City Grid Simulator

By Ogen Malek, Roy Goldstein, Nir Tsabar and Oren Shmuel.

Course: 236340 - Project in networking

Semester: Winter 2015/2016

Instructor: Prof. Reuven Cohen

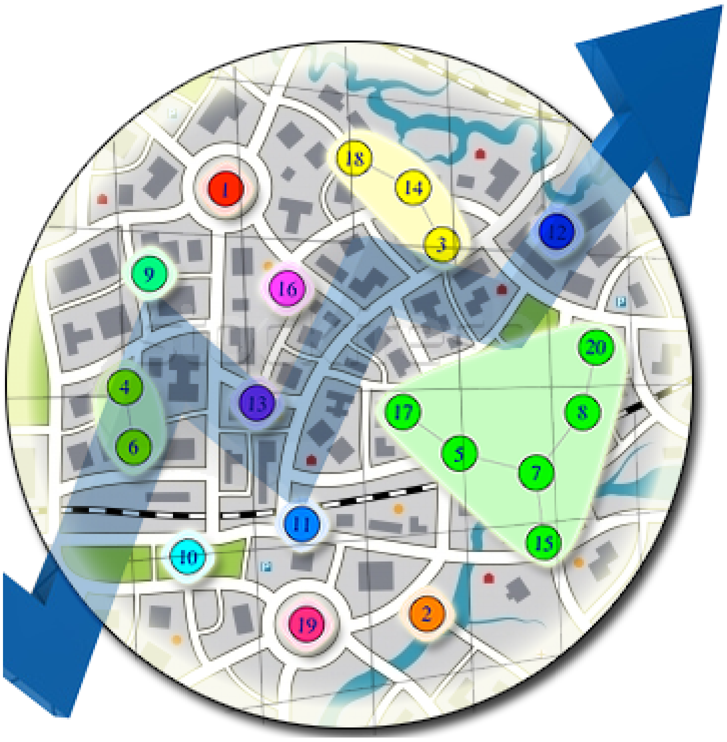


Table of Contents

Minimum system requirements

- Operation System: Windows 7, Windows 8 or Windows 10. (Wasn't tested on XP).
- Microsoft Excel 2013.
- .Net Framework 4.5.2 (Will be automatically downloaded in case missing).

It is recommended to run the simulator with number of threads that is less or equal to number of logical cores (and not more than that!).

Project Goals

- Simulate a wireless network containing thousands of units.
- Measure the network performance.
- Easy to use.
- Fast simulation.
- Maximize flexibility – the user can change almost everything!
- Easy to develop and measure new algorithms.
- Ability to analyze simulations.

System Modules

- Map
    - Grid Generation
        - Placing RSUs randomly and make sure it’s a strongly connected components graph.
    - Set Gateways locations
    - Build Trees
- Simulator
    - Events
    - Mac Layer + Link Layer
    - Application Layer
- Logging System
    - Creates xml file that log all the simulation events.
- Graph Generation using Excel

Running the Simulation

1. Set the parameters in “Advanced settings”. (not mandatory, defaults values are set)
2. Select the algorithms you’d like to compare at “automatic simulation”. (Both gateways and tree algorithms, there should be at least 1 algorithm from each category.
3. Select the parameters at “general settings”, number of iterations should be ~10 to get reliable results.
4. #RSUs/Grid^2 ratio should be about ~0.4.
5. Select X Coordinate variable, bounds and step.
6. press “Start”.
7. The excel will appear as soon as the simulation is done.

Recommended ratios for Transmission power 3:

Grid Length	Min #RSUs	Normal #RSUs
50	800	900
100	4000	4200
150	9000	9550
200	17150	18000
300	42000	43000
400	75500	76000

How to add a new tree building algorithm

Adding a new tree building algorithm can be done by the following steps:

(All the relevant steps are performed over the following path - "CityGridSimulator\Model\MapLogic\")

- Add a cost function to "BuildTreeMethods.cs".
  Cost function for example can be

  ```
  public static double MinimumDegree(Edge<RoadSideUnit> edge)
  {
      return edge.Source.Neighbors.Count();
  }
  ```

- Add a name for that algorithm + description at "EChooseTreeBuildingMethod.cs"
  For example:

  ```
  [Description("Minimum Degree")]
   MinimumDegree = 3,
  ```

- Add a translation from algorithm name to function name at "Map.cs" in the function "translateTreeBuildingMethod", for example:

  ```
  case (EChooseTreeBuildingMethod.MinimumDegree):
        return BuildTreeMethods.MinimumDegree;
  ```

Note, multiplication algorithms could be also added - for that use log function,

for example -

```
public static double LowestProductDegreePath(Edge<RoadSideUnit> edge)
{
      return Math.Log10(edge.Source.Neighbors.Count());
}
```

How to add a new gateway choosing algorithm

Adding a new gateway choosing algorithm can be done by the following steps:

(All the relevant steps are performed over the following path - "CityGridSimulator\Model\MapLogic\")

- Add a gateways choosing function to "GatewaysChooser.cs".
  Algorithm for example -

```
public void ChooseGatewaysByTopology(int numOfGateways)
 {
     _grid.CleanGateways();
     int numRegions = DivideRegionsByTopology(numOfGateways);
       Dictionary<int, StronglyConnectedRegion> regions =
       CalculateNumberOfGatewaysPerRegion(numRegions, numOfGateways);
     ChooseRegionGatewaysByTopology(regions);
 }
```

- Add a name for that algorithm + description at "EChooseGatewayMethod.cs"
  For example:

```
 [Description("By Topology")]
  Topology = 3,
```

- Add a translation from algorithm name to function name at "Map.cs" in the function "InitMap", for example:

```
case (EChooseGatewaysMethod.Topology):
       _gatewaysChooser.ChooseGatewaysByTopology(numOfGateways);
       break;
```

Logs Section

The logs can be enabled via advanced settings → "Enable Logs".

The logs are XML files that can be opened by any text viewer and are found at

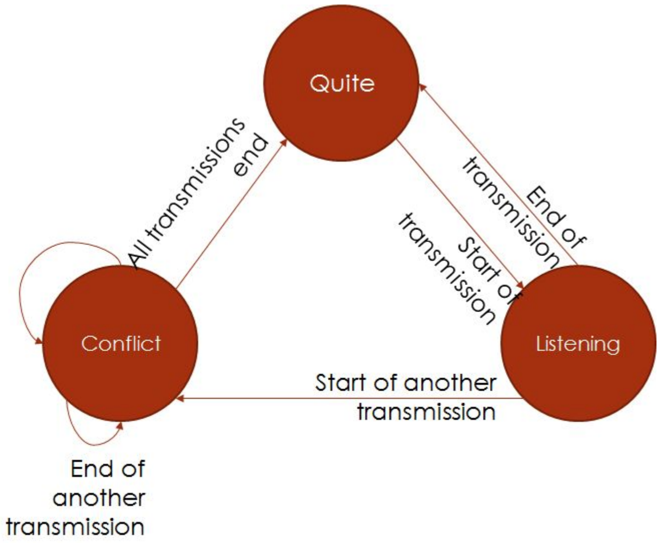
"<Running Directory> \ logs".

Note that logs are mainly used to check the correctness of the program and should not be used when running comparison simulations.

(The logs for one long simulation could get up to ~1GByte and more).

Event Sections

The state machine in each RSU



NewBroadcastAttemptEvent

- Adds a new StartBroadcastEvent that will take place after LBT msec.

StartBroadcastEvent

- If there was no conflict during LBT
    - Start transmitting and starts a backoff to all the relevant neighbors.
    - When transmission starts, update the neighbors state according to the state machine.
- Adds ‘StopBroadcastEvent’ that will take place in ‘message time’.

StopBroadcastEvent

- Stops the transmission.
- Updates all the neighbors’ state.
- Checks if there are additional consecutive transmissions of that message.
    - If yes, newBroadcastAttemptEvent is called.
    - Otherwise CheckBroadcastMessageFinishEvent

CheckBroadcastMessageFinishEvent

- Checks if all the children have received current message.
    - If yes, taking the next message from the queue and creates relevant event.
    - Otherwise –WakeUpBroadcastPenaltyEvent will take place after penalty time.

WakeUpBroadcastPenaltyEvent

- Stops the broadcast suspension.
- Checks if a unicast message is currently being sent,
    - If yes, do nothing, the broadcast will resume after the unicast is done.
    - Otherwise, start a newBroadcastAttemptEvent.

NewUnicastSendEvent

- Generates another NewUnicastSendEvent to 1 second from now.
- Checks if a new unicast message should be generated (random),
    - If yes – enqueues the new unicast message.
    - In case the RSU was inactive, starts NewUnicastAttemptEvent.

NewUnicastAttemptEvent

- Creates StartUnicastEvent after LBT time.

StartUnicastEvent

- If there was no conflict during LBT,
    - Starts transmitting and starts a backoff to all the relevant neighbors.
    - When transmission starts, update the neighbors state according to the state machine.
- Adds ‘StopUnicastEvent’ that will take place in ‘message time’.

StopUnicastEvent

- Stops the transmission
- Updates all the neighbors' state.
- Checks if the message was transmitted successfully.
    - If yes,
        - Starts Ack transmitting.
        - FinishUnicastAckEvent is called after Ack time.
        - Updates the state of all the neighbors of the destination of the unicast (the RSU who transmits the Ack).
    - Otherwise, generates NewUnicastAttemptEvent according to the backoff algorithm.

FinishUnicastAckEvent

- Stops the ack transmission.
- Updates all the neighbors' state.
- Dequeues the next task form the queue and applies the relevant event.

Gateways Choosing Algorithms

1. Random

given number of gateways G, choosing randomly G RSUs to be gateways.

2. Connection Quality

Stage 1: Divide the graph to Strongly Connected Components (SCC) based on link quality.

Stage 2: In each SCC pick the RSUs with most neighbors (with quality of the SCC).

Note: this was a primitive algorithm, afterwards it was tested and did not get better results than random.

For performance results, see appendix 1.

3. Shortest Links

Stage 1: Divide the graph to Strongly Connected Components (SCC)

based on link length using Binary Search to find the best SCCs division.

Stage 2: In each SCC pick the RSUs with most neighbors (with quality of the SCC).

Note: this was a primitive algorithm, afterwards it was tested and did not get better results than random.

For performance results, see appendix 1.

4. Topology

Stage 1: Divide the RSUs to regions with equal number of RSUs in each region.

The division is done by ordering the RSUs according to their location on the axis with the biggest delta between the two ends.

Stage 2: In each region set the gateway in the middle of the other axis.

Note: this was a primitive algorithm, afterwards it was tested and did not get better results than random.

For performance results, see appendix 1.

5. Topology Pieces

given a number of gateways G, lets define piece to be group of RSUs, lets define the first piece to be the group of all the RSUs.

Stage 1: find the longest distance in axis X between two RSUs in the piece, and the longest distance in axis Y between two RSUs in the piece, define the axis with the longer distance to be the chosen axis of the first piece.

Stage 2: until there are G pieces take each piece P and:

- split it into two pieces separated by the median RSU (when the RSUs is ordered by the coordinate of the chosen axis of the piece).
- define the chosen axis of the new pieces to be the other axis referring to the chosen axis of the initial piece P.

For performance results see appendix 2.

6. Topology Pieces V2

given a number of gateways G, lets define piece to be group of RSUs, lets define the first piece to be the group of all the RSUs.

Stage 1: put the first piece in a queue.

Stage 2: until the queue size is G do:

- dequeue a piece P from the queue.
- find the longest distance in axis X between two RSUs in the piece.
- find the longest distance in axis Y between two RSUs in the piece.
- if axis Y has longer distance split the piece P into two pieces separated by the median RSU (when the RSUs is ordered by their Y coordinate).
- otherwise split the piece P into two pieces separated by the median RSU (when the RSUs is ordered by their X coordinate).
- enqueue the new two pieces to the queue

Stage 3: for each piece in the queue calculate the average location of the piece and put gateway in the closest RSU (in euclidian distance) to the calculated average location.

For performance results, see appendix 3.

7. Equal Squares

Stage 1: Split the map to $\left(\sqrt{\#Gateways}\right)^2$ equal regions.

in each region set a gateway in the middle of the region.

(The closest RSU to the middle is promoted to be a Gateway)

Stage 2: for all the leftover gateways -

While there are still gateways to choose:

- Build a shortest distance tree from current gateways.
- Find the most distant RSU, for the path within the gateway and that RSU - set a gateway in the ⅔ of the way.

For performance results, see appendix 4.

8. Equal Rectangles

Same principle as the Equal squares,

in this algorithm split the map into rectangles regions with the following formula:

NumXRegions = $\left[\sqrt{\#Gateways}\right]$

NumYRegions = $\frac{\#Gateways}{numXRegions}$

RegionHeight = $\frac{GridLength}{NumXRegions}$

RegionWidth = $\frac{GridLength}{NumY\,Regions}$

Stage 2: Same as previous algorithm.

For performance results, see appendix 5.

9. Distance From Middle

- Set a gateway in the middle.
- While there are still gateways to choose:
  - Build a shortest distance tree from current gateways.
  - Find the most distant RSU, for the path within the gateway and that RSU - set a gateway in the ⅔ of the way.

For performance results, see appendix 6.

For comparison between algorithm 5-9, see Appendix 14.

Building Tree Method

All the methods below define weights to all graph's edges.

Afterwards, Using Dijkstra from each gateway – they find the radio distance to each RSU.

All the multiplication algorithm defines their weight to be logarithmic on the returned values.

1. Shortest Radio Distance

each edge gets its radio distance between the RSUs of its ends.

For performance results, see appendix 7.

2. Shortest Path

each edge gets its euclidian distance between the RSUs of its ends.

For performance results, see appendix 7.

3. Link Quality

each edge gets the value according to the his line quality as following:

- High Quality - value of ⅓
- Medium Quality- value of ½
- Low Quality - value of 1

For performance results, see appendix 8.

4. Minimum Degree

each edge gets its source number of neighbors.

For performance results, see appendix 9.

5. Maximum Degree

each edge gets subtraction between the total number of RSUs in the grid and the source number of neighbors.

For performance results, see appendix 10.

6. Lowest Product Degree Path

multiplication algorithm -

each edge gets its source number of neighbors.

For performance results, see appendix 11.

7. Quality Degree And Path

each get value according to the line quality, the source degre, and the path length. each of following criteria normalized to value between 0 to 1. on the following way:

- Quality:
    - High Quality - value of 0
    - Medium Quality - value of 0.6
    - High Quality - value of 1
- degree: the source degree div the max degree in the graph
- Path: value of 1 for each hop.

after that each of the criteria is multiplied by a factor, and the value edge is the sum of all the results.

In this algorithm the factors are:

- Line Quality Factor: 2
- Source Degree Factor: 5
- Path Factor: 3

For performance results, see appendix 12.

8. Degree Factor Quality

each edge gets value according to the following formula: Quality_Factor * Source_Degree + Path_Factor.

where the Quality_Factor is:

- High Quality: 1
- Medium Quality: 2
- Low Quality: 2.5

and the Path_Factor is: 7

For performance results, see appendix 13.

GUI Guide

General Overview

The GUI consists of:

1. Three main screens (Home, Automatic Simulation, Advanced Settings).
2. Two upper tabs (GUI Settings, About).
3. “Automatic Simulation In Progress” screen.
4. Back button (←).

Main Screens

1. Home: the welcome screen which the user gets on startup , includes the application’s logo, also see Appendix 15.
2. Automatic Simulation: the screen which the simulation is ran from, includes:
    - Gateways Choosing Algorithms multiple choice box.
    - Building Tree Methods multiple choice box.
    - General Settings box:
        - Number of iterations: how many times each configuration is being ran (important: each iteration includes the whole interval values of the X Coordinate variable).
        - Number of Threads: how many threads may participate in the simulation (in order to get best performance it is recommended to choose the number of logical Threads on the running machine).
        - Grid Length: the size of each axis of the grid (must be at least the square root of the Number of RSUs).
        - Number of Road Side Units: how many RSUs should be located on the grid (must be at most the minimum between the square of the Grid Length and the Number of Gateways).
        - Number Of Gateways: how many RSUs on the grid should be classified as gateways (must be at most the Number of RSUs).
    - X Coordinate variable single choice box: the chosen variable will be the measured topic of the simulation- its values will be defined in the X Coordinate bounds box, so the defined value of the relevant field in the General Settings box or Advanced Settings Screen will be overwritten, all the others parameters will be constant to all of the simulation progress, the choices:
        - Number of gateways.
        - Background traffic (will be explained in the Advanced Settings Screen section).
        - Number of consecutive transmissions (will be explained in the Advanced Settings Screen section).

    - Penalty time for broadcast failure (will be explained in the Advanced Settings Screen section).
  - X Coordinate bounds box:
    - Lower bound: the minimum value of the X Coordinate Variable.
    - Upper bound: the maximum value of the X Coordinate Variable (there is not necessarily a run with this value).
    - Step: the difference between each two consecutive values of X.
    - X Samplings: gives indication about how many runs are there according to the chosen Lower bound, Upper bound, Step fields.
  - Start Button: runs the simulation.

  also see Appendix 16.

3. Advanced Settings:
   - Transmission power: the radius on the that a transmission is being heard in (must be at least 1).
   - Background traffic: the probability for a single unicast to be sent from its source RSU (must be between 0 to 1).
   - Penalty time: the time in millisecond which a RSU stops to participate in broadcast transmissions, this penalty starts when RSU finished to send all of the consecutive transmissions of a single packet, and there is a destination (child in the tree) which did not get the packet even once.
   - Number of consecutive transmissions: the number of times which a broadcast sends consecutively the same packet to all of its children in the tree.
   - Number of broadcast packets: how many packets should be delivered from the gateways to all of the RSUs in the grid.
   - Enable Logs: this options enables each run to write to log, more about the logs in the Simulation Output section.

   also see Appendix 17.

Upper Tabs

1. Gui Settings:
   - an option to choose the color of the colorized fields (20 different colors).
   - an option to choose Theme for the Application (light, dark).
   - an option to choose the font size (small, large).

   also see Appendix 18.

2. About: information about the developers of the simulation, also see Appendix 19.

“Automatic Simulation In Progress” Screen

- Current Run Progress: percentage of the current run progress.

- Current Run Description: indicates information about the current run-
  - Simulator time: the current simulator time in this run.
  - Received: number of the packets that was received by any RSU from the total number (the total number is the number of RSUs multiplied by the number of broadcast packets).
- Total Progress: percentage of the simulation progress.
- Total run status: the number of runs that finished from the total number of runs in this simulation (the total number of runs is the number of chosen gateways choosing algorithms multiplied by the number of chosen building tree methods, and this result multiplied by the number of iterations).
- Estimated time left: gives an estimated indication about how many time is left until the simulator will finish. pay attention that the estimation is very rough and gets more accurate as the simulation progresses.
- Cancel button: cancel the simulation - may take few minutes if the simulation is in the middle of location Gateways.
- ❖ **when the simulation finishes (naturally or by cancellation), an Excel sheet will automatically open with the simulation data, results, and graphs, moreover, the sheet and all the photos of the graphs will be saved "SimOutput" Folder.**

  also see Appendix 20.

Back Button

allows to go back to the previous screen of the simulator, available in every screen except for the "Automatic Simulation In Progress" screen.

Simulation Output

Excel Simulation Results Output

the output of the finishes simulation is created inside **simOutput** folder followed by finish date of the simulation**.**

the output folder will consist of the following items:

- **excel simulation output file** - the file contain 3 worksheets:
    - **Summary** - list the different parameters that used by the current simulation run, also see Appendix 21.
    - **DataSet**

      upper dataset

        - Gateways Choosing Algorithm row - represent the common criteria by which the simulation runs
        - Building tree method row - the compared criteria that runs under common criteria
        - x variable type row - values of x variable run under the two criterias
        - iteration rows - represent the run time under the same 3 rows criterias , each iteration result is the simulation time with different grid.
        - average row - average the result of the all iterations

      bottom dataset - inversion of the upper dataset created by swap between Gateways Choosing Algorithm , and Building tree rows.

      also see Appendix 22.
    - **Graphs** - charts that were built according to the DataSet worksheet , each chart include all the graph lines with the same common criteria as title of the graph.
      graph line is plotted by taking the points of the x variable and the average simulation time value that matches to specific x value,

      also see Appendix 23.
- **graph snapshots -** a snapshot of each generated excel chart will be created with the title of the chart as file name.

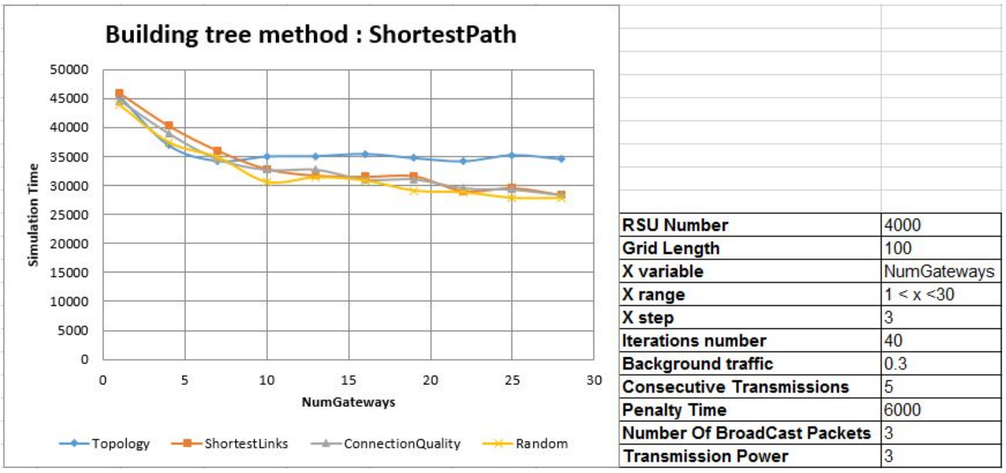
Simulation Log Output

the log file of each thread of each run from the simulation is created inside the **Logs** folder followed by Start date of the specific run, this xml file includes information regarding the simulation events and additional actions (such as backoff and penalty).

it is not recommended to enable logs for ordinary simulations, because it creates a lot of log data (Gigabytes), it provides information about the progress of each run, and may be enabled for single and short simulations (otherwise it is very hard to follow it), also see Appendix 24.

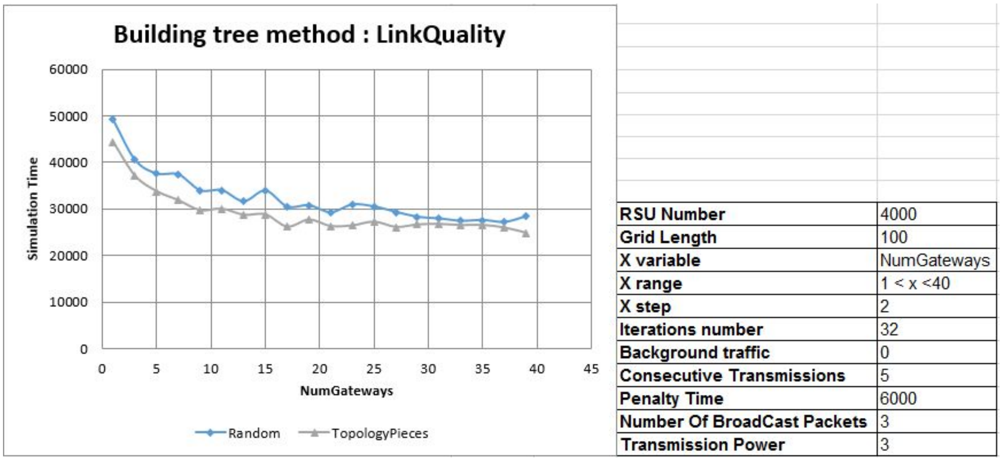
Appendixes

Appendix 1 - Topology vs. Shortest Links vs. Connection Quality vs. Random



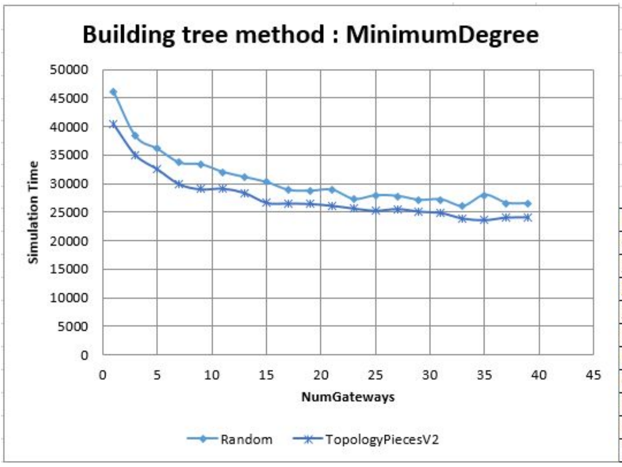
RSU Number	4000
Grid Length	100
X variable	NumGateways
X range	1 < x <30
X step	3
Iterations number	40
Background traffic	0.3
Consecutive Transmissions	5
Penalty Time	6000
Number Of BroadCast Packets	3
Transmission Power	3

Appendix 2 - Random vs. Topology Pieces.



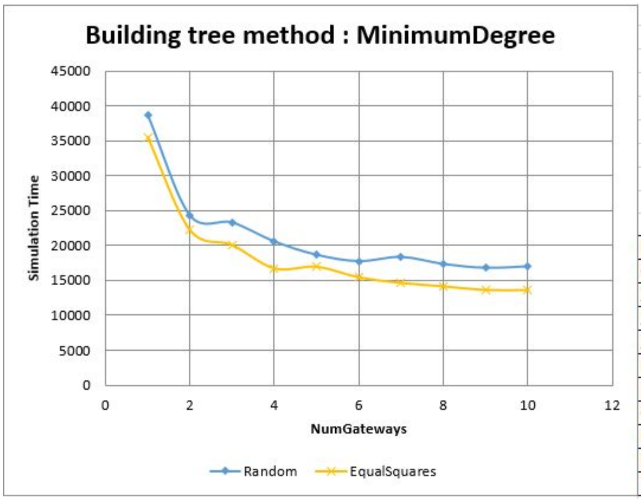
RSU Number	4000
Grid Length	100
X variable	NumGateways
X range	1 < x <40
X step	2
Iterations number	32
Background traffic	0
Consecutive Transmissions	5
Penalty Time	6000
Number Of BroadCast Packets	3
Transmission Power	3

Appendix 3 - Random vs. Topology Pieces V2



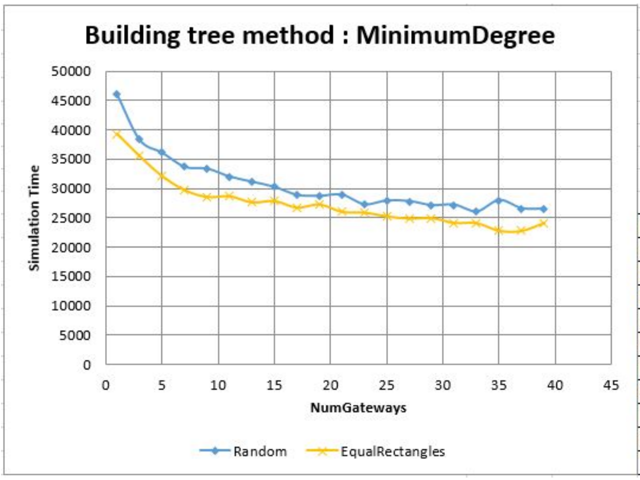
RSU Number	4000
Grid Length	100
X variable	NumGateways
X range	1 < x <40
X step	2
Iterations number	32
Background traffic	0
Consecutive Transmissions	5
Penalty Time	6000
Number Of BroadCast Packets	3
Transmission Power	3

Appendix 4 - Random vs. Equal Squares



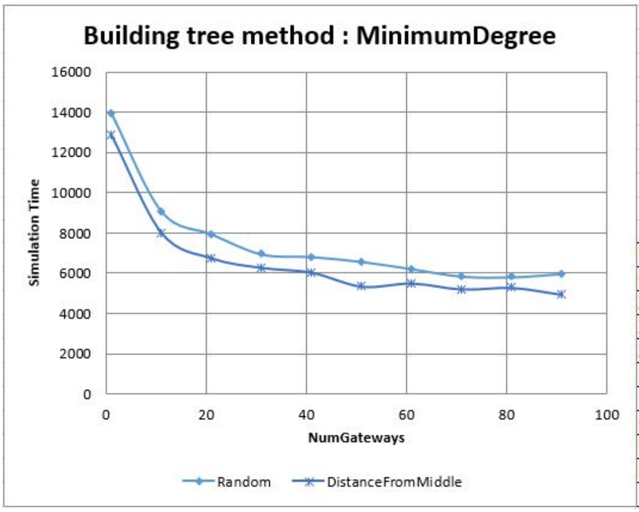
RSU Number	9000
Grid Length	150
X variable	NumGateways
X range	1 < x <100
X step	10
Iterations number	24
Background traffic	0.3
Consecutive Transmissions	5
Penalty Time	6000
Number Of BroadCast Packets	1
Transmission Power	3

Appendix 5 - Random vs. Equal Rectangles



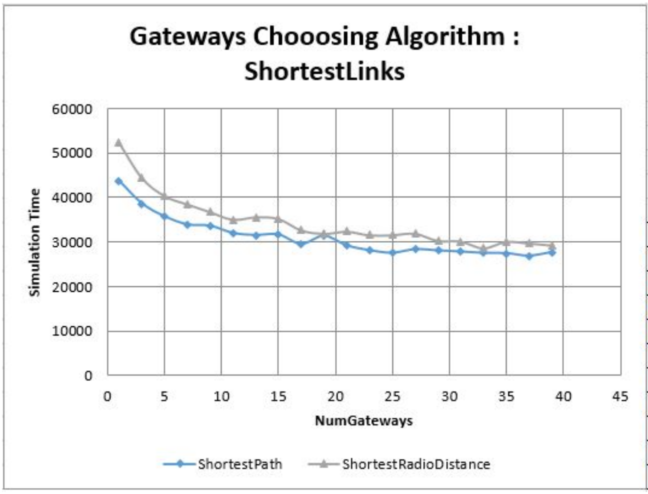
RSU Number	4000
Grid Length	100
X variable	NumGateways
X range	1 < x <40
X step	2
Iterations number	32
Background traffic	0
Consecutive Transmissions	5
Penalty Time	6000
Number Of BroadCast Packets	3
Transmission Power	3

Appendix 6 - Random vs. Distance from middle



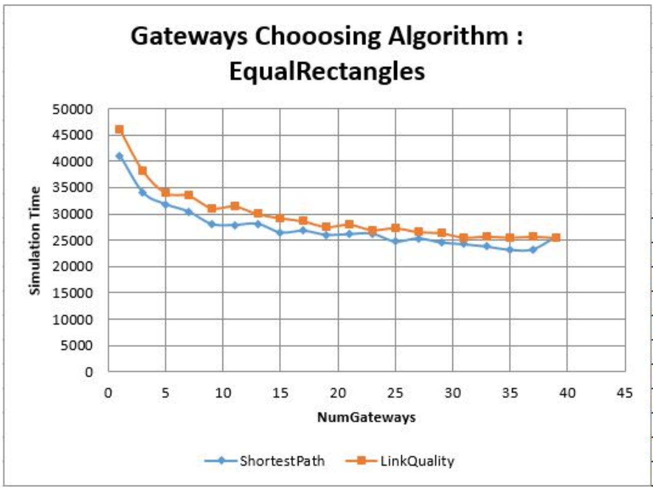
RSU Number	9000
Grid Length	150
X variable	NumGateways
X range	1 < x <100
X step	10
Iterations number	24
Background traffic	0.3
Consecutive Transmissions	5
Penalty Time	2000
Number Of BroadCast Packets	1
Transmission Power	3

Appendix 7 - Shortest Radio Distance vs. Shortest path



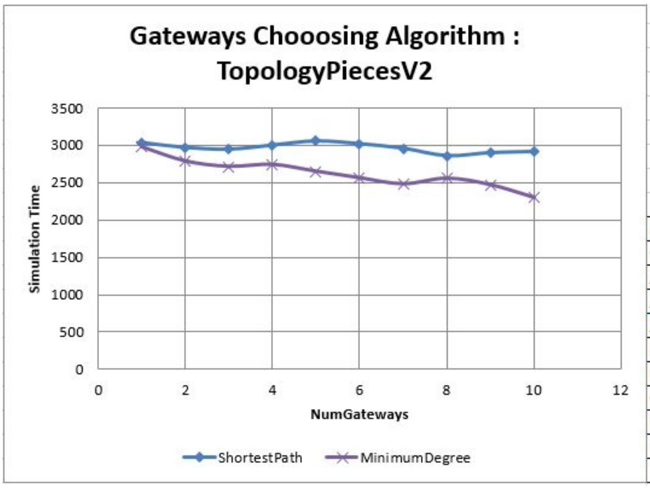
RSU Number	4000
Grid Length	100
X variable	NumGateways
X range	1 < x <40
X step	2
Iterations number	32
Background traffic	0
Consecutive Transmissions	5
Penalty Time	6000
Number Of BroadCast Packets	3
Transmission Power	3

Appendix 8 - Shortest Path vs. Link Quality



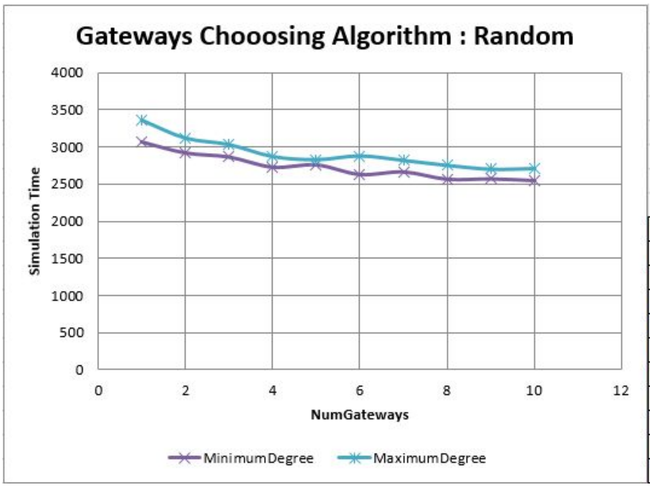
RSU Number	4000
Grid Length	100
X variable	NumGateways
X range	1 < x <40
X step	2
Iterations number	32
Background traffic	0
Consecutive Transmissions	5
Penalty Time	6000
Number Of BroadCast Packets	3
Transmission Power	3

Appendix 9 - Shortest Path vs. Minimum Degree



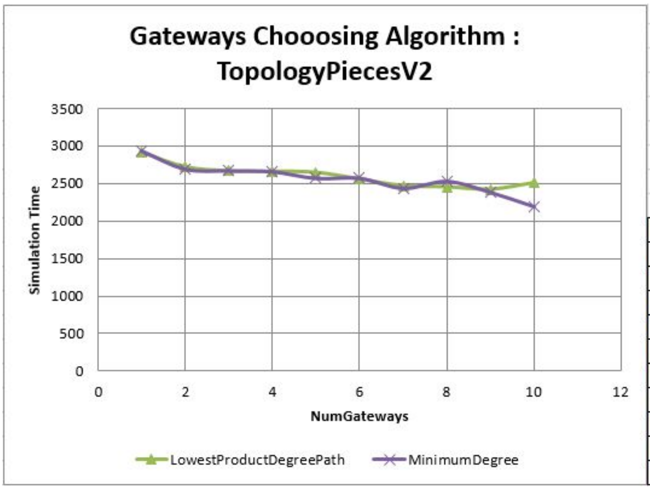
RSU Number	4500
Grid Length	150
X variable	NumGateways
X range	1 < x <10
X step	1
Iterations number	100
Background traffic	0
Consecutive Transmissions	4
Penalty Time	500
Number Of BroadCast Packets	1
Transmission Power	20

Appendix 10 - Minimum degree vs Maximum degree



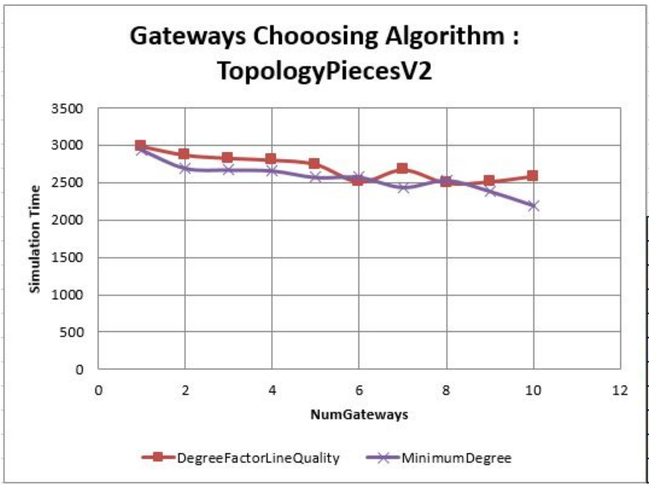
RSU Number	4500
Grid Length	150
X variable	NumGateways
X range	1 < x <10
X step	1
Iterations number	100
Background traffic	0
Consecutive Transmissions	4
Penalty Time	500
Number Of BroadCast Packets	1
Transmission Power	20

Appendix 11 - Minimum degree vs Lowest Product Degree Path



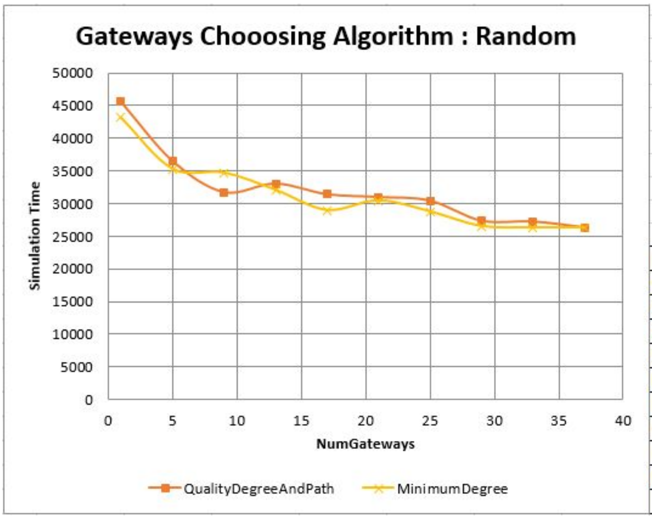
RSU Number	4500
Grid Length	150
X variable	NumGateways
X range	1 < x <10
X step	1
Iterations number	100
Background traffic	0
Consecutive Transmissions	4
Penalty Time	500
Number Of BroadCast Packets	1
Transmission Power	20

Appendix 12 - Minimum Degree vs Degree Factor Line Quality



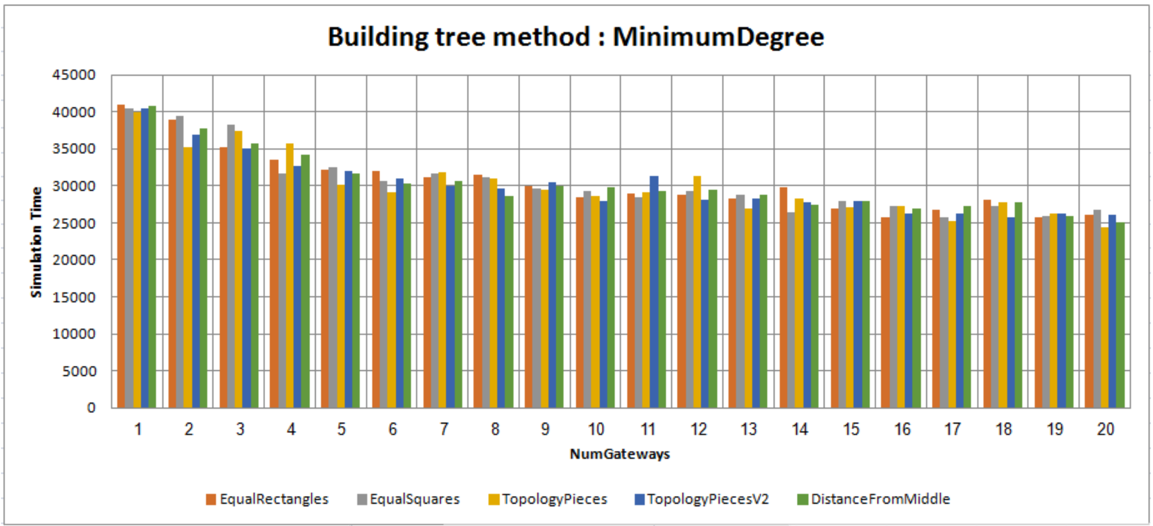
RSU Number	4500
Grid Length	150
X variable	NumGateways
X range	1 < x <10
X step	1
Iterations number	100
Background traffic	0
Consecutive Transmissions	4
Penalty Time	500
Number Of BroadCast Packets	1
Transmission Power	20

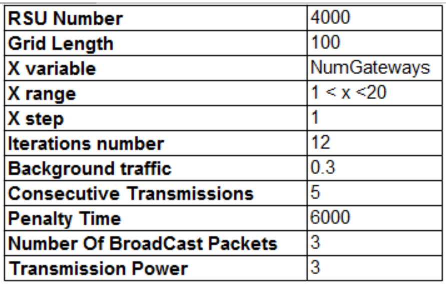
Appendix 13 - Minimum Degree vs Quality Degree and Path



RSU Number	4000
Grid Length	100
X variable	NumGateways
X range	1 < x <40
X step	4
Iterations number	12
Background traffic	0.3
Consecutive Transmissions	5
Penalty Time	6000
Number Of BroadCast Packets	3
Transmission Power	3

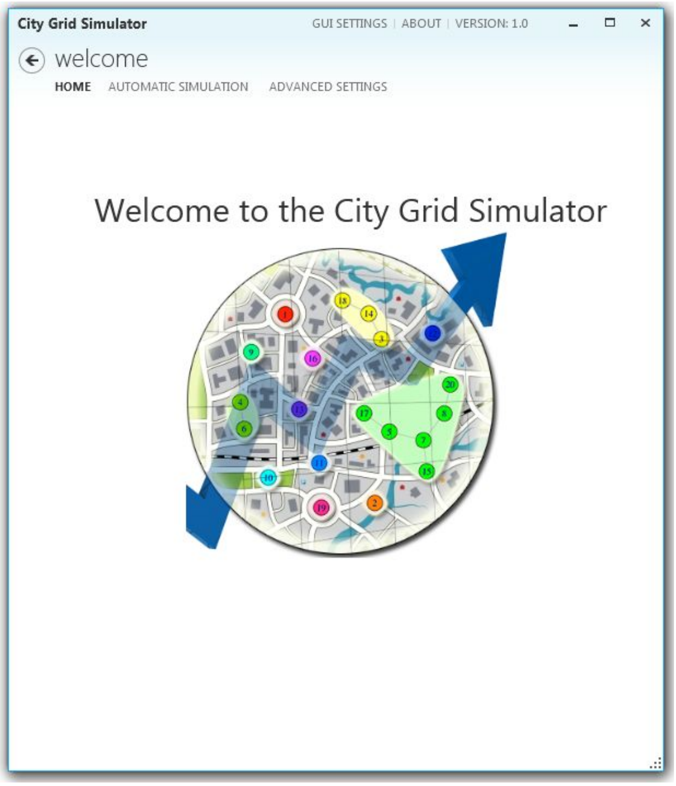
Appendix 14 - Comparison between top five Gateway Choosing Algorithms





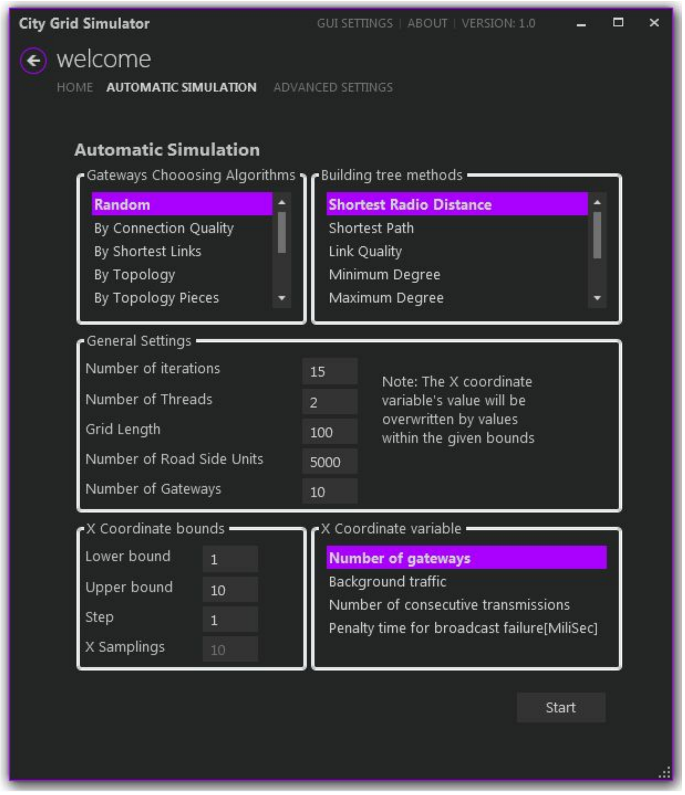
RSU Number	4000
Grid Length	100
X variable	NumGateways
X range	1 < x <20
X step	1
Iterations number	12
Background traffic	0.3
Consecutive Transmissions	5
Penalty Time	6000
Number Of BroadCast Packets	3
Transmission Power	3

Appendix 15 - Home Screen



Welcome to the City Grid Simulator

Appendix 16 - Automatic Simulation Screen



Automatic Simulation

Gateways Chooosing Algorithms
- Random
- By Connection Quality
- By Shortest Links
- By Topology
- By Topology Pieces

Building tree methods
- Shortest Radio Distance
- Shortest Path
- Link Quality
- Minimum Degree
- Maximum Degree

General Settings

Number of iterations	15
Number of Threads	2
Grid Length	100
Number of Road Side Units	5000
Number of Gateways	10

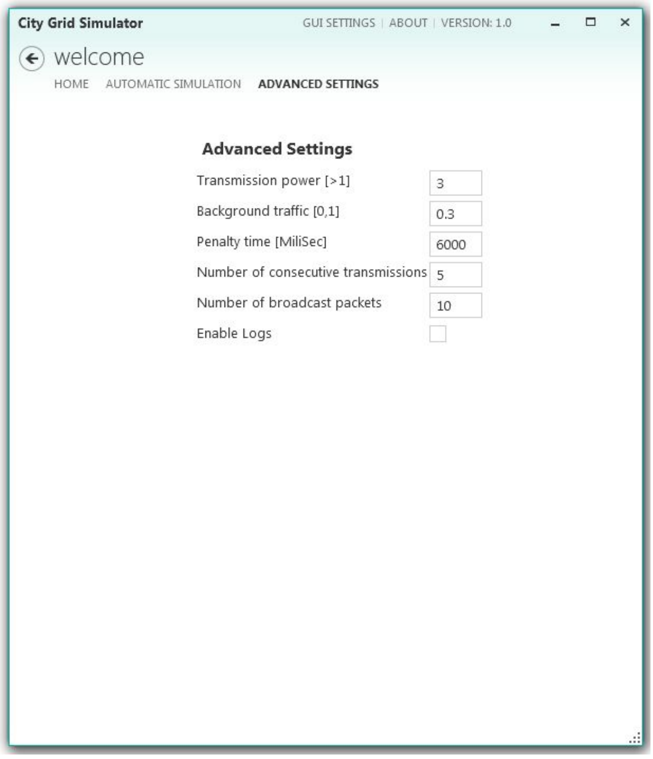
Note: The X coordinate variable's value will be overwritten by values within the given bounds

X Coordinate bounds

Lower bound	1
Upper bound	10
Step	1
X Samplings	10

X Coordinate variable
- Number of gateways
- Background traffic
- Number of consecutive transmissions
- Penalty time for broadcast failure[MiliSec]

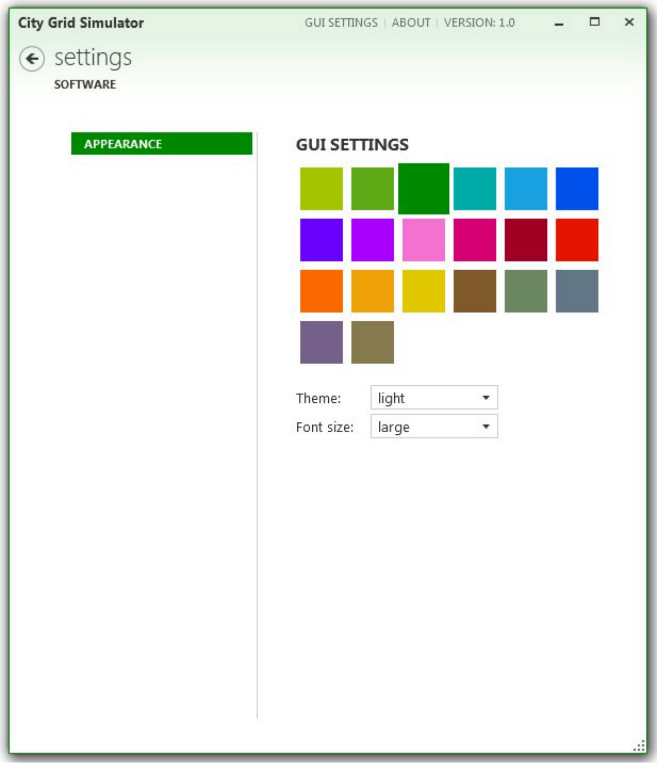
Appendix 17 - Advanced Settings Screen



Advanced Settings

Transmission power [>1]	3
Background traffic [0,1]	0.3
Penalty time [MiliSec]	6000
Number of consecutive transmissions	5
Number of broadcast packets	10
Enable Logs	□

Appendix 18 - GUI Settings Screen

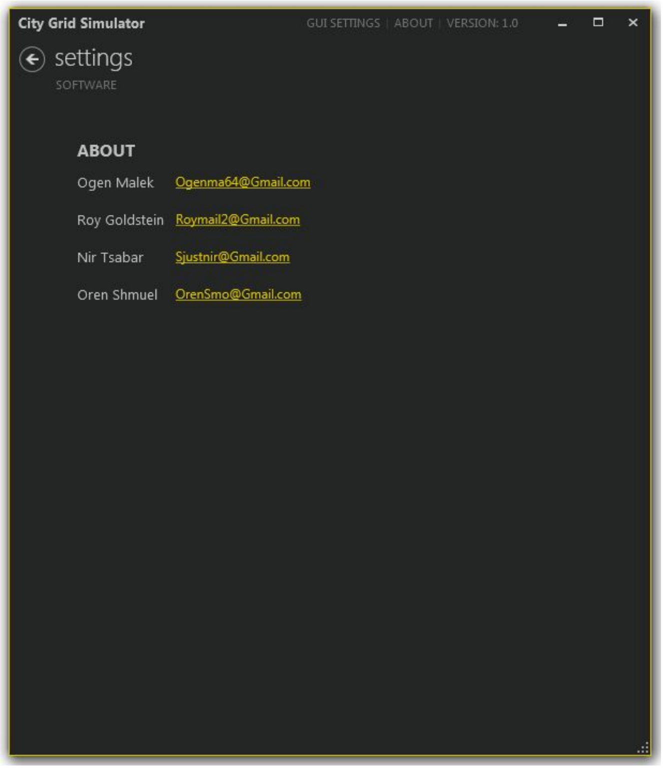


GUI SETTINGS

Theme: light

Font size: large

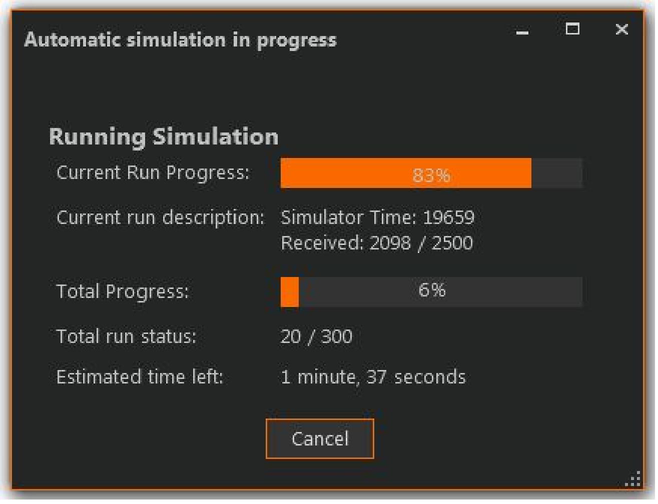
Appendix 19 - About Screen



ABOUT

Ogen Malek	Ogenma64@Gmail.com
Roy Goldstein	Roymail2@Gmail.com
Nir Tsabar	Sjustnir@Gmail.com
Oren Shmuel	OrenSmo@Gmail.com

Appendix 20 - Automatic Simulation In Progress Screen



Automatic simulation in progress

Running Simulation

Current Run Progress: 83%

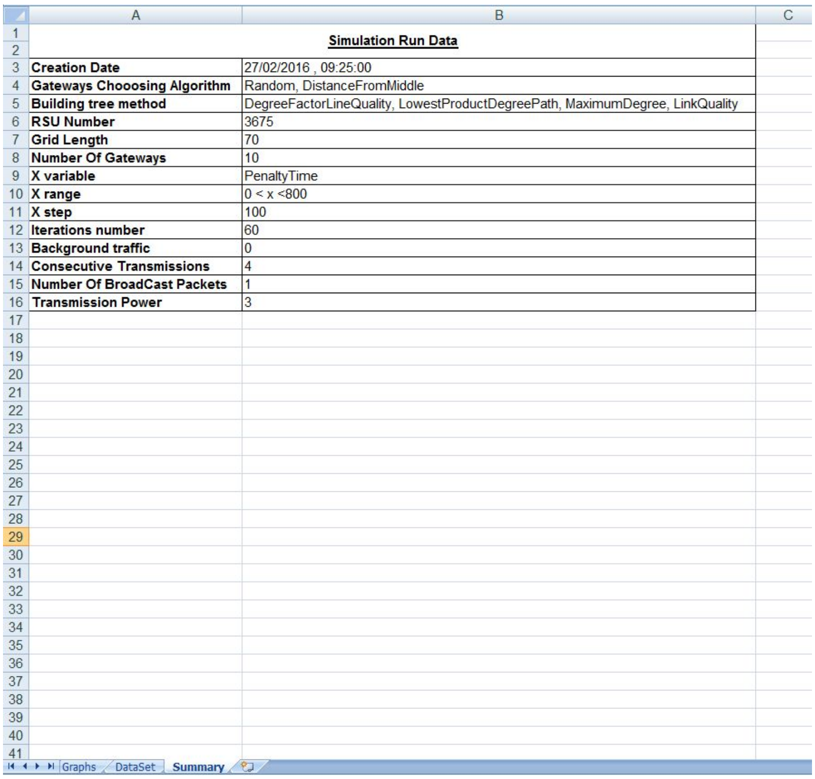
Current run description: Simulator Time: 19659
Received: 2098 / 2500

Total Progress: 6%

Total run status: 20 / 300

Estimated time left: 1 minute, 37 seconds

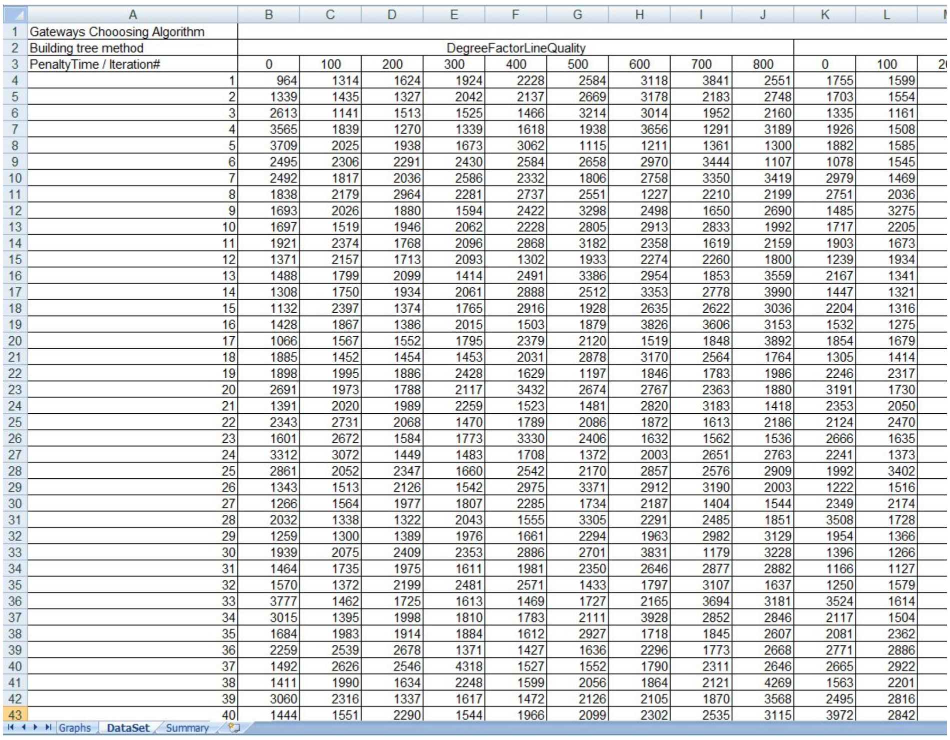
Appendix 21 - Excel Summary Sheet



Simulation Run Data

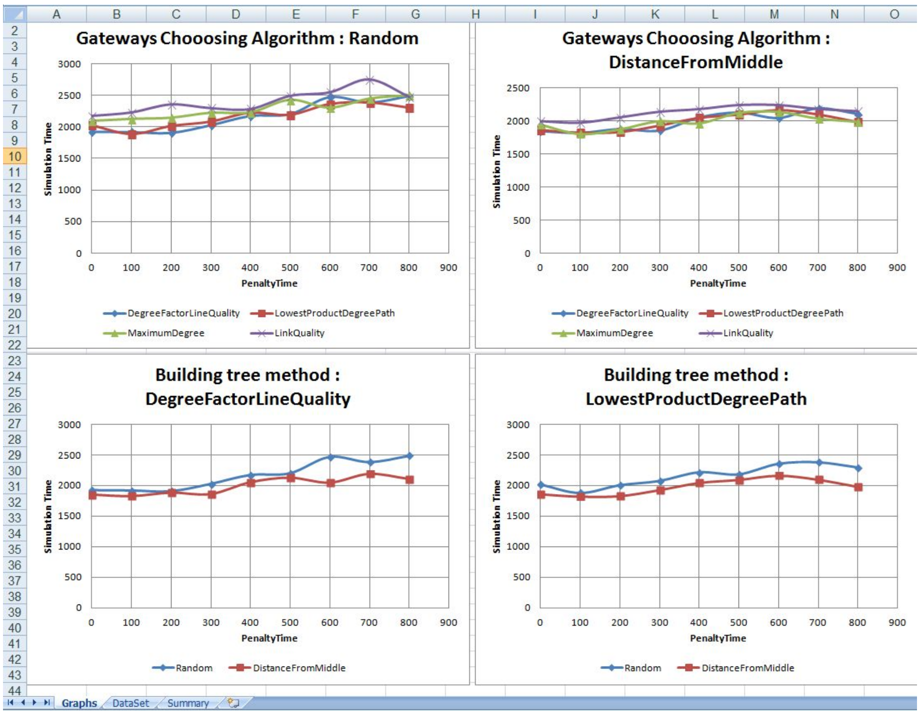
Creation Date	27/02/2016 , 09:25:00
Gateways Chooosing Algorithm	Random, DistanceFromMiddle
Building tree method	DegreeFactorLineQuality, LowestProductDegreePath, MaximumDegree, LinkQuality
RSU Number	3675
Grid Length	70
Number Of Gateways	10
X variable	PenaltyTime
X range	0 < x <800
X step	100
Iterations number	60
Background traffic	0
Consecutive Transmissions	4
Number Of BroadCast Packets	1
Transmission Power	3

Appendix 22 - Excel DataSet Sheet

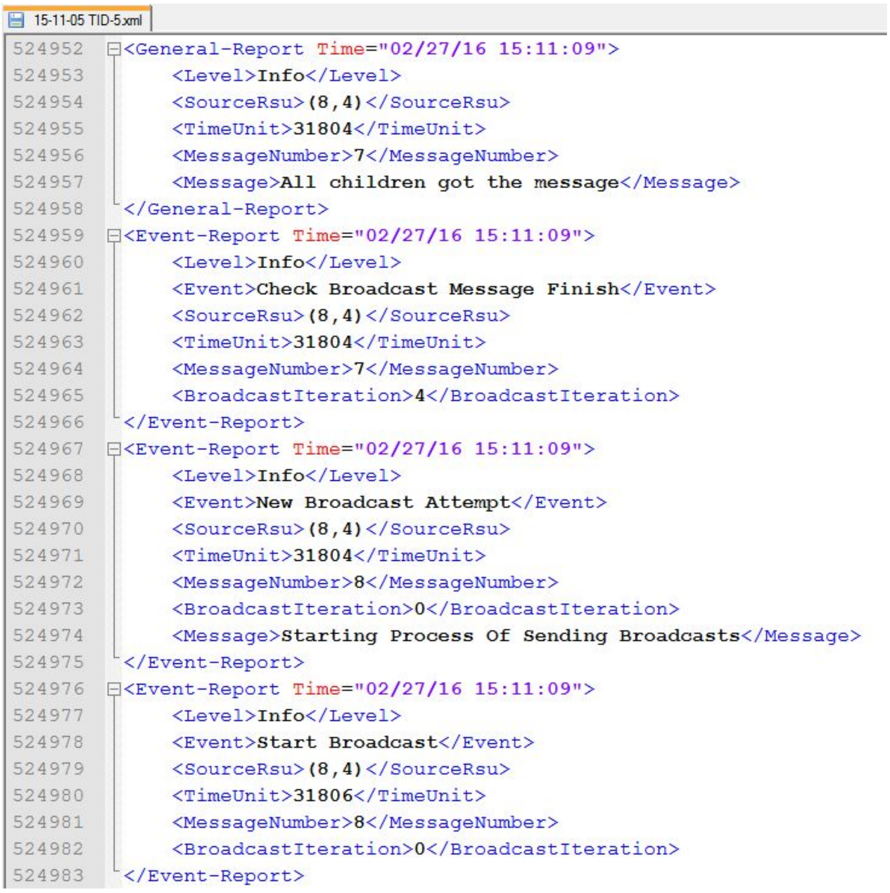


	A	B	C	D	E	F	G	H	I	J	K	L
1	Gateways Chooosing Algorithm											
2	Building tree method	DegreeFactorLineQuality										
3	PenaltyTime / Iteration#	0	100	200	300	400	500	600	700	800	0	100
4	1	964	1314	1624	1924	2228	2584	3118	3841	2551	1755	1599
5	2	1339	1435	1327	2042	2137	2669	3178	2183	2748	1703	1554
6	3	2613	1141	1513	1525	1466	3214	3014	1952	2160	1335	1161
7	4	3565	1839	1270	1339	1618	1938	3656	1291	3189	1926	1508
8	5	3709	2025	1938	1673	3062	1115	1211	1361	1300	1882	1585
9	6	2495	2306	2291	2430	2584	2658	2970	3444	1107	1078	1545
10	7	2492	1817	2036	2586	2332	1806	2758	3350	3419	2979	1469
11	8	1838	2179	2964	2281	2737	2551	1227	2210	2199	2751	2036
12	9	1693	2026	1880	1594	2422	3298	2498	1650	2690	1485	3275
13	10	1697	1519	1946	2062	2228	2805	2913	2833	1992	1717	2205
14	11	1921	2374	1768	2096	2868	3182	2358	1619	2159	1903	1673
15	12	1371	2157	1713	2093	1302	1933	2274	2260	1800	1239	1934
16	13	1488	1799	2099	1414	2491	3386	2954	1853	3559	2167	1341
17	14	1308	1750	1934	2061	2888	2512	3353	2778	3990	1447	1321
18	15	1132	2397	1374	1765	2916	1928	2635	2622	3036	2204	1316
19	16	1428	1867	1386	2015	1503	1879	3826	3606	3153	1532	1275
20	17	1066	1567	1552	1795	2379	2120	1519	1848	3892	1854	1679
21	18	1885	1452	1454	1453	2031	2878	3170	2564	1764	1305	1414
22	19	1898	1995	1886	2428	1629	1197	1846	1783	1986	2246	2317
23	20	2691	1973	1788	2117	3432	2674	2767	2363	1880	3191	1730
24	21	1391	2020	1989	2259	1523	1481	2820	3183	1418	2353	2050
25	22	2343	2731	2068	1470	1789	2086	1872	1613	2186	2124	2470
26	23	1601	2672	1584	1773	3330	2406	1632	1562	1536	2666	1635
27	24	3312	3072	1449	1483	1708	1372	2003	2651	2763	2241	1373
28	25	2861	2052	2347	1660	2542	2170	2857	2576	2909	1992	3402
29	26	1343	1513	2126	1542	2975	3371	2912	3190	2003	1222	1516
30	27	1266	1564	1977	1807	2285	1734	2187	1404	1544	2349	2174
31	28	2032	1338	1322	2043	1555	3305	2291	2485	1851	3508	1728
32	29	1259	1300	1389	1976	1661	2294	1963	2982	3129	1954	1366
33	30	1939	2075	2409	2353	2886	2701	3831	1179	3228	1396	1266
34	31	1464	1735	1975	1611	1981	2350	2646	2877	2882	1166	1127
35	32	1570	1372	2199	2481	2571	1433	1797	3107	1637	1250	1579
36	33	3777	1462	1725	1613	1469	1727	2165	3694	3181	3524	1614
37	34	3015	1395	1998	1810	1783	2111	3928	2852	2846	2117	1504
38	35	1684	1983	1914	1884	1612	2927	1718	1845	2607	2081	2362
39	36	2259	2539	2678	1371	1427	1636	2296	1773	2668	2771	2886
40	37	1492	2626	2546	4318	1527	1552	1790	2311	2646	2665	2922
41	38	1411	1990	1634	2248	1599	2056	1864	2121	4269	1563	2201
42	39	3060	2316	1337	1617	1472	2126	2105	1870	3568	2495	2816
43	40	1444	1551	2290	1544	1966	2099	2302	2535	3115	3972	2842

Appendix 23 - Excel Graphs Sheet



Appendix 24 - Log File



```
<General-Report Time="02/27/16 15:11:09">
    <Level>Info</Level>
    <SourceRsu>(8,4)</SourceRsu>
    <TimeUnit>31804</TimeUnit>
    <MessageNumber>7</MessageNumber>
    <Message>All children got the message</Message>
</General-Report>
<Event-Report Time="02/27/16 15:11:09">
    <Level>Info</Level>
    <Event>Check Broadcast Message Finish</Event>
    <SourceRsu>(8,4)</SourceRsu>
    <TimeUnit>31804</TimeUnit>
    <MessageNumber>7</MessageNumber>
    <BroadcastIteration>4</BroadcastIteration>
</Event-Report>
<Event-Report Time="02/27/16 15:11:09">
    <Level>Info</Level>
    <Event>New Broadcast Attempt</Event>
    <SourceRsu>(8,4)</SourceRsu>
    <TimeUnit>31804</TimeUnit>
    <MessageNumber>8</MessageNumber>
    <BroadcastIteration>0</BroadcastIteration>
    <Message>Starting Process Of Sending Broadcasts</Message>
</Event-Report>
<Event-Report Time="02/27/16 15:11:09">
    <Level>Info</Level>
    <Event>Start Broadcast</Event>
    <SourceRsu>(8,4)</SourceRsu>
    <TimeUnit>31806</TimeUnit>
    <MessageNumber>8</MessageNumber>
    <BroadcastIteration>0</BroadcastIteration>
</Event-Report>
```