Title

Netostat: analyzing dynamic flow patterns in high-speed networks

Permalink

https://escholarship.org/uc/item/8nz5w51s

Journal

Cluster Computing, 25(4)

ISSN

1386-7857

Authors

Murugesan, Sugeerth
Kiran, Mariam
Hamann, Bernd
et al.

Publication Date

2022-08-01

DOI

10.1007/s10586-022-03543-0

Peer reviewed

Netostat: Analyzing Dynamic Flow Patterns in High-Speed Networks

Sugeerth Murugesan · Mariam Kiran · Bernd Hamann · Gunther H. Weber

Received: 15 May 2021 / Revised: 16 October 2021 / Accepted: 9 December 2021

Abstract Understanding flow traffic patterns in networks, such as the Internet or service provider networks, is crucial to improving their design and building them robustly. However, as networks grow and become more complex, it is increasingly cumbersome and challenging to study how the many flow patterns, sizes and the continually changing source-destination pairs in the network evolve with time.

We present Netostat, a visualization-based network analysis tool that uses visual representation and a mathematics framework to study and capture flow patterns, using graph theoretical methods such as clustering, similarity and difference measures. Netostat generates an interactive graph of all traffic patterns in the network, to isolate key elements that can provide insights for traffic engineering. We present results for U.S. and European research networks, ESnet and GEANT, demonstrating network state changes, to identify major flow trends, potential points of failure, and bottlenecks.

Keywords Graph analysis, local clustering algorithm, difference graphs, wide area networks, network design.

Sugeerth Murugesan
Department of Computer Science
University of California
Davis, CA 95616
U.S.A.
E-mail: smuru@ucdavis.edu

Mariam Kiran
Berkeley Lab
1 Cyclotron Rd
Berkeley, California
U.S.A.
E-mail: mkiran@lbl.gov

Bernd Hamann
Department of Computer Science
University of California
Davis, CA 95616
U.S.A.
E-mail: bhamann@ucdavis.edu

Gunther. H. Weber
Berkeley Lab
1 Cyclotron Rd
Berkeley, California
U.S.A.
E-mail: ghweber@lbl.gov

1 Introduction

Computer networks are engineered to cope with challenges of traffic overhead, load-balancing, or prevent many potential points of failure [1]. However, network behavior is difficult to diagnose or comprehend, especially during pivotal time points, e.g., when unexpected traffic flows arise or an anomalous event or a burst of traffic occurs through the network. By modeling network behavior as a graph theory problem, one can characterize the flow data in great detail and understand which nodes connect frequently, how the addition or deletion of links affect network performance, or how this information can help with improving or building better networks. Studying network traffic flow patterns can provide insights relevant for better configuration and optimization of networks.

Using topological structure and historical flow data can reveal past network congestion points that have been resolved by updating routing table configurations [2], [3]. However, as networks grow and become increasingly complex, it is very cumbersome to study their behavioral patterns and make suggestions. Various techniques that are commonly used in social network analysis, e.g., centrality measures, connection degree, or community formations, can help determine how flow patterns change over time in a network. Such patterns provide us with a holistic network view, enabling comprehensive characterization of regular vs. non-regular or weekend vs. weekday patterns. For example, certain users coming online at particular days of times and an experimental detector running only some times in the year can cause consistent network flow traffic. When exploring a wide area network (WAN) setting, these techniques can reveal more intricate insights on source-destination movements that can help improve network design and engineering.

Current approaches used for network performance analysis fail to identify network states as a collection of time-points [4]. Further, solutions for visualizing dynamic graph changes are limited, due to change blindness. i.e., the difficulty to notice significant changes when similar images are placed adjacently[5]; non-compliance of mental-map preservation; and a lack of temporal visual scalability [6],[7]. Current techniques are not sufficient for depicting flow changes in network behavior patterns, e.g., new flows, new sources of data, newly formed connections, or effects on network bandwidth.

Our tool, Netostat, adapts network flow analysis techniques for WAN networks based on flow graphs, with nodes representing sites and edges indicating active flow transfers. Our approach is based on community-detection, similarity, and difference algorithms, making it possible to detect flow patterns in the network or identify flow pattern changes when a network state changes. Compared to the Internet, research and education (R&E) WAN networks are characterized by more unsystematic and erratic traffic patterns behavior as proven many times [8] [9] because of the high variability in files and users. Unlike the Internet exhibiting periodic patterns [10], research networks depend on the kinds of science experiments that are performed and what devices are running, or which groups are involved and what type of data transfers happen, varying from small to very large transfers requiring minutes or hours [11].

Our approach focuses on analyzing dynamic patterns, with state detection mechanism using difference graph techniques, to determine major changes in time-varying network flow data and identify topological flow changes. We visualize this behavior by encoding differences between current and adjacent time steps, by computing a difference graph and mapping states to find the dominant day and night patterns. Our analysis uses packet information routed via UDP, TCP, and ICMP network flows, captured by routers at network gateways. The data contains the source IP address, destination IP address, file size, port numbers, time sent, and relevant flags. The flows are time-stamped, sometimes with flow duration and transfer size. In this paper, our contributions support WAN flow analysis using difference and similarity graphs. Our analysis is based on dynamic graphs, allowing us to identify important information about site connections, daily patterns, and network growth as a consequence of sites starting (or shutting down). To the best of our knowledge, no such tools for WAN research network analysis exist.

1. We present a difference analysis framework based on social network analysis principles to identify growth and decay of flow data across networks and recognize potential points of failure ahead of time.
2. We develop a network visual analysis tool, Netostat, that processes time-varying network flow information to efficiently identify recurring day/night patterns, and detect load imbalance in the network flow infrastructure.
3. We apply our techniques to real WAN data sets – the U.S. and European research networks – demonstrating our method's capability to highlight flow characteristics and time-varying behavior that is hard to comprehend using existing network analysis techniques.

2 Background and Motivation

In this section, we present key issues of network change patterns and demonstrate motivating examples of developing techniques from social network analysis.

2.1 Network Analysis and Visualizations

Network monitoring tools can help model flow patterns, such as using parallel coordinates [12] and network maps [13] to understand overall network loads and topology [14]. Additionally, visualizing dynamic network patterns has gained much attention in both industry and research worldwide [15–17].

Network analysis methods have evolved to become very sophisticated supporting easy investigation for scientific and managerial purposes. For example, Erbacher [18] *et al.* and Ball *et al.* [19] employed a detailed approach to analyzing connectivity patterns from the intranet level to individual machines. Further, Goodall *et al.* [20] and Lakkaraju *et al.* [21] utilized aggregation and filtering mechanisms to reduce clutter and help users focus on regions of interest. Other techniques aided scalable exploration of data that involve sliders, dynamic queries [22], brushing, and linking[23].

The aforementioned techniques can be categorized into methods that utilize two or three-dimensional space. Examples utilizing three-dimensional techniques mostly require sophisticated interaction techniques such as zooming, filtering, rotating, and more [13][24][25]. Such methods increase the interaction load, cause occlusion, and clutter. In contrast, two-dimensional methods such as PortVis [26] provide an occlusion-free method to identify major events in dynamic networks.

Other techniques like seeNet [27] use abstraction techniques to identify and characterize major events in the network flow data and the tool by Teoh *et al.* [28] focuses on merging and utilizing multiple visualization views to explore complementary aspects of the data. Visual methods in other domains such as brain networks employ linked visualization views [29, 30] and flow-based techniques [31, 32] to better understand brain activity.

All of the mentioned systems do not satisfactorily focus on temporal aspects of network flows and fail to create situational awareness of network states. Netostat aims to automatically assess the topological effect of flow changes to better mitigate critical network bottlenecks. Further, graph-theoretical methods are used to model community changes to summarize all information flow details [13].

2.2 Social Network Analysis with Difference Graphs

Traditional dynamic graph visual analysis approaches suffer from change-blindness, (a phenomenon that occurs when we cannot recognize minute changes across two similar images [33]); it is often the consequence of overdrawing visual elements, therefore not conveying topological change effectively. Social network techniques such as difference graph methods solve this problem by only depicting the change between two-time steps. Given two graph states, only changes (concerning edges and nodes) are visualized [34]. The difference graph provides new insights into analyzing flow changes.

To deal with problems of scalability in difference graphs, Archambault *et al.* [35] used hierarchies to depict large areas where the entire graph changes, just providing general overview patterns. Subsequent work by Bourqui and Jourdan [36] analyzed edges having similar pathways to focus on structural similarity. Further work by Rufiange and McGuffin [37] used a hybrid method to build small-multiples and animations, to determine local topological changes between graphs.

Difference graphs alone, however, do not provide reasons for graph changes between time steps. This missing information can help fuel alerts and potentially network threats. This limitation of the lack of contextual information can be crucial for interpreting traffic patterns and low-level topological change over time. In our work, we go beyond traditional visual analytic methods by studying the context changes between two given time steps to best identify the change in centrality, community, and difference graphs.

We develop novel methods to help provide a difference-centrality metric [38] to define important changes as dynamic points, along with similarity for real WAN network data sets.

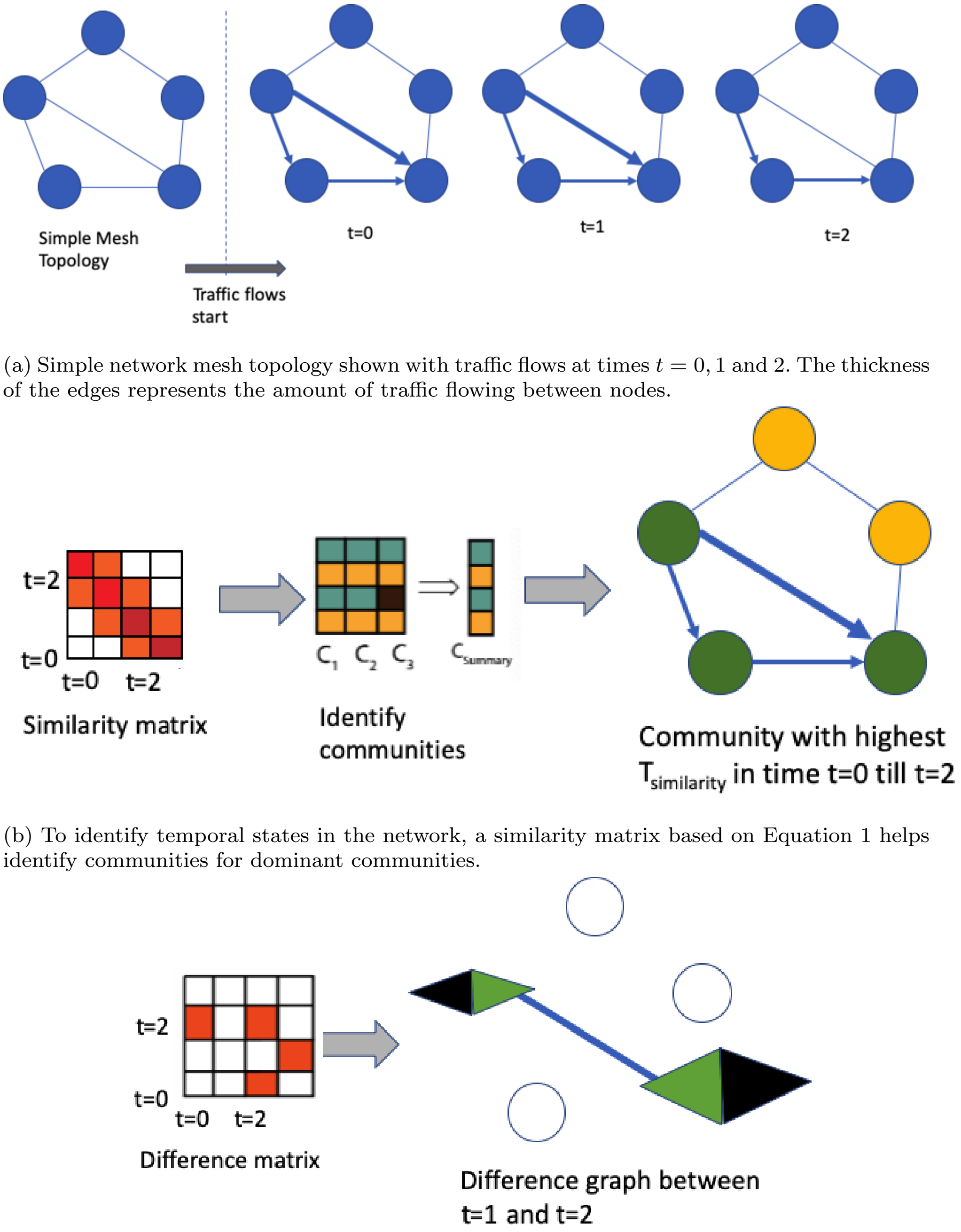
2.3 Understanding Network Flow Behavior

Software-defined networking (SDN) aims to provide flexible solutions to build agile networks, using active monitoring and informed decision-making [39]. Google [40] used SDNs to optimize link usage by doing ‘what-if’ scenarios to schedule transfers. Google’s B4 [40] and Microsoft’s SWAN (Software Driven WAN) [41] have proposed manners in which routers can greedily select routing patterns for arriving flows globally, to increase path utilization. However, these techniques require meticulously designed heuristics to calculate optimal routes and also do not distinguish between arriving flow characteristics. Studying network measurements can simultaneously detect, identify, and visualize attacks for anomalous traffic in real-time by passively monitoring packet headers [4]. However, reliably diagnosing flow-level behavioral patterns and how these can be linked to failures, improve routing paths, and develop better routing algorithms is still largely unexplored.

Understanding complex network behavior as a function of time in dynamic graphs can have an impact on network design and decision-making. We leverage social cluster analysis techniques for network flow analysis. The specific goals targeted by our approach are:

1. **Flow pattern recognition in large wide-area networks:** Concerning time, transfer behavior can reveal how much data is being transferred across sites and how long connections last. This insight provides a better understanding of network topology behavior.

2. **Linking time changes with flow patterns:** Understanding overall network behavior through flow changes between sites, over time. This is achieved by visualizing topological differences between graphs.



(a) Simple network mesh topology shown with traffic flows at times $t = 0, 1$ and 2. The thickness of the edges represents the amount of traffic flowing between nodes.

(b) To identify temporal states in the network, a similarity matrix based on Equation 1 helps identify communities for dominant communities.

(c) Difference graph between adjacent time steps, caused by addition or deletion of an edge. The left half of the rhombus represents the community of the node for time step G_t and right half for time step G_{t+1}. The sizes of the nodes depict the magnitude of change.

Fig. 1: Similarity and difference graph from a network flow topology. The graphs summarize topological behavior over time and depict low-level topological patterns characterizing state change.

3. **Identifying similarity communities with temporal network states:** Network changes can be viewed as continuous structural changes where sites that constantly engage can be grouped to form communities, e.g., by recognizing permanent flow communication between certain sites. This analysis can reveal normal and abnormal patterns, thereby supporting the detection of potential security threats to a WAN structure.

3 Netostat Methodology

The architecture 1 is based on a two-stage approach. The first stage performs *similarity analysis* to identify communities through community detection algorithms [42], as well as temporal states and day/night patterns. The second stage performs *difference analysis* and visualizes difference topologies across two timesteps for further detailed topological analysis. Furthermore, in order to find and explore the community detection and similarity results, Netostat provides the ability to interactively tweak and reiterate the metric and the community detection results.

3.1 Mathematical Notation

Figure 1a shows a simple network mesh topology used and flows simulated for three time steps $t = 0, 1$ and 2. This data is modeled as a graph $G = (V, E)$, consisting of vertices $V := \{v_1, \dots, v_n\}$ and edges $E \subseteq V \times V := \{e_1, \dots, e_m\}$. The edges may be weighted, i.e., a value $e_w \in \mathbb{R}$ may be attached to each $e \in E$ for a fixed time step.

3.2 Social Cluster Analysis

Sites that communicate frequently, can reliably be detected as sub-networks using the Louvain algorithm [42]. This algorithm uses a maximized objective modularity, measuring the quality of communities, where each community has dense intra-modular connectivity and sparse inter-modular connectivity. This metric is defined as,

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \tag{1}$$

where Q is the modularity metric with values in $[-1, 1]$, m is number of links in the graph, A_{ij} is the weight of the edge between node i and j, c_i and c_j depict the communities that nodes i and j belong to. The δ function is equal to 1 if the node communities i and j are the same, with k_i and k_j depicting the node degrees of i and j, respectively.

3.3 State Similarity Computation

To better characterize day or night patterns, transient evening patterns, or weekly, daily patterns, we need a metric that quantifies the amount of change. Netostat characterizes this behavior by detecting topological change between graphs and clustering similar time steps into a temporal state.

Using the metric introduced by Koutra et al. [43], first, we identify the similarity value across all pairs of time steps. Second, we transform similarity values into an adjacency matrix. Third, we use this matrix as an input to a community detection algorithm, Louvain [42], which determines a cluster of graphs that have similar topological behavior, *i.e.*, day/night. Once the states are detected, Netostat produces respective similarity and differences graphs based on the dynamic graphs.

Mathematically, we define the metric between graphs G_t and G_{t+1} as, $S(G_t, G_{t+1}) \in [0, 1]$, where a 1 represents two graphs being exactly the same with same edge-weights, while a 0 represents two graphs being completely dissimilar in its topology and edge-weight (flows). To determine this value, we define a vector $\vec{s_i}$ per node i, $\vec{s_i} = [s_{i,1} \dots s_{i,n}]$, where the influence scores start from i_{th} node and end at n_{th} node. This vector can then be stacked as an $n \times n$ vector-matrix S for every node in the graph. The similarity metric [43] identifies the flow changes in the dynamic graph as,

$$S = [s_{ij}] = [I + \epsilon^2 D - \epsilon A]^{-1} \tag{2}$$

Here $\epsilon = \frac{1}{1+max(d_{ii}))}$ is a constant that captures the influence between neighboring nodes, and D is a $n \times n$ diagonal matrix, where $d_{ii} = \sum_j a_{i,j}$ is the node degree. A is the adjacency matrix, and I is the identity matrix. We compute graph distance as,

$$d = RootED(S_1, S_2) = \sqrt[2]{\sum_{i=1}^{n} \sum_{j=1}^{n} (\sqrt{s_{1,ij}} - \sqrt{s_{2,ij}})^2} \tag{3}$$

The final similarity value of two graphs is defined as,

$$sim(G_1, G_2) = \frac{1}{1+d} \tag{4}$$

With the all-pairs similarity values (from Eq. 4) embedded in an adjacency matrix, the detected time intervals states, are then used to compute similarity graphs. The reduced similarity graph (fig. 1b) provides a summary of the topology prevalent during a particular state, *e.g.,* night time. The nodes in similarity graphs possess the community changes that happened during the state period while the edges depict the mean edge weight.

3.4 Similarity Graph Computation

To better understand the major pivotal sites, evolution patterns, and communities during temporal states, we devise a methodology that can detect a graph, that provides a summary of a temporal state. For a given period, the similarity graphs represent a simple abstraction of the graph-level complexities within the time frame. The visual representations ([38]) of the similarity graph depict summarized topological information, which includes the community, node membership, and edges. We use the algorithm from [38] to construct and depict the visual representation of the summarized graphs.

3.5 Difference Graph Computation

While the similarity graphs (as defined in [38]) depict general, overall trends within the dynamic network, lower level topological trends are hidden within the metric. The lower level patterns, like the addition/deletion of edges across time steps is important to depict how states are detected. Difference graphs [38] can help depict such topological patterns effectively. To best characterize change across time steps within a dynamic graph, we use the following criteria for change:

1. **Is there a change:** While comparing graphs, have the edges been added or deleted?
2. **The magnitude of change:** Given a change, to what effect have the edges been changed?
3. **Community membership change:** Have the community membership of the node changed across timestep?

Note, we assume our networks have stable topology node configurations, with only flow changes recorded as dynamic edges. To analyze the changes in difference graphs, we define the importance of edge-change through a metric known as Magnitude of Edge-Change. Furthermore, in our difference graph visualization, we encode edge-thickness, with the importance/magnitude of its change *w.r.t* to subsequent time steps.

3.6 Visualizing Change

For every difference graph, the topological change is characterized by visualizing only the change happening across two timesteps. This allows us to find core nodes that govern the entire network operation, potentially being vulnerable to caching or load-balancing. This metric is then provided to the visual topology renderer to scale the nodes based on the magnitude of change across two-time steps.

Magnitude of Edge-Change: To better identify critical nodes and potential sites of failure within a network, we need a mechanism to quantify the amount of change across time steps in a difference graph.

To visualize a particular edge-change, $C_i(t_k, t_{k+1})$ between two time steps t_k and t_{k+1} we define the metric with edge-change $C_i(t_k, t_{k+1})$,

$$C_i(t_k, t_{k+1}) \propto \|(E_k - E_{k+1}, f(N_{i,k}) - f(N_{i,k+1}))\| \tag{5}$$

where E_i, N_i are the edge and nodes in time step k and node id i. Equation 5 defines changes between two adjacent timesteps t_k and t_{k+1}. $E_k - E_{k+1}$ is the edge-set in difference graph and $f(N_k) - f(N_{k+1})$ is the difference in a flow movement measures in the graph, where $f(N)$ is a function describing the nodes centrality or its betweenness centrality for timestep k, and a nodeid l. Specifically, for *e.g.,* the change between $C_i(t_1, t_2)$ is directly proportional to the edge set of $(E_1 - E_2)$ and the $f(N_1) - f(N_2)$ where $f(N_1)$ and $f(N_2)$ are the centrality of the node, N_1 and N_2. Timestep is defined as k, where k is anything from $1..M$, where M is the end of the dataset.

Specifically, Equation 6 describes a measure of difference, *difference centrality* across two time steps t_k, t_{k+1} for nodes $N_{i,k}$ and $N_{i,k+1}$. This measure represents flow changes per node with time, providing information about possible new sites/nodes being vulnerable to link failures or needing additional caching support.

$$DC(t, N_i) \propto \|(\sum_{j=0}^{N_n} e_{i,j}^{t_k} * f(N_i)^{t_k}) - (\sum_{j=0}^{N_n} e_{i,j}^{t_k+1} * f(N_i)^{t_k+1})\| \tag{6}$$

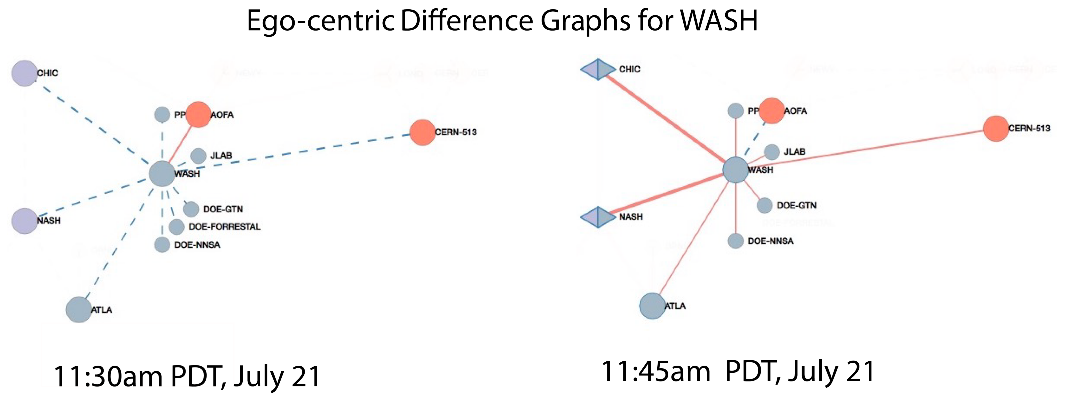


Fig. 2: Flow routing patterns of central site, Washington (ESNet), between *July 21 11:30am PDT - July 21 11:45am PDT*. Flow increases between NASH and WASH, causing a state change in NASH as the day progresses.The blue dotted lines (left graph) indicate the reduction of packets reaching WASH; however, the sudden increase in packets reaching NASH and CHIC results in the change of community, causing the change from light purple to blue color.

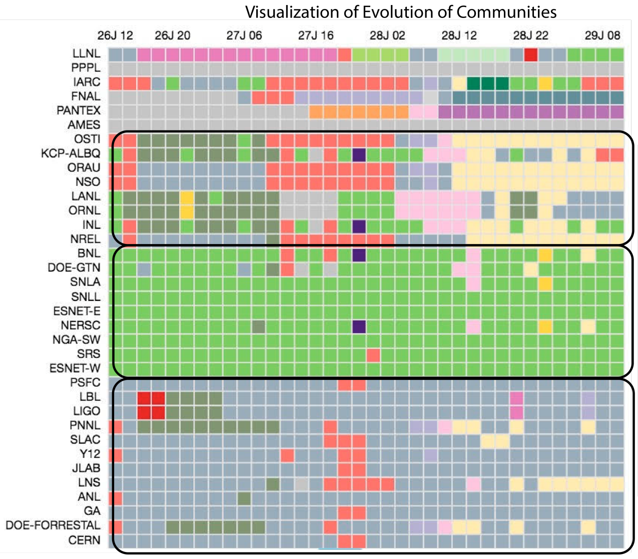


Fig. 3: Evolution of community membership in ESNet, with two major communities being formed stable friendly and dynamically changing communities. The x-axis represents time, and the y-axis represents the ESNet sites. Each cell in the matrix represents the community membership for a given ESNet site at a particular time point.

Two major visual encodings can be used for depicting the underlying visual information in the difference graph and similarity graph, including the following,

- Changes in community membership.
- Edge weight deviation.
- Addition or deletion of edges.

This approach was inspired by the encoding method discussed in [38]. For *similarity graphs* we visualize every node as a pie chart depicting the magnitude of different communities present for a certain period for a site, while the edges depict the standard deviation of the edge weight. Beyond a certain threshold for edge weights, the edges become dotted blue lines.

For *difference graphs*, the change in community memberships are represented by a rotated rhombus (Figure 1c), where the left half depicts community membership of the previous time step and its right half depicts community membership for the current time step. The dotted blue lines depict the deletion of an edge, and solid red edges depict the addition of edges relative from the previous to the current time step, Figure 2. Further, the larger the size of the node, the higher is the change in flow for that node across time, according to Eq. 6.

4 Experimental WAN Analysis

We have applied Netostat to two real-world WAN data sets to understand dynamic behavior.

4.1 Datasets

4.1.1 U.S. Research Network – ESnet

The Energy Science Network (ESnet), a Department of Energy (DOE) research network providing high-bandwidth, loss-less, provides reliable connectivity to scientists at U.S. national laboratories, universities, and other research institutions. ESnet monitors network connections,

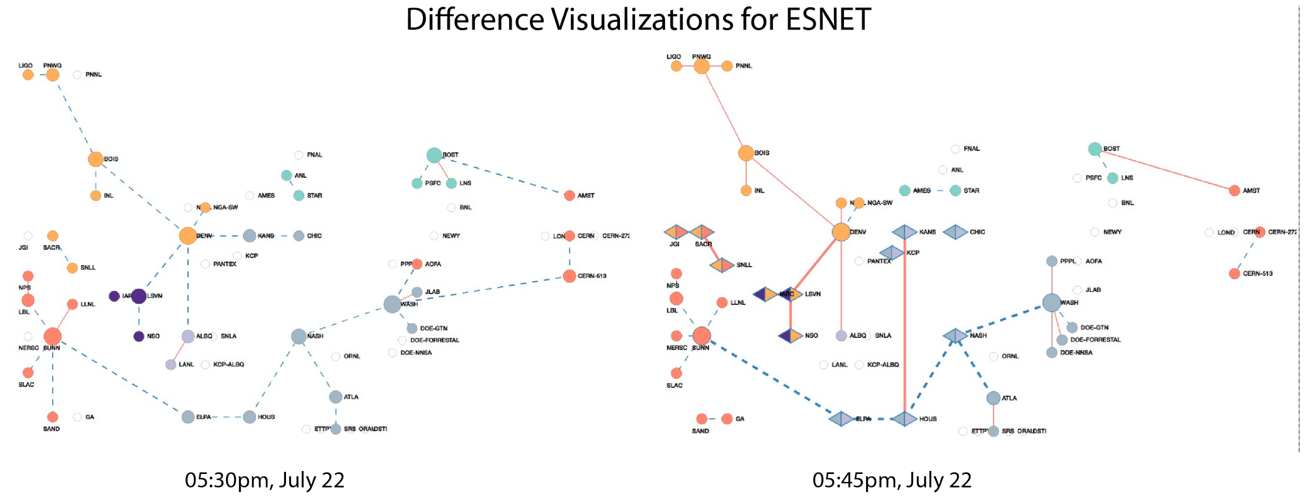


Fig. 4: Flow changes between 05:15pm - 05:30pm, July 22, and 05:30pm-05:45pm, July 22. Large node size indicates large flow change. Dashed blue lines show reduction in network flow, and red lines represent addition of new flow. One can see a core structure forming between IAR, LSVN, and NSO.

collecting statistics for bytes sent/received, and link performance logs. One monitoring tool, ESxSNMP, collects router-in and router-out bytes for every interface, every 30 seconds. The tool records the packets transferred between sites at different times of the day. While physical network topology is fixed, the virtual topology of data movement changes dynamically, depending on a site's access to data.

To examine the evolution of communities over time, we consider the traffic data collected for the three days from July 26, 2017, 12:00 pm PDT, to July 29, 2017, 8:00 pm PDT for analysis, see Figure 3. For the dynamic topology, we use traffic flow data recorded as SNMP for two days collected for 15-minute intervals, from *21 July 2017, 1:00 pm PDT* to *23 July 2017, 5:00 am PDT*, consisting of 80 time steps for 33 sites. For site abbreviations, we refer to `https://my.es.net/sites/list`. To handle the size of the data in the temporal dimension, we use a threshold to model the data as a dynamic graph. We use an undirected graph by averaging the bidirectional links to the sites, see Figures 2 and 4.

4.1.2 European Research Network – GEANT

GEANT, a European data network for research and education, has a connecting node in each European country, transporting data between universities and laboratories. To evaluate the effectiveness of our visual analysis system, we decided to use the GEANT backbone network [44]. The GEANT network includes 23 peer nodes and 120 undirected links. We use 2004 traffic data, sampled from the GEANT networks at 15-minute intervals. From the 10,772 traffic matrices, we use the most relevant 80-time steps for the analysis of our data sets. We discuss our results for this network for the period from *June 04, 5:00 pm GMT* to *June 05, 8:00 am GMT*.

5 Result Analysis

5.1 Visualizing Topology Information

Studying ESNet data sheds light on the inner workings of this vast U.S. network. The nodes, or sites, and edges depict the communication patterns between the sites.

The difference graphs are shown in Figure 4 represent changes across time points, 05:15 pm - 05:30 pm, July 22, and 05:30 pm-05:45 pm, July 22, respectively, showing state-change from day to night patterns. The dashed lines represent edges decreasing flow, while the solid orange lines show an increased flow rate. The Louvain community detection algorithm can identify group-like patterns in a graph for a given time step showing friendly and non-friendly sites in the network. Larger node sizes indicate a larger change in overall topology in the node of interest. Communities such as NSO, IAR, and LSVN form their core community (dark blue) only consumed by the orange community in the Northwest of the United States. One sees that the communities forming are spatially co-located with each other, implying that sites close to each other often communicate due to proximity. For example, LIGO, PNWG, and BOIS form an orange community. Another example is LBL, forming communities with CERN (in Europe) indicating distant experiment communication during the day.

To explore friendly stable vs. dynamically changing sites, we visualize community membership evolution by a heatmap see Figure 3. The figure shows community evolution detected by the algorithm for SNMP data from *July 26, 2017, 12.00 pm PDT*, to *July 29, 2017, 8.00 pm PDT*), showing groups, stable friendly communities, and dynamically changing communities.

5.2 Detecting Major Patterns in U.S. Network

Major evolving ESnet flow patterns need to be studied for an efficient re-design of the network [45,46]. For example, network engineers can optimize network links and routing behavior to best cater to different kinds of flows (large, small) over sites during different times.

Specifically, questions like, what sites are *friendly* and often collaborate? how do flow connections vary over time? do such communication patterns reveal common patterns between network sites? what are potential sites that may cause disruptions or are prone to a targeted attack?

Netostat can identify dynamic communities forming and recognizing temporal states in the recorded period. Using the approach in eq 1, 2, 3, 4, one can find two types of dominant network states corresponding to communication behavior during day and night, relative to the PDT timezone.

The similarity graphs and difference graphs, shown in Figure 5B and C, depict consistent topological and community patterns for four periods, also shown in Figure 5A:

1. State 1 ranges from *1:00pm - 5:30pm, Jul 21.*
2. State 2 lasts from *5:30pm Jul 21 - 5:00am Jul 22.*
3. State 3 ranges from *5:00am - 5:00pm PDT Jul 22.*
4. State 4 ranges from *5:00pm Jul 22 - 5:30am Jul 23.*

The similarity graph depicting an individual state, state 2, (Figure 5B) represents consistent evening-night-time operations in PDT time. During this period, three major communities, *green, orange, and purple* are detected. While, geographically closer sites like LIGO, and PNNL form communities, geographically far distant sites like BNL, NERSC, and NSO also form their communities, indicating experiments and interactions occur all across the network. Site CERN forms communities with GA, LIGO, NREL NSO, and ANL consistently although it is geographically located far away (in Europe). Further, the similarity graph in Figure 5C, conveys the overall flow behavior during the day and also indicates possible experiments/interactions running across time zones. As a network administrator considering a potential re-design of the network, one can take into account such frequently interacting sites and their routing behavior to reserve network resources and improve the underlying routing policies governing the network.

Figure 6 shows two contrasting patterns, pattern A, fig. 6, and pattern B, fig. 6 representing day and night flow patterns for site SNLL respectively. During the day, site SNLL plays a central dynamic role in transferring flow to a wide variety of geographic locations, SNL, SNLA, and SRS. Further, NGA-SW transitions from an orange to a dark-purple community, indicating its frequent dynamic collaboration with SNLL, SNLA, and SRS, suggesting a potential point of failure within the network. Pattern B, in fig. 6, on the other hand, depicts relative stability between selected sites, SNLL, SNLA, and SRS, characterized by green squares.

In summary, visualizing the time-varying flow behavior with Netostat makes it possible to determine stable and unstable time-varying connectivity behavior. It is possible to understand temporal data by automatically identifying similarities and differences supporting intuitive pattern identification. The similarity graphs show consistent *friendly* communication patterns over time, while the difference graph shows the underlying low-level flow changes causing the major state change. Such patterns can further be statistically explored in detail to construct alternative routing paths to better transfer information across sites.

5.3 Analysis over Larger Periods of Time

For evaluation of system usability and determining limitations, we perform analysis over larger time periods. We analyze network data from June 22, 2017, 4:00am to June 27, 2017, 6:00am for a period of about 203 time steps with an interval of 30 minutes. We want to better understand long-term patterns that are dominant across network sites. We explore these questions: What are the routing signatures for weekend and weekday patterns? What are potential points of failure during a state transition from weekday to weekend?

The metrics plot provides insight into the two major states established by the method, *i.e.,* weekend states and weekday states. Three temporal states are deter-

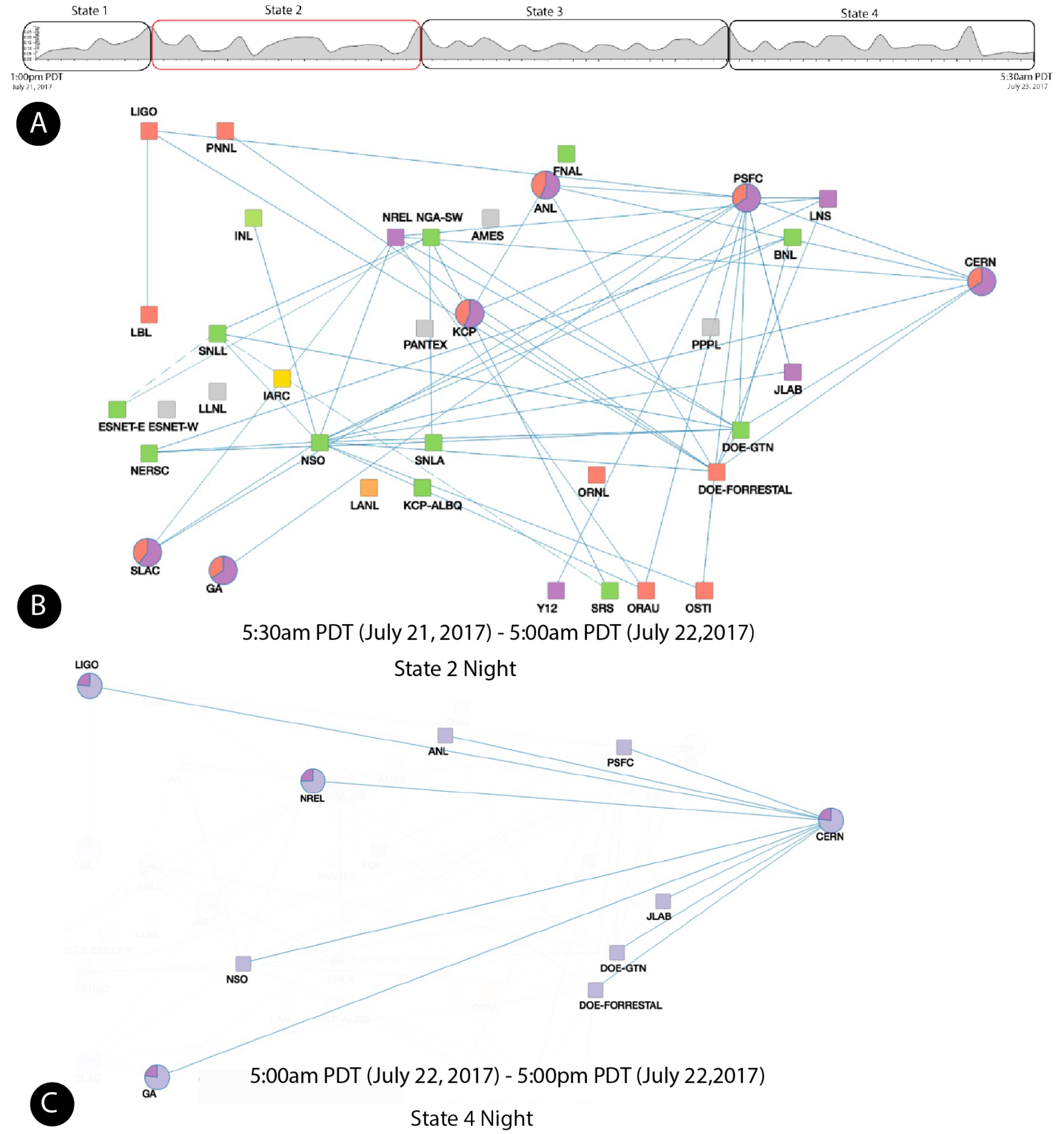


Fig. 5: Similarity and difference plots for ESNet flow. (A.) Plot of similarity metric showing four states, indicating day/night patterns. (B.) Similarity visualization for period from 21 July, 5:30 pm, to 22 July, 5:00 am, detecting six nodes dynamically changing community. (C.) Interaction patterns between CERN and other sites during day.

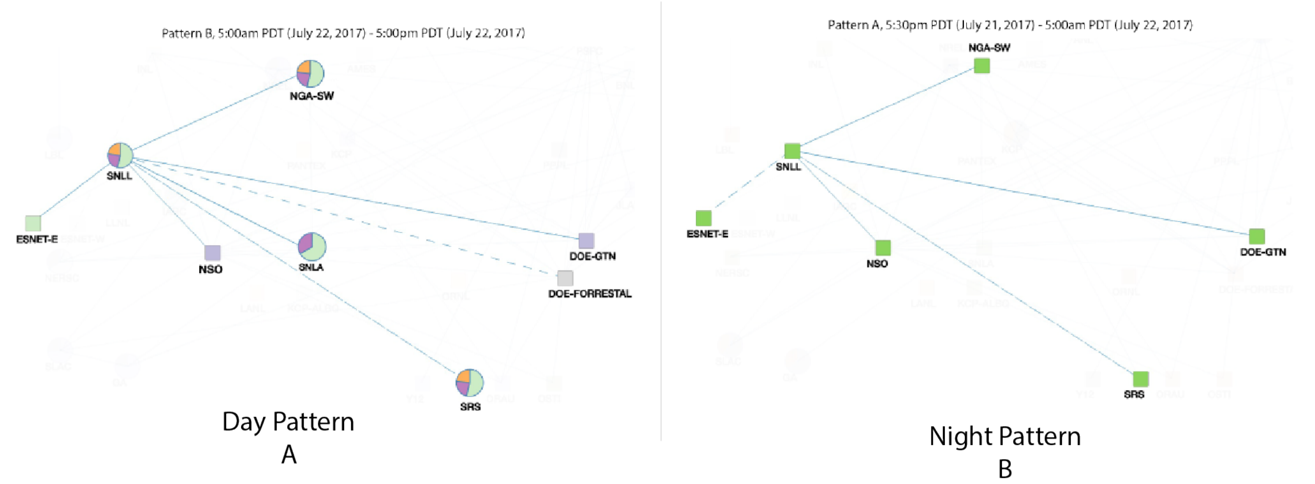

Pattern B, 5:00am PDT (July 22, 2017) - 5:00pm PDT (July 22, 2017)
NGA-SW
SNLL
ESNET-E
NSO
SNLA
DOE-GTN
DOE-FORRESTAL
SRS
Day Pattern
A
Pattern A, 5:30pm PDT (July 21, 2017) - 5:00am PDT (July 22, 2017)
NGA-SW
SNLL
ESNET-E
NSO
DOE-GTN
SRS
Night Pattern
B

Fig. 6: Similarity plots for ESNet network flow for SNLL. Overview of different states detected via corresponding similarity topology. Nodes SNLL and SRS switch communities frequently during this period and are stable during the night time interval.

mined, *state 1*: June 22, 2017, 4:00am – June 24, 2017, 1:00am; *state 2*: June 24, 2017, 1:00am – June 26, 2017, 2:00am; and *state 3*: June 26, 2017, 2:00am – June 22, 2017, 6:00am. Fig.7B shows the topological differences across the network during a weekend (top) and during the transition phase from weekend to weekday (bottom). A pattern can be seen clearly when comparing the left and right difference graphs: The left graph is a more sparse graph indicating less topological variation during the weekday, while the transition phase difference graph, shown on the right, indicates the dynamic routing nature of the transfer of packets across sites.

Specifically, the sites INL, IARC, NSO, and DOE-GTN dynamically route and manage multiple paths, causing changes of the communities they belong to during the transition period phase. Different behavior is shown by sites like GA, SLAC, and NERSC – not participating. Red edges in the graph indicate traffic slowly building up in the network, potentially causing bottlenecks around INL, PNNL, and PANTEX. Additional statistical analysis would make it possible to further investigate the findings of our method in more quantitative detail.

5.4 Flow Visualization in a European Network

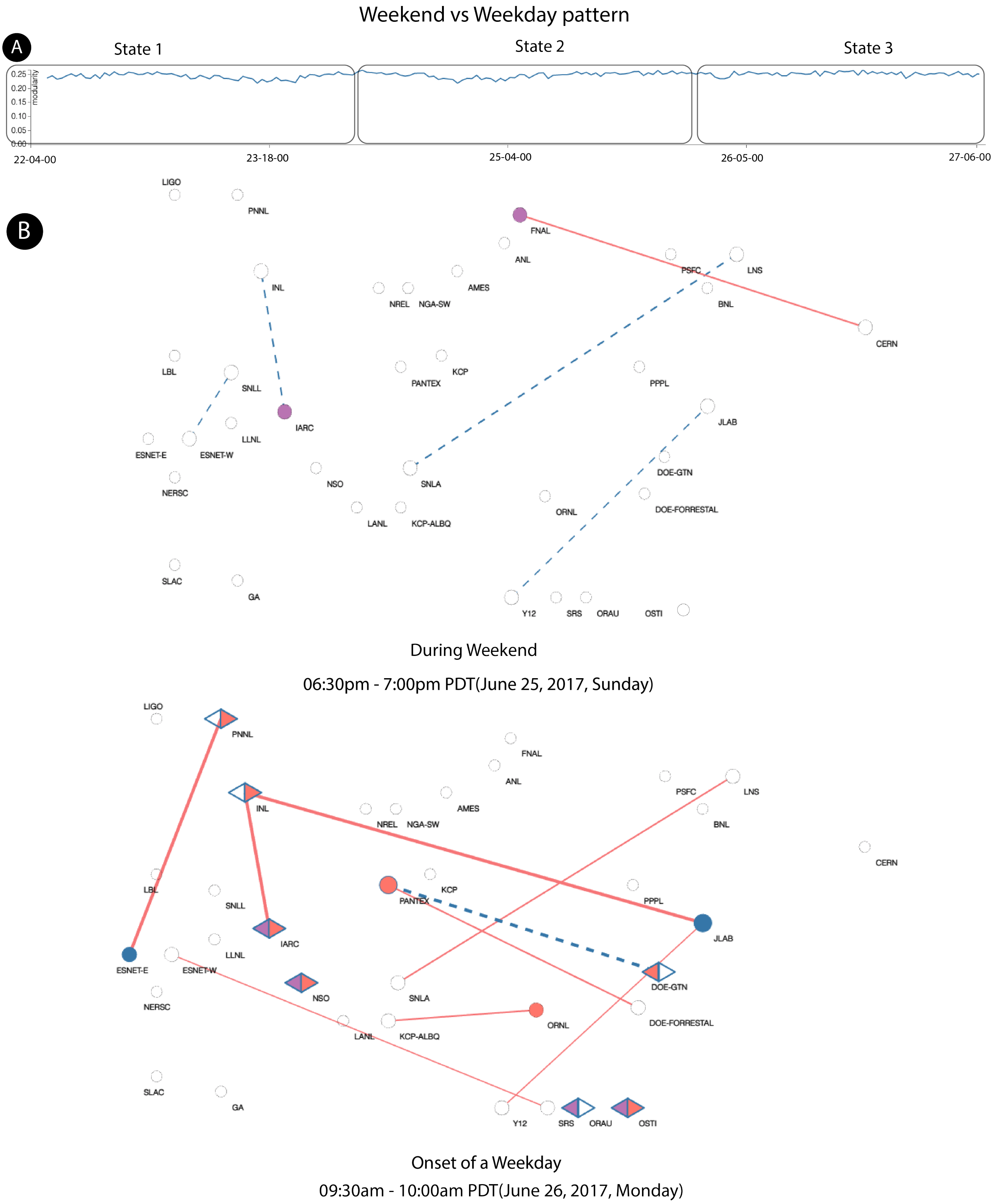
We also performed a detailed analysis of flow patterns in the GEANT data sets using Netostat. By primarily identifying day, evening, and night patterns, one wants to determine inherent changes in flow patterns and find out whether these patterns support a better understanding of potential network failure points. The data used is a 24-hour open data set available online with flow information. The time is GMT.

Netostat identifies dynamic communities forming three major temporal states. Shown in Figure 8A are for following time periods:

- State 1: 5:04pm, June 4 - 12:19am, June 5, 2004. State 1 represents evening;
- State 2: 12:19am, June 5 - 12:49am, June 5, 2004. State 2 a transient state;
- State 3: 12:49am, June 5 - 8:03pm, June 5, 2004. State 3 is night state.

Figure 8C shows similarity graphs. One can see the differences between evening and night patterns. As a general trend, the transient nodes 7, 8, 3, and 0 change their community memberships often (pie circles). Communication patterns during the night are quite stable when known sites and communities talk to each other without changing their community memberships (square glyphs).

The difference graph shown in Figure 8B shows the time point when the network transitions from a transient state to a night state. As a general trend, the visualization shows an overall reduction in the number of links in the graph. Few nodes, for example, 9, 8, 21, and 4, and 3, change their community memberships. An apparent difference is the size of node 11, where, despite not having changed its community, the large size indicates that it is the information hub of transfers during this transition.



During Weekend

06:30pm - 7:00pm PDT(June 25, 2017, Sunday)

Onset of a Weekday

09:30am - 10:00am PDT(June 26, 2017, Monday)

Fig. 7: Metrics plots and difference topology of network flow in ESNeT network over a larger time window. A.) Evolution of modularity metric for period *June 22, 4.00am – June 27, 6.00am PDT*. The method detects three major states pertaining to weekend and weekday temporal states; B.) Difference graphs, during weekend state (top) and state change from Sunday to Monday, weekend to weekday (bottom). Nodes IARC, NBO, SRS, OSTI change their community memberships often, in relation to other nodes that remain stable throughout. During weekday, the difference graphs have fewer connected nodes within them compared to the transition difference depicting multiple changes, including changes in the sizes of packets being transferred across sites.

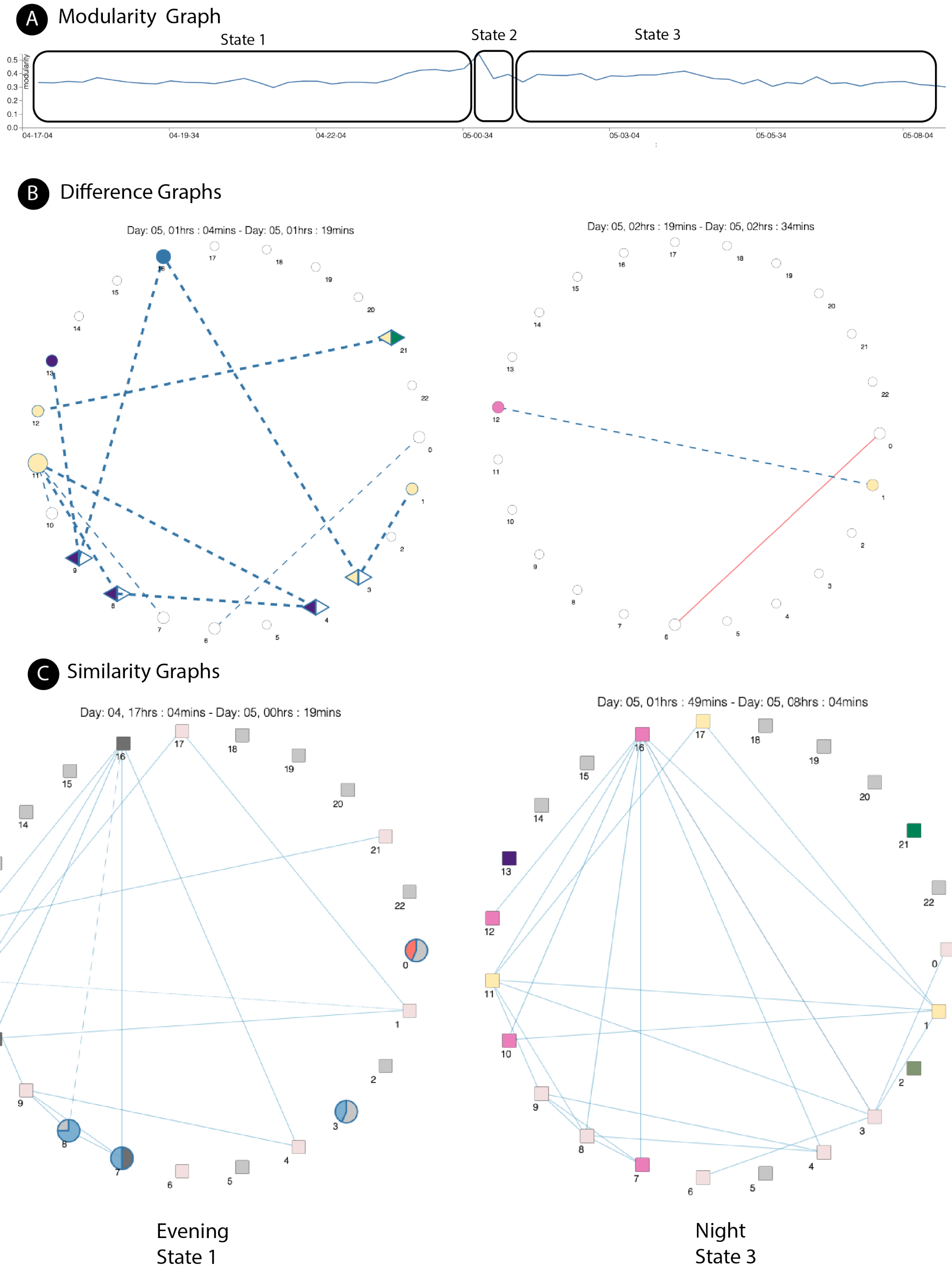


Fig. 8: Similarity and difference topology of network flow in GEANT network. (A.) Evolution of modularity metric for period *June 04, 5.00pm - June 05, 8.00am GMT.* (B.) Difference graphs, during state change from evening to night (top) and graph indicating change within the night state (bottom). Nodes 9, 8 and 4 switch their communities often when compared to other nodes that remain stable throughout. Node sizes in difference graphs are determined through equations 5 and 6, depicting traffic magnitudes handled by the routers at the respective sites (C.) Similarity graph showing evening and night states. During night, the similarity has less inter-connected nodes communicating with each other. Each site is colored based on its community affiliation.

The other difference snapshot depicted in Figure 8B shows the relative stability of the network during the middle of the night. The modularity metric depicted in Figure 8A represents the relative change in modularity during the transient state at 12:15 am.

A simple analysis of the data provides important information to network administrators, e.g. node 11, although not changing community, being a hub during state transition. Considering the similarity graphs, for example, nodes 16, 1, and 11 are always engaged in overall network operation, indicating that it might be advisable to improve bandwidth links or deploy additional infrastructure to avoid network congestion.

A preliminary analysis provides sufficient insight for a subsequent, more rigorous statistical analysis to determine and ensure overall network robustness. The growth of the GEANT network could, for example, indicate the need for providing additional resources to specific high-in-demand nodes.

5.5 Comparing Netostat with other techniques

We briefly compare our methodology with other existing methods used for graph visualization and analysis methods for dynamic networks and explain the conceptual advances of our approach.

Traditionally, existing network analysis tools use a hybrid version of animations and small multiples to visualize dynamic graph data. While a breadth of insights can be gleaned with such methods, users often cannot notice major topological differences between two adjacent graphs since recognizing changes is perceptually challenging. The identification of such patterns is important to effectively detect the onset of major community evolutionary or topological changes.

With techniques like small multiples, large changes with similar graphs can be relatively hard to find due to change blindness. Through animation, it may be even harder to keep track of changes, both sudden or minuscule changes due to limitations of our short-term memory. Our tool addresses these issues by explicitly showing the exact differences across time steps and providing a summary version of the dynamic graph that could not fit perceptually.

Further, Netostat supports the identification of stable and constantly evolving sites; it makes possible the exploration of the relationship of evolving graph topology and community membership that can be easily identified through the application of difference graphs over time.

6 Conclusions and Future Work

Identification of potential failures and understanding network evolution day and night is crucial to construct robust operating networks. Computing and visualizing these patterns over different periods helps inform, prevent, and diagnose any network alerts that reach a network administrator. For example, visual analysis capabilities used when diagnosing load-balancing issues, e.g., traffic congestion at a particular link due to network topology, improve the overall understanding and operation of the network. Visualization tools help network administrators understand the cause-and-effect relationships of network problems occurring over time.

Netostat uses principles from social network analysis to visualize flow communication patterns for time-varying networks. Our approach extracts the major differences in communication flows over time, identifying states within networks, and visualizes important changes. When applying Netostat to two R&E networks, it is possible to recognize day/night patterns helping network engineers to quickly identify unexpected communication patterns and provide visual insights into the operation of the network.

Concerning potential future research, the similarity and difference graphs can be employed as part of machine learning algorithms to help identify new network states that are unexpected and potential security threats. These states can also be selected to identify new communication patterns that can train a machine learning model to predict possible future bottlenecks. The bottlenecks describe the links that are badly designed with less capacity that becomes heavily loaded due to the traffic surges.

7 Funding Information

This work was supported by the U.S. Department of Energy, Office of Science Early Career Research Program for ‘Large-scale Deep Learning for Intelligent Networks’ Contract no FP00006145

8 Author Contribution

SM, MK, devised the visual analytics system; SM implemented and tested the system as a Native Node JS tool. MK was the PI of the project, and MK, GW, BH provided feedback for the system prototypes and the case studies. MK got the ESNet data and performed additional analysis of the network data set. SM, BH, MK and GW wrote the manuscript, and all authors read, revised, and approved the final manuscript.

9 Data Availability Statement

The Netostat source code, and the ESNet dataset will be made publicly available on Github at https://github.com/sugeerth/NetoStat. The GEANT dataset is already publicly available.

10 Competing Interests

The authors declare that they have no competing interests.

11 Informed Consent Statement

Not applicable.

12 Ethical Statement

Not applicable.

References

1. V. Bourassa and F. Holt, "Swan: Small-world wide area networks," in *Proceeding of International Conference on Advances in Infrastructures*, 2003.
2. J. Ros-Giralt, A. Bohara, S. Yellamraju, M. H. Langston, R. Lethin, Y. Jiang, L. Tassiulas, J. Li, Y. Tan, and M. Veeraraghavan, "On the bottleneck structure of congestion-controlled networks," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 3, no. 3, Dec. 2019. [Online]. Available: https://doi.org/10.1145/3366707
3. Y. Hong, S. Mandal, M. Al-Fares, M. Zhu, R. Alimi, K. N. B., C. Bhagat, S. Jain, J. Kaimal, S. Liang, K. Mendelev, S. Padgett, F. Rabe, S. Ray, M. Tewari, M. Tierney, M. Zahn, J. Zolla, J. Ong, and A. Vahdat, "B4 and after: Managing hierarchy, partitioning, and asymmetry for availability and scale in google's software-defined WAN," in *ACM Special Interest Group on Data Communication*, ser. SIGCOMM '18. New York, USA: Association for Computing Machinery, 2018, p. 74–87. [Online]. Available: https://doi.org/10.1145/3230543.3230545
4. Seong Soo Kim and A. L. N. Reddy, "A study of analyzing network traffic as images in real-time," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, 2005, pp. 2056–2067.
5. C. G. Healey, "Perception in visualization," 2013, https://www.csc2.ncsu.edu/faculty/healey/PP/.
6. G. Robertson, D. Ebert, S. Eick, D. Keim, and K. Joy, "Scale and complexity in visual analytics," *Information Visualization*, vol. 8, no. 4, pp. 247–253, 2009.
7. B. Yost, Y. Haciahmetoglu, and C. North, "Beyond visual acuity: the perceptual scalability of information visualizations for large displays," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2007, pp. 101–110.
8. O. Markelov, V. N. Duc, and M. Bogachev, "Statistical modeling of the internet traffic dynamics: To which extent do we need long-term correlations?" *Physica A: Statistical Mechanics and its Applications*, vol. 485, pp. 48–60, 2017. [Online]. Available: https://doi.org/10.1016/j.physa.2017.05.023
9. S. Uhlig, "On the complexity of internet traffic dynamics on its topology," *Telecommunication Systems*, vol. 43, no. 3, pp. 167–180, 2010. [Online]. Available: https://doi.org/10.1007/s11235-009-9213-6
10. k. claffy, "Internet traffic characterization," Ph.D. dissertation, UC San Diego, June 1994.
11. Q. Lu, L. Zhang, S. Sasidharan, W. Wu, P. DeMar, C. Guok, J. Macauley, I. Monga, S. Yu, J. H. Chen, J. Mambretti, J. Kim, S. Noh, X. Yang, T. Lehman, and G. Liu, "Bigdata express: Toward schedulable, predictable, and high-performance data transfer," in *IEEE/ACM Innovating the Network for Data-Intensive Science (INDIS)*, 2018, pp. 75–84.
12. X. Yin, W. Yurcik, M. Treaster, Y. Li, and K. Lakkaraju, "Visflowconnect: netflow visualizations of link relationships for security situational awareness," in *Workshop on Visualization and data mining for computer security.* ACM, 2004, pp. 26–34.
13. H. Shiravi, A. Shiravi, and A. A. Ghorbani, "A survey of visualization systems for network security," *IEEE Transactions on visualization and computer graphics*, vol. 18, no. 8, pp. 1313–1329, 2012.
14. L. Xiao, J. Gerth, and P. Hanrahan, "Enhancing visual analysis of network traffic using a knowledge representation," in *Visual Analytics Science And Technology.* IEEE, 2006, pp. 107–114.
15. T. Von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J.-D. Fekete, and D. W. Fellner, "Visual analysis of large graphs: state-of-the-art and future research challenges," in *Computer graphics forum*, vol. 30, no. 6. Wiley Online Library, 2011, pp. 1719–1749.
16. F. Beck, M. Burch, S. Diehl, and D. Weiskopf, "The state of the art in visualizing dynamic graphs," *EuroVis STAR*, vol. 2, 2014.
17. C. Aggarwal and K. Subbian, "Evolutionary network analysis: A survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, p. 10, 2014.
18. R. F. Erbacher, "Visual traffic monitoring and evaluation," in *International Symposium on the Convergence of IT and Communications.* International Society for Optics and Photonics, 2001, pp. 153–160.
19. R. Ball, G. A. Fink, and C. North, "Home-centric visualization of network traffic for security administration," in *Workshop on Visualization and data mining for computer security.* ACM, 2004, pp. 55–64.
20. J. R. Goodall, W. G. Lutters, P. Rheingans, and A. Komlodi, "Preserving the big picture: Visual network traffic analysis with tnv," in *Visualization for Computer Security.* IEEE, 2005, pp. 47–54.
21. K. Lakkaraju, W. Yurcik, and A. J. Lee, "Nvisionip: netflow visualizations of system state for security situational awareness," in *Workshop on Visualization and data mining for computer security.* ACM, 2004, pp. 65–72.
22. C. Ahlberg, C. Williamson, and B. Shneiderman, "Dynamic queries for information exploration: An implementation and evaluation," in *Proceedings of the SIGCHI*

conference on Human factors in computing systems, 1992, pp. 619–626.

23. P. Isenberg and D. Fisher, "Collaborative brushing and linking for co-located visual analytics of document collections," in *Computer Graphics Forum*, vol. 28, no. 3. Wiley Online Library, 2009, pp. 1031–1038.
24. T. Takada and H. Koike, "Tudumi: Information visualization system for monitoring and auditing computer logs," in *Information Visualisation*. IEEE, 2002, pp. 570–576.
25. A. Komlodi, P. Rheingans, U. Ayachit, J. R. Goodall, and A. Joshi, "A user-centered look at glyph-based security visualization," in *Visualization for Computer Security*. IEEE, 2005, pp. 21–28.
26. J. McPherson, K.-L. Ma, P. Krystosk, T. Bartoletti, and M. Christensen, "Portvis: a tool for port-based detection of security events," in *Workshop on Visualization and data mining for computer security*. ACM, 2004, pp. 73–81.
27. R. A. Becker, S. G. Eick, and A. R. Wilks, "Visualizing network data," *IEEE Transactions on visualization and computer graphics*, vol. 1, no. 1, pp. 16–28, 1995.
28. S. T. Teoh, K. L. Ma, S. F. Wu, and X. Zhao, "Case study: Interactive visualization for internet security," in *Visualization*. IEEE Computer Society, 2002, pp. 505–508.
29. S. Murugesan, K. Bouchard, J. A. Brown, B. Hamann, W. W. Seeley, A. Trujillo, and G. H. Weber, "Brain modulyzer: Interactive visual analysis of functional brain connectivity," *IEEE/ACM Tran. on Comp. Biology and Bioinformatics*, 2016.
30. V. D. Calhoun and T. Adali, "Time-varying brain connectivity in fmri data: Whole-brain data-driven approaches for capturing and characterizing dynamic states," *IEEE Signal Processing Magazine*, vol. 33, no. 3, pp. 52–66, 2016.
31. S. Murugesan, K. Bouchard, E. Chang, M. Dougherty, B. Hamann, and G. Weber, "Multi-scale visual analysis of time-varying electrocorticography data via clustering of brain regions." *BMC bioinformatics*, vol. 18, no. Suppl 6, p. 236, 2017.
32. S. Murugesan, K. Bouchard, E. Chang, M. Dougherty, B. Hamann, and G. H. Weber, "Hierarchical spatio-temporal visual analysis of cluster evolution in electrocorticography data," in *International Conference on Bioinformatics, Computational Biology, and Health Informatics*. ACM, 2016, pp. 630–639.
33. D. J. Simons and D. T. Levin, "Change blindness," *Trends in cognitive sciences*, vol. 1, no. 7, pp. 261–267, 1997.
34. D. Archambault, H. C. Purchase, and B. Pinaud, "Difference map readability for dynamic graphs," in *International Symposium on Graph Drawing*. Springer, 2010, pp. 50–61.
35. D. Archambault, "Structural differences between two graphs through hierarchies," in *Proceedings of Graphics Interface 2009*. Canadian Information Processing Society, 2009, pp. 87–94.
36. R. Bourqui and F. Jourdan, "Revealing subnetwork roles using contextual visualization: Comparison of metabolic networks," in *Information Visualisation, 2008. IV'08. 12th International Conference*. IEEE, 2008, pp. 638–643.
37. S. Rufiange and M. J. McGuffin, "Diffani: Visualizing dynamic graphs with a hybrid of difference maps and animation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2556–2565, 2013.
38. S. Murugesan, K. Bouchard, J. Brown, M. Kiran, D. Lurie, B. Hamann, and G. H. Weber, "State-based network similarity visualization," *Information Visualization*, vol. 19, no. 2, pp. 96–113, 2020.
39. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008. [Online]. Available: http://doi.acm.org/10.1145/1355734.1355746
40. S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, "B4: Experience with a globally-deployed software WAN," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, 2013. [Online]. Available: http://doi.acm.org/10.1145/2534169.2486019
41. C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving high utilization with software-driven WAN," in *Proceedings of the ACM SIGCOMM*, 2013, pp. 15–26.
42. V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and R. Lefebvre, "Fast unfolding of communities in large networks," *J. of Statistical Mechanics: Theory and Experiment*, 2008.
43. D. Koutra, N. Shah, J. T. Vogelstein, B. Gallagher, and C. Faloutsos, "Deltacon: Principled massive-graph similarity function with attribution," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 3, pp. 1–43, 2016.
44. S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 83–86, 2006.
45. E. Dart, L. Rotman, B. Tierney, M. Hester, and J. Zurawski, "The science DMZ: A network design pattern for data-intensive science," *SC13 – The International Conference for High Performance Computing, Networking, Storage and Analysis*, p. 173 – 185, 2014.
46. M. Kiran, E. Pouyoul, A. Mercian, B. Tierney, C. Guok, and I. Monga, "Enabling intent to configure scientific networks for high performance demands," *Future Generation Computer Systems*, vol. 79, pp. 205–214, 2018.