



Association Visualization Analysis for the Application Service Layer and Network Control Layer

Mianmian Shi^{1(✉)} and Huaping Cao²

¹ Institute of Network Technology, Beijing University of Posts
and Telecommunications, Beijing 100876, China
shimian@bupt.edu.cn

² National Computer Network Emergency Response Technical Team,
Coordination Center of China, Beijing 100029, China
caohuaping@cert.org.cn

Abstract. Most researches about complex networks are single-layer networks-based representation. However, in most cases, systems in the real world are not isolated but connective. In this paper, different from the traditional Open System Interconnection (OSI) model, our research pays attention to application service layer and network control layer for the view of application. Two layers connect with each other by using IP mapping relationship. Firstly, to avoid unnecessary loss of computational efficiency, we modify Louvain algorithm to divide the nodes in network control layer into several parts. Secondly, we add additional community attractive force and introduce Barnes-Hut force-calculation model to Fruchterman-Reingold algorithm in order to make nodes in network control layer aligned more structured and well-distributed efficiently. Finally, we merge the application service layer and the network control layer into a two-layer visualization model. Based on our two-layer model, the whole network trend, topology and incidence relation can be conveniently grasped.

Keywords: Two-layer layout · Network visualization · FR algorithm

1 Introduction

Visual analytics is based on the combination of automatic analysis technology and interactive visualization technology for efficacious exploring and decision making for users [1]. Converting complex datasets into graphics or images makes it easier for users to understand large-scale data and discover hidden information.

In many researches, network was just considered as a complex graph or a kind of data structure with nodes and edges. These studies are generally divided into two categories. One is visualization research for a single layer. The other is visualization of multi layers for complex network.

There are some works on displaying network status in single layer. For network control layer, VizFlowConnect [2] and PortVis [3] use Parallel—Coordinate and Scatter—Plot to visualize Netflow data. IPMatrix [4] and Netvis [5] use Node-Link and

TreeMap to visualize IDS alarm data. Romero-Gomez [6] designs THACO (THreat Analysis COnsole) for DNS-based network threat analysis. In addition, researchers also use topology diagram of network routing information to analyze network events. BGP data is regularly used in some systems, such as BGPlay [7], BGP Eye [8], BGPfuse [9] and so on. From the view of application, Alsaleh [10] presents an extension to PHPIDS, which correlates PHPIDS logs with the corresponding web server logs to plot the security-related events. ThousandEyes [11] monitors internal and external network performance to improve application delivery and reduce service interruptions. Similarly, RIPE Atlas [12] is a global network of probes that measure Internet connectivity and reachability, providing an unprecedented understanding of the Internet in real time.

Some researchers also propose multi-layer visual model to show the complex network. Zhang [13] considers that all the subsystems in the real world are not isolates but connective. Iyad Katib [14] separates the logical layer from physical layer, and propose a three-layer named IP-OTN-DWDM to improve the convenience of resource management and fault detection in communication. In 2017, Wei [15] develops a research from the view of application and proposes two-layer network with carrier layer and business layer. In order to display the different levels of network topology and the relationships between the multi layers, researchers usually set each network's nodes and edges in a 2D plane, and put these 2D planes in 3D space. This method is called as two-and-a-half-dimensional (2.5D) visualization [16].

However, in most cases, it is hard for single-layer visualization to show the relationship between different layers in network. Simultaneously, traditional multi-layer social or biological network visualization always focuses on displaying the same type of data hierarchically based on community detection. Each communities are placed on different 2d planes and straight lines connect those planes. These visualization models always ignore the connection relation among different types of layers in real network.

To solve problem mentioned above, we use the 2.5D visualization method to design our two-layer visual model. Concretely, different with the traditional Open System Interconnection (OSI) model, we just consider the application service layer and network control layer from the view of application. These two layers connect with each other by using IP mapping relationship. To avoid unnecessary loss of computational efficiency, we modify Louvain algorithm via pruning the leaf nodes to divide the network control layer into several parts. In order to make the view structured and well-distributed, we add additional community attractive forces to Fruchterman-Reingold algorithm. Then, to minimize the number of the crossing lines between these two layers, the method is given that the nodes' locations of the application service layer can be obtained by using the locations of nodes in the network control layer. Finally, we merge the application service layer and the network control layer into a two-layer visualization model. Based on our two-layer model, the whole network trend, topology and incidence relation can be easily observed.

2 Preliminary Knowledge

In network control layer, we can abstract the Internet topology at the inter-domain level into an Autonomous System (AS) connection graph. Through this graph, network is just considered as a complex graph with AS nodes and BGP routing sessions' links. Similar to social networks, there are communities in the real network. The AS nodes in the community interact closely and the relationships between communities are relatively sparse. In order to present a good community structure layout, this section introduces Louvain algorithm for community detection and Fruchterman-Reingold algorithm for nodes layout covered in this paper.

2.1 Louvain Algorithm

The existing community detection algorithms for complex network are mainly divided into two categories. One is based on the graph theory, such as k-clique algorithm [17], Label propagation algorithm (LPA) [18] and so on. Another is hierarchy-clustering algorithm, such as Fast Newman algorithm (FN) [19], Louvain algorithm [20] and so on. Currently, researchers consider the Louvain algorithm the best non-overlapping community detection algorithm.

Girvan Newman [21] first proposed the concept of modularity Q in 2002. Then the modularity Q in formula (1) is commonly used to measure the strength of the network community structure. As the value of the modularity Q increases, the community structure is more robust and compact. The value of modularity Q is up to one.

$$\begin{aligned}
 Q &= \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{K_i K_j}{2m} \right] \partial(c_i, c_j) \\
 &= \frac{1}{2m} \left[\sum_{i,j} A_{ij} - \frac{\sum_i K_i \sum_j K_j}{2m} \right] \partial(c_i, c_j) \\
 &= \frac{1}{2m} \sum_c \left[\sum in - \frac{(\sum tot)^2}{2m} \right]
 \end{aligned} \tag{1}$$

In which, $\sum in$ is the sum of the weights of all the edges in community c . $\sum tot$ is the sum of the weights of the edges connected to all nodes in community c .

Louvain algorithm is based on modularity Q optimization. The change of the modularity increment ΔQ can be derived by formula (1)

$$\Delta Q = \frac{1}{2m} (k_{i,in} - \frac{\sum tot k_i}{m}) \tag{2}$$

Where, $k_{i,in}$ is the sum of weight of edges which is connected to node n_i in community c . The use of greedy algorithm for large complex networks greatly improves computational efficiency.

Louvain algorithm's process is mainly divided into two steps:

The first step is to regard all nodes in the network as an independent community. Then we try to assign each node to another community which its neighbor node belongs to and calculate the change in modularity increment ΔQ . If $\Delta Q > 0$, we choose

the community which makes ΔQ the largest and then put this node into this community.

In the second step, we regard the nodes of the same community as a new node. The weights of the edges between communities are converted into the weight of edges between new nodes. Repeat step 1 until the modularity Q no longer changes.

2.2 Fruchterman-Reingold Algorithms

The most common graph drawing methods always rely in physical simulations, such as force-directed algorithm (FDA) [22], Kamada-Kawai algorithm (KK) [23], Fruchterman-Reingold algorithm (FR) [24] and so on. In this system, the edges between the nodes are equivalent to the spring or other physical connections, and the nodes are balanced by the interaction of the elastic force. Our research is based on FR algorithm. This method treats nodes as atoms in physical system and there is attractive force and repulsions force between each node. By calculating the total energy of the system, it can produce a beautiful and balance layout with a simple cooling table.

In order to make the nodes in graph well-distributed, FR algorithm thinks that nodes with edges connected should be as close as possible and nodes with no edges connected should be as far as possible. This method defines the concepts of attractive force (f_a) and repulsions force (f_r). There are attractive force among all nodes with edges connected and repulsions force among all nodes. K is used to control side length. The force can be calculated as following:

$$\begin{cases} f_a = d^2/k \\ f_r = -k^2/d \end{cases} \quad (3)$$

In which, $k = C\sqrt{\frac{S}{N}}$ is the balance coefficient. C is a constant. S is the layout area and N is the number of all nodes. Using FR algorithm, nodes are evenly distributed.

3 The Method of Two-Layer Network Topology Visualization

In our research, we just abstract the network and application as network control layer and application service layer. We focus on the topological structure of the network control layer as well as the relationships between each layer, so we use the 2.5D visualization method to design our two-layer model. This section introduces our work on network control layer topological visualization and the display algorithm of two-layer network model.

3.1 Visualization of Network Control Layer

Similar to social networks, there are communities in the real network. The AS nodes in the community interact closely and the relationships between communities are less interact. Traditional Louvain algorithm and FR algorithm have some defects in display

and efficiency. In order to make the view structured and well-distributed, we modify Louvain algorithm via pruning the leaf nodes and add additional community attractive forces to our modified FR algorithm.

The Modified Louvain Algorithm

Since there are a large number of leaf nodes which are only connected to one node in real network, it would cause unnecessary loss of computational efficiency if we calculate modularity increment ΔQ for every node. As shown in Fig. 1, the black nodes have five leaf nodes. Because community detection will avoid individual nodes belonging to a community, the leaf nodes and the black node must be in the same community. To avoid unnecessary loss of computational efficiency, in the first step of the Louvain algorithm, we can assign the leaf nodes to its adjacent non-leaf nodes directly. With the increase of the ratio of leaf nodes in the network, the algorithm efficiency is obviously enhanced.

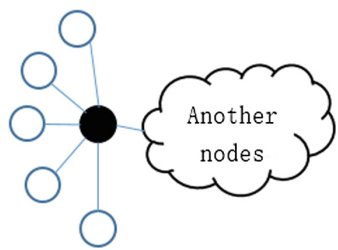


Fig. 1. Network with leaf nodes.

Our research obtains the routing data of the rrc00.ripe.net probe at 16:00 on May 7, 2018 from the RIPE Routing Information Service (RIS). We get the relationships between global AS nodes by Python processing. Since there are edges with higher repetition rate and edges with fewer paths in our routing data, we introduce the number of AS path as edge weights into the Louvain algorithm. Compared to the unweighted graph in real network, the nodes with high weight links are more likely in the same community.

We classify the nodes and edges by countries and select 338 nodes and 544 edges in China collection for experiment. We perform Louvain algorithm and modified Louvain one on this dataset. Compared with the Fast Newman algorithm (FN) and Label propagation algorithm (LPA), Table 1 shows the results of the experiment.

Table 1. Quantitative evaluation of FN algorithm, LPA algorithm, Louvain algorithm and modified Louvain one.

Algorithm	Time(s)	Number of community	Modularity Q
FN	0.9721	18	0.5620
LPA	0.0790	20	0.5469
Louvain	0.0870	15	0.6170
modified Louvain	0.0834	12	0.6231

As shown in Table 1, the Fast Newman algorithm performs worst. The Modularity Q is not bad but it takes too much time in iteration. The LPA algorithm is the fastest algorithm in those algorithms, but the Louvain algorithm performs much better in the modularity Q. Compared with the original Louvain algorithm, the modified one is better with 4.12% less time-consuming. At the same time, as the number of communities drops from 15 to 12, the modularity Q has also increased from 0.6137 to 0.6231. Therefore, this experiment proves the modified Louvain algorithm is the most suitable community detection for visualization of network control layer.

The Modified FR Algorithm

There are some issues on using FR algorithm to visualize large-scale data layout: (1) It is difficult for us to observe the community structure and the connection between communities when there are too many communities. (2) The time complexity of FR is $O(|E| + |V|^2)$. When using too many nodes, it will take a lot of time to calculate.

So, in our research, refer to FR algorithm, we redefine attractive forces (f_a) and repulsions forces (f_r), and add community force (f_{com}) to make the nodes in the same community more closely to each other while ensuring an evenly layout. In order to avoid the local optimum, we introduce the energy function into our method and use simulated annealing algorithm to approximate optimal solution. At the same time, Barnes-Hut force-calculation model [25] is introduced to reduce time complexity so that the modified algorithm can be applied to large-scale network layout.

Often, the nodes with high weight links should stay closer than other nodes, so we introduce the weight of each edges into the calculation of attractive forces (f_a). For another, the high-degree nodes usually belongs to different communities, so we hope these nodes farther away from each other in order to display a better visualization of communities. Therefore, we also introduce the degrees of two nodes when we calculate the repulsive forces (f_r) between all nodes. The forces of every node in network are calculated as following:

$$\begin{cases} f_a = \frac{d^2(n_1, n_2) * w(n_1, n_2)}{k} \\ f_r = -\frac{k^2}{d} \deg(n_1) * \deg(n_2) \end{cases} \quad (4)$$

For the nodes in the same community, we hope that edges with high weight connect nodes with each other tightly. Refer to the attractive force formula (3) defined in FR, we introduce the sum of edges weight in community (w_{com}) and define the formula for community forces as following:

$$f_{com} = \frac{d^2(n_1, n_2)}{k} * w(n_1, n_2) * \frac{w_{com}}{w_{all}} \quad (5)$$

In which $d(n_1, n_2)$ is the distance between node 1 and node 2, $w(n_1, n_2)$ indicates the weight of edge (n_1, n_2) . $\deg(n_1)$ is the degree of node n_1 . w_{com} is the sum of edge weight in a community. w_{all} is the sum of weight of all edges.

At the same time, in order to reduce the time complexity of the repulsion forces calculation, we introduce Barnes-Hut force-calculation model in the calculation of

repulsive forces. In our modified FR algorithm, the time complexity of the repulsion calculation is reduced from $O(|V|^2)$ to $O(|V| \log |V|)$.

We use this modified FR algorithm and another classic layout algorithms for experiment in the public Dolphins data set (62 nodes, 159 sides) and Football Club data set (115 nodes, 613 sides). The results are as shown in Figs. 2 and 3.

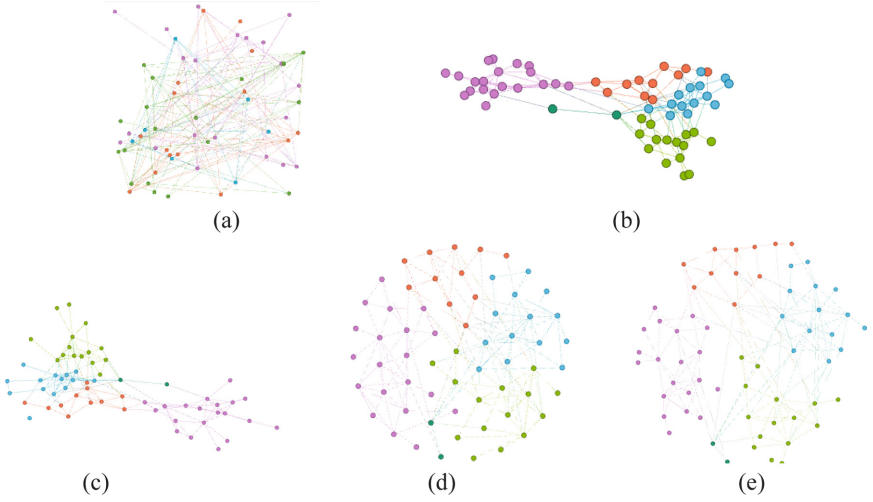


Fig. 2. Visualization of Dolphins Dataset (a. raw data without algorithm, b. Yifan Hu algorithm [26], c. ForceAtlas 2 algorithm [27], d. FR algorithm, e. modified FR algorithm).

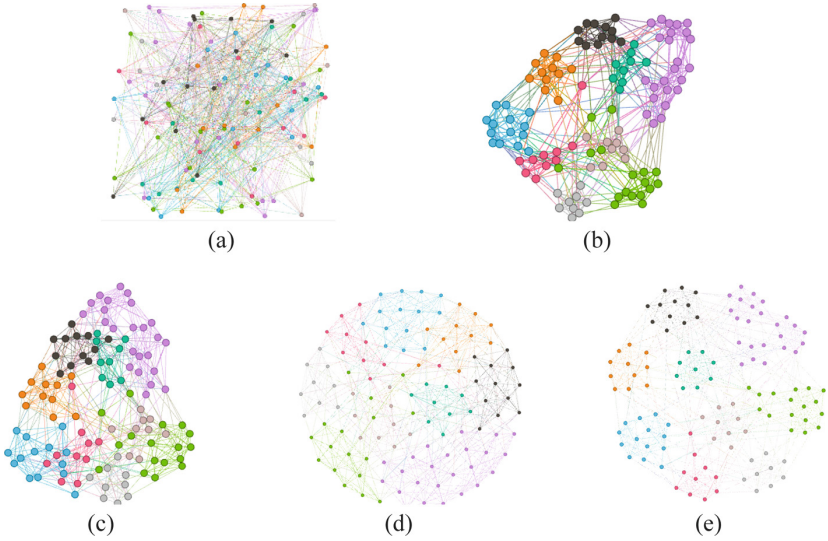


Fig. 3. Visualization of Football Club Dataset (a. raw data without algorithm, b. Yifan Hu algorithm, c. ForceAtlas 2 algorithm, d. FR algorithm, e. modified FR algorithm).

From the visualization of the two datasets in Figs. 2 and 3, it is easy to see that the modified FR algorithm shows the community relationships in network more clearly and structured than Yifan Hu and ForceAtlas2 algorithm. We can easily distinguish community connections with greater connectivity in the current network state.

3.2 Establishment of Two-Layer Network Model

In our research, the real network is considered as a two-layer model of the application service layer and the network control layer. We find that the locations of the application service layer's nodes are related to the locations of the network control layer nodes closely. In order to reduce the visual confusion caused by the inter-layer crossing, we need to calculate and adjust the location of the application service layer's nodes. Generally, an application service corresponds to multiple nodes in the network control layer. So in our 2.5D model, it is assumed that the node coordinates of the network control layer are $x_i, y_i, z_i = 0$. Then the corresponding node coordinates in application service layer are as follows:

$$\begin{cases} x_j = (x_i + x_{i+1} + \dots + x_{i+n-1})/n \\ y_j = (y_i + y_{i+1} + \dots + y_{i+n-1})/n \\ z_j = 1 \end{cases} \quad (6)$$

In which, x_i, y_i, z_i represent node coordinates in the network control layer and x_i, y_i, z_i represent node coordinates in the application service layer. n is the number of nodes in network control layer corresponding to an application service.

The layout algorithm established by two-layer layout model can be described as following:

Input: Network $G(V, E)$: a set of vertices V and edges E

Nodes: $\{V_{j1}, V_{j2}, V_{j3} \dots\}$

Output: Location of every nodes : $\{P_{i1}, P_{i2}, P_{i3} \dots P_{j1}, P_{j2}, P_{j3} \dots\}$

Step1: Merging all leaf nodes with their adjacent non-leaf nodes in network G

Step2: Covert the network G into an undirected weight graph and execute the modified Louvain algorithm. The result is community partition C .

Step3: Randomly input nodes in network G and initialize all node coordinates $P_{i1}, P_{i2}, P_{i3}, \dots$

Step4: Initialize the temperature coefficient and build Barnes-Hut hierarchical space model according to the position of Step3

Step5: Calculate the attractive force between adjacent nodes and the repulsive force between all nodes according to formula (4) and calculate another attractive force between adjacent nodes in the same community according to formula (5), then update the node position.

Step6: When the temperature coefficient is gradually reduced to zero and the node position tends to be stable, output the node coordinates $P_{i1}, P_{i2}, P_{i3} \dots$

Step7: According to the positions of nodes in network G , calculate node coordinates in application service layer by formula (6). Take the z value of network control layer as 0 and the application service layer as 1, output the application server node position $P_{j1}, P_{j2}, P_{j3} \dots$

4 Visual Analysis

We select 338 nodes and 544 edges in China collection in network control layer and 13 website nodes in application service layer for experiment. Using IP mapping to connect two layers, the result is shown in Fig. 4. The black nodes represent 13 websites and the remaining colorful nodes are the AS nodes in network control layer. Different colors represent different communities.

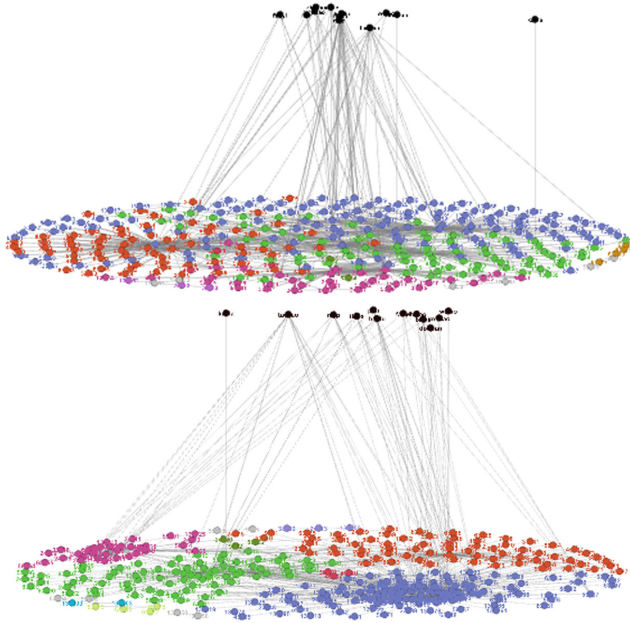


Fig. 4. Two-layer network topology for fault detection (FR algorithm and modified FR algorithm with community forces)

As shown in Fig. 4, compared the two-layer network model built by FR algorithm with two-layer network model built by modified algorithm, we can find that the community structure in our modified algorithm is more obvious and structured.

If you are interested in a website, you can select a node in the application service layer. Then, all business relationship related to it in network control layer are highlighted and we can get the AS node information from node labels. The result is shown in Fig. 5.

As shown in Fig. 5, we take the Baidu Fanyi node as an example, the AS nodes connect with it are AS4808, AS4847 AS9808, AS23724 and AS55967. Therefore, when these nodes are safe, we can ensure the proper operation of Baidu Fanyi.

Fig. 5. Select a node in application service layer.

If you want to observe the relationships of an AS node or the business carried by an AS node, you can select an AS node in the network control layer. The result is shown in Fig. 6

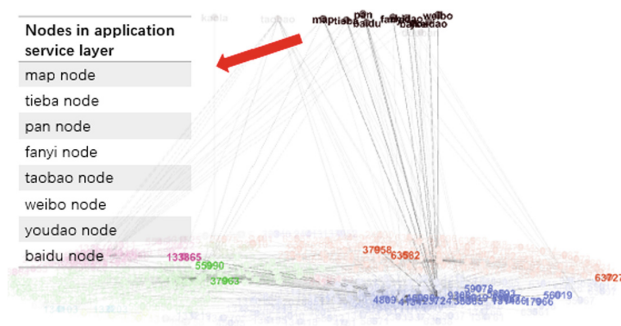


Fig. 6. Select an AS node (AS23724) in network control layer.

Figure 6 shows the result when we select an AS node (take AS23724 as an example) in network control layer. We can easily distinguish all nodes connected to AS23724 in network control layer and application service layer. If there is a problem with the AS23724 node, the nodes connected to it in network control layer and the website relied on it in application service layer may be also in trouble.

5 Conclusion

Currently, the visualization of real network mainly focused on the single-layer networks-based representation. Our research considered the application service layer and the network control layer with the IP mapping relationship between them. In network control layer, to avoid unnecessary loss of computational efficiency, we modified Louvain algorithm via pruning the leaf nodes to community detection. In order to make the view structured and well-distributed, we added additional community

attractive forces to FR algorithm to make nodes in network control layer structured and well-distributed. Finally, we merged application service layer and the network control layer into a 2.5D visual model to facilitate the user's further analysis on observing the network trend, topology and incidence relation.

The future task of our multi-layer networks is to introduce a geographic location layer to show the IP location of each node in application service layer. From the future multi-layer visual model, the operating status of the network and topology can be observed in multiple dimensions.

Acknowledgement. This paper is supported by National Natural Science Foundation of China and Xinjiang under Project U1603261 and National Key R&D Program of China (2016YFE0133000): EU-China study on IoT and 5G(EXCITING) and Research Fund of Ministry of Education- China Mobile under Grant No. MCM20160304.

References

1. Kohlhammer, J., Keim, D., Pohl, M., et al.: Solving problems with visual analytics. *Procedia Comput. Sci.* **7**(29), 117–120 (2011)
2. Yin, X., Yurcik, W., Treaster, M., et al.: 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*, pp. 26–34 (2004)
3. McPherson, J., Ma, K.L., Krystosk, P., et al.: Portvis: a tool for port-based detection of security events. In: *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, pp. 73–81 (2004)
4. Abdullah, K., Lee, C., Conti, G., et al.: Ids rainstorm: visualizing ids alarms (2005)
5. Kan, Z., Hu, C., Wang, Z., et al.: NetVis: a network security management visualization tool based on treemap. In: *2010 2nd International Conference on Advanced Computer Control (ICACC)*, pp. 18–21. IEEE (2010)
6. Romero-Gomez, R., Nadji, Y., Antonakakis, M.: Towards designing effective visualizations for DNS-based network threat analysis. In: *2017 IEEE Symposium on Visualization for Cyber Security (VizSec)*, Phoenix, AZ, pp. 1–8 (2017)
7. Colitti, L., Di Battista, G., Mariani, F., et al.: Visualizing interdomain routing with BGPlay. *J. Graph Algorithms Appl.* **9**(1), 117–148 (2005)
8. Teoh, S.T., Ranjan, S., Nucci, A., et al.: BGP eye: a new visualization tool for real-time detection and analysis of BGP anomalies. In: *Proceedings of the 3rd International Workshop on Visualization for Computer Security*, pp. 81–90. ACM (2006)
9. Papadopoulos, S., Theodoridis, G., Tzovaras, D.: Bgpfuse: Using visual feature fusion for the detection and attribution of BGP anomalies. In: *Proceedings of the Tenth Workshop on Visualization for Cyber Security*, pp. 57–64. ACM (2013)
10. Alsaleh, M., Alqahtani, A., Alarifi, A., et al.: Visualizing PHPIDS log files for better understanding of web server attacks. In: *Proceedings of the Tenth Workshop on Visualization for Cyber Security*, pp. 1–8. ACM (2013)
11. ThousandEyes. <https://www.thousandeyes.com/>, last accessed 2018/7/21
12. RIPE NCC. <https://atlas.ripe.net/landing/measurements-and-tools/>. Accessed 21 July 2018
13. Xitao, Z., Lingda, W., Huaquan, H., Shaobo, Y.: Tasks for visual analytics in multilayer networks. In: *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*, Shenzhen, pp. 368–371 (2017)

14. Katib, I., Medhi, D.: A study on layer correlation effects through a multilayer network optimization problem. In: Proceedings of the 23rd International Teletraffic Congress, pp. 31–38 (2011)
15. Wei, Y., Du, X.: Two-layer network visualization for comprehensive analysis. In: 2017 IEEE Second International Conference on Data Science in Cyberspace (DSC), pp. 363–367. IEEE (2017)
16. Dwyer, T.: Two-and-a-half-dimensional Visualisation of Relational Networks. School of Information Technologies, Faculty of Science, University of Sydney (2004)
17. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**, 814–818 (2005)
18. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **76**(3), 036106 (2007)
19. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **69**(6), 066133 (2004)
20. Blondel, V.D., Guillaume, J.L., Lambiotte, R., et al.: Fast unfolding of communities in large networks. *J. Stat. Mech: Theory Exp.* **10**, 10008 (2008)
21. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* **99**(12), 7821–7826 (2002)
22. Eades, P.: A heuristics for graph drawing. *Congressus Numerantium* **42**, 149–160 (1984)
23. Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. *Inf. Process. Lett.* **31**(1), 7–15 (1989)
24. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force directed placement. *Softw. Pract. Experience* **21**(11), 1129–1164 (1991)
25. Barnes, J., Hut, P.: A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature* **324**(6096), 446–449 (1986)
26. Hu, Y.: Efficient, high-quality force-directed graph drawing. *Math. J.* **10**(1), 37–71 (2005)
27. Jacomy, M., Venturini, T., Heymann, S., et al.: ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PLoS ONE* **9**(6), e98679 (2014)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

