Correlation animation tool

The correlation animation tool is created for a visual inspection of the correlation in our network test. The program takes in a CSV file, where the first line is used to setup the simulation:

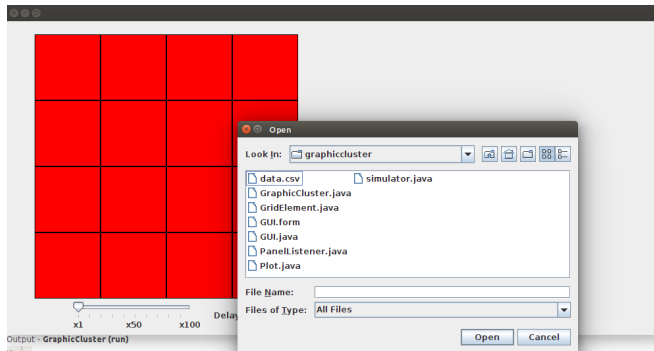<number of lines>, <Duration of the real test (in seconds)>, <number> , , , Number of element in grid -2.

Eks of 4 x 4 grid (16 raspberries), with 75868 measurements measured over one minutes:

75868,60,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

After this line, over 1 represent a received packet and 0 represent a lost packet:

0,1,0,1,0,1,0,1,1,1,1,1,1,1,0,0

The CSV file will be selected when starting up the programs:

When the program receives a 0 the squares will be colored red, else they will be green.
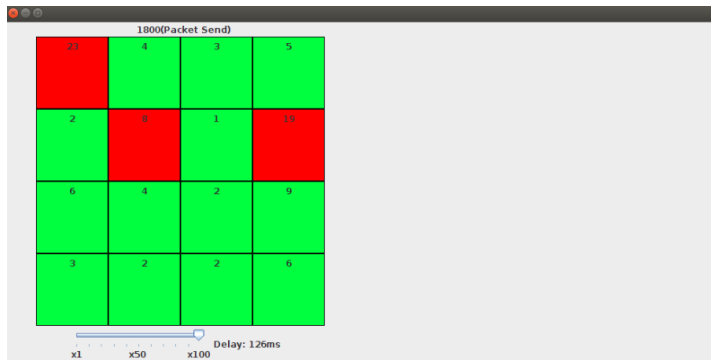
Since it can be hard to see the clusters of red squares, the bottom bar adds delay to the original time.

For a extra feature the nodes can be pressed, giving specific information about that node.

The Packet Loss Trace graph live updates when information is read from the CSV file.

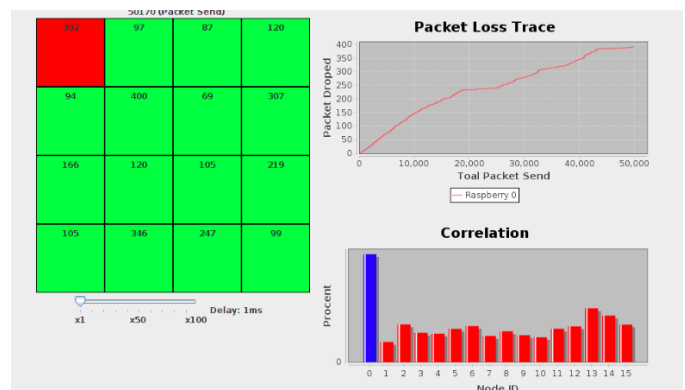The blue bar is the node itself and the red bars are its neighbors.

When talking about the correlation, we defined the correlation as:

$$cell(i,j) = \frac{(Number\ of\ same\ packet\ lost \in both\ i \wedge j)}{(Number\ of\ total\ packet\ lost \in both\ i \wedge j)}$$

This is the case in our simulation, since every node can reach each other. However in a real scenario, this is not true.
The correlation is described for a given node, n, and n's neighbors (nodes that can provide n with information).

So looking at n, what is the correlation in the subset of n and its neighbors.
This is not implemented in the simulation tool.

Code

The code consist of 5 classes as can be shown in the UML diagram.

Plot

- 🔖 ~int raspName
- 🔖 ~ChartPanel chartPanel
- 🔖 ~XYSeries series

- ◆ +Plot(int raspName)
- ● +void plotMe()
- ● +ChartPanel getChartPanel()
- ● +void addData(int time, int packetDroped)

1..1
plot

GraphicCluster

- 🔖 -JFrame frame
- 🔖 -GridElement[][] elements
- 🔖 -JLabel[][] labels
- 🔖 -JLabel delayHolder
- 🔖 -JLabel packetHolder
- 🔖 -JSlider slider
- 🔖 -JFileChooser fileChooser
- 🔖 ~final int SPEED_MIN
- 🔖 ~final int SPEED_MAX

- ◆ +GraphicCluster()
- ● -void initComponents()
- ● +JSlider getSlider()
- ● -void createGrid()
- ● +static void main(String args)

creates

1..*

GridElement

- 🔖 -int width
- 🔖 -int height
- 🔖 -int pos_x
- 🔖 -int pos_y
- 🔖 -int name
- 🔖 -int packetDroped
- 🔖 -int packetRecived
- 🔖 -int correlation
- 🔖 ~JPanel panel
- 🔖 ~JLabel label
- 🔖 ~Plot plot
- 🔖 ~ChartPanel chartPanel

- ◆ +GridElement(int width, int height, int pos_x, int pos_y, int name)
- ● +JPanel initGridElement()
- ● +void setColor(String color)
- ● +int getPacketDroped()
- ● +void setPacketDroped(int packetDroped)
- ● +int getPacketRecived()
- ● +void setPacketRecived(int packetRecived)
- ● +void addLabel(JLabel label)
- ● +JLabel getLabel()
- ● +void setLabelText()
- ● +void setPlot()
- ● +Plot getPlot()
- ● +void setPlotVisibility(boolean visible)
- ● +int getCorrelation()
- ● +void setCorrelation(int correlation)

1..* Update

Start
1..1

simulator

- 🔲 +GridElement[][] elements
- 🔖 -JSlider slider
- 🔖 -String csvFile
- 🔖 -JLabel delayHolder
- 🔖 -JLabel packetHolder
- 🔖 -int NUMBER_OF_LINES
- 🔖 -int EXECUTION_TIME

- ◆ +simulator(GridElement[][] elements, JSlider slider, String csvFile, JLabel delayHolder, JLabel packetHolder)
- ● +void run()
- ● +int checkCorrelation(int x, int y, String[][] raspberry)

1..*
Listens

PanelListener

- 🔖 ~GridElement elements

- ◆ +PanelListener(GridElement[][] elements)
- ● +void panelClicked(java.awt.event.MouseEvent evt)