Query Cycle Simulator Tutorial

Introduction

This document serves as a tutorial for the Query Cycle Simulator, a file sharing p2p network simulator. The program is a tool for modeling file downloads and identifies certain behavioral properties such as a peer's affinity towards a particular category of files, or a peer's malicious intent. The simulator currently uses the EigenTrust algorithm to assign peer trust values based on the success of a file upload. A description of this algorithm can be found in "Simulating a File-Sharing P2P Network." In order to illustrate the algorithm's effects, the GUI uses peer trust values to color code the network nodes.

Simulator Initialization

The following figure represents the initial view displayed by the simulator.

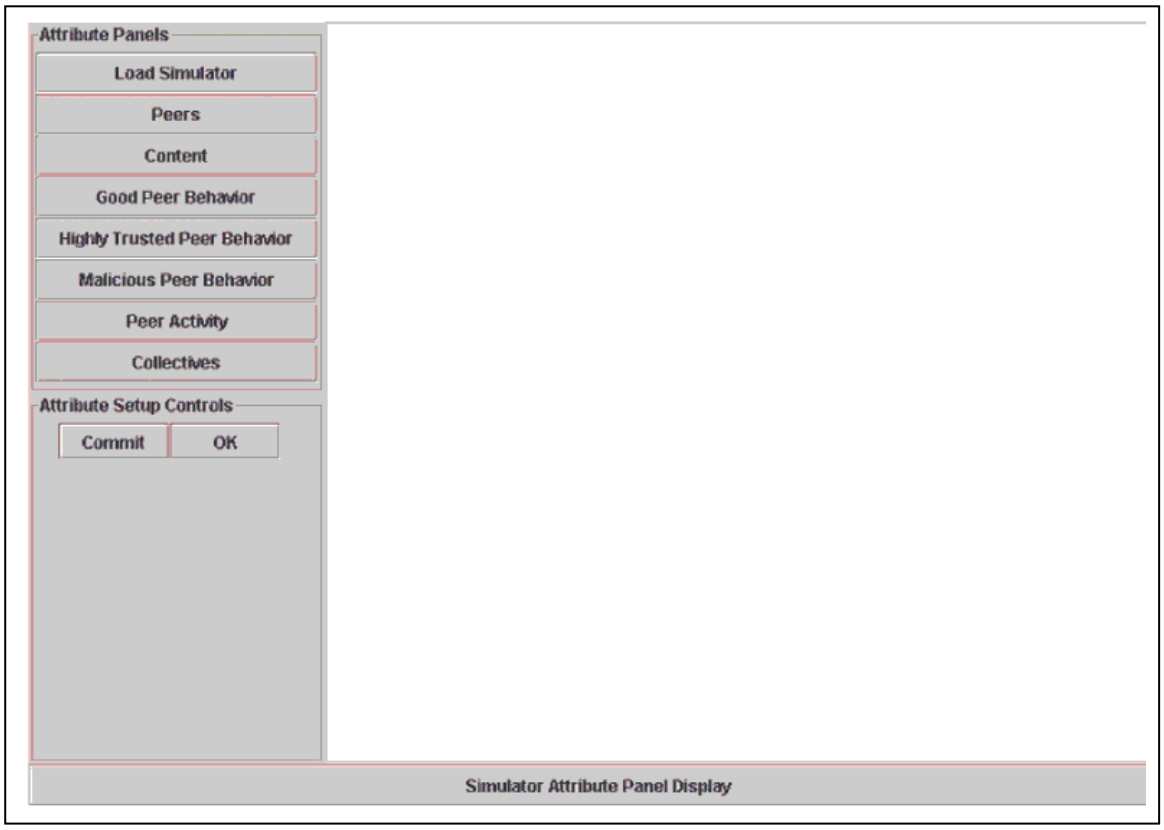

Figure 1.

There are two groups of controls that allow you to setup a desired network configuration. The first group is the Attribute Panels, which contains command buttons. These buttons

will display panels at the bottom of the view. The following describes the various initialization panels.

- Load Simulator: This displays a view that allows you to select from a list of stored simulations.
- Peers: This view presents a list of input fields used to setup the number of peers and the desired number of neighbors.
- Content: The following is the list of input fields and their descriptions displayed under this panel.
    - Distribution Type – determines how content categories are distributed among peers.
    - Total number categories – total number of categories in the system.
    - Minimum categories per peer – each peer will be assigned at least this many content categories
- Good Peer Behavior: This panel is used to setup a good peer's query response behavior. A peer will use this value to determine how it should choose download sources among query responses. An error fraction is also provided, which represents, on average, the fraction of upload errors a peer will make.
- Highly Trusted Peer Behavior: The view contains an error fraction field with the same semantics as described in Good Peer Behavior. It also has a Category Preference options which are top, bottom and random category. Selecting top or bottom category will initialize the peer's content with the top or bottom 20% content categories.
- Malicious Peer Behavior: This view is the same as the Highly Trusted Peer Behavior.
- Peer Activity: A peer's activity will determine its query behavior and/or uptime. The peer(s) activity can be set by highlighting the peer(s) and selecting the desired activity.
- Collectives: Collectives can be formed among malicious peers. You can setup a malicious collective by highlighting a set of malicious peers and selecting the type of collective and its configuration.

The second group is the Attribute Setup Controls, which contains "Commit" and "OK" buttons. After entering the desired values in an initialization panel, you need to press the "OK" button so that the program can accept these values. Once all values have been set, the commit command will create the simulator and display the simulation view.

Simulation View

After the values have been entered and committed, the simulation view will appear. The primary commands that drive the simulation can be found in the "Simulation Controls" group, shown in Figure 2. The peers and their download history are displayed on the right side of the window.

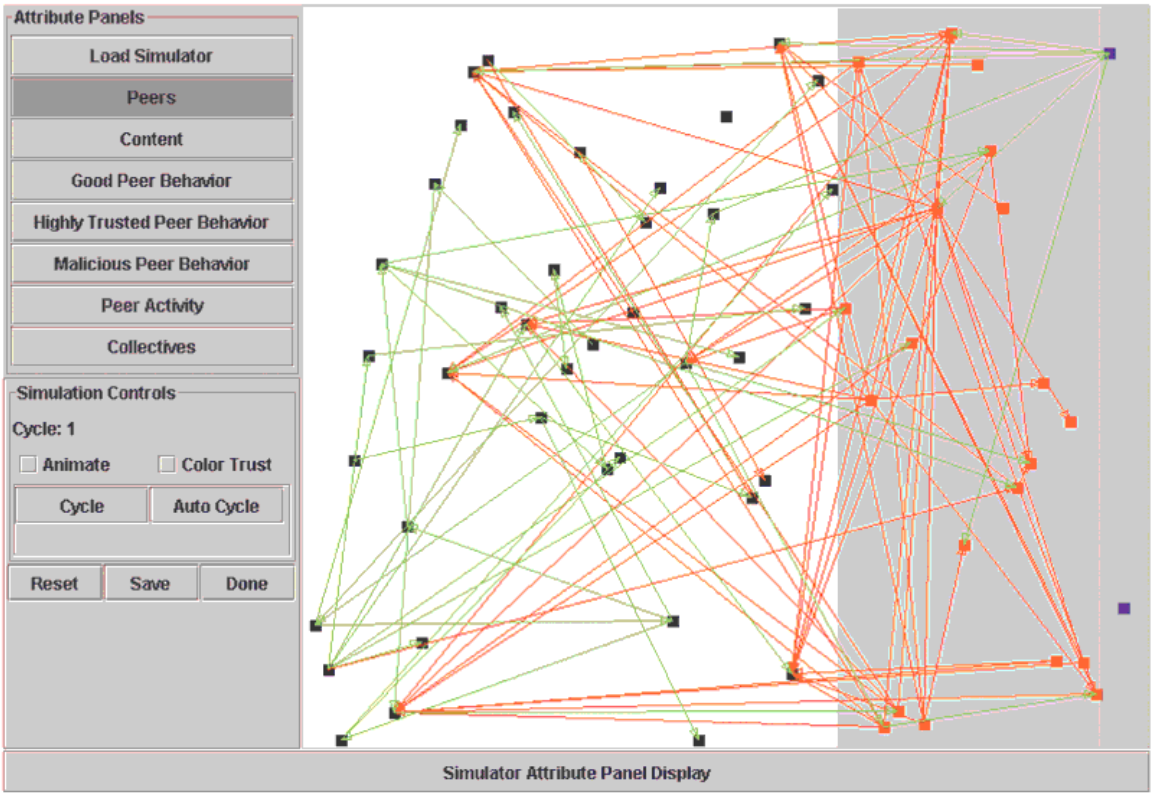

Figure 2.

Simulation Controls

The first item in the simulation controls is the current cycle label, which tracks the current cycle. This is followed by two check boxes labeled "Animate" and "Color Trust". The animate check box causes the peer view to animate peer downloads, once per cycle. The color trust will draw peers according to their global eigentrust values. The following are command buttons "Cycle" and "Auto Cycle". The cycle command will simply step through the simulation one cycle at a time, while auto cycle will continue cycling until "Halt Cycle" is pressed. The final set of commands will reset, save, or dispose of the current simulation.

If you have any questions, please send an email to tcondie@stanford.edu.