Properties and Evolution of Internet Traffic Networks from Anonymized Flow Data

MARK MEISS, Indiana University
FILIPPO MENCZER and ALESSANDRO VESPIGNANI,
Indiana University and Institute for Scientific Interchange

Many projects have tried to analyze the structure and dynamics of application overlay networks on the Internet using packet analysis and network flow data. While such analysis is essential for a variety of network management and security tasks, it is infeasible on many networks: either the volume of data is so large as to make packet inspection intractable, or privacy concerns forbid packet capture and require the dissociation of network flows from users' actual IP addresses. Our analytical framework permits useful analysis of network usage patterns even under circumstances where the only available source of data is anonymized flow records. Using this data, we are able to uncover distributions and scaling relations in host-to-host networks that bear implications for capacity planning and network application design. We also show how to classify network applications based entirely on topological properties of their overlay networks, yielding a taxonomy that allows us to accurately identify the functions of unknown applications. We repeat this analysis on a more recent dataset, allowing us to demonstrate that the aggregate behavior of users is remarkably stable even as the population changes.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network topology*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Applications*; C.2.3 [**Computer-Communication Networks**]: Network Operations—*Network management; network monitoring; public networks*

General Terms: Management, Measurement, Security

Additional Key Words and Phrases: Network flows, Internet usage, traffic statistics, behavioral networks, functional networks, application networks, application identification, power-law networks, latitudinal analysis, evolution of networks

ACM Reference Format:
Meiss, M., Menczer, F., and Vespignani, A. 2011. Properties and evolution of internet traffic networks from anonymized flow data. ACM Trans. Internet Technol. 10, 4, Article 15 (March 2011), 23 pages.
DOI = 10.1145/1944339.1944342 http://doi.acm.org/10.1145/1944339.1944342

1. INTRODUCTION

Understanding the structure and dynamics of the virtual networks formed by Internet users and applications has become a major focus of Internet-related research [Crovella and Krishnamurthy 2006]. While these networks are of great sociological interest,

This work was funded in part by NSF awards 0348940 and 0513650 to F. Menczer and A. Vespignani, respectively, and by the Indiana University School of Informatics and Computing.
Authors' addresses: M. Meiss, School of Informatics and Computing and Advanced Network Management Laboratory, Indiana University; email: mmeiss@indiana.edu; F. Menczer, School of Informatics and Computing, Indiana University, and Institute for Scientific Interchange, Torino, Italy; A. Vespegnani, School of Informatics and Computing, Indiana University, and Institute of Scientific Interchange, Torino, Italy.

DOI 10.1145/1944339.1944342 http://doi.acm.org/10.1145/1944339.1944342

understanding their properties is also important for research topics as varied as intrusion detection, application design, and network capacity planning. However, this broad applicability creates a tension in the Internet community: researchers in many areas want to mine network data, but the primary data sources used by most analysis systems—captured packets and network flow data—are vast and contain personal and sensitive information. Using these systems can require computing power and a level of access to the network unavailable to most interested researchers. Even systems such as BLINC [Karagiannis et al. 2005], which operate "in the dark" by using only network flow data, still associate flow records with users' actual IP addresses, raising privacy concerns that limit the availability of the data. The Internet2 network, for example, does not allow the distribution of nonanonymized network flow data outside the organization itself.

Given that packet inspection is not computationally tractable for high-speed data networks and that access to raw network flow data raises serious privacy concerns, the question becomes: If we restrict our data source to anonymized network flows, can it still yield useful insights for Internet researchers? If one takes the traditional view of network flow records as rows in a large relational database, the answer does not seem promising; we are left largely with information about the magnitude of flows between two unreliably identified ports on two unknown hosts. However, this purely relational view of flow data has been shattered by approaches that use flow data to build a complex host-to-host network, as proposed independently by our group [Meiss et al. 2005] and by others [Karagiannis et al. 2005]. Such approaches allow us to apply both machine learning and data mining techniques. Their success in traffic classification and identification of network anomalies suggests great promise in treating flow data more as feature vectors than as simple tuples.

The analytical framework we present in this article extends these approaches by using network flow records to build graph representations of network traffic in which the nodes are hosts, ports, or applications. This makes flow data amenable to both machine learning techniques and approaches from complex networks analysis. We present two case studies that illustrate the utility of this graph-centric approach, extending and integrating our preliminary findings as described in Meiss et al. [2007, 2008b]. The first study, involving behavioral networks, extends work described in Meiss et al. [2008b]; the second, involving application identification and classification, is novel to this presentation. In both cases, we compare results from a 2005 dataset and more recent data from 2008. The framework we present offers a number of original contributions to research in Internet traffic analysis and measurement.

— We define a general weighted directed graph representation of network flow data that allows for several single-mode projections, defining host-to-host (behavioral), port-to-host (functional), and port-to-port (application) networks.
— We use only anonymized network flow records, requiring no access to packet contents or the actual IP addresses associated with a flow.
— We apply analysis techniques from complex networks research to these graph structures, showing the utility of this approach in two practical applications: (i) we characterize different classes of traffic by their distributions and scaling relations derived from flow graphs, with implications for network modeling, capacity planning, and application design; and (ii) we demonstrate how the topological properties of flow networks can be used to develop a taxonomy of network applications which can then accurately identify the function of unknown applications.
— We argue for the computational tractability of our approach and its potential use in real-time analysis.

— We present evidence through our longitudinal analysis that the long-tailed distributions, scaling relations, and other large-scale properties we describe are not particular to our measurements, but rather inherent properties of Internet traffic data.

It is important to note that the overall aim of our flow analysis is not traffic classification. We are not concerned with what individual flows *are*, but rather with what they *do*: how they affect the network and what they allow us to predict about future activity on the network.

In Section 2, we introduce related research in network flow analysis and indicate how the present work compares to these approaches. In Section 3, we offer a technical description of our data source, placing particular emphasis on its origin, extent, and degree of anonymity. Section 4 describes the derivation of the data structures we propose as a framework for flow analysis. In Sections 5 and 6, we present case studies showing the utility of this framework. Finally, in Section 7, we summarize our findings and discuss the broader applicability of our approach for the Internet research community.

2. BACKGROUND

While a great deal of Internet measurement research [Claffy 2006; Krioukov et al. 2007; Shavitt et al. 2004] has focused on investigation of the structure and growth dynamics of the physical Internet [Alderson et al. 2005; Fabrikant et al. 2002; Jin et al. 2000; Li et al. 2004; Medina and Matta 2000; Pastor-Satorras and Vespignani 2004; Yook et al. 2002], researchers have devoted an increasing amount of effort to analyzing the virtual networks formed by transport-layer connections. Because these networks reflect the actual user-to-user interactions that make up network traffic, they serve as a primary data source for modeling user and application behavior as well as detecting malicious network activity. Internet routers facilitate the gathering of information on user-to-user communications by the abstraction of a *network flow*, which is uniquely defined by the IP address, protocol, and port used by both nodes involved in a network transaction during a particular period of time. The most common form of flow data, Cisco's *NetFlow*, includes a variety of attributes that describe high-level features of each flow; it does not contain the actual contents of any network conversation. Because of the large volume of data transmitted on modern networks—a single 10-Gbps link can transfer well over half a petabyte of information every day—routers derive flow information from a sample of actual network packets, often at a rate of 1:100 packets.

The flow-centered view of network activity has yielded substantial benefits already. For instance, interdomain traffic has been studied on a global level by looking at data representing all traffic received by specific service providers [Uhlig and Bonaventure 2001]. A similar strategy has been used by the CAIDA measurement infrastructure, which allows for the construction of traffic matrices representing the traffic between pairs of Autonomous Systems [Claffy 1999; Huffaker et al. 2000]. More recently, aggregated flows have been used to detect anomalies and for time modeling of traffic [Lakhina et al. 2004c]. A variety of tools have been developed to support these aggregation-based analysis techniques: in particular, Mark Fullmer's flow-tools,[1] CAIDA's cflowd,[2] and FlowScan[3] have been widely recognized and used in capacity planning and bandwidth management, as well as basic academic research.

[1] `http://www.splintered.net/sw/flow-tools/`
[2] `http://www.caida.org/tools/measurement/cflowd/`
[3] `http://www.caida.org/tools/utilities/flowscan/`

Commercial systems such as Arbor Networks' Peakflow[4] use network flow data for statistical trending and anomaly detection. Autofocus also allows for the discovery of dominant and unusual traffic clusters [Estan et al. 2003].

However, these tools, and most of the research they support, consider network flows in the context of a traditional relational model. They examine properties such as the proportion of traffic generated by particular applications or the longevity of certain classes of connection. While this approach has merit, it does not aid in exploring properties that relate to the patterns of interaction formed by network flow data. For these reasons, researchers are increasingly turning to more sophisticated analysis techniques borrowed from machine learning and data mining. We now highlight a number of recent projects in this area and indicate how our own proposed framework relates to each one.

Good results in detecting traffic anomalies have been obtained through principal component analysis of flow-based time-series data [Lakhina et al. 2004a, 2004b]. While this approach does consider network flows as contributing weight to the router graph of the network, it departs from our approach by presuming the existence of "typical" traffic patterns along subspaces of maximal data variance. Indeed, our previous experience [Meiss et al. 2005] and the data presented here suggest that this may be an unrealistic assumption; real-world network data can exhibit unbounded variance even under normal conditions. A more recent project has used manifold learning algorithms instead of principal component analysis to create a system that reduces the dimensionality of flow data for visualization [Patwari et al. 2005]. That system allows the user to explore relationships among a variety of entities represented in flow data, including TCP destination ports. Our second case study also regards ports as independent entities, but takes the additional step of clustering them hierarchically and using this hierarchy to predict the function of unknown applications.

Other recent projects extend machine learning techniques beyond anomaly detection to encompass a variety of traffic classification tasks. The previously mentioned BLINC system uses flow data to develop "graphlets" that describe the normal usage patterns of a variety of network applications; these structures are then used in conjunction with host information to predict the application associated with a given flow independently of the port numbers used [Karagiannis et al. 2005]. Our second case study takes a related approach in that we analyze flow patterns at multiple levels of detail, but it is focused on classifying applications themselves rather than individual flows; moreover, we do not incorporate any prior knowledge about the application protocols concerned when constructing our taxonomy. Other projects have used Bayesian analysis [Moore and Zuev 2005] and unsupervised clustering algorithms [Erman et al. 2006] to associate flows with applications; the latter has been extended to work in circumstances where only asymmetric flow data is available [Erman et al. 2007]. Our framework could be used in conjunction with any of these techniques to produce a hybrid system for traffic classification, but such an application is beyond the scope of the present article.

There have also been some recent projects that apply machine learning and clustering techniques to partial packet traces in order to classify flows in real time [Bernaille et al. 2006; Zhang et al. 2004]. These systems are quite promising for security applications in that they make classification possible even while a flow is still active and are quite scalable, but their requirement for actual packet data is incompatible with our goal of supporting research that relies only on anonymized flow data.

There has already been some application of complex systems analysis to application networks, mostly notably that of the Web, and it is these projects which have been

[4]http://www.arbor.net/products_platform.php

the most direct inspiration for the present work. The majority of Web mining studies focus on the social network built from the *link graph,* in which vertices and directed edges identify Web pages and hyperlinks, and links are seen as endorsements among pages. Data gathered in large-scale crawls of the Web have uncovered the presence of a complex architecture with small-world properties and long-tailed distributions that characterize the structure of the graph [Adamic and Huberman 2001; Barabási and Albert 1999; Broder et al. 2000; Kumar et al. 2000; Laura et al. 2003]. Examples of this complexity have included navigation patterns, community structures, congestion, and other social phenomena resulting from users' behavior [Huberman and Lukose 1997; Huberman et al. 1998; Adamic and Huberman 2001; Menczer 2002, 2004]. Our own work on analysis of Web request traffic has revealed important patterns of temporal predictability and insight for Web traffic modeling [Meiss et al. 2008a]; however, such analysis requires access to HTTP headers and is therefore outside the scope of the approach presented here.

Besides the Web, other overlay networks have been examined in similar fashion, most notably email interaction and peer-to-peer networks [Ebel et al. 2002; Newman et al. 2002; Ripeanu et al. 2002; Saroiu et al. 2002]. Other researchers in the field have applied graph analysis to security topics, focusing on monitoring and characterization of the spread of computer viruses on the Internet [Forrest et al. 1997; Moore et al. 2002; Newman et al. 2002; Pastor-Satorras and Vespignani 2001; Staniford et al. 2002] and other malicious activities [Garetto et al. 2003; Moore et al. 2001; Singh et al. 2004; Zou et al. 2004].

A great many studies have examined the structure of the overlay networks associated with individual peer-to-peer applications (e.g., Ripeanu et al. [2002] and Li and Chen [2007]), but the popularity of any particular peer-to-peer service has proven fleeting: the heyday of Napster is gone, Gnutella is fading away, and few even remember WinMX. Our present work avoids consideration of any particular P2P application in favor of treating them as a general class whenever possible. While this does make our conclusions less specific, recall that our emphasis is on the effect that flows have rather than their technical identity. Indeed, the results of our case studies imply that P2P applications affect the network in similar ways independently of the specific protocols they use.

3. DATA SOURCE

We now provide a technical description of the particular source of network flow data used in the present research. This anonymized source of flow data is typical of that available to a broad audience of interested researchers and does not provide access to host identities or captured packets.

The Abilene network, which is part of the Internet2 project,[5] provides an excellent source of network flow data for studying properties of the traffic network formed by interacting users. Abilene is a high-performance TCP/IP data network that spans the United States and provides high-speed connectivity to research laboratories, colleges, and universities throughout the nation. The backbone of the network consists of 10-Gbps fiberoptic links connecting eleven high-performance routers located in major metropolitan areas. At the time of our initial data collection in April 2005, this network nominally carried only academic and research traffic, and the several hundred institutions that participate in the network were required to maintain their own connections to the commodity Internet. These requirements have since been relaxed, and at the time of our more recent connection in April 2008, the Abilene network had

[5]`http://abilene.internet2.edu/`

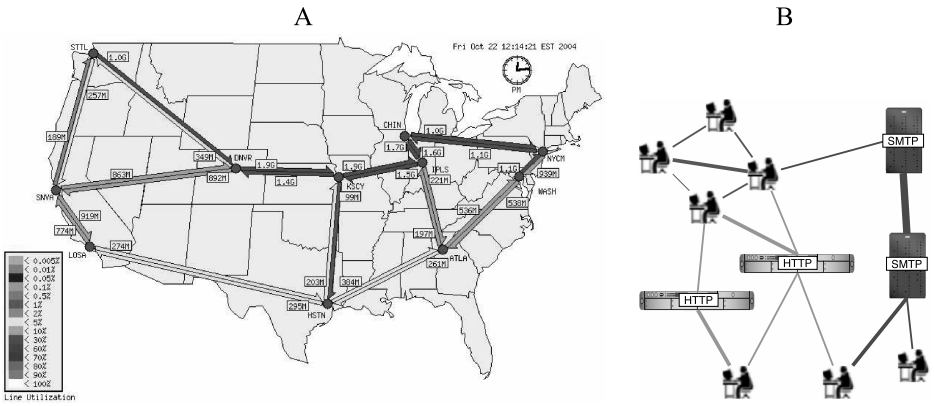

Fig. 1. (a) Typical activity levels between core routers in the Abilene network (sustained data rates in bits per second). Source: loadrunner.uits.iu.edu/weathermaps/abilene; (b) illustration of a behavioral network snapshot extracted from NetFlow data. Edge thickness represents amount of traffic. Flow records also specify traffic direction (not shown).

peering relationships with a variety of commercial network providers and carried a large quantity of traffic between colleges and the commodity Internet.

Among the users of the Abilene network are hundreds of thousands of undergraduate students who are among the first adopters of new network applications. In addition, the network provides transit for data from dozens of international academic and research networks, serving as a major transit path between Pacific Rim nations and Europe and giving an international character to its traffic. Abilene also offers the valuable property of never being congested even during peak usage, offering a view of what users do when the network itself does not impede their behavior. Typical levels of IPv4 traffic in the Abilene network are below 40% of capacity, as can be seen in Figure 1(a).

Current technology does not allow the collection of flow data for every single network conversation on Abilene; each core router samples about one in a hundred packets from their traffic load.[6] These packets are used to generate network flow information in Cisco's "netflow-v5" format,[7] which is sent from *each* of the core routers to an analysis system at our university. In accordance with the privacy policies of Internet2, this system removes the actual source and destination IP address of each flow, replacing them with index values that maintain their identity only over the course of a single day. Only this anonymized flow information is saved for analysis. On a typical day in 2005, the analysis system recorded around 700 million flow records; at 48 bytes per flow record, a full day of data thus consumes over 30GB of disk space and arrives at a mean rate of 3.1Mbps. By 2008, the daily number of flows had risen to nearly one billion, with proportionally greater demands on storage and analysis.

4. CONSTRUCTING BEHAVIORAL AND APPLICATION NETWORKS FROM FLOW RECORDS

The analysis of flow records enables the construction of different networks that depend on the way we aggregate the flow data. Each record describes the transmission of some quantity of data from some host and port to some other host and port, without

[6]Information is not formally available on the exact mechanism by which packets are sampled. We have empirically verified the sampling rate and conducted experiments to examine the effect of sampling bias on our results. While that analysis is beyond the scope of this article, we were able to conclude that sampling bias does not affect the reliability of the results presented here.

[7]`www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/nfc/nfc_3_0/nfc_ug/nfcform.htm`

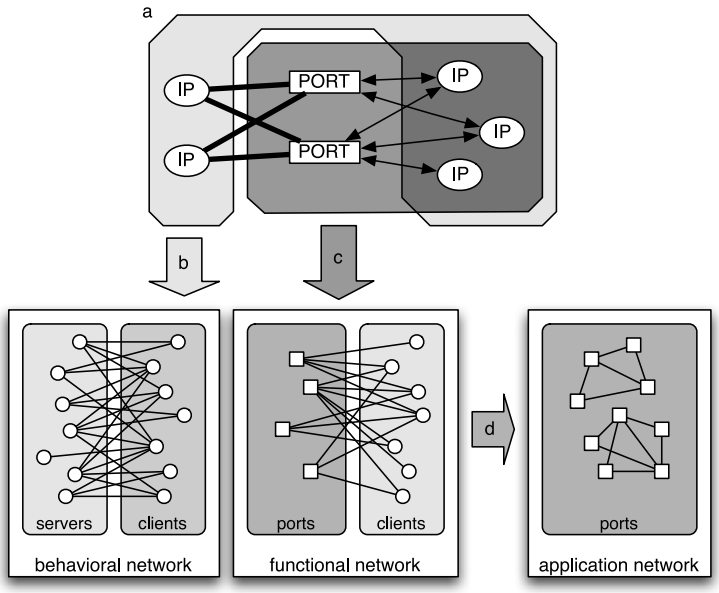

Fig. 2. The construction of behavioral, functional, and application networks from flow data. (a) Each raw flow record describes how many bytes are exchanged between two hosts, identified by their IP addresses, using application (TCP) ports on the sending and receiving hosts. Flows are aggregated over a day so that we have the total amount of data exchanged by two hosts for each pair of ports; (b) by aggregating flow data for each (anonymized) IP address across application ports, or focusing on individual standard ports, we can build networks of hosts (clients and servers) that describe how users are connected with each other and with services across the Internet; (c) by disregarding servers, and retaining TCP ports as entities, we can build a functional network describing the relationships between user hosts and applications; (d) by focusing on strength correlations among ports, we can cluster and identify application subnetworks.

identifying explicitly which host acts as a client and which acts as a server. Figure 2 illustrates different types of graphs that can be derived from the flow data, which we refer to as *behavioral, functional,* and *application* networks.

In deriving the behavioral network associated with an application or group of applications (Figure 1(b)), we begin by recovering the roles of clients and servers. This is done by examining the total number of flows that reference a particular port; because clients use ephemeral port numbers and server port numbers must be known in advance, in any particular record, the server will almost always be the system with the more frequently used port number. We can thus partition the set of all hosts into a subset $C = \{i_1, i_2, \cdots, i_{N_C}\}$ of systems that act as clients and a subset $S = \{j_1, j_2, \cdots, j_{N_S}\}$ of system that act as servers. Some computers on the Internet, especially those involved in peer-to-peer networks, act as both clients and servers and are thus assigned to both sets. Using the sets C and S, we construct a *behavioral* graph in which the nodes represent individual hosts operated by users or organizations and edges represent the directed transmission of data between a pair of hosts, aggregated over the course of a day. Each weight w_{ij} thus represents the total amount of sampled data sent from a particular client to a particular server over the course of a day, and w_{ji} represents the amount of data sent from a particular server to a particular client. This graph representation yields a bipartite digraph between clients and servers, weighted by aggregate volumes of traffic, as shown in Figure 2(b). The weighted representation of the behavioral networks of Internet2 hosts is the basis for the analysis in the following section.

When we retain port information and build a *functional network* among port numbers and client IP addresses, we are able to capture the variety of activities in which each particular user engages (Figure 2(c)). Each weight in the network represents the extent to which a host on the network has made use of a particular TCP port. (We consider TCP data exclusively because UDP data on the Internet2 network is fairly limited and dominated by network test traffic; there is nothing preventing the inclusion of this data in principle.) Since in general each port corresponds to a specific application, this functional network can be used to characterize applications by their profiles, that is, by the amounts of traffic exchanged by users over the corresponding ports. We can then study the associations among applications (ports) by comparing their host profiles; the basic intuition is that correlated use of two applications by users provides evidence that the applications have a similar purpose, similarly to the way in which two papers that are consistently cocited are likely to be related. This process allows for the construction of *application networks* (Figure 2(d)) having ports as nodes and weighted edges representing the usage correlations, and therefore similarity among ports as quantitatively detailed in Section 6. An application network can be used to classify *unknown* applications based on their observed usage patterns. In Section 6 we will present a rough taxonomy of Internet applications and include the results of an attempt to predict the function of over a dozen unknown applications without inspecting the content of any individual network packet.

5. CASE STUDY 1: BEHAVIORAL NETWORK ANALYSIS

We now present the results of a case study that shows how analysis of behavioral networks derived from flow data can yield insight into network capacity planning and application design.

5.1 Findings

This analysis is based on two sets of 24 hours of Internet2 flow data, the first gathered starting at midnight EST on April 14, 2005, and the second gathered starting at midnight EST on April 22, 2008. These were typical days in the life of the network, with no known major outages or disruptions of service, and our findings are consistent with those of earlier studies [Meiss et al. 2005].

For the 2005 dataset, in the course of the day, the flow collector received over 600 million flows involving almost 15 million hosts. Of these flows, 258 million (41.3%) were Web-related and 82 million (13.1%) were associated with known P2P applications. While these classifications are based on TCP port numbers and are thus individually suspect, the large and varied user population of Abilene strongly implies that a majority of flows are correctly identified by port. The remaining 285 million (45.6%) flows describe all other traffic, which includes network performance tests, pings, email, interactive logins, and a wide variety of miscellaneous and unidentified applications.

The 2008 dataset is larger, comprising over 980 million flows and involving just over 18 million hosts. The proportion of flows and edges associated with our categories changed somewhat, as shown in Figure 3. In particular, the proportion of traffic associated with the Web has grown, and that associated with P2P applications has apparently shrunk. However, this does not indicate an actual decline in the popularity of P2P applications, but rather evolution in the relative popularity of different applications and the ports they use. In Section 6, we describe how our application classification technique allowed us to identify new ports that have become associated with BitTorrent; with the addition of these ports, the volume of traffic we can associate with P2P networks is much greater in 2008 than in 2005. With the exception of Figure 3, all of the analysis we describe includes these ports, which carry a relatively small number

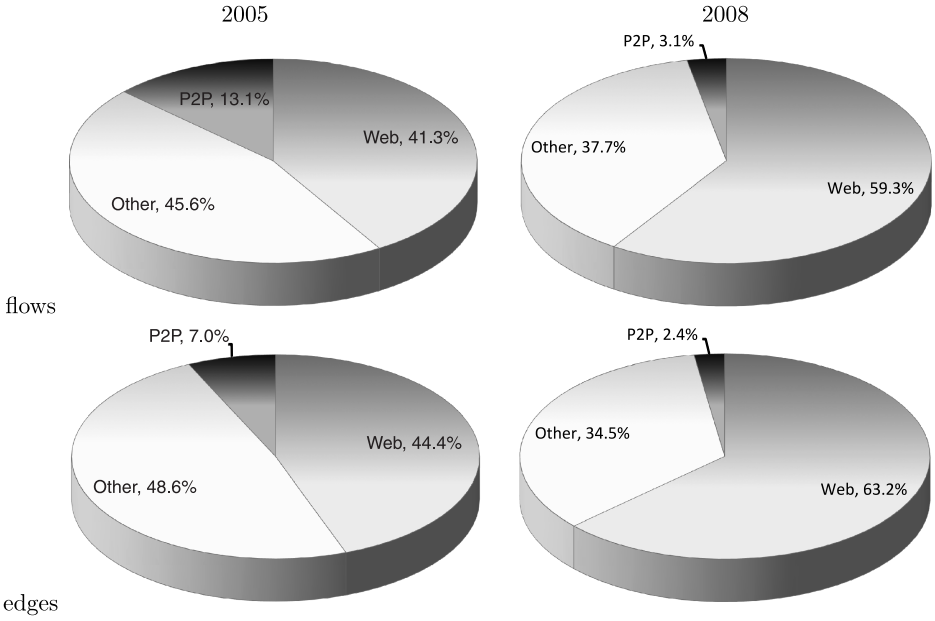

Fig. 3. Top: Proportion of collected flows generated by each category of traffic in 2005 (left) and in 2008 (right). Bottom: Relative sizes of bipartite graphs for each category of traffic as measured by number of edges in 2005 (left) and in 2008 (right). As explained in Section 6, these charts underestimate the contribution of P2P applications in 2008.

of very large flows that substantially increase the proportion of traffic associated with P2P applications.

In 2005, of the total number of hosts, 5.82 million were observed behaving as clients and nearly twice as many, 11.1 million, behaving as servers. Such a high proportion of servers to clients indicates the presence of scanning traffic on the network: rogue clients search for vulnerable servers. We find that the opposite is the case for Web and P2P applications. When we examine just Web flows, we find 3.97 million hosts behaving as clients and 0.68 million (less than one-fifth as many) behaving as servers. Similarly, for P2P traffic, there were 0.71 million clients and only 0.14 million servers. The remaining traffic shows 2.48 million clients and 10.6 million servers. The bipartite behavioral graph that includes all hosts and applications contains 131 million edges. If we examine subgraphs related only to particular classes of application, we find that the Web graph contains 50.1 million edges (38.0% as many as the full graph), the P2P graph contains 7.89 million edges (6.0%), and remaining TCP traffic contains 54.9 million edges (41.6%). (See Figure 3.)

For each category of traffic (Web, P2P, and the remainder), it is also instructive to examine the degree of overlap between C and S, which we represent with the quantity

$$O = (|C| \cap |S|)/(|C| \cup |S|).$$

When $O = 0$, no host acts as both client and server; when $O = 1$, every host does so. We would expect O to be lower for traditional client-server applications than modern P2P applications, and indeed, we find that $O = 0.013$ for Web traffic, compared to 0.097 for P2P traffic. This is a strong indication that hosting content for the Web is much less of a participant sport than sharing personal files, in that relatively few users run both Web clients and servers. However, we also note that $O = 0.15$ for other traffic, which suggests the presence of significant amounts of covert P2P traffic within

Table I. Volume of TCP Traffic Observed for Major Classes of Network Application as Determined by TCP Port Number

	Web		P2P		Other	
	2005	2008	2005	2008	2005	2008
Proportion of traffic	17.4%	28.5%	4.0%	7.1%	78.6%	64.4%
Mean data (client)	81 kB	146 kB	105 kB	395 kB	586 kB	250 kB
Mean data (server)	471 kB	936 kB	515 kB	1270 kB	137 kB	243 kB

As discussed in Section 6, the port assignments for P2P applications were updated for the 2008 dataset.

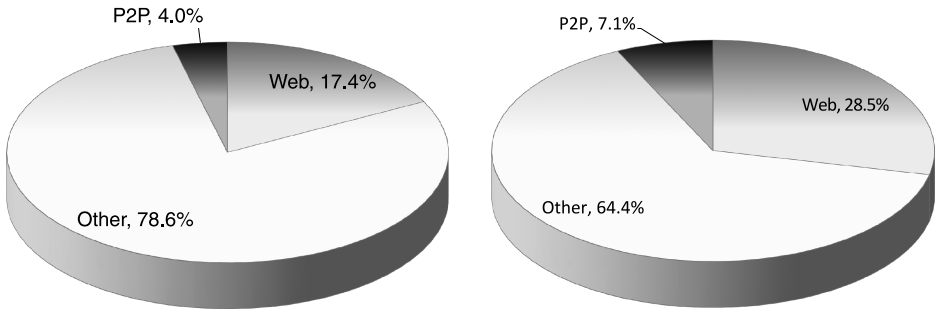

Fig. 4. Proportion of traffic volume consumed by each category of traffic in 2005 (left) and in 2008 (right). As discussed in the text, the set of ports associated with P2P applications has been revised for the 2008 dataset.

the unclassified data. In the next section, we expand on this by classifying traffic irrespective of TCP port numbers.

The 2008 dataset indicates less scanning traffic, with clients outnumbering servers by a healthy margin (12.8 million to 9.0 million). However, it duplicates the overwhelming margin of clients over server in the categories of Web and P2P traffic (6.0 million to 940 thousand and 540 thousand to 160 thousand, respectively). Furthermore, the small degree of overlap between C and S is still present: O is again close to 0.01 for Web traffic. Surprisingly, we observe a large decrease in the amount of overlap for P2P traffic, which has fallen by more than half, amounting to only 0.04. This low degree of overlap may be the result of many participants in P2P networks operating in bad faith: the ratio of clients to servers is over three to one, making it clear that many prefer downloading files to providing them to others. Other possible causes include the rise of projects monitoring the content of P2P networks and apparent asymmetries introduced by failed connections to servers that are no longer participating in file swarms.

The total volume of traffic recorded in 2005 was approximately 1.85 trillion bytes, with a mean of 124kB per host. In 2008 it had risen to 6.18 trillion bytes, with a mean of 343kB per host. However, because of the sampling involved in constructing the flow data at the routers, the true amount of traffic was actually about 100 times greater than these values. In Table I, we break this traffic down into the same broad application classes as before. We also note that values for "other" traffic are influenced by large volumes of *iperf* test traffic generated by the Abilene network operations center, which may exaggerate this category relative to other major data networks. This test traffic is difficult to separate from the rest of the data because the port numbers used are common to a number of applications in active use on the Abilene network. In Figure 4, we provide a visual comparison of the proportions of traffic by category in 2005 and 2008.

Unfortunately, the statistics just described provide little insight into the actual behavior of the user community; they tell us little about the role a typical user plays in the network. We thus turn our attention to the structure of the subsets of the behavioral network corresponding to the Web, P2P applications, and everything else.

We begin by considering the distributions of *degree* and *strength* for the nodes in the behavioral network. Given a node i with k_i^{out} outbound edges and k_i^{in} inbound edges, we define the *degree* as

$$k_i = k_i^{out} + k_j^{in}$$

and the *strength* as

$$s_i = \sum_{j=1}^{k_i^{out}} w_{ij} + \sum_{j=1}^{k_i^{in}} w_{ji},$$

where $w_{i,j}$ denotes the weight of the edge between nodes i and j. In other words, the degree of a node in the behavioral network reflects the total number of users with which it has exchanged data, and the strength reflects the total amount of data it has exchanged. In addition, by aggregating traffic by specific ports, it becomes possible to inspect the behavioral subnetworks concerning individual applications.

Because both the degree and strength distributions reflect the decisions made by a large population of individual users, it might seem plausible for their form to be roughly normal. This turns out to be far from the case, however, as shown in Figure 5. The extreme length of the tails of all of the degree and strength distributions, some spanning almost ten orders of magnitude, necessitates plotting their probability distributions on double logarithmic axes. As an example of this extreme diversity, the mean strength of a client in 2005 was approximately 318kB, but the standard deviation of the distribution was 72.6MB; the level of statistical fluctuation is over two orders of magnitude larger than the mean value. Indeed, the distributions are so skewed that in the case of all traffic and the Web behavioral network, we are able to approximate both the degree and strength distributions with a power-law approximation $P(n) \sim n^{-\gamma}$ over several orders of magnitude. In particular, for the Web behavioral network, we find that γ is roughly 2.4 for client degree, 2.1 for client strength, 1.8 for server degree, and 1.7 for server strength. These properties are consistent between the datasets, with little change in the exponents even after three years of rapid chance in the network itself.

The slope of these power-law approximations is of great significance in analyzing user behavior. When $\gamma < 3$, the second moment of a quantity n

$$\langle n^2 \rangle = \int n^2 P(n) dn$$

diverges; the standard deviation is not an intrinsic value of the distribution and is bounded only by the size of the data sample. In such a case, the average value $\langle n \rangle$ is no longer typical, and we lack any characteristic mean for the system; this is the often-mentioned "scale-free" behavior. We have an appreciable probability of finding a client that has contacted any arbitrary number of servers or downloaded any arbitrary amount of data, without any bound other than the size of our sample. The averages appear to be of no value in predicting the behavior of users.

When $\gamma < 2$, as is the case for both the degree and strength distributions of Web servers, we have an even more dramatic situation. In this case, even the first moment

$$\langle n \rangle = \int n P(n) dn$$

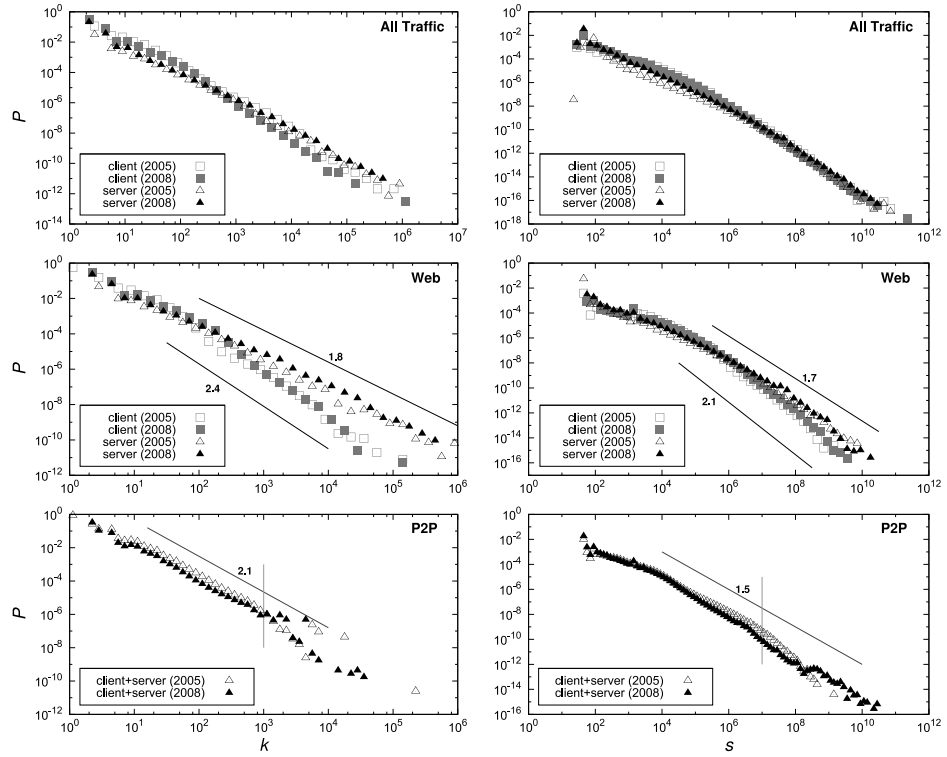

Fig. 5. Probability distributions for degree (left) and strength (right) in the Internet2 behavioral network in 2005 and 2008, shown for all data (top), the Web (middle), and peer-to-peer applications (bottom). The data are grouped into logarithmically-sized histogram bins normalized by the width of the bin and size of the distribution, so that we are estimating a probability density function. The annotated lines in the Web plots show statistically significant best-fit power-law approximations to the actual data, with $R^2 \geq 0.995$. Reference power-law fits are also included for P2P applications, together with a visual indication of the onset of exponential decay for the 2005 data.

diverges and is bounded only by the size of the sample. Neither the mean number of connections nor the mean amount of data transmitted are intrinsic to the system.

In the case of P2P networks, we do observe heavy-tailed distributions, but there is evidence of an exponential cutoff, after which the probability function decays more quickly than a power-law fit would predict. This effect is more evident in the 2005 data, which leads us to conjecture that the cutoff may be due to the limited computing and network capacity of most individual computers participating in peer-to-peer networks. As the resources available to individual users increase over time, so do the widths of the degree and strength distributions, and the cutoffs move to the right.

The interaction between degree and strength, which describes the relationship between the number of hosts contacted and the amount of data exchanged, is also of considerable interest in understanding user behavior. Because of the power-law nature of distributions of degree and strength considered separately, it is unsurprising for strength to increase as a function of degree, again following a power law, as shown in Figure 6, which again uses a double logarithmic scale.

Basic power-law behavior $\langle s(k) \rangle \sim k^{\beta}$ may be expected of the interaction between degree and strength; it is the value of the exponent β that is of critical interest. If it

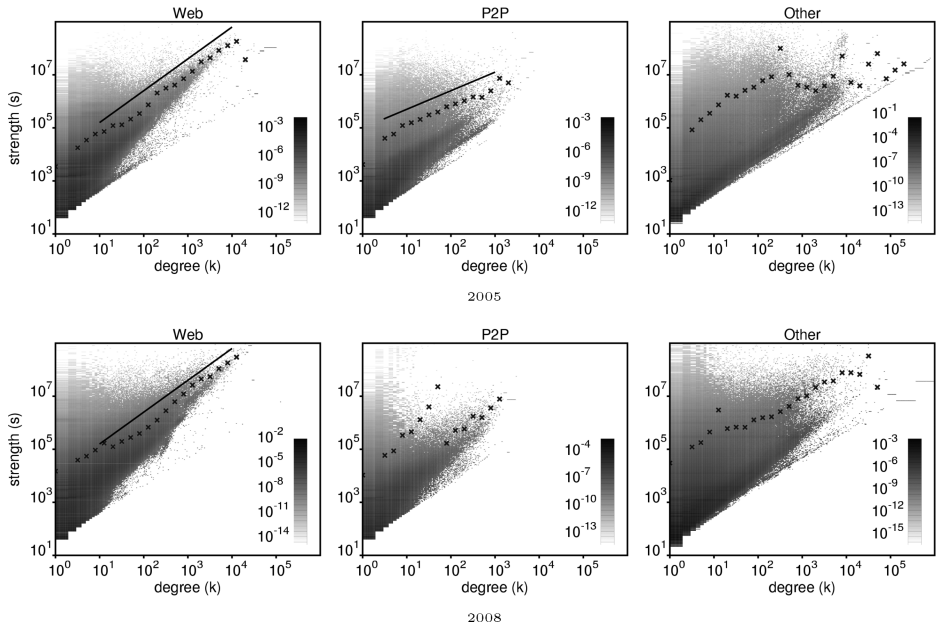

Fig. 6. Behavior of strength (total data) s as a function of degree (number of hosts contacted) k for Web, P2P, and other traffic, in 2005 (top) and in 2008 (bottom). The tones represent the frequencies of strength values, normalized within each degree bin, on a log scale. The plotted points show the mean strength for each degree bin, and the lines serve as a reference to power-law approximations to the actual data. For both datasets, the behavior of Web traffic is roughly linear on a double logarithm scale, with $R^2 \geq 0.999$ over the full dataset.

is the case that $\langle s(k) \rangle \sim k^{\beta}$, then $\beta < 1$ implies a sublinear relationship, in which case a diminishing amount of traffic is induced by each additional contact; $\beta = 1$, a trivial linear relationship, in which the magnitude of each contact is independent of the number of contacts; and $\beta > 1$, a superlinear relationship, in which case every additional contact tends to generate even more traffic—a statistically explosive situation. In the case of server behavior we find a linear or sublinear relationship ($\beta \leq 1$), but in the case of Web clients we see evidence that $\beta = 1.2 \pm 0.1$, giving clear indication of a superlinear relationship. This finding remains consistent between the 2005 and 2008 datasets even though the 2008 dataset includes a greater proportion of Web traffic, much of it commercial rather than academic.

5.2 Implications

The heterogeneity seen in the degree and strength distributions of Web servers, fitting a power-law distribution with $\gamma < 2$, is so extreme as to make their mean values no longer well-defined. We can infer no global average quantity for either the number of clients or the amount of data transmitted for a Web server, implying that no single scale is most appropriate for the design of general-purpose Web server software.

As mentioned previously, the existence of an exponential cutoff in the same distributions for the P2P networks may be a result of the limited processing power and network capacity of individual workstations. If these are the limiting factors in these distributions, we can expect the tails of the distributions to lengthen over time as the average computer continues to become more powerful and have access to greater network resources. This indeed seems to be the case; the distribution of P2P traffic for 2008 is over half an order of magnitude longer than it was in 2005.

The superlinear relationship observed between strength and degree for Web clients implies that 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 of these sites. Such a nonlinear growth mechanism may assist in techniques for disambiguating the behavior of individual Web surfers and large-scale crawlers. Individual users clearly lack the ability to digest an even larger body of information from each site as they surf the Web, whereas Web crawlers are designed to do exactly that. Crawlers can thus be expected to appear toward the upper-right of the degree-strength plot. We can obtain additional evidence to support such a classification can by examining whether the servers contacted are generally high or low in traffic: actual humans will tend to visit popular Web servers, while crawlers will visit a preponderance of obscure servers for the sake of completeness.

The relationship between the in- and out-distributions of strength may also facilitate the discovery of unrestricted proxy servers that are the launch points for a wide variety of security attacks on the Internet. Most of the traffic associated with a proxy for an application will be repeated: requests from a client to the proxy are retransmitted from the proxy to a server, and the server response is likewise retransmitted by the proxy back to the client. We would thus expect a proxy to exhibit an unusually high level of symmetry in its role in a behavioral network; not only would it function as both client and server, but its in-strength and out-strength would be nearly equal to one another. Unfortunately, the effects of sampling make this an unsuitable technique for early detection of open proxies: it is only substantial use that is likely to make their anomalously symmetric traffic distinguishable from other network servers.

6. CASE STUDY 2: APPLICATION NETWORK ANALYSIS

We now present the results of a second case study in which we aggregate flows across server hosts and project them onto ports. We then construct an application network in which the nodes are applications and the edges are measures of behavioral similarity between the applications. We perform hierarchical clustering on the nodes of this network to yield a taxonomy that is able to predict the function of a collection of unknown network applications.

The large volume of data in the "everything else" subgraph of the behavioral network serves as a strong example of how nonstandard applications comprise much of Internet traffic: none of the applications included in this category is an individually large contributor to overall traffic, but the cumulative weight of this long tail is a significant portion of the total. As discussed before, researchers and system administrators have in many cases a very incomplete knowledge of what is "out there" in the cyberworld, even in the face of increasing legal and ethical demands for reliable categorization of application traffic. The central problem is that in general, applications are simply identified by their *port*, which is a numeric label used by Internet protocols to multiplex communications; we ourselves have taken this approach in the analysis of the previous section. This yields accurate classification for a large portion of user interactions, but we must consider the growing proportion of Internet traffic generated by applications running on nonstandard ports or *covert channels*; for example, many users evade local firewall policies by running peer-to-peer applications on the port normally associated with the Web. Users may thus disguise their activities, for example, exchange email using the port of another application or an "ephemeral" port that Internet standards do not associate with any particular application. They can also evade most network security systems through encryption, packet fragmentation, and a variety of other techniques. The consequence is that while we can monitor

and be aware of the *existence* of applications that act as an interaction mechanism among users, we often do not know what *kind* of communication and function they support.

6.1 Findings

To approach the problem of identifying applications, we investigate the relationship between application and hosts by considering only the behavior of clients, as servers are likely to be devoted to hosting a single application and are much less likely to represent the actions of a single user. To describe the behavior of a client node, let us define the *port strength* of a client node $i \in C$ as

$$s_p(i) = \sum_{j \in C \cup S} w_p(i, j),$$

where p is an application port. In other words, the port strength of a node reflects the total amount of data it has exchanged using the associated application. We thus find a strength vector for each application, whose elements correspond to the amount of data exchanged in that application by each host.

$$\vec{p} = (s_p(1), \ldots, s_p(|C|))$$

We can then measure the correlation of use between two applications $\vec{p}$ and $\vec{q}$ by calculating, for instance, the cosine similarity of their vectors.

$$\sigma(\vec{p}, \vec{q}) = (\vec{p} \cdot \vec{q})/(\|\vec{p}\| \cdot \|\vec{q}\|)$$

This quantity ranges from zero in the case of completely orthogonal use, to one in the case in which every host uses the applications with proportional strength. The resulting application network is a connected graph (Figure 2(d)) having ports as nodes and strength correlations (as measured by cosine similarity) as the weights of their associations.

We used a standard clustering algorithm to group the 38 most highly used TCP ports in the 2005 dataset according to their strength correlations; the results can be seen in Figure 7 together with the correlations between applications. The Web is so pervasive among users so as to be strongly correlated with virtually every application, though the different ports associated with the Web form a very strong cluster among one another. The groupings of the remaining applications also capture their functions. BitTorrent, the most popular P2P network, uses a variety of different ports, but we can clearly see that they form a tight cluster (A). Another strong cluster (B) identifies the Gnutella network, the next most popular file sharing application. Standard client-server applications also form clusters. One (C) includes ports used by email, chat, and file transfer protocols; another (E) includes applications for listening to streaming music and logging into work from home. Several other P2P applications are also clustered together (D).

The Internet is a rapidly changing environment, and the popularity of network applications can change drastically in a brief period of time. The purpose of applications is likewise dynamic: at one time, Usenet (NNTP) was used primarily for the exchange of news and personal communication, but now its traffic is overwhelmingly dominated by people broadcasting media files. Similarly, IRC was developed as a chat protocol, but is now used just as frequently for peer-to-peer file transfers. To understand the pace and structure of this evolution in applications, we repeated our clustering analysis on the 2008 dataset, again using the ports with the highest traffic; the results are

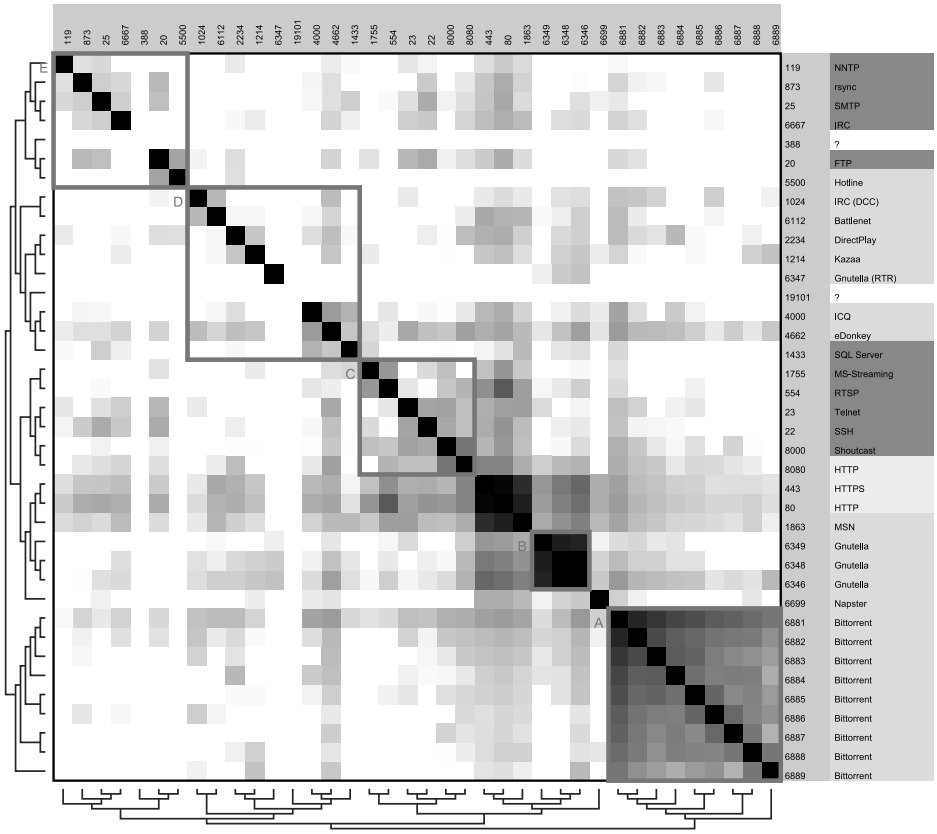

Fig. 7. Correlation of client use of network application as measured by cosine similarity of strength vectors in the 2005 dataset. We show the 38 ports with highest traffic, two of which are used by unknown applications (see text). The symmetric matrix shows the correlation between each pair of ports (with a threshold of 10^{-4}). The dendrogram used to sort the ports is obtained by applying Ward's hierarchical clustering algorithm [Ward 1963] after converting cosine similarity to a distance measure $(1/\sigma) - 1$. Alternative clustering algorithms, such as *k-means,* yield similar groupings. The ports have been manually labeled and shaded according to two broad classes: P2P (lighter) and traditional client-server (darker) applications. We use a third shade for ports associated with the Web (HTTP/HTTPS), since the use of the Web is so pervasive as to be almost synonymous with computer use in general. Furthermore, these ports are often used by other applications, for example, file transfers in Gnutella. The labeled boxes within the matrix highlight clusters of related applications as described in the text.

shown in Figure 8. It was immediately clear from this analysis that between 2005 and 2008, several new BitTorrent clients had become popular—even more so than the original client—and that these clients used a different set of default ports (20000–20029 and 40000–40007). We therefore adjusted the set of ports which we associate with P2P applications in the 2008 dataset, as mentioned earlier. This results in a larger collection of ports that nevertheless comprise a comparable portion of overall traffic as in 2005. Also included in the 2008 clustering are 14 unknown applications; we discuss how we may deduce their function in the following section.

6.2 Implications

The clustering of P2P applications based on correlation of use suggests that a significant body of users employ more than one file-sharing application over the course of the

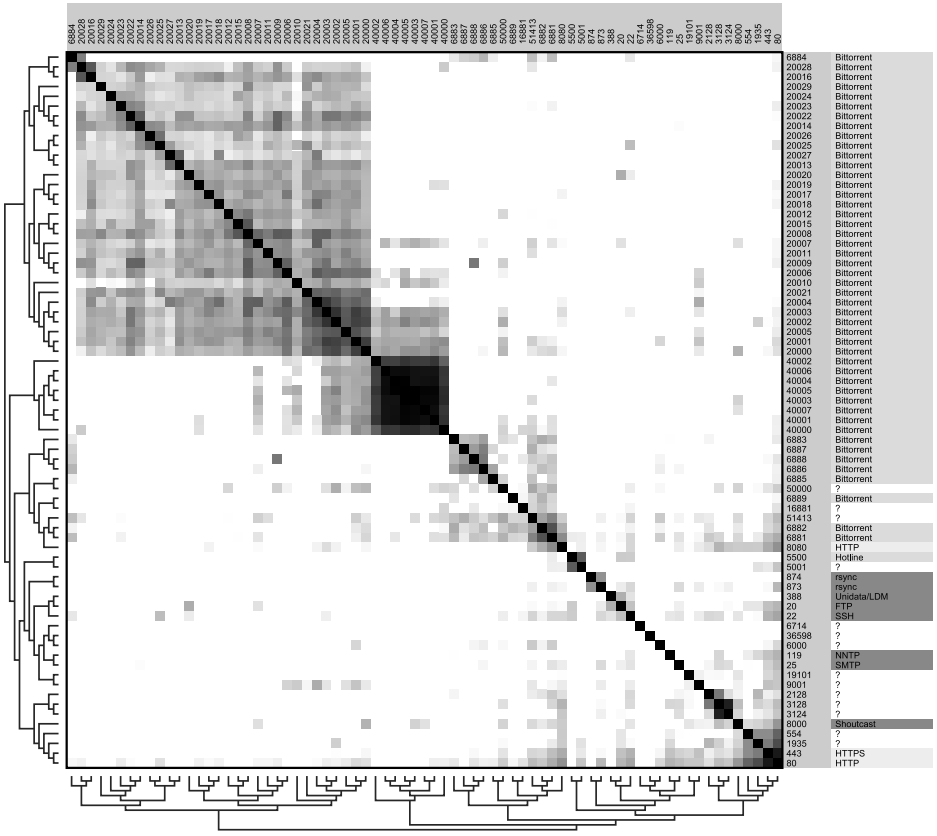

Fig. 8. Clustering based on correlation of client use of network application for the 2008 dataset. The methodology and labeling are the same as in Figure 7. Also included are 14 unknown applications with no formal port assignment, whose purpose we deduce from their correlation of use to other applications.

day. Not only are these users likely to be conduits of material from one P2P network to another, but any institutional attempts to manage P2P networks focus on a single application at their own peril.

Although this clustering mirrors our previous understanding of network applications, it is of little practical value unless it can be used to predict the nature of previously unknown applications. To explore the usefulness of our analysis, we used the 2005 application correlation data to classify the ports associated with 16 applications unknown to us at that time, either because of their obscurity or because their ports had not been formally assigned. Two such applications are included in Figure 7, represented by a "?" next to their port numbers. Port 388 is coupled most strongly with FTP and Hotline (an older P2P file-sharing application); subsequent investigation showed it to be assigned to "Unidata/LDM," a file transfer application used for moving large sets of meteorological data between research centers. Port 19101 was grouped with both instant messaging and P2P applications, suggesting that it might be a P2P application that relies on individual contact to initiate file transfers. This prediction allowed us to construct search engine queries to discover that this port is used by "Clubbox," a Korean file-sharing program that allows users to trade entire seasons of television programs on large virtual hard drives. In this case, sniffing the applica-

Table II. Predicted Uses of High-Traffic TCP Ports in the 2005 Dataset Running Lesser-Known Applications, Based on the Hierarchical Clustering Data; and Their Actual Uses, as Derived from Security Bulletins, Web Searches, Application Homepages, etc

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

The ports marked as "unknown" were in use only transiently and did not carry appreciable traffic on other days.

tion data would have been of little use to a network engineer unfamiliar with the Korean language; the application network gave us information that packet analysis alone would not.

The predictions based on clustering and the actual identities of the applications for all sixteen unknown applications in 2005 are shown in Table II. To verify or disprove these predictions, we consulted security bulletins, search engines, application homepages, and other related resources, in some cases locating information that would have been difficult to discover without the initial prediction. There were eight successful predictions, six partial predictions, and two predictions that cannot be verified. The partial predictions result from applications that were clustered with both P2P file-sharing applications and applications strongly associated with malicious activity (IRC and SQL Server). In these cases, we lacked sufficient data to make a judgment as to which purpose was more likely. In a practical application, network administrators would be advised to examine such cases closely. This ambiguity is also an indication that systems involved with P2P applications may be more likely to be compromised by malicious software, possibly through the P2P applications themselves. We could not verify our predictions for two of the ports because they were in use only transiently during our data collection period and no longer carry more traffic than any other randomly selected port.

We also note that while Web proxies predate the P2P file-sharing networks, their function is essentially that of a moderator between peers. A Web proxy draws data from a collection of information providers and then shares that same content with a community of users, making its traffic not only somewhat symmetric, but directly analogous to that of P2P applications that download a file (or parts of a file) and then share them with other users. Though they are not usually described as such, they are thus actually an early form of P2P application, so we do count this as a match for our prediction.

Because our technique met with substantial success in the 2005 dataset, we applied it to the 2008 dataset as well, selecting 14 more unknown applications and attempting to classify them based on their roles in the application network. These 14

Table III. Predicted Uses of High-Traffic TCP Ports in the 2008 Dataset, Using the Same Methodology as in 2005

Port	Predicted	Actual	Match?
50000	P2P, probably BitTorrent	BitTorrent	yes
16881	P2P, probably BitTorrent	BitTorrent	yes
51413	P2P, probably BitTorrent	BitTorrent	yes
5001	traditional file transfer	iperf and several trojans	partial
6714	traditional client/server	Internet Backplane Protocol	yes
36598	traditional client/server	*(unknown)*	unknown
6000	traditional client/server	X Window System	yes
19101	centralized file xfer svc.	file sharing (still ClubBox)	yes
9001	centralized file xfer svc.	Tor network	partial
2128	Web-related service	Net Steward dist. firewall	partial
3128	Web-related service	Squid Web cache	yes
3124	Web-related service	PlanetLab Web proxy network	yes
554	Web-related service	real-time streaming protocol	yes
1935	Web-related service	Macromedia Flash	yes

applications include port numbers that were also among the unknown applications in the 2005 dataset. We felt this to be appropriate because of the even greater rate of change for informally allocated ports: the 2005 guesses might no longer be valid in 2008. The results, shown in Table III, reflect a similar level of success: we had ten successful matches, three partial matches, and another instance of a port that was in transient use only. The partial match for port 5001 results from the dual use of this port for both *iperf* test traffic and several well-known backdoor applications. We regard the Tor network as only a partial match because it is a general-purpose network application that provides anonymous transit for any kind of data; while it is likely that much of this traffic involves sharing copyrighted media files, we cannot prove this. Our guess that the Net Steward distributed firewall system was a Web-related service provided the final partial match: while the firewall itself is not necessarily Web-related, the application includes a popular Web content filtering system.

We must caution not every network application is amenable to detection and classification through straightforward application of the techniques described in this section. A particular vulnerability lies in the assumption that port numbers remain stable during the execution of an application, whereas some modern peer-to-peer applications now migrate from port to port in order to evade traffic shaping and firewalls. One promising avenue of future research is to mitigate this vulnerability through link analysis: even if only a small proportion of the users participating in a BitTorrent swarm are using a common port number, it may be possible to infer the participation of the others from the overall pattern of connections.

7. CONCLUSIONS

Our first case study shows that a graph-centered view of network flow data reveals properties of user behavior that are essential for agent-based modeling of user populations. These properties affect applications such as Internet epidemiology, where highly nonuniform contact patterns among hosts affect the rate at which worms and viruses spread and which methods are most effective in combating them. Network design and capacity planning are also greatly affected: if there is no well-defined mean for the amount of traffic introduced by the users of the network, a service provider cannot easily estimate the incremental cost of each new customer. Broad-tailed distributions of traffic make it difficult to draw a line above which consumption of network resources

is excessive; when the standard deviation is two orders of magnitude greater than the mean, what behavior is truly aberrant?

This pervasive presence of distributions with extremely long tails implies that user behavior rarely follows the normal distributions that might be expected, but is actually so diverse as to defy characterization with a mean value. Superlinear behavior in Web clients especially demands that any behavioral models be able to account for nontrivial coupling of degree and strength. Furthermore, the differences observed between the Web and P2P application groups imply that behavioral analysis can yield statistical signatures for different types of application, allowing network managers to identify applications being run covertly on nonstandard ports. As mentioned previously, current network security products commonly employ rate-based thresholds to detect traffic anomalies. However, our results show that "normal" traffic would cause many false alarms, no matter what the threshold. We also demonstrate through our comparison of the two datasets that these statistics are consistent across a three-year time period, even as the character of the traffic on the network has changed from purely academic to substantially commercial. The analysis of behavioral networks may thus offer more effective methods of detecting malicious or otherwise anomalous behavior on the Internet.

The application clusters identified by the technique presented in our second study show that system administrators or network managers can easily infer traits of the activity carried out on a particular port, even if the application for that port is unknown or the port is being used covertly. They also give us an opportunity to group applications together by the way they affect the network rather than the origin of their codebase or even their nominal purpose. The potential of this approach becomes clear when we consider how these clusters evolve over time as people use the same applications and the same protocols in radically different ways. For example, because our clustering is based on aggregate behavior rather than pattern-matching against captured data, we were able to recognize the ports associated with new BitTorrent clients without ever examining the payload of a single packet.

The framework we present here offers a practical way of understanding the collective behavior of individual Internet users through analysis of the behavioral and application networks implicitly formed by their actions. Because we avoid any reliance on captured packets or nonanonymized flow data, a much wider audience of network researchers are able to test these techniques for themselves than has been the case with previous studies. Of course, network administrators can use the scalable techniques presented here in conjunction with more resource-intensive packet analysis applications, which could confirm suspicious activity suggested by network-based analysis of flow data.

None of the processing steps we describe requires extensive computing resources; a single high-end workstation can perform the analysis described in the first case study in less than half an hour. The analysis we describe in the second case study is quadratic in the number of applications considered, but even it can be performed in a fraction of the data collection time window.

Finally, we are working with the technology transfer office of our university to make our analysis tools publicly available.

ACKNOWLEDGMENTS

The authors would like to thank the Advanced Network Management Laboratory at Indiana University for support and infrastructure, as well as Internet2 for its generous polices on the use of anonymized network flow data. We also thank our anonymous reviewers for a variety of constructive suggestions that have strengthened this work.

REFERENCES

ADAMIC, L. A. AND HUBERMAN, B. A. 2001. The Web's hidden order. *Comm. ACM 44,* 9, 55–60.

ALDERSON, D., LI, L., WILLINGER, W., AND DOYLE, J. C. 2005. Understanding Internet topology: Principles, models, and validation. *IEEE/ACM Trans. Netw. 13,* 6, 1205–1218.

BARABÁSI, A.-L. AND ALBERT, R. 1999. Emergence of scaling in random networks. *Science 286*, 509–512.

BERNAILLE, L., TEIXEIRA, R., AND SALAMATIAN, K. 2006. Early application identification. In *Proceedings of the Conference on Emerging Network Experiment and Technology (CoNEXT).*

BRODER, A., KUMAR, S., MAGHOUL, F., RAGHAVAN, P., RAJAGOPALAN, S., STATA, R., TOMKINS, A., AND WIENER, J. 2000. Graph structure in the Web. *Comput. Netw. 33*, 1-6, 309–320.

CLAFFY, K. 1999. Internet measurement and data analysis: Topology, workload, performance and routing statistics. In *Proceedings of the National Academy of Engineering Workshop (NAE'99)*. CAIDA.

CLAFFY, K. 2006. A day in the life of the Internet: Proposed community-wide experiment. *ACM SIGCOMM Comput. Comm. Rev. 36*, 5, 39–40.

CROVELLA, M. AND KRISHNAMURTHY, B. 2006. *Internet Measurements: Infrastructure, Traffic and Applications*. Wiley & Sons.

EBEL, H., MIELSCH, L.-I., AND BORNHOLDT, S. 2002. Scale-Free topology of e-mail networks. *Phys. Rev. 66*, 035103.

ERMAN, J., ARLITT, M., AND MAHANTI, A. 2006. Traffic classification using clustering algorithms. In *Proceedings of the ACM SIGCOMM Workshop on Mining Network Data.* 281–286.

ERMAN, J., MAHANTI, A., ARLITT, M., AND WILLIAMSON, C. 2007. Identifying and discriminating between Web and peer-to-peer traffic in the network core. In *Proceedings of the International World Wide Web Conference (WWW).* 883–892.

ESTAN, C., SAVAGE, S., AND VARGHESE, G. 2003. Automatically inferring patterns of resource consumption in network traffic. In *Proceedings of the ACM SIGCOMM Conference.*

FABRIKANT, A., KOUTSOUPIAS, E., AND PAPADIMITRIOU, C. H. 2002. Heuristically optimized trade-offs: A new paradigm for power laws in the Internet. In *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP).*

FORREST, S., HOFMEYR, S., AND SOMAYAJI, A. 1997. Computer immunology. *Comm. ACM 40*, 10, 88–96.

GARETTO, M., GONG, W., AND TOWSLEY, D. 2003. Modeling malware spreading dynamics. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom).*

HUBERMAN, B. AND LUKOSE, R. 1997. Social dilemmas and Internet congestion. *Science 277*, 535.

HUBERMAN, B., PIROLLI, P., PITKOW, J., AND LUKOSE, R. 1998. Strong regularities in World Wide Web surfing. *Science 280*, 5360, 95–97.

HUFFAKER, B., FOMENKOV, M., MOORE, D., NEMETH, E., AND CLAFFY, K. 2000. Measurements of the Internet topology in the Asia-Pacific region. In *Proceedings of the Annual Conference of the Internet Society (INET'00)*. The Internet Society.

JIN, C., CHEN, Q., AND JAMIN, S. 2000. INET: Internet topology generators. Tech. rep. CSE-TR-433-00, Electrical Engineering and Computer Science Department, University of Michigan.

KARAGIANNIS, T., PAPAGIANNAKI, K., AND FALOUTSOS, M. 2005. BLINC: Multilevel traffic classification in the dark. In *Proceedings of the ACM SIGCOMM Conference.* 229–240.

KRIOUKOV, D., CLAFFY, K., FOMENKOV, M., CHUNG, F., VESPIGNANI, A., AND WILLINGER, W. 2007. The workshop on Internet topology (WIT) report. *ACM SIGCOMM Comput. Comm. Rev. 37*, 1, 69–73.

KUMAR, S., RAGHAVAN, P., RAJAGOPALAN, S., SIVAKUMAR, D., TOMKINS, A., AND UPFAL, E. 2000. Stochastic models for the Web graph. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science.* IEEE Computer Society Press, 57–65.

LAKHINA, A., CROVELLA, M., AND DIOT, C. 2004a. Characterization of network-wide anomalies in traffic flows. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference.*

LAKHINA, A., CROVELLA, M., AND DIOT, C. 2004b. Diagnosing network-wide traffic anomalies. In *Proceedings of the ACM SIGCOMM Conference.*

LAKHINA, A., PAPAGIANNAKI, K., CROVELLA, M., DIOT, C., KOLACZYK, E. D., AND TAFT, N. 2004c. Structural analysis of network traffic flows. In *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems.* 61–72.

LAURA, L., LEONARDI, S., MILLOZZI, S., MEYER, U., AND SIBEYN, J. F. 2003. Algorithms and experiments for the Webgraph. In *Proceedings of the European Symposium on Algorithms*.

LI, C. AND CHEN, C. 2007. Gnutella: Topology dynamics on phase space. Preprint cs/0702022.

LI, L., ALDERSON, D., WILLINGER, W., AND DOYLE, J. 2004. A first-principles approach to understanding the Internet's router-level topology. In *Proceedings of the ACM SIGCOMM Conference*. 3–14.

MEDINA, A. AND MATTA, I. 2000. BRITE: A flexible generator of Internet topologies. Tech. rep. BU-CS-TR-2000-005, Boston University.

MEISS, M., MENCZER, F., AND VESPIGNANI, A. 2005. On the lack of typical behavior in the global Web traffic network. In *Proceedings of the 14th International World Wide Web Conference*. 510–18.

MEISS, M., MENCZER, F., AND VESPIGNANI, A. 2007. A framework for analysis of anonymized network flow data. In *Proceedings of the NSF Symposium on Next Generation of Data Mining and Cyber-Enabled Discovery for Innovation*.

MEISS, M., MENCZER, F., FORTUNATO, S., FLAMMINI, A., AND VESPIGNANI, A. 2008a. Ranking Web sites with real user traffic. In *Proceedings of the 1st ACM International Conference on Web Search and Data Mining (WSDM)*.

MEISS, M., MENCZER, F., AND VESPIGNANI, A. 2008b. Structural analysis of behavioral networks from the Internet. *J. Phys. A: Math. Theor. 41*, 22.

MENCZER, F. 2002. Growing and navigating the small world Web by local content. *Proc. Nat. Acad. Sci. 99*, 22, 14014–14019.

MENCZER, F. 2004. The evolution of document networks. *Proc. Natl. Acad. Sci. 101*, 5261–5265.

MOORE, A. W. AND ZUEV, D. 2005. Internet traffic classification using Bayesian analysis techniques. In *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*. 50–60.

MOORE, D., VOELKER, G., AND SAVAGE, S. 2001. Inferring Internet denial of service activity. In *Proceedings of the USENIX Security Symposium*.

MOORE, D., SHANNON, C., AND BROWN, J. 2002. Code-Red: A case study on the spread and victims of an Internet worm. In *Proceedings of the 2nd Internet Measurement Workshop*.

NEWMAN, M. E. J., FORREST, S., AND BALTHROP, J. 2002. E-Mail networks and the spread of computer viruses. *Phys. Rev. E 66*, 035101.

PASTOR-SATORRAS, R. AND VESPIGNANI, A. 2001. Epidemic spreading in scale-free networks. *Phys. Rev. Lett. 86*, 3200–203.

PASTOR-SATORRAS, R. AND VESPIGNANI, A. 2004. *Evolution and Structure of the Internet*. Cambridge University Press, Cambridge, UK.

PATWARI, N., HERO III, A. O., AND PACHOLSKI, A. 2005. Manifold learning visualization of network traffic data. In *Proceedings of the ACM SIGCOMM Workshop on Mining Network Data*. 191–196.

RIPEANU, M., FOSTER, I., AND IAMNITCHI, A. 2002. Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Comput. 6*, 1, 50–57.

SAROIU, S., GUMMADI, P. K., AND GRIBBLE, S. D. 2002. A measurement study of peer-to-peer file sharing systems. In *Proceedings of the Multimedia Computing and Networking Conference (MMCN'02)*.

SHAVITT, Y., SUN, X., WOOL, A., AND YENER, B. 2004. Computing the unmeasured: An algebraic approach to Internet mapping. *IEEE J. Select. Areas Comm. 22*, 1, 67–78.

SINGH, S., ESTAN, C., VARGHESE, G., AND SAVAGE, S. 2004. Automated worm fingerprinting. In *Proceedings of the ACM/USENIX Symposium on Operating System Design and Implementation*.

STANIFORD, S., PAXSON, V., AND WEAVER, N. 2002. How to own the Internet in your spare time. In *Proceedings of the 11th USENIX Security Symposium (Security'02)*.

UHLIG, S. AND BONAVENTURE, O. 2001. The macroscopic behavior of Internet traffic: A comparative study. Tech. rep., Infonet-TR-2001-10, University of Namur.

WARD, J. H. 1963. Hierarchical grouping to optimize an objective function. *J. Amer. Statist. Assoc. 58*, 301, 236–244.

YOOK, S.-H., JEONG, H., AND BARABÁSI, A.-L. 2002. Modeling the Internet's large-scale topology. *Proc. Nat. Acad. Sci. 99*, 13382–13386.

ZHANG, Y., SINGH, S., SEN, S., DUFFIELD, N., AND LUND, C. 2004. Online identification of hierarchical heavy hitters: Algorithms, evaluation, and applications. In *Proceedings of the Internet Measurement Conference*. 101–114.

ZOU, C., TOWSLEY, D., AND GONG, W. 2004. Email worm modeling and defense. In *Proceedings of the 13th International Conference on Computer Communications and Networks (ICCCN'04)*.

Received June 2010; revised September 2010; accepted November 2010