Behavior Analysis of Internet Traffic via Bipartite Graphs and One-Mode Projections

Kuai Xu, *Member, IEEE, ACM*, Feng Wang, *Member, IEEE*, and Lin Gu, *Member, IEEE*

***Abstract*—As Internet traffic continues to grow in size and complexity, it has become an increasingly challenging task to understand behavior patterns of end-hosts and network applications. This paper presents a novel approach based on behavioral graph analysis to study the behavior similarity of Internet end-hosts. Specifically, we use bipartite graphs to model host communications from network traffic and build one-mode projections of bipartite graphs for discovering social-behavior similarity of end-hosts. By applying simple and efficient clustering algorithms on the similarity matrices and clustering coefficient of one-mode projection graphs, we perform network-aware clustering of end-hosts in the same network prefixes into different end-host behavior clusters and discover inherent clustered groups of Internet applications. Our experiment results based on real datasets show that end-host and application behavior clusters exhibit distinct traffic characteristics that provide improved interpretations on Internet traffic. Finally, we demonstrate the practical benefits of exploring behavior similarity in profiling network behaviors, discovering emerging network applications, and detecting anomalous traffic patterns.**

***Index Terms*—Behavior graph analysis, bipartite graph, clustering algorithms, one-mode projection, traffic profiling.**

I. Introduction

As INTERNET hosts and applications continue to grow, it becomes increasingly important to understand traffic patterns of end-hosts and network applications for efficient network management and security monitoring. A number of research studies [3]–[6] have focused on traffic behavior analysis of individual hosts and applications. However, an increasingly large number of end-hosts, a wide diversity of applications, and massive traffic data pose significant challenges for such fine-granularity analysis for backbone networks or enterprise networks.

This paper proposes a new approach of profiling traffic behavior by identifying and analyzing clusters of hosts or applications that exhibit similar communication patterns. With each cluster abstracting behavior patterns of a plurality of hosts or applications, the cost of traffic analysis is significantly reduced. We first use bipartite graphs to model network traffic of Internet backbone links or Internet-facing links of border routers in enterprise networks. As one-mode projections can effectively extract hidden relationships between nodes within the same vertex sets of bipartite graphs [7], we subsequently construct one-mode projections of bipartite graphs to connect source hosts that communicate with the same destination host(s) and to connect destination hosts that communicate with the same source host(s).

The derived one-mode projection graphs enable us to further build similarity matrices of Internet end-hosts, with similarity being characterized by the shared number of destinations or sources between two hosts. Based on the similarity matrices of end-hosts in the same network prefixes, we apply a simple yet effective spectral clustering algorithm to discover the inherent *end-host behavior clusters*. Each cluster consists of a group of hosts that communicate with similar sets of servers, clients, or peers. The behavior clusters not only reduce the number of behavior profiles for analysis compared to traffic profiling on individual hosts, but also reveal detailed behavior patterns for a group of end-hosts in the same network prefixes.

Similarly we use a vector of graph properties including clustering coefficient to capture the similarity of traffic behavior for end-hosts engaging in the same Internet applications and discover the inherent *application behavior clusters*, each of which consists of a number of applications. For each application cluster, we examine characteristics of the aggregated traffic, such as host symmetry, the fan-out degree of source IP addresses, and the fan-in degree of destination IP addresses. The experimental results based on real Internet traffic confirm that application behavior clusters indeed capture applications with similar traffic characteristics and behavior patterns.

To demonstrate the practical benefits of end-host behavior clusters and application behavior clusters, we show how behavior clusters could be used to discover emerging applications and detect anomalous traffic behavior such as scanning activities, worms, and denial-of-service (DoS) attacks through synthetic traffic traces that combine IP backbone traffic and real scenarios of worm propagations and denial-of-service attacks. Thus, our proposed technique could become a valuable tool for network operators to gain a deep understanding of network traffic and to detect traffic anomalies.

The contributions of this paper are summarized as follows.

- We use bipartite graphs to represent communication patterns between source and destination hosts, and construct one-mode projection graphs to capture behavior similarity of Internet end-hosts.

Manuscript received January 17, 2012; revised May 26, 2012; October 01, 2012; February 14, 2013; and May 01, 2013; accepted May 12, 2013; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Kasera. Date of publication June 05, 2013; date of current version June 12, 2014. This work was supported in part by the NSF under Grant CNS-1218212, ASU under an SRCA grant, and the State Key Laboratory of Advanced Optical Communication Systems and Networks, Shanghai, China, under Grant 2011GZKF030902.

K. Xu and F. Wang are with the School of Mathematical and Natural Sciences, Arizona State University, Glendale, AZ 85306 USA (e-mail: kuai.xu@asu.edu).

L. Gu is with the Hong Kong University of Science and Technology, Kowloon, Hong Kong.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2013.2264634

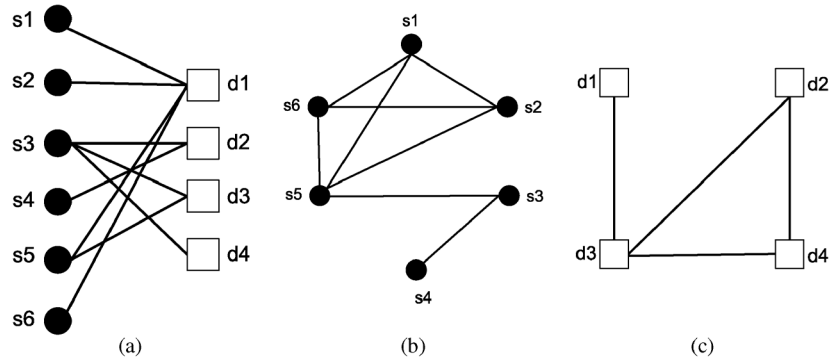


Fig. 1. Modeling host communications using bipartite graphs and one-mode projection graphs. (a) Example of bipartite graphs. (b) One-mode projection graph of source hosts. (c) One-mode projection graph of destination hosts.

- We explore behavior similarity of Internet end-hosts in the same network prefixes using clustering algorithms and discover the distinct *end-host behavior clusters*.
- We apply clustering algorithms to group Internet applications into distinctive *application behavior clusters* based on clustering coefficient and other graph properties of one-mode projection graphs built from the underlying network traffic.
- We demonstrate practical benefits of exploring behavior similarity of Internet end-hosts in profiling network prefixes and emerging applications and detecting anomalous traffic patterns such as scanning activities, worms, or denial-of-service attacks through synthetic traffic traces.

This paper is organized as follows. Section II discusses bipartite graphs for modeling data communication in network traffic and the one-mode projection for capturing behavior similarity of end-hosts. Section III describes the similarity matrices and clustering coefficient of one-mode projection graphs, while Section IV uses clustering algorithms to leverage similarity matrices and clustering coefficient for discovering behavior clusters of end-hosts in the same prefixes or engaging in the same applications. Section V presents the distinct characteristics of end-host behavior clusters within the same network prefixes and uses behavior similarity to discover traffic patterns in network prefixes and detect anomalous behaviors. Section VI explores behavior clusters of network applications for detecting emerging applications and anomalous traffic patterns. Section VII discusses related work, and Section VIII concludes this paper.

II. Modeling Host Communications With Bipartite Graphs and One-Mode Projections

A. Bipartite Graphs of Host Communications

Host communications observed in network traffic of Internet backbone links or Internet-facing links of border routers for enterprise networks could be naturally modeled with a bipartite graph $\mathcal{G} = \mathcal{A}, \mathcal{B}, \mathcal{E})$, where $\mathcal{A}$ and $\mathcal{B}$ are two disjoint vertex sets, and $\mathcal{E} \subseteq \mathcal{A} \times \mathcal{B}$ is the edge set [7]. Specifically, all the `source` IP addresses observed in network traffic from one single direction of an Internet backbone link form the vertex set $\mathcal{A}$, while the vertex set $\mathcal{B}$ consists of all the `destination` addresses observed in the same traffic. Each of the edges, e_k in $\mathcal{G}$ connects one vertex $a_i \in \mathcal{A}$ and another vertex $b_j \in \mathcal{B}$. Note that an Internet backbone link carries network traffic from two directions, thus we separate network traffic based on traffic directions and use bipartite graphs to model network traffic from two directions separately.

To analyze the traffic behavior for network prefixes that include end-hosts with the same network bits in their IP addresses, we could further decompose the bipartite graph of all the traffic into a set of smaller disjoint bipartite subgraphs such that each bipartite subgraph captures the host communications for a single source or destination IP prefix, e.g., *source behavior graph* (SBG) $\mathcal{G}_{\mathcal{P}} = (\mathcal{A}_{\mathcal{P}}, \mathcal{B}, \mathcal{E}_{\mathcal{P}})$ and *destination behavior graph* (DBG) $\mathcal{G}_{\mathcal{Q}} = (\mathcal{A}, \mathcal{B}_{\mathcal{Q}}, \mathcal{E}_{\mathcal{Q}})$ representing the bipartite subgraphs of host communications for the source IP prefix $\mathcal{P}$ and the destination IP prefix $\mathcal{Q}$, respectively.

Similarly, for a given application port, its traffic also forms a natural subgraph of the bipartite graph. Let $\mathcal{A}_{\texttt{port\_number}}$ and $\mathcal{B}_{\texttt{port\_number}}$ denote the sets of source and destination IP addresses engaging in the application port `port_number`, respectively. Then, we could build two bipartite subgraphs SBG $\mathcal{G}_{\texttt{srcport}} = (\mathcal{A}_{\texttt{srcport}}, \mathcal{B}, \mathcal{E}_{\texttt{srcport}})$ and DBG $\mathcal{G}_{\texttt{dstport}} = (\mathcal{A}, \mathcal{B}_{\texttt{dstport}}, \mathcal{E}_{\texttt{dstport}})$ for representing host communications for the source port `srcport` and the destination port `dstport`, respectively.

B. One-Mode Projections of Bipartite Graphs

To study the social-behavior similarity of end-hosts in network traffic, we leverage one-mode projection graphs of bipartite graphs that are used to extract hidden information or relationships between nodes within the same vertex sets [7]. Fig. 1(a) illustrates an example of a simple bipartite graph that shows data communications between six source IP addresses $(s_1 - s_6)$ and four destination IP addresses $(d_1 - d_4)$. Fig. 1(b) illustrates the one-mode projection of the bipartite graph on the vertex set of the six left-side nodes, i.e., the source hosts $(s_1 - s_6)$, while Fig. 1(c) is the one-mode projection on the four destination addresses. An edge connects two nodes in the one-mode projection if and only if both nodes have connections to at least one same node in the bipartite graph. Thus, studying one-mode projection graphs could potentially reveal interesting traffic patterns of Internet end-hosts and network applications.

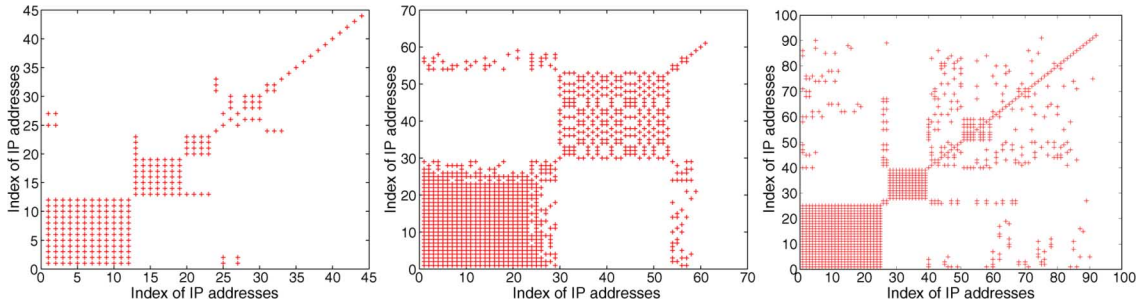


Fig. 2. Visualization of the adjacency matrix for one-mode projections of bipartite graphs for three network prefixes.

The one-mode projection of the bipartite graphs uses edges between end-hosts in the same network prefixes or engaging in the same application to quantify the similarity of their network connection patterns. For example, in Fig. 1(b), the edge between s_1 and s_2 reflects the observation that both s_1 and s_2 talk with the same destination host d_1 in the bipartite graph [Fig. 1(a)], and the edge between d_2 and d_3 in Fig. 1(c) captures the observation that s_3 talks with both destinations d_2 and d_3. Therefore, given a bipartite graph $\mathcal{G}_{\mathcal{P}} = (\mathcal{A}_{\mathcal{P}}, \mathcal{B}, \mathcal{E}_{\mathcal{P}})$ for a source prefix $\mathcal{P}$, we could construct the one-mode projection graph of *SBG on source prefix* $\mathcal{P}$, $\mathcal{G}'_{\mathcal{A}_{\mathcal{P}}} = (\mathcal{A}_{\mathcal{P}}, \mathcal{E}'_{\mathcal{A}_{\mathcal{P}}})$, where $\mathcal{A}_{\mathcal{P}}$ consists of all source hosts observed in $\mathcal{P}$ and $\{p_i, p_j\} \in \mathcal{E}'_{\mathcal{A}_{\mathcal{P}}}$ if and only if two hosts p_i and p_j talk with at least one same destination host. The similar process could generate the one-mode projection graph of *DBG on destination prefix* $\mathcal{Q}$ for any destination prefix $\mathcal{Q}$ as well. Using the same approach, we could build one-mode projection graphs of *SBG on port* `srcport` and of *DBG on port* `dstport`. In this study, we leverage one-mode projection graphs to explore the social-behavior similarity of source or destination IP addresses that share the same network prefixes or engage in the same Internet applications.

III. Similarity Matrices and Clustering Coefficient of One-Mode Projection Graphs

A. Similarity Matrices

To capture the information on the degree of the social-behavior similarity among end-hosts, we use the normalized weight for the edges in the one-mode projection graph. Let $\mathcal{N}_{p_i}$ and $\mathcal{N}_{p_j}$ represent the numbers of Internet hosts with which two hosts p_i and p_j in the prefix $\mathcal{P}$ have communicated, respectively. We then use $w_{\{p_i,p_j\}}$ to denote the weight for the edge between p_i and p_j in the one-mode projection

$$w_{\{p_i,p_j\}} = \frac{|\mathcal{N}_{p_i} \bigcap \mathcal{N}_{p_j}|}{|\mathcal{N}_{p_i} \bigcup \mathcal{N}_{p_j}|} \tag{1}$$

where $|\mathcal{N}_{p_i} \bigcap \mathcal{N}_{p_j}|$ denotes the total number of the shared destination hosts in the bipartite graph between the two hosts p_i and p_j, and $|\mathcal{N}_{p_i} \bigcup \mathcal{N}_{p_j}|$ denotes the total number of the uniquely combined destinations of p_i and p_j. Note that $w_{\{p_i,p_i\}} = 1$. The weighted adjacency matrix of the one-mode projection graph for the network prefix $\mathcal{P}$ then becomes $\mathcal{M}_{\mathcal{P}} = (m_{i,j})_{|\mathcal{P}|\times|\mathcal{P}|}$, $m(i,j) = w_{\{p_i,p_j\}}$. The similar process could lead to the weighted adjacency matrices $\mathcal{M}_{\mathcal{Q}}$, $\mathcal{M}_{\texttt{srcport}}$, and $\mathcal{M}_{\texttt{dstport}}$ of the one-mode projection graph for the destination prefix $\mathcal{Q}$, the source port `srcport`, and the destination port `dstport`, respectively.

One interesting observation of the one-mode projection graphs for host communications lies in the clustered patterns in the weighted adjacency matrix. The scatter plots in Fig. 2 visualize the adjacency matrices of the one-mode projection graphs for three different network prefixes with 44, 61, and 92 end-hosts, respectively. For each prefix, we sort the IP addresses based on the hosts' degree (number of neighbors in the one-mode project graph) in a nonincreasing order. Both x-axis and y-axis represent the indices of IP addresses in the same prefix, and each "+" point (i, j) in the plots denotes an edge with a positive weight between two sorted hosts p_i and p_j in the one-mode projection graph, i.e., $m(i, j) = w_{\{p_i,p_j\}} > 0$. As shown in Fig. 2, each prefix has a few well-separated blocks that divide end-hosts into different clusters. This observation on the adjacency matrix motivates us to further explore clustering techniques and graph partitioning algorithm [8] to uncover these behavior clusters of end-hosts that share the same network prefixes or engage in the same Internet applications.

B. Clustering Coefficient

Clustering coefficient is a widely used measure to study the "closeness" or the "small-world" patterns of nodes in one-mode project graphs [9]. This measure can be applied to individual nodes as *local clustering coefficient* (LCC) and can also be applied to the entire graph as *global clustering coefficient* (GCC). For a given node u, the local clustering coefficient LCC_u is provided by the number of the edges among u's neighbors over the number of all possible edges among u's neighbors. Let N_u represent the set of all the neighbors of the node u, where $|N_u| = m$, and let E_u represent the set of edges among these neighbors. The number of all possible edges among m neighbors is $m \times (m-1)/2$. The LCC of u is calculated as

$$\mathrm{LCC}_u = \frac{|E_u|}{\frac{(m*(m-1))}{2}} = \frac{|E_u| * 2}{m * (m-1)}. \tag{2}$$

Clearly, $\mathrm{LCC}_u \in [0, 1]$. LCC_u is 0 if there is no edge among u's neighbors, while LCC_u is 1 if u's neighbors form a complete graph (clique). Note that the LCC for nodes with 0 or 1 neighbor is 0 due to zero edges. The GCC of the entire graph, GCC_G is the average LCC over all n nodes, where $\mathrm{GCC}_G = 1/N * \sum \mathrm{LCC}_u$, where $u \in G$. Because of the existence of nodes with

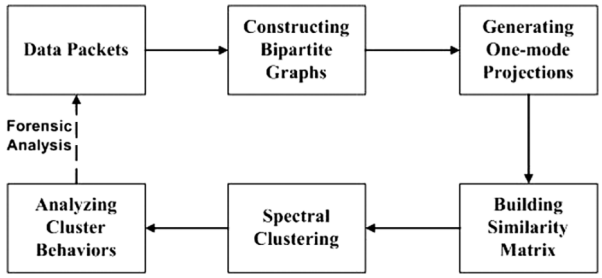


Fig. 3. Schematic process of network-aware behavior clustering algorithm for discovering behavior clusters of network prefixes.

0 or 1 neighbor that affects the calculation of the global clustering coefficient, we adopt an adaptive global clustering coefficient (AGCC), introduced in [10], $\mathrm{AGCC}_G = 1/1-\theta * \mathrm{GCC}_G$, where θ is the percentage of the isolated nodes in one-mode project graphs. In addition, we also measure the percentage of the nodes that have at least two neighbors (or nonisolated nodes) in the graphs. In Section VI, we will show how clustering coefficient captures social behavior of source or destination IP addresses engaging in the same applications with real network traffic datasets and helps uncover groups of Internet applications exhibiting similar traffic patterns.

IV. Discovering Behavior Clusters via Clustering Algorithms

A. Partitioning Similarity Matrix With Spectral Clustering Algorithm

In this study, we focus on the social behavior of end-hosts in data communications through bipartite graphs and one-mode projection graphs, and we are interested in exploring the social-behavior similarity of end-hosts to discover inherent traffic clusters. Fig. 3 illustrates the schematic process of our clustering approach from constructing bipartite graphs based on IP packets to discovering and analyzing behavior clusters of network prefixes.

An important starting point of a clustering algorithm is to define the appropriate similarity matrix between data points. In this paper, we use the weighted edge between two hosts u and v of the same prefix in the one-mode projection graph as the similarity measure $s_{u,v}$ between u and v because the weighted edges capture and quantify the social-behavior similarity of host communications in network traffic. Therefore, the weighted adjacency matrix of the one-mode projection graphs for the prefix $\mathcal{M}_{\mathcal{P}}$ essentially becomes the similarity matrix $\mathcal{S}_{\mathcal{P}}$, which will be used as an input to the spectral clustering algorithm outlined here.

This study applies a simple spectral clustering algorithm developed in [8] due to its wide applications in graph partitioning and its small running time. The original spectral clustering algorithm [8] requires an explicit input of k as the expected number of clusters. Given the infeasibility of predicting the optimal number of behavior clusters in network prefixes without analyzing the traffic data, we therefore augment the algorithm by adding a step of automatically selecting an appropriate value of k as the desired number of the clusters based on the eigenvalue distribution. The detail of this step is explained in the following algorithm.

Algorithm 1: Algorithm of discovering behavior clusters using an augmented spectral clustering algorithm

Input: network flow traces during a given time window and a source or destination prefix $\mathcal{P}$;
1: Construct bipartite graphs of host communications from flow traces;
2: Generate the one-mode projection of bipartite graphs and its weighted adjacency matrix $\mathcal{M}_{\mathcal{P}}$ for end-hosts in the prefix $\mathcal{P}$, and then obtain the similarity matrix $\mathcal{S}_{\mathcal{P}} \in \mathbb{R}^{n \times n}$ for the prefix $\mathcal{P}$;
3: Let A be the diagonal matrix with $A(i,i) = \sum_{j=1}^{n} s_{i,j}$, where $i = 1, \ldots, n$;
4: Compute the Laplacian matrix $L = A^{-1/2} S A^{-1/2}$;
5: Find the largest k eigenvalues, $\lambda_1, \lambda_2, \cdots, \lambda_k$ such that $\sum_{i=1}^{k} \lambda_i \geq \alpha \times \sum_{j=1}^{n} \lambda_n$ and $(\lambda_k - \lambda_{k+1}) \geq \beta \times (\lambda_{k-1} - \lambda_k)$;
6: Use the corresponding k eigenvectors $(e_1, e_2, \cdots, e_k)$ as columns to construct the matrix $E = [e_1 e_2 \cdots e_k] \in \mathbb{R}^{n \times k}$;
7: Construct the matrix $\mathcal{Z}$ through renormalizing E such that each row has a unit length, and consider each row as a point;
8: Run k-means clustering algorithm to cluster the points of $\mathcal{Z}$ into k clusters $(\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_k)$
9: Assign the original IP address p_i to the cluster $\mathcal{C}_j$ if the row i of $\mathcal{Z}$ is assigned to the cluster $\mathcal{Y}_j$.
Output: clusters $\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_k$, where $\mathcal{C}_i = \{p_j | z_j \in \mathcal{Y}_j\}$.

Algorithm 1 outlines the major steps of the spectral clustering algorithm with the augmented change of automatically selecting k clusters based on the traffic patterns. The input of this algorithm is network flow traces during a given time window and a source or destination prefix $\mathcal{P}$. The first step is to use flow traces to construct bipartite graphs of host communications, while the second step is to generate the one-mode projection of bipartite graphs and its weighted adjacency matrix $\mathcal{M}_{\mathcal{P}}$ for end-hosts in the prefix $\mathcal{P}$, and then to obtain the similarity matrix $\mathcal{S}_{\mathcal{P}} \in \mathbb{R}^{n \times n}$.

Next, we compute the Laplacian matrix $L = A^{-1/2} S A^{-1/2}$, where A is the diagonal matrix with $A(i,i) = \sum_{j=1}^{n} s_{i,j}$ and $i = 1, \ldots, n$. Then, in the augmented step, we search for the largest k eigenvalues, $\lambda_1, \lambda_2, \cdots, \lambda_k$ such that $\sum_{i=1}^{k} \lambda_i \geq \alpha \times \sum_{j=1}^{n} \lambda_n$ and $(\lambda_k - \lambda_{k+1}) \geq \beta \times (\lambda_{k-1} - \lambda_k)$. In other words, the augmented step searches an appropriate value for k by finding the largest k eigenvalues that account for at least α of the total variances and stopping at the eigenvalue λ_k where the distribution of eigenvalues exhibits a sharp slope change. In our experiments, we have evaluated a variety of values for α and β and found that there are not significant changes for α in the range of [0.8, 0.95] and β in the range of [1.5, 2.5]. For example, Fig. 4(a) shows the similar numbers of discovered clusters by the proposed algorithm for all source network prefixes during a 1-min time window with $\beta = 2$ and α being set as 0.8, 0.85, 0.09, and 0.95, respectively, while Fig. 4(b) also shows the

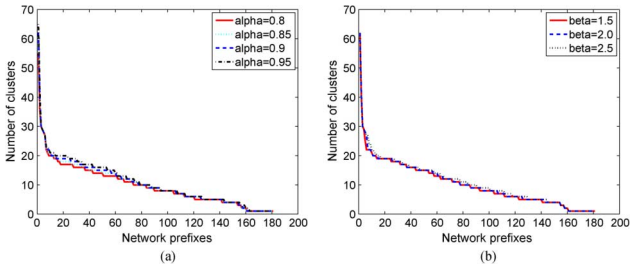


Fig. 4. Sensitivity analysis of (a) α and (b) β used in the proposed algorithm of discovering behavior clusters.

similar numbers of clusters for the same set of network prefixes with $\alpha = 0.9$ and β being set as 1.5, 2.0, and 2.5, respectively. Thus, in the remainder of this paper, we use 0.9 and 2 for the α and β, respectively, to present the experimental results.

In our experiment with real traffic traces, we find that it is common to observe that a few eigenvectors account for the majority of the variances in the similarity matrix for IP prefixes. Thus, we use the corresponding top k eigenvectors $(e_1, e_2, \cdots, e_k)$ as columns to construct the matrix $E = [e_1 e_2 \cdots e_k] \in \mathbb{R}^{n \times k}$, and subsequently construct the matrix $\mathcal{Z}$ through renormalizing E such that each row has a unit length. Considering each row as a point, the final step of the algorithm is to run a k-means clustering algorithm to cluster the points of $\mathcal{Z}$ into k clusters $(\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_k)$, and then assign the original IP address p_i to the cluster $\mathcal{C}_j$ if the row i of $\mathcal{Z}$ is assigned to the cluster $\mathcal{Y}_j$.

The output of this algorithm is a set of k clusters $(\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_k)$, each of which includes a group of end-hosts sharing similar social-behavior patterns in network traffic. In Section V, we will study traffic characteristics of end-host behavior clusters discovered by the spectral clustering algorithm, and then demonstrate the practical benefits of these clusters for discovering traffic patterns and detecting anomalous behaviors.

B. Clustering Analysis of Internet Applications

For the source and destination behavior graphs generated from the traffic of each Internet application, we calculate the *adaptive global clustering coefficient*. In this study, we consider a unique combination of port number and transport protocol (TCP or UDP) as one Internet application. For example, all network traffic on port 80/TCP is considered as an Internet application. In addition, we focus on network applications with consistent port numbers. Some applications, e.g., peer-to-peer file sharing that uses random port numbers to obfuscate their traffic behavior, require additional information, e.g., packet payload and hosts with labeled traffic patterns, to study social behavior of source and destination hosts. In the rest of this paper, we refer to the *adaptive global clustering coefficient* as *clustering coefficient* for simplicity. Fig. 5 shows the distribution of clustering coefficient for all Internet applications observed from an OC-192 Internet backbone link during a 1-min time window. An interesting observation is the clustered pattern of clustering coefficient, which leads to our next step of applying clustering algorithms to discover the inherent clusters formed by Internet applications sharing similar behavior patterns.

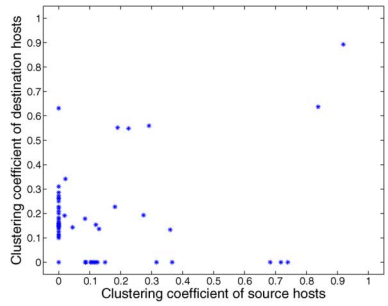


Fig. 5. Distribution of adaptive global clustering coefficient for source and destination behavior graphs of Internet applications.

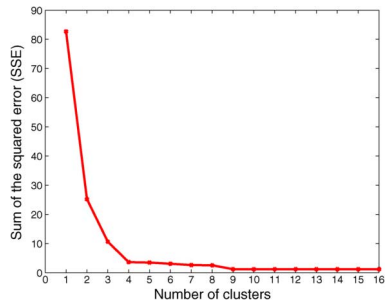


Fig. 6. Determining the optimal k based on SSE.

Based on clustering coefficient and other graph properties of Internet applications, we apply a simple K-means clustering algorithm [11] to group them into distinct application behavior clusters. The choice of selecting this algorithm is due to its simplicity and wide usage. The features used in the clustering algorithm include clustering coefficients of source and destination behavior graphs, and the ratios of nodes with two or more neighbors in these graphs. In other words, for each source or destination port p, we obtain a vector of four features, i.e., $\mathrm{AGCC}_{\mathcal{S}_p}$, $\mathrm{AGCC}_{\mathcal{D}_p}$, $r_{\mathcal{S}_p}$, $r_{\mathcal{D}_p}$, where the first two features are clustering coefficients of source and destination hosts engaging in the application port p, and the last two features are ratios of hosts with at least two or more neighbors in one-mode project graphs on source and destination hosts.

A challenging issue of applying K-means clustering algorithms is to find an optimal value of k since the choice of k plays an important role of archiving the high quality of clustering results. Toward this end, we search the optimal value of k by running K-means algorithms using a variety of k values and evaluate the best choice of k by comparing the sum of squared error (SSE) with Euclidian distance function between nodes in each cluster [11]. For example, Fig. 6 illustrates the distribution of SSE with varying values of k from 1 to 16. We select $k = 9$ as the choice since increasing k from 9 to 10 and above does not bring significant benefits of reducing SSE.

V. Making Sense of End-Host Behavior Clusters

A. Datasets

The datasets used in our analysis are collected from CAIDA's equinix-chicago and equinix-sanjose network monitors [12] on bidirectional OC-192 Internet backbone links of a large Internet

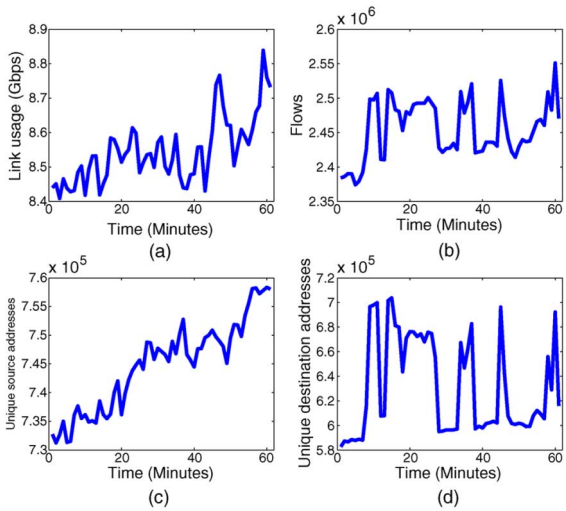


Fig. 7. Statistics of CAIDA Internet traffic trace. (a) Link usage. (b) Flows. (c) Source addresses. (d) Destination addresses.

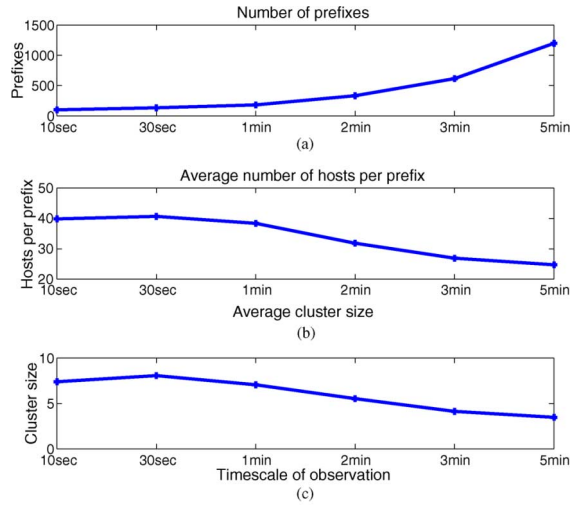


Fig. 8. Observations with using varying timescales to analyze 1-h traffic data: (a) number of prefixes; (b) average number of hosts per prefixes; (c) average cluster size.

service provider during December 17, 2009. The CAIDA Internet traffic traces are anonymized using CryptoPAn *prefix-preserving* anonymization [13] for privacy reasons, however such *prefix-preserving* process does not affect our analysis that explores behavior similarity of end-hosts within the same network prefixes or engaging in the same Internet applications.

Similar to the observations in previous studies [14], Internet backbone links carry large volumes of network traffic, which poses a challenging problem for real-time or near real-time traffic analysis. The total size of the compressed dataset used in this study is over 200 GB. As a first step to reduce the data size, we aggregate packet traces into the well-known 5-tuple network flows. Fig. 7(a) shows an average of 8.6 Gb/s link usage during a 1-h duration, and Fig. 7(b) illustrates millions of network flows for every minute during this period. In our analysis, we use a 1-min time bin to analyze traffic data due to vast amounts of packets and flows to be processed in each time bin. In addition, Fig. 7(c), (d) show the total numbers of unique source and destination IP addresses, respectively. Such a large number of unique IP addresses in the packet traces makes it very challenging to analyze traffic behavior at host level [3], therefore the focus on behavior clusters of network prefixes becomes an intuitive alternative for scalable analysis on Internet backbone traffic.

In our analysis, we use the /24 block as the network prefix granularity for analysis for two reasons. First, /24 is a common block size of BGP routing prefixes based on the observations on BGP routing tables. Based on the block size distribution of BGP prefixes in a recent snapshot of BGP routing table from the RouteView project [15], the /24 blocks account for over 50% of all the total prefixes on the Internet. In addition, multiple /24 prefixes could form larger prefixes by prefix aggregations. For example, two neighboring /24 prefixes could form /23 prefixes, thus the clusters identified in these two /24 prefixes could become separate clusters or be merged together to form a large cluster due to common traffic behavior. Second, the prefix-preserving anonymization process makes it impractical to aggregate IP addresses into real BGP prefixes or larger network prefixes. On the other hand, our proposed algorithm could be applied to BGP prefixes if data packets are not anonymized in other datasets. Our analysis is applied to both source and destination prefixes since the bipartite graphs and one-mode projection graphs in the previous sections could be established for both sides.

To determine an appropriate timescale for analyzing network traffic, we run the proposed algorithms with six different timescales including 10 s, 30 s, 1 min, 2 min, 3 min, and 5 min. Fig. 8(a)–(c) illustrates the number of prefixes, the average number of hosts per prefix, and the average number of clusters per prefix, respectively, for these timescales. Apparently, the number of prefixes increases as a result of increasing scale of observations. However, the average number of hosts and clusters per prefix tend to decrease when the timescale increase from 1 min to 2 and more minutes. Our in-depth study reveals that during the longer time windows we tend to observe more single-packet and short-lived flows to a smaller number of random hosts in the same network prefixes due to pervasive scanning activities on the Internet. During time windows with a smaller timescale, we mainly observe normal multiple-packet and long-lived flows such as traffic from server farms of popular Web sites and video streaming services. As a result, the decreasing number of hosts per prefix leads to smaller traffic clusters. The three timescales 10 s, 30 s, and 1 min have the highest numbers of average cluster size. In addition, Fig. 9 shows the percentage of hosts in the first cluster over all hosts in the prefix across varying timescales. As the timescale of observation increases, each prefix tends to include additional hosts in the prefix that do not share similar traffic behavior with other hosts. In other words, the increased timescale of observation leads to an increased number of clusters with one or a few hosts. As shown in Fig. 9, the timescales 10 s, 30 s, and 1 min have the highest percentages of hosts in the top cluster. Thus, we consider these three timescales as good candidates for appropriate timescales for traffic analysis.

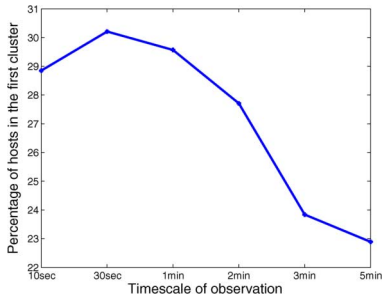


Fig. 9. Percentage of hosts in the first cluster over all hosts in the prefix.

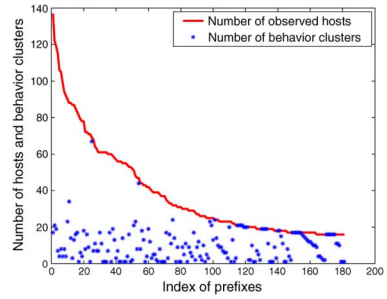


Fig. 11. Number of observed hosts and behavior clusters in all the prefixes with at least 16 hosts during 1-min time window.

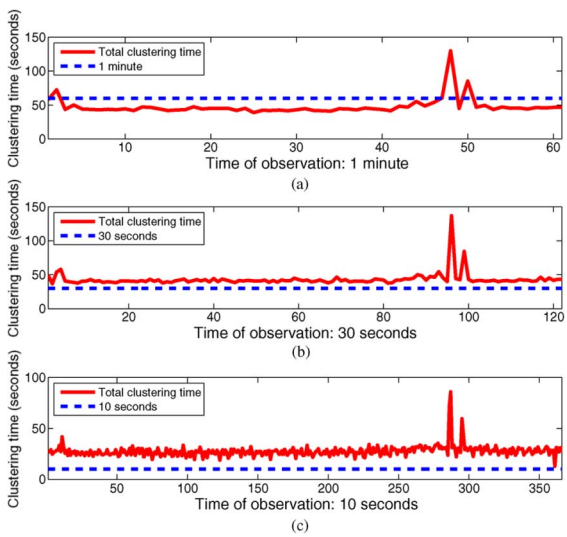


Fig. 10. Running time for clustering source and destination IP prefixes for a variety of timescales: (a) 1 min; (b) 30 s; (c) 10 s.

To evaluate the operational feasibility of the clustering algorithm, we run the clustering process on a commodity Linux server with a 2.93-GHz CPU and 2G memory using the traffic data. Fig. 10 illustrates the running time of the clustering process in discovering end-host behavior clusters of both source and destination network prefixes for three timescales: 10 s, 30 s, and 1 min, respectively. In average, it takes 27.5, 42.9, and 47.8 s to complete the clustering process for both source and destination IP prefixes observed in 10-s, 30-s, and 1-min timescales, respectively. The clustering step is able keep up with the continuous input of 1-min traffic data, but is unable to keep up with the input of 30- or 10-s traffic data. Thus, we choose 1 min as the timescale for our further analysis.

B. Distinct Traffic Characteristics of Behavior Clusters

The network-aware behavior clustering of end-hosts shifts traffic analysis from host-level to prefix-level clusters and increases the granularity of traffic analysis compared to host-level traffic profiling, thus it could successfully reduce the number of behavior profiles for analysis. Fig. 11 illustrates the size of the prefixes with at least 16 end-hosts and the number of their clusters during a 1-min time window. As we can see, the number of clusters is much smaller than the size of prefixes, as each behavior cluster groups many end-hosts together due to their common social-behavior patterns. This observation holds for other time windows as well. From Fig. 11, it is also interesting to see that there exists little correlation between the number of observed hosts and the number of behavior clusters. The number of behavior clusters for an IP prefix largely depends on the similarity of the social behavior patterns among the observed hosts, rather than the count of observed hosts. For example, the IP prefixes of data center networks that include hundred of servers tend to have less diverse behavior, while the IP prefixes of residential Internet service providers could have more diverse behavior since the hosts in residential networks could have very different communication patterns.

After obtaining separate behavior clusters, the next question we ask is the following: *Do end-host behavior clusters indeed exhibit distinct traffic characteristics?* Toward answering this question, we study the distributions of traffic features in each of behavior clusters, and then compare them to the aggregated traffic of the prefixes. We use an information-theoretic measure, relative uncertainty (RU) introduced in [14], to analyze the traffic features in individual clusters and the aggregated traffic. Given a variable X with a probability distribution, $p(x_i) = n_i/n, x_i \in X, i = 1, 2, \cdots, m$, where n_i is the number of times X is observed with the value x_i, the relative uncertainty (RU) on the variable X is defined as follows:

$$\mathrm{RU}(X) = \frac{H(X)}{H_{max}(X)} = \frac{H(X)}{\log m} \tag{3}$$

where the entropy, $H(X)$, is defined as $H(X) = -\sum_{x_i \in X} p(x_i) \log p(x_i)$, and $H(X)$ measures the variety or diversity in the observed values of X [16]. The relative uncertainty value $\mathrm{RU}(X)$ quantifies the randomness or uniqueness of the observed values. In general, $\mathrm{RU}(X)$ being 1 or approximately to 1 shows that the observed values of X are closer to being uniformly distributed, while $\mathrm{RU}(X)$ being 0 or approximately to 0 indicates that the values of X are concentrated to one or a few frequently observed values [14].

Our results show that behavior clusters separate different traffic patterns of the same prefixes for improved understanding and interpretation. Fig. 12 shows the distribution of relative uncertainty on destination IP addresses, source ports, and destination ports, respectively, for all the source prefixes and their behavior clusters during a 1-min time window. Compared to relative uncertainty values for network prefixes, the behavior clusters have much larger percentages of relative uncertainty values on all of these features being 0 and 1 or approximately being 0 and 1, which reveal concentrated patterns on a few

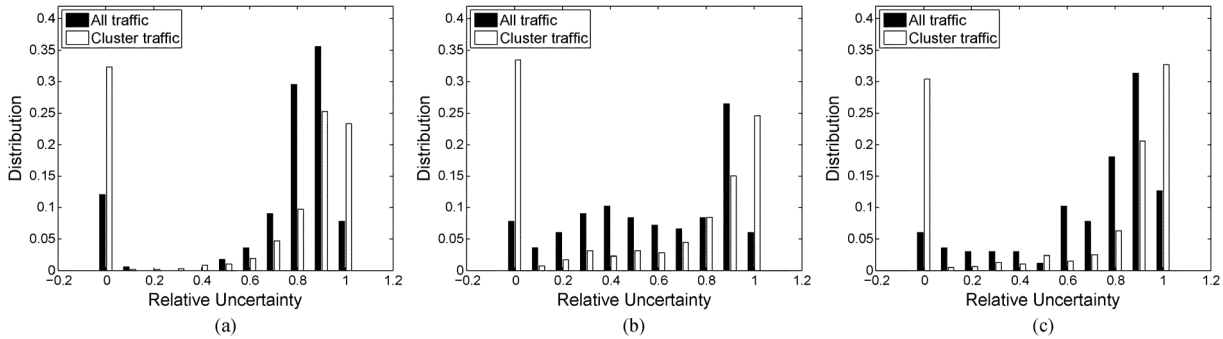


Fig. 12. Histograms of relative uncertainty distributions for behavior clusters and the aggregated traffic. (a) Destination IP addresses. (b) Source ports. (c) Destination ports.

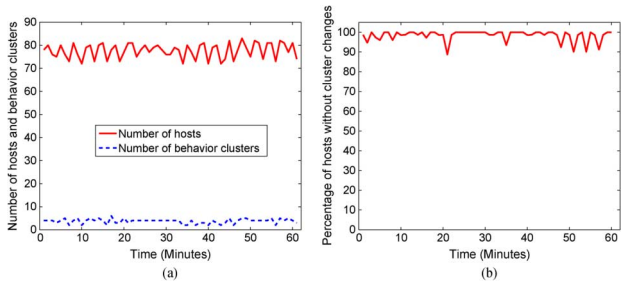


Fig. 13. Temporal stability of behavior clusters in a network prefix. (a) Number of hosts and behavior clusters over time. (b) Percentage of end-hosts without changing clusters.

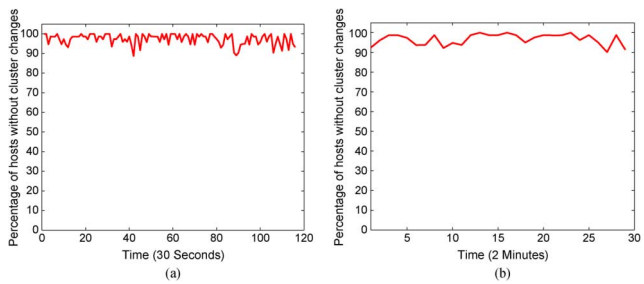


Fig. 14. Temporal stability of behavior clusters in a network prefix for different timescales. (a) Percentage of end-hosts without changing clusters (30 s). (b) Percentage of end-hosts without changing clusters (2 min).

ports and IP addresses or random patterns on ports and addresses. This result shows that the clustering algorithm extracts behavior clusters with distinct traffic characteristics from the aggregated traffic in the network prefixes, thus significantly improving the understanding of the traffic patterns with detailed and meaningful interpretations.

C. Temporal Stability of Behavior Clusters

The second question on the characteristics of end-host behavior clusters we ask is the following: *Are the clusters stable over time?* In other words, *do end-hosts of network prefixes change clusters over time?* To address this question, we study the temporal stability of behavior clusters and the dynamics of cluster changes for end-hosts over time. Fig. 13(a) illustrates the high temporal stability of behavior clusters for one IP prefix during the 1-h time window. As shown by the top line in Fig. 13(a), the number of end-hosts in the prefix fluctuates slightly over time since some hosts do not continuously send or receive traffic. More importantly, the number of behavior clusters, illustrated by the bottom line in Fig. 13(a), also exhibits slight fluctuations over time. Similar observations hold for other prefixes.

In addition, we find the majority of end-hosts stay in the same behavior cluster over time. Fig. 13(b) shows the high percentage of end-hosts in the network prefix in Fig. 13(a) without changing clusters over consecutive 1-min time windows. In average, 71.8% of all the end-hosts in the traffic traces do not change clusters during the 1-h time period. Our experiments with varying timescales also show similar observations hold for other timescales as well. For example, Fig. 14(a) and (b) illustrates the high percentages of end-hosts do not change clusters over continuous 30-s and 2-min time windows, respectively. These observations confirm that network-aware behavior clustering separates end-hosts of network prefixes into distinct and stable behavior clusters.

TABLE I
TRAFFIC CLUSTERS OF AN EXAMPLE DESTINATION PREFIX

Cluster ID	Size	Flows	Patterns
1	20	422	(sip [87], spt *, dip [20], dpt 9050)
2	8	15	(sip [15], spt *, dip [8], dpt 80)
3	8	79	(sip [38], spt 80, dip [8], dpt *)
4	33	33	(sip [1], spt *, dip [33], dpt 445)

D. Practical Benefits of Exploring End-Host Behavior Clusters

1) Discovering Traffic Patterns in Network Prefixes: One major motivation of exploring behavior similarity is to gain a deep understanding of Internet traffic in backbone networks or large enterprise networks. Therefore, we first demonstrate the practical benefits of network-aware behavior clustering on discovering traffic patterns. The end-host traffic clusters discovered in each prefix reveal groups or clusters of traffic activities in the same prefixes, and understanding these patterns could be used for fine-grained traffic engineering.

End-host behavior clusters provide an improved understanding of traffic patterns in network prefixes compared to the aggregated traffic of network prefixes. For example, Table I lists four traffic clusters for one destination prefix with 69 active end-hosts during one time window. The first cluster consists of 20 destination hosts (*dip [20]*) to which 87 source hosts

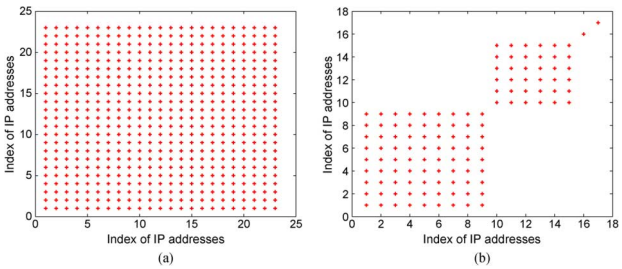


Fig. 15. Behavior clusters formed by scanning activities toward end-hosts in the same prefixes. (a) Scanning case 1. (b) Scanning case 2.

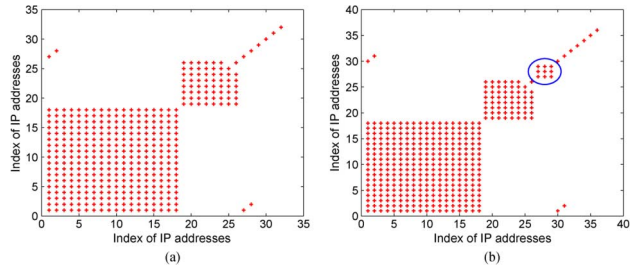


Fig. 16. Emerging behavior clusters formed by worm propagations. (a) Behavior clusters of a network prefix before Witty worm propagations. (b) Behavior clusters of the prefix during the first few minutes of Witty worm propagations.

(*sip [87]*) talk on destination port 9050 (*dpt [9050]*) with random source ports (*spt* *), while the second cluster consists of 8 hosts to which 15 source hosts talk on destination port 80. In the third cluster, 38 source hosts talk to 8 hosts using source port 80. Finally, the last cluster consists of 33 hosts to which a single source host talks on the destination port 445 that is associated with well-known vulnerabilities. In other words, the last cluster is very likely corresponding to a scanning activity toward these hosts. If the traffic of this prefix is mixed together for analysis, it becomes very difficult to interpret and understand since there are multiple behavior patterns simultaneously occurring within the same prefix. However, by separating the traffic into different clusters, the behavior of each cluster becomes much easier for network operator to understand and take necessary actions.

2) Detecting Scanning Activities With Behavior Clusters of Destination Prefixes: One interesting finding on the behavior clusters of network prefixes is that many prefixes with tens of end-hosts have only a single cluster, i.e., all hosts in each of these prefixes talk with the same set of hosts. For example, Fig. 15(a) shows one case of such activity toward one prefix with 23 end-hosts in one time window. Upon close examination, we find that in this case one particular source IP scans all 23 IP addresses, thus explaining the single traffic cluster of the network prefix.

Detecting such simple scanning scenarios is not surprising since many other existing approaches could reveal these patterns. However, the behavior clusters of destination prefixes are also able to reveal more challenging scanning cases from the massive traffic data. For instance, Fig. 15(b) shows four behavior clusters of an IP prefix. The first cluster includes nine end-hosts, while the second includes six hosts. Each of the last two clusters includes a single host since they do not share any social-behavior similarity with other hosts. By studying network traffic in each cluster, we find that the first two clusters are corresponding to two independent scanning behaviors at the same time. The first cluster is due to one scanner targeting nine different hosts, while the second cluster is caused by a different scanner targeting six other hosts. It is very interesting to note that in terms of packet counts, the last two small-sized clusters account for 99.76% of network traffic (6655 out of 6671 data packets), while the first two clusters, having only nine and seven packets respectively, accounting for a very small percentage of the traffic. If traffic analysis is simply focused on the entire prefix, such low-volume anomalous patterns could simply be missed. Therefore, this suggests that behavior analysis on host communication patterns is complementary to existing volume-based techniques for detecting scanning behavior patterns.

3) Detecting Worm Behavior in Its Early Phases: To demonstrate practical benefits of network-aware behavior clustering in detecting worm behavior, we use real traces of Witty worm collected by CAIDA [17] and combine it with the backbone network traffic into synthetic traffic. The behavior clustering is able to detect a new cluster in one of the prefixes during the very beginning of worm propagations. Fig. 16(a) and (b) shows behavior clusters of this prefix before and after worm propagations, respectively. An emerging small cluster consisting of three end-hosts marked by the circle in Fig. 16(b) is actually triggered by data packets containing the Witty worms. Such emerging behavior clusters of a network prefix triggered by worm propagation events or other suspicious activities serve as strong alarm signals to network operators for immediate response and in-depth analysis.

4) Detecting Distributed DoS (DDoS) Attacks: Detecting and mitigating DDoS attacks is one of the challenging tasks facing network operators or security analysts at edge networks due to the nature of these attacks in saturating network links. However, we argue that pushing the detection from edge networks to backbone networks is beneficial since backbone networks have sufficient bandwidth and diverse routing paths compared with edge networks. By combining backbone traffic from a large ISP and real cases of DDoS attacks identified in the previous work [18], we demonstrate the usage of behavior similarity in detecting DDoS attacks in Internet backbone networks.

Fig. 17(a)–(d) illustrates the behavior clusters of two source IP prefixes before and during DDoS attacks based on the synthetic traffic traces. The spectral clustering reveals emerging clusters or cluster changes during DDoS attacks for both source prefixes [Fig. 17(b) and Fig. 17(d)]. For the first prefix, 39 end-hosts form an emerging cluster in Fig. 17(b), while in Fig. 17(d), the existing cluster of 25 end-hosts of the second prefix in Fig. 17(c) is expanded to a much larger cluster with 52 end-hosts. The reason for the abnormal expansion of the cluster in the second prefix is that the existing 25 hosts join other 27 hosts in the same prefix in launching the DDoS attacks while sending normal data traffic as well. Compared to other methods of detecting DDoS attacks, the advantage of behavior

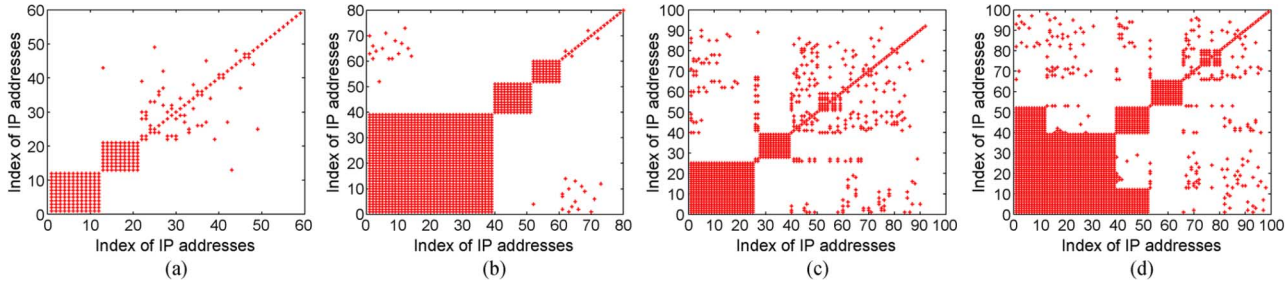


Fig. 17. Emerging behavior clusters of two independent source prefixes formed during DDoS attacks. (a) Prefix_1 before the attack. (b) Prefix_1 during the attack. (c) Prefix_2 before the attack. (d) Prefix_2 during the attack.

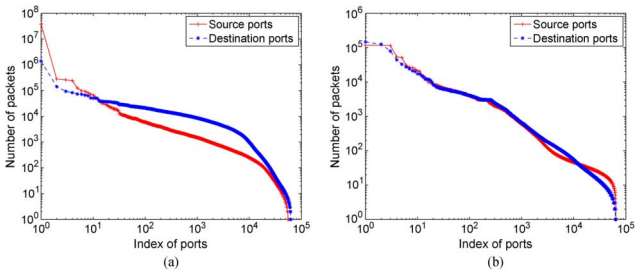


Fig. 18. Distribution of network traffic for Internet applications observed from Internet backbone links. (a) Application ports with TCP protocol. (b) Application ports with UDP protocol.

clusters is to leverage the small emerging clusters and the dynamics of existing clusters for capturing interesting events such that the attacks could be detected before the traffic arrives at edge networks and saturates network links connecting edge networks to the Internet.

VI. Exploring Similarity of Internet Applications

A. Traffic Characteristics of Internet Applications

Fig. 18(a) and (b) illustrates the distribution of IP packets for Internet application traffic observed from one backbone link during a 1-min time window for TCP and UDP ports, respectively. It is interesting to observe that a large number of application ports, regardless of transport protocols (TCP or UDP) and traffic directions (source ports or destination ports), carry non-trivial data traffic. For example, there are over 2550 TCP destination ports with more than 5000 IP packets on the link during the 1-min time window. In other words, the traditional top N approaches of focusing on a few top ports with the largest amount of traffic is not sufficient since it is also very important to study the other applications with significant volumes of IP traffic.

Building source and destination behavior graphs for each application port in our proposed method provides an opportunity to understand the social behavior of source and destination hosts engaging in the same applications. In addition, grouping these applications based on clustering coefficient of source and destination behavior graphs into distinct clusters helps understand unknown applications that share similar patterns with well-known applications.

To evaluate the quality of the clustering results, we study traffic characteristics of application clusters and compare the similarity in traffic characteristics among application ports in the same clusters as well as the dissimilarity among ports in different clusters. Our experiment results show that the application clusters indeed exhibit distinctive traffic characteristics. Specifically, for each application port p, we study IP symmetry ipsym_p, fan-out degree of source hosts fanout_p, and fan-in degree of destination hosts fanin_p. The IP symmetry ipsym_p is given by the ratio between unique source hosts and unique destination hosts engaging in the application port p. The fan-out degree for the application is the average fan-out degree of all source hosts involving in the application, while the fan-in degree is the average fan-in degree of all destination hosts.

Fig. 19(a)–(c) illustrates the distinctive characteristics in IP symmetry, fan-out degrees of source hosts, and fan-in degree of destination hosts for application clusters during one time window, respectively. The similar observations hold for other time windows as well. This observation confirms that our proposed method of behavioral graph analysis on Internet applications is indeed able to discover distinct clusters of application ports that not only exhibit similar clustering coefficient in their source and destination behavior graphs, but also share similar traffic characteristics in IP symmetry, fan-out, and fan-in degrees.

B. Usage of Application Behavior Clusters

To demonstrate the usage of application behavior clusters, we use case studies to illustrate how these clusters could aid in identifying emerging applications and detecting anomalous traffic patterns.

1) Detecting Emerging Applications: As the Internet continues to grow in end-users, mobile devices, and applications, classifying Internet applications becomes more complicated due to the rapid growth of new applications, mixed uses of application ports, and traffic hiding using well-known ports for avoiding firewall filtering. On the other hand, detecting emerging applications is very important for traffic engineering and security monitoring. In our experiment results, we find that application clusters are a feasible approach of finding new applications that share similar clustering coefficient or similar social behaviors of end-hosts with existing known applications. For example, we find that an unknown port consistently follows in the same clusters with service ports TCP port 25 (SMTP), TCP port 80 (Web), and TCP port 443 (HTTPS). We conjecture that this port very likely corresponds to a service application since the source and destination hosts engaging in this port

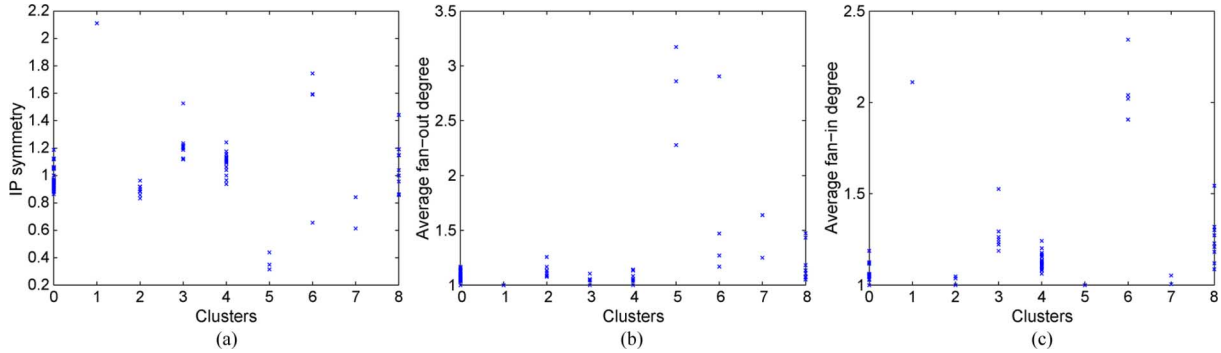


Fig. 19. Distinctive traffic characteristics of application clusters. (a) IP symmetry. (b) Fan-out degree of source IP addresses. (c) Fan-in degree of destination IP addresses.

exhibit similar social behaviors with existing known applications. Such findings could provide very valuable information for network operators for in-depth analysis.

2) Detecting Anomalous Traffic Patterns: The usage of application behavior clusters on Internet traffic also includes detecting anomalous traffic patterns. For example, the clustering results of application traffic show a cluster of TCP destination ports 135, 1433, and 22. The first two are ports associated with well-known vulnerabilities, thus it is not surprising to observe these two ports in the same cluster. However, port 22 is mostly used for SSH traffic, and it is expected to be grouped into clusters that include other major Internet service ports. An in-depth analysis reveals that during that particular time window, six source IP addresses in the same `/26` network prefix send only TCP SYN packet to 28 unique destination address on destination TCP port 22. These hosts likely scan SSH ports on Internet hosts. The scanning traffic together account for 66% of total flows toward destination TCP port 22 during the time window, which explains why TCP port 22 is clustered with ports associated with well-known vulnerabilities.

In summary, our experiment results show that application behavior clusters are able to group Internet ports into distinct clusters based on clustering coefficient and other graph properties of source and destination behavior graphs. These behavior clusters could aid network operators in understanding emerging applications and detecting anomalous traffic toward Internet applications.

VII. Related Work

Most of the prior work has focused on profiling network behavior of individual end-hosts [14], [19], [20] or classifying the roles and communities of end-hosts based on their traffic patterns [21]. In [19], the authors study the host behavior at the social, functional, and application levels for classifying traffic flows, while [14] builds behavior profiles of end-hosts using traffic communication patterns, and [20] merges packet header data into clusters based on the similarity of network traffic features. These studies focus on communication patterns of individual hosts for accurate traffic classification and behavior profiling, while the goal of this paper is to explore behavior similarity of Internet end-hosts in the same network prefixes and to discover the distinct end-host behavior clusters as well as application behavior clusters. In addition, this work focuses on groups of end-hosts in the same network prefixes, while some earlier studies [14] are interested in significant individual hosts. Many insignificant hosts might not be selected for profiling due to low traffic volume, however these hosts in the same prefixes will be collectively analyzed in this work.

Much recent research has been focused on individual Internet applications, such as Web [22], DNS [23], and HTTPS [24]. For example, [22] studies the co-locations of Web servers on the Internet and their corresponding authoritative DNS servers to discover the relationships among Web servers. In [23], Schatzmann *et al.* develop an approach of identifying HTTPS mail traffic from netflow data, while [24] analyzes DNS query traffic for in-depth understanding of the overall network traffic and detects unusual or unwanted network traffic on edge networks. However, little attempt has been made to systematically study all Internet applications. Although [25] develops an application-aware traffic measurement and analysis system, its major interest is on accurate usage-based accounting. Different from these works, this study examines social behavior of Internet applications with an ultimate goal of understanding their traffic behavior.

Graph analysis has been widely used in monitoring and visualizing network traffic [26], studying network traffic behavior [6], discovering shared-interest relationships based on e-mail communication history [27], and localizing botnets members based on the communication patterns used for command and control [28]. In [26], Iliofotou *et al.* use traffic dispersion graphs to model the social-behavior of hosts, while [6] uses traffic activity graphs to capture the interactions among hosts engaging in specific types of communications. Similar in spirit, our work is complementary to the studies in [6] and [26] by leveraging one-mode projections of bipartite communication graphs established between source and destination hosts for discovering behavior clusters within the same IP prefixes. The early work [27] constructs e-mail communication graphs and employs interest-clustering algorithms for discovering e-mail users with particular interests or expertise. Reference [28] develops an inference algorithm to search botnet communication structures from the background communication graphs constructed from the collected network traffic. Inspired by these studies, our work also uses graph analysis to uncover the social-behavior similarity among end-hosts and Internet applications.

VIII. Conclusion

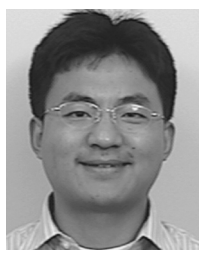
This paper uses bipartite graphs and one-mode projection graphs to analyze social behavior of end-hosts within in the same network prefixes or engaging in the same Internet applications. By applying clustering algorithms on the similarity matrices of one-mode projection graphs, we find the clustered behavior of end-hosts in the same network prefixes. Through clustering coefficient and other graph properties, we also find interesting similarity of social behavior among different Internet applications and discover distinctive application behavior clusters that group applications with similar social behavior. Our experiments demonstrate practical benefits of behavior clusters in profiling traffic patterns in IP prefixes, discovering emerging Internet applications, and detecting anomalous traffic behaviors through synthetic traffic.

Acknowledgment

The authors would like to thank CAIDA for providing the Internet traces dataset in this research.

References

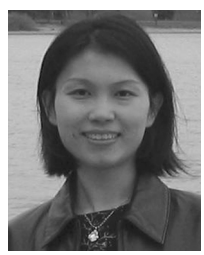
[1] K. Xu, F. Wang, and L. Gu, "Network-aware behavior clustering of Internet end hosts," in *Proce. IEEE INFOCOM*, Apr. 2011, pp. 2078–2086.
[2] K. Xu and F. Wang, "Behavioral graph analysis of internet applications," in *Proc. IEEE GLOBECOM*, Dec. 2011, pp. 1–5.
[3] S. Wei, J. Mirkovic, and E. Kissel, "Profiling and clustering internet hosts," in *Proc. Int. Conf. Data Mining*, 2006, pp. 269–275.
[4] K. Xu, Z.-L. Zhang, and S. Bhattacharyya, "Profiling internet backbone traffic: Behavior models and applications," in *Proc. ACM SIGCOMM*, Aug. 2005, pp. 169–180.
[5] H. Jiang, Z. Ge, S. Jin, and J. Wang, "Network prefix-level traffic profiling: Characterizing, modeling, and evaluation," *Comput. Netw.*, vol. 54, no. 18, pp. 3327–3340, 2010.
[6] Y. Jin, E. Sharafuddin, and Z.-L. Zhang, "Unveiling core network-wide communication patterns through application traffic activity graph decomposition," in *Proc. ACM SIGMETRICS*, Jun. 2009, pp. 49–60.
[7] J.-L. Guillaume and M. Latapy, "Bipartite graphs as models of complex networks," *Physica A, Stat. Theor. Phys.*, vol. 371, no. 2, pp. 795–813, 2006.
[8] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. NIPS*, 2001, pp. 849–856.
[9] J. Ramasco, S. Dorogovtsev, and P. Romualdo, "Self-organization of collaboration networks," *Phys. Rev.*, vol. 70, no. 3, p. 036106, 2004.
[10] M. Kaiser, "Mean clustering coefficients: The role of isolated nodes and leafs on clustering measures for small-world networks," *New J. Phys.*, vol. 10, pp. 083042–083052, Aug. 2008.
[11] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Reading, MA, USA: Addison-Wesley, 2006.
[12] Cooperative Association for Internet Data Analysis (CAIDA), La Jolla, CA, USA, "Internet traces," 2009 [Online]. Available: http://www.caida.org/data/passive/passive_2009_dataset.xml
[13] J. Fan, J. Xu, M. Ammar, and S. Moon, "Prefix-preserving IP address anonymization: Measurement-based security evaluation and a new cryptography-based scheme," *Comput. Netw.*, vol. 46, no. 2, pp. 253–272, Oct. 2004.
[14] K. Xu, Z.-L. Zhang, and S. Bhattacharyya, "Internet traffic behavior profiling for network security monitoring," *IEEE/ACM Trans. Netw.*, vol. 16, no. 6, pp. 1241–1252, Dec. 2008.
[15] University of Oregon, Eugene, OR, USA, "RouteViews project," 2005 [Online]. Available: http://www.routeviews.org/
[16] T. Cover and J. Thomas, *Elements of Information Theory*, ser. Telecommunications. New York, NY, USA: Wiley, 1991.
[17] C. Shannon and D. Moore, "The spread of the Witty worm," *IEEE Security Privacy*, vol. 2, no. 4, pp. 46–50, Jul.–Aug. 2004.
[18] A. Hussain, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks," in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 99–110.
[19] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel traffic classification in the dark," in *Proc. ACM SIGCOMM*, Aug. 2005, pp. 229–240.
[20] K. Theriault, D. Vukelich, W. Farrell, D. Kong, and J. Lowry, "Network traffic analysis using behavior-based clustering," BBN Technol., Tech. Rep., 2010.
[21] W. Aiello, C. Kalmanek, P. McDaniel, S. Sen, O. Spatscheck, and J. Merwe, "Analysis of communities of interest in data networks," in *Proc. Passive Active Meas. Workshop*, 2005, pp. 83–96.
[22] C. Shue, A. Kalafut, and M. Gupta, "The Web is smaller than it seems," in *Proc. ACM IMC*, Oct. 2007, pp. 123–128.
[23] D. Schatzmann, W. Muehlbauer, T. Spyropoulos, and X. Dimitropoulos, "Digging into HTTPS: Flow-based classification of webmail traffic," in *Proc. ACM IMC*, Nove. 2010, pp. 322–327.
[24] D. Plonka and P. Barford, "Context-aware clustering of DNS query traffic," in *Proc. ACM IMC*, Oct. 2008, pp. 217–230.
[25] T. S. Choi, C. H. Kim, S. Yoon, J. S. Park, B. J. Lee, H. H. Kim, H. S. Chung, and T. S. Jeong, "Content-aware internet application traffic measurement and analysis," in *Proc. IEEE/IFIP NOMS*, Apr. 2004, pp. 511–524.
[26] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh, and G. Varghese, "Network monitoring using traffic dispersion graphs," in *Proc. ACM SIGCOMM IMC*, Oct. 2007, pp. 315–320.
[27] M. Schwartz and D. Wood, "Discovering shared interests using graph analysis," *Commun. ACM*, vol. 36, no. 8, pp. 78–89, Aug. 1993.
[28] S. Nagaraja, P. Mittal, C.-Y. Hong, M. Caesar, and N. Borisov, "BotGrep: Finding P2P bots with structured graph analysis," in *Proc. USENIX Security Symp.*, Aug. 2010, p. 7.



Kuai Xu (M'12) received the B.S. and M.S. degrees from Peking University, Beijing, China, in 1998 and 2001, respectively, and the Ph.D. degree from the University of Minnesota, Twin Cities, Minneapolis, MN, USA, in 2006, all in computer science.

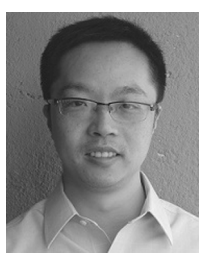
He is currently an Assistant Professor with Arizona State University, Glendale, AZ, USA. His research interests include network security and cloud computing.

Dr. Xu is a member of the Association for Computing Machinery (ACM).



Feng Wang (M'12) received the B.S. degree from Wuhan University, Wuhan, China, in 1996, the M.S. degree from Beijing University, Beijing, China, in 1999, and the Ph.D. degree from the University of Minnesota, Twin Cities, Minneapolis, MN, USA, in 2005, all in computer science.

She is currently an Associate Professor with the School of Mathematical and Natural Sciences, Arizona State University, Glendale, AZ, USA. Her research interests revolve around network modeling, optimization, and analysis of various networks, including online social networks, wireless networks, and data center networks.



Lin Gu (M'05) received the B.S. degree from Fudan University, Shanghai, China, in 1996. the M.S. degree from Peking University, Beijing, China, in 2001, and the Ph.D. degree in computer science from the University of Virginia, Charlottesville, VA, USA, in 2006, all in computer science.

He is an Assistant Professor with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology (HKUST), Hong Kong. His research interest includes cloud computing, operating systems, computer networks, and wireless sensor networks.