Analysis of Communities of Interest in Data Networks

William Aiello[1], Charles Kalmanek[2], Patrick McDaniel[3],
Subhabrata Sen[2], Oliver Spatscheck[2], and Jacobus Van der Merwe[2⋆]

[1] Department of Computer Science, University of British Columbia,
Vancouver, B.C. V6T 1Z4, Canada
aiello@cs.ubc.ca
[2] AT&T Labs – Research,
Florham Park, NJ 07932, U.S.A.,
{crk,sen,spatsch,kobus}@research.att.com
[3] Department of Computer Science and Engineering, Penn State University,
University Park, PA 16802, U.S.A.
mcdaniel@cse.psu.edu

Abstract. *Communities of interest* (COI) have been applied in a variety of environments ranging from characterizing the online buying behavior of individuals to detecting fraud in telephone networks. The common thread among these applications is that the historical COI of an individual can be used to predict future behavior as well as the behavior of other members of the COI. It would clearly be beneficial if COIs can be used in the same manner to characterize and predict the behavior of hosts within a data network. In this paper, we introduce a methodology for evaluating various aspects of COIs of hosts within an IP network. In the context of this study, we broadly define a COI as a collection of interacting hosts. We apply our methodology using data collected from a large enterprise network over a eleven week period. First, we study the distributions and stability of the size of COIs. Second, we evaluate multiple heuristics to determine a stable core set of COIs and determine the stability of these sets over time. Third, we evaluate how much of the communication is not captured by these core COI sets.

1 Introduction

Data networks are growing in size and complexity. A myriad of new services, mobility, and wireless communication make managing, securing, or even understanding these networks significantly more difficult. Network management platforms and monitoring infrastructures often provide little relief in untangling the *Gordian knot* that many environments represent.

In this paper, we aim to understand how hosts communicate in data networks by studying host level *communities of interest* (COIs). A community of interest is a collection of entities that share a common goal or environment. In the context of this study, we broadly define a community of interest as a collection of interacting hosts. Using data collected from a large enterprise network, we construct community graphs representing the existence and density of host communications. Our hypothesis is that the

⋆ This research was conducted when the authors were with AT&T Labs – Research.

behavior of a collection of hosts has a great deal of regularity and structure. Once such structure is illuminated, it can be used to form parsimonious models that can become the basis of management policy. This study seeks to understand the structure and nature of communities of interest ultimately to determine if communities of interest are a good approximation of these models. If true, communities of interest will be useful for many purposes, including:

- *network management* - because of similar goals and behavior, communities will serve as natural aggregates for management
- *resource allocation* - allocating resources (e.g., printers, disk arrays, etc.) by community will increase availability and ensure inter-community fairness
- *traffic engineering* - profiles of communal behavior will aid capacity planning and inform prioritization of network resource use
- *security* - because communities behave in a consistent manner, departure from the norm may indicate malicious activity

Interactions between social communities and the Web have been widely studied [1, 2]. These works have shown that the web exhibits the *small world phenomena* [3, 4], i.e., any two points in the web are only separated by a few links. These results indicate that digital domains are often rationally structured and may be a reflection of the physical world. We hypothesize that host communication reflects similar structure and rationality, and hence can be used to inform host management. In their work in network management, Tan et. al. assumed that hosts with similar connection habits play similar roles within the network [5]. They focused on behavior within local networks by estimating host *roles*, and describe algorithms that segment a network into host role groups. The authors suggest that such groups are natural targets of aggregated management. However, these algorithms are targeted to partitioning hosts based on some *a priori* characteristic. This differs from the present work in that we seek to identify those characteristics that are relevant. Communities of interest can also expose aberrant behavior. Cortes et. al. illustrated this ability in a study of fraud in the telecommunications industry [6]. They found that people who re-subscribed under a different identity after defaulting on an account could be identified by looking at the similarity of the new account's community.

This paper extends these and many other works in social and digital communities of interest by considering their application to data networks. We begin this investigation in the following section by outlining our methodology. We develop the meaning of communities of interest in data networks and then explain how our data was collected and pre-processed. While the data set that we analyze is limited to traffic from an enterprise network, we believe that the methodology is more broadly applicable to data networks in general. In Section 3 we present the results of our analysis and conclude the paper in Section 4 with a summary and indication of future work.

2 Methodology

In this section we consider the methodology we applied to the COI study. First we develop an understanding of what COI means in the context of a data network. Then we

explain how we collected the data from an enterprise network and what pre-processing we had to perform on the data before starting our analysis.

2.1 Communities of Interest

We have informally defined COI for a data network as a collection of interacting hosts. In the broadest sense this would imply that the COI of a particular host consists of *all* hosts that it interacts with. We call the host for which we are trying to find a COI the **target-host**. We begin our analysis by exploring this broad COI definition, by looking at the total *number* of hosts that target-hosts from our data set interact with. Thus in this first step we only look at the COI set size and its stability over time.

Considering all other hosts that a target-host ever communicates with to be part of its COI might be too inclusive. For example, this would include one-time-only exchanges which should arguably not be considered part of a host's COI. Intuitively we want to consider as part of the COI the set of hosts that a target-host interact with *on a regular basis*. We call this narrower COI definition the *core* COI.

In this work it is not our goal to come up with a single core COI definition. Instead, it is our expectation that depending on the intended *application* of COI, different definitions might be relevant. For example, in a resource allocation application the relevant COI might be centered around specific protocols or applications to ensure that the COI for those applications receive adequate resources. On the other hand an intrusion detection application might be concerned about deviations from some "normal" COI. However, in order to evaluate our methodology, we do suggest and apply to our data two example definitions of a core COI:

- **Popularity**: We determine the COI for a *group* of target-hosts by considering a host to be part of the COI if the percentage of target-hosts interacting with it exceeds a threshold T, over some time period of interest Y.
- **Frequency**: A host is considered to be part of the COI of a target-host, if the target-host interacts with it at least once *every* small time-period Z (the bin-size) within some larger time period of interest Y.

Intuitively these two definitions attempt to capture two different constituents of a core COI. The most obvious is the *Frequency* COI which captures any interaction that happens frequently, for example access to a Web site containing news that gets updated frequently. The *Popularity* COI attempts to capture interactions that might happen either frequently or infrequently but is performed by a large part of the user population. An example would be access to a time-reporting server or a Web site providing travel related services.

From the COI definitions it is clear that the *Popularity* COI becomes more inclusive in terms of allowing hosts into the COI as the threshold (T) decreases. Similarly the *Frequency* COI becomes more inclusive as the bin-size increase. For the *Popularity* case where the threshold is zero, *all* hosts active in the period-of-interest are considered to be part of the COI. Similarly, for the *Frequency* case where the bin-size is equal to the period-of interest, all hosts in that period are included in the COI. When the period-of-interest, Y, is the same for the two core COI definitions, these two special cases (i.e.,

$T = 0$ for the *Popularity* COI and $Z = Y$ for the *Frequency* COI), therefore produce the same COI set.

Notice that the *Popularity* COI defines a core COI set for a "group" of hosts, whereas the *Frequency* COI defines a per-host COI. We have made our core COI definitions in the most general way by applying it to "hosts", i.e., not considering whether the host was the initiator (or client) or responder (or server) in the interaction[4]. While these general definitions hold, in practice it might be useful to take directionality into account. For example, the major servers in a network can be identified by applying the *Popularity* definition to the percentage of clients initiating connections to servers. Similarly, the *Frequency* definition can be limited to clients connecting to servers at least once in every bin-size interval to establish a per-client COI.

In the second step of our analysis we drill deeper into the per-host interactions of hosts in our data set to determine the different core COI sets. Specifically, we determine the *Popular* COI and the *Frequency* COI from a client perspective and consider their stability over time.

Ultimately we hope to be able to predict future behavior of hosts based on their COIs. We perform an initial evaluation of how well core COIs capture the future behavior of hosts. Specifically, we combine all the per-host *Client-Frequency* COIs with the shared *Popularity* COI to create an **Overall** COI. We construct this COI using data from a part of our measurement period and then evaluate how well it captures host behavior for the remainder of our data by determining how many host interactions are *not* captured by the *Overall* COI.

2.2 Data Collection and Pre-processing

To perform the analysis presented in this paper we collected eleven weeks worth of flow records from a single site in a large enterprise environment consisting of more than 400 distributed sites connected by a private IP backbone and serving a total user population in excess of 50000 users. The flow records were collected from a number of LAN switches using the Gigascope network monitor [7]. The LAN switches and Gigascope were configured to monitor *all* traffic for more than 300 hosts which included desktop machines, notebooks and lab servers. This set of monitored hosts for which we captured traffic in both directions are referred to as the **local hosts** and form the focal point of our analysis. In addition to some communication amongst themselves, the local hosts mostly communicated with other hosts in the enterprise network (referred to as **internal hosts**) as well as with hosts outside the enterprise environment (i.e., **external hosts**). We exclude communication with external hosts from our analysis as our initial focus is on intra-enterprise traffic. During the eleven week period we collected flow records corresponding to more than 4.5 TByte of network traffic. In our traces we only found TCP, UDP and ICMP traffic except for some small amount of RSVP traffic between two test machines which we ignored. For this initial analysis we also removed weekend data from our data set, thus ensuring a more consistent per-day traffic mix. Similarly, we also excluded from the analysis any hosts that were not active at least once a week during the measurement period.

[4] We provide an exact definition of client and server in the next section.

Our measurement infrastructure generated unidirectional flow-records for monitored traffic in 5 minute intervals or bins. A flow is defined using the normal 5-tuple of IP protocol type, source/destination addresses and source/destination port numbers. We record the number of bytes and number of packets for each flow. In addition, each flow record contains the start time of the 5 minute bin and timestamps for the first packet and last packet of the flow within the bin interval. The collected "raw" flow-records need to be processed in a number of ways before being used for our analysis:

Dealing with DHCP: First, because of the use of Dynamic Host Configuration Protocol (DHCP), not all IP addresses seen in our raw data are unique host identifiers. We use IP address to MAC address mappings from DHCP logs to ensure that all the flow records of each unique host are labeled with a unique identifier.

Flow-record processing: The second pre-processing step involves combining flows in different 5 minute intervals *that belong together from an application point of view*. For example, consider a File Transfer Protocol (FTP) application which transfers a very large file between two hosts. If the transfer span several 5 minute intervals then the flow records in each interval corresponding to this transfer should clearly be combined to represent the application level interaction. However, even for this simple well-known application, correctly representing the *application* semantics would in fact involve associating the FTP-control connection with the FTP-data connection, the latter of which is typically initiated from the FTP-server back to the FTP-client.

Applying such application specific knowledge to our flow-records is not feasible in general because of the sheer number of applications involved and the often undocumented nature of their interactions. We therefore make the following simplifying definition in order to turn our flows records into a data set that captures some application specific semantics. We define a **server** as any host that listens on a socket for the purpose of other hosts talking to it. Further, we define a **client** as any host that *initiates* a connection to such a server port. Clearly this definition does not perfectly capture application level semantics. For example, applying this definition to our FTP interaction, only the control connection would be correctly identified in terms of application level semantics. This client/server definition does however provide us with a very general mechanism that can correctly classify all transport level semantics while capturing some of the application level semantics.

To summarize then, during the second pre-processing step we combine or splice flow-records in two ways: First, flow-records for the same interaction that span multiple 5 minute intervals should be combined. Second, we combine two uni-directional flow-records into a single record representing client-server interaction.

To splice flow-records that span multiple 5-minute intervals, we use the 5-tuple of protocol and source/destination addresses and ports. We deal with the potential of long time intervals between matching flows by defining an *aggregation time* such that if the time gap between two flow records using the same 5-tuple exceed the aggregation time, the new flow-record is considered the start of a new interaction. If the aggregation time is too short, later flow-records between these hosts will be incorrectly classified as a new interaction. Making the aggregation time too long can introduce erroneous classification for short lived interactions. We experimented with different values of aggregation time

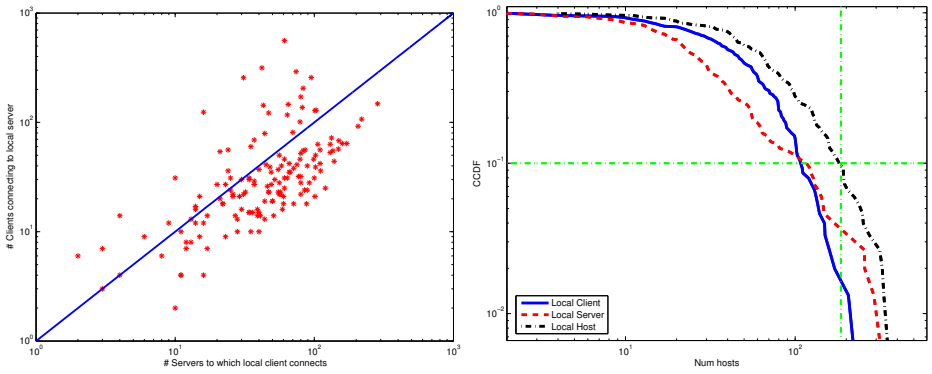
and found a value of 120 minutes provided a good compromise between incorrectly splitting flows that fit together and incorrectly combining separate flows.

The 5-tuple is again used to combine two unidirectional flows into a single interaction. For TCP and UDP, two flow-records are combined into a single record if the flows are between the same pair of hosts and use the same port numbers in a swapped fashion (i.e., the source port in one direction is the same as the destination port in the reverse direction). For ICMP traffic, flow-records are combined if they are between the same pair of hosts. The result of splicing two unidirectional flows together is an *edge-record* and we present the data as a directed graph in which each edge represents a communication between a client and a server and each node represents a unique host. The direction of the edge represents client/server designation and the labels on the edge indicate the number of packets and bytes flowing in each direction between the two nodes.

We evaluated the experimental error introduced by our flow-record processing as follows. We consider a 5 week subset of our total 11 week data set for this evaluation. We note that flows labeled with a client port number below 1024 and a server port number above 1024 is highly likely to be incorrect for all but a few services (as it is not consistent with the normal use of reserved ports), and the reverse (server port $<$ 1024, and client port $>$ 1024) are likely to be correct. We bound experimental error by calculating the ratio of incorrect to correct labeled flows based on this heuristic (after removing known services that violate this property, e.g., `ftp-data`, NFS traffic through `sunrpc`). This approximation yields a 2.187% role assignment error for all traffic, while the numbers for TCP and UDP are 2.193% and 2.181%, respectively. Each instance of mis-interpreted directionality introduces an additional flow into the data set. Hence, such errors do not change the structure of the community, but slightly amplify a host's role as a client or server.

Removing unwanted traffic: Since we are interested in characterizing the "useful" traffic in the enterprise network the third pre-processing step involves removing all graph edges for suspected unwanted traffic, such as network scans or worm activity. Doing such cleaning with 100% accuracy is infeasible because unwanted traffic is often indistinguishable from useful traffic. We use the following heuristics:

- *TCP:* We clean the data by removing all edges which do not have more than 3 packets in each direction. We chose the number three since a legitimate application layer data transfer needs more than three packets to open, transfer and close the TCP connection. This cleaning removes 16% of all edges indicating that a large fraction of traffic in the monitored network does not complete an application-level data transfer.
- *UDP:* We observe that there are two types of legitimate UDP uses. One is request/response type interaction such as performed by DNS and RPC. The other is a long lived UDP flow as used by many streaming applications. In both cases we expect an edge which performs a useful task to be associated with at least two packets, either in the same direction or in opposing directions. Therefore, we remove all edges for which the sum of packets in both directions is smaller than 2.
- *ICMP:* We do not perform any cleaning on the ICMP data since a single ICMP datagram is a legitimate use of ICMP.



(a) (b)

Fig. 1. (a) Scatterplot of 151 *local* hosts: Clients using the local host as a server and the local host talking to servers as a client. (b) CCDF:local host communication for total 11 week period.

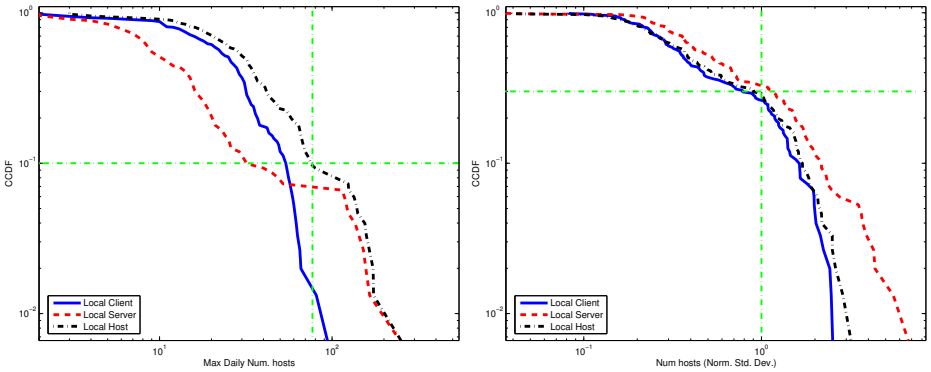
3 Results

In this section we present the COI analysis as applied to the enterprise data we collected. After pre-processing, the final data set we used for the analysis consisted of 6.1 million edge-records representing 151 *local* hosts and 3823 *internal* hosts and corresponding to 2.6 TBytes worth of network traffic. We will characterize only the set of 151 local hosts, but consider *all* their interactions, with both other *local* and *internal* hosts.

3.1 Community of Interest Set Size

First we evaluate the COI of the set of local hosts in our data set based on the broadest definition of COI. Specifically we consider the *number* of other hosts that each local host interact with. We look at the total number of such hosts and then do a breakdown based on whether the target local host was acting as a client or a server.

We first perform this analysis for all hosts over the *entire* measurement period. Figure 1(a) shows a scatter plot of the in/out-degree of the set of 151 local hosts considering all observed traffic. The Y-axis shows the number of clients connecting to the local host acting as a server (i.e., in-degree). The X-axis shows the number of servers that the local host connects to acting as a client (i.e., out-degree). Observe from Figure 1(a) that most hosts act as both client and server over the observation period. Indeed for the total traffic breakdown shown, *all* hosts act as both client and server during the measurement period. The general observation that most hosts act as both client and server, hold when data is analyzed on a per-protocol basis. Specifically, counting the number of hosts that acted purely as clients on a per protocol basis we get only 3 for TCP, 2 for UDP and 1 for ICMP. Similarly, counting the number of hosts acting purely as servers on a per protocol basis we get none for TCP, 2 for UDP and 5 for ICMP. Further, as indicated by the density below the diagonal line, the majority of local hosts are mostly acting as clients. For the plot shown, 111 hosts are below and 35 hosts above the diagonal line. The implication of the simple observation that most hosts act as both clients and servers, is that security schemes that rely on hosts acting exclusively as clients or servers, are likely to be infeasible in current enterprise networks.



(a) Maximum (b) Normalized standard-deviation

Fig. 2. CCDFs for the number of hosts communicated with on daily basis.

Figure 1(b) shows the empirical Complementary Cumulative Distribution Function (CCDF) of the number of machines that our local hosts communicate with for all traffic over the entire 11 week measurement period. The "Local Host" curve corresponds to the total number of hosts (either *local* or *internal*) that a particular local host interacts with, whether as a client or as a server. The plot shows that each of the local hosts communicates with a fairly small community of other hosts even over a period of several weeks. For example, 90% of the local hosts talks to fewer than 186 other hosts. Considering the client/server breakdown, the same holds true with local hosts interacting with a fairly small number of servers and clients. The final 10% of the "Local Server" curve shows that a small number of local machines acting as servers have higher numbers of clients talking to them than the other 90% of the local servers. These machines most likely correspond to "real" servers that serve a significant client population as opposed to hosts that are servers on the basis of the protocol interaction only.

We next look at the COI of each host on a *daily* basis and examine the statistical properties of these daily values over the complete observation period. First, Figure 2(a) shows the CCDF of the maximum daily number of hosts that each local host communicates with over the entire eleven weeks. These maximum number per day CCDFs are similar to those for the maximum over the entire measurement period, Figure 1(b), but the numbers are lower (i.e., the curves are "shifted" to the left). For example, the 90th percentile number for the "Local Host" curve in Figure 2(a) is only 77 compared with 186 for the same percentile in Figure 1(b). Also similar to Figure 1(b), there is an inflection at the 10% point in Figure 2(a) for the "Local Server" (and "Local Host") curves which is likely caused by "real" servers.

The relatively small sizes of the total number of hosts communicated with over the entire period as well as the small per-day maximums for the vast majority of hosts, suggest that a simple anomaly detection approach based on monitoring the normal COI size, has the potential to detect abnormal activities like port scans and worm spreads. These anomalies are often marked by a host communicating with a large number of other machines within a very short time span.

Next we consider the variability of the per-day COI size for each local host over the entire measurement period. Figure 2(b) shows the resulting CCDF of the normal-

ized standard deviation (normalized by the mean for each local host). Note that some of the variability is a result of hosts being inactive on some days, one contributing reason being telecommuting users. Hosts for these users might either be inactive because they are not being used, or in the case of notebooks, might not be visible to our monitoring infrastructure. The graph shows that approximately 70% of the local hosts have normalized standard-deviations in their per-day COI size that is less than 1. Assuming that all of the traffic in our data set was indeed legitimate, this would mean a simplistic approach to detect abnormal behavior for these hosts, based on a policy that restricts "normal" per-day COI size to 3 times the respective per-day means, would result in false alarms being generated only 5% of the time. Note also from Figure 2(b) that the standard deviation for the "Local Client" curve is less skewed than the "Local Server" curve. This suggests that on a daily basis the number of servers which a local client talks to, is more stable than the number of clients that talk to a local server. The implication of this is that network management policies derived from observations close to the initiator of communication (client) is likely to be more stable than policies derived from traffic close to the communication responder (server).

3.2 Core Communities of Interests

We next explore our two example core COI definitions *Popularity* and *Frequency* core COIs and their interactions.

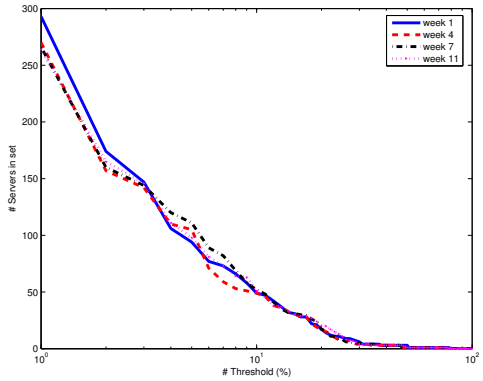
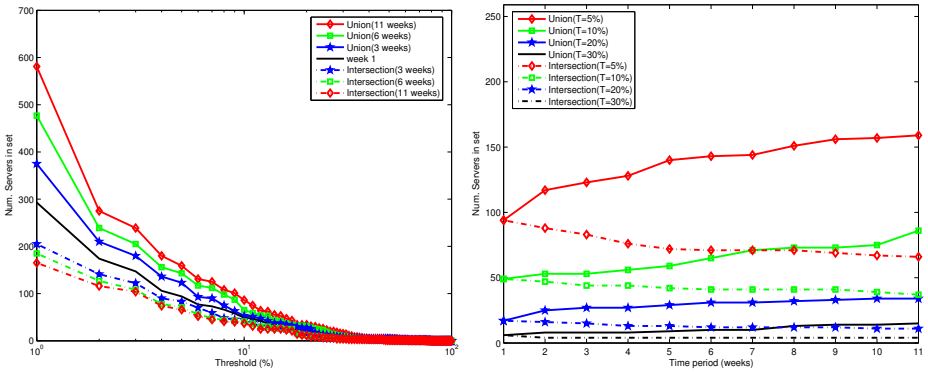


Fig. 3. Size of *Popularity* COI set for all traffic.

***Popularity* COI:** Recall that for the *Popularity* COI we consider a host to be part of the COI for a group of target-hosts if the percentage of target-hosts interacting with it exceeds a threshold T over some period of interest Y. Here we identify the *Popularity* COI of the local hosts from a client view point, for each of the 11 weeks in our data set (i.e., Y is one week). Figure 3 shows the size of the *Popularity* core COI set as a function of the threshold T for 4 equally spaced weeks out of the total 11 weeks, for traffic across all protocols. The graphs shows the expected decline of the set size as one progresses from a threshold of 0% (which would include all hosts) to a threshold of



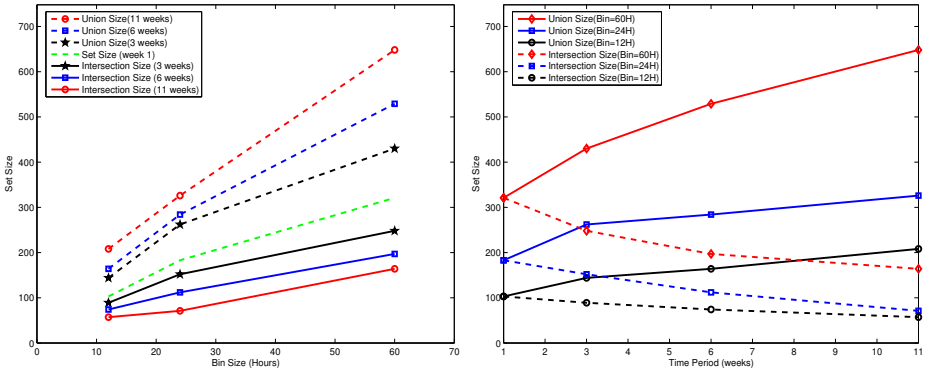
(a) (b)

Fig. 4. *Popularity* COI: Union and intersection set size (a) As a function of threshold T (b) As a function of length of time window N weeks ($1 \leq N \leq 11$).

100% at which point the size is expected to be very small as it would require all target-hosts to communicate with each member of the set. We observe that the size of the core COI set as a function of the threshold is very similar across the different weeks. This suggests that, deviations from the *Popularity* COI size distribution, for a set of hosts monitored over time, would be a strong indication of a network anomaly.

While the stability in the core COI set size is encouraging, we are also interested in the stability and predictability of the core COI set *membership*. To evaluate this, we determine the core COI set for each week in our data and then explore how the membership of these sets change over the measurement period. We do this by calculating the union (the set of servers that belong to the core set in at least one week) and the intersection (the set of servers that belong to the core set in every week) of the COI sets. For any two sets the difference between the size of the union and intersection represents a measure of the "churn" between the two sets - that is the total number of elements that needs to be added or removed from one set to transform it to the other set. Therefore, for a window of N COI sets, the difference between the union and intersection of **all** the sets, represents an upper bound on the churn between any two pairs in N. By looking at this bound we get a worst case estimate of how much the COI membership changes over the time window (N). By progressively increasing the length of the time window, we determine how this worst case estimate changes over time.

Figure 4(a) depicts the sizes of the union and intersection of core COI sets for weeks 1 to 3, 1 to 6 and 1 to 11, as a function of the threshold T for all traffic. For comparison the core COI set size for week 1 is also shown. For all curves (i.e., for all time periods considered), the difference between the union and intersection set sizes, i.e., the churn, tends to decrease as the threshold increases. Figure 4(b) shows the same data, but in this case we show the union and intersection set sizes for selected thresholds for increasing time windows N of interest (1 to 11 weeks), starting from week 1, i.e., 1 to 2 weeks, 1 to 3 weeks, etc. As expected, the union set size size increases and the intersection set size decreases for a given threshold as the time window increases. Notice though from Figure 4(b) that for any threshold, the union and intersection set sizes change in a sub-linear fashion with increasing N. In fact the intersection seems to flatten within 6 to



(a) (b)

Fig. 5. *Overall Frequency* COI: Union and intersection set size (a) As a function of bin size Z (b) As a function of length of time window N weeks ($1 \leq N \leq 11$).

8 weeks. While the union set size shows a continued small growth, the maximum union set size, for the thresholds considered, did not increase beyond a factor of 2.5 over the entire time window of interest. The observed intersection behavior implies that servers present in the *Popularity* COI in the first week, have a high probability of remaining in the COI for the entire period. This holds true independent of the threshold. For example for the 5% threshold, after the 11 weeks 66 of the initial 94 servers are still in the intersection set. The relatively small growth of the union set implies that even though servers are constantly added to the set, the number of additional servers added in a week is low. Even for the 5% threshold, the number is low enough that applications requiring "manual" verification of the status of new servers, i.e., whether they are legitimate servers, would be feasible.

The above results indicate that rapid changes in the *Popularity* COI membership would be an additional indication of anomalous network behavior. Note that this holds true if a large number of popular servers are either rapidly added to **or** removed from the COI set.

***Frequency* COI:** We have also defined a core COI that captures the frequency of interaction. Recall that our *Frequency* Core COI considers a host to be part of the COI if a target-host interacts with it at least once in every time bin Z over some larger time period of interest Y. To evaluate this core COI definition we calculated the *Frequency* COI (client perspective) for each host for each week of our data (i.e., Y is one week) for bin-size (Z) values 12, 24, 60 and 120 hours. For each week, we define the *Overall Frequency* COI to be the union of all per-host *Frequency* COIs for a given bin-size. We explore how the membership of this set changes over time.

Similar to the approach above for *Popularity* COI, we determined the union and intersection of all the *Overall Frequency* COI sets for a specific time window of interest N. Figure 5(a) shows the size of the union and intersection sets for N equal to 3 weeks, 6 weeks and 11 weeks, for different bin sizes. (Note that we did not include a bin size of 120 hours for this plot as that would include all hosts that communicated in a week as part of the core COI for the week, which would be too inclusive for a core COI.) As a reference point, Figure 5(a) also shows the size of the *Overall Frequency* COI for

the first week. As the bin size increases, the COI set becomes more inclusive and as expected the set size (shown for week 1) increases as the bin size increase to 60 hours. The same holds true for the union and intersection set sizes, i.e., for a particular size of N (e.g., 3 weeks), the size of both the union and intersection sets increase as the bin size increases. Next consider how the union and intersection set sizes for a particular bin size change for different values of N. For example, for a bin size of 24 hours, we see that the union set size increases as N increases from 3 to 6 to 11 weeks, while the intersection set size decreases for the same values of N. Again this behavior is expected, but it is interesting to note that this increase and decrease is not linear with respect to the increase in N. For example, doubling N from 3 weeks to 6 weeks does not result in doubling the union set size or halving the intersection set size. This is best shown in Figure 5(b), which depicts the union and intersection set sizes for each value of N (1 week, 3 weeks, 6 weeks and 11 weeks).

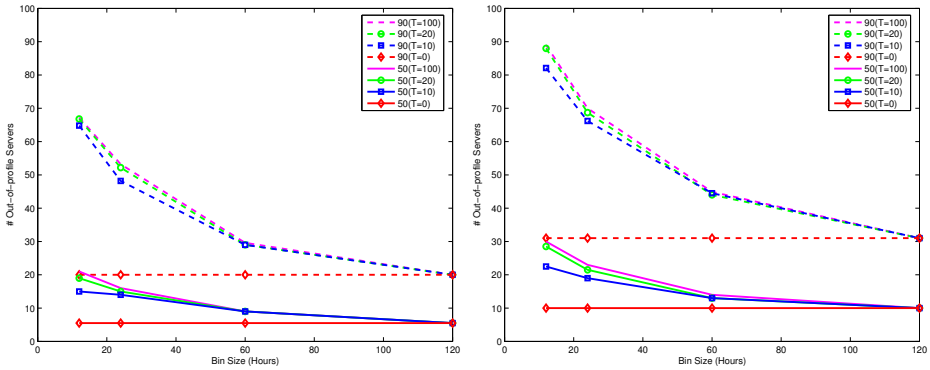
The above behavior of the *Overall Frequency* COI as a function of increasing N is similar to the behavior of the *Popularity* COI as a function of increasing N as shown in Figure 4(b). As for the *Popularity* COI, the results for the *Overall Frequency* COI indicate that rapid changes in the COI membership would be an indication of anomalous network behavior.

Overall COI: Recall that the *Popularity* and *Frequency* COI definitions attempt to capture different types of interactions that should be considered part of a core COI. Above we explored the churn in the *Popularity* and *Frequency* COIs separately, and focused on the churn in the membership of these sets. In contrast to this aggregate view, another way to explore variability is to inspect how the ability to capture the communication behavior for individual hosts is impacted by the churn in these COI sets. This is the goal of the study described next.

The *Popularity* COI is a function of the threshold parameter (T), while the *Frequency* COI is a function of the bin-size (Z) as defined earlier. For this part of the study, we computed, for a range of $\{$threshold,bin-size$\}$ pairs, the *Overall* COI set for the **first week** of our data by combining (using set union) the *Overall Frequency* COI with the *Popularity* COI of the total local host set. Should this *Overall* COI accurately capture the core interactions of the target hosts in subsequent weeks, then one would expect that few of the target-hosts' interactions would be with hosts not in this set. We define interactions with hosts outside of the *Overall* COI to be *out-of-profile*.

For each local host we determined the number of out-of-profile interactions for subsequent weeks of our data. We calculate a distribution of the out-of-profile interactions across all hosts for each of the $\{$threshold,bin-size$\}$ pairs. The results for this analysis are shown in Figures 6(a) and (b) for 6 and 11 weeks respectively. The figures depict the 90^{th} and 50^{th} percentiles for these distributions for a number of threshold values and as a function of bin-size.

We had discussed in Section 2.1 the situations under which the *Popularity* and *Frequency* COIs are identical. This explains why each set of curves (50^{th}, 90^{th} percentiles) converge in the 120 hour bin-size value in Figures 6(a) and (b). It also explains the horizontal lines (i.e., the cases where threshold is zero): in these latter cases the *Popularity* COI already includes all servers, so the union with the *Frequency* COI does not add any



(a) 6 weeks (b) 11 weeks

Fig. 6. Out-of-profile interactions for *Overall* COI for different threshold values and as a function of bin-size.

members to the *Overall* COI, and the number of out-of-profile interactions is therefore independent of the bin-size.

The two horizontal lines in each figure correspond to the case where the threshold is zero, i.e., where the *Popularity* COI includes all hosts that acted as servers in week 1. From Figure 6(a), (the six week period), 50% of the local hosts (50^{th} percentile line), had less than 6 out-of-profile interactions, while 90% of the local hosts (90^{th} percentile line) did not exhibit more than 20 out-of-profile interactions over the entire period. The corresponding numbers for the full 11 week period shown in Figure 6(b) are 10 and 31 out-of-profile interactions for 50% and 90% of the local hosts respectively.

Now we consider the impact of the *Popularity* COI set by looking at the graphs in both figures for a fixed bin-size, (e.g., 60 hours). For this bin size, consider the 4 values of the six week distributions (Figure 6(a)). Notice that there is a significant difference between the case where the threshold is zero and the case where the threshold is 10%, when compared to the difference between the 10% and 20% (or even 100%) points. This also holds true for the 11 week distributions. This suggests that as the *Popularity* set becomes more inclusive (i.e., as the threshold gets closer to zero), it contributes *more significantly* to the *Overall* COI set. This seems to suggest that in this region the *Popularity* set indeed captures the important infrequent interactions that should be part of a host's COI. The relatively small difference between the values of the out-of-profile interactions for each set for thresholds between 10% and 100% seem to suggest that the *Overall Frequency* COI already captures most of the servers that the *Popularity* set would capture at such thresholds. This in turn suggests there may be significant overlap between the servers that communicate with a larger fraction of clients and those that interact frequently with clients, in our data set. Note that the increase in violations going from 6 to 11 weeks, does not increase proportionally with the increase in time.

In summary the graphs show that an *Overall* COI derived from one week's worth of data is not sufficient to fully capture host interactions for subsequent weeks. However, using the most inclusive variant of this COI definition (i.e., where threshold T is zero or where the bin size Z is 120), 90% of the hosts experienced on the average less than 3 violations per week (for the 11 week graph). Similarly, for the most restrictive COI

we considered (i.e., T equal to 100 and Z equal to 12), 90% of the hosts experienced less than 9 violations per week over the 11 week period. Both rates are low enough that they would not preclude COI derived application that require human involvement. In fact, 50% of the hosts would only experience one third of these violations.

4 Conclusions

In this paper we presented our methodology and initial results for characterizing communities of interest (COIs) for hosts communication in data networks. We presented example definitions for COI that attempt to capture different characteristics of the underlying communities. We explained how we collected our measurement data and the pre-processing steps that were required before analysis. While this work is still maturing our initial results indicate that:

- Hosts typically act as both clients and servers which implies that any management applications or policies will have to explicitly deal with this.
- Using a very broad COI definition we saw similar distributions for the COI size over daily and monthly timescales, suggesting some stability in the COI for the community as a whole.
- COI definitions that represent core host interactions, showed significant stability of the COI over timescales of several weeks.
- Core COIs calculated over a part of our measurement period were also able to capture the actual host interaction in the remainder of the data fairly well.

We are continuing the presented work by moving from the presented aggregate COI characterization to finer grained per-host characterization. Our ongoing work aims to provide models that accurately capture host behavior. And our ultimate goal is to be able to apply such models to the many challenging network management tasks presented in the introduction.

References

1. Emily M. Jin et. al., "The structure of growing social networks," *Physics Review E*, vol. 64, pp. 845, 2001.
2. Ravi Kumar et.al., "The web and social networks," *IEEE Computer*, vol. 25, no. 11, pp. 32–36, 2002.
3. J. Kleinberg, "The Small-World Phenomenon: An Algorithmic Perspective," in *Proceedings 32nd ACM Symposium on Theory of Computing*, 2000, pp. 163–170.
4. J. Kleinberg, "Navigation in a small world," *Nature*, vol. 405, pp. 845, 2000.
5. Godfrey Tan et. al., "Role Classification of Hosts within Enterprise Networks Based on Connection Patterns," in *Proceedings of 2003 USENIX Annual Technical Conference*, June 2003, pp. 15–28, San Antonio, TX.
6. Corinna Cortes, Daryl Pregibon, and Chris T. Volinsky, "Communities of interest," *Intelligent Data Analysis*, vol. 6, no. 3, pp. 211–219, 2002.
7. Chuck Cranor et. al., "Gigascope: a stream database for network applications," in *Proceedings of ACM SIGMOD*, June 2003.