

# MULTI-NODE MONITORING AND INTRUSION DETECTION

Robert F. Erbacher, Zhouxuan Teng, and Siddharth Pandit  
Department of Computer Science, LI 67A  
University at Albany-SUNY  
Albany, NY 12222, USA  
(erbacher, zxteng01, pandits)@cs.albany.edu

## Abstract

The monitoring of systems connected to the Internet is critical for the maintenance of security and privacy. The threat of hackers, terrorism, and internal misuse are major concerns of any organization. In this paper, we expand our visual monitoring environment to support multiple monitored systems and provide an effective layout of the nodes (hosts) for the analysis of the networked environment. We discuss the analysis and correlation strategies needed in such a multi-host environment in order to identify unusual activity. The effectiveness of the correlation and analysis activities is directly related to the node organization. We will show that the node layout we have developed leads to a very effective organization in that line intersections and line orientations are designed to be informative and indicative of unusual activity. Given the effectiveness of line intersections and line orientations as visual attractors, as they are discerned pre-attentively [1], this leads to a very effective monitoring environment. Given our goal is to provide an additional tool to system administrators with the understanding that this is not their sole task then the ready discrimination and identification of activity needing attention is crucial.

**Keywords:** Information Visualization, Computer Security, Intrusion Detection

## 1. Introduction

Identification of intrusions and misuses is dependent on the identification of anomalous behavior. A single event will rarely identify sufficient activity to indicate an intrusion or misuse as these individual events will occur frequently. It is multiple events in correlation that is representative of a true attack. We have shown how temporally related events in conjunction with a single monitored host can be indicative of an attack [2]. These temporally related events may come from the same machine or from different machines. However, most organizations maintain large networks of computer systems that are subject to attack. We must be able to monitor all of these systems simultaneously and identify spatially as well as temporally related events. This can take one of three forms:

1. When an attack is identified against a single system within the organization, it is likely that many or all

of the systems have been so attacked. Were any successfully broken into?

2. After attackers have successfully compromised one system they will generally use that system to attack others, both within the organization and without. This is generally how it is learned that a system is compromised, through reports from other organizations that attacks are originating from a given system.
3. Third, an attack may be coordinated from more than a single machine and these events must be correlated.

In all three cases, it is critical that we are able to monitor the activity and interactivity between the systems, both internal and external. Suppose a user connects from one machine within the organization to another. Is this expected usage or unexpected usage [2]? Identification of the actual user and which systems are being accessed are critical to identification of the user's intentions and the typical expectations of the environment.

## 2. Previous Work

System administrators are currently very limited as to the tools they have available to aid in the monitoring and identification of intrusions and misuses of computer systems under their auspices. While many tools are available to identify different types of attacks, such as Portsentry's identification of port probes [3, 4], BlackIce [5], TripWire [6], and Cops [7], they generally provide their results as additional messages appended to the standard system log files. No single tool is currently the end-all be-all of intrusion detection. Therefore, it is necessary to run many such tools and correlate their results. This will assist identification of real full blown attacks amid many false alarms, limited probes, and the morass of day to day activities.

Consequently, the system log files are a system administrator's principal line of defense against intrusions. The textual nature of such log files makes the analysis task daunting as the administrator must examine an enormous number of messages, on the order of thousands a day for a moderately sized network, and correlate the messages over time to determine if activity is a true threat or inconsequential. Our

intrusion detection monitoring environment [8, 9] is geared towards aiding in this analysis.

The original environment showed a single monitored system in the center of the window and connecting systems in concentric circles around the monitored system. Node positions are maintained over time to aid correlation of activity. Initial node positions are determined based on the remote systems IP address and additionally on a first-come first-serve basis. The environment uses glyphs to visually represent parameters from the system log files, including: time of day, system load, type of connection, number of users, number of connections, state of connections, etc.

### 3. Multiple Monitored Systems

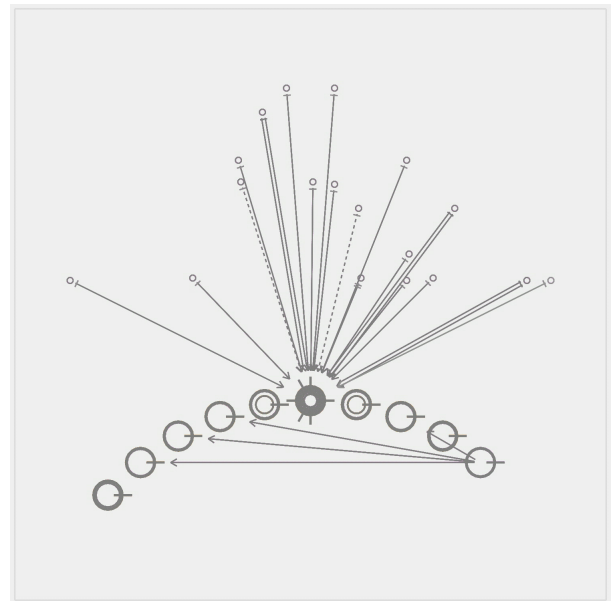
Our first goal in expanding on our visualization of a single monitored system was the identification of an effective node layout for the many nodes that will be needed to represent a typical organization's networked infrastructure. The goal is to improve the readability of the graph. "Readability issues can be expressed by means of aesthetic criteria, such as the minimization of crossings between edges and the display of symmetries." [10] Typical graph drawing algorithms are insufficient as they are geared towards an arbitrary layout of the nodes. The application of force-based techniques maps better as we can apply these forces to map local nodes and remote nodes to separate locations on the screen and to differentiate remote nodes as we did with the single monitored node technique. The placement of nodes will have an inherent meaning or implication as to the organization of the network for the system administrator. This must be reconciled with the somewhat arbitrary nature of automated node layout algorithms. We have the additional constraint in that the representation must maximize the number of nodes presented on a single display, minimizing the amount of white space.

A final issue that tends to provide a significant obstacle to most graph drawing techniques is the dynamic nature of our data. Since we are providing an online monitoring environment, we do not know ahead of time details about the presence of individual nodes, their connectivity, duration, or changes in connectivity. We must also ensure that node positions are maintained over time to aid correlation, a requirement not provided by most layout algorithms. Since our goal is to provide system administrators with immediate identification of suspicious activity, the online nature of the environment is critical. An off-line approach runs every now and then will give a hacker a substantial amount of time to cause damage to the system and compromise additional facilities. Ultimately, post-mortem static visualization techniques will be used to reinforce and review activity, acting as an additional line of defense.

Other techniques, such as radial graphs [11] or core trees [12], are far too arbitrary in nature, and do not provide consistency of node placement, and do not

provide other positioning criteria needed in our layout requirements. Macroscopic techniques [13] do not provide sufficient detail for widely applicable intrusion detection and analysis. Ultimately, the techniques we are looking at can be related to flattened cone trees [14] without the arbitrariness of such layouts and with additional specialized requirements for the layouts.

We ultimately resolved an effective discriminating layout. Maturation of this layout required extensive experimentation as we wished to minimize the number of crossed edges except for where the crossed edge provided meaning. Figure 1 provides a clean example of this node layout. Remote connections can be clearly seen to be separated from remote connections, eliminating any unnecessary crossed edges.



**Figure 1:** Application of heuristic measures to strengthen node relationships by providing locality of spatial positioning in conjunction with locality of temporal accesses.

#### 3.1. Multi-node Layout

The layout exemplified in figure 1 was derived from work relating to gravity and force-based graph drawing and can be thought of as incorporating two opposing forces, figure 2. More specifically, the layout can be envisioned as having two gravity wells. This concept differs from other work with force-based graph drawing techniques as we allow forces to exist without the presence of nodes. The local nodes are attracted to the gravity well at the bottom of the diagram while the remote nodes are attracted to the gravity well at the top.

Remote nodes are additionally impacted by the difference between their IP address and that of the local systems. Thus, the greater the deviation in IP addresses the closer to the gravity well the node will be placed. Nodes are placed adjacent to each other along the force line, beginning in the center of the force line and working outwards. When no further nodes may be

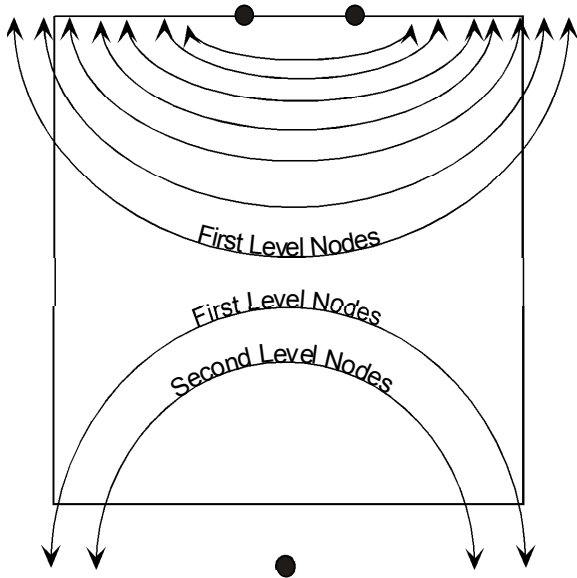


Figure 2: Gravity-based force mapping diagram.

placed on the current force line then the strength of the field is envisioned to increase and the nodes are placed closer to the gravity well. Several adjacent rings are made available for this filling process. The number of available such rings is based on the expected number of nodes in a given ring. Thus, the number of expected nodes with only a single component differing in the IP addresses is significantly less than that with all four components differing. Spacing is incorporated into the equations for thirteen distinct force lines before overlapping will occur. The environment for remote nodes can be modeled with elliptical equations as follows:

$$a_i = 1.0, b_i = 0.5, x_i = 0.0$$

$$r = (d + 1) + \lambda$$

$$b = b_i - r * \frac{0.5}{13}$$

$$a = a_i - \frac{r}{13}$$

$$x = \pm N_r * 2$$

$$y = \sqrt{b^2 - \frac{(x^2 * b^2)}{a^2}}$$

$N_r$  = Radius of nodes

$r$  = Vertical radius of force lines

$d$  = Number of different components between remote and local IP addresses

$\lambda$  = Number of force lines filled for a given  $d$  value

The local nodes can be modeled similarly with circular equations.

The decision on the appearance of the force lines was made based on the expected interaction between the various nodes. We will never have details as to activity from one remote node to another. We will also

not be aware of connections from local nodes directed to remote nodes. Therefore, the layout is designed to optimize the visual clarity of connectivity between local nodes and from remote nodes to local nodes. The curvature of the force lines ensures that we will be able to effectively show connections from any remote node to any local node and from any local node to another local node. This explains the chosen curvatures for the force lines associated with local versus remote nodes.

### 3.2. Heuristic Approaches

It should be noted that the example in figure 1 has begun to incorporate heuristics into the node positioning algorithm. These heuristics help ensure that node positioning is effective. For example, the main node receiving many external connections is a local server expecting such connections. Should we not have been using heuristics, this node could have been placed anywhere, including ineffective very positions, which was in fact the cause for our inclusion of such techniques.

The heuristics can be related to the virtual memory management protocols from operating systems research [16]. As mentioned in relation to our environment for monitoring a single host, we attempt to maintain a node's position over time in order to provide a sense of temporal continuity. This greatly aids the identification of temporally related events. However, in many environments, nodes will connect on a *very* infrequent basis or only as a one time event. With these types of events, it is valuable to be able to *reuse* node positions. This is where such heuristics come into play. We can monitor the activity of individual hosts and replace their node positions over time when deemed appropriate.

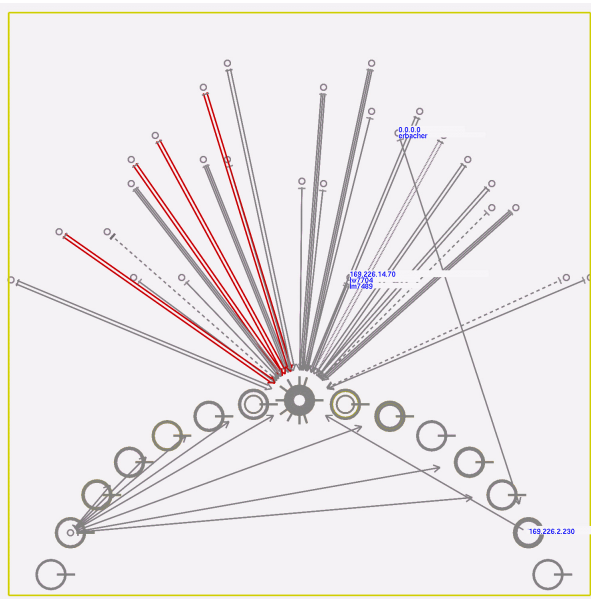
In the current model, each active connection is given a temporally increasing weight. When a remote node's connection weight exceeds that of another remote node that is connecting to the same local node and the latter remote node is spatially located closer to the local node then the node positions are swapped. With local nodes a similar methodology is applied such that over time a highly active node will have its weight factor increased, allowing it to replace nodes spatially closer to the remote nodes, i.e., moving closer to the apex of the semicircle of nodes.

This paradigm is related to spring forces in which the weight of the interconnections corresponds to the spring strength. Given a finite number of node positions, in order for the spring force to draw a remote node closer to a local node the remote node must *knock out* some other remote node in order to take its place. While we described this as swapping positions in essence it will be a repetitive process in which the node just knocked out will now potentially be able to replace other nodes. This process repeats until a stable state is achieved.

## 4. User-Based Correlation

An additional direction of analysis is the correlation of the activities of individual users over time. This is critical for identification of misuse and compromised user accounts. Given the number of users typically connecting to the systems within the university, or any other organization for that matter, visually representing each user would be an impossibility. Humans are not able to differentiate the 150-200 different visual characteristics that would be needed to represent the individual users [17].

An alternate strategy is providing user information through feedback and user interaction. For example, an individual connecting to the university's primary server would not raise attention. However, if a user was connecting directly to a workstation rather than the primary server then suspicion would be raised. Additionally, should a connection be made from one local workstation or PC to another then additional analysis is likely warranted. Why connect from one workstation to another? Is something wrong with the workstation?



**Figure 3:** User interaction, selection, and feedback example. IP Address is shown above user list when available. Identifiers are shown in blue on the screen to differentiate them from other characteristics.

Once additional analysis is deemed warranted then the user may select the node or link, getting feedback as to the user or users making such connections. This can then be correlated with other user activity. Is this same user connected to other systems? This can be represented visually through the application of user initiated probing and feedback, figure 3. An enhancement would be to allow the highlighting of selected users or nodes, greatly enhancing the visual response. Figure 4 shows a zoomed in view of the visual response of the user directed interaction. As can be seen in the figures, we have selected three nodes. Each node provides the IP

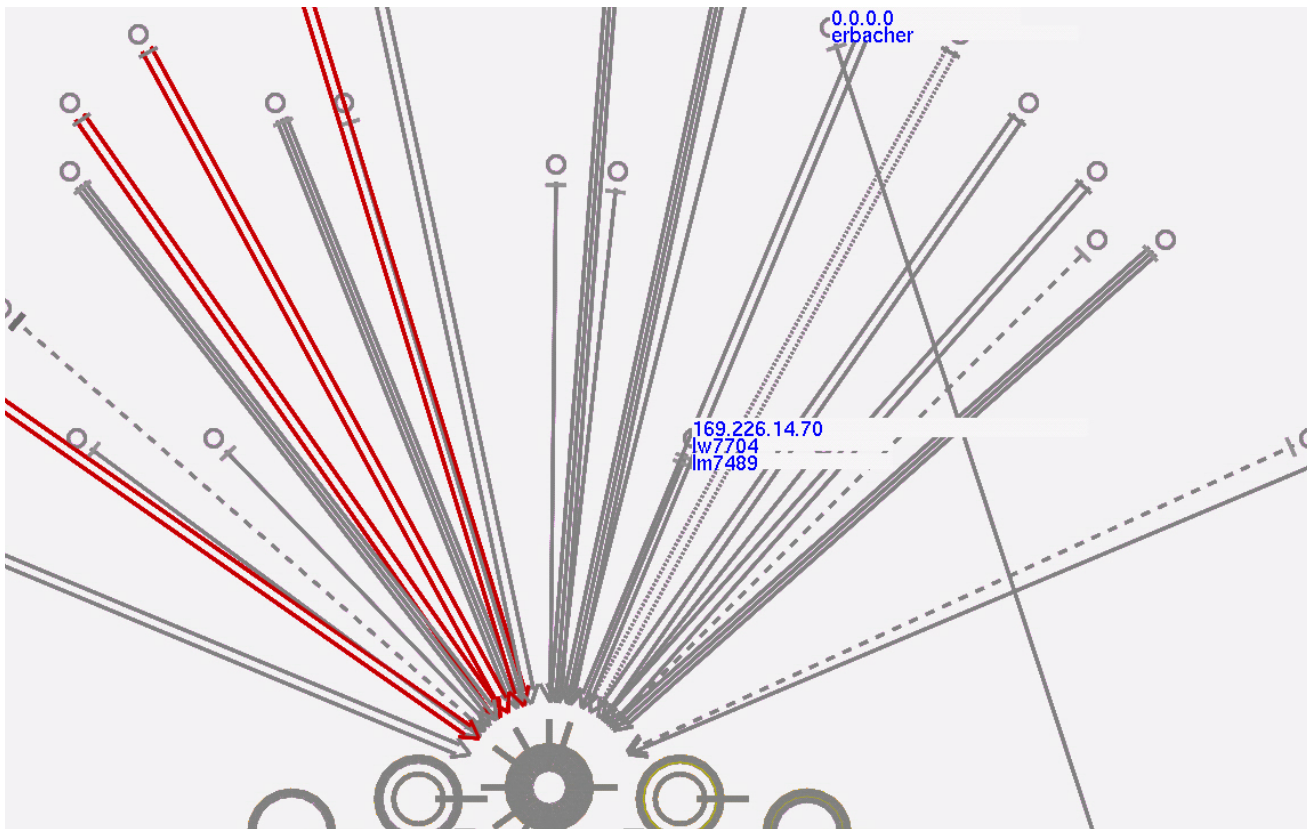
Address of the host represented by that node and any user information available from the host. The selected host at the top of the display has a single user connecting from it to a local system. The node in the middle has two users. This is indicated by the two hash marks below the node itself and reaffirmed by the node's feedback, providing the individual's user names. The final node highlighted is a local system. While there is an individual connecting from this node the user information is not available.

In general, user information will not be available from systems in which we are only identifying an inetd related event. These inetd events indicate an initial port connection before authentication has been completed. Obviously if authentication has not completed then no user information will be available. Inetd connections are indicated by a node with a set of parallel lines drawn between it and the node it is connecting to. The style of the lines is indicative of the port connected to, i.e., the type of connection initiated. Other systems do not provide the ability to collect the related user information, which significantly hinders the analysis process. This impact will greatly effect the user policies and logging mechanisms employed by an organization.

## 5. Multi-node Attack Analysis

In our prior work, we discussed how attacks on a single system can be visually discriminated within the monitoring environment. With multiple monitored systems this methodology must be extended. First, we must continue to correlate activity temporally. However, we must also correlate activity spatially. The environment provides the ability to visually correlate activities spatially and temporally that otherwise would not be easily identifiable. This is critical in a typical computing environment in which many systems are available but widely distributed throughout the organization. Examples of a few of the types of activity of interest and which cannot be identified with a single host approach, include:

- Users unexpectedly connecting to non-server based machines. These *workstations* are expected to be used locally only.
- Connections to many hosts simultaneously or in succession from a single remote host. Since all the systems are essentially configured the same, should the primary server not be sufficient for a user's need then any single workstation should do.
- A user connecting through a local machine to get to other local machines when not expected to. This is a common behavior once a single system has been compromised in order to expand the intruder's domain and available resources.
- A user from a local machine connecting to another local machine. This is similar to the prior example except the user appears to be logged on locally to a machine.



**Figure 4:** Zoomed in view showing feedback from a user initiated probe of network hosts. Examples show both one and two users connecting from the given hosts. The identifiers can be made out more easily in this zoomed view.

- Excessive CPU load on many systems simultaneously. This could be an indication of misuse or an actual attack on the systems.
- Excessively short or exceedingly long connection times, particularly when this activity occurs repeatedly.

As can be seen in the example figures, each of these types of activity will stand out clearly given our node layout algorithm.

## 6. Attack Examples

Figure 1 provides one example of activity on a collection of monitored systems. In this scenario, we have a large number of remote users connecting to one principal server, as is expected. We also have a single internal workstation connecting to many other internal systems. Is this expected? Is it acceptable? In fact, this is a system administrator connecting to the various systems for typical administrative duties. This fact is easily verifiable but is something that *needs* verification. Had this not been a system administrator's activity then the activity would have been considered far more unsavory. Additional types of attacks are similarly identifiable in the visual monitoring environment. These new analysis capabilities enhance our previous capabilities by providing monitoring intrinsic to a multi-host environment.

Additional suspicious activities are shown in figures 3 and 4. First, there is a user connecting from

the outside world to an internal system indicated by the IP Address 169.226.2.230. There is then a connection from this system to the main server. This is in fact the same user connecting through an intermediary before connecting to the main server. This connection is the result of one of the authors connecting to his lab-based server and subsequently connecting to the main server in order to verify that the data collection scripts were operating correctly. The important aspect of this display is the clarity of the visualization. The intersecting lines, as the connection crosses all of the other remote connection lines, stands out very clearly and cries out to be probed and analyzed. The fact that someone is hopping through a local workstation also stands out quite visibly, creating a characteristic V shape with one system acting as the foci.

## 7. Future Work

Even with the described effectiveness of the environment there remain improvements to be made. First, a more extensive heuristically-based approach is needed, including an analysis of the effectiveness of various heuristic measures for the node layout algorithm.

Second, the interaction and feedback capabilities need extending. As suggested earlier, we need the ability to select connections on user, such that we can easily identify all connections related to a single selected user. This will then need to be extended such that we can select and isolate the activity of a given

host or user. In this way, when suspicious activity is identified, either the host or the user can be selected and the associated activity over time can then be reviewed in isolation from other activity. This will greatly aid the accuracy and level of analysis performed.

Finally, the ability to provide a static display of activity occurring over time is needed. The amount of data makes this useless without the described monitoring capabilities but will be used in close conjunction with the user and host-based selection and isolation facilities.

## 8. Conclusions

We have described the enhancements to our intrusion monitoring environment for multi-node monitoring. In particular, we focused on the advantages of our layout algorithm and the incorporation of heuristically-based techniques for improving the layout algorithm. The application to pre-attentive vision ensures the visual model will provide the needed insights in a large networked infrastructure.

Additionally, we discussed the incorporation of selection and feedback capabilities that allow for the identification and monitoring of users and hosts within the environment. In conjunction with the user-based analysis and node layout algorithm we described typical intrusion and misuse identification activities through the visualization environment, providing some rather powerful examples of the techniques effectiveness.

## 9. References

- [1] Charles A. Kesley, "Detection of Vision Information," *The Perception of Visual Information*, William R. Hendee and Peter N.T. Well Editors, 2nd Edition, Springer-Verlag, 1997.
- [2] Robert F. Erbacher, "Visual Behavior Characterization for Intrusion Detection in Large Scale Systems," *Proceedings of the IASTED International Conference On Visualization, Imaging, and Image Processing*, Marbella, Spain, September 3 - 5, 2001, pp. 54-59.
- [3] Stuart McClure, Joel Scambray, and George Kurtz, *Hacking Exposed: Network Security Secrets & Solutions*, Third Edition, Osborne/McGraw-Hill, 2001.
- [4] <http://www.psionic.com/products/>.
- [5] D. Rae and D. Ludlow, "Halt! Who goes there? [Internet intrusion detection benchtest]," *Network News (UK Edition)*, February 16, 2000, pp. 31-37.
- [6] G.H. Kim, E.H. Spafford, "Writing, supporting, and evaluating Tripwire: a publically available security tool," *Proceedings of the 1994 USENIX UNIX Applications Development Symposium*, April 1994, pp. 89-107.
- [7] J.A. Fore, "System security: when enough is not enough," *Proceedings of Expanding Expectations in Integrated Online Library Systems (IOLS)*, New York, 1997, pp. 53-62.
- [8] Robert F. Erbacher, Kenneth L. Walker, and Deborah A. Frincke, "Intrusion and Misuse Detection in Large-Scale Systems," *Computer Graphics and Applications*, Vol. 22, No. 1, January/February 2002, pp. 38-48.
- [9] Robert F. Erbacher and Deborah Frincke, "Visualization in Detection of Intrusions and Misuse in Large Scale Networks," *Proceedings of the International Conference on Information Visualization '2000*, London, UK, July, 2000, pp. 294-299.
- [10] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Joannis Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice Hall, 1999.
- [11] Ka-Ping Yee, Danyel Fisher, Rachna Dhamija, and Marti Hearst, "Animated Exploration of Dynamic Graphs with Radial Layout," *Proceedings of the IEEE Symposium on Information Visualization*, San Diego, CA, October 22-23, 2001.
- [12] Cheng-Zen Yang and Chiun-How Kao. "Visualizing Large Hierarchical Information Structures in Digital Libraries." In *Proceedings of the Second Asian Digital Library Conference*, Taipei, Taiwan, ROC, November 8-9, 1999, pp. 217-225.
- [13] [http://www.caida.org/analysis/topology/as\\_core\\_network/](http://www.caida.org/analysis/topology/as_core_network/)
- [14] G. G. Robertson, J. D. Mackinlay, and S. K. Card, "Cone trees: Animated 3d visualizations of hierarchical information," In S. P. Robertson, G. M. Olson, and J. S. Olson, editors, *Proc. ACM Computer-Human Interaction '91 Conference on Human Factors in Computing Systems*, New York, 1991, ACM Press, pp. 189-194.
- [15] Ingo Brass and Arne Frick, "Fast interactive 3-D graph visualization," *Proceedings of Graph Drawing '95*, 1995, pp. 99-110.
- [16] William Stallings, *Operating Systems*, Fourth Edition, Prentice Hall, 2001.
- [17] Edward R. Tufte, *Envisioning Information*, Graphics Press, 1990.