# A Framework for Analysis of Anonymized Network Flow Data

Mark Meiss
Dept. of Computer Science and Adv. Network Mgmt. Lab
Indiana University, Bloomington, IN, USA
mmeiss@indiana.edu

Filippo Menczer
Dept. of Computer Science and Dept. of Informatics
Indiana University, Bloomington, IN, USA
fil@indiana.edu

Alessandro Vespignani
Dept. of Informatics
Indiana University, Bloomington, IN, USA
alexv@indiana.edu

## Abstract

*Many projects analyze application overlay networks on the Internet using packet analysis and network flow data. This is infeasible on many networks: either the volume of data makes packet inspection intractable, or privacy concerns forbid packet capture and require the dissociation of network flows from users' identities. We describe a framework for exploration of usage patterns even under circumstances where the only available data is anonymized flow records. We offer two proofs of concept using data gathered from Internet2. In the first, we uncover distributions and scaling relations in host-to-host networks with implications for capacity planning and application design. In the second, we classify network applications based on properties of their overlay networks, yielding a taxonomy that allows us to identify the functions of unknown applications.*

## 1 Introduction

Understanding the structure and dynamics of the virtual networks formed by network users has become a major research focus. While these networks are of great sociological interest, understanding their properties is also important for research topics as varied as intrusion detection and network capacity planning. This results in a methodological challege: researchers in many areas want to mine network data, but the primary data sources — captured packets and network flow data — are vast and contain sensitive personal information. Using this data can require computing power and network access unavailable to most interested researchers. Even more recent systems that use only network flow data still associate flow records with users' IP addresses, raising privacy concerns that limit data distribution. The Internet2 network, for example, forbids distribution of non-anonymized flow data outside the organization.

Given that packet inspection is intractable for high-speed networks and access to raw data raises privacy concerns, the question becomes: Can anonymized network flows still yield useful insights for network researchers? The traditional view of flows as rows in a large relational database makes this sound unpromising; all we know is the magnitude of flows between two unreliably identified ports on two unknown hosts. However, this purely relational view has been superceded by approaches that use flows to build a complex host-to-host network, as proposed independently by our group [6] and by Karagiannis *et al.* [4]. Such an approach allows the application of many machine learning and data mining techniques. Their success in traffic classification and anomaly identification suggests great promise in treating flows as feature vectors rather than simple tuples.

We present an analysis framework that extends these approaches by using flow records to build graph representations in which the nodes are hosts, ports, or applications, allowing us to apply machine learning techniques and approaches from complex networks analysis. This framework offers a number of novel contributions to Internet data mining: (1) We define a weighted digraph representation of flow data that allows for several single-mode projections, defining host-to-host (behavioral), port-to-host (functional), and port-to-port (application) networks; (2) We use only anonymized flow data, requiring no access to raw packets or the actual addresses used by a flow; (3) We analyze these graphs using complex networks research techniques, showing the utility of this approach in two applications: (i) we distinguish different classes of traffic by their distributions and scaling relations, with implications for modeling, capacity planning, and design; and (ii) we show how the topological properties can be used to develop a taxonomy of ap-

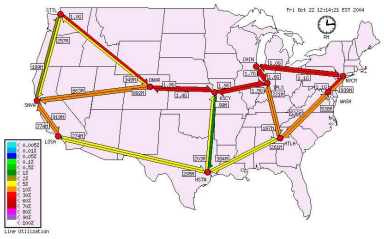

**Figure 1. Typical Abilene network activity.**

plications and use this hierarchy to identify the function of unknown applications; and (4) We argue for the tractability of our approach and its utility in real-time analysis. Our system is not a traffic classifier. We are concerned not with what flows *are*, but with what they *do*: how they affect the network and what they can predict about future activity.

Though space forbids detailed background on network flow analysis and related research efforts, several contemporary projects are similar to our described approach, but differ in privacy considerations, the entities modeled, etc. [5, 2, 4]. For detailed information on network flows and analysis tools, we recommend CAIDA's Web site [1].

## 2 Graph construction

Our data source consists of anonymized flow data typical of that available to a broad audience of researchers and includes neither host identities nor captured packets. The Abilene network (a part of Internet2[3]) provides an excellent source of flow data for studying user behavior. This high-speed TCP/IP data network spans the US and provides connectivity to several hundred research laboratories, colleges, and universities. Its backbone consists of 10-Gbps fiberoptic links connecting eleven high-performance routers. The network's primary intent is for academic traffic, but it is increasingly peering with the commodity Internet as well. Among its users are hundreds of thousands of undergraduates who are among the first adopters of new applications. Abilene also provides transit for dozens of international academic networks, serving as a major path between Asia and Europe and giving its data international character. Abilene is also uncongested even during peak hours (see Figure 1), offering a view of what users do when the network does not impede their behavior.

Current technology prevents collection of flow data for every network connection; each router samples about 1% of packets to generate flow information, which is sent from each router to our analysis system. In accordance with Internet2 policy, this system removes the actual IP addresses of each flow, replacing them with index values that are maintained for a single day. We save only these anonymized flows, which total 700 million flows on a typical day. At 48 bytes per record, a full day of data thus consumes over 30 GB of disk. Each record describes the directed transmission of data between a pair of hosts and ports without identifying which is the client and which is the server. We generate several graphs from these flow records, which we term *behavioral*, *functional*, and *application* networks.

We derive the behavioral network for an application by first recovering the roles of clients and servers. This is done using the number of flows that use a port: because clients use ephemeral port numbers and servers' ports must be known, the server will almost always be the endpoint with the more common port number. We thus partition all hosts into a subset $C = \{i_1, \cdots, i_{N_C}\}$ that act as clients and a subset $S = \{j_1, \cdots, j_{N_S}\}$ that act as servers. Some computers, especially those in P2P networks, act as both and are assigned to both sets. We use the sets $C$ and $S$ to construct a *behavioral* graph in which the nodes represent individual hosts and edges represent the directed transmission of data, aggregated over the course of a day. Each weight $w_{ij}$ represents the (sampled) data sent from a client to a server during a day, and $w_{ji}$ represents the amount of data sent in the opposite direction. This yields a bipartite digraph between clients and servers, weighted by aggregate traffic. This representation is used for the analysis in the following section.

When we build a *functional graph* among server ports and client hosts, we can capture the variety of activities in which each user engages. Each weight represents the amount that a host has used a particular port. (We consider only TCP data simply because UDP data on Abilene is minor and dominated by test traffic.) Since each port roughly corresponds to an application, this graph can be used to characterize applications by their host profiles: the traffic volumes exchanged by their users. We can then study the associations among applications by comparing their profiles, using the intuition that correlated use of applications is evidence they have a similar purpose, just as papers with consistent co-citations are likely related. We thus construct *application graphs* having ports as nodes and weighted edges representing their similarities, which can be used to classify unknown applications based on their observed usage. In Section 4 we present a rough taxonomy of applications and predict the function of unknown applications without inspecting any actual packet.

## 3 Behavioral network analysis

Our first case study shows how analysis of behavioral graph yields insight into capacity planning and application design. This analysis is based on 24 hours of Internet2 flow data gathered on April 14, 2005. This was a typical day, with no known outages or disruptions, and our findings are consistent with earlier studies [6]. On this day, our analysis system received over 600 million flows involving 15 million

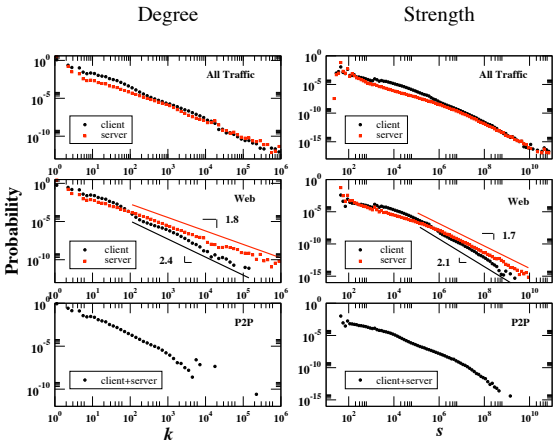


**Figure 2. PDFs for degree and strength in the behavioral graph, shown for all data, the Web, and P2P. We use log-sized histogram bins normalized by bin width and distribution size. The annotated lines show power-law approximations, with $R^2 \geq 0.995$.**

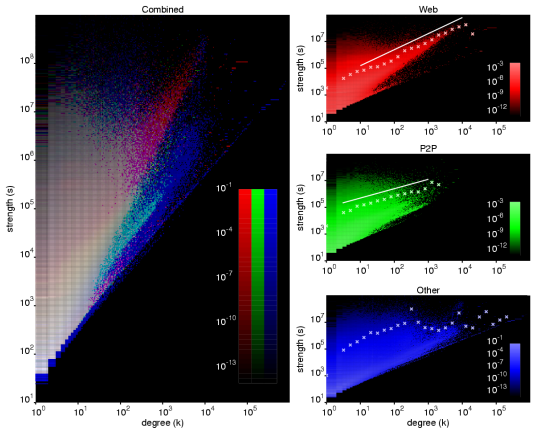


**Figure 3. Behavior of $s$ as a function of $k$ for all categories of traffic. The tones represent the density of $s$ normalized within each $k$-bin, on a log scale. The plotted points show $\langle s(k) \rangle$, and the lines show power-law approximations to the data, with $R^2 \geq 0.999$.**

hosts. Of these flows, 41.3% were Web-related and 13.1% involved known P2P applications. The remaining flows describe all other traffic, including attacks and test traffic. Of the hosts, 5.8 million were observed behaving as clients and nearly twice as many behaving as servers. This high ratio of servers to clients indicates scanning activity: rogue clients search for vulnerable servers. The opposite is the case for both Web and P2P applications, where we find only 20% as many servers as clients. The bipartite graph containing all hosts and applications has 131 million edges. If we examine the subgraphs for particular classes of application, we find that the Web graph contains 38.0% as many edges as the full graph and the P2P graph has only 6.0%. The total volume of traffic recorded is 1.85 TB (17.4% of which is Web-related); because of packet sampling, the true amounts are actually 100 times greater.

Unfortunately, these statistics tell us little about the role a typical user plays. We thus turn to the *structure* of these subsets of the behavioral network, first examining distributions of *degree* and *strength* in the graph. Given a node $N$ with $i$ initial edges and $j$ terminal edges, we define the degree as $d_N = i + j$ and the strength as $s_N = \sum_{k=1}^{i} w_{N,N_k} + \sum_{k=1}^{j} w_{N_k,N}$, where $w_{a,b}$ is the weight of the edge from $a$ to $b$. The degree of a node reflects the number of users with which it has exchanged data, and the strength reflects the amount of data. We can also aggregate traffic by specific ports to inspect the subgraphs for individual applications.

Because these distributions reflect decisions made by many individual users, we might expect them to be normal. We see in Figure 2 that this is far from the case: all distributions shown have extremely long tails, some spanning almost ten orders of magnitude. For instance, the mean strength of a client is 318 kB, but the standard deviation is 72.6 MB, making the level of fluctuation 100 times larger than the mean. So skewed are these distributions that for the Web and total traffic, we are able to approximate both $k$ and $s$ with a power-law approximation $P(n) \sim n^{-\gamma}$ over several orders of magnitude. The slope of these approximations says much about user behavior. When $2 < \gamma < 3$, the second moment $\langle n^2 \rangle = \int n^2 P(n) dn$ diverges; the variance is not intrinsic to the distribution and is bounded only by the size of the sample. In this case, $\langle n \rangle$ is no longer typical; lacking any characteristic mean, we have "scale-free" behavior. We may find a client that has contacted any number of servers or downloaded any amount of data, bounded only by the sample size. When $\gamma < 2$, as for Web servers, the situation is more dramatic. In this case, even the first moment $\langle n \rangle = \int nP(n) dn$ diverges and is bounded only by sample size. This extreme heterogeneity indicates that no "best scale" exists for the design of general-purpose Web server. In the case of P2P networks, the heavy-tailed distributions have a definite exponential cutoff, after which the PDF decays more quickly than a power law. This may be due to the limited capacity of most individual hosts on P2P networks, in which case we can expect the tail to lengthen over time as the average computer becomes more powerful.

The relationship between the number of hosts contacted (degree) and the amount of data exchanged (strength) also

helps to describe user behavior. Because of the power-law nature of these distributions, it is unsurprising that strength increases as a function of degree, again following a power law $\langle s(k) \rangle \sim k^{\beta}$, as shown in Figure 3. The value of $\beta$ is of primary interest: $\beta < 1$ implies a sublinear relationship; $\beta = 1$, a linear relationship; and $\beta > 1$, a superlinear relationship. The case of server behavior is linear or sublinear ($\beta \leq 1$), but for Web clients, $\beta = 1.2 \pm 0.1$, a clearly superlinear relationship. This means the amount of data exchanged with *each* Web server tends to increase as a user contacts more servers: the more sites surfed, the more data is received from each site. This non-linearity may be the basis for a method to disambiguate single users from Web crawlers. The relationship between $s_{in}$ and $s_{out}$ may also aid in discovering of open proxy servers, which should exhibit symmetry in their role in the behavioral network.

## 4 Application network analysis

In our second case study, we aggregate flows to build a graph in which the nodes are applications and the edges are measures of behavioral similarity among them. Hierarchical clustering of the nodes yields a taxonomy that can predict the function of unknown applications.

Non-standard applications comprise much of Internet traffic (over 40% of our sample). Researchers have only sketchy knowledge of the fauna of cyberspace even as the demand for reliable identification increases. Applications are also only *generally* identified by their port number, as we assumed in the previous section. This is accurate for a lot of traffic, but we must consider data generated by applications running on non-standard ports; for example, many users evade firewalls by running P2P applications on the Web port. Users disguise their activity by masquerading as other applications or using "ephemeral" ports not used by any known protocol. They can also evade security systems through encryption and tricks such as packet fragmentation. As a result, while we can monitor the *existence* of applications, we often do not know what kind of communication they support.

We approach this problem by examining the relationships between applications and clients (servers are less likely to use multiple applications or represent the actions of a single user). To describe client behavior, we define the *port strength* of a client node $i \in C$ as $s_i^p = \sum_{j \in C \cup S} w_{ij}^p$, where $p$ is a port. The port strength of a node reflects the amount of data it has exchanged using an application. We thus obtain a vector for each application, whose elements are the volume of data for that application for each host: $\vec{p} = (s_1^p, \ldots, s_{|C|}^p)$. We then measure the correlation of use between two applications $\vec{p}$ and $\vec{q}$ using their cosine similarity: $\sigma(\vec{p}, \vec{q}) = (\vec{p} \cdot \vec{q})/(\|\vec{p}\| \cdot \|\vec{q}\|)$. This quantity is zero in the case of orthogonal use and one in the case in which every host makes equal use of the applications. The result is a connected graph having ports as nodes and strength correlations as weights. We used a standard clustering algorithm to group the most highly used TCP ports by their correlations, as shown in Figure 4. The Web is so pervasive so as to be strongly correlated with nearly every application, though the different Web ports form a strong cluster among themselves. The groupings of the remainder also capture their functions. Bittorrent and Gnutella use a variety of ports that combine to form tight clusters. Standard client-server applications form groups, one including email, chat, and file transfer protocols, and another including those for streaming music and remote logins. Other P2P applications are clustered together, suggesting that many users employ more than one file-sharing application.

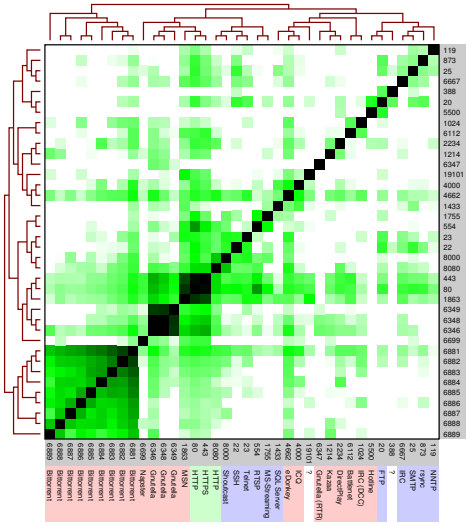


**Figure 4. Correlation of use of applications. We show the 38 ports with highest traffic, two of which are used by unknown applications (see text). The matrix shows the correlation between each pair of ports. The dendrogram is obtained using Ward's algorithm [8] after mapping similarity to distance. The ports are manually colored by group: P2P (pink); traditional client-server (blue); and Web (green).**

This clustering mirrors our existing understanding of the network but is of little use unless it can predict the nature of *unknown* applications. To confirm the utility of our technique, we used the similarity data to classify 16 ports unknown to us because of their unofficial use or their obscurity. Two such applications are included in Figure 4. Port 388 is coupled most strongly with FTP and Hotline; we found it to be assigned to "Unidata/LDM," a file transfer system used for moving large sets of meterological data.

**Table 1. Predicted uses of ports running unknown applications and their actual uses, as derived from online sources.**

| Port | Predicted | Actual | Match? |
|---|---|---|---|
| 388 | traditional file transfer | weather data transfer | yes |
| 19101 | P2P chat or file transfer | individual file shares | yes |
| 9080 | P2P with central index | team collaboration | yes |
| 8090 | Windows P2P w/ Web svc. | Weblog server | yes |
| 5020 | Windows P2P file transfer | BBFTP file transfer | partial |
| 42899 | P2P file sharing or trojan | *(unknown)* | unknown |
| 8301 | P2P file sharing or trojan | several trojans | partial |
| 1025 | trojan | many different trojans | yes |
| 20000 | P2P, probably BitTorrent | BitTorrent | yes |
| 59174 | P2P file sharing or trojan | *(unknown)* | unknown |
| 20001 | P2P file sharing or trojan | several trojans | partial |
| 15002 | P2P file sharing or trojan | biology collab. tool | partial |
| 16881 | P2P, probably BitTorrent | BitTorrent | yes |
| 9000 | P2P file sharing or trojan | several trojans | partial |
| 3124 | Windows P2P file transfer | Web proxy (Windows) | yes |
| 39281 | P2P file sharing or trojan | grid-based computing | partial |

Port 19101 was grouped with both IM and P2P applications, suggesting that it might be P2P, but relying on individual contact for file transfers. This allowed us to form search queries to find that the port is used by "Clubbox," a Korean file-sharing program used to trade entire TV programs. Sniffing Clubbox data would have been of little use to anyone unfamiliar with Korean; the application graph gave us information that packet analysis could not.

The predictions for all 16 ports are shown in Table 1; eight were entirely successful. The partial predictions stem from applications clustered with both P2P ports and those strongly associated with malicious activity (IRC and SQL Server). We lacked sufficient experience to judge which purpose was more likely. This ambiguity also hints that systems involved with P2P applications may be compromised more often than usual, possibly through the applications themselves. We could not verify our predictions for two ports because they were in use only transiently during our sample period; they no longer carry appreciable traffic. We also observe that while Web proxies predate P2P networks, their function is similar.

## 5 Conclusions

In our first case study, our graph-centric analysis reveals aspects of user behavior that are essential for agent-based modeling approaches, with implications for network management and epidemiology. The pervasiveness of heavy tails implies that user behavior rarely follows normal distributions, but is so diverse as to make mean values meaningless. Superlinear behavior in Web clients demands that models be able to account for non-trivial coupling of degree and strength. The differences between the Web and P2P graphs imply that we can construct signatures for different application types, allowing the identification of hidden applications without violating user privacy. Finally, while many security products use rate thresholds to detect traffic anomalies [7], our results show that we can expect many false alarms even from normal traffic. The analysis of behavioral networks may offer an effective approach for detecting malicious and anomalous behavior.

The application clusters identified in our second study show that one can easily infer traits of the activity on a port even if the actual application is unknown. We can group applications by the way they *affect* the network rather than their code base or stated purpose. The potential becomes clear when we consider how these clusters may evolve as people use existing applications in radically different ways. Usenet began as a system for propagating small text articles and NNTP has not changed — but its traffic now consists mostly of CDs, DVDs, and other audiovisual data. The protocol has not changed, but the way people use it has.

Our framework offers a way of examining user behavior through analysis of the graph structures formed by their actions. Because we avoid any use of raw packets or non-anonymized data, a wide audience of researchers can test these techniques for themselves. None of our processing steps require unusual resources; a single workstation can perform the analysis of the first case study in less than 30 minutes. The analysis of the second case study is quadratic in the number of applications but can still be performed in less time than taken to collect the data. Finally, we are working to make our analysis tools publicly available.

The authors thank the Advanced Network Management Laboratory at Indiana University for support and infrastructure and Internet2 for its generous use of anonymized flow data. This work is funded in part by NSF awards 0348940 and 0513650 to FM and AV, and by the Indiana University School of Informatics.

## References

[1] The Cooperative Association for Internet Data Analysis (CAIDA). http://www.caida.org/home/.

[2] J. Erman, M. Arlitt, and A. Mahanti. Traffic classification using clustering algorithms. In *ACM SIGCOMM Workshop on Mining Network Data*, pages 281–286, 2006.

[3] Internet2 project. http://abilene.internet2.edu/.

[4] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: Multilevel traffic classification in the dark. In *SIGCOMM*, pages 229–240, 2005.

[5] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. In *ACM SIGMETRICS*, pages 61–72, 2004.

[6] M. Meiss, F. Menczer, and A. Vespignani. On the lack of typical behavior in the global Web traffic network. In *Proc. 14th WWW Conference*, pages 510–518, 2005.

[7] A. Networks. Peakflow. http://www.arbor.net/.

[8] J. H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.