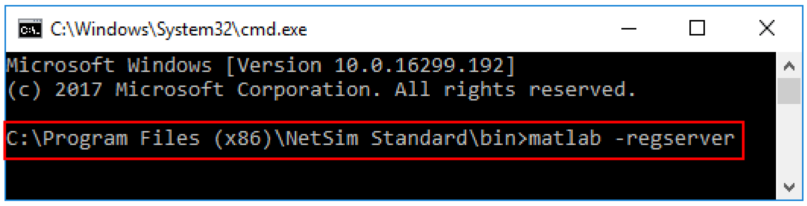
Variable name: MATLAB_PATH
Variable value: C:\Program Files (X86)\MATLAB\R2015b

9. Make sure that the following directory is in the PATH(Environment variable)
<Path where MATLAB is installed>\bin\win32



```
C:\Program Files\MATLAB\R2015b\bin\win32
C:\Program Files\MATLAB\R2015b\runtime\win32
%C_EM64T_REDIST11%bin\Intel64
C:\WINDOWS\system32
C:\WINDOWS
C:\WINDOWS\System32\Wbem
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
C:\Program Files\Microsoft SQL Server\120\Tools\Binn\
C:\Program Files (x86)\Windows Kits\10\Windows Performance Toolkit\
C:\Program Files\MATLAB\R2016a\runtime\win64
C:\Program Files\MATLAB\R2016a\polyspace\bin
C:\Program Files\MATLAB\R2016a\bin\win64
```

(**Note:** To run this code 32- bit version of MATLAB must be installed in your system. If you are interfacing for the first time then open command window and go to the **<NetSim installed directory>\bin** and type **matlab -regserver**)

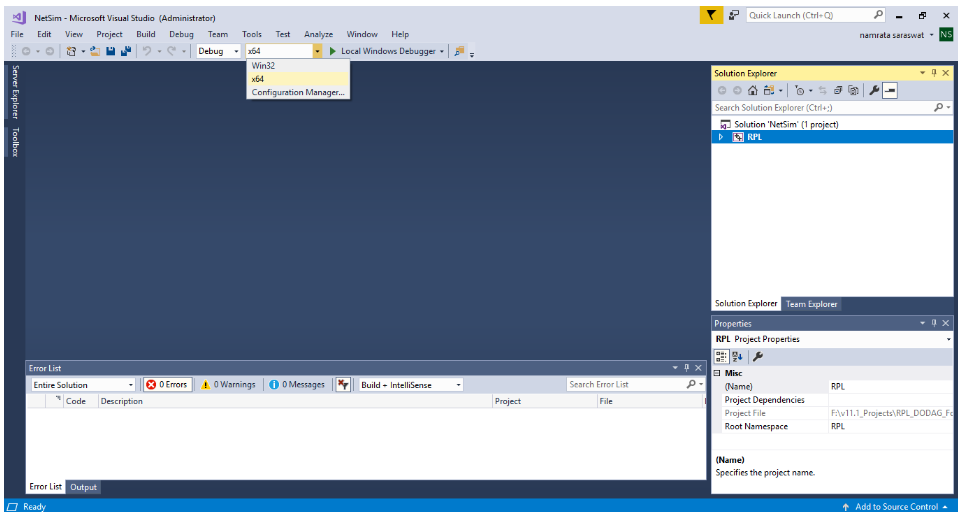


```
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Program Files (x86)\NetSim Standard\bin>matlab -regserver
```

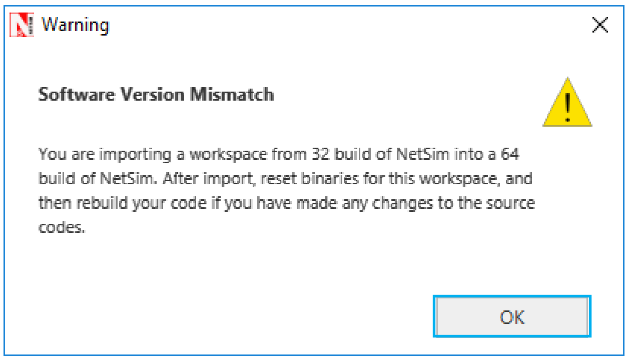
10. Now Right Click on RPL project and select Rebuild.
11. Upon rebuilding, **RPL.dll** will automatically be updated in the respective bin folder of the current workspace.

Note:

- Based on whether you are using NetSim 32 bit or 64 bit setup you can configure Visual studio to build 32 bit or 64 bit Dll files respectively as shown below:



- While importing the workspace, if the following warning message indicating Software Version Mismatch is displayed, you can ignore it and proceed.



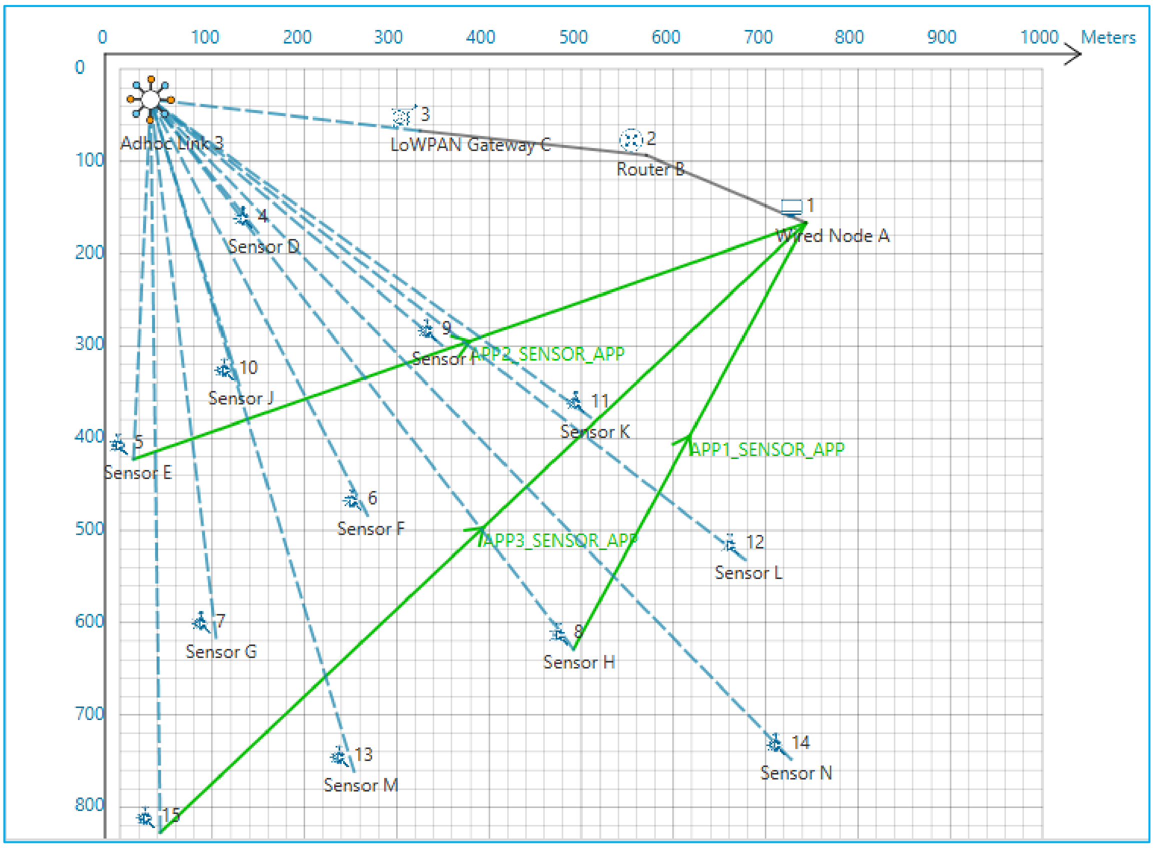
Warning

Software Version Mismatch

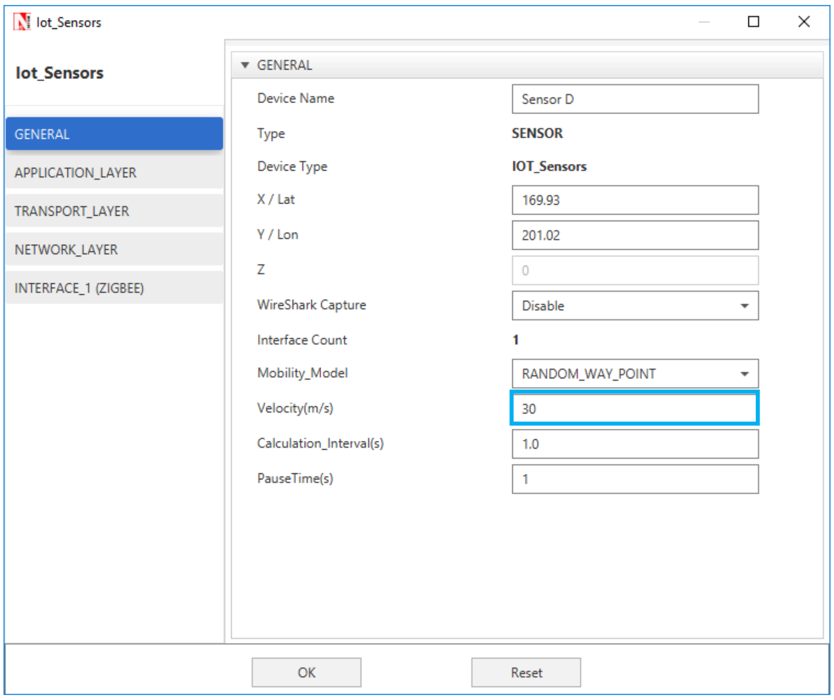
You are importing a workspace from 32 build of NetSim into a 64 build of NetSim. After import, reset binaries for this workspace, and then rebuild your code if you have made any changes to the source codes.

OK

- Go to NetSim home page, click on **Open Simulation**, Click on **DODAG_Visualization_Example**.



Set Velocity to the sensors



Iot_Sensors

Iot_Sensors

GENERAL
APPLICATION_LAYER
TRANSPORT_LAYER
NETWORK_LAYER
INTERFACE_1 (ZIGBEE)

GENERAL

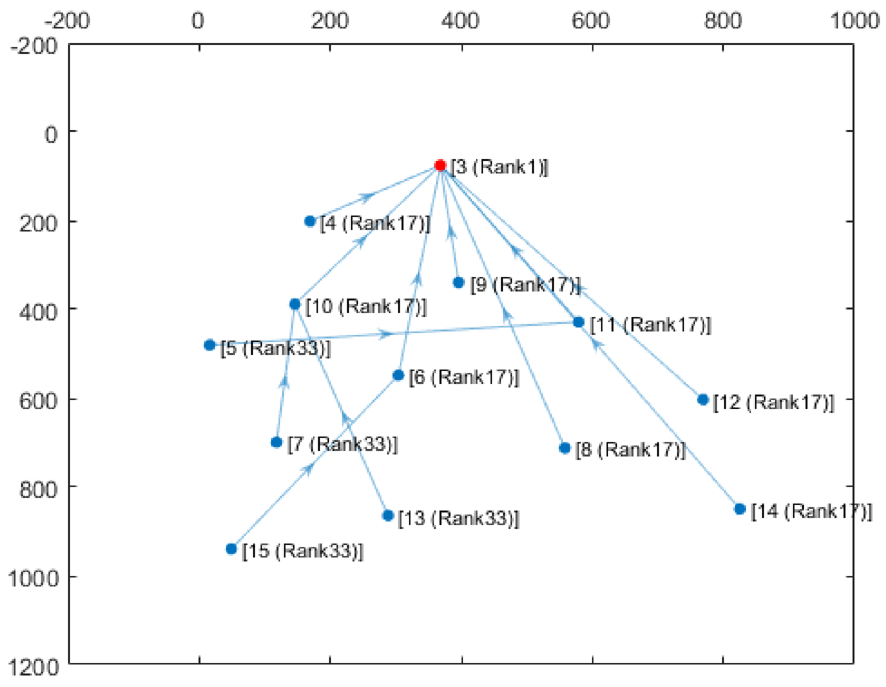
Device Name	Sensor D
Type	SENSOR
Device Type	IOT_Sensors
X / Lat	169.93
Y / Lon	201.02
Z	0
WireShark Capture	Disable
Interface Count	1
Mobility_Model	RANDOM_WAY_POINT
Velocity(m/s)	30
Calculation_Interval(s)	1.0
PauseTime(s)	1

OK Reset

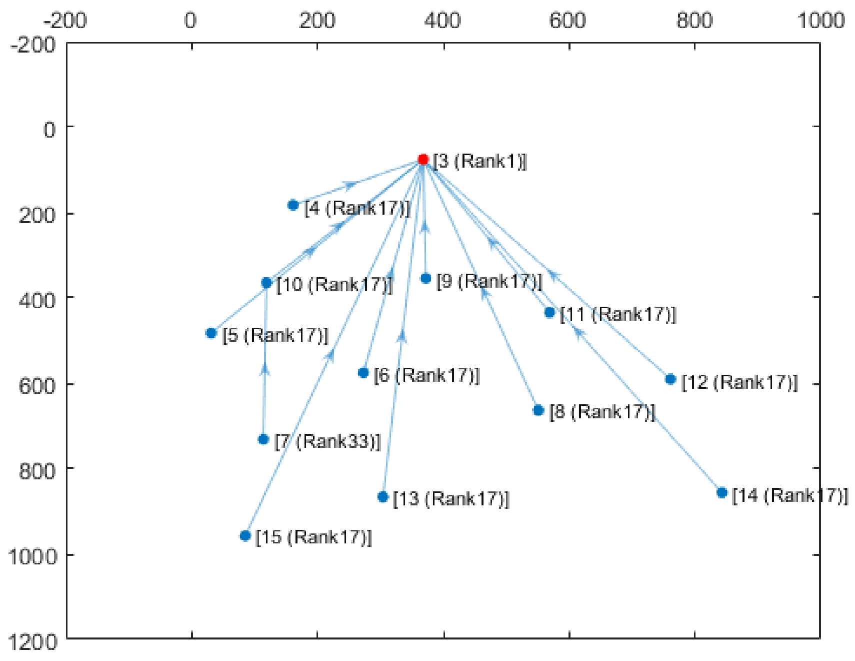
Output:

A plot will open, showing the DODAG when the simulation is started and the first route is formed between sink node and the sensor. And the DODAG will be dynamically updated.

Initially formed DODAG



DODAG formed after some time due to movement in sensors



After simulation press any key in the NetSim command window to close the MATLAB.