Communities of Interest for Internet Traffic Prioritization

Andrew J. Kalafut
Indiana University
Bloomington, IN 47401
Email: akalafut@cs.indiana.edu

Jacobus van der Merwe
AT&T Labs – Research
Florham Park, NJ 07932
Email: kobus@research.att.com

Minaxi Gupta
Indiana University
Bloomington, IN 47401
Email: minaxi@cs.indiana.edu

Abstract—**Communities of Interest (COI) have been studied in the past to classify traffic within an enterprise network, and to mitigate denial-of-service (DoS) attacks. We investigate the use of Communities of Interest (COIs) to prioritize known good traffic on the Internet. Under our system, an ISP may construct a COI for each of its enterprise customers. The COI would contain entities which have previously had good communications with the customer. These COIs could then be used in combination with traffic differentiating mechanisms during periods of heavy traffic in order to prioritize traffic from communicating entities known to be good. We show that it is possible to construct an effective COI from information which would be available to an ISP about its customers, specifically sampled Netflow data. We investigate various heuristics to determine which flows actually represent good traffic whose endpoint should be inserted into the COI, and show that our heuristics are effective in differentiating wanted and unwanted traffic.**
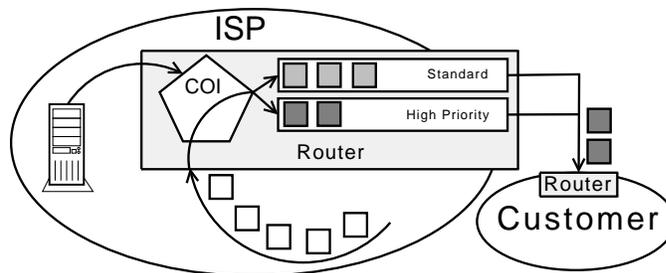
Fig. 1. A good-COI for a customer is computed offline by the provider and sent to its router at the interface to that customer. Packets arriving at this router are checked against the COI and prioritized into the appropriate queue.

I. INTRODUCTION

The Internet's best-effort communication model is capable of supporting a wide variety of services and applications. However, it performs poorly in terms of differentiating the relative importance of traffic. This deficiency of the Internet is particularly acute when it comes to differentiating between wanted and unwanted traffic. In particular, the best-effort, unaccounted service model directly enables denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks.

A *proactive* DDoS mitigation strategy, in which the prediction of communication patterns based on historic communication is used to prioritize between wanted and unwanted traffic, recently illustrated promising results [1]. A particularly attractive aspect of this work is that *network derived intelligence* can be used to inform the use of existing low-level router mechanisms to perform differentiation. The approach holds the promise of providing basic DDoS protection at a massive scale, provided that appropriate network intelligence can be readily derived.

This paper focuses on the feasibility of deriving network intelligence to inform proactive DDoS mitigation. Specifically, our goal is to determine whether network intelligence derived from data that is readily available to most service providers, historic sampled *Netflow* [2], can be used to predict future traffic patterns. We approach this problem from the point of view of an ISP who might want to provide differentiated services to its customers based on their historic traffic characteristics. We therefore perform the analysis at the granularity of ISP customers connecting to a backbone network, where "customers" are enterprise networks or smaller ISPs.

For each customer network, our approach is to attempt to derive the set of "entities" with which each of these networks engage in "wanted" communication and to use that to predict future wanted communication. We use the term *community-of-interest (COI)* to refer to this set of communicating entities [1], [3], [4] and further refer to a "good-COI" as the set of communicating entities engaged in wanted communication. Specifically, the good-COIs in our approach consist of network prefixes derived from source IP addresses of traffic destined to each customer. Though various mechanisms for deriving this information are possible, it is easiest to think of the process as a periodic, offline process. When a good-COI is derived, it is pushed to the ISP routers at the perimeter, where they interface with the relevant customer. (Recent work in the IETF [5] would allow for the dissemination of good-COIs in a scalable fashion.) The routers would match traffic for each customer against the corresponding good-COI and prioritize "wanted" traffic over other traffic. The higher priority can be accomplished simply by queuing the wanted traffic in a higher priority queue [6]. This process is shown in Figure 1. Other existing quality-of-service (QoS) mechanisms [7], [8] that allow for differentiated treatment of packets could also be utilized. Note that most modern routers readily support many mechanisms to enable differentiated treatment of packets. For example, router vendors already offer products with sophisticated classifiers and policing mechanisms [9], [10].

The main contributions of our work are:

- We develop heuristics that can be applied to sampled flow data to derive good-COIs for Internet traffic. These heuristics are based on generally-accepted characteristics of good traffic.
- We evaluate these heuristics using ground truth based on traffic destined to a single enterprise network, and traffic observed within this network's firewall. The heuristics based on history of frequent communication perform the best. They produce good-COIs that are very effective in differentiating wanted traffic from unwanted traffic.
- We evaluate the utility of using good-COIs to predict wanted traffic for additional ISP customers using a large corpus of Internet data from more than 50,000 customers and find that the system is likely to be beneficial to at least 78%.

The outline of the remainder of the paper is as follows. Section II discusses related work to provide context for our work. In Section III we discuss the heuristics we use in constructing our good-COIs. We describe our data in Section IV. The performance of these heuristics is presented in Section V and Section VI. Finally, we conclude the paper in Section VII.

II. RELATED WORK

Using the inherent structure afforded by communities-of-interest (COIs) to impose structure on communicating entities has been performed in the context of the phone network [11] as well as enterprise data networks [3], [4]. The proactive DDoS mitigation work [1] that motivates our own work also used COIs derived from Internet data. However, in that work the COI was derived from unsampled flow records which does not offer a scalable solution for deriving COIs for **all** customers connected to an ISP. Further, [1] made use of both a good-COI (as we do) and a bad-COI representing a set of unwanted communicating entities. In our work we limit out attention to deriving the good-COI. This is motivated in part by recent results that indicate very high churn in botnet populations [12], which indicate that unwanted traffic might in general not have stable communication patterns and thus not provide significant utility to our approach.

Most DDoS mitigation approaches [13]–[18] are reactive in nature rather than the proactive approach adopted for our work. Our work has some similarity with the "off-by-default" [19] approach where an end system explicitly indicates to the network infrastructure what traffic it wants to receive. Rather than explicitly signaling this information to the network, however, in our case the network indirectly derives this information from traffic patterns. Similar to our own, a proactive protection approach is also followed in the surge protection work presented in [20]. However, in [20] the emphasis is on protecting the network infrastructure by exploiting stability in *aggregate* traffic flows, whereas the focus of our work is to provide fine grained per-customer protection by using stability in the communication patterns of enterprise networks.

III. HEURISTICS TO IDENTIFY GOOD COIS

Our goal is to find heuristics that when applied to sampled Netflow data on each customer interface of an ISP can extract information to produce accurate good-COIs for that customer. The ISPs can in turn use these good COIs to prioritize good traffic over unwanted traffic as desired. Here, we outline the basic heuristics we tested for their efficacy in identifying good flows. While it is clear that none of these heuristics will classify all flows correctly, since we are prioritizing traffic and not filtering it a small amount of misclassification can be considered acceptable.

1) **Low Port Number Heuristic:** Low port numbers generally require root access. Even though services on these ports could be compromised, a low port number is in general an indication of a legitimate service. The low port number heuristic leverages this observation to identify good flows. Notice that higher port numbers may sometimes indicate good traffic depending on the specific applications used at a specific site. Further, low port numbers may not have a reason to be viewed as good in specific cases. However, obtaining such site-specific information is not always be possible. In this paper, we use this heuristic assuming the lack of site-specific information. Hence, we consider a flow to be good if either of the port numbers involved in the connection are below 1024.

2) **Low Packet Count Heuristic:** This heuristic leverages the fact that most flows on the Internet are short [21]. It checks if a flow has seen five or fewer packets, as measured by sampled Netflow. This translates to approximately $2,500$ or fewer actual packets in the flow due to the 1 in 500 sampling in our primary data set. If a flow passes this test, we consider it good. We note that this heuristic may also be used to exclude short flows resulting from port scans and denial-of-service (DoS) related connection attempts. However, we are unable to isolate the later due to the 1/500 packet sampling rate.

3) **Reverse Flow Seen Heuristic:** If a flow in one direction is reciprocated by a flow in the reverse direction, it is in indication of a two-way connection. Although two-way communication does not guarantee the goodness of traffic in cases such as spam or a visit to a malware-containing Web site, most good traffic tends to have a request-response nature. For each flow, this heuristic checks to see if we have seen a flow between the same pair of IP addresses in the opposite direction within the previous seven days. If we have, then we consider the flow good, otherwise, we consider it suspect. A similar heuristic has been used previously in [22] to identify good traffic, although not for sampled data.

4) **Recent History Heuristic:** This heuristic is based on the assumption that frequent communication indicates goodness. A common example supporting this heuristic is Web browsing, when a client may visit multiple pages on the same Web site. For each sending IP address,

this heuristic checks if the IP address was seen in the previous hour. If so, the flow is considered good. This in effect requires a sender to be seen twice within an hour to be added to the COI. Recall that these heuristics are being used to construct COIs, not to classify flows on their own. Therefore, no IP address would be considered as bad the first time it is seen. If it returns within an hour, it will be marked as good and inserted in the future good-COI.

5) **Non-recent History Heuristic:** This heuristic is in some ways the complement of the recent history heuristic. Like the recent history heuristic, it checks if an IP address has been seen previously. However, instead of checking if it was seen in the previous hour, it checks if it was seen in the previous seven days except for the previous hour. The motivation behind this heuristic is the observation that most attacks are short-lived [23], so if an IP address has been seen multiple times in flows not close together, it is likely to be legitimate.

6) **No Heuristic:** Where appropriate, we compare the results of using the heuristics with that of not applying any heuristic, instead simply adding all observed IP addresses to the COI. This serves as a baseline to tell how much of an effect the heuristics have.

IV. DATA SOURCES AND OVERVIEW

While our aim is to apply the heuristics to sampled Netflow data available to ISPs, we must test them against more complete data to validate their effectiveness. Here, we describe the data sets we use toward this goal.

A. Data Sources

Sampled Data: Our primary data set is sampled Netflow data, to which we apply the heuristics. We collect this data from over 200 routers on a tier-1 ISP network. The Netflow data contains many pieces of information at flow granularity, including the start and end times of flows, source and destination IP addresses and ports, transport layer protocol, and number of packets in the flow. It also contains the ingress and egress interfaces where the flow enters and leaves the ISPs network, useful in identifying the origin and destination sites for each flow.

The Netflow data is sampled in two ways. First, the routers do packet sampling at a rate of 1 in 500. In addition, they perform *smart sampling* [2], a technique to get a reliable estimate of detailed usage from only a subset of flow records by exploiting the fact that a large fraction of usage is contained in a small fraction of flows. By preferentially sampling larger flows over small ones, one can control the volume of statistics while simultaneously controlling the variance of statistical estimates derived from them. Smart sampling entails balancing those two objectives in an optimal manner.

Unsampled Data: In order to test the efficacy of our heuristics, we utilize two unsampled data sets. The first, referred to as *unsampled good data* subsequently, was collected inside of a firewall at one "customer" site. Assuming the firewall is

accurate and has already filtered out the bad flows, this data would serve as a ground truth for testing if a heuristic does well in identifying good data from sampled Netflow. This data was collected using a Gigascope device [24] and contained good flows to one /16 prefix. The second data set, referred to as *unsampled full data* subsequently, is unsampled data destined to the same customer site but before it was subject to the firewall. Combined, these two data sets serve as ground truth and provide us a full view of all the flows to this customer site. By taking flows observed both before and after being subject to the firewall as good, and counting those which did not pass the firewall as bad flows, they allow us to distinguish good traffic from the rest of the traffic.

B. Basic Statistics

We now present in Table I basic statistics for Netflow (sampled), unsampled good data, and unsampled full data for the customer site where we performed a detailed analysis. All numbers presented are averages per day, over the 16 days we use to test the good-COIs.

TABLE I
BASIC STATISTICS FOR EACH DATA SET

	Sampled	Unsampled Good	Unsampled Full
Number of Flows	13,738	2,224,698	3,695,850
Number of Packets	106,972	146,850,472	168,076,457
Traffic Volume	146 MB	152.0 GB	181.7 GB
Number of IPs	919	21,487	62,720
Number of /24s	658	12,599	48,880

V. EFFECTIVENESS OF HEURISTICS

We begin by evaluating how the heuristics we proposed in Section III perform individually and in combination with other heuristics. We test each heuristic under various parameters. The analysis presented in this section focuses on a single customer site since unsampled ground truth is required for this purpose.

A. Individual Heuristics

A good heuristic would correctly classify a large percentage of Netflow records. Here, we examine how effective individual heuristics are in doing so. We build good-COIs for each heuristic by applying the heuristic to Netflow data for 30 days in a row. Any time the heuristic marks a flow as good, we add the /24 containing the sender's IP address to the good-COI. Once the good-COI is built, we apply it to the flows from the following day of ground truth data, checking which it classifies as good or bad, and how they were classified in the ground truth data itself. We repeat this process for 16 days of ground truth, building a new good-COI shifted forward by a day each time.

In order to directly compare the heuristics, we plot the percentage of flows from each day of ground truth correctly classified by each heuristic in Figure 2. *The history-based heuristics and the one based on the presence of a reverse flow outperform all others on most days.* The low port heuristic also is best on a few days, but not many. The packet count
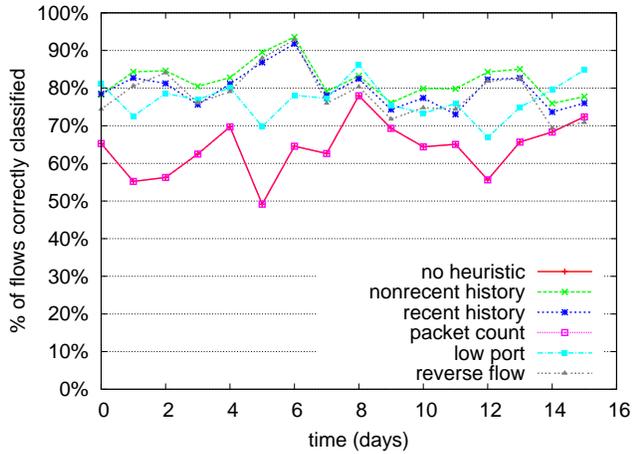
Fig. 2. Overall effectiveness of good-COIs based on each heuristic using 30 days of Netflow
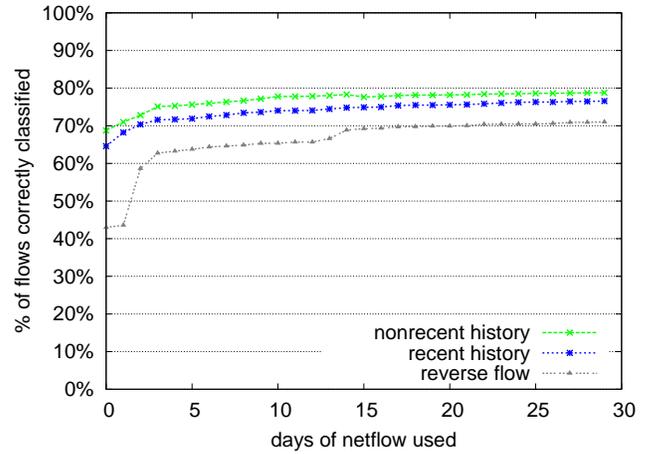

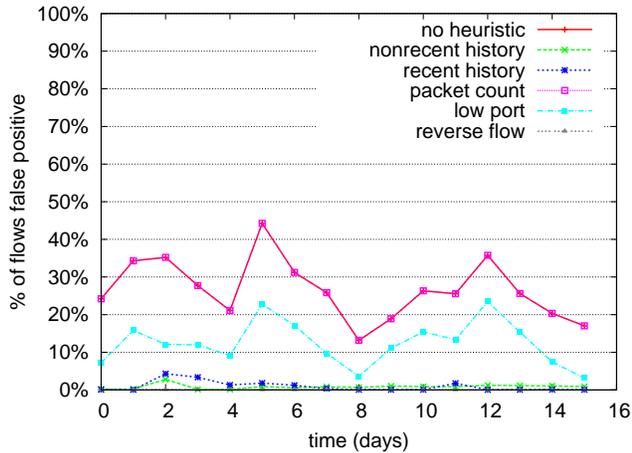Fig. 4. Effect of varying number of days of Netflow used to construct good-COI for a single day

Fig. 3. Daily false positive rate for good-COI based on each heuristic using 30 days of Netflow

heuristic has little effect, it is visually indistinguishable from no heuristic in the graph.

Given that our goal is to prioritize good traffic, it is important for our classification to produce low false positives, i.e., fewer bad flows classified as good. Having low false negatives, i.e., fewer good flows classified as bad, is an important metric, but more important in scenarios where traffic is being filtered. Figure 3 shows the false positives each day for each heuristic. *The good-COI based on the reverse flow heuristic produces almost no false positives, a maximum of 0.1%. The recent history and non-recent history heuristics produce slightly more, with maximums of 4.3% and 2.8%, while the remaining heuristics do noticeably worse.* Incidentally, the non-recent history, recent history, and reverse flow heuristics classify up to 23.0%, 26.2% and 30.5% of flows as false negatives. Even though the other heuristics have less false negatives, this is not a major concern since our goal is to prioritize, not filter.

Experimenting with amount of Netflow used: The analysis

thus far used 30 days of Netflow data to construct good-COIs. Ideally, it would be best to use as little information as possible without compromising on the accuracy, for this would require less information to be maintained and less processing to build the good-COI for each customer interface at the ISP. We now experiment with varying number of days of Netflow data. We focus our analysis here on the three best heuristics.

Figure 4 shows the effect of varying the number of days of Netflow data used to construct a good-COI to classify a single day of data (day 15, the last day, from Figure 2). We see similar trends for non-recent and recent history heuristics. The amount classified correctly has a notable increase for the first few days, but near day 5 it levels off having only marginal increases in accuracy per day by the time it gets to day 10. The reverse flow heuristic shows a similar pattern except for a much greater increase in the first few days. *This indicates that at least for the day used here, more than about 10 days worth of history is probably not providing enough benefit to justify its use.* When investigating false positives and negatives, we find that using more Netflow decreases false negatives while false positives depend very little on how much Netflow is used. Similar to the overall accuracy, the change in false positives is most significant for small numbers of days of history, having little effect beyond 10 days of Netflow used.

Results from classifying a single day are promising, but we must look across more days to be certain 10 days is sufficient. In Table II we show the accuracy of the top three heuristics when using 5, 10, or 30 days of Netflow to build the good-COI. We see only a small difference between using 10 days of Netflow, and using 30. The greatest difference here is 1.4% for the reverse flow heuristic, while the most accurate heuristic, non-recent history, only has a difference of 0.7%. The difference between 5 and 10 days is greater, 2.9% for non-recent history. *Although there is some variation based on the heuristic and the day being considered, it appears that 10 days of history used to construct the COI produces COIs almost as good as those with 30 days, without requiring as*

much history to be maintained.

Heuristic	% correctly classified		
	5 days	10 days	30 days
Non-recent history	77.4%	80.3%	81.0%
Recent history	75.1%	77.8%	78.5%
Reverse flow	71.7%	75.5%	76.9%

Varying parameters for heuristics: A few of the heuristics can be varied in various ways. Focusing again on just the top three heuristics, the amount of history considered for the recent history heuristic, or excluded for the non-recent history heuristic can be varied. There are not any obvious variations for the reverse flow heuristic. We now explore the effect of varying these parameters, using 10 days of Netflow to build the good-COI.

In addition to the usual one hour of history, we consider two hours and one day of history for the recent history. Similarly, for non-recent history we consider seven days of history except for the most recent one hour, two hours, and one day. We additionally consider a *simplified history heuristic*, consisting of the full seven days without anything removed. For the variations of the non-recent history heuristic, on average the simplified history heuristic performs best, correctly classifying 81.2% of flows, a slight gain over the variation we started with, which removes a single hour. However, this varies day to day, with our original version sometimes performing better. The situation is more clear for the recent history heuristic. Considering greater lengths of time for this heuristic clearly improves the accuracy of the COI based on it, with the seven days of the simplified history heuristic being best. Although the recent and non-recent history heuristics were based on different reasoning, the performance of this simplified history heuristic indicates that what really matters is whether a similar flow is seen twice within a window of several days, without regard to whether the sightings are within an hour of each other. The marginal gain of this heuristic over the other history heuristics also shows that they mostly were identifying the same flows. There is still, however, a trade-off to consider. Though we do best with seven days of history, this requires maintaining an additional seven days of Netflow. Using one day instead of seven provides 80.0% accuracy, while not requiring as much extra history to be maintained.

B. Combined Heuristics

Thus far, we have only discussed how well the heuristics perform individually. We also evaluated if combining multiple heuristics would produce better results than individual ones. There are two possible methods of combination, add a /24 to the good-COI if it meets either one of the heuristics (*logical OR*), or add it to the good-COI only if it meets both of the heuristics (*logical AND*). We investigated both methods of combination for a pair of heuristics at a time. *Overall, we found that combining the heuristics did not gain much, with*
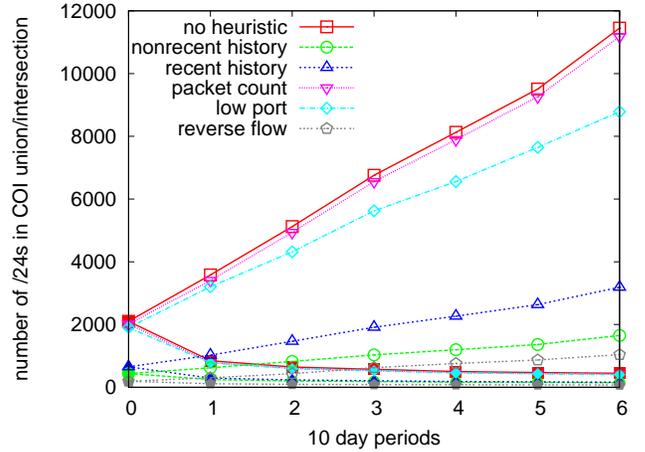

Fig. 5. Overlap of COIs based on non-overlapping 10 day periods of Netflow. Upward sloping lines with unfilled markers show unions of COIs. Downward sloping lines with filled markers show intersections.

the best combination classifying only 0.03% additional flows correctly.

C. Other good-COI Properties

Aside from accuracy, there are other properties we would like in the good-COI, especially since the accuracy we receive from the three best-performing heuristics is similar. One of these is the size of the good-COI. The smaller the COI is, the less space it will take up and the more efficient it will be to apply. The other is degree of churn. If the churn in the IP prefixes that belong to the good-COI is low, the COI will not need to be regenerated as often.

Figure 5 plots the union and intersection of good-COIs built using consecutive non-overlapping 10 day periods of Netflow data. We can see how both properties are represented in each heuristic in this figure. In the first 10 day period considered, the recent, non-recent history, and reverse flow heuristics put 651, 436, and 191 /24s in their good-COIs respectively, while the others more than double this. We also see that the difference between union and intersection grows slower for these than for others, indicating that these heuristics have more consistency.

VI. GOOD-COIs FOR MULTIPLE CUSTOMERS

In Section V, we investigated how well the heuristics classify traffic for one site. Because we can not make measurements inside other customer sites, we have no way of judging how the heuristics will perform at other customer sites. However, we can look for properties which may give an indication of how effective a COI would be. One property essential to a COI is *consistency in traffic*. A good-COI classifies traffic based on the information built up from the traffic seen on previous days. A lack of consistency would make the good-COI less useful since traffic from previous days would not be able to tell us what to expect on the next one.

We can get some idea of this property by building good-COIs and testing how much of the overall traffic seen on the next day matches the COI. For this purpose, we build COIs
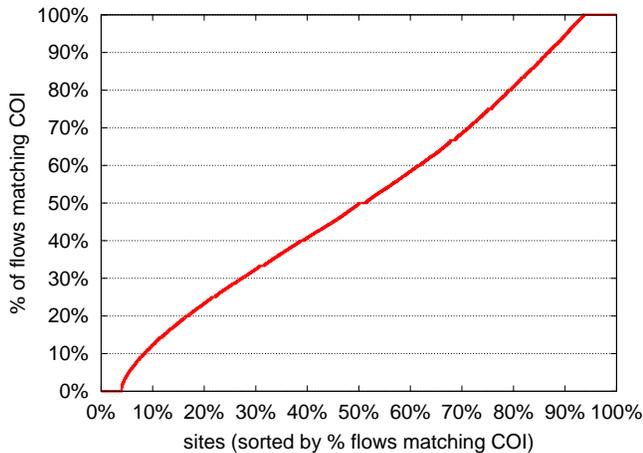
Fig. 6. Percent of flows matching good-COI for over 50,000 sites

for in excess of 50,000 sites, chosen by selecting the subset of network egress points which are the eventual destinations for traffic observed at a single router on a single day.

A. Results

Figure 6 shows for each site what percentage of flows seen on the test day matched the good-COI for each site. We see a wide variation on how many flows match a good-COI for each site. *In 6.3% of sites, all of the test data matches the good-COI*, indicating a COI would work well for these. In 4.0% of sites there is no match between the COI and the flows observed on the testing day. For these sites, it is likely that the system we propose will not be helpful since there is no consistency in their traffic.

While it is very likely that a COI would help those with 100% match and not help those with 0% match it is less certain how it would affect those sites in between. The single site we tested in detail averages 25.7% of flows matching the COI using this method. *78.3% of the sites we tested have a higher match with their testing data than this single site.* This indicates that while a good-COI may not be beneficial to every site, it is likely to be beneficial to a majority of them, since most have better consistency than the single site for which we have shown the system would work.

VII. CONCLUSION

We have shown that it is feasible to develop good-COIs that can help ISPs prioritize traffic per customer. The COIs can be derived only based on easily-available sampled-Netflow data. Specifically, simple history-based heuristics, or a heuristic based on seeing a flow in both directions, applied to the data produce good-COIs which are reasonably accurate for the purpose of traffic prioritization. While we investigated the issue extensively for one site and looked into the efficacy of the system for many other sites, there are various other aspects that need to be analyzed in detail in order to ascertain the feasibility of our proposal. Specifically, it would be useful to examine other properties of those customer sites where the good-COIs

resulting from our approach perform poorly. Further, our entire evaluation was on real traffic. It may be useful to simulate the efficacy of our system under a wider variety of traffic patterns, including those of common attacks not covered by our data.

REFERENCES

[1] P. Verkaik, O. Spatscheck, J. van der Merwe, and A. Snoeren, "Primed: A community-of-interest-based DDoS mitigation system," in *SIGCOMM Workshop on Large Scale Attack Defense (LSAD)*, September 2006.
[2] N. Duffield, C. Lund, and M. Thorup, "Properties and prediction of flow statistics from sampled packet streams," in *Proc. ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, Nov. 2002, pp. 159–171.
[3] W. Aiello, C. Kalmanek, P. McDaniel, S. Sen, O. Spatscheck, and J. Van der Merwe, "Analysis of Communities of Interest in Data Networks," in *Proc. PAM Workshop*, Mar. 2005.
[4] P. McDaniel, S. Sen, O. Spatscheck, J. Van der Merwe, W. Aiello, and C. Kalmanek, "Enterprise security: A community of interest based approach," in *Proc. NDSS*, Feb. 2006.
[5] P. Marques, N. Sheth, R. Raszuk, B. Greene, J. Mauch, and D. McPerson, "Dissemination of flow specification rules," IETF Internet-Draft, August 2005.
[6] S.Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," IETF RFC 2475, December 1998.
[7] M. A. El-Gendy, A. Bose, and K. G. Shin, "Evolution of the Internet QoS and support for soft real-time applications," *Proceedings of the IEEE*, vol. 91, no. 7, pp. 1086–1104, Jul. 2003.
[8] X. Xipeng and L. M. Ni, "Internet QoS: A big picture," *IEEE Network*, vol. 13, no. 2, pp. 8–18, Mar./Apr. 1999.
[9] C. Systems, "Cisco ios quality of service solutions configuration guide," www.cisco.com.
[10] J. Networks, "Policy framework configuration guide," www.juniper.net.
[11] C. Cortes, D. Pregibon, and C. T. Volinsky, "Communities of interest," *Intelligent Data Analysis*, vol. 6, no. 3, pp. 211–219, 2002.
[12] A. Karasaridis, B. Rexroad, and D. Hoeflin, "Wide-scale botnet detection and characterization," in *Conference on Hot Topics in Understanding Botnets (HotBots)*, April 2007.
[13] J. Ioannidis and S. M. Bellovin, "Implementing Pushback: Router-Based Defense Against DDoS Attacks," in *Proc. Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Feb. 2002.
[14] A. Yaar, A. Perrig, and D. Song, "Pi: A path identification mechanism to defend against DDoS attacks," in *Proc. IEEE Symposium on Security and Privacy*, Oakland, CA, 2003.
[15] ——, "SIFF: A stateless Internet flow filter to mitigate DDoS flooding attacks," in *Proc. IEEE Symposium on Security and Privacy*, Oakland, CA, May 2004.
[16] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attacks and defense mechanisms," *ACM Computer Communications Review*, vol. 34, no. 2, pp. 39–54, 2004.
[17] S. Kandula, D. Katabi, M. Jacob, and A. Berger, "Botz-4-Sale: Surviving Organized DDoS Attacks That Mimic Flash Crowds," in *Proc. 2nd USENIX NSDI*, Boston, MA, May 2005.
[18] M. Walfish, H. Balakrishnan, D. Karger, and S. Shenker, "DDoS defense by offense," in *Proc. ACM SIGCOMM*, Pisa, Italy, Aug. 2006.
[19] H. Ballani, Y. Chawathe, S. Ratnasamy, T. Roscoe, and S. Shenker, "Off by default!" in *Proc. ACM HotNets Workshop*, Nov. 2005.
[20] J. Chou, B. Lin, S. Sen, and O. Spatscheck, "Minimizing collateral damage by proactive surge protection," in *SIGCOMM Workshop on Large Scale Attack Defense (LSAD)*, August 2007.
[21] S. Anderson and D. Hogrefe, "Mouse trapping: A flow data reduction method," in *Proc. International Conference on Internet Monitoring and Protection*, 2008.
[22] T. M. Gil and M. Poletto, "MULTOPS: a data-structure for bandwidth attack detection," in *USENIX Security Symposium*, 2001.
[23] Z. M. Mao, V. Sekar, O. Spatscheck, J. van der Merwe, and R. Vasudevan, "Analyzing large ddos attacks using multiple data sources," in *SIGCOMM Workshop on Large Scale Attack Defense (LSAD)*, September 2006.
[24] C. Cranor, T. Johnson, O. Spatscheck, and V. Shkapenyuk, "Gigascope: a stream database for network applications," in *ACM SIGMOD*, 2003.