Visualization and Analysis of Email Networks

Xiaoyan Fu*
National ICT Australia

Seok-Hee Hong†
National ICT Australia
School of IT, University of Sydney, Australia

Nikola S. Nikolov‡
Department of CSIS, University of Limerick, Ireland

Xiaobin Shen§
Department of Civil and Environmental Engineering, University of Melbourne, Australia

Yingxin Wu¶
National ICT Australia
School of IT, University of Sydney, Australia

Kai Xu‖
National ICT Australia

Abstract

This paper presents various methods for visualization and analysis of email networks; visualization on the surface of a sphere to reveal communication patterns between different groups, a hierarchical drawing displaying the centrality analysis of nodes to emphasize important nodes, a 2.5D visualization for temporal email networks to analyze the evolution of email relationships changing over time, and an ambient display for finding social circles derived from the email network. Each method was evaluated with various data sets from a research organization. We also extended our method for visual analysis of an email virus network.

Keywords: Visualization, Email network, SOM, Centrality, Ambient Display, Email virus network.

Index Terms: H.5.2 [INFORMATION INTERFACES AND PRESENTATION]: User Interfaces—Theory and methods

1 Introduction

Recently, email networks have been popular for both analysis and visualization. For example, analysis of email networks was used to identify the informal communication structure within an organization [14, 30], to discover the shared interests between people [28] and in relation to the spread of computer viruses [26]. Further, visualization of email networks has been widely applied to assist the users to understand email data and analyze the social network it reflects[3, 34, 2, 32].Based on these results, several visualization methods, such as “Thread Arcs”, have been used to help users track email threads [17], where a variety of information regarding email threads is visualized using a curved tree structure.

An email network visualization tool, the “Email Mining Toolkit”, is used to identify possible spam and viruses [21]. In [23], an email network was used to study information seeking and workplace collaboration, followed by many visualization tools such as the “Collaborative Innovation Networks” [13], “Social Network Fragments” [6] and “Rhythms in Email Experience” [22]. Another interesting development of email visualization is an application of ambient display, i.e., visualization exploiting peripheral vision. An example is the “Info-Lotus” [35] for email notification visualization.

In this paper, we consider two specific types of email networks: *small-world* email networks to analyze social networks and *email virus* networks to analyze an email virus attack. In general, visualizing small world networks is very challenging due to the short diameter of the network. For techniques and methods for drawing small-world networks, see [31, 9].

This paper presents various methods for visualization and analysis of email networks; visualization on the surface of a sphere to reveal the relationships between different groups, a hierarchical drawing displaying the centrality analysis of nodes to emphasize important nodes, a 2.5D visualization for temporal email networks to analyze the evolution of email relationships over time, and an ambient display for finding social circles that may reflect collaboration. Each method was evaluated with various data sets from a research organization. These were exhibited at public demonstrations in order to obtain informal feedback. We also extended our method for visual analysis of an email virus network.

This paper is organized as follows: In the next section, we present simple statistics of the email network. We then present four different methods for visual analysis of email networks: sphere drawing to reveal communication patterns between groups, hierarchical drawing to display the centrality analysis of nodes inside a group, temporal email networks to analyze the evolution of email relationships changing over time, and ambient display for identifying social circles. We next present a method for visual analysis of an email virus network. Finally, we conclude with an open problem.

2 Statistical Analysis of Email Networks

The data was collected from the email server of National ICT Australia (NICTA) from July to August 2004. Specifically, an email network was derived from an email log file from the email server. In the email network, each node represents an email address and each edge between two nodes represents an email exchange between these two email addresses.

The original email network has 604 nodes and 8605 edges in total. The network has some disconnected nodes. The *giant component*, that is the largest connected component, has 470 nodes. The diameter of the network is 5, and the average path length is merely 2.2, which means that the email network is an “ultra-small-world” network with a small diameter and short graph distance between any pair of nodes. The clustering coefficient is 0.489, which means that the network is relatively highly clustered [7].

3 Spherical Drawing of the E-mail Network

In this section, we describe a new method to visualize an email network on the surface of a sphere using a Self-Organizing Map (SOM). This section is organized as follows: we briefly introduce the SOM and its application to graph drawing, followed by the advantages of graph layout on a sphere, and then discuss the detail of our method.

Here, we used a slightly modified data set from the previous section. We omitted the emails that had an external origin or destination to analyze relationships between groups inside the organiza-

*e-mail: xiaoyan.fu@nicta.com.au
†e-mail:shhong@it.usyd.edu.au
‡e-mail:nikola.nikolov@ul.ie
§e-mail:xrshen@unimelb.edu.au
¶e-mail:chwu@it.usyd.edu.au
‖e-mail:kai.xu@nicta.com.au

tion. An edge is created between two users if they had exchanged (sent and received) emails at least five times. This results in a smaller network with 277 nodes and 1975 edges. Figure 1 shows the email network using the force-directed layout in Pajek [27].

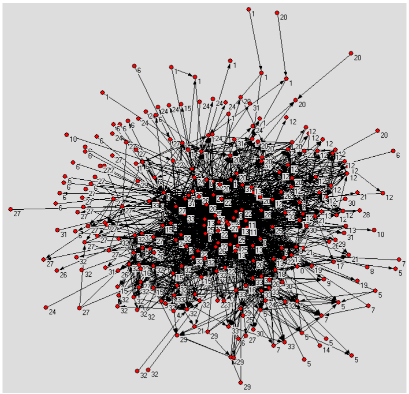


Figure 1: The NICTA email network.

The self-organizing map [20] is an unsupervised competitive artificial neural network. It projects high-dimensional data onto a low-dimensional space. The projection preserves the topological relationships of the original data: data close to each other in high dimensional space are projected to similar places in a low dimensional space. The neuron's neighborhood relationship is fixed and defined by a 2D rectangular or hexagonal lattice so that every grid unit has 4 or 6 neighbors. During the training phase, all neurons compete with each other for the input signals. The winner and its neighbors within a specific distance (update radius) adjust their weight vectors towards the input signal.

$$n.weight := n.weight - \alpha * h(d) * (n.weight - s) \qquad (1)$$

Here, α is the learning rate which decreases with the training time; s is the input signal; d is the distance between the neuron's weight vector and the input signal; $h(d)$ is the neighborhood function.

Previously, the SOM has been applied to graph drawing [5, 4, 24]. They considered the whole graph as a neural network: each node is a neuron and the edges define the neighborhood relationships. It is claimed that the algorithm is able to lay out positive or negative weighed graphs, directed graphs and large graphs [4]. The computational complexity is quadratic.

It is mentioned that the algorithm can be easily extended to layout graphs on spherical surfaces[24]. The spherical surface may provide a natural fisheye effect which enlarges the focus point and shows other portions of the image with less detail. This effect can be useful for small-world network visualization. As pointed out in [24], a spherical 3D layout that allows interactive rotation can be a novel interaction technique for graph navigation. Based on this idea, an interactive spherical projection display, the ViBall, was developed in our lab [18]. Using the ViBall, the spherical image can be rotated not only by mouse or keyboard, but also physically by hand. We made use of this device to visualize a small world email network.

We made several changes to the algorithms in [5, 4, 24], as they need to be adjusted for small-world social networks. First, we needed to determine the update radius in the training of the SOM.

1. In [24], the initial update radius is 3, which means the neighbors within 3 steps from the winning neuron (node) will be updated. However, the email network has a small-world property: the average distance between any pair of nodes is 2.2. Thus we chose an update radius smaller than 2.
2. Email networks do not have the transitivity characteristic: if A communicates with B, and B communicates with C, it doesn't mean that A communicates with C.

Based on this, we chose an update radius of 1.

Secondly, we chose a logarithmic neighborhood function instead of an exponential function:

$$h(d) = \frac{\log_e\left(\frac{0.1}{d}\right)}{\log_e(w * 0.9)} \qquad (2)$$

Here, w is a weight of an edge, which indicates the number of emails exchanged between two people. In our implementation, it is normalized to the range of [0,1]. This controls the amount of adjustment in position. The bigger the w, the bigger the value of $h(d)$. d is the geodesic distance between the winning node and its direct neighbor on the sphere. 0.1 is the desired distance between the nodes. This function will be negative/positive if the distance is less/bigger than 0.1, and the neighbor will be pushed away/dragged closer from the winning neuron (node).

Algorithm 1: SOM Sphere Layout

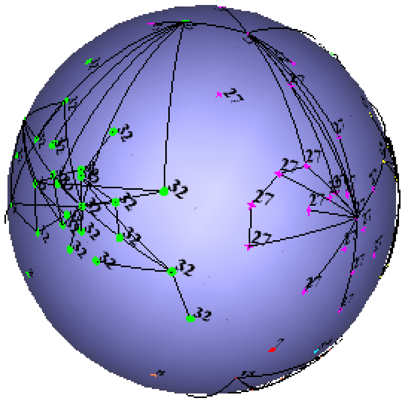
input : Graph G=(V,E);
Epoch: t_{max};
Initial learning rate: α;
output: Spherical Layout of Graph G

Initialization: Place nodes in random locations on the sphere;
while $t < t_{max}$ **do**
  Generate a random vector v on the sphere;
  Find the closest node n;
  Update n's position: $n.pos := n.pos - \alpha * (n.pos - v)$;
  foreach *n's direct neighbor m* **do**
    $\beta := \alpha * \frac{\log_e\left(\frac{0.1}{d}\right)}{\log_e(w*0.9)}$;
    $m.pos := m.pos + \beta * (n.pos - m.pos)$
  endfch
  $t = t + 1$;
  $\alpha = \alpha * \frac{t_{max} - t}{t_{max}}$;
endw

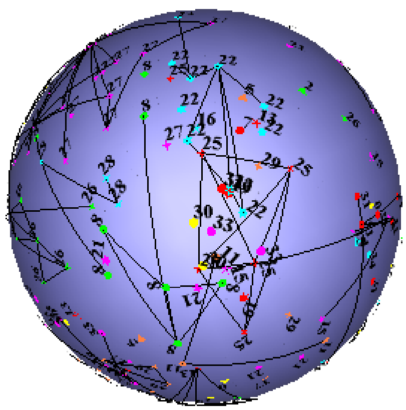
In our implementation, the initial learning rate α is 0.9 and t_{max} is 500. There are 34 different groups (research groups, administration and management groups) in NICTA. We use different node shapes to display people in different groups. Each person is also labeled with a number which represents the group he/she belongs to.

In Figure 2, only edges representing intra-group communications are shown to reduce visual complexity. Inter-group communications can be observed by the closeness of the groups: the closer the groups, the more communication between them. Some communication patterns can be seen. People in research groups such as 27 and 32 tend to exhibit the same communication pattern. Their intra-group communication edges almost form cliques. Each research group is well separated; they do not communicate or collaborate each other (see Figure 2(a)). However, people in administration or management groups such as 8 (The CEO office), 22 (Finance) and 25 (Human Resource) are mixed together. This means that they often communicate and collaborate with each other in order to complete a task (see Figure 2(b)).

Compared to the force-directed layout in Figure 1, the SOM layout shows communication patterns between groups more clearly. The nodes are distributed more evenly on the surface of the sphere, instead of collapsing at the center. However, as pointed out by [24], the main disadvantage of using the SOM for graph layout is the overlapping between nodes and edges.



(a) Two research groups.



(b) The management groups.

Figure 2: Spherical drawing of the NICTA email network.

4 Displaying Centrality Analysis of an E-mail Network Using Hierarchy

Centrality in social network analysis is a measure of the importance of a node embedded in the network. Hierarchical layout is popularly used to visualize centrality analysis of a network. This involves higher placement of a node with a high centrality value, than a node with lower centrality value, so that the centrality value can be interpreted with the height of a node position.

The considered e-mail network of a specific research group is small, but very dense with 32 nodes and 328 edges. The number of emails between two nodes are represented using a *weight* of an edge between the nodes. As there are edges with weights ranging from 1 to 2229, it is meaningful to consider subsets of edges when analyzing the network. If, for example, we consider only the edges with weight of at least 100, we are left with one big component with a few isolated nodes. The giant component, shown in Figure 3, has 22 nodes and 72 edges. We now visualize this giant component using a hierarchical layout in order to display centrality analysis of each node.

Recently, 2.5D hierarchical layout has been introduced [16], as an extension to the classical 2D hierarchical layout (also well-known as the Sugiyama method) for drawing directed graphs [29]. In the 2.5D hierarchical layout, each layer was further divided into k parallel walls, as an efficient way of using the third dimension for reducing the visual complexity and minimizing occlusion. Roughly speaking, there are four steps similar to the 2D Sugiyama method for producing a 2.5D hierarchical layout:

1. Partition the node set into layers;

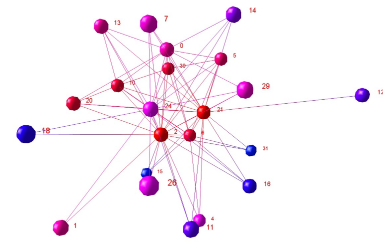


Figure 3: The giant component of the e-mail network, with edges representing at least 100 e-mail messages.

2. Split each layer into k walls, $k \geq 2$;

3. Order the nodes in each layer and wall;

4. Assign x-, y-, and z-coordinates to all nodes.

In general, in the 2.5D layout, the hierarchy is further split into k parallel planes (or walls), each containing a 2D hierarchy. Step 2 can be achieved according to various criteria. In the examples below we employ a balanced min-cut algorithm that minimizes the number of edges between two walls with balanced partitioning of vertices [16]. In the case of more than two walls, we use the barycenter split, i.e. the wall node v is assigned to the barycenter of the walls of its neighbors on the layer below [15].

As the network was modeled as an undirected graph, we made the following modifications to the 2.5D hierarchical layout [16, 15] by using centrality values in order to define hierarchy and edge directions.

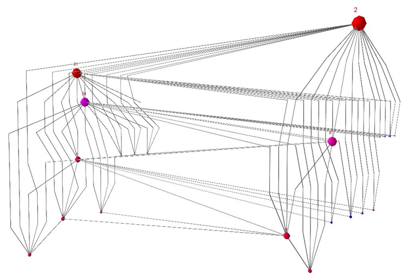
At step 1, the node set is partitioned into an ordered collection of layers $\mathscr{L} = \{L_1, L_2, \ldots, L_h\}$, so that if $u \in L_i$ and $v \in L_j$ for edge (u,v), then $i < j$. That is, when layers are drawn on parallel lines, all edges point into the same direction, e.g. downwards. Thus, the direction of the edges plays a significant role for partitioning the node set into layers.

We now explain how the direction of the edges can be used to emphasize properties of the network. Consider the undirected edge $\{u,v\}$, and let d_u and d_v be the degree centrality values of nodes u and v respectively. We can appoint u as a source of the edge if $d_u > d_v$, and v as a target. If $d_u > d_v$, then v is the source and u is the target. In a hierarchical layout, the layer a node belongs to and the degree centrality of the node will be loosely connected. Each node will be placed above all its neighbors with lower centrality values and below all its neighbors with higher centrality value. The resulting drawing will contain hierarchy in the strongest sense, i.e. without edges between nodes in the same layer, and still a loose relation between the centrality values and the vertical position of the nodes.

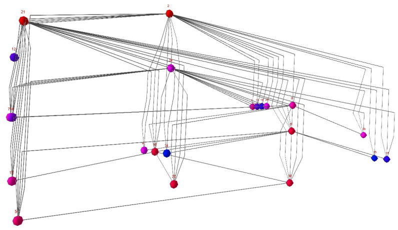
Figure 4(a) shows a 2.5D layout with two parallel walls. In the drawing, the direction of the edges is assigned according to the degree centrality values of the nodes. The size of the nodes also represents their degree centrality values. The relationship between the layers and the centrality values makes it easier to understand the underlying prominence (or influence) structure of the network. A similar drawing, but with 4 parallel walls, is shown in Figure 4(b).

Once a hierarchy with edge directions related to the degree centrality values is obtained, we can further map another centrality value to the size of nodes. For example, in Figure 5, the eigenvector centrality values are mapped to the node size, simultaneously displaying the result of two centrality analyses in a single drawing.

The drawings demonstrate how 2.5D hierarchical drawings, in combination with visual properties of the nodes, can be used for efficient visualization of several centrality values in a single drawing.



(a) Two parallel walls.



(b) Four parallel walls.

Figure 4: The giant component of an email network with 2.5D hierarchical layout

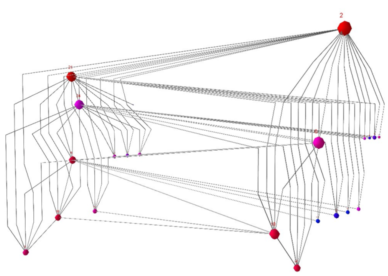


Figure 5: Combined visual representation of two centrality values: edge directions related to degree centrality values; node size related to eigenvector centrality values.

The hierarchical layout makes the graph easier to navigate and facilitates the understanding of the structure of the network from the perspective of the centrality measure mapped to the edge directions.

5 Temporal Email Network Visualization

Recently, temporal networks played an important role in social network analysis due to network dynamics. Good visualization methods for time-series networks can provide better understanding on network evolution[8], thus becoming an important supplement to current social network analysis methods. For example, temporal email networks have been studied for analysis and visualization [12, 3, 34].

The email data set we use records email traffic between July 2004 to March 2005. Therefore, eight data files were generated, with each containing the email communications for one month. To simplify, the direction of the communication is not considered.

Previously, temporal networks have been visualized in two ways:

- a smooth animation between a series of visualizations of networks at consecutive time points [25, 12];

- a 2.5D visualization method, which draws each network in 2D and then stacks them up into 3D using parallel planes [10, 8].

Preserving a mental map is one of the most important criteria for evaluating methods for visualizing temporal networks. Animations seem to be a good choice for an overview; however, the user may fail to remember the details. For small-size temporal networks, a 2.5D visualization method can show the entire history of network evolution without introducing overwhelming visual complexity. As the size of email network of each group is relatively small, we chose a variation of the 2.5D visualization method.

In our 2.5D visualization method, nodes that represent people in the data set are placed into plates; nodes in the same plate are connected by edges representing email communication; plates of consecutive times are stacked in order. A force-directed layout is applied for each plate to draw each network at that time frame. Degree centrality and betweenness centrality [33] measures are also applied in order to provide a further analysis. Finally, as an improvement to existing 2.5D methods [10, 8], edges are added between the same nodes in different time plates, so that the evolution can be easily highlighted.

As new inter-plate edges are introduced in our framework, we can define a new criteria for a good 2.5D temporal layout to minimise the total inter-plate edge lengths.

Note that the force-directed method implies some randomness. That is, if we naively apply a force-directed method for drawing each plate and connect inter-plate edges, this may result in the type of drawing shown in Figure 6. Here, inter-plane edges are drawn as long edges, resulting in occlusion, and hiding the real evolution of the temporal network.

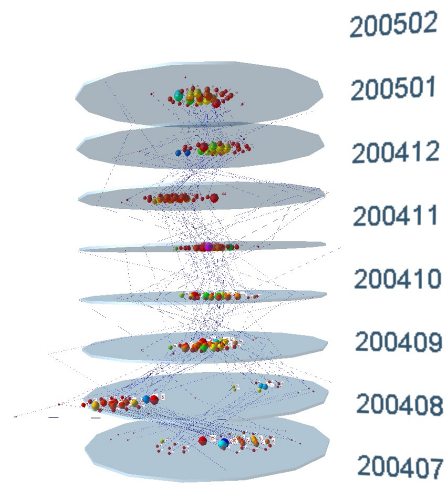
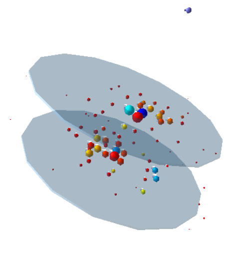


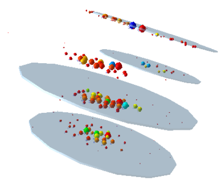
Figure 6: Long inter-plate edges.

We devised two methods to address this problem. The first method is to define a *supergraph* that consists of each plate, plus inter-plate edges. We then apply the force-directed algorithm for the supergraph. Inter-plate edges are considered as part of the supergraph, and are assigned corresponding edge weights. When the force-directed algorithm reaches the equilibrium, the inter-plate edges tend to be drawn as straight lines with less occlusion. However, due to the size of the supergraph, it tends to take longer time. Figure 7 shows the process of the method.

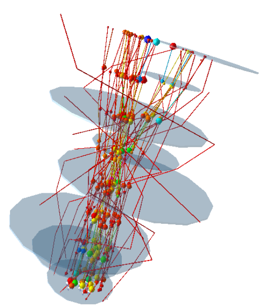
Another solution is to draw each plate separately, initializing the location of the same node in the next plate with the location in the previous plate. More specifically, we assign random positions, only to the first plate. When the layout of the first plate is completed, the



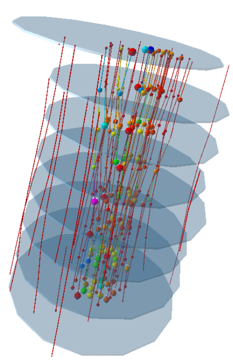
(a) Draw the plates.



(b) Apply a forced directed layout to each plate separately.

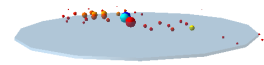


(c) Add inter-plate edges.

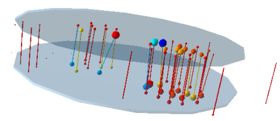


(d) Apply a forced directed layout to the supergraph.

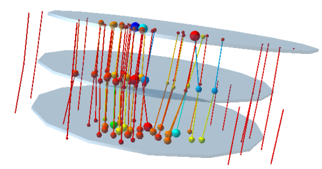
Figure 7: Using a supergraph with added forces between plates.



(a) The first plate.



(b) The second plate.



(c) The third plate.

Figure 8: Draw one plate after another using good initialization.

positions of the nodes are saved. From the second plate, the positions of nodes are initialized with the positions of the corresponding nodes in the previous plate. This method can also minimize the difference between the layouts of two consecutive networks in a time series, which helps the user to preserve his/her mental map. It also speeds up the computation of the drawing in the next plate, reaching the equilibrium faster, as most nodes have similar relations in each plate. Figure 8 shows this process.

Compared to the visualization in Figure 6, both methods produce layouts that make it easier to understand the network evolution: nodes with no change are connected with almost parallel inter-plate edges; a node with change is highlighted by an inter-plate edge with two end points at considerably different locations.

Moreover, the framework is flexible and extendable. As the graph layout in each plate is relatively independent, it is easy to replace the layout algorithm in the plates with other avaliable 2D layout algorithms. The framework can also be used to visualize other types of networks, such as multiple relational networks, evolution networks, dynamic networks or for network comparisons with minor modification.

6 Visual Analysis of Email Virus and Propagation Network

A real data set always comes with unexpected events; in many cases, such events are treated as noise and filtered in the early data processing stage. However, sometimes they also contain useful information that can lead to interesting results [26]. In this section we present a method of visual analysis of email virus attack - an unexpected event.

The email virus attack recorded in the data set happened on November 10, 2004. The virus was coded: W32.Mydoom.AI@mm. It is a mass-mailing worm which spreads by sending an email to the email addresses that it finds in the address book. An infected computer will act as a fake email server and send virus emails to others [1].

In general, email network analysis uses a "one-mode" network approach; in other words, the email network represents only the interaction between email-users. Although, in fact, a lot more information was monitored by the server and recorded in log files, it is hard to represent it.

On the other hand, a *two-mode* network, which represents two types of nodes in the graph, can be a better representation. An email transaction has the following three stages:

- Client (sender) sends an email to mail server
- Email exchanges between servers
- Client (receiver) receives an email from mail server

We define a *two-mode* email transaction network which contains both user nodes and server nodes. More precisely, it contains both client (sender and receiver) side and server side of email transactions.

For example, a normal email transaction network within a one-hour period of our data set can be represented as in Figure 9. Here, red nodes represent servers while yellow nodes represent clients. To distinguish the sending and receiving processes, green and blue edges are used to display them, respectively. The red node in the center represents the main email server in the data set.

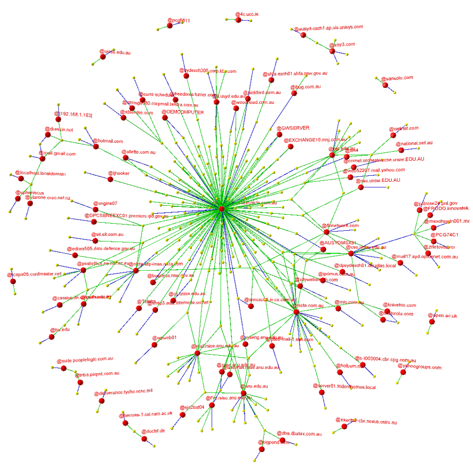


Figure 9: Two-mode Email Network.

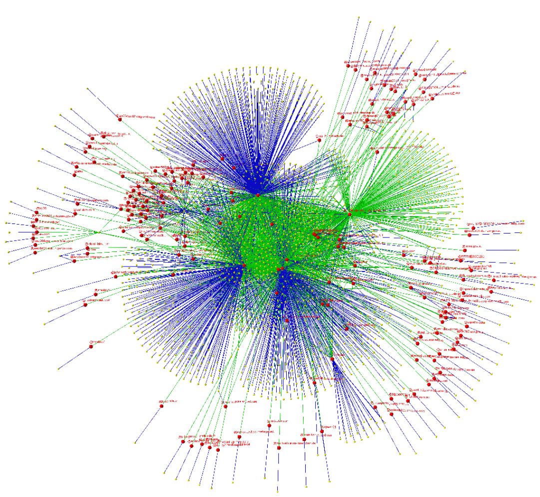


Figure 10: Virus Infection.

In Figure 10, we see a quite different picture. It is a visualization of an email network from 9am-10am, November 10, 2004, when the virus attacked the network. It is quite easy to see that something extraordinary is happening, as the email traffic increased tremendously. Although the sudden increase of email traffic can also be seen by checking the log file, it is more insightful to display the same information using the visualization. In Figure 10, obviously some red nodes were much more active than the normal pattern in Figure 9: a huge number of emails was sent by them.

To identify such nodes using visualization, we again use centrality analysis. As mentioned previously, centrality indices measure the importance of a node in the network. As we want to highlight those sending lots of emails, the degree centrality is appropriate for this. As we deal with a two-mode network here, we need to extend the measure to a two-mode network. To meet our requirement, we only need a simple variation: we compute the degree centrality of server nodes and client nodes separately. Figure 11 shows the result, with degree centrality mapped to the size of the node. Three servers were highlighted. Not surprisingly, they are not the normal servers (see Figure 12); they are virus-infected computers which acted as “fake” servers.

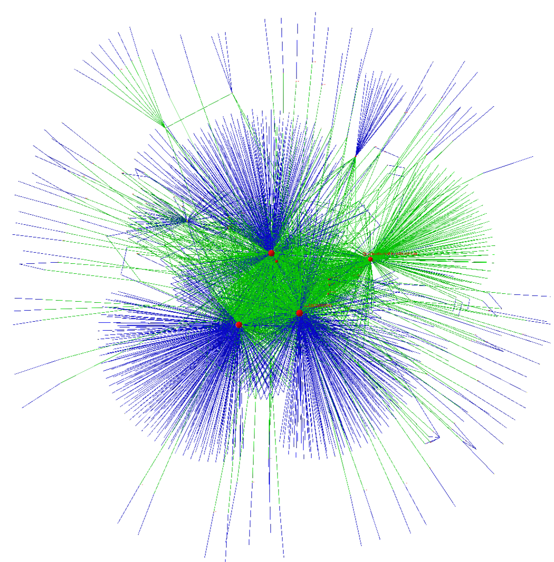


Figure 11: Highlight the infected server by applying degree centrality.

We can further visualize a temporal email propagation network. Figure 13 shows an example. In every one hour, a layout of a two-mode email network is drawn in a plate, showing the traffic of that time period; then those plates are stacked together, as a time-series network. Edges between plates are also added to highlight propagation of the email virus. This example clearly demonstrates the power of visualization combined with proper analysis methods.

7 Ambient Display of Email Networks

In this section, we use ambient display to represent email network collaboration inside a group. The aim of ambient display is to provide useful information which blends in aesthetically with the surroundings. E-mail communications, as a method of human collaboration, have become an integral part of our lives. We use real-time email logs as the data source, and represent collaboration relationships inferred from the data source in a synthesized painting of stars in the sky.

To meet the aesthetic requirement, we use a watercolor image as our final picture. In the drawing, the size of each star represents the amount of personal emails, and the distance between two stars represent collaboration between two people via email (See Figure 14).

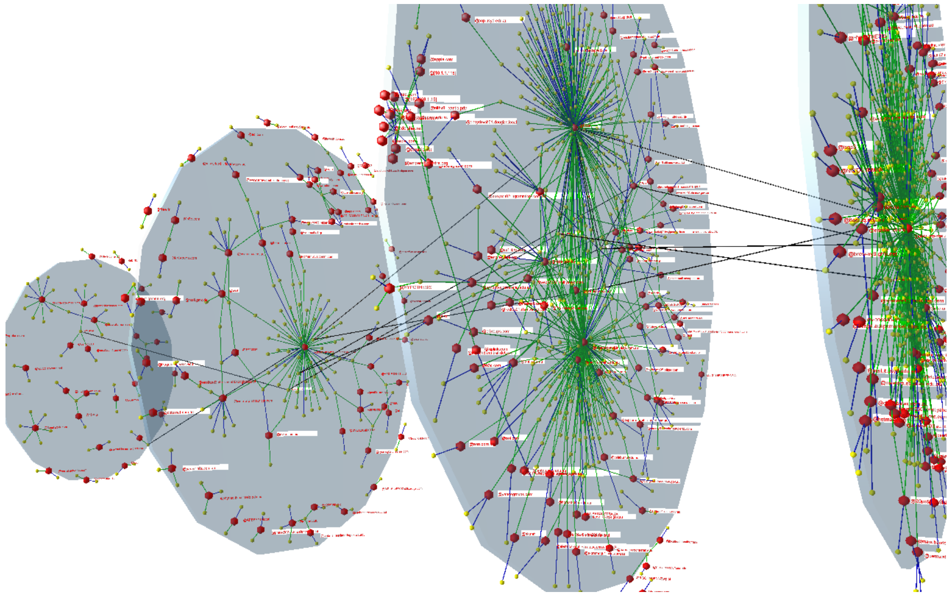


Figure 13: Email virus propagation.

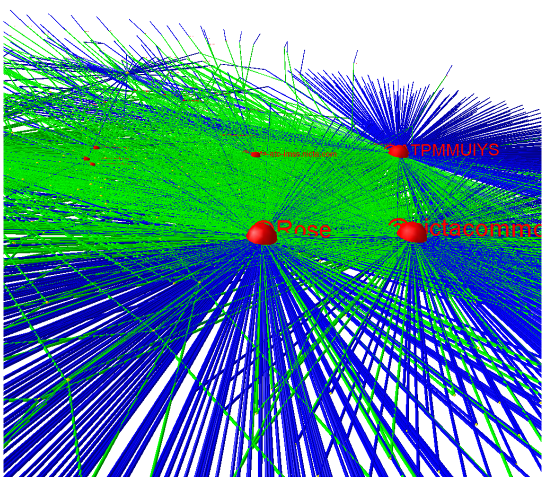


Figure 12: Infected computers acted as fake servers

Figure 14: Ambient display of an email network.

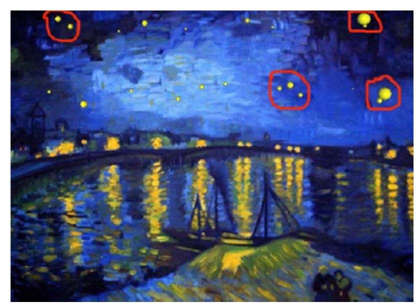


Figure 15: Social circles.

Figure 16: Ambient display in general environment.

Specifically, we model the email network as weighted graphs. For the layout, we used a modification of a spring algorithm [11], so that the distance between the stars may depend on the weight of the edges of the email network. That is, if two people exchange emails frequently, the stars corresponding to the people are drawn closely.

The ambient display represents real time visualization of an email network with 30 people in the same research group. We can easily locate social circles (see Figure 15 for red circle). This may be interpreted as potential collaboration between people inside the same research group working on the same research projects.

We created a traditional picture, using a picture frame around a monitor, for the ambient display (see Figure 16).

8 Conclusions

This paper presents various methods for visualization and analysis of small-world email networks and email virus networks. We now plan to conduct a formal evaluation of each method, which will include comparisons between the different methods. Also, visualization methods suggested by other researches [10, 8] will be considered. Our next research challenge it to design a method for visual analysis for large and complex temporal email networks, such as the ENRON email data set [19].

References

[1] Symantec security response. http://www.sarc.com/avcenter/venc/data/w32.mydoom.ai@mm.html, 2003.

[2] L. A. Adamic and E. Adar. How to search a social network. *Social Networks*, 27(3):187 – 203, 2005.

[3] E. Ben-Naim, H. Frauenfelder, and Z. Toroczkai, editors. *Information Dynamics in the Networked World*, Lecture Notes in Physics. Springer, 2003.

[4] E. Bonabeau. Graph multidimensional scaling with self-organizing maps. *Information Sciences*, 143:159 – 180, 2002.

[5] E. Bonabeau and F. Hnaux. Self-organizing maps for drawing large graphs. *Information Processing Letters*, 67:177 – 184, 1998.

[6] D. Boyd and J. Potter. Social network fragments: an interactive tool for exploring digital social connections. In *GRAPH '03: Proceedings of the SIGGRAPH 2003 conference on Sketches & applications*, pages 1–1, New York, NY, USA, 2003. ACM Press.

[7] U. Brandes and T. Erlebach, editors. *Network Analysis: Methodological Foundations*, volume 3418 of *Lecture Notes in Computer Science*. Springer, 2005.

[8] E. H. Chi, J. Pitkow, J. Mackinlay, P. Pirolli, R. Gossweiler, and S. K. Card. Visualizing the evolution of web ecologies. In *CHI '98: ACM CHI 98 Conference on Human Factors in Computing Systems*, pages 400–407, 644–645, New York, NY, USA, 1998. ACM Press.

[9] F. J. D. Auber, Y. Chiricota and G. Melancon. Multiscale visualization of small world networks. In *IEEE Symposium on Information Visualization 2003*, pages 75–81, 2003.

[10] T. Dwyer. A scalable method for visualising changes in portfolio data. In *Proceedings of the Australasian Symposium on Information Visualisation (InVis.au'03)*, pages 17–25. CRPIT, 2003.

[11] P. Eades, W. Lai, and X. Mendonca. A visualizer for e-mail traffic. In *Proceedings of 4th International Pacific Graphics Conference / CADDM'94*, 1994.

[12] P. Gloor. Capturing team dynamics through temporal social surfaces. In *Proceedings of 9th IEEE International Conference on Information Visualisation IV05*, pages 6–8, 2005.

[13] P. A. Gloor, R. Laubacher, S. B. C. Dynes, and Y. Zhao. Visualization of communication patterns in collaborative innovation networks - analysis of some w3c working groups. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 56–60, New York, NY, USA, 2003. ACM Press.

[14] R. Guimer, L. Danon, A. Daz-Guilera, and F. G. Y. A. Arenas. The real communication network behind the formal chart: Community structure in organizations. In *7th Granada Seminar on Computational and Statistical Physics*, Granada, Spain, 2002.

[15] S. Hong and N. Nikolov. Hierarchical layout of directed graphs in three dimensions. In *Proceedings of 13th International Symposium on Graph Drawing*, page to appear, 2005.

[16] S. Hong and N. S. Nikolov. Layered drawings of directed graphs in three dimensions. In S. Hong, editor, *Information Visualisation 2005, Asia-Pacific Symposium on Information Visualisation (APVIS2005)*, volume 45, pages 69–74. CRPIT, 2005.

[17] B. J. Kerr. Thread arcs: An email thread visualization. In *IEEE Symposium on Information Visualization 2003 (INFOVIS 2003)*, pages 211– 218, Oct. 2003.

[18] S. Kettner, C. Madden, and R. Ziegler. Direct rotational interaction with a spherical projection. In *Interaction: Systems, Practice and Theory Proceedings*, 2004.

[19] B. Klimt and Y. Yang. Introducing the Enron corpus. In *Proceedings of First Conference on Email and Anti-Spam (CEAS)*, 2004.

[20] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin Heidelberg, 3rd. edition, 2003.

[21] W.-J. Li, S. Hershkop, and S. J. Stolfo. Email archive analysis through graphical visualization. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 128–132, New York, NY, USA, 2004. ACM Press.

[22] M. Mandic and A. Kerne. Using intimacy, chronology and zooming to visualize rhythms in email experience. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1617–1620, New York, NY, USA, 2005. ACM Press.

[23] D. W. McDonald. Recommending collaboration with social networks: a comparative evaluation. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 593–600, New York, NY, USA, 2003. ACM Press.

[24] B. Meyer. Self-organizing graphs - a neural network perspective of graph layout. In S. Whitesides, editor, *Proceedings of the 6th International Symposium on Graph Drawing*, pages 246 – 262, London, UK, 1998. Springer-Verlag.

[25] D. Moody, J. McFarland and S. Bender-deMoll. Dynamic network visualization. *American Journal of Sociology*, 110(4):1206–41, January 2005.

[26] M. E. J. Newman, S. Forrest, and J. Balthrop. Email networks and the spread of computer viruses. *Physical Review*, 66:1 – 4, 2002.

[27] W. Nooy, A. Mrvar, and V. Batagelj. *Exploratory Social Network Analysis with Pajek*. CAMBRIDGE UNIVERSITY PRESS, 40 West 20th Street, New York, NY 10011-4211, USA, 2005.

[28] M. F. Schwartz and D. C. M. Wood. Discovering shared interests using graph analysis. *Communications of the ACM*, 36:78 – 89, 1993.

[29] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, February 1981.

[30] J. R. Tyler, D. M. Wilkinson, and B. A. Huberman. Email as spectroscopy: Automated discovery of community structure within organizations. *Communities and technologies*, pages 81 – 96, 2003.

[31] F. van Ham and J. J. van Wijk. Interactive visualization of small world graphs. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04)*, pages 199–206, Washington, DC, USA, 2004. IEEE Computer Society.

[32] G. D. Venolia and C. Neustaedter. Understanding sequence and reply relationships within email conversations: a mixed-model visualization. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 361–368, New York, NY, USA, 2003. ACM Press.

[33] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applicaitons*. Cambridge University Press, 40 West 20th Street, New York, NY 10011-4211, USA, 1st. edition, 1995.

[34] F. Wu, B. A. Huberman, L. A. Adamic, and J. R. Tyler. Information flow in social groups. *Physica A*, 337:327 – 335, 2004.

[35] L. Zhang, N. Tu, and D. Vronay. Info-lotus: a peripheral visualization for email notification. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1901–1904, New York, NY, USA, 2005. ACM Press.