Flamingo: Visualizing Internet Traffic

Jon Oberheide, Michael Goff, Manish Karir
Networking Research and Development
Merit Network Inc.
Ann Arbor, MI 48104 USA
{jonojono,goffm,mkarir}@merit.edu

Abstract— **In this paper we describe a set of visualization techniques that can help the task of operating and managing a network by representing network traffic information in a concise and intuitive manner. We have implemented these visualization techniques in Flamingo, a software tool that can be used to explore Internet traffic flow data. Flamingo is able to process live Netflow data in real-time and present a set of interactive visualizations and associated manipulation tools that can help users in network data analysis. Flamingo is comprised of a server and a client component. The Flamingo server is responsible for receiving raw Netflow feeds from devices in the network that can sample traffic, and then sending processed information to the client for display. The Flamingo client receives data from the server and provides concise intuitive data visualizations, 3D space navigation, as well as filtering capabilities that can help the operator to extract or monitor specific information of interest. Flamingo also supports a playback mode which allows users to select specific historical Netflow data for visualization. We illustrate with the help of simple examples, based on traffic data from a busy Internet backbone router, how Flamingo can be used to perform network monitoring tasks as well as network security related data forensics.**

I. Introduction

Visualizing Internet traffic data is a challenging problem. It is difficult to graphically represent large amounts of information and at the same time be able provide sufficient level of detail for the visual representation to be meaningful. In this paper we describe our attempts to tackle two related problems. The first is to be able to develop an effective visual representation of Internet traffic data. The second is to be able to provide various data manipulation controls that can allow the user to filter the represented information and extract specific details of interest.

In order to tackle the difficult task of effectively representing complex multi-dimensional Internet data, we provide a set of five different types of data visualizations, each of which presents a different view of the same dataset. All the visualizations are based on extensions of a basic quad-tree based algorithm that can map IP addresses and prefixes into a fixed size two dimensional square grid. We make creative use of additional dimensions in order to build the visualizations described above. We overcome the problem of representing multi-dimensional Internet data by presenting it in a set of relatively simple visualizations versus attempting to combine them into a single complex display.

The Flamingo server/client system implements our visualization techniques. The Flamingo server is able to receive live Internet traffic summary data in the form of Netflow feeds. It then performs some basic data manipulation and accounting before exporting information to the client for display. A single Flamingo server can receive Netflow data from multiple sources and present them to attached clients. The Flamingo client receives processed data from the server, and displays it in an intuitive graphical user interface. In addition to the data visualizations, the client also provides various capabilities to help users filter out extraneous information and to quickly extract relevant information via simple slider bar controls.

The rest of this paper is organized as follows. Section II describes in detail our Internet traffic visualization techniques. In section III we provide some details of how these are implemented in Flamingo, as well as the various controls available to users that allow them to explore the data representations. In section IV we provide some simple examples that illustrate potential uses of Flamingo, in exploring and examining in detail traffic patterns in a network, network security tasks, as well as in identifying anomalous events. Section V provides a brief overview of some related work, and finally section VI outlines our conclusions as well as our plans for extending our current implementation.

II. Internet Traffic Visualizations

Effectively visualizing Internet traffic data is an extremely challenging problem. One of the primary causes of this difficulty is the fact that IPv4 addresses are 32 bits in size. This implies that there are 2^{32} total unique IP addresses each of which may need to be represented. In

addition, Internet traffic is generally composed of flows which consist of at the very least, source IP address, destination IP address, source port and destination port. Netflow [1] is a popular mechanism for obtaining information about network traffic. Netflow (version 5) records consist of IP addresses, source and destination port numbers, peering AS numbers, as well as protocol numbers. Representing these in a systematic visualization without losing valuable details is difficult. We have to make creative use of various techniques, color, size, and three dimensional views. It is also important to be able to perform meaningful aggregations where possible, and at the same time be able to represent data with sufficient fidelity. We have developed a sequence of visualizations that together are able to provide relevant details extracted from Netflow samples of Internet traffic at a particular site. In the following sections we first provide a brief description of our extended quad-tree algorithm and then various describe each of our visualizations in detail.

A. Quad-Tree based IP Address Space Visualization

The basic quad-tree based visualization algorithm has been presented in literature before [2], where it was applied to represent BGP routing information. The underlying algorithm for a quad-tree representation of data is quite simple. A square grid is divided into 4 equal parts. The top-left is assigned the bit value 00, the bottom-right the bit value 11, the top-right is assigned the bit value 01, and finally the bottom-left is given the values 10. In this way any 2 bit binary quantity can be represented by simply coloring the representative quadrant. For each quadrant, we can repeat the algorithm to allow us to represent another 2 bit value. In this way we can encode a 4 bit binary value by coloring the appropriate sub-quadrant. Repeating this algorithm 16 times with increasingly smaller sub-squares allows us to represent a 32-bit IP address within a single larger square.

While the basic technique is simple, there are several practical issues that limit the use of this technique. The first is that in order to represent a 32-bit value, we have to repeat the algorithm 16 times. This often results in a display which is very small in size and therefore hard to discern visually. In addition, with two dimensions already used to represent a single 32-bit value, it is difficult to represent additional data. A prefix such at 10.0.0.0/8 can be visualized by selecting the first and the last IP address of that prefix which will give us the top-left and bottom-right corners of a square/rectangle which represents the entire prefix.

We address the display resolution issue, by adding the ability to zoom in/out to various portions of the visualization. This allows users to examine even very small representations. More importantly, in order to represent data that has multiple dimensions, we extend the basic two dimensional visualization method into the third dimension by using a cube-based 3 dimensional frame. Any single side of the cube can be used to represent the two dimensional quad-tree IP address space, but this still leaves us some some additional dimensions to represent additional information. Aside from the spatial elements at our disposal, color, and line thickness also give us other ways to encode information visually. In the following subsections, we describe how we are able to combine all these methods in order to effectively visualize Internet traffic. We describe some simple visualizations first, gradually moving on to more complex displays.

B. Traffic Volume by IP Address or Prefix

The most basic visualization of interest to network operators is one that depicts the volume of traffic from or to various IP addresses or prefixes. Each Netflow [1] record contains 32 bit source and destination IP addresses, as well as a count of the number of flows that were seen. While it would be straightforward to represent the full addresses, it would be difficult to visually discern such small representations if data for the entire 32 bit IP address space was being visualized. Therefore, we provide users the ability to view data in aggregated or non-aggregated mode. Aggregation mode is beneficial when viewing data from a network backbone which is routing a large amount of traffic from a diverse IP address space, while non-aggregating mode is preferable when a small network subnet is being examined. Here we only describe the visualization of traffic in aggregation mode, later in section III.B we describe the non-aggregation mode.

In this visualization type, a recent copy of an Internet routing table is used and the source and destination IP addresses are simply mapped to the corresponding prefix via a longest prefix match algorithm. We then keep statistics for that prefix that track the traffic volume for that prefix over a configurable interval of time. At the end of the time interval we display the {prefix,count} pairs of values. The prefix is displayed on the base (x and y axis) of a three dimensional cube by using the quad-tree mapping algorithm described earlier. The size of the square depicts the prefix length. The flow count value associated with this prefix is used to determine the height of the bar that is drawn on the z-axis. Figure 1 shows an example where this technique is used to display traffic volume on the basis of source IP addresses. A unique

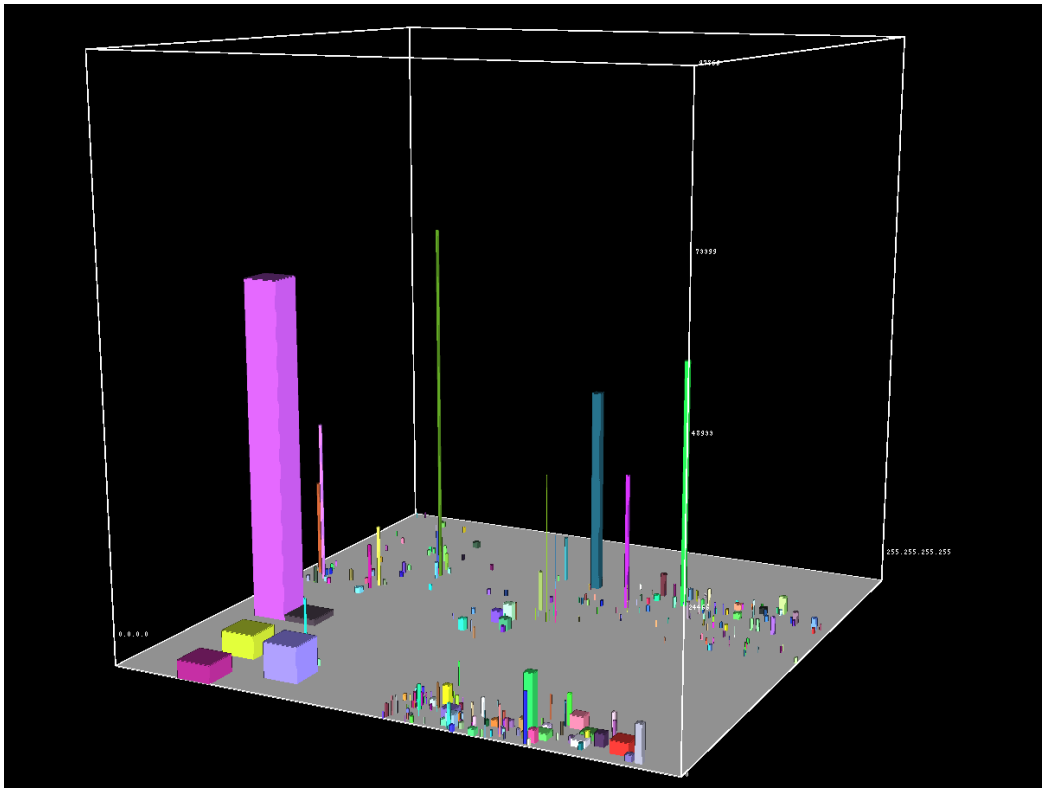

Fig. 1. Traffic Volume by Source IP Prefix: Each bar represents a prefix in the BGP routing table; height represents the volume of traffic over a configurable interval of time

color is used to represent each prefix. One can easily identify hosts or networks that are generating relatively large amounts of traffic by simply looking for the largest vertical bars. Similar to the representation in Figure 1, we can also generate an identical representation of traffic volume by destination prefix, by using destination IP addresses instead of source addresses in our mapping.

C. Traffic Volume by Peer AS

In addition to source and destination IP address information, Netflow records also contain information regarding source and destination AS peers from which the traffic was received. This information can either represent the AS number of the immediate next-hop peer from which traffic was received, or the origin AS number to which the source and destination IP addresses belong. Per peer AS traffic volume information can be useful in understanding traffic distributions across multiple peers at a peering point. Unlike an IP address, an AS number is a 16 bit value, therefore there are a total of 65K different AS numbers possible.

Once again for our visualization we use a technique similar to the one described in the previous section. However, instead of mapping 32-bit IP prefixes onto the base of a cube, we map 16-bit AS number values. We only need to apply the quad-tree algorithm 8 times resulting in a visual representation where even a single AS is adequately visible on most medium resolution displays. We then use the z-axis or height to indicate the volume of traffic to or from that AS. As the resulting representation is similar in nature to the one shown in 1 we omit it here due to space considerations.

D. Source/Destination Port Traffic

In addition to traffic volume information, source and destination ports being used also provide valuable information. In order to visually represent this information, we once again start by representing aggregated or non-aggregated IP address information on the base of a cube. In this visualization, the z-axis is used to represent destination port numbers from 1 - 65536. In order to represent port information, we reproduce the quad-tree generated IP address representation at various heights for each port number where there was traffic. For example, if prefix 192.168.0.0/16 is seen generating traffic on ports 42, 80, 135, and 137, we will draw 4 representations of that prefix at different planes on the z-axis where each plane corresponds to a single port. Volume information regarding traffic destined for a particular port number is represented by using text labels next to each square.

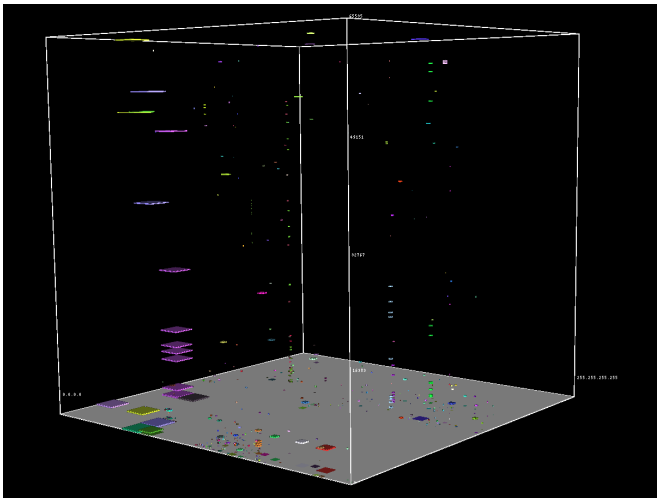

Fig. 2. Destination Port information per Prefix: The IP address representation is reproduced at various heights, to indicate traffic at different ports

An example of this technique is shown in Figure 2. We can easily spot source or destination ports that are being by traffic on the network. In particular this visualization can be useful for identifying if there is traffic originating from or destined to particular ports of interest. It should be noted that due to the complex nature of information that is represented in this compact visualization it is important to be able to effectively navigate the 3 dimensional space. This will allow a user to take full advantage of this method to explore all the data being displayed and extract or monitor relevant information. We describe these navigation controls later in this paper.

E. Source IP - Destination IP Traffic Flows

Information regarding traffic flows between source and destination prefixes at a given monitoring site is extremely important. This information can help network operators diagnose problems and optimize their networks. In order to represent traffic flows between source and destination IP addresses or prefixes, we utilize the two inside surfaces of a cube. The left surface is used to represent the source of the flow, and the right side is used to represent destination. In aggregation mode, information extracted from Netflow samples regarding source and destination IP addresses is aggregated into IP address prefixes as seen from a recent Internet routing table. We then draw lines between source and destination IP prefixes on the two planes of the cube to indicate traffic between them. The thickness of the connecting lines indicates the relative volume of traffic. The color of the line is chosen to be the same as the color of the source prefix.

Figure 3 shows the resulting visualization based on data from an Internet backbone router. It is easy to visually identify large flows almost immediately in this representation. It also illustrates which hosts or networks the traffic is originating as well as where it is destined to. This visualization summarizes a large amount of data into a single image, and allows us to gain an understanding of the overall Internet traffic pattern at a particular site. In the next section we describe the tools that allow users to further dissect this data representation.

F. Source IP,Port - Destination IP,Port Traffic Flows

In this representation we combine portions of two previous visualization methods into a single comprehensive representation that displays information about the quintuple {source IP address, source port, destination IP address, destination port}. The source/destination addresses are represented together on a single plane, at the bottom of a cube. The source/destination port numbers are represented on the z-axis. We then draw lines to indicate various traffic flows between (source IP, source port) and (destination IP, destination Port) pairs. The thickness of the line indicates the volume of traffic between the different pairs, and line color is used to represent the direction of the flow. The color of the line is chosen to be the same as the color used to indicate the source IP address.

Figure 4 shows an example visualization based on data from a live Internet backbone router. This is an extremely powerful visualization, as it represents a large amount of data into a compact display. This visualization method can easily show various events that might be of interest to network operators such as network scans or traffic hotspots. Network scans are usually characterized by a single source IP, port and a wide range of destination ports. This manifests itself visually as a vertical ”fan” shape. In addition, large numbers or flows from or to specific prefixes, or flows with large volumes are also easily identified in this visualization. In order to extract useful information from such a complex representation, we need ways to control, filter or limit the amount of information that is displayed. We provide various slider bars to control the display. How we couple the visualization methods with simple controls to build an easy to use Internet traffic exploration tool is described in detail in the next section.

III. FLAMINGO: AN INTERNET TRAFFIC EXPLORATION TOOL

Flamingo is a unique software tool that enables Internet traffic data exploration in real-time. Flamingo

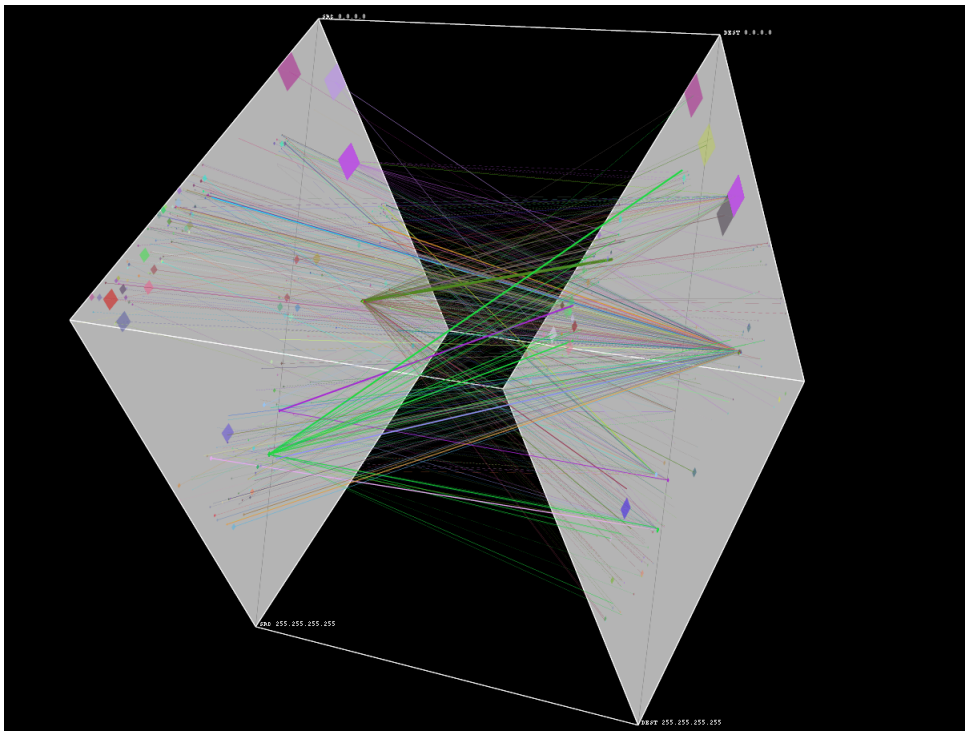

Fig. 3. Source-Destination Traffic Flows: Left plane displays source IP prefix; Right plane displays destination IP prefix; line thickness represents volume of traffic between specific source-destination pairs

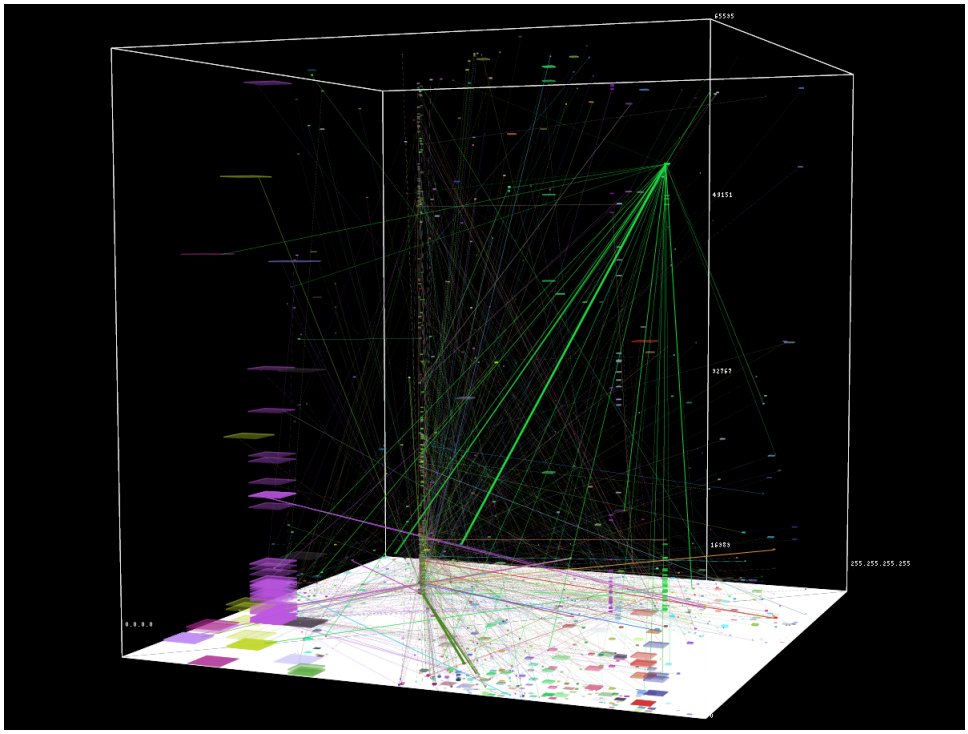

Fig. 4. Source IP, Port - Destination IP, Port Traffic Flows: Both source IP prefix and source port as well as destination IP prefix and destination port are displayed within a single 3D cube; lines are used to connect specific flows from source to destination; line thickness represents the volume of traffic for each flow over a configurable amount of time

implements the various visualizations described in the previous section in addition, it provides the necessary navigation and filtering controls to allow users to manipulate the visual representations. Flamingo is comprised of two parts, the client and the server. The Flamingo server receives Netflow feeds from various routers. This Netflow data is then processed, before it is transmitted to the client for visualization. A single Flamingo server can support multiple Netflow feeds as well as multiple Flamingo visualization clients. Each client can select to explore different types of data from the same server.

A. The Flamingo Server and Client

The Flamingo server is responsible for receiving live Netflow feeds, and processing the information into a format suitable for viewing by the clients. The server on startup reads in a current Internet routing table. This table provides basic prefix information that the server can use to perform aggregations. In aggregation mode, when a Netflow record is received, the server performs a longest prefix match on the source and destination IP addresses and updates the related statistics. In addition, the server also maintain information that maps a unique color to each prefix. Maintaining this information at the server ensures a consistent display across multiple clients. After the configured time period, the server sends the aggregated information to each client for display. The server is responsible for maintaining state about various attached clients, in order to track which visualization type they are currently subscribing to. In addition to the live mode, the Flamingo server also has the ability to playback data from stored Netflow files. It uses timestamps requested by the client to lookup the relevant files and can then stream data from those at the requested aggregation interval and playback rate.

The Flamingo client makes extensive use of OpenGL to provide high fidelity visualizations. The client simultaneously displays a three dimensional as well as a two dimensional view of the same data. The relative sizes of these can be changed via a split pane window. The two dimensional view is the basic quad-tree algorithm and is used to keep users properly oriented as they navigate the three dimensional space in the left pane. The client allows users the ability to easily navigate the three dimensional space via simple mouse controls. Left-click drag is used to rotate, middle-click drag is use to zoom in and out, and right-click drag is used to translate the cube left/right/up/down. In order to make it easy for users to interpret the data and to prevent clutter, the client allows users to enable/disable text labels in the three dimensional view. This makes it easy for users to immediately extract information about what the prefixes are, and what the bars represent. As enabling the labels adds a lot of information to the display, this is meant to be used in conjunction with the slider bar controls that are provided by the client. The slider-bar controls, enable the user to control what data is being displayed. Data that is above or below certain volume thresholds, or not in the requested port range can be dropped from the display. In addition to the graphical display, a sorted list of the data being displayed is also maintained in a separate information window, and customized watch lists can be created to monitor specific items over time.

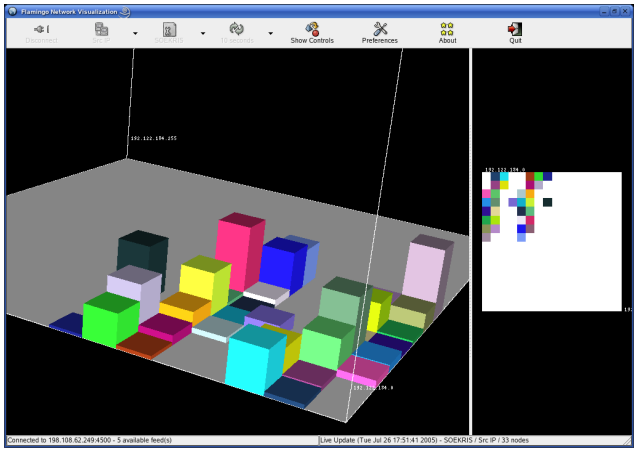

Fig. 5. Source Traffic Volume from Individual Hosts: Each bar represents a single IP address; the height represents traffic volume

Upon startup the client requests a list of available feeds from the server. It also sends to the server, the aggregation mode, and the source and destination IP address range that the user has selected. The server then only sends relevant information to the client. The IP address range, data feed source, and aggregation mode can be changed at anytime by the client. The ability to switch between aggregated and non-aggregated mode as well as the navigation controls and slider-bars, allows the user to rapidly zero in on potentially interesting data.

B. Aggregated versus Non-Aggregated Modes

Most of the examples we have presented in the previous sections have been from the perspective of a large network operator which is observing a large number of flows from a large number of hosts at a Internet backbone router. However, Flamingo is not limited in its use to only presenting aggregated statistics. The Flamingo server also implements a non-aggregation mode which can be extremely useful in order to visualize detailed data for a specific smaller subnet. In this mode The Flamingo server does not perform any aggregation on Netflow feeds that it receives, instead it collects statistics

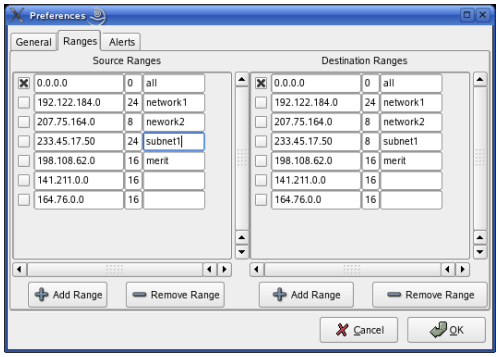

Fig. 6. Preferences Dialog Box: Selecting IP address space of interest

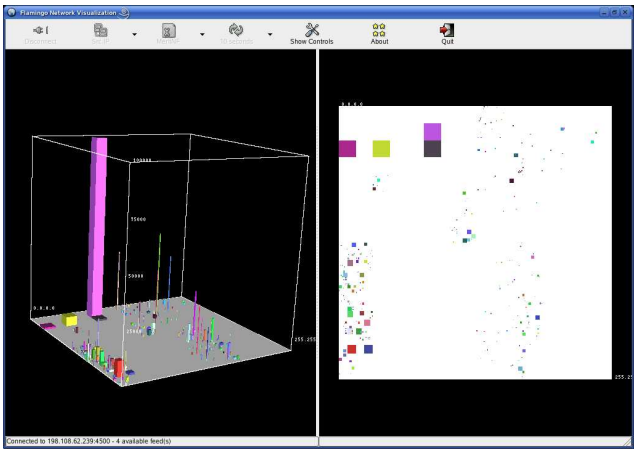

Fig. 7. Traffic Volume Aggregated by Source IP Prefix: Left window displays the 3D representation; right window displays the corresponding 2D representation

about individual IP addresses. The client then proceeds to display these full 32-bit IP addresses. In this mode individual hosts appear as equally sized squares.

Flamingo allows users to specify the IP address ranges that they are interested in. In the preceding examples, we have examined the entire 32-bit IP address space. In non-aggregation mode, it is often preferable to limit the visualization to a smaller portion of the address space, so that we can discern more detail. Figure 6 shows how the address ranges of interest can be specified.

Figure 5 shows an example where the Flamingo server has been configured in non-aggregation mode for a Netflow feed originating from a class C (/24) subnet. The figure shows the volume of traffic generated by each host in the subnet. The particular subnet that was being monitored only contained roughly 30 hosts. As expected traffic is seen originating from hosts with IP addresses at the beginning of the subnet. The two dimensional visualization in the right pane clearly shows this distribution. Similar to the visualizations described above, we can easily explore data from this subnet and extract information regarding traffic volume, port information, and individual flows. A typical usage scenario for this would be as follows; a user notices an unusually large amount of traffic originating from an particular IP prefix, Flamingo can then be used in non-aggregating mode to inspect the details of individual hosts within that prefix.

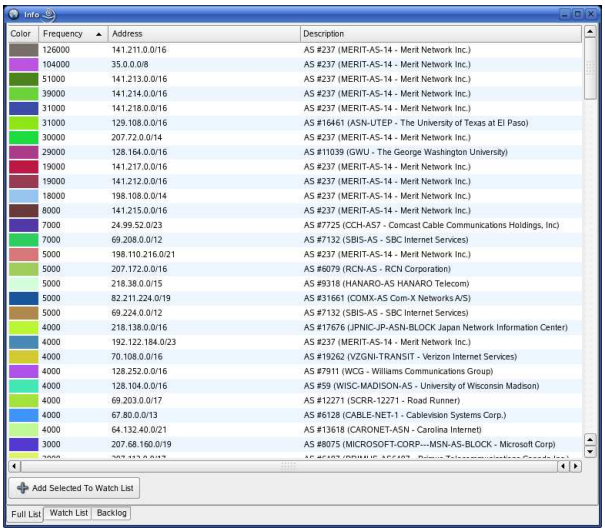

Fig. 8. Detailed Information Window: This window is used to display text entries that are being visualized; the color mapping, volume information, and description are also displayed

IV. Flamingo Case Studies and Usage Scenarios

A. Case Study: Exploring Network Traffic Patterns

In this section we describe how Flamingo can be used to obtain estimates of traffic patterns in a network, and to examine Netflow data in detail. We examine live sampled Netflow data from a busy Internet backbone router. Figure 7 shows an example of instantaneous traffic volume summary by source IP prefix. The data

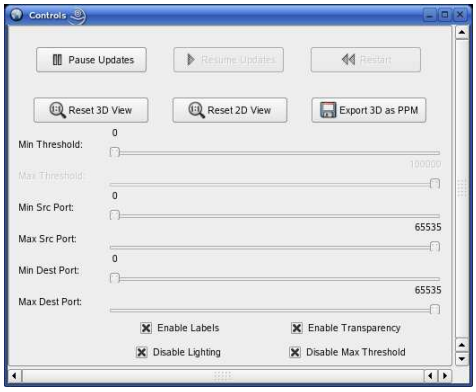

Fig. 9. Flamingo Visualization Controls: Slider bars for traffic volume, source port, and destination port ranges; various radio buttons to enable/disable text labels, transparency, and lighting; buttons to pause/resume data visualization, and to export the current visualization to high resolution image file

aggregation interval in this case was ten seconds. The figure provides an example of the flamingo client display, which shows both the three dimensional as well as the two dimensional views side-by-side. We can pan and zoom to adjust the display in both views, however, the two dimensional view cannot be rotated. This is used to help keep the users oriented as they navigate (zoom/rotate/translate) the three dimensional view. The height of the bars represents traffic volume being originated by different prefixes. Each prefix is represented by a different color, and the width of the bars represents the length of the prefix. The figure clearly shows that traffic volume is being dominated by traffic from one particular prefix. In addition to the visualization window, Flamingo can also be used to examine a sorted list of the information being displayed in a separate window. Figure 8 shows this listing sorted by decreasing traffic volume for various source IP prefixes. The list shows us the color used to represent the prefix, the volume of traffic, as well as the origin AS to which this prefix belongs.

The data filtering controls are an extremely important component of Flamingo. Figure 9 shows the controls that are available to users to allow them to control the information that is being displayed. Using slider bars, users can filter out flows that are below a certain threshold, or that are above a threshold. Similarly for port based visualizations, the source and destination min/max slider bars can be used to filter out flows that do not meet the display criteria. In addition, the control panel provides users with the ability to pause and resume updates as well as the ability to save the current visualization to a high-resolution image file for future viewing.

In addition to basic traffic volume summaries, Flamingo can be used to examine detail information regarding specific flows in the network. Figure 10 shows such an example. The Flamingo client was used to view source-destination IP address visualization of Netflow data similar to the representation in Figure 3. The live visualization was then paused, the slider bars were used to eliminate non-essential data (small flows were eliminated via the threshold slider bar control), and the three dimensional visualization was manipulated to display a zoomed in image of the the side of the cube that represents source IP prefixes. Text labels were then enabled in order to display the prefixes that were originating traffic flows. The resulting display is shown in Figure 10.

The ability to examine traffic for specific ports is also of importance to network operators, because traffic on certain ports might have some security implications. Figure 11 shows a zoomed in image where the Flamingo

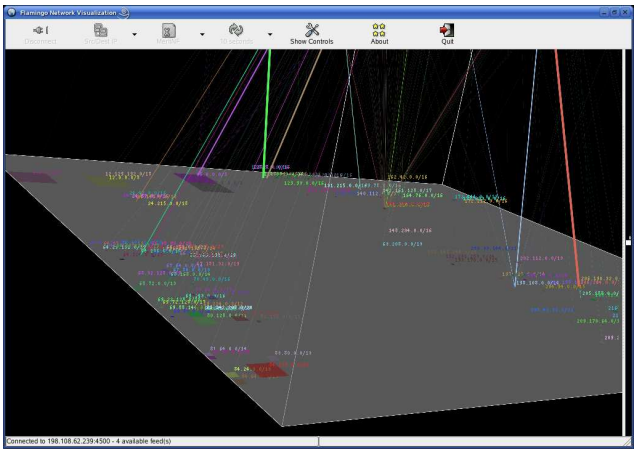

Fig. 10. Exploring Specific Flows in the Network: Using the navigation and filter capabilities we are able to examine traffic flows originating from various source IP prefixes

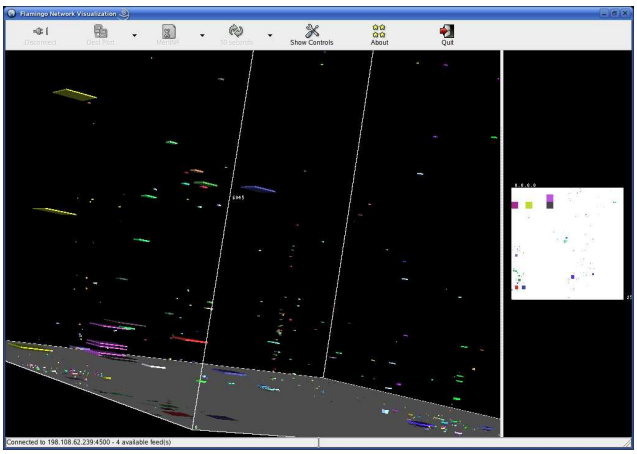

Fig. 11. Traffic Distribution by Destination Port: 3D Navigation allows us to closely examine traffic on different ports, as represented on the z-axis

controls were used to navigate the three dimensional visualization of destination port information. Text labels identify the prefix, the port and the volume of traffic. For each prefix and port the volume of traffic is shown in three dimensional space allowing users to examine specific data points of interest. The unique feature of this visualization is that the entire dataset is represented in a single display, allowing users to visually identify elements of interest. This can potentially illuminate zero-day security events where the attack data profile might not be known beforehand. We address this capability of Flamingo in more detail in the next section.

B. *Case Study: Visual Security Analysis of Network Data*

Flamingo can be an be an extremely useful aid in detecting and monitoring network security events. This analysis includes not only examining live events, but also

the ability to perform basic forensic analysis by playing back stored Netflow data from specific time intervals. Flamingo lets users examine suspect network flows, trace their target and origin, as well as look for patterns over time and IP address space in order to gain rapid insight into their potential magnitude and scope.

One concrete example is the traffic flows related with the Blaster worm [3]. The Blaster worm first appeared in August 2003. It used TCP Port 135 to propagate, and even though traffic from this worm has diminished greatly, we continue to see suspect flows from un-patched hosts. Flamingo can be used to monitor destination port 135 activity on any Netflow enabled network. It can be used to trace suspect flows from or to a particular prefix, or from particular hosts within a subnet based on the use of aggregation or non-aggregation mode for that Netflow feed. Figure 12 shows an example where Flamingo was used to monitor, in real-time, traffic passing a network backbone router for flows whose destination was TCP port 135. The figure shows visually, potential worm propagation attempts. Detailed exploration of the data can reveal the source and destination end points of these suspect flows if further action is needed.

An interesting example where the visual analysis technique is particularly useful, is in detecting network scanning activity. Figure 14 provides an example where Flamingo was used to monitor traffic with destination port 42. Destination port 42 traffic has been considered of potential interest for security purposes [4] as it has been associated with a remotely exploitable buffer overflow vulnerability in the Windows Internet Naming Service (WINS). The figure illustrates how a scan event, attempting to detect vulnerable target hosts running this service, can be easily detected using the source IP, source port - destination IP, destination port visualization method described in section II.F. In this case the scan event is represented by flows with the same source and destination prefixes as well as the same destination port but with different source ports, thereby giving it a fan-like appearance in the figure. Figure 13 shows an example of a traffic anomaly amongst other normal network traffic The fan-shaped traffic anomaly is easily spotted and consists of a single source prefix and port, and is destined towards 2 distinct prefixes on a wide range of destination ports, most likely a Denial of Service attack.

The Dabber worm [5] is an example of a worm that uses multiple ports. It uses ports 9898 as well as port 5554. The first port is used to listen for data and provides attackers with a system-level remote shell on the infected system. Port 5554 is used to identify and scan for other potential targets. Figure 15 illustrates an example

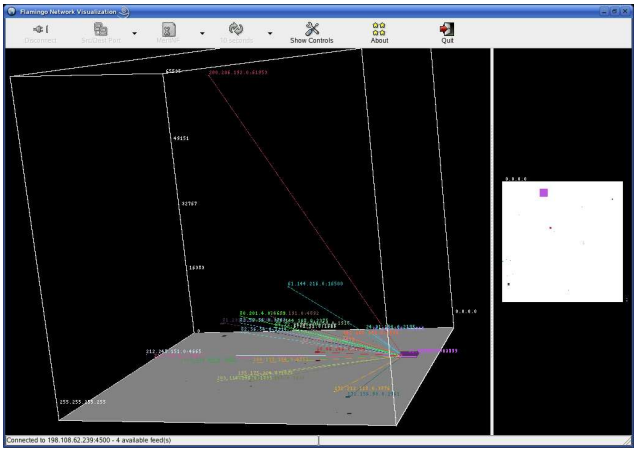

Fig. 12. Port 135 Traffic: Slider bar controls are used to only display traffic flows with destination port 135; the colors of the lines indicates that these were all incoming flows into a single IP prefix from a wide variety of originating prefixes

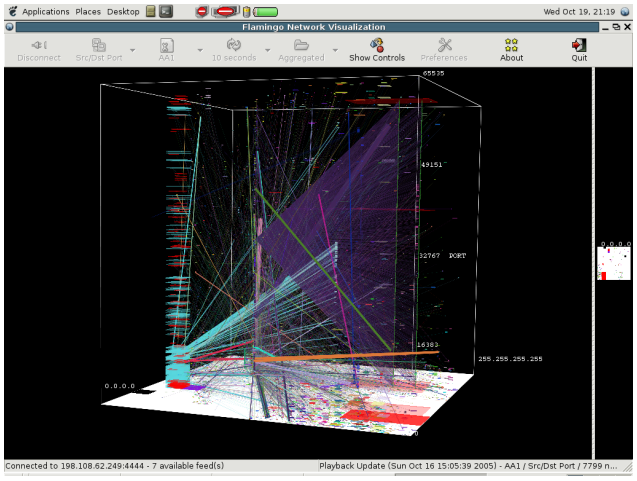

Fig. 13. Traffic Anomaly: This figure shows a traffic anomaly amongst other live network traffic; the anomaly appears visually in the shape of a triangular fan.

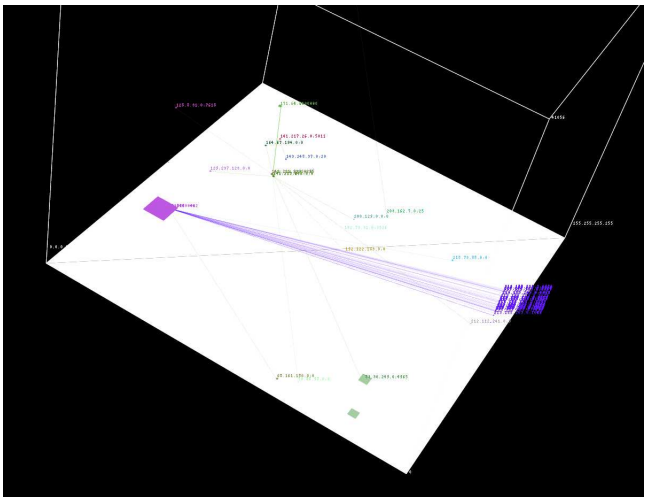

Fig. 14. Port 42 Scans: The scan shown in this figure appears to originate from a single IP prefix and is targeted at a single specific prefix

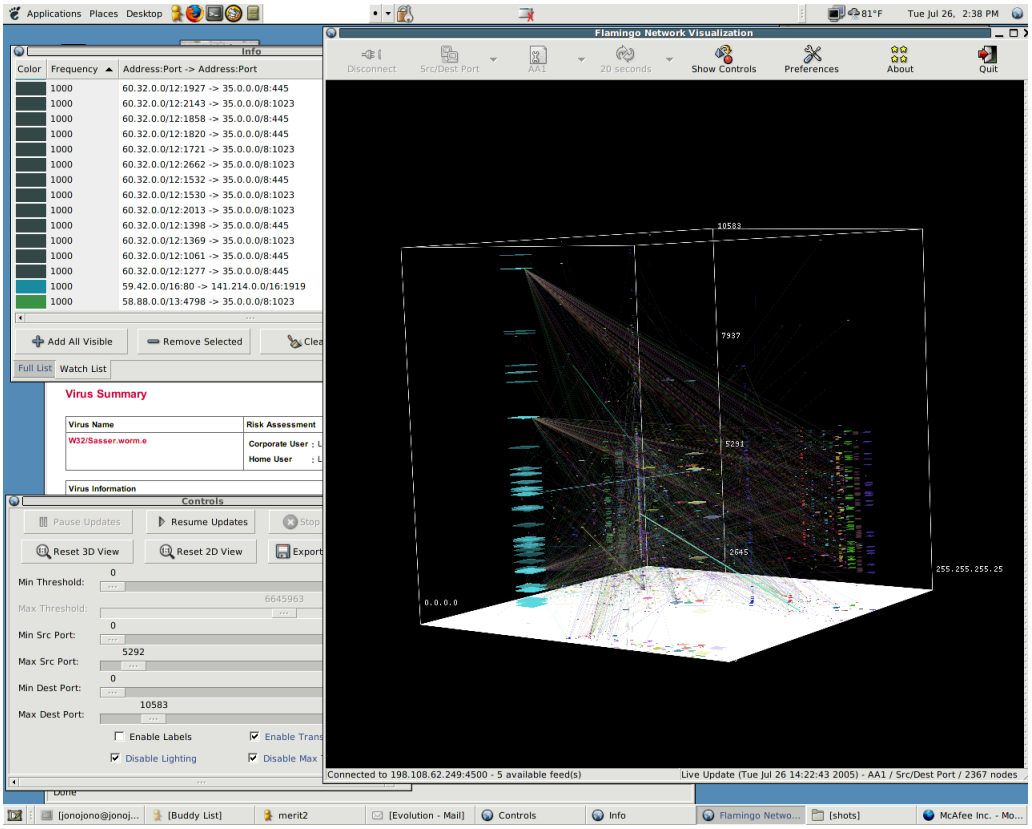

Fig. 15. Dabber Worm Traffic: The main Flamingo window displays potentially interesting traffic pattern, a large number of incoming flows targeted at 2 specific ports within an IP prefix; in the background we see Flamingo slider bar controls as well as the detailed information window

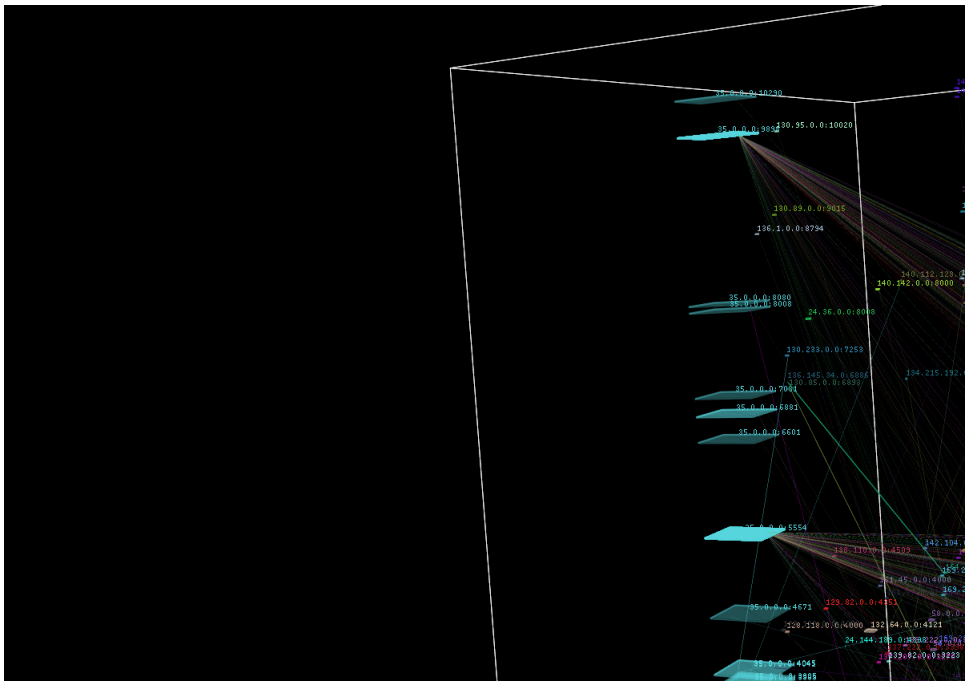

Fig. 16. Dabber Worm Traffic: A closer examination of the traffic pattern in the above figure reveals the specific target ports within the target IP prefix, which in turn indicates potential Dabber Worm activity

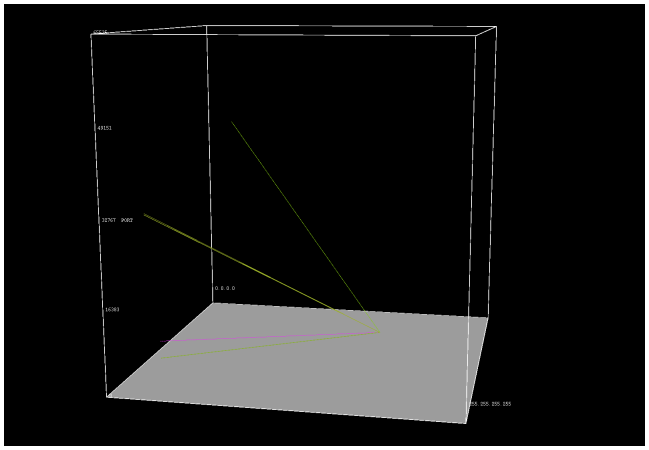

Fig. 17. Before Slashdot Event: number of flows per minute directed at a particular web-server

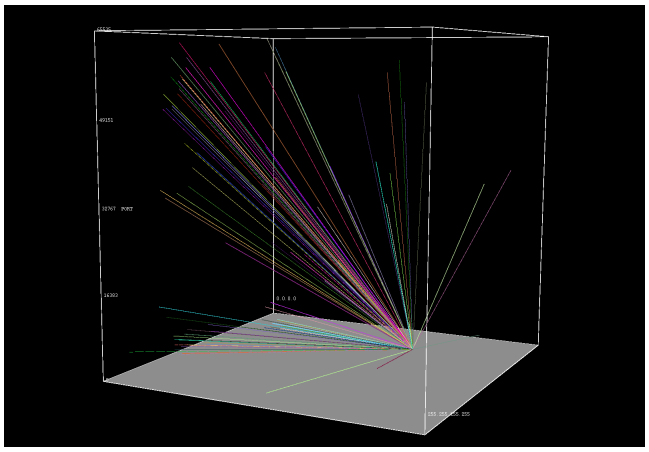

Fig. 18. During Slashdot Event: number of flows per minute directed at a particular web-server

where Flamingo was used to observe network flows with destination ports ranging from 0-10,000. Observing this range clearly captured evidence of Dabber infections probes in the network traffic. Figure 15 shows a large number of flows converging towards two specific ports for a single IP address prefix. Figure 16 shows how closer examination of this traffic pattern reveals that this burst of activity is on ports 9898 and port 5554, thereby identifying it as being related with the Dabber worm and also illustrating the source and destination prefixes.

C. Case Study: The Slashdot Effect

The slashdot.org website hosts one of the most popular discussion forums on topics of technical interest. It is viewed on a regular basis by an extremely large number of readers. One of the drawbacks of this popularity is that it has become a vehicle of unintentional DDoS-like attacks. Any URL that gets posted on this site results in a very large and rapid increase in traffic directed at that website. Very often this results in the listed website being unusable for a certain duration of time. This has commonly come to be known as the Slashdot Effect.

We were able to obtain a dataset containing traffic samples for one such event that occurred on October 31, 2004. Using Flamingo we were able to analyze this data. We used Flamingo in non-aggregating mode to view traffic directed at hosts within the particular /24 subnet where we knew the target web-server was located. We also limited our view to only include traffic directed at port 80. Figures 17 and 18 show visually the impact of slashdot effect on the particular web-server. Normal traffic for this fairly nondescript web-server is roughly 2-3 flows per minute, however, during the slashdot event this increases several fold and is easily discernable visually as an anomalous event. Other views of the same dataset illustrate a corresponding increase in the volume of traffic being sent to that specific host. The time at which this sudden increase in traffic occurs coincides very closely to the time at which the URL was posted on Slashdot. This example shows how Flamingo can be used not only for live data analysis, but also for network forensics after an event has already occurred.

V. Related Work

Recently there has been significant interest in the area of network data visualization. Visualization techniques and methods are increasingly being applied to help manage, monitor, and secure networks. However, network data visualization, both routing as well as traffic data, continues to be a challenging problem.

SeeNet [6] was one of the earliest attempts at visualizing network data. SeeNet proposed the use of matrix displays to illustrate data between any 2 nodes in a network. It also provide simple data filtering and manipulation capabilities as well as the use of zoom and color to allow users to control the amount of data being displayed. Several of these basic techniques have been adapted for use by successive tools over time, including Flamingo.

The basic quad-tree algorithm described in this paper has been applied in the past to help visualize BGP routing data anomalies. [2] and [7] proposed a visual technique to detect anomalies in BGP routing information by animating BGP datasets. [8] used a similar technique to display IP address space usage in order to determine IP address space delegation approximations. Flamingo is based on the same basic technique, but extends it in order to be able to display and explore multi-dimensional traffic data in real-time.

[9] uses a three dimensional method in order to visualize intrusion detection logs. They do not use a

quad-tree based approach, instead they list monitored address space on one axis, and the entire Internet address space on another, using the third axis to represent port information. Unlike Flamingo, this tool could only present a single visualization, and could not be used for live Netflow information exploration. Moreover, this tool lacks the ability to switch between aggregation and non-aggregation modes which makes Flamingo much more flexible.

VisFlowConnect [10] and NVisionIP[11] are prototype applications developed at the National Center for Supercomputing Applications(NCSA) to visualize Netflow data. While VisFlowConnect, NVisionIP and Flamingo share the same goals, the approach taken by each one is quite different. VisFlowConnect focuses on representing the data in a two dimensional plane by using the left and right axis to represent source and destination IP addresses. It is limited by its inability to display the entire 32-bit IP address space and instead displays smaller "domains". NVisionIP is a tool that uses a series of three increasingly detailed graphical representations to display data about a class B network. Flamingo is unique in its use of a three dimensional modified quad-tree based algorithm to represent the complete IP address space. A user can navigate this space in order to locate data points of particular interest. Different visualizations within Flamingo display different parameters of the network data being visualized.

PortVis [12] is a tool that can be used to analyze network events based entirely on port level information. The vertical axis is used to represent time and the horizontal axis is used to represent network traffic port range. Flamingo is able to easily incorporate port level information within its visualization suite.

VI. Conclusions and Future Work

In this paper we have described a set of novel visualization methods for Internet traffic data. A basic quad-tree based algorithm was extended to allow us to represent the multi-dimensional data extracted from Netflow records. We combine the visualization techniques with navigation and data filtering controls in a client-server software tool called Flamingo. We have provided some details of our implementation, as well as some illustrative case studies which serve to demonstrate how Flamingo can be used to help in network operations and management tasks. In particular we describe how Flamingo can be used to examine network traffic profiles, for visual security analysis of traffic flowing in a network, as well as for identifying anomalous events. Flamingo has been used by network operators to easily detect infected machines on a subnet. In one particular scenario for example, we were able to use it to identify the source of large bursts of DDoS attack packets. The affected host was then promptly disconnected from the network preventing it from causing any further damage.

Due to the the complex nature of network traffic data, we argue that a set of visualizations and associated manipulation tools is better able to represent the dataset, rather than a single complex visualization. Our work in implementing Flamingo also highlighted the need for further research in fast algorithms and data structures that are able to process the large volumes of Internet data in real-time. We are actively working on extending Flamingo to further enhance its usability as well as its utility in an operational environment. One feature currently under development is the ability to generate automated alerts based on user specified traffic profiles. We anticipate releasing Flamingo to the networking operator and research communities to obtain further feedback which can guide our future development.

References

[1] Cisco Systems. Cisco CNS netflow collection engine: Netflow services solutions guide. *http://www.cisco.com/go/netflow*, October 2004.

[2] S.T. Teoh, K-L. Ma, S.F. Wu, and X. Zhao. A visual technique for internet anomaly detection. *Proceedings of the 14th IFIP/IEEE Workshop on Distributed Systems: Operations and Management*, October 2003.

[3] D. Knowles, F. Perriot, and P. Szor. W32.blaster.worm. *http://securityresponse.symantec.com/avcenter/*, February 2004.

[4] E. Cooke, J. Nazario, and D. McPherson. Tcp/42 wins report. *http://ims.eecs.umich.edu/reports/index.html*, January 2005.

[5] K. Ha. Symantec security response: W32.dabber.b. *http://securityresponse.symantec.com/avcenter/*, May 2005.

[6] R.A. Becker, S. Eick, and A. Wilks. Visualizing network data. *Proceedings of IEEE Transactions on Visualization and Computer Graphics*, March 1995.

[7] R.S. Teoh, T.J. Jankun-Kelly, Kwan-Liu Ma, and S.F. Wu. Visual data analysis for detecting flaws and intruders in computer network systems. *IEEE Computer Graphics and Applications*, September 2004.

[8] P.D. McDaniel. Origin authentication in interdomain routing. *Invited Talk, University of Michigan*, October 2004.

[9] Stephen Lau. The spinning cube of potential doom. *ACM Communications Viewpoint Column*, June 2004.

[10] X. Yin, W. Yurcik, M. Treaster, Y. Li, and K. Lakkaraju. Visflowconnect: Netflow visualizations of link relationships for security situational awareness. *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security(VizSEC)*, October 2004.

[11] K. Lakkaraju, W. Yurcik, A. Lee, R Bearavolu, Y. Li, and X. Yin. Nvisionip: Netflow visualizations of system state for security situational awareness. *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security(VizSEC)*, October 2004.

[12] J. McPherson, Kwan-Liu. Ma, P. Krystosk, T. Bartoletti, and M. Christensen. Portvis: A tool for port-based detectino of security events. *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security(VizSEC)*, October 2004.