Computer Networks 100 (2016) 12–27

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Social relationship discovery of IP addresses in the managed IP networks by observing traffic at network boundary

Ahmad Jakalan[a,b,*], Jian Gong[a,b], Qi Su[a,b], Xiaoyan Hu[a,b], Abdeldime M.S. Abdelgder[c]

[a] School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[b] Jiangsu Key Laboratory of Computer Network Technology, Nanjing 210096, China
[c] School of Information Science and Engineering, Southeast University, Nanjing 210096, China

ARTICLE INFO

Article history:
Received 26 June 2015
Revised 12 January 2016
Accepted 14 February 2016
Available online 23 February 2016

Keywords:
Computer networks
Networks security
Clustering
IP relationship discovery
Profiling IP networks
Community detection

ABSTRACT

The continuous growth of Internet and its applications caused more difficulties for analyzing Internet communications which are becoming more and more complex, this has caused new challenges for monitoring and managing the huge and vast network traffic. It is not efficient to monitor and analyze individual IP addresses, so it is more useful to monitor groups of IP addresses that have similar behavior, which represents a certain application activity. Nowadays, such a grouping is either based on network prefixes that does not meet the requirement mentioned above as difference of traffic behavior of individual IP address not being considered, or clustering IP hosts based on their traffic patterns, which requires information about TCP/UDP port numbers (which are occasionally obfuscated) or packet payloads (which are sometimes encrypted or unavailable from aggregated flow records). This paper proposes a new methodology of clustering IP addresses within a managed network domain such as campus network or ISP clients with similar social relationship based on inter-IP connectivity structure. The key idea of this methodology is to split the entire IP address space into Internal (inside the managed domain) and External (outside) ones. The clustering strategy is to group inside IP addresses that communicate with common outside IP addresses, the similarity measure of two inside IP addresses is the unique number of the common outside IP addresses. We propose a novel approach with an approximation algorithm to discover communities on a large scale in the managed domain based on the bipartite networks and one mode projection and the basis of graph partitioning of the similarity graph. Bipartite networks were built using NetFlow datasets collected from a boundary router in an actual environment, and then a one-mode projection has been applied to build a social relationship similarity graph of the inside IP addresses. We propose a community detection algorithm to extract communities. Experimental results demonstrate that our approach can discover communities from real large scale managed domain networks with a high quality. We experimentally validate our approach in terms of IP networking by applying deep flow inspection (DFI) and deep packet inspection (DPI) on related traffic to prove that hosts with the same cluster tend to have some dominant network behavior. We demonstrated the practical benefits of exploring social behavior similarity of IP hosts

* Corresponding author at: School of Computer Science and Engineering, Southeast University, Nanjing 210096, China. Tel.: +8615366165651.
E-mail addresses: ahmad@njnet.edu.cn, ahmad.jakalan@hotmail.com (A. Jakalan), jgong@njnet.edu.cn (J. Gong), qsu@njnet.edu.cn (Q. Su), xyhu@njnet.edu.cn (X. Hu), abdeldime@hotmail.com (A.M.S. Abdelgder).

http://dx.doi.org/10.1016/j.comnet.2016.02.012
1389-1286/© 2016 Elsevier B.V. All rights reserved.

in understanding application usage, users' behavior, detecting malicious users, and users of prohibited applications.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

With the continuous growth in the number and diversity of Internet hosts and applications, it is becoming more increasingly important to understand traffic patterns of end-hosts and network applications for efficient network management and security monitoring. Different approaches have thoroughly analyzed Internet traffic and end host behavior [1–17]. Some studies have focused on analyzing traffic behavior of individual hosts. Illiofotou et al. [17] used IP communication graph and information about some applications used by few IP-hosts for the purpose of profiling Internet backbone traffic. Karagiannis et al. [4] adopted an activity graphlet to profile end-host systems based on their transport-layer behavior. Flow data of a host were compressed into a compact representation. The authors concluded that a user's behavior can undergo large changes over time. This underscores the need for clustering end hosts with similar profiles for large scale network management, to understand users' behavior for resource provisioning, load balancing and efficient network monitoring. In addition, increasingly large number of end-hosts, wide diversity of applications and massive traffic data pose significant challenges for such fine-granularity analysis for backbone networks, large enterprise networks and Internet service providers. These challenges make it difficult for researchers to study traffic patterns of end hosts independently, therefore, it is more important to find groups of hosts with similar behaviors.

Discovering communities in networks is one of the important and challenging research topics of network management and network security, as well as the research works in the social network analysis [18–28]. The problem of community detection has been addressed by researchers from different disciplines where systems are often represented as graphs, such as in sociology, biology and computer science. This problem is quite hard and not yet satisfactorily solved. Huge efforts of large interdisciplinary community of scientists have been spent on it over the past few years. The essential problem addressed by researchers when they study community detection in complex networks is the number of communities. In most of the cases, since the number of communities that the network should be partitioned into, and the number of members in each community are both unknown in advance, it is important to know which level of cutting edges of the input graph should be applied to deduce a well and an efficient graph-partitioning. The minimum cut approaches has been adopted for graph partitioning which requires to know the minimum number of edges needed to disconnect a graph [16]. However, the community structure problem differs crucially from graph partitioning in that the sizes of the communities are not usually known in advance. Community detection methods operate under the intuition that intra-community connections are more common than inter-communities connections.

Motivated by research works in community detection, the primary objective of this work is to solve the aforementioned problems by developing an approach for clustering IP hosts inside a managed domain network based on their relationship with the outside Internet. The word community usually refers to a social context. People naturally tend to form groups, within their work environment, family, and friends [29]. Similar to community detection in social networks, this paper proposes a novel community detection strategy in IP networks to discover such smaller communities in large scale IP networks that share similar behavior. The proposed work is implemented in a real large-scale IP network (China Education and Research Network CERNET) and practically proved as an effective tool for network operators to discover communities of hosts with similar connectivity with the outside network. The results presented here are an actual observations obtained from CERNET border router. Practical implementation of the method showed that it is possible to find clusters with similar hosts' behavior based only on the IP relationship. The proposed approach is for clustering IP addresses within a managed domain network based on their inter-IP communication structure with the outside network. The objective is to find groups of similar behavior to setup hosts' behavior profiles. The overall scenario of the problem is illustrated in Fig. 1. The key idea of the proposed method is to explicitly add location information (inside/outside) for IP clustering by splitting the entire IP address space into inside (the managed domain) and outside ones. The clustering method is to discover groups of inside IP addresses that communicate with common outside IP addresses. The similarity measure of two internal IP addresses is the unique number of the common outside IP addresses. The primary aim of this methodology is to find clusters with similar social behavior, which are expected to have similar network behavior. This methodology is implemented to analyze *NetFlow* dataset obtained from a border router in an actual environment real network, evaluated mainly on the basis of graph modularity, and validated using deep flow inspection and deep packet inspection.

Specifically, the contributions of this paper include:

- An intuitive methodology based on global communication structure is presented, i.e., inside–outside communication pattern represented as a bipartite graph, which is used to represent communication patterns between inside and outside networks, and construct one-mode projection graphs to deduce social relationship similarity of inside IP addresses.
- This paper adopts a new efficient clustering algorithm to discover communities of similar social behavior IP addresses.

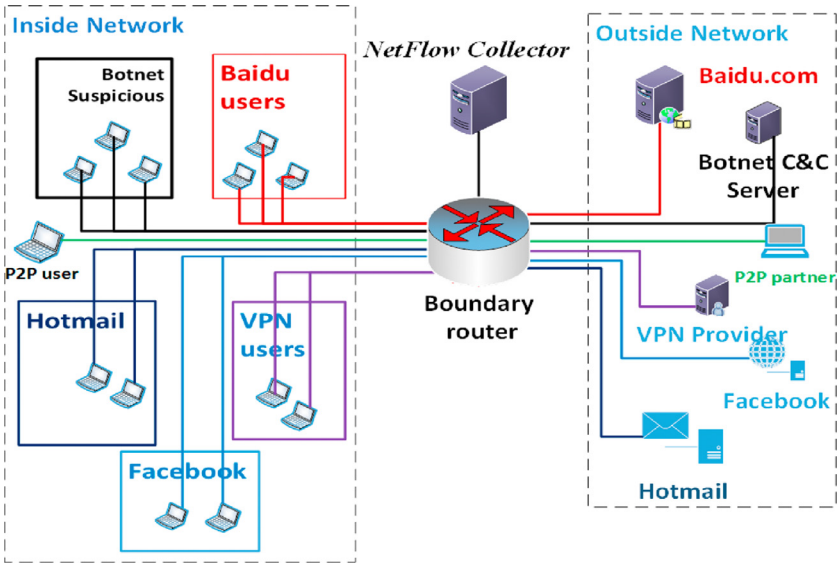


Fig. 1. The overall scenario of the problem: community structure detection within the managed network by observing their traffic at network boundary.

- This methodology is based only on IP addresses and does not require information about TCP/UDP port numbers (which are occasionally obfuscated) or packet payloads (which are often encrypted or unavailable from aggregated flow records), the use of an actual measured dataset is also the strength of this paper.
- We demonstrate practical benefits of exploring social behavior similarity of Internet hosts in understanding application usage, users' behavior, finding malicious users, and/or finding users of prohibited applications.

The rest of this paper is organized as following: In Section 2 we discuss the already existed works in the field of community detection related to our work. The details of our methodology is given in Section 3, and its experimental results are showed in Section 4. In Section 5 we evaluate the proposed algorithm and discuss the results. In Section 6 we analyze the possible semantics of these clusters found by the experiment. Finally we present our conclusion and future work in Section 7.

2. Related works

Discovering clusters of hosts with similar behavior has a great importance and usefulness for network operators. This problem has attracted significant attention of network researchers. However, the problem is far from being solved. Many researchers tried to discover clusters with similar host behaviors based on traffic patterns of end hosts [3–6,11–15,30]. Unsupervised classification of internet hosts based on their communication patterns in a space of traffic features are proposed in [3]. In this context, an unsupervised machine learning techniques were also applied in [14] to discover clusters of hosts with similar traffic behaviors based on traffic patterns of individual hosts. Unsupervised clustering algorithm was applied on fifteen direct and indirect features extracted from the flows caused by observed IP addresses to cluster the most significant IP addresses into groups of similar traffic behavior. The focus was on clustering the most significant active IP addresses (which initiate more than 90% of the overall traffic) based on their network traffic patterns. Unlike previous works, this paper concentrates on clustering inside IP hosts by creating groups of similar network connectivity with the outside ones without relying on packet and flow level information, which can be obfuscated, therefore, there is no need to collect information about the protocols, ports, or any other traffic features, instead it just relies on the connectivity between the inside and the outside IP addresses (who has a relationship with whom). In this sense, we refer to it as the IP address social relationship discovery of IP addresses in the managed domain. This connectivity is analyzed using bipartite graph and a one mode projection.

In complex networks, communities are defined as groups of densely interconnected nodes that are sparsely connected with the rest of the network [31]. Community detection has different applications. For instance, Krishnamurthy et al. [32] introduced clustering Web clients who have similar interests and are close together topologically and likely to be under common administrative control may improve the performance of services provided on the World Wide Web. Krishna et al. [33] Identify clusters of customers with similar interests in the network of purchase relationships between customers and products of online retailers which enables to set up recommendation system that guides customers and enhances the business opportunities. Ferreira et al. [34] used community detection for time series clustering by transforming set of time series into a network using different distance functions, then, applied community detection algorithms to identify groups of strongly connected vertices to identify time series clusters. Community detection in computer networks has different purposes such as detecting network traffic anomalies [35,36], behavior analysis of internet traffic [13],

and application identification [15]. The community structure discovery in networks provides an understanding of relationship between entities in this network. This is a quite helpful mechanism for protecting networks from attacks because any change in communities' structures will be easily discovered. On the other hand, managing computer networks on the level of enterprise networks or even on the higher levels (e.g., campus networks or ISPs) would become much easier when information about how entities being connected with each other is available. This will also improve the provided quality of service (QoS). A discovered community is called a community of interest which is a collection of hosts that share a common goal or environment, or it may be a collection of interacting hosts [30].

Community detection in graphs aims at identifying the modules by only using the information encoded in the graph topology. The problem has a long tradition and it has appeared in various forms in several disciplines. Newman et al. [37] proposed an algorithm, aiming at the identification of edges lying between communities and their successive removal, a procedure that after some iterations leads to the isolation of the communities. Intercommunity edges are detected based on the importance of the role of the edges in processes where signals are transmitted across the graph following paths of minimal length. Newman [31] has examined the problem of detecting community structure in networks as an optimization task to find the maximal value of the quantity called as modularity over possible divisions of a network. Modularity is one measure of the structure of networks or graphs. It was designed to measure the strength of division of a network into modules (also called groups, clusters or communities) [31]. Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes from different modules. Modularity is easy to compute and widely applicable. However, modularity optimization methods suffer from a resolution limit problem that depends on the size and connectivity of the network [38]. Spectral and min-cut techniques have been applied, but exhibit a bias such that aggressive maximization of certain community score functions can destroy intuitive notions of cluster quality [10].

Bipartite graphs have been used to analyze complex networks [39], Internet traffic [15], and social networks [40]. Kuai Xu et al. [13,15] used graph analysis to construct the bipartite graphs from host communication and then to generate the one-mode projection graphs for uncovering the communication patterns behavior similarity among the end hosts within the same network prefix by applying spectral clustering algorithm. Bipartite networks are graphs with two parties with links connecting vertices between different parties, and not possible to have links between two nodes from the same part. In [13,15] the two sides of the bipartite graph are the source IP addresses and the destination IP addresses. Unlike [13,15], this work constructs the bipartite graphs from hosts communications provided by NetFlow records at the boundary router where the two separated groups of entities are IP addresses from the two sides of the Internet, one is the managed domain inside IP addresses, and the other is the outside IP addresses, regardless of the direction of the flow record. Since the managed domain IP addresses can be mapped, and the NetFlow records communications between hosts from two different sides, any other IP address observed in the trace is considered as an outside IP address. Our focus is to detect communities from the total managed domain which may contain tens or hundreds of thousands of IP nodes, not only detecting communities from the hosts within the same network prefix as in [13], so it is important to adopt a robust algorithm which can perform the clustering in an efficient manner to cluster the expected large number of hosts. Furthermore, the work proposed in [13,15] performed partitioning similarity matrix with spectral clustering algorithm which impose high computational complexity because it firstly requires performing a quite complex Laplace transformation to compute the number of clusters k and later applies K-mean unsupervised clustering. This indicates that the complexity will greatly increase with the increase of the number of the IP addresses.

We apply one mode projection on the bipartite graph over the outside nodes, the result of the one-mode projection is the social similarity graph, the vertices in this graph are the inside IP addresses, each two nodes have an edge connecting them if both IP addresses have at least one common outside IP address, and the weight of the link is the number of common outside IP addresses. The adjacency matrix of this graph is called the similarity matrix. The clustering is done by a heuristic approximation algorithm based on an affiliation factor which measures the degree of affiliation of each node to a group of nodes. Therefore, we put each IP address in a group of IP addresses which have similar social behavior. The proposed algorithm outperforms the previous algorithms from the theoretical viewpoint and useful for the actual problem instances. There is no need to know the number of clusters in advance, instead, the clustering is based on creating clusters of explicitly neighbored nodes.

3. The proposed methodology

Community detection plays an important role in research on network behavior and characteristics of network elements and in the mining of network information. A variety of algorithms have previously been proposed, but with the continuous growth of network scale, few of them can detect community structure efficiently at a large scale networks. We study the community detection at a large scale networks based on *NetFlow* records collected from the Internet boundary routers. Actually this approach could be applied on any type of datasets that provide the trace of IP connectivity captured by border routers. This methodology is based only on IP addresses and does not require information from IP packets, like TCP/UDP port numbers (which are occasionally obfuscated) or packet payloads (which are often encrypted or unavailable from aggregated flow records), and this is the strength of this approach. Any dataset provides IP addresses connectivity between the two sides of the border router can be used. This approach is not limited to the managed domain, but it is more general. The main focus is to be able to setup a model to detect communities with similar social relationship behavior of IP addresses in one side of the Internet

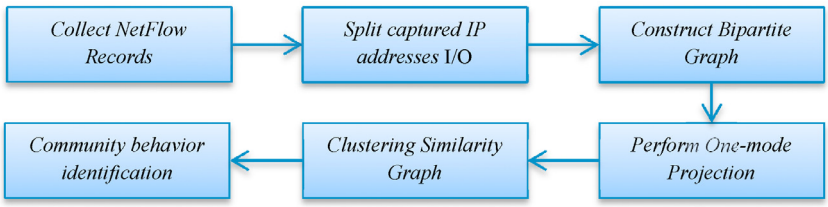


Fig. 2. Schematic process of discovering social behavior communities within the managed domain network.

based on their connectivity with the IP addresses from the other side. Each IP address is considered as an entity, and processed as an individual node. Fig. 1 shows the overall scenario of the problem, community structure detection within the managed network by observing their traffic at network boundary. As we show in Fig. 1, our intention is to group inside IP addresses that are connected to the same IP addresses from the outside network in one group, this will be useful to have a better understanding of what services are requested from or provided to the outside network, besides, it will be helpful to identify some closed user groups such as botnets. Fig. 2 shows the schematic process of our methodology. This methodology is defined in the following steps:

3.1. Split the observed IP addresses into inside and outside

This approach has been implemented in China Education and Research Network CERNET. The managed network includes about 350 network prefixes. By mapping observed IP addresses based on the network prefixes, it is possible to determine that an observed IP address belongs to the inside (managed) network, or to the outside network. *NetFlow* records are collected from the border router and aggregated into new flow records by another system called NBOS [41], each flow record represents several similar *NetFlow* records. The aggregated flow records are stored in files with time slots of 5 min. This is a moderate time, and offers some advantages. It reduces the possibility that a host is given a new IP, or the IP is being used by another host during this period. Besides, it reduces the input number of IP addresses to the clustering algorithm. The question now is which list of IP addresses will be analyzed? In the next section, the distribution of observed IP addresses over 24 h at the backbone router will be shown in Fig. 5(a). It is not useful to analyze all observed IP addresses, therefore, the focus will be on the active hosts that appear in the flow records as source IPs. From Fig. 5(a), the total number of observed inside IP addresses is much less than the total number of inside IP addresses observed in the flow records. Therefore, the focus was on the active inside IP addresses observed in flow records within a time slot of 5 min. By implementation, there is an ability to analyze the most significant IP addresses based on traffic caused by them. In a Previous research [42] it has been validated experimentally that 10% of the total IP addresses observed at the border routers, cause about 90% of the total traffic. Nevertheless, the experiments conducted here and results presented in this paper are based on the whole scope of active IP addresses observed in the flows.

3.2. Construction of the bipartite graph

Bipartite graph is a graph whose vertices can be divided into two disjoint sets (totally independent sets) such that every edge connects two vertices, each of them belongs to one of the two independent sets, and no edges can exist within one of these groups. The output of the previous process is two separate sets of IP addresses, they represent the two sets of vertices in the bipartite graph. The proposed approach is to discover communities of similar social relationship within the managed domain based on their connectivity with the IP addresses from the outside network. Let X be the list of inside IP addresses, Y is the list of outside IP addresses. The bipartite graph is represented with its adjacency matrix. Let n = |X| be the number of inside IP addresses (internal vertices), p = |Y| is the number of outside IP addresses (external vertices), and then: G = (X, Y, E) is the bipartite graph. Fig. 3(a) illustrates a bipartite graph constructed using two separate lists on nodes, the inside on the left and the outside on the right. These edges are not weighted, for an inside IP address if it has a single or multiple connections with an outside IP address, it is involved in this relationship, and this relationship is considered in clustering. For a vertex v, the number of adjacent vertices is called the degree of the vertex and is denoted as $deg(v)$. The degree sum formula for a bipartite graph states that:

$$\sum_{x \in X} \deg(x) = \sum_{y \in Y} deg(y) = |E| \tag{1}$$

The adjacency matrix of the bipartite graph is defined as the following:

$$B_{n*p} = \begin{cases} 1 & if\ there\ exists\ at\ least\ one\ flow\ between\ i\ and\ j \\ 0 & if\ there\ is\ no\ flows\ between\ the\ nodes\ i\ and\ j \end{cases}$$

3.3. One-mode projection

After the construction of the bipartite graph representing connectivity between the two sets of inside and outside IP addresses, a one mode projection over the external vertices is performed. In one mode projection of a bipartite graph; an edge connects two nodes from the same side of the bipartite graph if and only if both nodes have connections to at least one same node on the other side of the bipartite graph. Fig. 3(b) illustrates one mode projection of the internal nodes over the external nodes of the

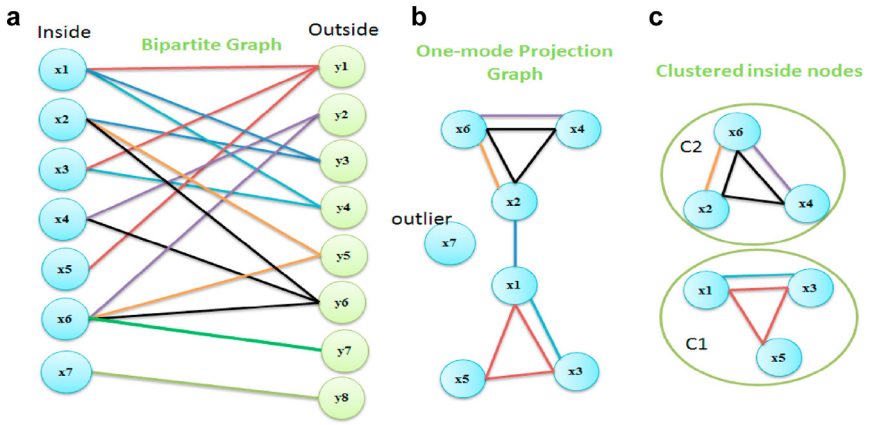
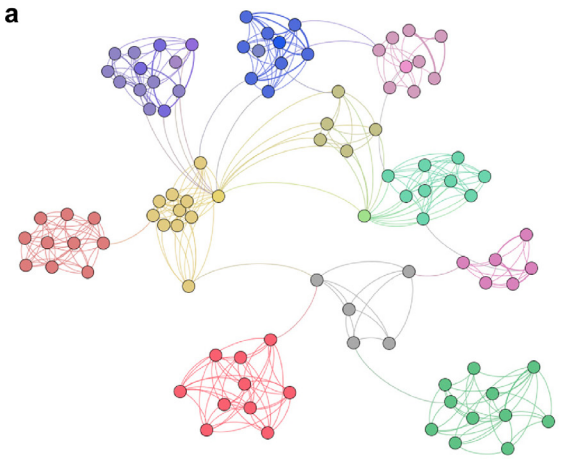


Fig. 3. (a) The bipartite network constructed from the inside and outside network connectivity (b) the one-mode projection of the inside nodes (c) the discovered communities.



b

150	3	0	1	0	7	2	2	0	0	1
3	92	0	0	0	0	0	0	2	0	0
0	0	72	0	0	0	0	0	0	0	0
1	0	0	90	0	0	0	0	0	0	0
0	0	0	0	90	0	0	0	0	0	0
7	0	0	0	0	90	0	0	1	0	0
2	0	0	0	0	0	64	6	0	0	0
2	0	0	0	0	0	6	6	0	0	0
0	2	0	0	0	1	0	0	50	0	0
0	0	0	0	0	0	0	0	0	38	0
1	0	0	0	0	0	0	0	0	0	20

Fig. 4. (a) A graph generated after a one-mode projection of 100 randomly selected inside IP addresses from the flows observed at time slot 14:00–14:05. (b) the sum of weighted links inside sub communities and between members of different communities.

bipartite graph in Fig. 3(a). We may notice that all vertices in this new generated graph are the internal set of IP addresses, and the one mode projection deduces such kind of similar social relationship between internal IP addresses. We call the new graph as the social behavior similarity graph, its vertices are the internal IP addresses, an edge appears between two nodes if they have a connectivity to at least one common external IP address, the weights of the edges represent the number of "distinct" common external IP addresses. We call the adjacency matrix of the one mode projection as the similarity matrix S which represents the similarity in social behavior between IP addresses. $S_{n*n} = [s_{ij}]$ Where s_{ij} is the number of common external IP addresses between i and j. Similarity matrix is a symmetric matrix; all entities on the main diagonal are zeros $s_{ii} = 0$.

3.4. Clustering

Communities are defined as groups of densely interconnected nodes that are only sparsely connected with the rest of the network. We have selected 100 IP addresses from the inside network observed in the flow records from the time slot 14:00–14:05, then we built the bipartite graph which represents the inside/outside connectivity between IP addresses in this list and the outside network. Then we applied a one mode projection to get the similarity graph of the selected 100 IP addresses as illustrated in Fig. 4(a) using Gephi [43] (an open source software for exploring and manipulating networks). We may notice the existence of sub groups of nodes that are connected to each other more than nodes in other groups. Fig. 4(b) shows the sum of weights links from these groups. It is clear that if some low weight links were removed, we can get several small communities, and this is what the clustering algorithm is doing.

The proposed algorithm allows an IP address to stay in the community which has the highest similarity with its members (maximum number of weighted edges), and to be removed from other communities where it has a lower number or no edges with their members in the similarity graph. From prospective of graph theory, each line in the adjacency matrix represents a sub-graph in which the vertices are the element at the row index with

Algorithm 1 Algorithm for discovering communities of similar social relationship.

Input: flow traces from border router during a given time window T_f

1: Split the observed IP addresses into X: inside and Y: outside.
2: Construct bipartite graph of IP connectivity G = (X, Y, E); an edge $e \in E$ connects a vertex $u \in X$ to a vertex $v \in Y$ if there is at least one flow between u and v in T_f.
3: Perform a one-mode projection over Y to get the Social Similarity Graph G' = (X, E'). S is the adjacency matrix of G' (the social relationship similarity matrix of X).
4: Let the initial clusters be as the following:
   a. Add each row index i in S to a new cluster C_i
   b. Add each column index j where $S[i,j]>0$ to C_i
   c. Clusters with single member are added to the outliers list and excluded from the remaining processing.
5: Remove clusters that are subsets of other clusters.
6: Check if an element x_i exists in two clusters C_k , C_l then:
   a. Calculate:
      i. the affiliation factor of x_i to C_k as: $AF(x_i\ ,\ C_k) = \sum_{j \in C_k} S_{ij}$, and
      ii. the affiliation factor of x_i to C_l as: $AF(x_i\ ,\ C_l) = \sum_{j \in C_l} S_{ij}$
   b. if $AF(x_i\ ,\ C_k) > AF(x_i\ ,\ C_l)$ thenx_i stays in C_k and removed from C_l ; Otherwise,
   c. if $AF(x_i\ ,\ C_k) < AF(x_i\ ,\ C_l)$then node x_i stay in C_l and removed from C_k;
   d. If $AF(x_i\ ,\ C_k) = AF(x_i\ ,\ C_l)$ then the element x_i will be removed from the cluster with less number of elements.
7: After removing an element x_i from a cluster C_k , if $|C_k| = 1$; then add the remaining element in C_k to C_l and remove C_k

each column index element, where the value of the cell in the matrix is larger than 0. These sub-graphs are the initial clusters, and they are identified by an ID which is the row index. In other words, we consider each node with all of its neighbors as an initial cluster. For example, the ith element in the similarity matrix and all elements in the same line where $s_{ij}>0$ (they have a common external IP with the ith) are considered as one cluster. The maximum initial number of clusters is n. It is true that the similarity matrix is a symmetric matrix, however, we experimentally realized that taking the entire matrix lines as initial clusters attain superior results, however, it costs more processing time. To eliminate the total number of initial clusters and reduce the clustering processing time, clusters which are subsets of others are removed.

As we have mentioned, the problem here is that neither the number of communities, nor the number of members in each community are known. For that reason, we remove members from certain communities based on an Affiliation Factor $AF(x_i\ ,\ C_k)$which represents the degree of affiliation of a node x_i to a cluster C_k and is calculated by:

$$AF(x_i\ ,\ C_k) = \sum_{j \in C_k} S_{ij} \tag{2}$$

AF is a metric used for the partitioning process that determines which cluster is appropriate for a specific element. Therefore, for each element x_i from cluster C_k, we check if it exists in another cluster C_l, then its affiliation for both clusters is calculated, that consequently leads to one of three cases:

- if $AF(x_i\ ,\ C_k) > AF(x_i\ ,\ C_l)$ then the element x_i will stay in C_k and removed from C_l;
- if $AF(x_i\ ,\ C_k) < AF(x_i\ ,\ C_l)$the node x_i will stay in C_l and removed fromC_k;
- Otherwise, if $AF(x_i\ ,\ C_k) = AF(x_i\ ,\ C_l)$ then the element x_i will be removed from the cluster with less number of elements.

It is worth mentioned that empty clusters are deleted after each removal operation. Algorithm 1 outlines the major steps of the proposed approach. The input of this algorithm is network flow traces during a given time window. The output of this algorithm is the inside IP addresses and each IP address is assigned to a cluster, while the members of each cluster have similar social relationship with the outside network. The strength of this algorithm is that it is intuitive and easy to implement, and the outcome is unsupervised, i.e. no need to provide any parameters such as the number of clusters or the number of nodes in each cluster or the number of nodes at which the algorithm will stop partitioning the graph. The most significance of this work also is its implementation using real datasets, as will be discussed in the following sections. Moreover, the proposed approach will be validated by means of network traffic behavior of the resulting clusters members. After clustering the IP addresses based on similarity of their connectivity with the outside network, the deep flow and packet inspections show that each resulting cluster has a dominant network behavior over the period of study.

4. Experimental results

The datasets used in our experiments are aggregated NetFlow records, each dataset is collected from the border router of CERNET over a period of 5 min. An aggregated flow record represents a group of original NetFlow records with the same 5-tuples in the same time slots. These datasets were already collected and stored in files. The proposed approach was implemented using C++ on a server with Intel CPU running CentOS release 6.4. The proposed approach is implemented to run offline, this is because it is required to provide all the dataset within the time window to be able to construct the full bipartite graph. If the time window is too short, the obtained results would be useless. The conducted experiments were applied with a time windows of 5 min. Fig. 5(a) illustrates the distribution of the number of inside and outside IP addresses observed in the flows for 24 h with 5 min time slots on dataset collected on 2015/6/03[1] by border routers. Inside IP addresses

[1] Readers who are interested can request datasets used in this implementation by email to be able to reproduce the results.

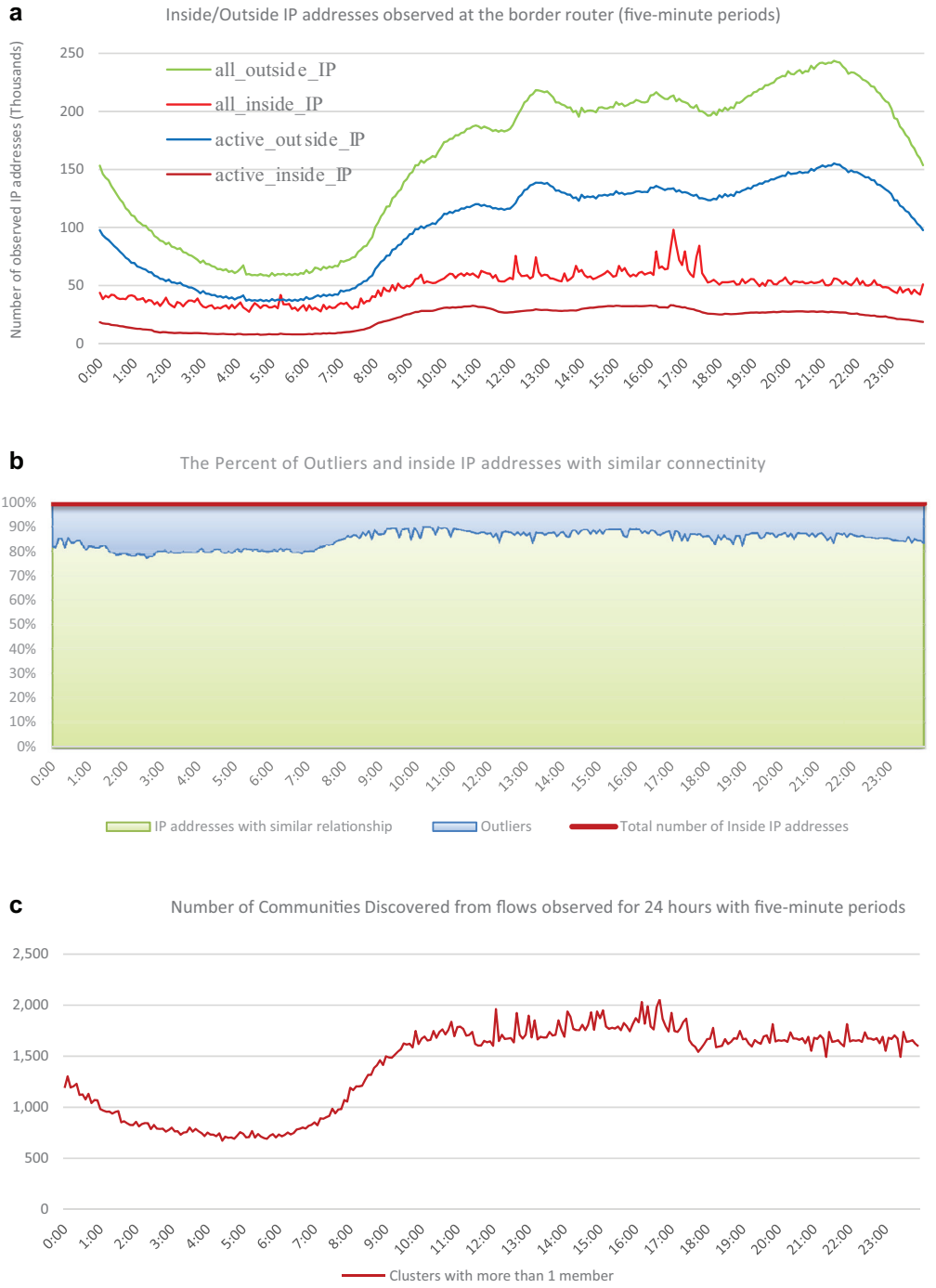


Fig. 5. The experimental results of clustering algorithm for 24 h, (a) compares the number of the observed inside and outside IP addresses with the number of active inside/outside IP addresses. (b) The percent of outliers from the total active inside IP addresses. (c) The number clusters for 24 h.

has been clustered based on the proposed approach (social relationship similarity). Fig. 5(b) shows the percent of IP addresses that do not have any common outside connectivity with other inside IP addresses. Fig. 5(c) shows the distribution of total number of clusters, including clusters with single member (outliers), and the number of clusters with at least 2 IP addresses. Experiments conducted on the above mentioned datasets for the whole day demonstrate that about 14% of the total inside IP addresses are outliers, that means they do not have social similarity with others. The remaining 86% of the inside IP addresses are clustered with at least 2 members and the number of these clusters is about 6.47% of the total number of the inside IP addresses.

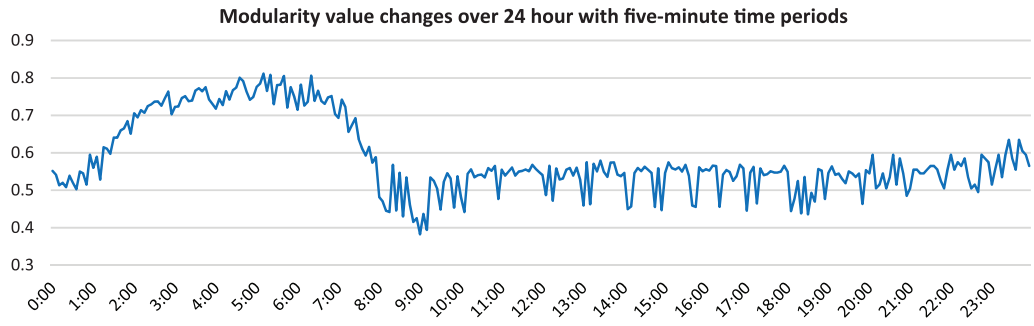


Fig. 6. Modularity value calculated after clustering internal IP addresses observed in flows over a duration of 24 h with a time slot of 5 min. We notice that most of the modularity values are higher than 0.3 which reflects the efficiency of the clustering. It's clear that the modularity value during the idle time from 1:00AM to 8:00AM is the highest.

Members of any cluster tend to have the same dominant network behavior, so it is more efficient to pick one or several IP addresses from each cluster to analyze their network behavior instead of the whole cluster members. By this work, traffic analysis has been shifted from host-level to community or cluster level which increases the granularity of traffic analysis compared to host-level traffic analysis by reducing the number of behavior profiles for analysis to an average of 20% of the clustered IP addresses (outliers 14%, and clusters 6%). And the other benefit is to know that there is a reduced list of hosts that have "outlier" behavior (connected to non-common outside hosts).

5. Discussion and evaluation

5.1. Modularity

Modularity was proposed by Newman et al. [31] and had been used as a standard to measure the strength of division of a network into modules or the quality of community detection algorithms [44]. It compares the number of edges inside a cluster with the expected number of edges that one would find in the cluster if the network were a random network with the same number of nodes and where each node keeps its degree, but edges are otherwise randomly attached. It is positive if the number of edges within groups exceeds the number expected on the basis of chance. The value of the modularity lies in the range $[-1, 1)$. Modularity is often used in optimization methods for community structure discovery in networks. It reflects the concentration of edges within modules compared with random distribution of links between all nodes regardless of modules. Networks with high modularity have dense connections between the nodes in the same cluster but sparse connections between nodes from different clusters. The main consideration of modularity is the degree of distribution of the nodes in the network. In our network G' = (X, E'), the adjacency matrix is given by the Similarity matrix S; the network contains a total of n nodes (vertices) and m edges, and d_i, d_j are the degrees of nodes i and j respectively. For any node, differences between the actual interactions and the expected numbers of connections can be obtained by calculating $S_{ij} - \frac{d_i d_j}{2m}$, therefore, for a community C, the strength of the community effect can be defined as: $\sum_{i \in C,\ j \in C} S_{ij} - \frac{d_i d_j}{2m}$ So for the network G' divided into k communities, its modularity can be calculated by the following equation:

$$Q = \frac{1}{2m} \sum_{l=1}^{k} \sum_{i \in C,\ j \in C} S_{ij} - \frac{d_i d_j}{2m} \tag{3}$$

The division by 2 m is to normalize the Q value between -1 and 1. If the number of edges inside communities is no better than random, we will get $Q = 0$. The maximum value $Q = 1$ indicates that there is a strong community structure in the network (No edges connect members from two different communities). Practical implementation of this measurement by different research works confirm that a division of a network is considered a "good" division if the Q value lies between 0.3 and 0.7 [44]. We calculated modularity after each clustering of the IP addresses observed in the flows within the time slot (5 min) for a whole day to evaluate our algorithm, Fig. 6 illustrates Q values for community structure detection by our algorithm with most of the Q values higher than 0.3. It is clear from Fig. 6 that during the idle time (from 01:00AM to 08:00AM) there exists a high level of modularity. Actually this is normal, because there is a steady behavior of internal hosts during this time. An example on such behavior can be big files downloads/uploads, backup, update, P2P. This kind of connectivity have a stable nature and there is no big changes in connectivity with outside network. On the other side, the users behavior during daytime tend to be more random, more diversity of Internet applications are used, therefore, each internal IP address tend to communicate with a massive number of external IP addresses, also a big number of internal hosts use multiple Internet applications simultaneously which make it more difficult to separate nodes into different clusters, because more number of links in the graph need to be cut off, which makes the Q value to drop as illustrated in Fig. 6. As we have mentioned earlier, the value of modularity reflects the quality of partitioning the network to discover community structure, but based on the objectives of this research, we need to find clusters that have similar connectivity to validate its traffic behavior similarity, some works has focused on achieving the highest value of modularity like in [45] where the authors focus only

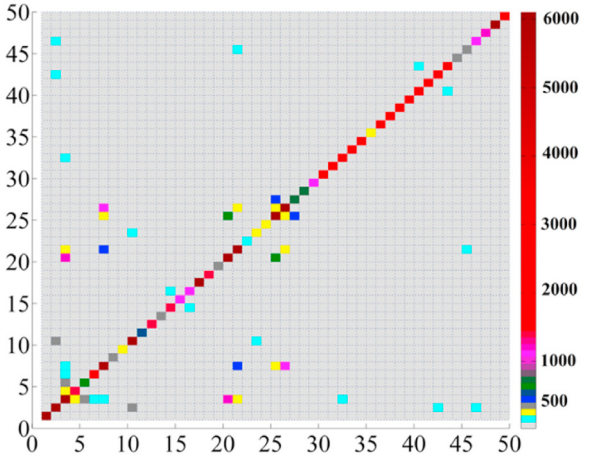


Fig. 7. A color scaled matrix of the sum of weighted edges between nodes from 50 communities, the main diagonal represents sum of weighted edges inside communities.

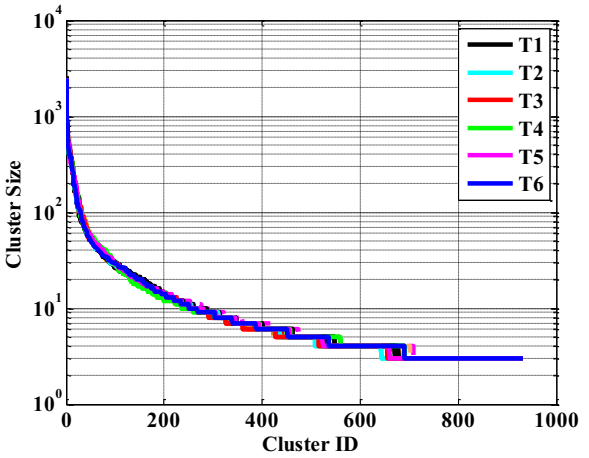


Fig. 8. A 10-base log-scaled sizes of communities discovered over one hour on 6 periods each one is 5 min sorted by size in a descending order.

on getting the highest value of modularity, therefore, it may outcomes a very big cluster which may itself should be partitioned into several similar behavior clusters when we talk about network traffic behavior. This approach creates clusters of explicitly neighbored nodes. Therefore, the graph is divided into subgraphs that does not include long paths between its nodes to be focused only on the objective which is detecting clusters of IP addresses that have a similar connectivity with the outside network.

5.2. Inter- communities vs. Intra- communities links

Communities are defined as groups of densely interconnected nodes that are only sparsely connected with the rest of the network. To evaluate the clustering algorithm, we compare the sum of weighted edges within the same community with the sum of weighted edges between nodes from different communities. Fig. 7 illustrates a color scaled matrix of the sum of weighted edges between members of a randomly selected 50 communities from the clustering results of the first 5-min slot after 2:00PM. The main diagonal represents the sum of all weighted edges connecting members from the same community, while the rest area represents sum of weighted edges connecting nodes from different communities (Inter-communities edges). It's clear from the figure that the number of edges between members of the same community are much higher than the number of edges connecting nodes from different communities. And based on the definition of communities, this attests the existence of the communities discovered by this approach. It is true that there are some clusters that have some or many edges connecting between nodes from different communities, but they are much less than the sum of edges connecting nodes from the same community. From the example illustrated in Fig. 7, the total sum of the inter-communities weighted links was calculated, it represents about (7.23%) of the total sum of weighted links in the whole graph, with a Modularity $Q = 0.759013$.

5.3. Sizes of communities

Fig. 8 illustrates the sizes of communities discovered in six periods of successive five minutes time slots sorted by size in a descending order. We noticed that the number of communities bigger than 100 is less than 5% of all communities, while the size of about 40% of the communities is less than 5 members. It is important to realize that the sizes of discovered communities over successive periods of time are almost the same. That means, there is no major changes in clusters' sizes which indicates that the clustering algorithm is greatly stable over successive periods.

5.4. Clusters stability

In cluster analysis, stability is intensely based on the dataset, especially on how well separated and how homogeneous the clusters are. In our case, the cluster instability indicate anomalous behavior in the network. The stable cluster also guarantee that the behavior of each member of the cluster can be reasonably used to represent the behavior of the whole cluster members. Therefore, the host profiling process will be more efficient for one host rather than many, which in turn significantly reduce the network management's burden and complexity. However, the number of clusters may slightly fluctuate over time, as illustrated in Fig. 5(c). The reason for that is that some hosts may do not continuously send or receive traffic, or they are not observed in *NetFlow* records. To evaluate clusters' stability over time, we define the popular IP list for a period T as the list of IP addresses which are observed in flow records in each time slot analyzed for the time period T. The stability is evaluated on a period of 20 min from 14:00 to 14:20, which consists of four time slots of five minutes length. The popular IP list of the observed inside IP addresses over T is calculated, it represents about 50% of total IP list observed in each time slot. The algorithm was run on this time period, the numbers of clusters of each run is calculated and sorted in a descending order, each cluster from the first time slot with its matching cluster from

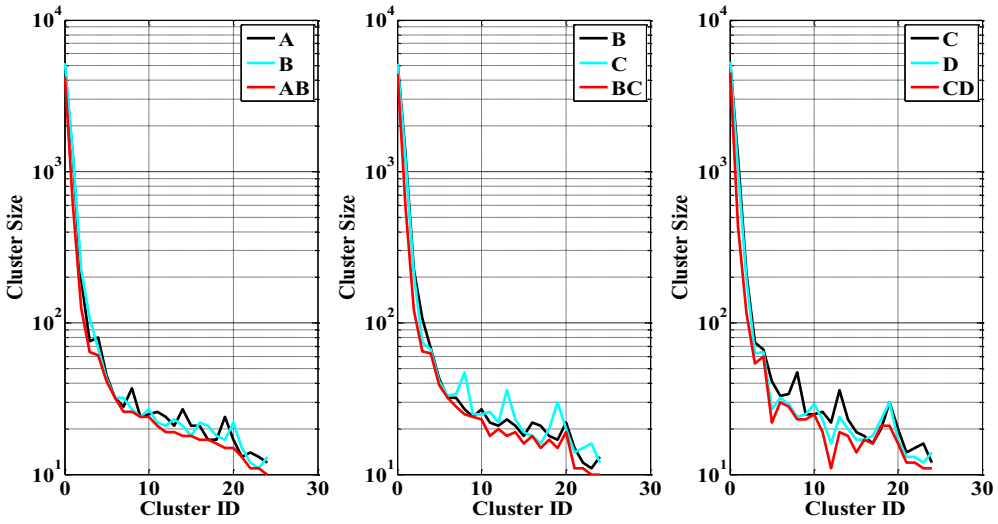


Fig. 9. Clusters stability.

the second, third, and fourth one. The 25 largest clusters were selected, and illustrated as in Fig. 9, where A, B, C and D represent the clusters' sizes in each time slot. AB represents the number of common members between A and B, and so for BC and CD. It is clear from Fig. 9 that the proposed approach demonstrated a significant stability, as the curves are almost overlapping during the whole evaluated period. It is also clear that the number of common members within two corresponding clusters from two successive periods remain the same as shown in Fig. 9(a), (b) and (c).

5.5. Time complexity of the proposed approach

Time Complexity of the proposed approach is the estimate of the amount of resources required to perform the task, from reading the input dataset, splitting the observed IP addresses, constructing the bipartite graph based on the active IP addresses, perform a one mode projection, and finally clustering the similarity graph. The time complexity is defined as the sum of the complexities of all steps. This is based on the length of the flow records, and the number of IP addresses desired to be analyzed, in addition to the connectivity overlapping between hosts. Several approaches appeared to find communities in graph, but they may not be suitable for the problem. Here we need to create clusters of explicitly neighbored nodes in similarity graph, while other optimization based approaches focus on increasing the value of modularity [26,31] which is not the good choice for this problem.

Let's denote the number of aggregated flows collected in a period of time (T_f) as N_f, then the observed IP addresses are split into n inside IP addresses and p outside IP addresses in $O(N_f)$, the bipartite graph is constructed in $O(N_f)$. The one-mode projection of bipartite graph needs $O(n^2)$ to be computed [46]. The Initial clusters are extracted in $O(n)$. Subset clusters are removed in $O(n^2)$. After removing outliers and subsets, the new number of clusters is (k). The separation of overlapping clusters requires $O(\alpha^* k^2)$. Where α represents the maximum number of common nodes between two overlapping clusters, this value has been significantly reduced in the previous step when clusters that are subset of others are removed. Consequently, empirical results show that the value of k is very low compared with of the value of n. In addition to that this value is reduced continuously during this step to reach the final number of clusters. So, theoretically, the time complexity of the proposed approach is $O(n^2)$, which means it outperforms common known algorithms for community detections which require $O(m^2n)$ in GN [37], $O(md^*log\ n)$ in Newman et al. [44] and $O(mn^2)$ in Ref. [47].

Memory consumption is also considered, sine very big matrixes are required to perform this task, vectors of vectors were appropriate for this case. The bipartite graph adjacency matrix is the biggest matrix, but using a bool type matrix considerably reduced the size and remarkably improved the computation time. vector of bool typed in C++ is a space-efficient specialization bool where each element occupies a single bit instead of sizeof(bool) bytes [48].

The proposed approach is implemented to run offline to analyze historical hosts' behavior to setup hosts' profiles. This means no need to analyze the whole flow records, instead, it is suggested to analyze the first and the seventh flow records of each twelve five-minute periods in one hour. Fig. 10 illustrates the actual memory consumption and time required to process flow records collected on 23/11/2015 by CERNET border router. The first and the seventh five-minute periods of each hour are processed on the aforementioned machine. The experiments conducted to measure the time required to analyze flow records collected by the border router over a whole day using the proposed approach illustrated in Fig. 10 shows that the processing time is correlated with the number of edges (m) in the similarity graph. It is clear from Fig. 10 that the number of edges in the similarity graph increases significantly with any increase in the number of inside IP addresses. The time and memory required to run the whole

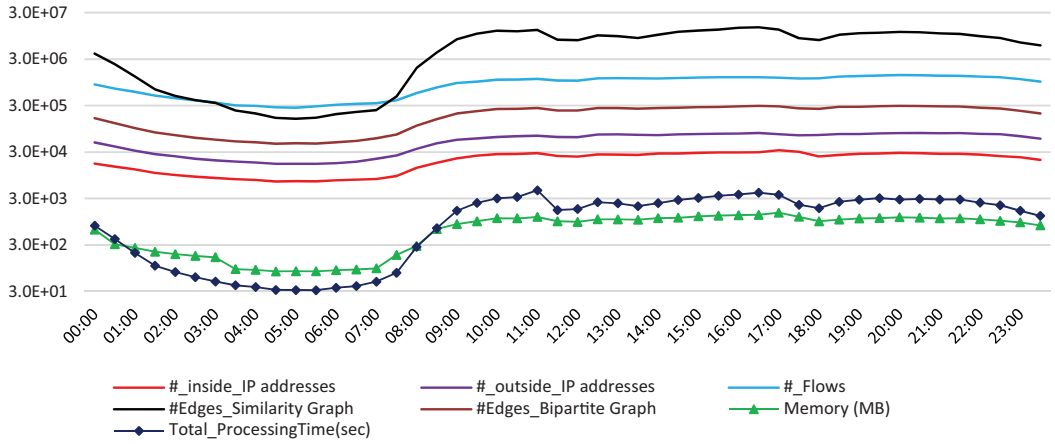


Fig. 10. Memory consumption (MB) and Time (Seconds) required to process Flow records collected by the border router during the first five minutes of each hour for a duration of one day.

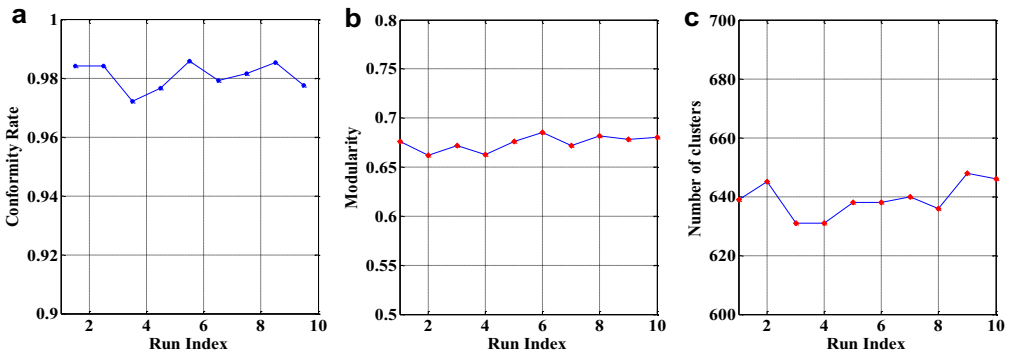


Fig. 11. Clustering error due to the elements order for the same dataset (a) the modularity changes (b) The conformity rate between the corresponding clusters (c) the number of clusters for each run.

proposed approach over different times of the day assure the efficiency of this approach within reasonable limits of time and resources consumption.

5.6. Limitations and errors

The similarity matrix measures the social similarity between two inside IP addresses. The strategy used to generate the similarity matrix imposes arranging the elements of the matrix according to the order of NetFlow records. However, we experimentally realized that the change in the order of the inside IP address consequently lead to a slight change in the clustering results. To measure the error resulting due to this problem, the algorithm is applied several times on the same dataset in a random order of the inside IP addresses. Noteworthy, the number of inside IP addresses list used to perform this test is 8163. The similarity matrix is rearranged based on the input order. For each run, we measured the modularity, the number of clusters, and the conformity factor rate. The conformity factor rate is the average percentage of the element conformity between the current state of the cluster and its previous state using the same dataset. The results are demonstrated in Fig. 11. As it is clearly shown from Fig. 11(a) the conformity rate is very high (more than 97%). And from Fig. 11(b) the change in the modularity value is very small (the difference between the maximum value and the minimum value is 0.067). Moreover, the number of clusters has not greatly changed as shown in Fig. 11(c). The limitation of this approach is that it cannot avoid this error, however, the result indicates that the error due to the arrangement is quiet small and has no great effect on the overall results.

6. Semantics of clusters in terms of IP networking

As we have mentioned above, the objective of this paper is to investigate whether a similar social behavior of IP hosts can reasonably represent the similar network behavior. In the following section, the proposed approach will be experimentally validated in terms of IP networking.

6.1. Dominant behavior of communities

By selecting some clusters resulted from a single period of clustering, and doing deep flow inspection on the flow records where the members of these clusters observed to see the whole network traffic behavior of the members

Table 1
Dominant behaviors of some selected clusters to validate results in terms of IP networking Time period 15:00–15:05.

Cluster ID	Cluster size	Protocol usage	Application usage	Notes about most frequent common external IP addresses
0	2623	TCP: 92.8% UDP: 6.69%	http: 90% P2P: 8.5%	All IP addresses in this cluster accessed baidu.com website
1	1160	TCP: 96% UDP: 3.56%	http: 93.81% P2P: 5.23%	All IP addresses in this cluster accessed lzu.edu.cn website
3	762	UDP: 9.61% TCP: 0.24%	service: 99.74% http: 0.22%	UDP, 112.124.100.*:53, aliyun.com is the source and all IP addresses in this cluster are destinations, source port is 53, destination ports are totally different
13	253	UDP: 99.9%	P2P: 88.53% Service: 11.19%	P2P connections with 69.22.142.* using different src/dst ports
18	168	TCP: 98.30% UDP: 1.55%	http: 73.47% P2P: 25.41%	All IP addresses in this cluster accessed google.com
63	51	TCP: 90.18% UDP: 7.05%	P2P: 71.77% Web: 17.79% Service: 10.12%	All IP addresses are connected with 23.81.109.* internal port: 3389 (Microsoft Terminal Server RDP). external port: 6000 (remote graphical user) direction send/receive
69	47	UDP: 58% TCP: 42%	P2P: 50.32% Service: 46.75%	IP addresses from chinatelecom.com.cn had P2P connections with the members of this cluster, also they requested MSSQL on the internal port: 1433, external port: random
107	27	TCP: 100%	Web: 96.77% P2P: 3.22%	Port This 80, 97%, External IP is *. This is an abnormal behavior: a single external IP address with random port numbers communicating over TCP (Web service) with 27 internal IP address as if they are all web service providing service on port 80.

Table 2
Some identified closed groups.

ID	Size	Outside peers	Protocol usage	Application usage	Local port	Remote port	Notes about most frequent common external IP addresses
10	32	13	TCP 98%	Unknown: 95%	3389: 100%	*	All IP addresses are connected with one or both of two outside IP addresses (46% the first, 30% the second)
104	4	29–31	TCP 100%	FTP: 68%, Unknown, 32%,	80, 100%	*	Strange behavior: About 30 outside IP addresses using ftp and another unknown service from inside 4 hosts on port 80 only.
177	3	3–8	TCP 100%	Unknown: 45% FTP: 15% http: 10% Multimedia: 10% Voice: 10%	80: 60% 8080: 25% 8081: 15%	*	A single outside IP address is using several services from 3 inside hosts. The requested services include only 10% http, and the rest 90% for other services, but the port used by inside hosts are only 80, 8080, 8081 which are normally used for http.
184	2	80–85	TCP 99%	FTP: 75% Unknown, 19% http: 2%	80: 77%	80, 20%	The two inside hosts request ftp service on port 80
323	2	11–21	TCP 100%	FTP: 65% Unknown: 35%	80: 100%	*	Outside hosts requesting ftp and other services provided from inside hosts on port 80.
331	2	1	TCP 100%	Unknown: 100%	3389: 100%	*	A single outside IP connected with two inside IPs

Table 3
Network traffic behavior of some identified outliers IP addresses.

IP	Outside peers	Protocol Usage	Application Usage	Local port	Remote port	Notes about most frequent common external IP addresses
IP1	1	UDP:100%	Service: 100%	7643: 100%	1196: 100%	From 00:00 to 24:00 and in each time slot it connects with a single outside IP address X, average packets 3packets/5 minutes, average packet size is 200.
IP2	159	UDP: 98.7%	Unknown: 98.7%	53: 98.7%	*	It seems to be a DNS server

of each cluster, it was very clear that there is a dominant behavior of IP addresses in the same cluster, such as the most frequently used application, furthermost accessed website and the highest requested services. Table 1 shows that some clusters are very huge, and the service they are using is very common as shown in cluster (0) where the most common service accessed by the cluster members is searching the web using baidu.com, or using other services provided by the giant website in China.

6.2. Closed groups

Closed user groups are groups of hosts that have similar behaviors. These groups use some specific applications for a relatively long time, they stay connected with one or a list of outside IP addresses all the time, and they have the same purpose. Some examples of these closed groups are botnets. This approach allows to identify such closed user groups. The idea behind this approach is to find IP connectivity between IP addresses. After clustering the inside IP addresses over a long time, it is possible to check if some clusters stay available and what kind of changes happened on the members. The experiments conducted here demonstrate that there are some clusters remain active with a relatively small change in their members. The experiments conducted here are to cluster the inside IP addresses, hence, the closed groups are inside groups of hosts having the same traffic behavior, and they are connected to the same outside IP address for a long time. If the clustering were done on the outside IP addresses, the results would be able to find outside closed groups with an intention in some specific inside hosts. Within one hour from 14:00 to 17:00, it was possible to identify 7 closed groups. Closed groups are identified when the same outside IP address/s remain connected with the same list of the inside IP addresses for a long time. Table 2 shows DFI summary applied on the corresponding clusters' members for the observation period. An abnormal behavior was noticed in the behavior of members of cluster id 177, where a single outside IP address requesting several services on common ports (like 80, 8080, and 8081). We compared the clustering results of successive periods for a long time, and we found that the traffic behavior has not changed.

6.3. Outliers

Outliers are IP addresses that do not have any similar connectivity with other inside IP addresses. Experiments are conducted to identify some of these outliers. Table 3 shows the network behavior of some outlier inside IP addresses. We found that most of these IP addresses use some network services over UDP protocol. They send/receive small size packets (average packet size is less than 80bytes/packet). Some of these IP addresses remain active for a relatively long time. From the described dataset, the most active list of outlier IP address was extracted from the period 14:00-15:00. A list of 39 inside IP addresses were found active in each 5-minutes time slot over the study period.

7. Conclusion

This paper presented an approach to discover IP relationship to setup a clustering model based on IP connectivity of IP addresses inside the managed domain network based on their connectivity with the outside network by observing traffic at the border router. The objective is to setup hosts' profiles, however, since it is not efficient to setup such profiles for each IP address, it is more efficient to discover clusters of IP addresses with similar behaviors. Instead of clustering hosts based on their traffic patterns, this paper proposed a clustering strategy based only on the IP connectivity without any information about protocol, port, packets. Our experimental results demonstrated that this approach can discover communities from real managed domain networks. The approach is discussed and evaluated using concepts from graph partitioning, such as modularity and community detection definitions, it has been also validated by deep flow inspection DFI. To the best of our knowledge, this is the first step forward in the research to discover social communities of IP networks by splitting network into inside and outside networks and discover communities' structure among inside network based on similarity of connectivity with the outside networks. The proposed approach is implemented in a real network, and the quality of clustering significantly fulfilled our expectations. This work has practical benefits in network security, network management, and the monitoring and analysis of large networks. The future work will include improving the algorithm for further reduction in the calculations complexity of the proposed approach.

Acknowledgment

This work was conducted under the support of Jiangsu Key Laboratory of Computer Networking Technology and the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of

Education. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of those sponsors.

References

[1] Y. Himura, K. Fukuda, K. Cho, P. Borgnat, P. Abry, H. Esaki, Synoptic graphlet: bridging the gap between supervised and unsupervised profiling of host-level network traffic, IEEE-ACM Trans. Netw. 21 (2013) 1284–1297.

[2] L. Bin, L. Chuang, Q. Jian, H. Jianping, P. Ungsunan, A NetFlow based flow analysis and monitoring system in enterprise networks, Comput. Netw. 52 (2008) 1074–1092.

[3] G. Dewaele, Y. Himura, P. Borgnat, K. Fukuda, P. Abry, O. Michel, et al., Unsupervised host behavior classification from connection patterns, Int. J. Netw. Manag. 20 (2010) 317–337.

[4] T. Karagiannis, K. Papagiannaki, N. Taft, M. Faloutsos, Profiling the end host, Passive and Active Network Measurement, Springer, 2007, pp. 186–196.

[5] S. Wei, J. Mirkovic, E. Kissel, Profiling and clustering Internet hosts, in: Proceedings of the International Conference on Data Mining, 2006.

[6] X. Kuai, Z. Zhi-Li, S. Bhattacharyya, Internet traffic behavior profiling for network security monitoring, IEEE/ACM Trans. Netw. 16 (2008) 1241–1252.

[7] B. Li, M.H. Gunes, G. Bebis, J. Springer, A supervised machine learning approach to classify host roles on line using sFlow, in: Proceedings of the first edition Workshop on High Performance and Programmable Networking, New York, New York, USA, 2013.

[8] H. Qiao, J. Peng, C. Feng, J.W. Rozenblit, Behavior analysis-based learning framework for host level intrusion detection, in: Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, 2007.

[9] K. Xu, Z.-L. Zhang, S. Bhattacharyya, Profiling internet backbone traffic: behavior models and applications, in: 1515 Broadway, 17th Floor, New York, NY 10036-5701, United States, 2005, pp. 169–180.

[10] Z. Zhang, B.-Q. Wang, H.-C. Chen, H.-L. Ma, Internet traffic classification based on host connection graph, Dianzi Yu Xinxi Xuebao(J. Electron. Inf. Technol.) 35 (2013) 958–964.

[11] T. Karagiannis, K. Papagiannaki, M. Faloutsos, BLINC: multilevel traffic classification in the dark, ACM SIGCOMM Comput. Commun. Rev. (2005) 229–240.

[12] S. Bhattacharyya, K. Xu, and Z.-L. Zhang, "Identifying significant behaviors within network traffic," ed: US Patent 8,204,974, 2012.

[13] K. Xu, F. Wang, L. Gu, Behavior Analysis of Internet traffic via bipartite graphs and one-mode projections, IEEE-ACM Trans. Netw. 22 (2014) 931–942.

[14] A. Jakalan, G. Jian, W. Zhang, S. Qi, Clustering and profiling ip hosts based on traffic behavior, J. Netw. 10 (2015) 99–107 2015-03-03.

[15] K. Xu, F. Wang, L. Gu, Network-aware behavior clustering of Internet end hosts, in: INFOCOM, 2011 Proceedings IEEE, 2011, pp. 2078–2086.

[16] G.W. Flake, S. Lawrence, C.L. Giles, Efficient identification of web communities, in: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, 2000, pp. 150–160.

[17] M. Iliofotou, B. Gallagher, T. Eliassi-Rad, G. Xie, M. Faloutsos, Profiling-by-association: a resilient traffic profiling solution for the internet backbone, in: Proceedings of the 6th International Conference, Philadelphia, Pennsylvania, 2010.

[18] J. Xiang, Y.-N. Tang, Y.-Y. Gao, Y. Zhang, K. Deng, X.-K. Xu, et al., Multi-resolution community detection based on generalized self-loop rescaling strategy, Phys. A: Stat. Mech. Appl. 432 (2015) 127–139.

[19] R. Shang, S. Luo, Y. Li, L. Jiao, R. Stolkin, Large-scale community detection based on node membership grade and sub-communities integration, Phys. A: Stat. Mech. Appl. 428 (2015) 279–294.

[20] J. He, D. Chen, A fast algorithm for community detection in temporal network, Phys. A: Stat. Mech. Appl. 429 (2015) 87–94.

[21] J. Eustace, X. Wang, Y. Cui, Overlapping community detection using neighborhood ratio matrix, Phys. A: Stat. Mech. Appl. 421 (2015) 510–521.

[22] A. Clementi, M. Di Ianni, G. Gambosi, E. Natale, R. Silvestri, Distributed community detection in dynamic graphs, Theor. Comput. Sci. 584 (2015) 19–41.

[23] W. Liu, M. Pellegrini, X. Wang, Detecting communities based on network topology, Sci. Rep. 4 (2014) 5739.

[24] C. Chang, C. Tang, Community detection for networks with unipartite and bipartite structure, New J. Phys. 16 (2014) 093001.

[25] X. Yongcheng, C. Ling, Z. Shengrong, Community detection from bipartite networks, in: Web Information System and Application Conference (WISA), 2013 10th, 2013, pp. 249–254.

[26] O.R. Zaïane, R. Goebel, and J. Chen, "Detecting communities in social networks using max–min modularity," in: Proceedings of the 2009 SIAM International Conference on Data Mining, pp. 978–989.

[27] Z. Xia, Z. Bu, Community detection based on a semantic network, Knowl. Based Syst. 26 (2012) 30–39.

[28] M.E.J. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (2004) 066133.

[29] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (2010) 75–174.

[30] W. Aiello, C. Kalmanek, P. McDaniel, S. Sen, O. Spatscheck, J. Van der Merwe, Analysis of communities of interest in data networks, in: Proceedings of the 6th international conference on Passive and Active Network Measurement, Boston, MA, 2005.

[31] M.E. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. U S A 103 (2006) 8577–8582.

[32] B. Krishnamurthy, J. Wang, On network-aware clustering of Web clients, SIGCOMM Comput. Commun. Rev. 30 (2000) 97–110.

[33] P. Krishna Reddy, M. Kitsuregawa, P. Sreekanth, S. Srinivasa Rao, A graph based approach to extract a neighborhood customer community for collaborative filtering, in: S. Bhalla (Ed.), Databases in Networked Information Systems, vol. 2544, Springer, Berlin Heidelberg, 2002, pp. 188–200.

[34] L.N. Ferreira, L. Zhao, Time series clustering via community detection in networks, Inf. Sci. 326 (2016) 227–242.

[35] W.X. Liu, J. Cai, A new method of detecting network traffic anomalies, Appl. Mech. Mater. 347 (2013) 912–916.

[36] Z.Z. Chen, W. Hendrix, N.F. Samatova, Community-based anomaly detection in evolutionary networks, J. Intell. Inf. Syst. 39 (2012) 59–85.

[37] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. 99 (2002) 7821–7826 June 11, 2002.

[38] S. Fortunato, M. Barthelemy, Resolution limit in community detection, Proc. Natl. Acad. Sci. U S A 104 (2007) 36–41.

[39] J.L. Liu, J. Cai, Complex network community structure of user behaviors and its statistical characteristics, in: Proceedings of the 2011 Third International Conference on Multimedia Information Networking and Security, 2011.

[40] E.A. Horvat, K.A. Zweig, One-mode projection of multiplex bipartite graphs, in: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2012, 2012, pp. 599–606.

[41] Z. WeiWei, G. Jian, D. Wei, Z. XiaoGuo, NBOS: a fine-grained network management system, in: presented at the CERNET2012, 2012.

[42] A. Jakalan, J. Gong, S. Liu, Profiling IP hosts based on traffic behavior, in: Proceedings of the IEEE International Conference on Communication Software and Networks (ICCSN), 2015, 2015, pp. 105–111.

[43] M. Bastian, S. Heymann, M. Jacomy, Gephi: An Open Source Software for Exploring and Manipulating Networks, 2009.

[44] M.E. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E Stat. Nonlin. Soft Matter Phys. 69 (2004) 026113.

[45] D.B. Vincent, G. Jean-Loup, L. Renaud, L. Etienne, Fast unfolding of communities in large networks, J. Stat. Mech.: Theory Exp. 2008 (2008) P10008.

[46] K. Zweig, M. Kaufmann, A systematic approach to the one-mode projection of bipartite graphs, Soc. Netw. Anal. Min. 1 (2011) 187–218.

[47] P. Pons, M. Latapy, Computing communities in large networks using random walks, Computer and Information Sciences-ISCIS 2005, Springer, 2005, pp. 284–293.

[48] cppreference.com. Available: 2016 http://en.cppreference.com/w/

Ahmad Jakalan is a Ph.D. Candidate in School of Computer Science and Engineering, Southeast University, Nanjing, P. R. China. His research interests include computer networks and security, intrusion detection, network traffic and host profiling. In 2011, he received his MSc in Computer Science and Technology from Southeast University, China. In 2005, He received his BSc in Informatics Engineering from Aleppo University, Aleppo, Syria.

Jian Gong is a professor in the School of Computer Science and Engineering, Southeast University. His research interests are network architecture, network intrusion detection, and network management. He has received his BS in computer software from Nanjing University, and his PhD in computer science and technology from Southeast University.

Qi Su is a Ph.D. candidate in School of Computer Science and Engineering, Southeast University, Nanjing, P. R. China. His research interests are network measurement and network management. He received B.S degree in computer science and technology from the Southeast University, Nanjing, P. R. China.

Xiaoyan Hu is an assistant professor in School of Computer Science and Engineering, Southeast University. She focuses her research interests on information centric networking, in-network caching and scalable name-based routing. She received her BS in software engineering from Nanjing University of Science and Technology in 2007, MS and PhD in computer architecture from Southeast University in 2009 and 2015 respectively. She visited netsec lab in Colorado State University, a research group working on NDN, from Sep. 2010 to Aug. 2012.

Abdeldime Mohamed Salih Abdelgader, got his BSc from Sudan university of Science and technology in 2000, MSc degree from Karary University in 2003. Since that time he is a lecturer at Karary University in the department of electrical and computer engineering, as well as consultant for many IT companies in Sudan. Currently, he is a PHD candidate in Southeast University, Nanjing, China in the School of information science and communication engineering. His research interest is related to computer networks, mobile communications, and the physical layer of the vehicular networks.