SuperViz: An Interactive Visualization of Super-Peer P2P Network

Anthony (Peiqun) Yu
pqyu@cs.ubc.ca

Abstract: The Efficient Clustered Super-Peer P2P network is a novel P2P architecture, which overcomes the efficiency and scalability problems of existing P2P systems. A flooding algorithm, Efficient Flooding Algorithm (EFA) is also introduced to work on this type of network system. However, the Super-Peer network architecture and the behavior of the flooding algorithm are not intuitive. SuperViz project creates a software system that applies various information visualization techniques. With this system, people can interact with the network topology and the routing algorithms, and therefore understand the Super-Peer network architecture and the behavior of the different routing algorithms.

1. Introduction

Peer-to-peer computing is becoming more and more popular nowadays, because of its attractive features, such as self-organization, load balancing, availability, fault tolerance, and anonymity. However, locating content in an efficient and scalable way is one of the most challenging issus in P2P research. Juan Li proposes a novel Efficient Clustered Super-peer (ECSP) P2P model [1]. In this model, Peers are grouped into clusters according to their topological proximity. Peers that have more resource capabilities in a cluster are selected from regular peers to act as cluster leaders. Each cluster has one super-peer. These super-peers are responsible for locating content and maintaining the network structure for client peers. Super-peers are connected with each other, to construct a backbone overlay network (see Figure 1).

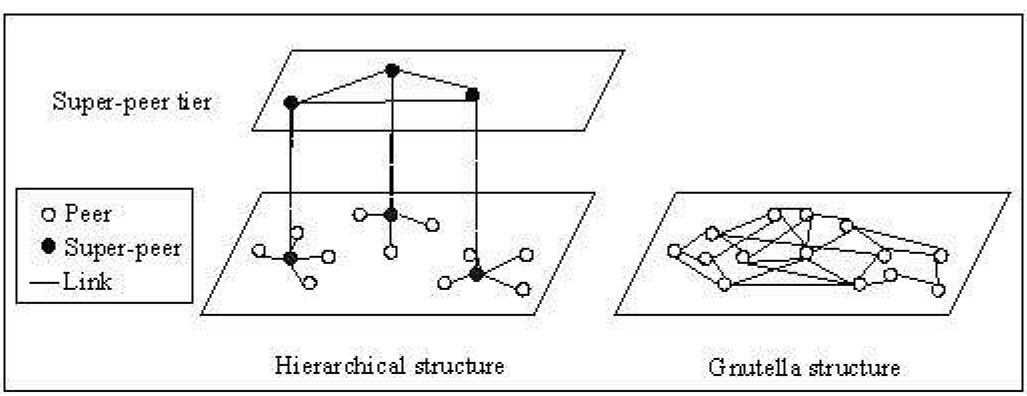

Figure 1: Hierarchical structure vs. Gnutella structure[1]

[1] The figure is from [1]

For unstructured P2P system, because there is no clue about where the content is located, the queries must be flooded through the network. Therefore, a lot of duplicated queries are generated, especially in highly connected network. These duplicated queries will not increase the chances of finding the contents, but will add extra work on each node in the network. The Super-Peer P2P network can reduce the duplicated queries by its hierarchical architecture. Instead of being flooded to every nodes in the network, queries are flooded to each super-peer only.

In addition, the thesis also introduces a flooding algorithm, Efficient Flooding Algorithm (EFA) for the Super-Peer architecture, which even further reduces the duplicated queries. The basic idea of this algorithm is that a node will not send a query to a particular neighbor, if it knows that this neighbor should receive the same node from another node.

However, it's not an easy task to understand the structure of the Super-Peer P2P network architecture and the behavior of the EFA content search algorithm. This is simply because the structure is hierarchical, and the structure is a logical architecture, and there is no real world representation. Meanwhile, the flooding algorithm is even harder to understand because the flooding happens in the whole network in parallel, and the states of each node are changing dynamically.

2. SuperViz

SuperViz allows a network developer, or a researcher to view the structure of a Super-Peer network and comprehend the behavior of the flooding algorithms through interacting with the system. This system applied various information visualization techniques to make it as effective as possible.

2.1 Data

A program for generating several topologies such as grid and random topologies is provided by Juan Li. I use this program to generate super-peer network topologies.

2.2 Interactive GUI Design

SuperViz provides a GUI that is intended to be easier for users to interact with the system. A control panel sits on the right hand side of the GUI (see Figure 2). This control panel plays the most important role of the interaction. Users are allowed to specify the size of the network, the degree of the connectivity, the routing algorithms, and the speed of the animation. The control panel is closable, so users can save some space for the topology frames if they do so. By having a control panel, users can find all the main functionalities of the system in a single place.

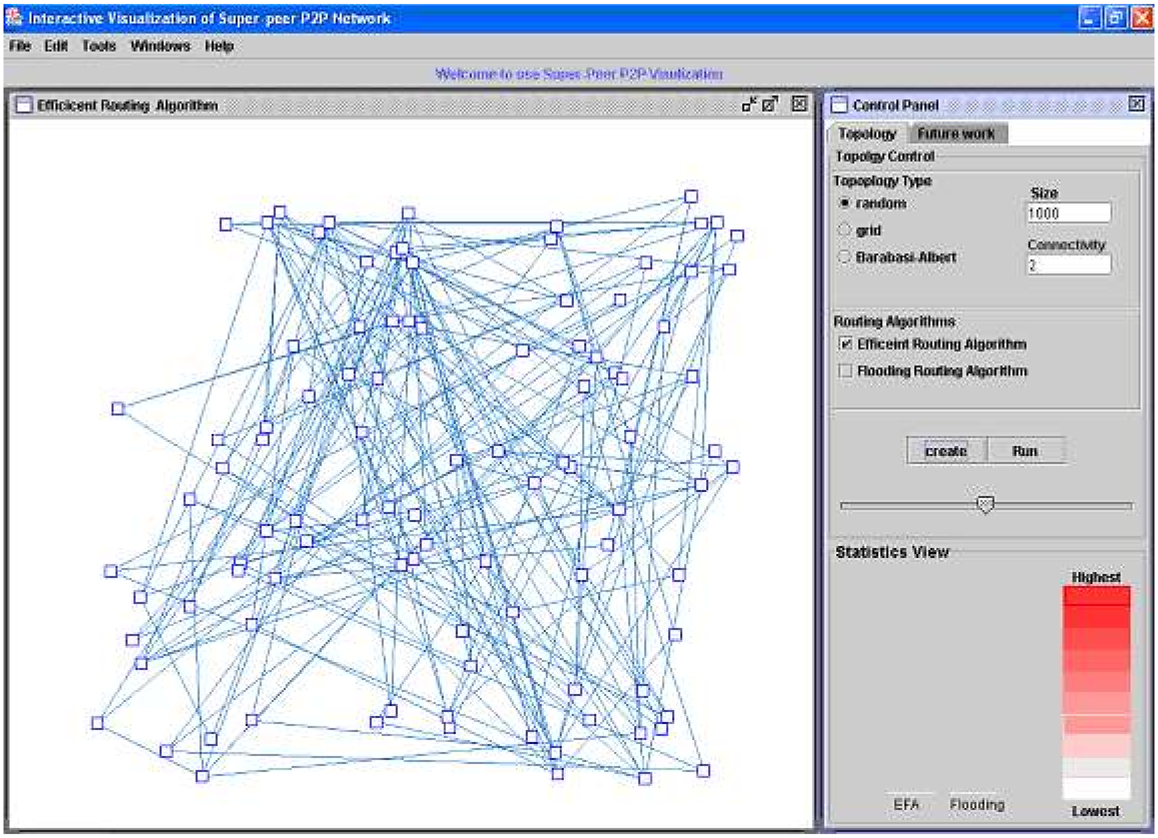

Figure 2: Interactive GUI Design

The topology graphs are shown in the middle of the GUI. The GUI can show multiple internal frames at the same time. The internal frames are movable and minimizable. These features are useful, because they can used to support side-by-side comparison, which is one of the information visualization techniques being used to compare the behavior of the different flooding algorithms. Users are also allowed to run the algorithms repeatedly to investigate the flooding behaviors.

2.3 Semantics Zooming and Navigation

Users can freely zoom and pan over the content to fulfill their interests to a particular area of the topology. The system supports the following zooming and navigation features:

- Automatically centering. If users resize the topology frame, the content will be automatically centered in the middle of the frame.
- Panning over the contents
- Zooming in and out the contents freely.
- Semantics zooming. The different amount of the details will be shown, when users zoom into different levels.

Users may want to resize the topology frame for many reasons. For example, when they close or open the control panel, the topology frame size would be enlarged or reduced. The automatically centering functionality can make sure that the topology graph is always shown in the middle of the window, so the users don't need to re-position the topology graph.

Users can also choose to zoom in and out, or pan over the content manually to fit their special interests in the particular area of the topology graph.

One of the important features this system presented is the semantics zooming. This is effective for showing the hierarchical structure of the Super-Peer network system. I first tried to use the overview and details to fulfill this task, but later on, I found the semantics zooming seems more effective to this kind of tasks.

The idea is that, users would first see a network topology in the higher level, in which, each square representing a cluster of peers (see Figure 3). This level gives a user an overview of the Super-Peer P2P network topology. When users zoom into the graph, more contents of the graph are shown. For example, after users zoom into a certain level, they see the peers appear in each square (see Figure 4). If they zoom further, they see the strings that display the contents held by each peer. The semantics zooming not only helps users to understand the hierarchical structure of the super-peer P2P network topology, but also makes the limited number of pixels of the window to show as many contents as possible in different levels and contexts without overwhelming the users.

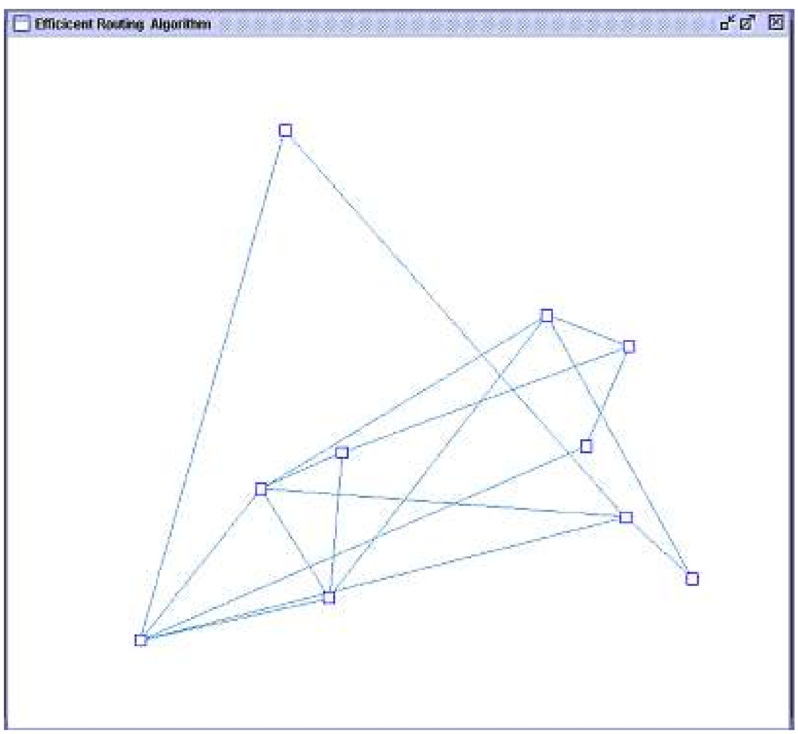

Figure 3. Topology graph before zooming

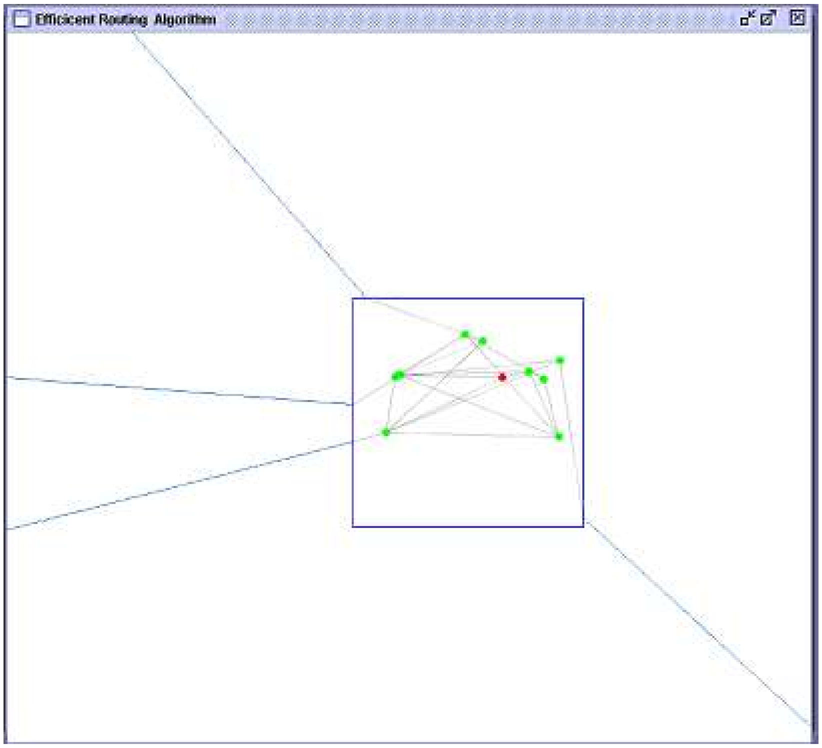

Figure 4: After zooming into a certain level, the actual peers in each cluster are shown, and the super-peer is in red color

2.4 Color Coding

Another important task of the system is to make sense of the flooding algorithms. Color is effectively useful here for indicating the number of duplicated queries received by each super-peer. In the initial topology graph, the square boxes are in white to indicate that they haven't received any queries yet. When the algorithm is running, the color of each square box is continuously changing from white to red gradually (see Figure 5). The level of redness depends on the number of the duplicated messages received. The more the square boxes are getting red, the more duplicated messages they received. This technique is especially effective, when two algorithms are being compared side-by-side. One can easily justify which algorithm generates the most duplicated messages by looking at the overall redness of the nodes in each topology graphs.

The links among the square nodes are also colored. In the initial stage, the links are in blue. When an algorithm is running, the link, that has a query passed through, turns into red. Therefore, after the flooding is finished, the links that have never been used for transmitting queries are still in blue, and the others are in red. People can easily distinguish these two type of link by the color. By comparing the two algorithms, we found that normal flooding algorithm simply used all the links, while EFA only used partial links of the network.

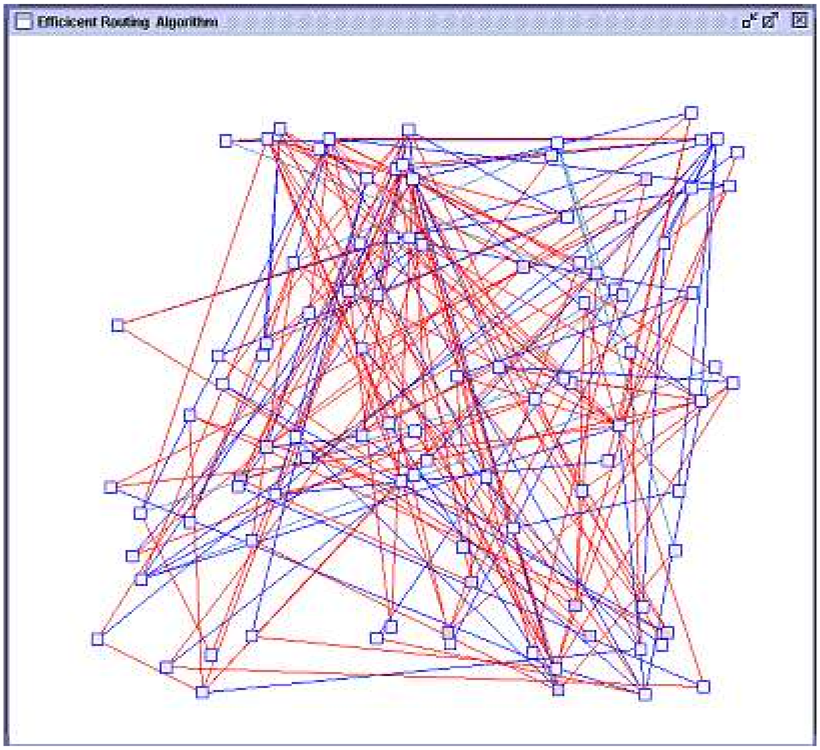

Figure 5: The color of each node is changing from white to red
to indicate that the they are receiving duplicated queries

A color scheme is provided on the control panel (see Figure 1) to indicate the mapping between the level of redness and the amount of duplicated messages.

2.5 Animation

To understand the flooding algorithm, a technique that can demonstrate the queries passing around in the network in a dynamic fashion will be helpful. Animation is effectively used in doing this job. Once users click the run button to start the animation, a node sends a query to the network. Red round circles representing the query messages start to move through the links between the nodes and their neighbors (see Figure 6). Users can view how the queries are flooded over the network by viewing the red circles moving around. Therefore, that is the animation makes it possible for users to view the behavior of the flooding, which is impossible otherwise.

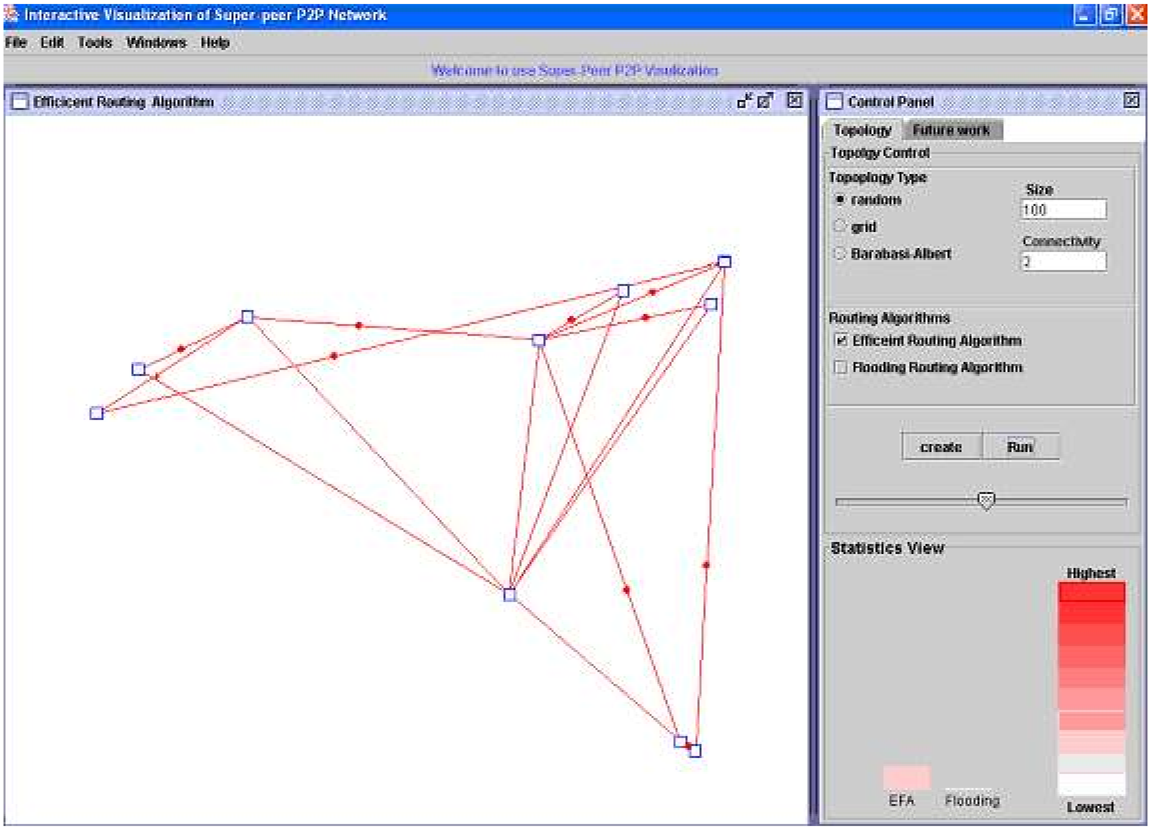

Figure 6: Red Circles represent query messages. They are flooding in the network. Links that have queries going through become red

Users also have the choice to change the speed of animation. The sliding bar in the control panel allows users to slow down or speed up the animation. This is helpful when users want to view the flooding behavior for different purposes. For example, one can choose to view the flooding behavior of the overall network (using faster speed), or view the flooding behavior of a particular node (using low speed).

2.6 Side-by-side Comparison

To compare the flooding behavior of two algorithms, I applied the side-by-side comparison technique. When users select both of the check boxes of the two algorithms, EFA and flooding, at the same time, two topology frames are displayed and positioned on the top and the bottom (see Figure 7). After the animation starts, the two algorithms are running in each frame in parallel. By doing that, users can easily understand the difference between the EFA and flooding. For example, flooding tends to take longer time to stop than EFA, this is obvious especially when one runs the two algorithms side-by-side in parallel. At same time, users may observe that the simple flooding algorithm is more overwhelmed by the duplicated query messages because the graph is very red, while the EFA algorithm is more efficient, because it's more blue, which indicates that fewer duplicated query messages emerges in this network.

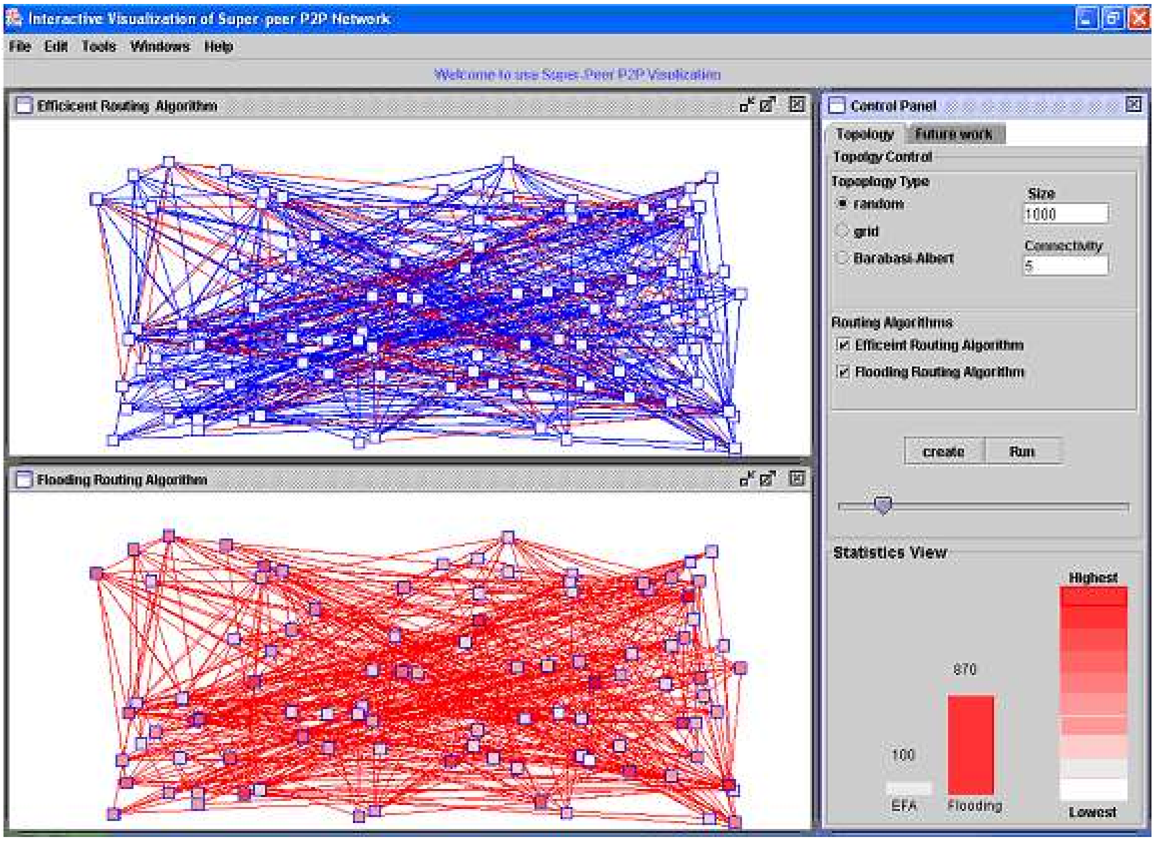

Figure 7: Side-by-side comparison

The bar charts are used to help users quickly get the difference ratio of messages emerges in each algorithm. The color also enhances the idea that the more red, the more duplicated messages.

2.7 Highlighting Directly Connected Links for Avoiding Occlusion

Users may want to see the directly connected links of a specific node. However, the links of each node are not obvious when the nodes are cluttered, especially when the network size is large and the connectivity degree is high. In this case, highlighting the node and its directly connected links helps to distinguish the links of this node from the links for other nodes. When users move the mouse over a specific node, this node is highlighted and its directly connected links as well (see Figure 8).

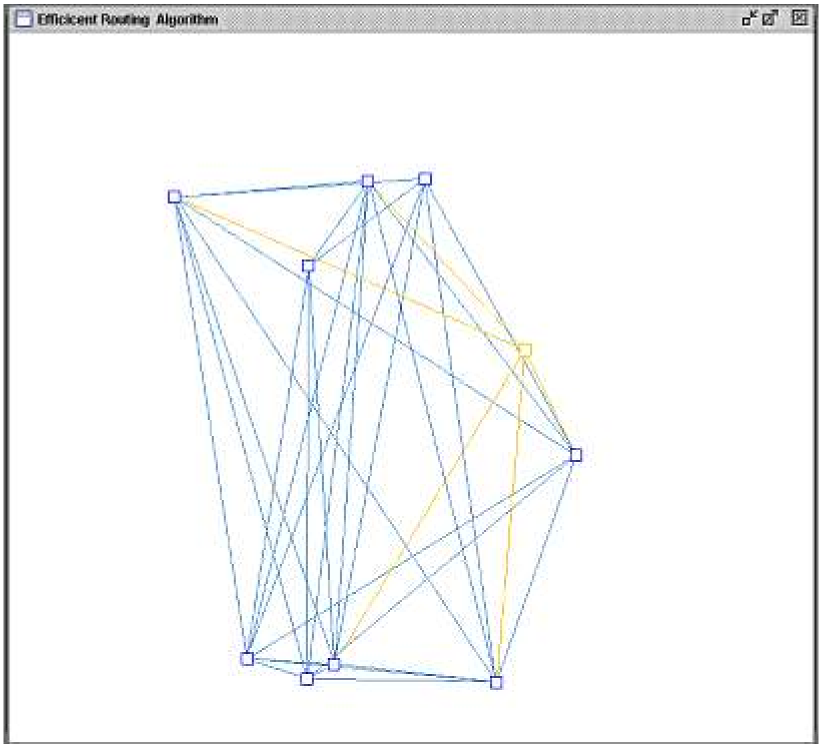

Figure 8: Highlighting directly connected links

3. Implementation

SuperViz is implemented using Java 2D and the zoomable user interface toolkit, Jazz 1.3. I used the Java IDEs such as Netbeans and Eclipse for developing the system. I used Netbeans, because its form editor is very helpful in GUI developing. I used Jazz to create the zoomable topology graph. It has been a learning curve because I have little graphics background.

I used Java Thread for implementing the animation. When a query reaches a node, the node starts to run as a thread. After the thread updates the node's states and passes the query to another node via a link, the thread is killed. Each link is also a thread, the link thread's job is to animate the message passing, and starts the node thread at the other end.

I designed and programmed SuperViz in a way that makes the system extensible. Other routing algorithms and network topologies can be added with a little code change. One may use it as a framework for visualizing other network topologies.

3. Scenario of use

Now, I present 2 scenarios where SuperViz can be used effectively.

3.1 Understanding the Super-Peer P2P Network Structure

A professor in the Department of Computer Science wants to introduce the Super-Peer P2P network to his students. The professor uses SuperViz as a tool to help his students to understand the concept of Super-Peer P2P network architecture. He asks the students to download the system, and play with it at home before coming to the school the next day, when the professor will give a lecture on the Super-Peer P2P network architecture. The professor finds that students have understood the architecture very well after using the tool.

In this scenario, users first generate a topology by specifying the number of super peers in the network, the degree of connectivity among nodes, and the topology types such as random topology and grid topology. After that, users can create a topology graph based on the parameters just entered. Users first see a normal network topology graph. Each node represents a cluster of peers. Users then zoom into a particular node. While the node is blown up to a certain level, the peers in this cluster are shown, and after being further zoomed in, the contents that each peers contains are also shown. After these steps, users understand that the hierarchical structure of the Super-Peer P2P network.

3.2 Comparing EFA and Flooding Routing Algorithm

The professor then wants to prove that the Efficient Flooding Algorithm (EFA) is more efficient than simple flooding routing algorithm. He then asks the students to use the animation functionality of the SuperViz.

A student first selects both of the routing algorithms, and then follows the steps described in the first scenario. Two identical topologies are created and displayed on the top and bottom. The student then starts the animation. The student sees that the query messages are flooding in the both networks. However, he observes that the nodes in the topology graph that is running simple flooding become more red than the nodes in the network that is running the EFA algorithm. Meanwhile, the links in the simple flooding algorithm becomes all red, while partial links are red in the EFA . Finally, two bar charts are created showing that the height of the bar corresponding to simple flooding is much higher than the one corresponding to EFA, which means more duplicated messages emerges in the simple flooding network.

4. Related Work

Visualizing Network Agents[1] by Ken Deeter presents a visualization tool for mobile agents, in which agents are moving in the network. The animation of the query message passing is inspired by this paper. Multiscale Visualization of Small World Networks[2] introduces the techniques for visualizing clustered hierarchical networks, which is similar to the situation of the super-peer P2P network architecture. I applied the idea of semantics zooming in this paper to SuperViz. Automatically centering and zooming are effectively used in the Visualization of Web History [3] by Rajiv Gandhi et al. When the panel that displaying the web history tree is resized, the web history tree is automatically centered and zoomed to fit in the middle of the panel. I used this idea in SuperViz for automatically positioning topology graph.

5. Lesson Learned

The whole process, from finding the project idea to making a proposal of the project, and finally implementing this project, is certainly a good practice for a research activity. I have made an intensive reading for searching a suitable topic. Although this topic is finally chosen, but I developed my ability of searching and finding a potential topic. This ability is definitely valuable to my future research.

Finding the most suitable information visualization techniques to apply on the project is another challenging issue. Ideas may be changed over and over. The ideas you have at the beginning may not be the ones you use in the end. Best ideas often come up during the implementation.

This is my first time to be exposed to Java 2D and Jazz. I have little graphics background, so understanding some concepts is a learning curve for me. Therefore, I learned a lot from implementing the project. I found that Java 2D is such a fun.

6. Evaluation

6.1 Strength

SuperViz is an effective visualization tool for people to understand hierarchical network architecture, in particular, Super-Peer P2P network, which has no real world representation. The tool is also effective for demonstrating the behavior of flooding algorithm in work, which is also impossible for people to see in the real world. By SuperViz, users are not only able to visualize the behavior of the flooding algorithms, but also able to visualize the amount of duplicated queries in each node through color coding.

The system is also effective for its interactive features. A control panel is handy and intuitive for the first time users without pre-training. Users are allowed to interact with the tool by specifying different parameters, so they feel more entertaining and being in control of the whole situation.

6.2 Weakness

The project can be improved by adding more features. I applied automatically centering, but automatically zooming will also be useful, because it can take the best use of the pixels when the frame size is enlarged.

In the current implementation, interaction is limited to the highest level. That is, the peers in each cluster are not interactive and animating. It would be better, if the system

could show how a query message reaches the real super peer in each cluster, and how the peers interact with super-peers for sharing data contents.
Message query is always initiated by the system selected node, it could be better to let the users to select a node to initiate the query, therefore, users could observe more interesting behaviors of the flooding algorithms when the flooding starts from different positions of the network.

7. Conclusion

This paper presents the SuperViz project, which is an interactive visualization tool for Super-Peer P2P Network. SuperViz is an effective visualizing tool for visualizing Super-Peer P2P network architecture, and for comparing the simple flooding algorithm with the Efficient Flooding Algorithm (EFA). By applying various information visualization techniques such as semantics zooming, color coding, animation, and side-by-side comparison, SuperViz turns out to be an effective visualization tool for the specific tasks.

Reference:

[1] Juan Li. ECSP: An Efficient clustered super-peer architecture for P2Pnetworks. Master thesis at UBC, 2003

[2] Visualizing Network Agents. Course project of cs533 at UBC, 2003

[3] Multiscale Visualization of Small World NetworksAUBER D. ; CHIRICOTA Y. ; JOURDAN F. ; MELANÇON G. InfoVis Symposium on Information Visualization , 2003, pp. 75-81

[4] Domain Name Based Visualization of Web Histories in a Zoomable User Interface. R.Gandhi, G.Kumar, B. Benderson, B. Shneiderman. In Proceedings of the Second International Workshop on Web-based Information Visualization (WebVis'00), pages 591-598, Sep. 2000.

[5] Prawn: An interactive Tool for Software Visualization. Andrew Chan, Reid Holmes. Course project of cs533 at UBC, 2003