SPLAT: A visualization tool for mining Internet measurements

Joel Sommers[1], Paul Barford[1], and Walter Willinger[2]

[1] University of Wisconsin-Madison, Madison, WI, USA
[2] AT&T Labs-Research, Florham Park, NJ, USA

Abstract. Visualizations provide a natural means for organizing large complex data sets and mining them for characteristics of interest. In this paper, we describe SPLAT, a **s**catter and **p**hase p**l**ot **a**nimation **t**ool. SPLAT offers a broad set of capabilities for investigating Internet measurement data sets based on scatter and phase plots—two well-known techniques for exploratory data analysis. An important feature of SPLAT is that it can animate the scatter and phase plots over time to reveal dynamic characteristics of the data at hand. We demonstrate SPLAT's capabilities through a series of case studies that show how both general profiles and important non-obvious details in large Internet data sets can be identified thereby illustrating its utility for a diverse set of network research areas.

1 Introduction

Empirical measurements are the cornerstone of scientific discovery and evaluation of Internet structure and behavior. However, the quantity, diversity and complexity of Internet data greatly complicate the process of analysis. Many methods for assessing Internet data have been proposed and developed over the years, but most of them are quantitative in nature, suited for specific data types, and designed with a particular purpose in mind. There exist surprisingly few general-purpose tools for qualitative exploratory analysis of Internet data—an important precursor to hypothesis-driven discovery, analysis, and validation. On the other hand, standard plot-based visualization methods are common in Internet data analysis. These include plots to evaluate statistical properties of time series, scaling properties (*e.g.*, [8, 13]), and protocol behavior (*e.g.*, [6]). There are also several visualization tools which have been developed for Internet data analysis; see [2] for a partial listing. These can be classified as tools for connectivity/structural analysis (*e.g.*, [15]), network monitoring (*e.g.*, [3, 10]), and for elucidating the dynamics of particular networking protocols (*e.g.*, [5]).

Visualizations provide a natural means for exploratory analysis of large complex data sets. While there are many standard textbook references on exploratory data analysis that highlight visual methods, Tukey [20] remains one of the classics and a rich source for inspiration. The high level objectives in such analyses are to identify both interesting general patterns in the data and relevant domain-specific details in parts of the data. Standard methods for visualizing data are mainly comprised of 2D plots or graphs that may include reference information (*e.g.*, regression curves). However, the key to effective visual analysis is to present the data in ways that offer great flexibility, thereby facilitating the detection and identification of critical characteristics. This

suggests several requirements for visualization tools in the context of Internet data, including the ability to handle large, high-dimensional data sets, the flexibility to support a variety of views of the data based on Internet context, and the applicability to a variety of different, network-specific problems. In [17], Paxson also calls for visualization tools to support a range of exploratory analyses.

In this paper we describe SPLAT, a **s**catter and **p**hase p**l**ot **a**nimation **t**ool that has been developed for Internet data analysis. As the name suggests, SPLAT offers visualization capabilities based on 2D scatter and phase plots of data. We have enhanced the tool by providing a set of pruning, zooming and feature selection (*e.g.*, filtering) capabilities developed specifically for large high-dimensional Internet data analysis. A distinguishing feature of SPLAT is that it can display animations of phase and scatter plots as they evolve over time. This 3D capability greatly facilitates the discovery and identification of subtle features in the data that may reveal interesting aspects of Internet structure and behavior but would typically be overlooked when using a purely static display of the data. We are not aware of any widely-used visualization tools that have the combination of capabilities provided by SPLAT.

We demonstrate the capabilities of SPLAT in a set of case studies that consider (i) characteristics of TCP packet traffic, (ii) characteristics of Internet flows, (iii) calibration of active measurement tools, and (iv) the dynamic structure of source/destination addresses in IP traffic. In these examples, we show how the tool can be used to identify both general patterns and unique features in each of the data sets. These examples illustrate the practical attributes and benefits of this tool in a broad set of problem domains.

2 SPLAT Design and Implementation

Phase plots and scatter plots are well-known exploratory analysis tools for determining association and examining relationships among different variables. In its basic form, phase plot analysis considers two time-dependent variables $x(t), y(t)$. A phase plot is a graph of all points $x(t_i), y(t_i)$ over a specified period of time where the x-variable is plotted on the horizontal axis and the y-variable on the vertical axis. Since Internet data sets almost always have time components associated with them, they are naturally suited to phase plot analysis. Scatter plots are, in essence, identical to phase plots but do not have an implicit time axis. A time axis can be trivially added, however, to study the temporal evolution of the two variables.

The basic design requirement for SPLAT is to display a phase plot, to allow one to zoom and pan on specific regions, and to view the animations of the plot over time. In addition, SPLAT has a number of annotation capabilities, including display of relative point densities along each dimension of the plotting region, listing of the current time in the trace file (for animations), coloring of data points to highlight possible associations with higher-layer entities (*e.g.*, packets associated with flows), and labeling of these higher-level entities (*e.g.*, a string representation of the five-tuple that defines a particular flow).

For large, multivariate data sets, a key design requirement is the ability to view subsets of the main phase plot data, conditioned along one or more dimensions. To enable filtering, SPLAT can load auxiliary data sets (*e.g.*, time series data that are synchronized with the main phase plot data) and various kinds of categorical summary data (*e.g.*, estimates of flow round-trip times or flow sizes). For example, assume that our basic phase

plot data consists of spacings between individual packets of a flow as the packets *enter* a congested router queue and the spacings of the same packets as they *exit* the queue, as depicted in Figure 1(a). We could make use of time series data of the queue length to enable visual detection of correlations between phase plot features and congestion events. Similarly, we may wish to restrict our view of the ingress-egress phase plot to consider the largest flows that also have round-trip times within a certain range, or to view only phase plot data for flows having destination IP addresses matching a given prefix. These built-in capabilities of SPLAT distinguish it from more general-purpose visualization tools such as GGOBI [4] that are not designed to handle Internet-specific data sets. Additional concrete examples of some of SPLAT's filtering capabilities are described in the case studies, below.

SPLAT consists of about 7,000 lines of C++ and uses the cross-platform Trolltech Qt libraries [7] for its graphical capabilities. Plotting areas are drawn using OpenGL widgets, enabling relatively simple zoom, translation, and rotation by manipulating the world-to-screen and projection matrices. Saving a plot for later reference is handled by converting the raw frame buffer data to a common image format. All scatter plots in this paper were produced using this capability in SPLAT. SPLAT includes the ability to read a variety of common Internet-related data types.

3 Case Studies

In this section we examine a range of SPLAT's capabilities through four case studies of scatter/phase plot construction, interpretation, and filtering. Our intent is not to demonstrate deep analysis of different kinds of Internet data, but rather to demonstrate a diverse range of data exploration tasks that can be accomplished with SPLAT.

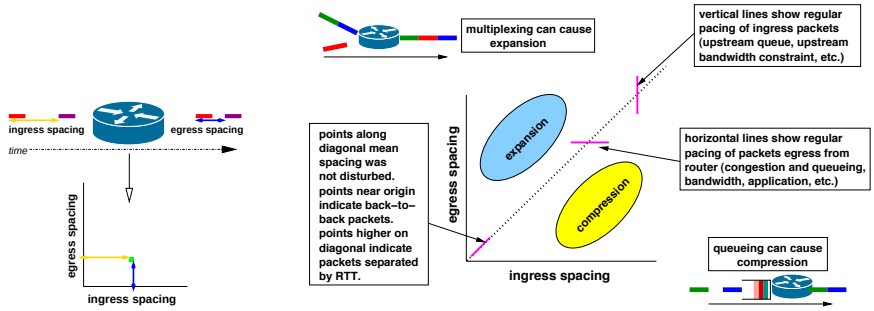
3.1 Case Study I: TCP packet traffic characteristics

Many prior studies of packet traffic behavior (*e.g.,* [13]) have been based on measurements taken at a single point in the network such as the ingress to a router. These measurements usually consist of full IP packet headers and lend themselves directly to many types of analysis, but are generally insufficient for capturing the most basic characteristics of IP networks, *i.e.*, effects of statistical multiplexing and queuing at routers. A simple extension to this basic measurement capability is to gather timestamped IP packet headers at *both* the ingress and egress of a router (or collaborating end hosts), which provides considerably more information and exposes a much wider range of traffic characteristics.

Given the capability of taking ingress and egress measurements at a router, we can consider two packets from the same source that are emitted close to one another in time. If we measure the time delay between these packets as they arrive at the router ($Ingress\_Spacing = s_i$) and again as they exit the router ($Egress\_Spacing = s_e$), then there are three possibilities for the ratio $s_r = s_e/s_i$. If $s_r = 1$ then spacing remained unchanged by the router. If $s_r > 1$ then other packets enqueued between the two packets causing *expansion*. If $s_r < 1$ then the first packet was delayed because of a queue that has diminished by the time the second packet arrives, causing *compression*. TCP ACK compression is one manifestation of this latter phenomenon[3].

[3] There are, of course, many potential causes of compression and expansion. For example, expansion might be caused by a router-induced delay that is unrelated to congestion, while both

The basic construction of a two-dimensional phase plot from packet traces is depicted in Figure 1(a). Thus far, we have defined our two measurement points as two links of a path through a router. More generally, these ingress-egress measurements can be considered at arbitrary points along a path. For example, we might collect packet traces at the end hosts of a path, constructing a phase plot that includes effects of all routers and intermediate devices along the path.



(a) Construction of a phase plot from two synchronized packet traces.

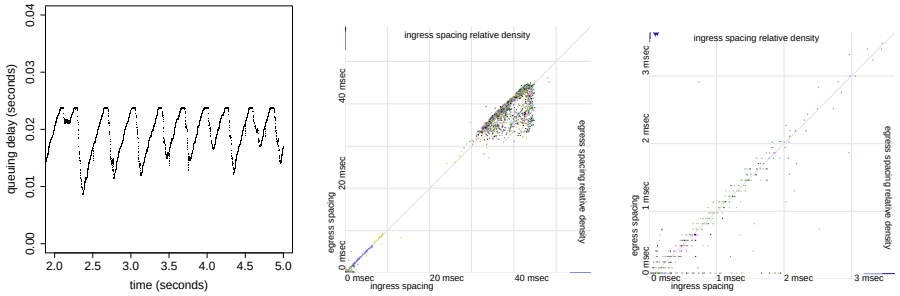
(b) Interpreting phase plot features of flow dynamics.

Fig. 1. Basic construction and interpretation of phase plots for analyzing packet traffic dynamics from measurements taken at the ingress and egress of a router.

As an example of this kind of phase plot analysis, consider the well-known synchronization behavior of long-lived TCP flows with similar round trip times [14, 21]. Reno-based TCP implementations in congestion avoidance cause the bottleneck queue to fill and drain as the sources oscillate between linear increase and multiplicative decrease in sending rate. We created a standard dumbbell topology in a laboratory environment with commodity workstations and routers and measured the traffic generated by 40 infinite TCP transfers at both ingress and egress of the bottleneck router. The bottleneck link in this setup is an OC-3 (155 Mb/s) and its interfaces are configured with drop-tail queues. A round-trip time of 20 milliseconds is configured using NetPath [9]. Figure 2(a) shows the familiar "sawtooth" characteristic of the time series of packet delays due to queuing. Figures 2(b) and 2(c) show the corresponding phase plots for the traffic at two resolutions. There are two main features in these plots: a cluster of points close to the origin, and a relatively large triangle-shaped element with a lower left corner at an ingress and egress spacing of about 30 milliseconds. From the density indications along the right and upper axes of Figure 2(b), we see that the majority of packets are sent back-to-back or closely spaced. Although the triangle-like cluster is eye-catching, most packets of each flow arrive closely spaced.

The triangle-like feature of the phase plot has meaningful dimensions and location. As the competing flows increase their congestion windows during the congestion

compression and expansion may be due to a bandwidth differential between the two measurement points.



(a) Queuing delay over a three second interval of the trace. Well-known oscillating ("sawtooth") behavior results from synchronization among TCP flows.

(b) Phase plot representation of long-lived TCP connection synchronization. Grid lines are spaced 10 milliseconds apart. Note that most volume is located at lower left (as shown by relative density indications on right and upper axes).

(c) Close-up of lower left corner of (b). Grid lines are spaced 1 millisecond apart.

Fig. 2. Phase plots and an associated queuing delay time series plot created from a laboratory testbed trace of long-lived TCP sources. Phase plot of (c) is a close-up of the lower-left corner of (b). Note that grid lines are not at the same scale for each phase plot. Vertical and horizontal lines along top and left edge of plots indicate density of points along each axis.

avoidance phase, the queue builds, the round-trip time increases, and so does the spacing between packet trains from each TCP source. This effect results in the line of points along the diagonal of the phase plot. When the queue fills and packet loss occurs, the sources drop their congestion windows, causing the queue to drain, and some packet spacings to be compressed [16, 22]. The triangle feature is approximately 15 milliseconds wide and tall, corresponding to the range of oscillation of the queue. Also, the location of the lower left of the triangle is offset by 10 milliseconds from the round-trip propagation delay of 20 milliseconds. This offset corresponds to the smallest delay measured through the congested queue, as shown in Figure 2(a).

Finally, we note that when viewing the phase plot as an animated series of consecutive time slices, the points on the triangle of Figure 2(b) appear in a clockwise manner, following the rise and fall of the queue. SPLAT can display a synchronized view of a time series plot similar to Figure 2(a) alongside the phase plot to make this connection visually apparent.

Looking at a close-up of the back-to-back spacing region in Figure 2(c), we first notice a clear discretization of egress spacings. As closely spaced packets of one flow arrive at the bottleneck link, they are multiplexed with packets from other flows and are respaced by an integral number of 1500 byte packets as they are transmitted at 155 Mb/s. The horizontal striations are separated by approximately 80 microseconds, corresponding to the transmission time of a 1500 byte packet.

In this case study, SPLAT was helpful in revealing interesting dynamic characteristics of the eye-catching triangular feature and relating those characteristics to the well-understood sawtooth queuing behavior typical for a homogeneous (but unrealistic) network environment (*e.g.*, all TCP flows are long-lived and have similar RTT). The tool

also helped to illustrate that while the observed triangular feature is distinctive, it is not the dominant characteristic.

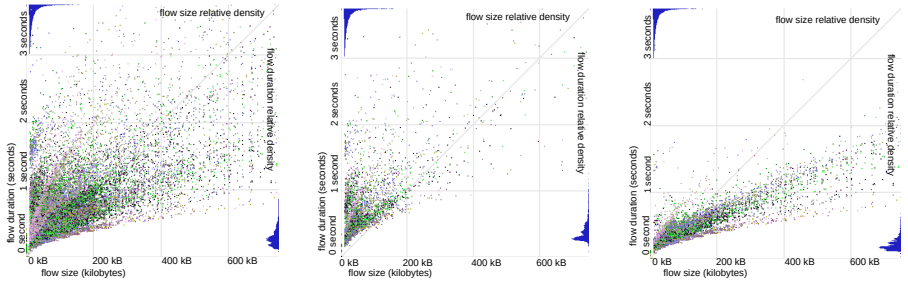
3.2 Case Study II: Flow-level Traffic Characteristics

In this case study we illustrate the dynamic characteristics of a scatter plot by comparing flow size and duration. This application of scatter plots could be useful for studies similar to Zhang *et al*. [23], in which they compared characteristics of Internet flow rates with aspects such as flow size and duration. The scatter plots in this section are formed by setting the *x*-axis to represent flow size in bytes and setting the *y*-axis to represent flow duration in seconds. A natural choice for the time axis is the starting time of the flow.

Figure 3(a) shows a scatter plot of flow sizes and durations, with a focus on small flows. The data for this example was created using the same basic laboratory testbed dumbbell topology as described in § 3.1. The background traffic for this example consists of web-like self-similar traffic created using Harpoon [18] that varies in utilization over the duration of the experiment, from about 50 Mb/s to 130 Mb/s, averaged over 1 second intervals. Propagation delays in the testbed are configured to be between 35 and 65 millseconds, with a mean of 50 milliseconds, and the queue is configured to buffer approximately 50 milliseconds of packet data. Although the general profile of the figure supports the expected correlation between flow size and duration, we also notice that there are a number of the smallest flows with significantly longer durations than most other flows. If we filter the data to show only time periods when the queue was nearly full (Figure 3(b)), we observe a shift in the data points corresponding to generally longer durations for all flows during congested time periods. If, alternatively, we restrict our view to time periods when the queue was nearly empty (Figure 3(c)), we observe a shift towards generally lower flow durations, as we might expect. Additional filtering capabilities of SPLAT, such as restricting our view to flows with RTTs within a certain range, or confining our view based on the amount of data transferred between a source/destination pair, could be used to draw out subtle characteristics of the plots and determine relevant cofactors.

3.3 Case Study III: Calibrating Available Bandwidth Estimation Tools

Available bandwidth estimation tools (ABETs) such as SPRUCE [19] and PATHLOAD [11] are designed to send packets pairs or streams at well-defined spacings (on time scales of tens to hundreds of microseconds), measure the spacings of the same packets at a receiver, and, according to a tool-specific model, infer the amount of available bandwidth along the path. Calibration, in the context of ABETs, is commonly understood to mean comparison of available bandwidth estimates with measures that have been obtained through, *e.g.*, packet traces with timestamps of sufficient quality. An alternative ABET calibration exercise involves comparing measured packet level characteristics of a probe stream with characteristics that the tool *should have produced*. Figure 4(a) depicts this kind of ABET calibration. Considering a series of packet pairs or streams emitted by an ABET in the context of a phase plot, the *x* (ingress) dimension should reveal any differences between spacings that are intended by the ABET, and the spacings



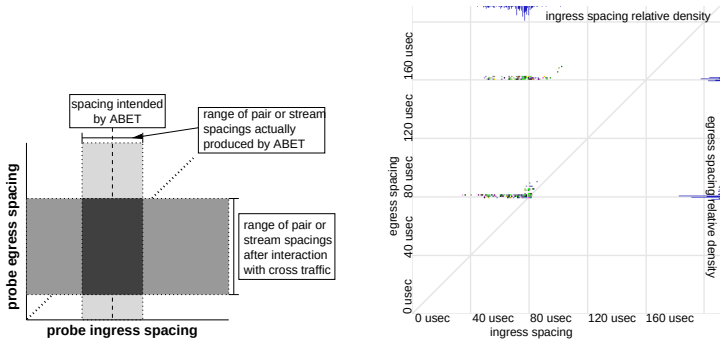
(a) Scatter plot produced from all flows during the experiment. Figure shows a zoom of the smallest flows (up to 750 kBytes).

(b) The same data as in 3(a), but filtered to show periods when the queue was within 90% of its capacity.

(c) The same data as in 3(a), but filtered to show periods when the queue was no more than 10% full.

Fig. 3. Scatter plots of flow size (*x*-axis) versus flow duration (*y*-axis) generated from traffic produced in a laboratory environment.

actually produced. This provides the ability to assess bias introduced into the measurement process by imprecise commodity hardware and operating systems. The *y* (egress) dimension of the phase plot should reveal the spacings on which inferences should be made by the receiver after interaction with cross traffic, though they may differ from the spacings actually measured by the receiver. This enables calibration of both the inference method as well as providing a baseline for calibrating the receiving host.



(a) Interpreting scatter plot features for available bandwidth estimation tool calibration.

(b) An example scatter plot created from a series of 12 SPRUCE runs (1200 packet pairs, packets of length 1500 bytes).

Fig. 4. Application of scatter plots to available bandwidth estimation tool analysis and calibration.

Figure 4(b) shows a phase plot created from a series of 12 probe streams generated by SPRUCE in a laboratory testbed. The testbed setup is based on the same dumbbell topology used in § 3.1 and 3.2. A simple constant bit-rate UDP source of 100 Mb/s was used as background traffic in this setup. Given the bottleneck bandwidth of 155

Mb/s, SPRUCE packet pairs should be configured to be separated by approximately 80 microseconds upon departure [19].

The phase plot shown in Figure 4(b) was created by collecting time-synchronized packet traces using DAG monitors before the probes entered the bottleneck router (and prior to interaction with any other traffic), and just after exiting the router. The phase plot immediately exposes two potential sources of measurement bias. First, it is easy to see that there is a wide range of interpacket spacings on ingress which can be attributed to inaccuracies introduced by the sending host. Second, it is also evident that an effect of the CBR cross traffic is to cause a respacing of probe packets on egress to either exactly back-to-back ($\approx$ 80 microseconds) or with one cross traffic packet interposed ($\approx$ 160 microseconds). Closer examination reveals that packets spaced farther apart by the ABET are more likely to experience expansion by a cross traffic packet than to be transmitted back-to-back on the tight link. This can be seen in the figure by the perceptible shift to the right in the upper cluster of points. Additional benefits of using SPLAT in this situation include the possibility for dynamic analysis of ABET bias through SPLAT animations and the potential to filter the data, *e.g.*, to focus on time segments of high intensity background traffic or rapidly changing background traffic conditions.

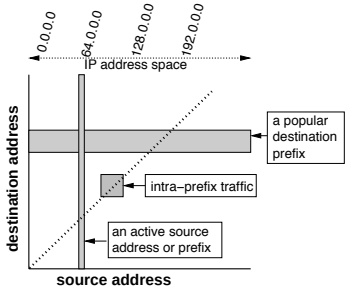
3.4 Case Study IV: Structure of Addresses in IP Traffic

In [12], Kohler *et al.* consider the structural characteristics of destination addresses in IP traffic. One way to represent these characteristics in scatter plot form is to consider the source address of a flow as the x dimension, the destination address as the y dimension, and the flow start time as the time dimension. To derive values for the x and y dimensions, IP addresses can be linearly mapped to a given scale. Figure 5(a) depicts this application of scatter plots. In their study, Kohler *et al.* observed that destination address structure was stable over short time scales as seen at a particular location, and that the observed structure can make a practical fingerprint of the aggregate traffic seen at a given vantage point.

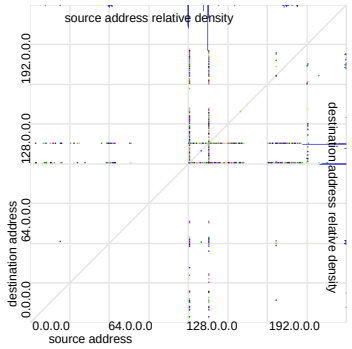
Figure 5(b) shows an IP source/destination address scatter plot created from one hour of flow records collected at a border router of the University of Wisconsin-Madison. The two dominant lines for both source and destination addresses in the plot indicate that most traffic is sourced or sinked by the University's two class B address spaces. The less dominant characteristics are also interesting. For example, a data point in the lower left quadrant indicates a traffic flow between two networks in the traditional class A address range. Also, a set of points in the upper right quadrant indicates one or more multicast sources with a large set of destinations in the class B and C range.

Figure 5(c) shows another IP source/destination address scatter plot created from one hour of flow records collected at the Houston, TX router of the Abilene network [1]. The basic features of this plot compared with Figure 5(b) show the effect of connectivity and perspective of a router on the addresses in traffic observed at that router. Figure 5(d) is produced from the same underlying data as Figure 5(c), but has been filtered to show only points representing source and destination pairs that transferred more than 1 MByte. From the relatively density indications along the upper and right sides of the plot, we see that there are indeed very few source/destination pairs responsible for most of the traffic. We also see a number of potentially interesting spatial outliers, *e.g.*, in the lower right quadrant.

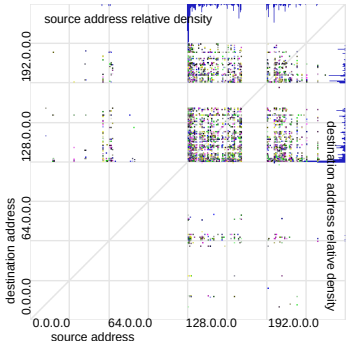
This case study suggests that SPLAT can be useful for examining spatial fingerprints or signatures of traffic flows and for helping to understand some of the causes behind these signatures. The animation capabilities of SPLAT make it well-suited as a tool for qualitatively assessing the stability of these signatures over a range of time scales, and also for helping to uncover causes behind possible changes to a signature.



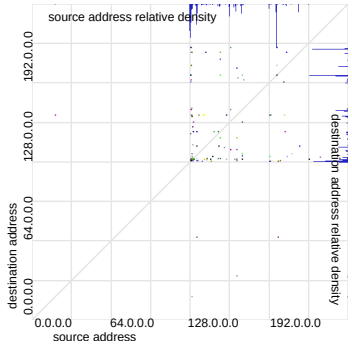
(a) Construction and basic interpretation of an IP source/destination scatter plot



(b) Source/destination scatter plot constructed from flow records collected at a border router of the University of Wisconsin-Madison.



(c) Source/destination scatter plot constructed from flow records produced by the Houston, TX router of the Abilene network.



(d) The same scatter plot as in (c), but filtered to show only source/destination pairs that transferred more than 1 MByte.

Fig. 5. Application of scatter plots to analyzing the structure of source and destination addresses in IP traffic.

4 Summary and Conclusions

The description and illustrations of SPLAT provided in this paper highlight the effectiveness of this simple visualization tool as a means for mining very different types of data sets, collected from either the real Internet, from laboratory testbeds with commodity hardware and software, or from commonly-used network simulation environments (not considered here). The measurements can have a temporal component (*e.g.*,

ingress/egress spacings), a spatial component (*e.g.*, IP addresses), some other network-specific component (*e.g.*, flow attributes), or any combination thereof, and they can be very fine-grained (*e.g.*, packets, IP addresses) or more coarse-grained (*e.g.*, flows, prefixes). SPLAT offers a basic set of useful capabilities for mining and exploring these voluminous and semantically rich data sets, and we have illustrated its utility for exploring such diverse and challenging problems as assessing traffic dynamics (case studies 1 and 2), calibrating active measurement tools (case study 3), or the potential for detecting/identifying network traffic anomalies through spatial fingerprinting (case study 4). SPLAT will be made openly available to the community, and the set of built-in capabilities is expected to increase as users experiment with it in the context of more diverse applications.

Acknowledgments

This work is supported in part by NSF grant numbers CNS-0347252, ANI-0335234, and CCR-0325653 and by Cisco Systems. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF or of Cisco Systems.

References

1. Abilene backbone network. http://abilene.internet2.edu/, 2006.
2. CAIDA visualization tools. http://www.caida.org/tools/visualization/, 2006.
3. Ethereal: A network protocol analyzer. http://www.etheral.com/, 2006.
4. Ggobi data visualization system. http://www.ggobi.org/, 2006.
5. Nam: Network animator. http://www.isi.edu/nsnam/nam/, 2006.
6. tcptrace—official homepage. http://jarok.cs.ohiou.edu/software/tcptrace/tcptrace.html, 2006.
7. Trolltech—creators of Qt—the multi-platform C++ GUI/API. http://www.trolltech.com/, 2006.
8. P. Abry and D. Veitch. Wavelet analysis of long range dependent traffic. *IEEE Transactions on Information Theory*, 44(1):2–15, 1998.
9. S. Agarwal, J. Sommers, and P. Barford. Scalable network path emulation. In *Proceedings of IEEE MASCOTS '05*, September 2005.
10. C. Estan, S. Savage, and G. Varghese. Automatically inferring patterns of resource consumption in network traffic. In *Proceedings of ACM SIGCOMM '03*, Karlsruhe, Germany, 2003.
11. M. Jain and C. Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp throughput. In *Proceedings of ACM SIGCOMM '02*, Pittsburgh, Pensylvania, August 2002.
12. E. Kohler, J. Li, V. Paxson, and S. Shenker. Observed structure of addresses in IP traffic. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop '02*, Marseilles, France, October 2002.
13. W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, pages 2:1–15, 1994.
14. M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communications Review*, 27(3), July 1997.
15. T. Munzner. *Interactive Visualization of Large Graphs and Networks*. PhD thesis, Stanford University, 2000.
16. V. Paxson. End-to-end Internet packet dynamics. In *Proceedings of ACM SIGCOMM '97*, Cannes, France, September 1997.
17. V. Paxson. Strategies for sound Internet measurement. In *Proceedings of ACM IMC '04*, October 2004.
18. J. Sommers and P. Barford. Self-configuring network traffic generation. In *Proceedings of ACM SIGCOMM Internet Measurement Conference '04*, 2004.
19. J. Strauss, D. Katabi, and F. Kaashoek. A measurement study of available bandwidth estimation tools. In *Proceedings of ACM IMC '03*, Miami, Florida, October 2003.
20. J.W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.
21. L. Zhang and D.D. Clark. Oscillating behavior of network traffic: A case study simulation. *Journal of Internetworking: Research and Experience*, 1:101–112, 1990.
22. L. Zhang, S. Shenker, and D.D. Clark. Observations on the dynamics of a congestion control algorithm: the effects of two-way traffic. In *Proceedings of ACM SIGCOMM '91*, pages 133–147, 1991.
23. Y. Zhang, L. Breslau, V. Paxson, and S. Shenker. On the characteristics and origins of internet flow rates. In *Proceedings of ACM SIGCOMM '02*, Pittsburgh, Pensylvania, August 2002.