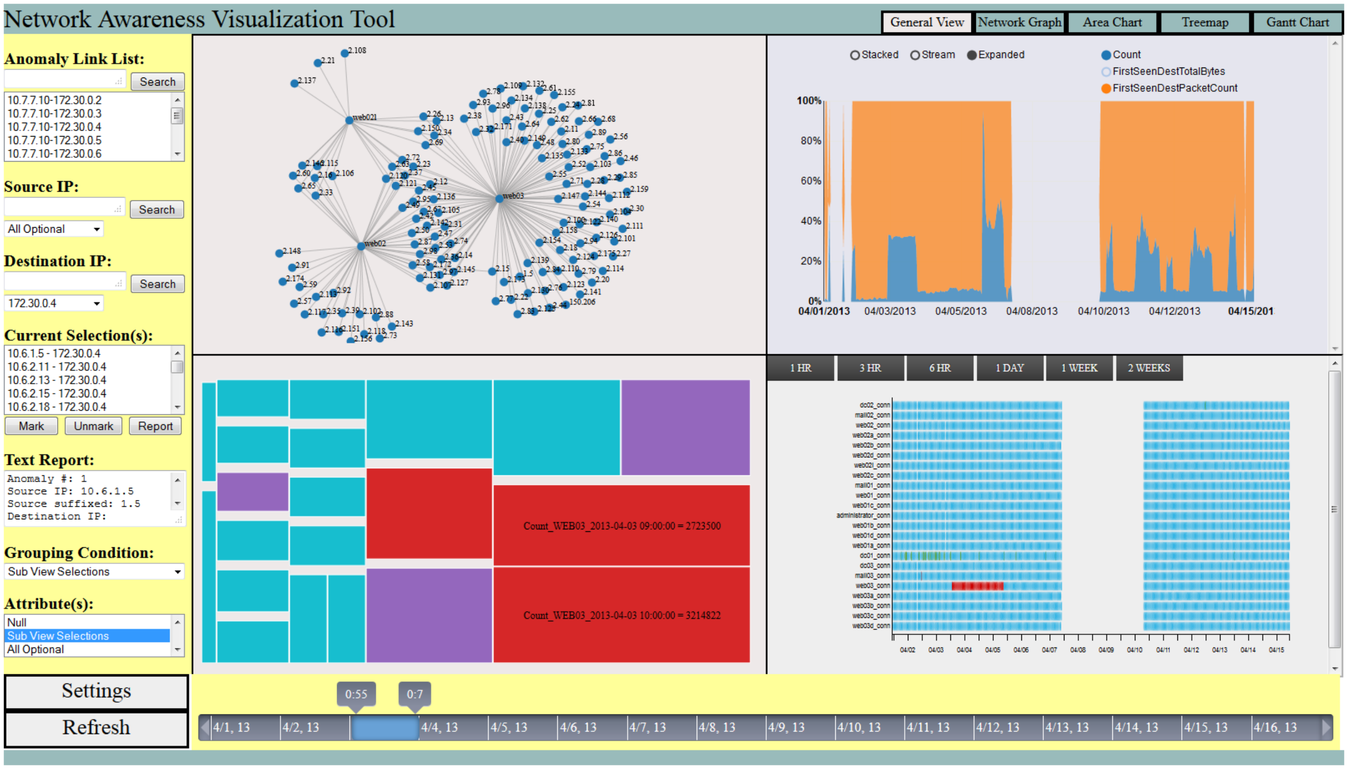
Bridging the Gap of Network Management and Anomaly Detection through Interactive Visualization

Tao Zhang[*] Qi Liao[†] Lei Shi[‡]



Network Awareness Visualization Tool

Anomaly Link List:
10.7.7.10-172.30.0.2
10.7.7.10-172.30.0.3
10.7.7.10-172.30.0.4
10.7.7.10-172.30.0.5
10.7.7.10-172.30.0.6

Source IP: All Optional

Destination IP: 172.30.0.4

Current Selection(s):
10.6.1.5 - 172.30.0.4
10.6.2.11 - 172.30.0.4
10.6.2.13 - 172.30.0.4
10.6.2.15 - 172.30.0.4
10.6.2.18 - 172.30.0.4

Text Report:

```
Anomaly #: 1
Source IP: 10.6.1.5
Source suffixed: 1.5
Destination IP:
```

Grouping Condition: Sub View Selections

Attribute(s): Null, Sub View Selections, All Optional

Count_WEB03_2013-04-03 09:00:00 = 2723500

Count_WEB03_2013-04-03 10:00:00 = 3214822

Figure 1: The overview of web-based visualization tool for analyzing the network and system anomalies in standard log files.

Abstract

Large-scale networks have become increasingly challenging to manage. It is vital for a system administrator or network manager to be able to analyze the vast amount of log data in order to detect suspicious behaviors or patterns, possibly due to malicious users/applications or faulty devices. While an intrusion detection system (IDS) log can provide a large number of warnings, exactly which alarms are true while the others are false, and more importantly what are the underlying causes are still difficult to know. To bridge the gap between network log and anomaly discovery, we design and implement a visualization tool that combines multiple commodity visualizations with minimum learning curve. While each individual view is well understood, the effects of such views in analyzing network anomalies are not well studied. Since each visualization technique has advantages as well as limitations in addressing a particular task, we show that these views, when combined and linked together, may provide an effective and lightweight network anomaly analysis tool. The web-based open platform may simplify network administration as well as promote collaborative analysis among researchers.

1 Introduction

One important daily task of network administrators and operators is to detect potentially bad or anomalous activities from a large amount of standard log data such as netflow, packet traces, syslog and IDS. Network management dataset includes different types of records reporting a wide range of network and system status. The challenges not only lie in the large volume of data but also the highly dynamic nature of network traffic (i.e., users may come and go, connections may be built and torn down at any moment). The network data is usually highly dimensional with dozens or even hundreds of attributes. Choosing which attributes to examine can be a daunting task and computationally infeasible by automatic processes. Therefore, bringing domain experts into the loop through interactive visualization is promising. While there have been network management tools, few of them are lightweight enough and actually geared towards anomaly detection in dynamic traffic data. In the paper, we design and implement a generic network log analysis and visualization tool for situation awareness, anomaly detection and event investigation.

[*] e-mail: zhang3t@cmich.edu, Department of Computer Science, Central Michigan University.

[†] e-mail: liao1q@cmich.edu, Department of Computer Science, Central Michigan University.

[‡] e-mail: shil@ios.ac.cn, State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences.

In particular, we utilize a few commodity visualization methods, i.e., area charts, Gantt charts, Treemaps and network graphs, by taking advantages of user familiarity and robustness. While each method has been well studied, it is still not clear which view is suitable for detecting which type of anomalies and for analysing which type of network log with different characteristics. Network data types have different properties, which makes them hard to be dealt with using any particular visualization technique. For example, a network graph could be constructed by flow or packet data collected during a specific time period. Unlike social networks, computer networks are more dynamic in terms of topological changes. Although we could extract the changing information and utilize graph visualization solutions to fit the dynamic nature, other attributes or types of log data may not be appropriate using a graph view. Area charts or Gantt charts, on the other hand, may be a better choice for quantitatively comparing time dynamics and trends.

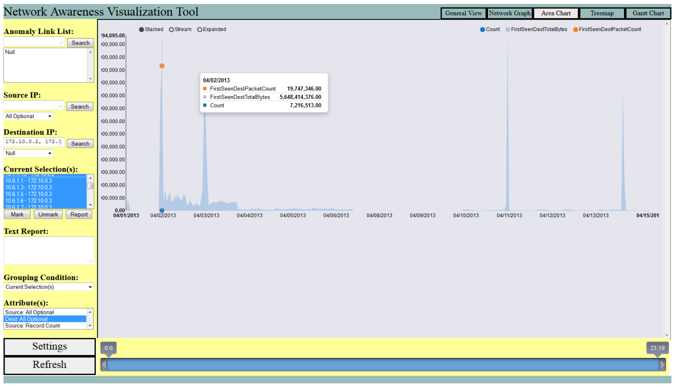
With this motivation, we study the relationship between different visualizations in analyzing different network characteristics and finally integrate them in a unified view. Based on their pros and cons, we study how each individual view is useful at different granularity levels to help network administrators address both obvious and subtle events. We note that our contributions are on how these general views, when combined, may help detect network anomalies from common log files, and on the study of the feasibility of such views for particular network anomaly detection tasks. Implemented with D3, the lightweight, web-based visualization platform allows network administrators and other researchers to easily view and collaborate on security data analysis and visualization.

2 Related Work

There have been common visualization methods that are potentially useful for analyzing general data. For example, Gantt charts [13] are widely used in project managements, job scheduling [8], etc. Area charts such as line charts are well established for understanding quantitative data. It is intuitive to show trends over time among some attributes of network log data such as packet transmission, data usage, etc. The comparison of attributes by both numbers and percentages usually helps to pin down exact start and end times for interesting events. Since networks can be naturally organized into trees and graphs, there has been a vast amount of work to design different visualization solutions addressing issues of viewability or usability, e.g. directed-edge representation [7], edge bundling method [4], fisheye tree [1,5], cone tree [6], information cube [10], maxent-stress model [3], and dynamic ego network visualization [11]. Treemap [1], as another visualization solution, has a different role in our work. Considering the comparable design of sizes and colors of Treemap, we embed quantities to show significant differences between nodes' unsynchronized performances and keep them in a time-node hierarchical structure.

Other visualizations have been designed that may potentially help administrators to investigate and detect anomalies in network traffic data. For example, a parallel axes [14] view has been used to display NetFlow records and generate network traffic patterns of both normal and malicious behavior. FlowScan [9] uses one area-chart-like plotting solution to analyze flow data and provides a continuous view of the network traffic. In addition, network anomalies may also be predicted by directly visualizing the statistical result [2].

While parallel axes [14] and FlowScan [9] primarily target at the flow data, networked systems also include many other important aspects such as machines status, IDS alarms, etc. Using only one pattern layer without considering other attributes may have issues for detecting actual anomalous network activities. Visualizing statistical results [2] that have a tendency for predicting anomalies is like a double edged sword. Although accurate prediction will efficiently help investigators, wrong results will be possible to mislead administrators. We try to unitize multiple visualizations to process a variety of network log data of standard formats. A collaborative information linking method [12] may enhance the current general view by bridging visualizations. Visual links across different views could quickly lead the user to discover the relative knowledge in other views after selecting an item.



Anomaly Link List: Null

Source IP: All Optional

Destination IP: 172.10.0.2, 172.1 Null

Current Selection(s):
10.6.1.1 - 172.10.0.3
10.6.1.3 - 172.10.0.3
10.6.1.5 - 172.10.0.3
10.6.1.6 - 172.10.0.3
10.6.1.7 - 172.10.0.3

Grouping Condition: Current Selection(s)

Attribute(s):
Source: All Optional
Dest: All Optional
Source: Record Count

04/02/2013
FirstSeenDestPacketCount 19,747,346.00
FirstSeenDestTotalBytes 5,648,414,376.00
Count 7,216,513.00

Figure 2: Attributes of server machines are summed by time slices. In the stacked area charts, each of the attributes (log record counts, first-seen-total-bytes, and first-seen-packet-count) is plotted with a unique color.

Lastly, there are also commercial network management tools such as HP OpenView, IBM Tivoli, CA NSM, Splunk, etc. While these are more heavy weight solutions that manage, aggregate and visualize network and system log data in one central place, we focus on a more lightweight version that focuses on security related events and network anomaly detection.

3 Visualizations For Network Anomaly Analysis

When a network administrator or manager starts to analyze the log data for potential network anomalies, he faces two challenges. First, how to locate the interesting events and pin down to specific time periods. Second, in order to find the interesting event, how to narrow the whole dataset down to a few important attributes. In other words, how to find the useful data fields that we should care. These challenges are real due to the vast amount of log data sent to the manager's workstation and the dynamic nature of the data as well.

In terms of the choice of visualizations, the basic design principle is to keep it simple and general. Our intuition is to use existing and well-established views by taking the advantage of user familiarity and visualization robustness. Among these views, we choose area charts, Gantt charts, Treemaps, and network graphs since each view may have limitations and may only be suitable for certain tasks. We study the pros and cons of these views in the setting of network management and anomaly detection tasks, and more importantly, how these views, when linked together, can provide a much better situation awareness and investigation assistance to system administrators and network managers.

Figure 1 shows an overview of the developed visualization tool. On the left panel, there are options for users to interact with the network log data. Users can select the source and destination nodes, connections, and combinations of attributes on each system. Upon selection, the views on the right will be automatically updated. Specifically, the top-left panel reflects the network (sub)graphs based on the node/edge selections. The top-right panel shows the area chart for the trend of selected attributes on systems. The bottom-left panel provides quick view of attribute value magnitudes over selected time. The bottom-right panel demonstrates through Gantt charts of attribute values of systems across the customized time period. There are additional interactive visual items

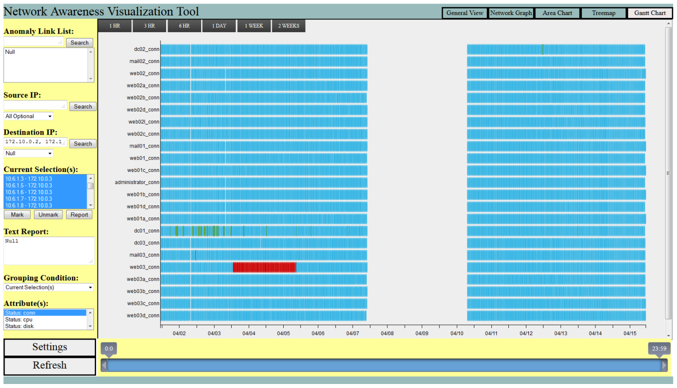


Figure 3: A Gantt chart showing the *connection* status of servers (y-axis) by the timeline (x-axis). Colors indicate the severity of events (similar to syslog levels). It is obvious that server ‘web03’ has abnormality. To pin down the exact start and end times, one can click the upper left buttons to zoom along x-axis of time units.

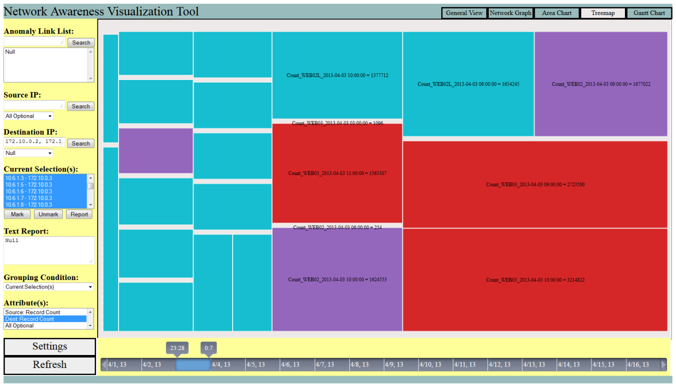


Figure 4: Rectangles in the Treemap represent the record counts grouped by time unit from the server machines with unique colors. The rectangle of the server ‘web03’ from 9 to 12am dominates on April 3rd. In addition, servers ‘web02’ and ‘web02l’ are also noticeable.

such as tabs for switching views, buttons for refreshing/settings, and slider bars for selecting arbitrary investigation time windows.

3.1 Area Charts and Gantt Charts

We formulate anomaly detection tasks into layers. As any investigation process, there must be a starting point. Summary statistics in this case can provide a good starting point as for where in the dataset to begin an investigation. To that end, we apply an area chart with summary values, i.e., we collect data records by attributes and plot total summations of attributes at each time unit. The administrator could quickly identify the “interesting attributes” if the lines fluctuate significantly during the time of investigation. Correlation of different attributes may be referred by viewing series together in one chart. Not only the numerical but also categorical data that tells a machine’s degree may be summed with a user defined rule.

Next, after having the basic clue for detecting anomalies, almost certainly, the administrator will need to drill down further and discover a more precise knowledge such as which machine(s) that actually causes the problem. The challenge at this level will be different: extract the specific machine, and find the machine’s specific attributes that are related to the anomalous event. To address this challenge, we use the Gantt chart with the data broken down from the summary data in two ways. First, for each selected attribute in the event’s time period, we plot every machine as a single visual element and use different colors representing the related attributes at each time slice during the selected investigation period. An intuitive horizontal comparison between machines will highlight every target whose trend of changing values is significantly different from others. For instance, in Figure 3, sever_03’s health status differs from others. Second, we plot each attribute as a single visual element and reflect its value changing trends by color. At this time, the result from horizontal comparison is the relations between attributes. In particular, it means that we visually compare the patterns of changing attributes’ values and group the attributes which have the similar changing trend. A recursive processing may be required to help generate more precise results. In addition, when the administrator has events which last in a long term, zooming the time axis could help the investigator to recognize the trends easily.

3.2 Treemaps and Network Graphs

At the next anomaly analysis level, the administrator already has an anomalous-machine-candidate list and the most representative data attributes for event trends. The next target for the administrator is to find which client(s) mainly cause the event. The difficulty at this level is to distinguish the strong anomalous candidates and less anomalous candidates. To solve the problem, we try to illustrate the true anomaly by using the Treemap to compare the magnitudes of nodes corresponding to the event’s representative attribute. In the Treemap, we define one possible hierarchical structure as follows. We treat every machine in the list as individual and divide each machine’s sum value into blocks during the event period. For each rectangle, the area corresponds to the sum value of the attribute of a specific machine during the unique time unit. The biggest area of a rectangle represents the nodes which mostly affect the event. In addition, we could save the color property to separate different items, e.g., machines in Figure 4. We could then embed another attribute into the color and show the color changes in granularity. The two attributes may help the investigator to gain more knowledge. Treemap rectangles with smaller sizes could also be viewed via a zoom and pan interaction.

After the above process, the administrator has one final task with two challenges: first, how to find correlations among anomalies, and second, how to find the out-list nodes which belong to the dependency path of anomalous events. Some nodes may only cause the event but are not part of the event results. Connection information is stored in network data such as network flows and packet traces. To that end, we build and utilize network graphs by extracting the src/dst pairs within the event time period. Link attributes (e.g. total data transmission per connection) may be built into the graph and shown as edge width. The investigator could directly mark the existing edges between a pairwise in-list nodes as anomalous link. For the links consisting only one node in the anomalous list, we treat them as anomalous links at first. Then we substitute link attributes into the above visual analytics as inputs and create new presentations. Finally, after a comprehensive analysis of all new charts, the administrator will determine whether a link is an anomaly or not.

3.3 Linking and Brushing

The individual views we discussed above focus on creating presentation with node’s attributes which describe a network element’s information, e.g. system status. However, network data records usually correlate to two nodes, and pairwise nodes have common attributes to describe the link. For instance, a network flow record has a source IP and a destination IP, and their relevant attributes such as first seen packet count, total bytes transmit, etc., which are also related to two nodes’ properties in the current conversation. Considering every connection record as an isolated item will make the workload to increase exponentially.

Cross consulting and linking between each individual view is another important procedure during the entire network anomaly analysis task. The idea is to combine the above visualization methods to overcome the shortcomings of a single technique. We create the visualization tool by connecting the four views together and establishing the contact between visual items of each view. With the interaction such as selecting a specific node in one view or from the anomalous list menu, other views will automatically reflect the updated information with respect to the selected item. For example, if the administrator selects a specific node in the network graph chart in the visualization tool, by treating the selected node as the source node, Gantt chart will automatically plot all connections' destination IP nodes as visual elements. Link attributes such as first seen package count and total byte transmit could be shown as granularity changed color. Meanwhile, Treemap uses rectangles' sizes to show which destination IP dominates a particular link attribute. In addition, area chart will plot the rise and fall of link attributes' sum value to highlight the interesting area as reference.

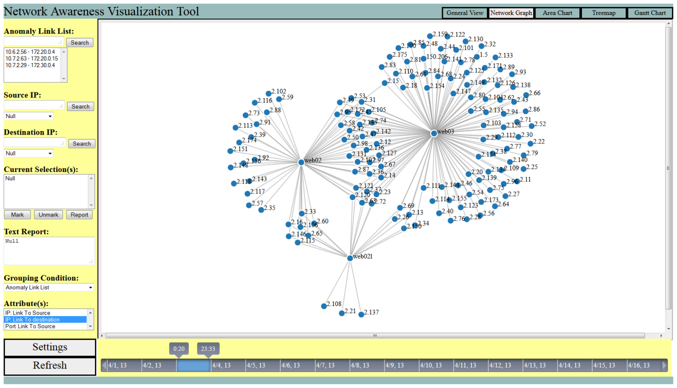
The idea behind this design is that by connecting multiple views through interactive linking and brushing, the visualization tool provides more information than considering each component independently. This is especially useful for network anomaly analysis and investigation. The investigator will consider all evidence and make the conclusion such as which connection and system are anomalous. Furthermore, linking will empower the administrator to aggregate scattered anomaly into a correlated event and a bigger picture. Additional conclusions could also be made, such as which node is the attacker in a malicious intrusion. Attack types could also be determined with several connections. For instance, if one attacker is demonstrated to many other victims, it might be a portscan behavior, and reversely, if one victim node has many anomalous links included, it may have suffered from a DDoS attack.

4 Case Study

We use VAST Challenge 2013 Mini Challenge 3 dataset as an example of the case study. The dataset contains network security data from an international marketing company (Big Marketing) that consists of around 1200 workstations and servers. Besides the common network traffic logs such as network flow data and intrusion detecton/prevention system (IDS/IPS) data, the 2-week dataset also contains a commercial network monitoring program that provides network health and status data for every single workstation and server, which periodically reports status updates such as CPU, memory and disk usage.

Suppose the administrator of Big Marketing receives user trouble reports stating that one of the corporate websites becomes unresponsive from Apr. 2nd to Apr. 5th. The administrator reboots machines in the data center and has all websites and relevant network functionalities on other servers back online. After solving the problem, the administrator realizes that the reason for the website crash is still unknown, which might be important because similar or worse events could happen in the future if current issue is due to hardware-level hidden faults in the data center's machines, or even worse, from malicious attacks.

To investigate, the administrator pulls the entire corporate network logs from monitoring facilities, such as the Netflow collector and big brother program, around the periods of problem, and tries to figure out two questions. First, which part of the network is the main anomaly that causes the unresponsiveness? Second, what is the reason for the anomaly happened? To find out, the administrator uploads the data into the web-based network awareness visualization tool. At first, the administrator almost gets overwhelmed by the too much information about different attributes over such a long time window. With the guidance of the tool, the administrator observes an area chart (Figure 2) for data transmission trend of all servers as destinations in the network. The administrator observes a few spikes. All the rise and fall of counters, first seen total bytes, and first seen packet counts from network flow data are synchronized. The administrator notes all suspicious events by marking their start and end times.



Anomaly Link List:
Search
10.6.2.56 - 172.20.0.4
10.7.2.63 - 172.20.0.15
10.7.2.29 - 172.30.0.4

Source IP:
Search
Null

Destination IP:
Search
Null

Current Selection(s):
Null
Mark Unmark Report

Text Report:
Null

Grouping Condition:
Anomaly Link List

Attribute(s):
IP: Link To Source
IP: Link To destination
Port Link To Source

Settings
Refresh

Figure 5: Examine and correlate events by selecting anomalous nodes in a network graph.

Next, the administrator tries to find out the exact problem machines for each event. He examines the servers' status changes for relative attributes in Gantt chart section. For example, in Figure 3, one can clearly find that only the connection status of server 'web03' is at a warning level sequentially during the event's time period. An inference could be made that the event may relate to actions by server 'web03.' In order to verify his hypothesis, the administrator then digs into the Treemap and views the sum value of every server's hourly flow records. He notes that the two rectangles of server 'web03' are significantly larger than others, which makes them barely to be shown in Figure 4. The administrator concludes that the server 'web03' is an anomalous node and its action time is pinned down to a two-hour time period.

After confirming the most important anomalous node and event happening time, the administrator proceeds to the network graph and discovers the correlation between the server 'web03' and its neighbors (Figure 5). By selecting the particular node, the investigator observes relevant link attributes in the main visualization view (Figure 1). After linking among visualizations and comprehensive investigation, the administrator excludes the candidate cause that 'server03' is offline due to functional failures, but rather server 'web03' is under DoS attack during the event time.

5 Conclusion

Network anomaly detection is important yet hard due to the vast data volume, a large number attributes, interconnectivity/causality, and high dynamics. We study how general visualization methods might be suitable to address different aspects of network management and monitoring tasks, and especially, when these views are linked and combined, what information gains might be particularly valuable for network anomaly analysis tasks. A general, lightweight, web-based visualization prototype has been built and evaluated. We believe that the tool provides a time-efficient alternative for system administrators and network managers to analyze their common log data. Further research will be conducted to study possibly more useful views and their relationship to network anomaly tasks as well as to polish the user interaction process and develop more advanced features.

Acknowledgements

This work was supported in part by CMU Early Career grant (C61920), NSFC grant 61379088 and China National 973 project 2014CB340301.

References

[1] J. Abello, S. G. Kobourov, and R. Yusufov. Visualizing large graphs with compound-fisheye views and treemaps. In *Proceedings of the 12th international conference on Graph Drawing*, pages 431–441, New York, NY, Sep.29-Oct.2 2004.

[2] M. Celenk, T. Conley, J. Willis, and J. Graham. Predictive network anomaly detection and visualization. *IEEE Transactions on Information Forensics and Security*, 5(2):288–299, 2010.

[3] E. Gansner, Y. Hu, and S. North. A maxent-stress model for graph layout. In *IEEE Pacific Visualization Symposium (PacificVis)*, pages 73–80, 2012.

[4] E. Gansner, Y. Hu, S. North, and C. Scheidegger. Multilevel agglomerative edge bundling for visualizing large graphs. In *IEEE Pacific Visualization Symposium (PacificVis)*, pages 187–194, 2011.

[5] E. R. Gansner, Y. Koren, and S. C. North. Topological fisheye views for visualizing large graphs. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):457–468, 2005.

[6] M. Hemmje, C. Kunkel, and A. Willett. Lyberworld – a visualization user interface supporting fulltext retrieval. In *the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 3-6 1994.

[7] D. Holten, P. Isenberg, J. van Wijk, and J. Fekete. An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs. In *IEEE Pacific Visualization Symposium (PacificVis)*, pages 195–202, 2011.

[8] H. Jia, J. Fuh, A. Nee, and Y. Zhang. Integration of genetic algorithm and gantt chart for job shop scheduling in distributed manufacturing systems. *Computers & Industrial Engineering*, 53(2):313 – 320, 2007.

[9] D. Plonka. Flowscan: A network traffic flow reporting and visualization tool. In *In USENIX LISA*, pages 305–317, 2000.

[10] J. Rekimoto and M. Green. The information cube: Using transparency in 3d information visualization. In *Proceedings of the Third Annual Workshop on Information Technologies & Systems*, pages 125–132, 1993.

[11] L. Shi, C. Wang, and Z. Wen. Dynamic network visualization in 1.5d. In *IEEE Pacific Visualization Symposium (PacificVis)*, pages 179–186, 2011.

[12] M. Waldner and D. Schmalstieg. Collaborative information linking: Bridging knowledge gaps between users by linking across applications. In *IEEE Pacific Visualization Symposium (PacificVis)*, pages 115–122, 2011.

[13] J. M. Wilson. Gantt charts: A centenary appreciation. *European Journal of Operational Research*, 149(2):430 – 437, 2003.

[14] X. Yin, W. Yurcik, M. Treaster, Y. Li, and K. Lakkaraju. Visflowconnect: Netflow visualizations of link relationships for security situational awareness. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC '04)*, pages 26–34, Washington DC, USA, 2004.