

PViz

A visualization tool for PeerSim

PViz

- A visualizer for PeerSim simulations.
- Allows the user to visually examine multi-agent, P2P simulations and “replay” their effects, even in situations where node number is large.
- Fully integrated with PeerSim engine, allows for various user-defined protocols to be visualised
- Chord is used as a test showcase since it is easy to implement and understand.

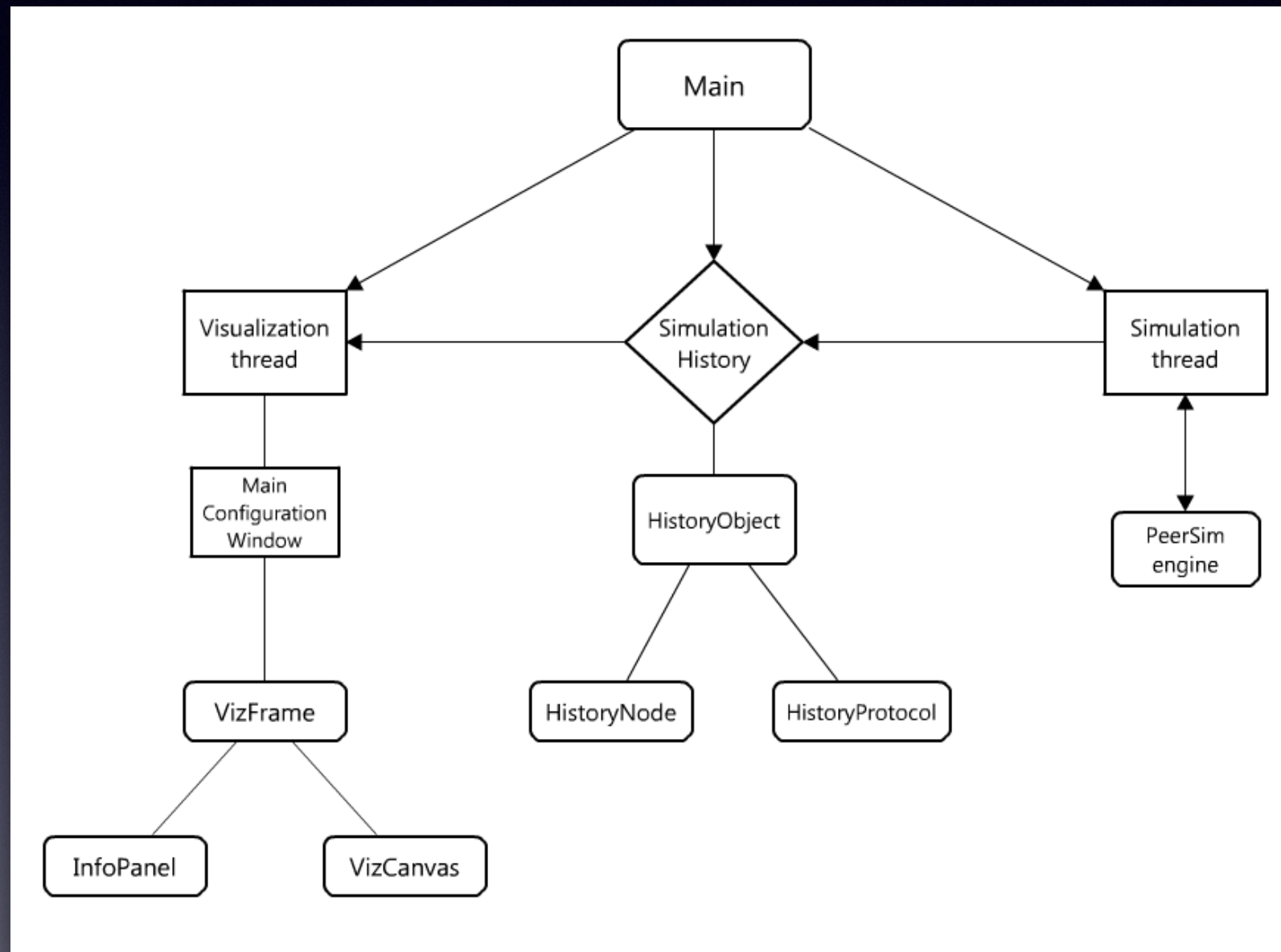
PViz

- Expands PeerSim, almost no simulator code is altered, P-viz sits on top of simulation engine
- Allows recording of protocol changes (e.g. insert/remove/alter)
- Complete network action history is passed to the visualisation thread
- Information presented to the user

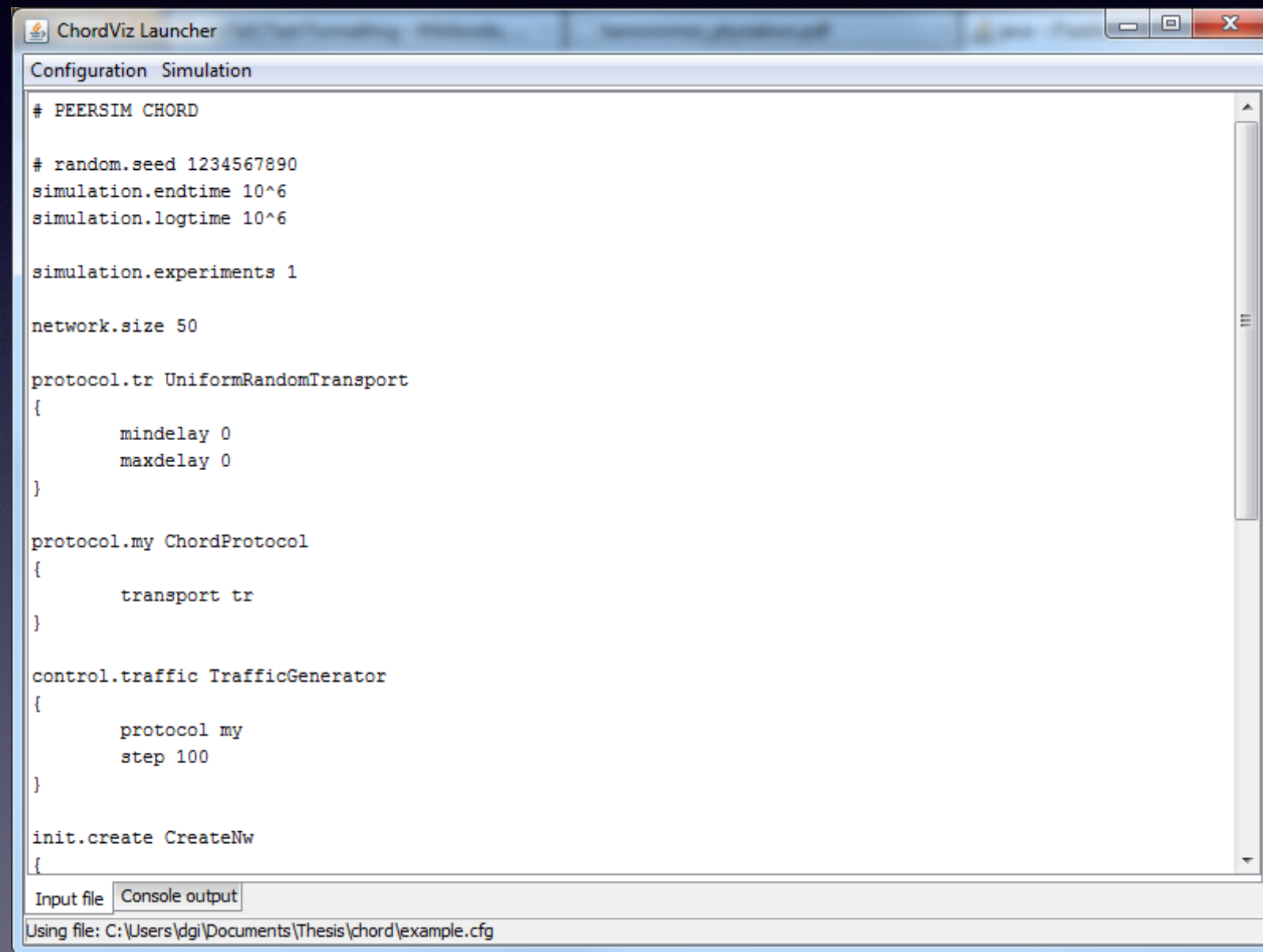
PViz

- Visualisation has a complete simulation history loaded into memory (minimal information logging so that even large simulations can be replayed)
- Creates a user interface with said information
- User is able to interact, load another simulation and so on

How it works



PViz: Step-by-step



The screenshot shows a window titled "ChordViz Launcher" with a tabbed interface. The "Configuration" tab is active, displaying a text area with a configuration file. The file content is as follows:

```
# PEERSIM CHORD

# random.seed 1234567890
simulation.endtime 10^6
simulation.logtime 10^6

simulation.experiments 1

network.size 50

protocol.tr UniformRandomTransport
{
    mindelay 0
    maxdelay 0
}

protocol.my ChordProtocol
{
    transport tr
}

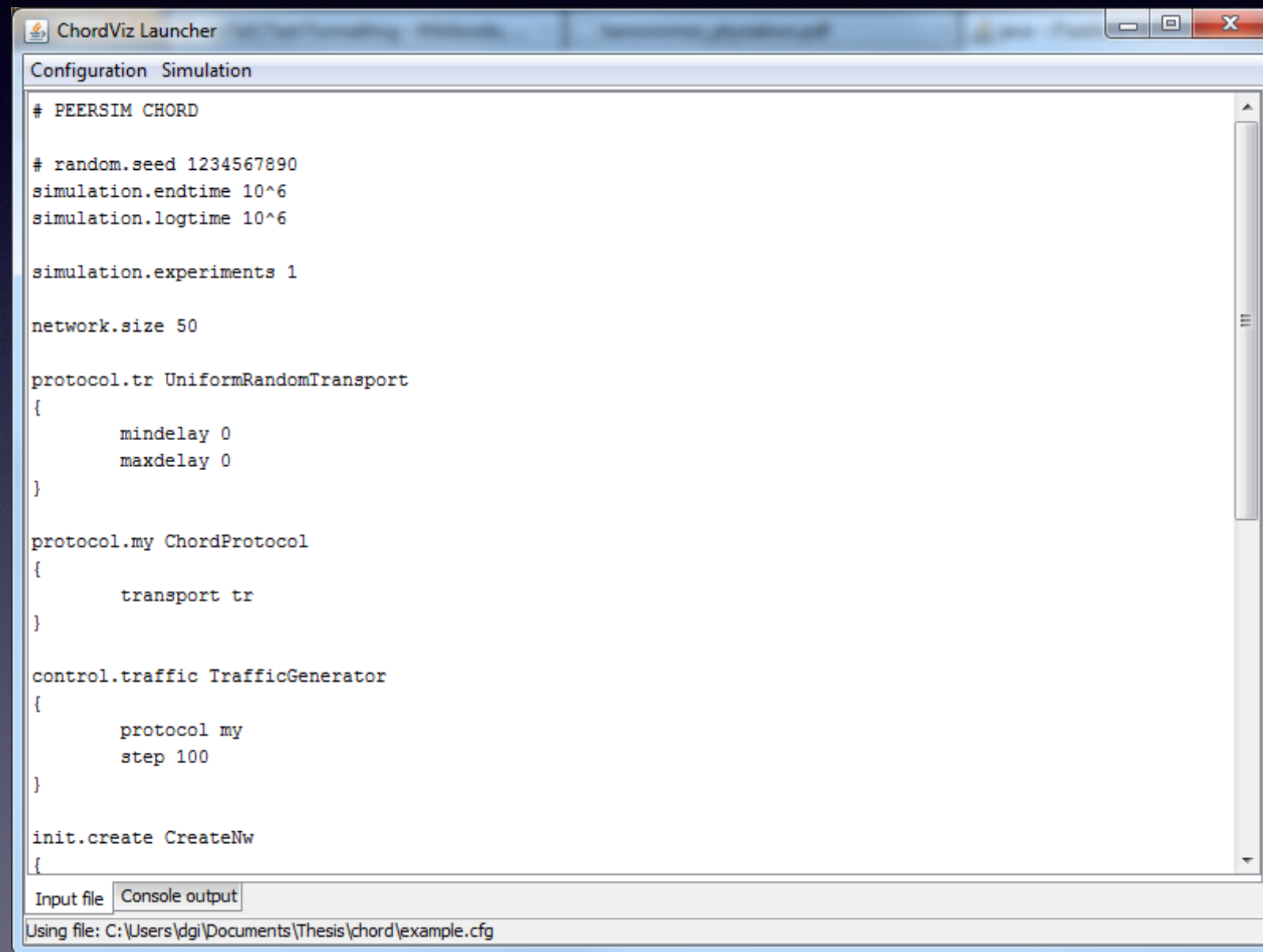
control.traffic TrafficGenerator
{
    protocol my
    step 100
}

init.create CreateNw
{
```

At the bottom of the window, there are two tabs: "Input file" and "Console output". The "Input file" tab is selected, and it shows the file path: "Using file: C:\Users\dgi\Documents\Thesis\chord\example.cfg".

Input
Configuration

PViz: Step-by-step



The screenshot shows a window titled "ChordViz Launcher" with a tabbed interface. The "Configuration" tab is active, displaying a text area with a configuration file. The file content is as follows:

```
# PEERSIM CHORD

# random.seed 1234567890
simulation.endtime 10^6
simulation.logtime 10^6

simulation.experiments 1

network.size 50

protocol.tr UniformRandomTransport
{
    mindelay 0
    maxdelay 0
}

protocol.my ChordProtocol
{
    transport tr
}

control.traffic TrafficGenerator
{
    protocol my
    step 100
}

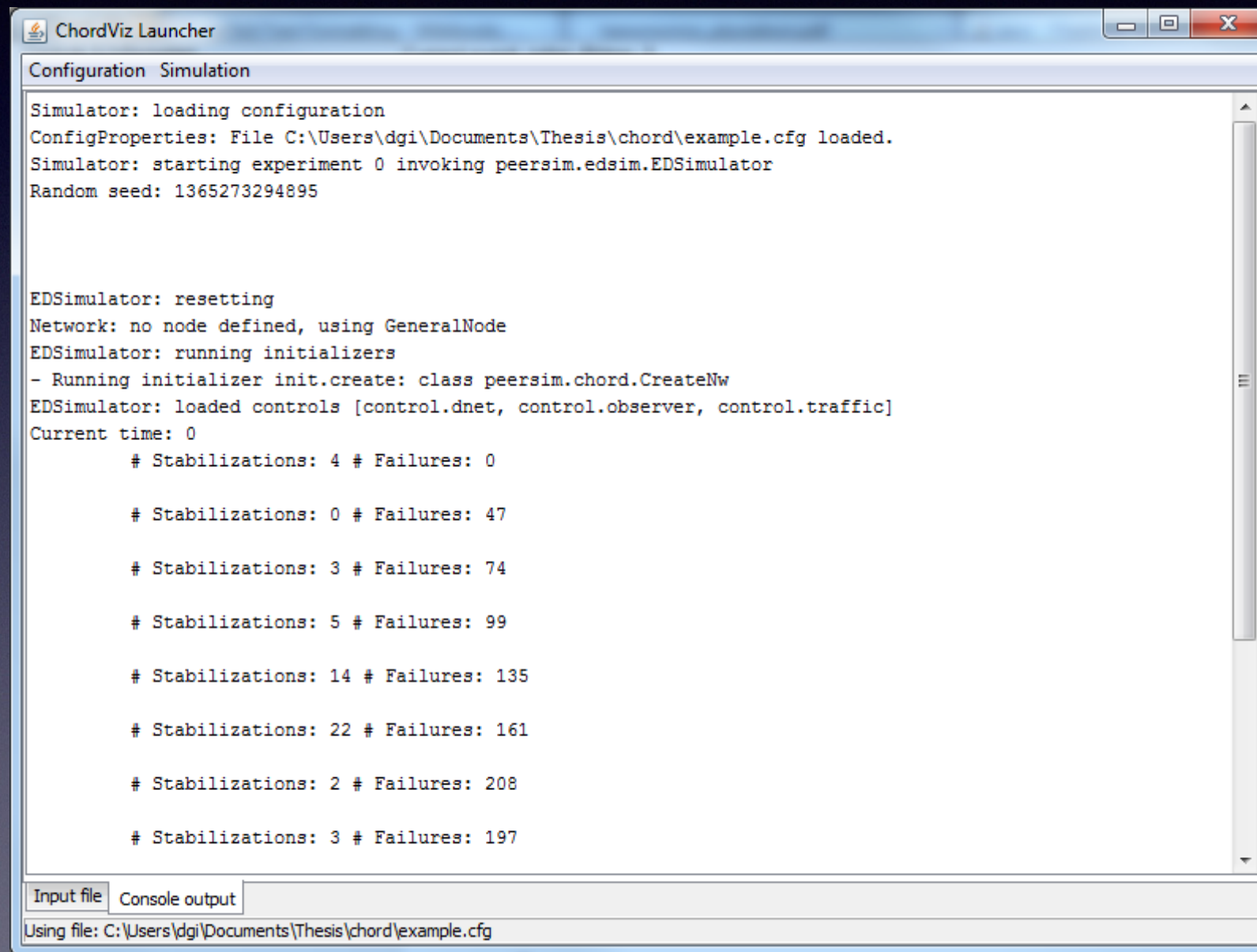
init.create CreateNw
{
```

At the bottom of the window, there are two tabs: "Input file" and "Console output". The "Input file" tab is selected, showing the file path: "Using file: C:\Users\dgi\Documents\Thesis\chord\example.cfg".

Input Configuration

- Accepts PeerSim files
- Allows customisation

PViz: Step-by-step



The screenshot shows a Windows application window titled "ChordViz Launcher". It has a menu bar with "Configuration" and "Simulation". The main area is a text box displaying the following output:

```
Simulator: loading configuration
ConfigProperties: File C:\Users\dgi\Documents\Thesis\chord\example.cfg loaded.
Simulator: starting experiment 0 invoking peersim.edsim.EDSimulator
Random seed: 1365273294895

EDSimulator: resetting
Network: no node defined, using GeneralNode
EDSimulator: running initializers
- Running initializer init.create: class peersim.chord.CreateNw
EDSimulator: loaded controls [control.dnet, control.observer, control.traffic]
Current time: 0
    # Stabilizations: 4 # Failures: 0

    # Stabilizations: 0 # Failures: 47

    # Stabilizations: 3 # Failures: 74

    # Stabilizations: 5 # Failures: 99

    # Stabilizations: 14 # Failures: 135

    # Stabilizations: 22 # Failures: 161

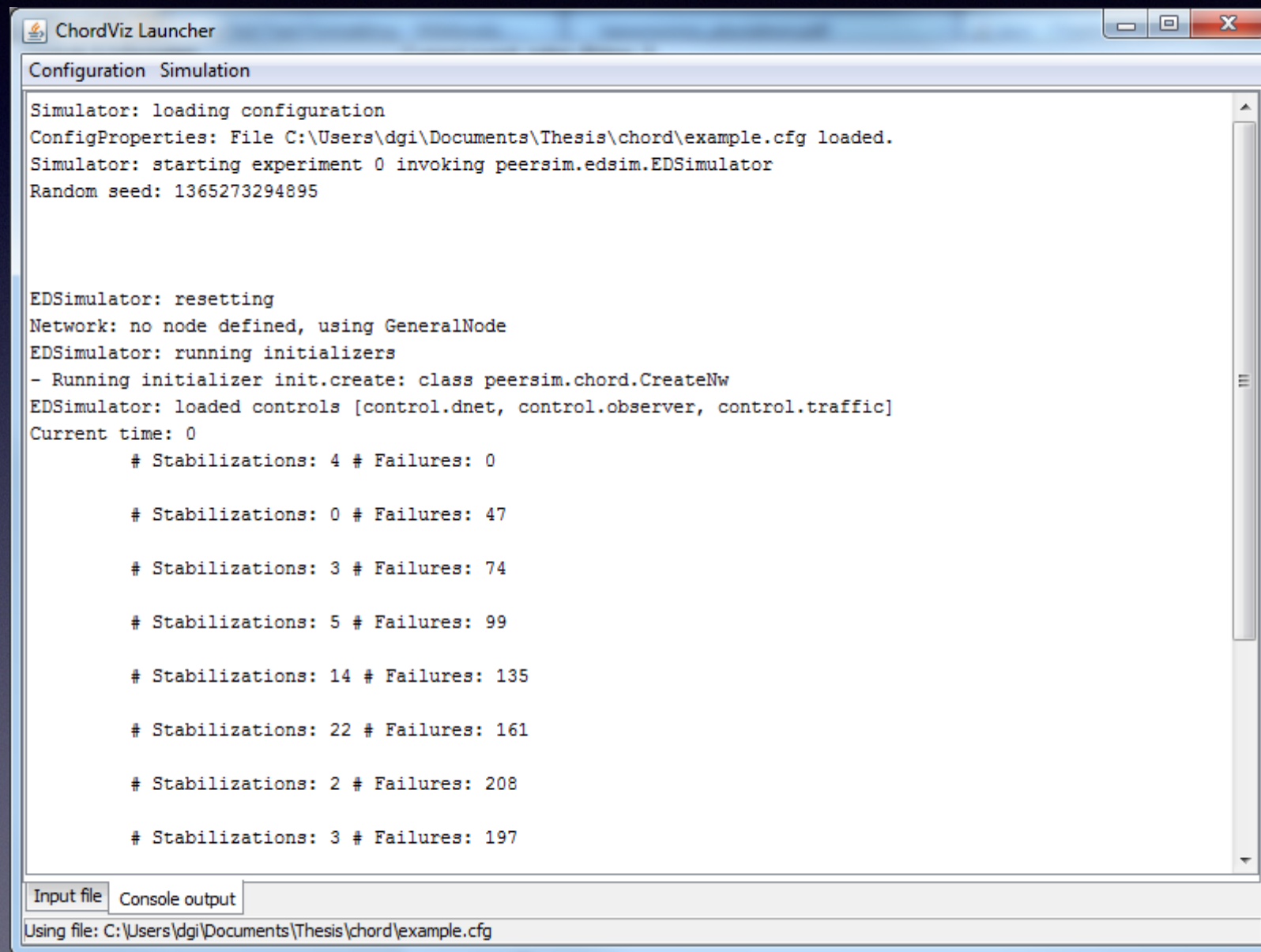
    # Stabilizations: 2 # Failures: 208

    # Stabilizations: 3 # Failures: 197
```

At the bottom, there are tabs for "Input file" and "Console output". The "Input file" tab is selected, showing the path: "Using file: C:\Users\dgi\Documents\Thesis\chord\example.cfg".

Command-line output

PViz: Step-by-step



The screenshot shows a Windows application window titled "ChordViz Launcher". It has two tabs: "Configuration" and "Simulation". The "Simulation" tab is active, displaying a text area with the following output:

```
Simulator: loading configuration
ConfigProperties: File C:\Users\dgi\Documents\Thesis\chord\example.cfg loaded.
Simulator: starting experiment 0 invoking peersim.edsim.EDSimulator
Random seed: 1365273294895

EDSimulator: resetting
Network: no node defined, using GeneralNode
EDSimulator: running initializers
- Running initializer init.create: class peersim.chord.CreateNw
EDSimulator: loaded controls [control.dnet, control.observer, control.traffic]
Current time: 0
    # Stabilizations: 4 # Failures: 0

    # Stabilizations: 0 # Failures: 47

    # Stabilizations: 3 # Failures: 74

    # Stabilizations: 5 # Failures: 99

    # Stabilizations: 14 # Failures: 135

    # Stabilizations: 22 # Failures: 161

    # Stabilizations: 2 # Failures: 208

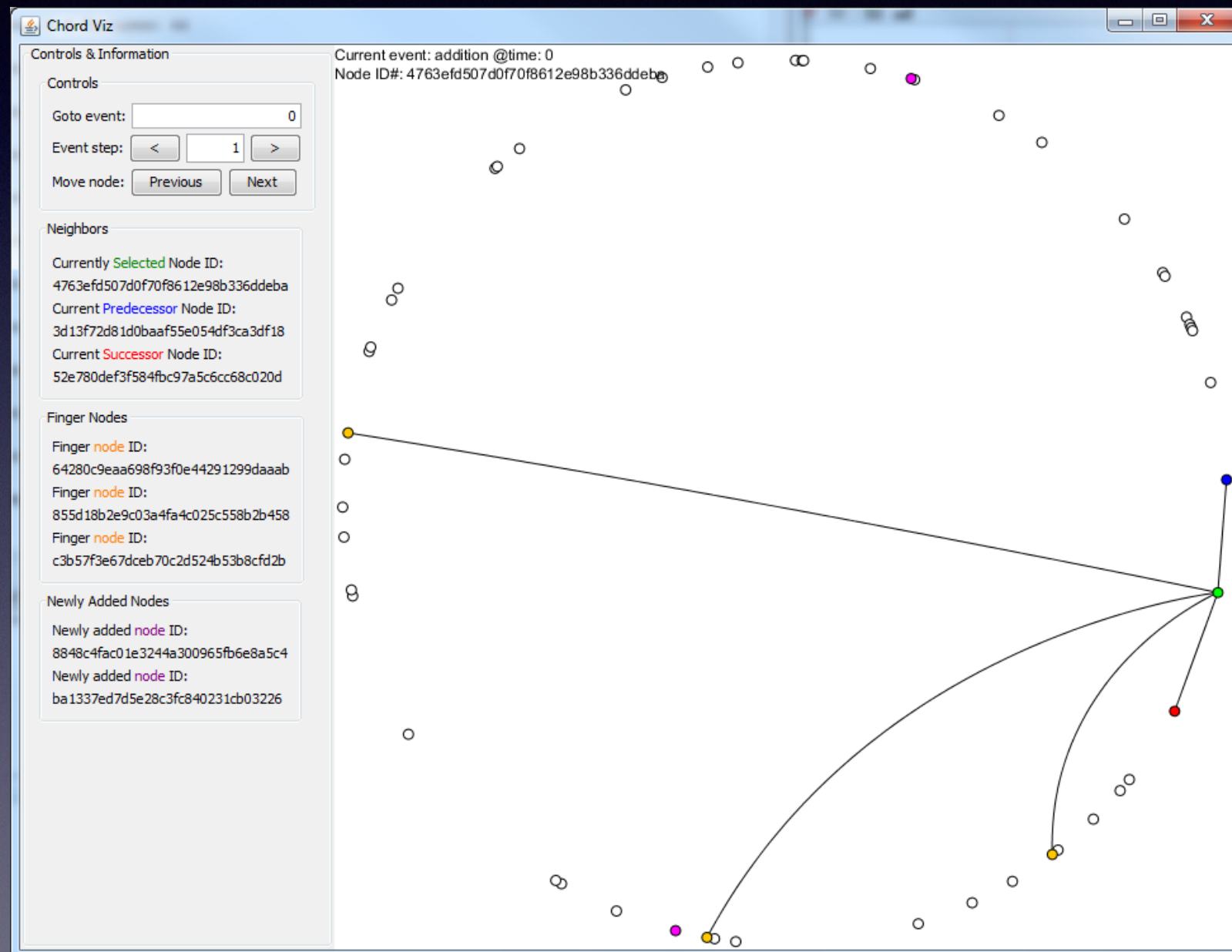
    # Stabilizations: 3 # Failures: 197
```

At the bottom of the window, there are two tabs: "Input file" and "Console output". The "Input file" tab is selected, showing the path: "Using file: C:\Users\dgi\Documents\Thesis\chord\example.cfg".

Command-line output

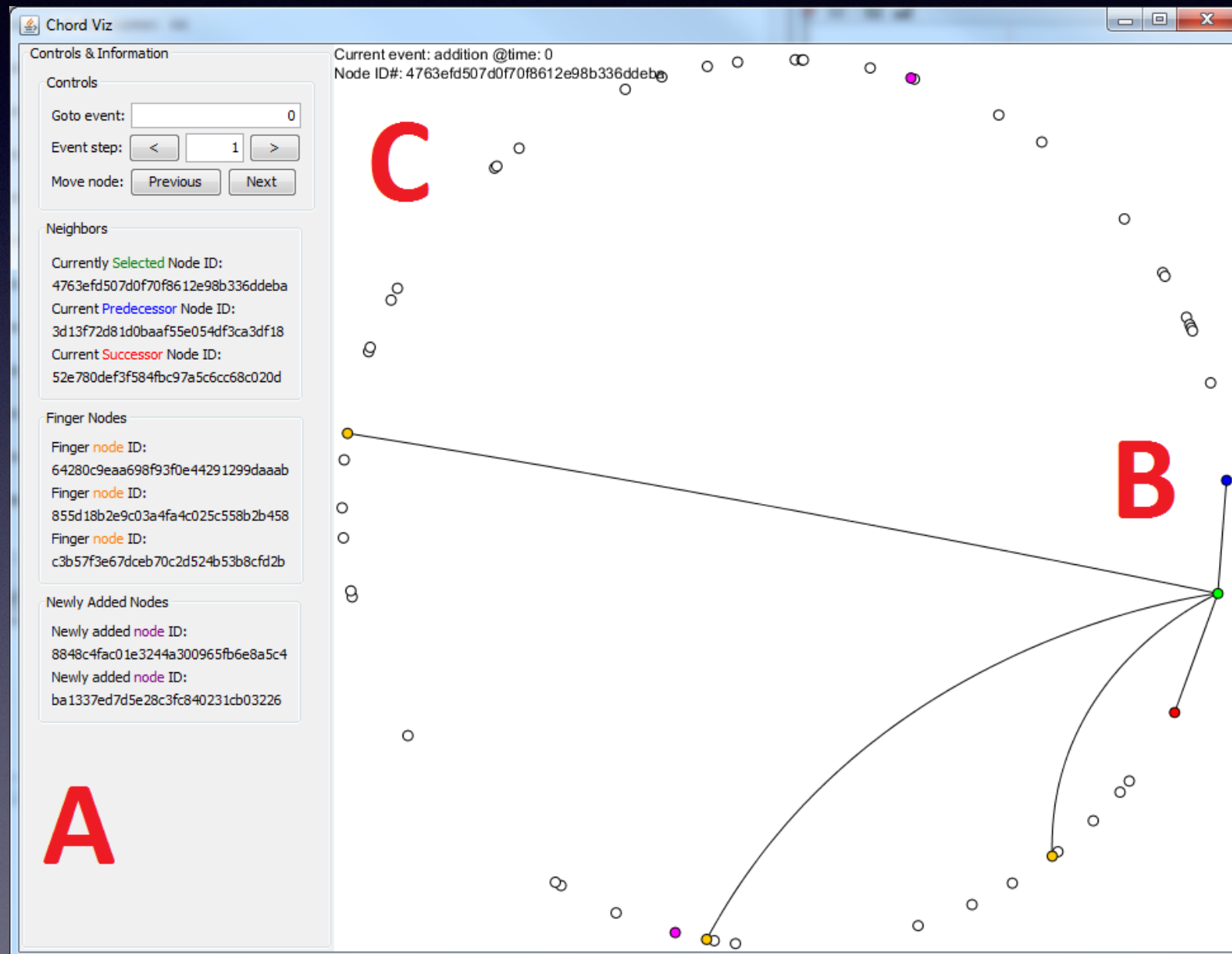
- PeerSim output log
- Provides additional information
- Accompanies visualisation

PViz: Step-by-step



Visualization Window

PViz: Step-by-step



Visualization Window

A. Information Panel

B. Interactive Visualisation

C. Current event

PViz: Step-by-step

Controls & Information

Controls

Goto event:

Event step:

Move node:

Neighbors

Currently **Selected** Node ID:
425b58bcc8d8cec4d0773044b97556ab

Current **Predecessor** Node ID:
3f612c76a17220eed5b30338e3f9431a

Current **Successor** Node ID:
5978efecc0cfb3b6899247a38bd00523

Finger Nodes

Finger **node** ID:
5b6ddc13e0f0330c83f3da8c2df0408d

Finger **node** ID:
7d709948a9adc3827512e5826473ac84

Finger **node** ID:
c24a415e9a5168c5081bc5ab46287a8f

Newly Added Nodes

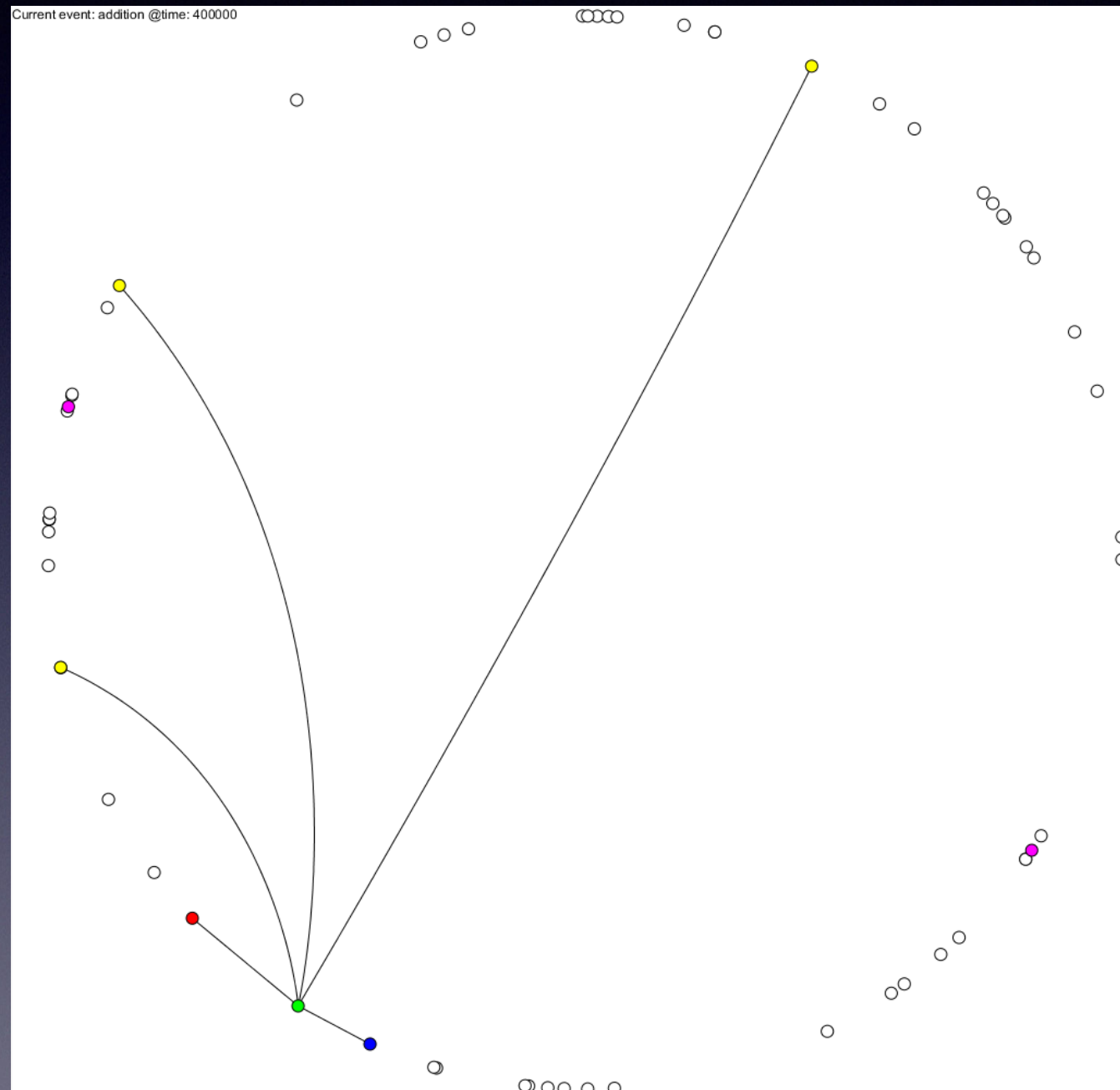
Newly added **node** ID:
73226101dc20c61c4269b55be848b09b

Newly added **node** ID:
67c4c107ea8c97882525a123dd8b4ac0

A) Information Panel

- Control event, node, simulation sequence
- List neighbours, fingers, new nodes via mouse

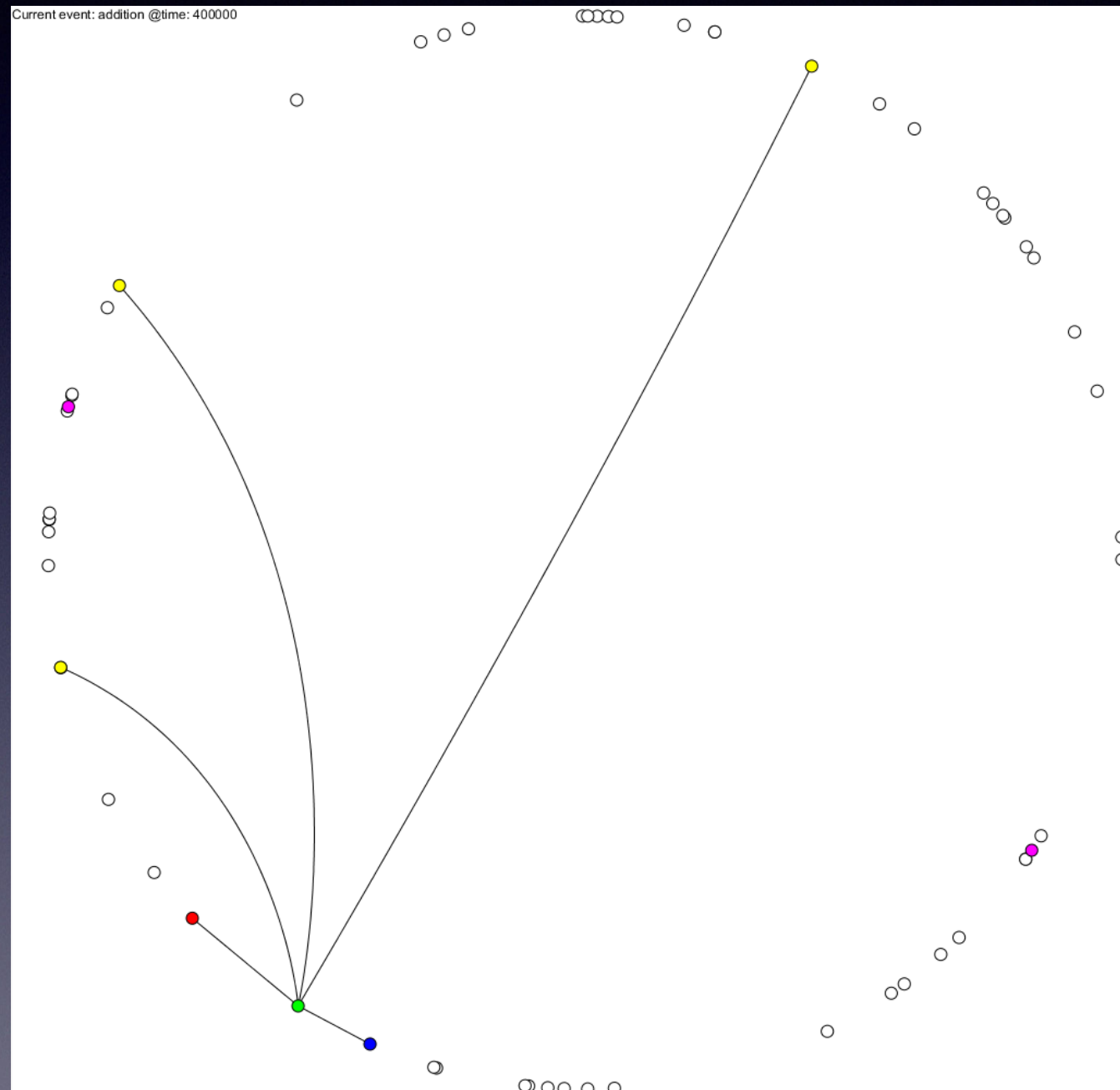
PViz: Step-by-step



B) Main Visualization
Node selection
reveals neighbours,
successor,
predecessor

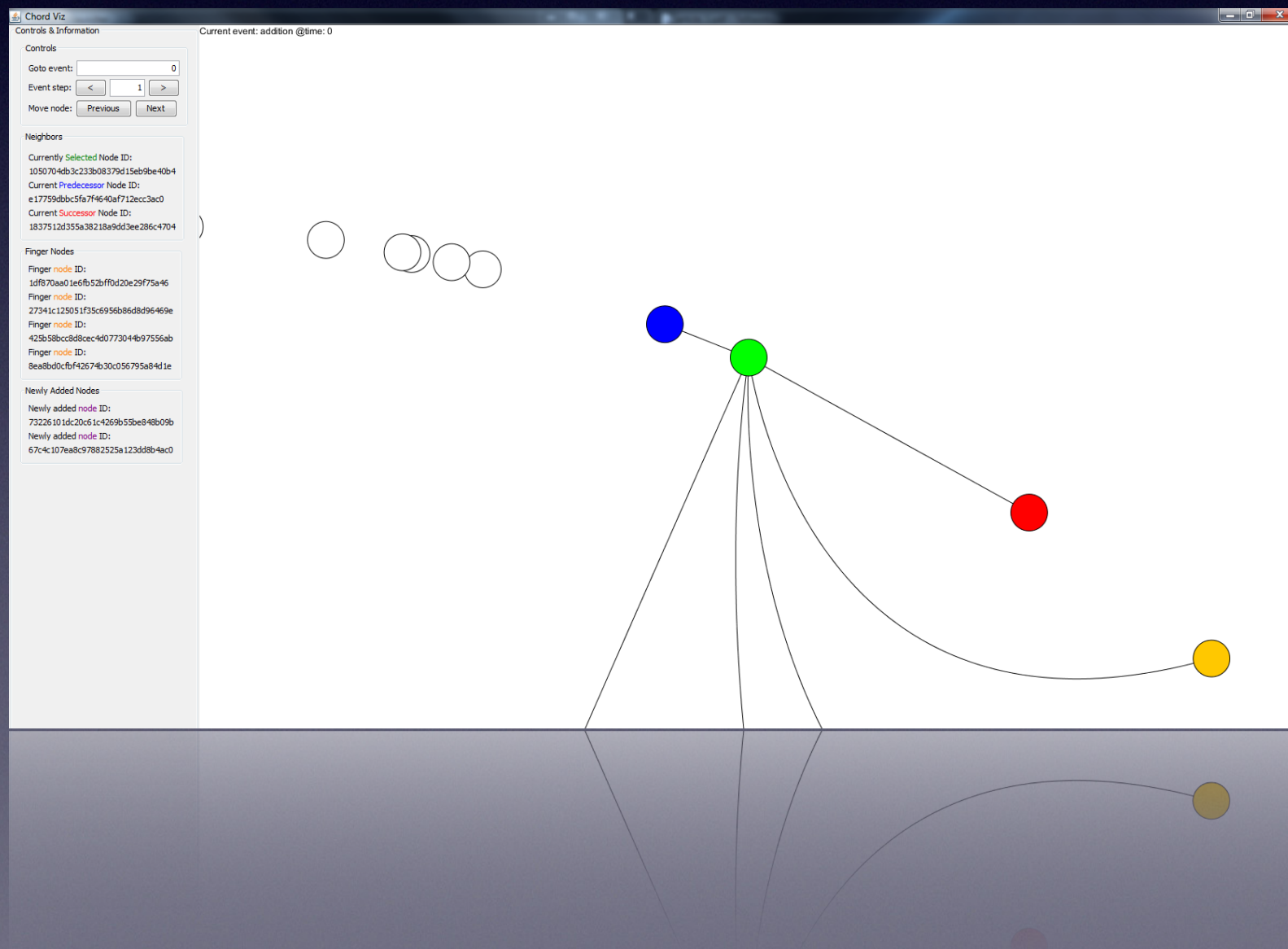
- Successor: red
- Predecessor: blue
- Fingers: yellow

PViz: Step-by-step



- B) Main Visualization
- Events shown on top-left corner
 - On event change, new nodes are highlighted
- New nodes: magenta

PViz: Step-by-step



Zooming

- Visualization allows dynamic zoom in or zoom out
- User is able to cherry-pick parts of the network to examine

PViz:Questions

- Any questions?

PViz

Thank you!