Googling the Internet: Profiling Internet Endpoints via the World Wide Web

Ionut Trestian, *Student Member, IEEE,* Supranamaya Ranjan, Aleksandar Kuzmanovic and Antonio Nucci

***Abstract*—Understanding Internet access trends at a global scale, *i.e.*, how people use the Internet, is a challenging problem that is typically addressed by analyzing network traces. However, obtaining such traces presents its own set of challenges owing to either privacy concerns or to other operational difficulties. The key hypothesis of our work here is that most of the information needed to profile the Internet endpoints is already available around us — on the web.**

In this paper, we introduce a novel approach for profiling and classifying endpoints. We implement and deploy a `Google`-based profiling tool, that accurately characterizes endpoint behavior by collecting and strategically combining information freely available on the web. Our Web-based 'unconstrained endpoint profiling' (UEP) approach shows advances in the following scenarios: (i) Even when no packet traces are available, it can accurately infer application and protocol usage trends at arbitrary networks; (ii) When network traces are available, it outperforms state-of-the-art classification tools such as BLINC; (iii) When sampled flow-level traces are available, it retains high classification capabilities. We explore other complementary UEP approaches, such as p2p- and reverse-DNS-lookup-based schemes, and show that they can further improve the results of the Web-based UEP. Using this approach, we perform unconstrained endpoint profiling at a global scale: for clients in four different world regions (Asia, South and North America and Europe). We provide the first-of-its-kind endpoint analysis that reveals fascinating similarities and differences among these regions.

I. Introduction

Understanding what people are doing on the Internet at a global scale, *e.g.*, which applications and protocols they use, which sites they access, and who they try to talk to, is an intriguing and important question for a number of reasons. Answering this question can help reveal fascinating cultural differences among nations and world regions. It can shed more light on important social tendencies (*e.g.*, [30]) and help address imminent security vulnerabilities (*e.g.*, [29], [35]). Moreover, understanding *shifts* in clients' interests, *e.g.*, detecting when a new application or service becomes popular, can dramatically impact traffic engineering requirements as well as marketing and IT-business arenas. `YouTube` [18] is probably the best example: it was unexpected and it currently accounts for more than 10% of the total Internet traffic [21].

The most common way to answer the above questions is to analyze operational network traces. Unfortunately, such an approach faces a number of challenges. First, obtaining 'raw' packet traces from operational networks can be very hard, primarily due to privacy concerns. As a result, researchers are typically limited to traces collected at their own institutions' access networks (*e.g.*, [23], [24]). While certainly useful, such traces can have a strong 'locality' bias and thus cannot be used to accurately reveal the diversity of applications and behaviors at a global Internet scale. Moreover, sharing such traces among different institutions is again infeasible due to privacy concerns.

Even when there are no obstacles in obtaining non-access, *i.e.*, core-level traces, problems still remain. In particular, accurately classifying traffic in an online fashion at high speeds is an inherently hard problem. Likewise, gathering large amounts of data for off-line post-processing is an additional challenge. Typically, it is feasible to collect only flow-level, or *sampled* flow-level information. Unfortunately, some of the state-of-the-art packet-level traffic classification tools (*e.g.*, [23]) do not perform as well in such scenarios, as we demonstrate below.

In this paper, we propose a fundamental change in approaching the 'endpoint profiling problem': depart from strictly relying on (and extracting information from) network traces, and look for answers elsewhere. Indeed, our key hypothesis is that the large and representative amount of information about endpoint behavior is available in different forms all around us.

For communication to progress in the Internet, in the vast majority of scenarios, information about servers, *i.e.*, which IP address one must contact in order to proceed is publicly available (not necessarily on Google). In p2p-based communication, in which all endpoints can act both as clients and servers, this means that association between some of the endpoints and such an application becomes publicly visible. Even in classical client-server communication scenarios, information about *clients* does stay publicly available for a number of reasons (*e.g.*, at website user access logs, forums, proxy logs, *etc*.). Given that many other forms of communication and various endpoint behavior (*e.g.*, game abuses) does get captured and archived, this implies that enormous information, invaluable for characterizing endpoint behavior at a global scale, is publicly available.

The first contribution of this paper is the introduction of a novel methodology, that we term 'unconstrained endpoint profiling.' The methodology uses publicly-available information about endpoints, e.g., available on the web or accessible by crawling p2p systems, to profile endpoints. The approach is "unconstrained" in the sense that it looks for and uses external information, beyond that available in network traces, to predict application trends or complement the existing traffic classification schemes. Hence our approach is different by design (not necessarily better) from other traffic classification

A subset of this work appears in the Proceedings of ACM Sigcomm '08 [34].

I. Trestian and A. Kuzmanovic are with the EECS Department, Northwestern University, Evanston, IL 60208 USA (e-mail: ionut@northwestern.edu; akuzma@northwestern.edu).

S. Ranjan and A. Nucci are with Narus Inc., Mountain View, CA, 94043 USA (e-mail: soups@narus.com; anucci@narus.com).

approaches (*e.g.*, BLINC). We compare these approaches for given networks later in the paper. In this paper, we focus on Web-oriented UEP approach that aims to characterize endpoint behavior by strategically combining information from a number of different sources available on the web. The key idea is to query the `Google` search engine [5] with IP addresses corresponding to arbitrary endpoints. In particular, we search on text strings corresponding to the standard dotted decimal representation of IP addresses, and then characterize endpoints by extracting information from the responses returned by `Google`. The core components of our methodology are (i) a *rule generator* that operates on top of the `Google` search engine and (ii) an *IP tagger*, that tags endpoints with appropriate features based solely on information collected on the web. The key challenge lies in *automatically* and accurately distilling valuable information from the web and creating a semantically-rich endpoint database.

We demonstrate that the proposed methodology shows advances in the following scenarios: (i) even when *no* operational traces from a given network are available, it can accurately predict traffic mixes, *i.e.*, relative presence of various applications in given networks, (ii) when packet-level traces are available, it can help outperform state of the art traffic classification algorithms such as BLINC , *e.g.*, [23], both quantitatively and qualitatively and, (iii) when sampled flow-level traces are available, it retains high classification capabilities when other state-of-the-art schemes do not perform as well. It should be noted that the examined networks belong to Tier-1 ISPs which is an unfriendly environment for one of the compared approaches [25]. Still not all information is available on the Web. Hence, results may be improved by using additional sources of information, some of which come at a high cost (*e.g.*, joining and crawling a p2p network). We explore other complementary UEP approaches, such as p2p- and reverse-DNS-lookup-based schemes, and show that they can further improve the results of the Web-based UEP.

Our second contribution lies in exploiting our methodology to perform, to the best of our knowledge, the first-of-its-kind Internet access trend analysis for four world regions: Asia, S. and N. America and Europe. Not only do we confirm some common wisdom, *e.g.*, `Google` massively used all around the world, `Linux` operating system widely deployed in France and Brazil, or multiplayer online gaming highly popular in Asia; we confirm fascinating similarities and differences among these regions. For example, we group endpoints into different classes based on their application usage. We find that in all explored regions, the online gaming users strongly protrude as a separate group without much overlap with others. At the same time, we explore locality properties, *i.e.*, where do clients fetch content from. We find strong locality bias for Asia (China), but also for N. America (US), yet much more international behavior by clients in S. America (Brazil) and Europe (France).

This paper is structured as follows. In Section II we explain our Web-based unconstrained endpoint profiling methodology that we evaluate in a number of different scenarios in Section III, and apply this approach to four different world regions in Section IV. In Section V, we compare the Web-based unconstrained endpoint profiling approach with two complementary UEP profiling schemes: p2p crawling and reverse DNS lookups. We discuss related issues and provide an overview of related work in Section VI. Finally Section VII concludes the paper.

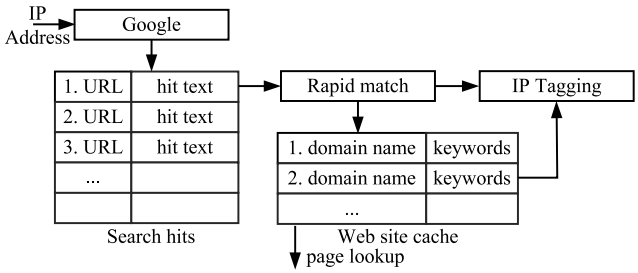

Fig. 1. Web-based endpoint profiling tool. Generates IP address tags based on information found via Google.

II. Methodology

Here, we propose a new methodology, that we term Web-based 'Unconstrained Endpoint Profiling' (UEP). Our goal is to characterize endpoints by strategically combining information available at a number of different sources on the web. Our key hypothesis is that records about many Internet endpoints' activities inevitably stay publicly archived. Of course, not all active endpoints appear on the web, and not all communication leaves a public trace. Still, we show that enormous amounts of information does stay publicly available, and that a 'purified' version of it could be used in a number of contexts that we explore later in the paper.

A. Unconstrained Endpoint Profiling

Figure 1 depicts our web-based endpoint profiling tool. At the functional level, the goal is straightforward: we query the `Google` search engine by searching on text strings corresponding to the standard dotted decimal representation of IP addresses. For a given input in the form of an IP address, *e.g.*, 200.101.18.182, we collect search hits returned by `Google`, and then extract information about the corresponding endpoint. The output is a set of *tags* (features) associated with this IP address. For example, `forum user`, `kazaa node`, `game abuser`, `mail server`, *etc*. In general, an endpoint could be tagged by a number of features, *e.g.*, a `forum user` and a `p2p client`. Such information can come from a number of different URLs.

At a high level, our approach is based on searching for information related to IP addresses on the web. The larger the number of search hits returned for a queried IP address, and the larger number of them confirming a given behavior (*i.e.*, a `streaming server`), the larger the confidence about the given endpoint activity. The profiling methodology involves the following three modules: (i) Rule generation, (ii) Web classification and (iii) IP tagging, that we present in detail below.

1) Rule Generation: The process starts by querying `Google` [5] using a sample 'seed set' of random IP addresses from the networks in four different world regions (details in Section III) and then obtaining the set of search hits. Each search hit consists of a URL and corresponding *hit text*, *i.e.*, the text surrounding the word searched. We then extract all the

words and biwords (word pairs) from the hit texts of all the hits returned for this seed set. After ranking all the words and biwords by the number of hits they occur in and after filtering the trivial keywords (*e.g.*, 'the'), we constrain ourselves to the top N keywords[1] that could be meaningfully used for endpoint classification. By meaningfully used we mean that the keyword found implies an application or application class associated with network activity.

Then, in the only manual step in our methodology, we construct a set of rules that map keywords to an interpretation for the functioning of that website, *i.e.*, the *website class*. The rules are as shown in the relationship between Column 1 and 2 in Table I. For example, the rules we develop in this step capture the intelligence that presence of one of the following keywords: `counter strike`, `world of warcraft`, `age of empires`, `quake`, or `game abuse` in either the URL or the text of a website implies that it is a gaming website (either gaming server list or abuse list). Table I shows a few rules to differentiate the information contained in websites. For instance, if a website only contains the keyword `mail server` from the set of keywords, then it is classified as a site containing list of mail servers. However, if a website contains one of the following words, `spam` or `dictionary attacker` besides `mail server`, then it is classified as one containing list of *malicious* mail servers, *e.g.*, one that is known to originate spam. Similar rules are used to differentiate between websites providing gaming servers list and gaming abuse list.

2) Web Classifier: Extracting information about endpoints from the web is a non-trivial problem. Our approach is to first characterize a given webpage (returned by `Google`), *i.e.*, determine what information is contained on a website. This approach significantly simplifies the endpoint tagging procedure.

Rapid URL Search. Some websites can be quickly classified by the keywords present in their domain name itself. Hence, after obtaining a search hit we first scan the URL string to identify the presence of one of the keywords from our keyword set in the URL and then determine the website's class on the basis of the rules in Table I. For instance, if the URL matches the rule: {`forum` | ... | `cafe`} (see last row in Table I) then we classify the URL as a Forum site. Typically, websites that get classified by this rapid URL search belong to the Forum and Web log classes. If the Rapid URL search succeeds, we proceed to the IP tagging phase (Section II-A3). If rapid match fails, we initiate a more thorough search in the hit text, as we explain next.

Hit Text Search. To facilitate efficient webpage characterization and endpoint tagging, we build a website cache. The key idea is to speed-up the classification of endpoints coming from the same web sites/domains under the assumption that URLs from the same domain contain similar content. In particular, we implement the website cache as a hashtable indexed by the domain part of the URL. For example, if we have a hit coming from the following URL: www.robtex.com/dns/32.net.ru.html, the key in the hashtable becomes robtex.com. Hence, all IPs that return a search hit from this domain can be classified in the same way.

Whenever we find a URL whose corresponding domain name is not present in the cache, we update the cache as follows. First, we insert the domain name for the URL as an index into the cache with an empty list (no keywords) for the value. In addition, we insert a counter for number of queried IP addresses that return this URL as a hit along with the corresponding IP address. High values for the counter would indicate that this domain contains information useful for classifying endpoints. Thus, when the counter for number of IP addresses goes over a threshold we retrieve the webpage based on the last URL. We currently use a threshold of 2. We have chosen this threshold in order to filter out websites that carry information about a single IP address only. At the same time, this approach maximizes the amount of traffic that we can classify while filtering out the above sites. Then, we search the webpage for the keywords from the keyword set and extract the ones that can be found.

Next, we use the rule-based approach to determine the class to which this website (and hence the domain) belongs. Finally, we insert an entry in the cache with the domain name as the key and the list of all associated keywords (from Table I) as the value. For instance, if the URL matches the rule: `mail server` & {`spam` | `dictionary attacker`}, then the domain gets classified as a list of malicious mail servers. Further, we insert all the keywords in the cache. When a URL's domain name is found in the cache, then we can quickly classify that URL by using the list of keywords present in the cache. In this way, the cache avoids having to classify the URL on every hit and simplifies the IP-tagging phase, as we explain next.

3) IP tagging: The final step is to tag an IP address based on the collected information. We distinguish between three different scenarios.

URL based tagging. In some scenarios, an IP address can be directly tagged when the URL can be classified via rapid search for keywords in the URL itself. One example is classifying eMule p2p servers based on the emule-project.net domain name. Another example is the torrent list found at torrentportal.com. In such scenarios, we can quickly generate the appropriate tags by examining the URL itself. In particular, we use the mapping between a website class (Column 2) and IP tags (Column 3) in Table I to generate the tags. In majority of the cases, such rapid tagging is not possible and hence we have to examine the hit text for additional information.

General hit text based tagging. For most of the websites, we are able to accurately tag endpoints using a keyword based approach. The procedure is as follows. If we get a match in the website cache (for the specific URL we are currently trying to match), we check if any of the keywords associated with that domain match in the search hit text. Surprisingly, we typically find at least a *single* keyword, that clearly reveals the given IP's nature and enables tagging. Table I provides the mapping between the domain class and IP tags.

For hit texts that match multiple keywords, we explain the generation of tags via an example. For instance, a URL such as projecthoneypot.org provides multiple information about an IP address, *e.g.*, not only that it is a mail server but also a spammer. Due to a match with both the keywords, this

[1]We find and use the top 60 keywords in this paper.

TABLE I
KEYWORDS - WEBSITE CLASS - TAGS MAPPING

Keywords	Website Class	Tags
{'ftp' | 'webmail' | 'dns' | 'email' | 'proxy' | 'smtp' | 'mysql' | 'pop3' | 'mms' | 'netbios'}	Protocols and Services	<protocol name> server
{'trojan' | 'worm' | 'malware' | 'spyware' | 'bot'}	Malicious information list	<issue name> affected host
'spam'	Spamlist	spammer
{'blacklist' | 'banlist' | 'ban' | 'blocklist'}	Blacklist	blacklisted
'adserver'	Ad-server list	adserver
{'domain' | 'whois' | 'website'}	Domain database	website
{'dns' | 'server' | 'ns'}	DNS list	DNS server
{'proxy' | 'anonymous' | 'transparent'}	Proxy list	proxy server
'router'	Router addresses list	router
'mail server'	Mail server list	mail server
'mail server' & {'spam' | 'dictionary attacker'}	Malicious mail servers list	mail server [spammer] [dictionary attacker]
{'counter strike' | 'warcraft' | 'age of the empires' | 'quake' | 'halo' | 'game'}	Gaming servers list	<game name> server
{'counter strike' | 'warcraft' | 'age of the empires' | 'quake' 'halo' | 'game'} & {'abuse' | 'block'}	Gaming abuse list	<game> node [abuser] [blocked]
{'torrent' | 'emule' | 'kazaa' | 'edonkey' | 'announce' | 'tracker' | 'xunlei' | 'limewire' | 'bitcomet' | 'uusee' | 'qqlive' | 'pplive' }	p2p node list	<protocol name> p2p node
{'irc' | 'undernet' | 'innernet' | 'dal.net'}	IRC servers list	IRC server
{'yahoo' | 'gtalk' | 'msn' | 'qq' | 'icq' | 'server' | 'block'}	Chat servers	<protocol name> chat server
{'generated by' | 'awstats' | 'wwwstat' | 'counter' | 'stats'}	Web log site	web user [operating system] [browser][date]
{'cachemgr' | 'ipcache'}	Proxy log	proxy user [site accessed]
{'forum' | 'answer' | 'resposta' | 'reponse' | 'comment' | 'comentario' | 'commentaire' | 'posted' | 'poste' | 'registered'| 'registrado' | 'enregistre' | 'created' | 'criado' 'cree' | 'bbs' | 'board' | 'club' | 'guestbook' | 'cafe' }	Forum	forum user [date][user name] [http share][ftp_share] [streaming node]

URL's domain would be entered in the website cache as a malicious mail servers' list. Then queries to an ip-address that is listed at projecthoneypot.org could return either: (i) both the keywords `mail server` and `spam`, in that case, the ip-address would be tagged by both the tags `mail server` and `spammer`, (ii) only the keyword `mail server` where the ip-address would be tagged as a `mail server` only and (iii) only the keyword `spam` where the ip-address would be tagged as `spammer` via the one-to-one mapping but also as `mail server`. This expansion of tags (from spam to mail server) can be done unambiguously because there is no rule in Table I with only one keyword `spam`. Similarly, regardless of the combination of keywords found in the hit text for gaming servers list or gaming abuse list, their rules can be disambiguated as well.

In some cases, such as for Web logs and Proxy logs, we can obtain additional tags (labeled by square brackets in Column 3 of Table I). For Web logs we can obtain the access date and, if the data exists, the operating system and browser that was used. Similarly, in the case of Proxy logs, we can obtain the site that was accessed by the IP address.

Hit text based tagging for Forums. The keyword-based approach fails when a URL maps to an Internet forum site. This is because a number of non-correlated keywords may appear at a forum page. Likewise, forums are specific because an IP address can appear at such a site for different reasons. Either it has been automatically recorded by a forum post, or because a forum user deliberately posted a link (containing the given IP address) for various reasons.

In the case of forums, we proceed as follows. First, we use a post-date and username in the vicinity of the IP address to determine if the IP address was logged automatically by a forum post. Hence, we tag it as the `forum user`. If this is not the case, the presence of the following keywords: http:\, ftp:\, ppstream:\, mms:\, *etc.* in front of the IP address string in the hit text suggests that the user deliberately posted a link to a shared resource on the forum. Consequently, we tag the IP address as an `http share` or `ftp share`, or as a `streaming node` supporting a given protocol (ppstream, mms, tvants, sop, *etc.*).

Because each IP address generates several search hits, multiple tags can be generated for an IP address. Thus aggregating all the tags corresponding to an IP address either reveals additional behavior or reaffirms the same behavior. For the first case, consider the scenario where an IP address hosts multiple services, that would then be identified and classified differently and thereby generate different tags for that IP address, revealing the multiple facets of the IP address' behavior. In the second case, if an IP address' behavior has been identified by multiple sites, then counting the unique sites that reaffirm that behavior would generate higher confidence. In this paper, we consider this confidence threshold as 1, *i*.e., even if one URL hit proclaims a particular behavior then we classify the endpoint accordingly. We make this choice in order to maximize the amount of traffic classified.

B. Where Does the Information Come From?

Here, we attempt to answer two questions. First, which sites 'leak' information about endpoints? While we have already hinted at some of the answers, we provide more comprehensive statistics next. Second, our goal is to understand if and

TABLE II
WEBSITE CACHES - TOP ENTRIES

N. America				Asia				S. America			
Nr	Site	Hits	Info	Nr	Site	Hits	Info	Nr	Site	Hits	Info
1	whois.domaintools.com	338	D	1	jw.dhu.edu.cn	1381	S	1	weblinux.ciasc.gov.br	395	S
2	en.wikipedia.org	263	F	2	projecthoneypot.org	377	M	2	projecthoneypot.org	371	M
3	robtex.com	255	BDN	3	info.edu.sh.cn	268	S	3	robtex.com	252	BDN
4	projecthoneypot.org	217	M	4	czstudy.gov.cn	227	S	4	redes.unb.br	252	S
5	extremetracking.com	202	S	5	qqdj.gov.cn	181	S	5	pt.wikipedia.org	200	F
6	botsvsbrowsers.com	182	W	6	zhidao.baidu.com	176	F	6	appiant.net	136	S
7	cuwhois.com	151	D	7	1bl.org	154	B	7	www.tracemagic.net	116	S
8	proxy.ncu.edu.tw	132	P	8	cqlp.gov.cn	149	S	8	www.luziania.com.br	91	F
9	comp.nus.edu.sg	116	S	9	cache.vagaa.com	142	T	9	pgl.yoyo.org	90	A
10	quia.jp	108	M	10	bid.sei.gov.cn	122	S	10	netflow3.nhlue.edu.tw	76	S
Cache size: 827				Cache size: 892				Cache size: 728			
A:adservers, B:blacklist, D:domaindb, F:forum, M:mail/spam, N:dnsdb , P:proxy cache, S:Web logs, T:torrent, W:bot detector											

how such 'information-leaking' sites vary in different world regions.

Sites containing information about endpoints could be categorized in the following groups:

• *Web logs:* Many web servers run web log analyzer programs such as AWStats, Webalizer and SurfStats. Such programs collect information about client IP addresses, statistics about access dates, host operating systems and host browsers. They parse the web server log file and generate a report or a statistics webpage.

• *Proxy logs:* Popular proxy services also generate logs of IP addresses that have accessed them. For instance, the Squid proxy server logs the requests' IP addresses, and then displays them on a webpage.

• *Forums:* As explained above, Internet forums provide wealth of information about endpoints. Some forums list the user IP addresses along with the user names and the posting dates in order to protect against forum spam. Examples are inforum.insite.com.br or www.reptilesworld.com/bbs. Likewise, very frequently clients use Internet forums to post links containing (often illegal) CDs or DVDs with popular movies as either ftp, http, or streaming shares. We explained above how our methodology captures such cases. Some IP logging forums also provide information about clients participating in chat applications. These forums also ask for (Yahoo, MSN etc.) messenger ID's upon registration in order to display the online status of the forum user. When searching the IP address on Google one also finds this related information.

• *Malicious lists:* Denial of service attacks and client misbehavior in general, are a big problem in today's Internet. One of the ways to combat the problem is to track and publicize malicious endpoint behavior. Example lists are: banlists, spamlists, badlists, gaming abuse lists, adserver lists, spyware lists, malware lists, forum spammers lists, *etc.*

• *Server lists:* For communication to progress in the Internet, information about servers, *i.e.*, which IP address one must contact in order to proceed, must be publicly available. Examples are domain name servers, domain databases, gaming servers, mail servers, IRC servers, router (POP) lists, *etc.*

• *P2P communication:* In p2p communication, an endpoint can act both as a client and as a server. Consequently, an IP's involvement in p2p applications such as eMule, gnutella, edonkey, kazaa, torrents, p2p streaming software, *etc.*, becomes visible when contacting the first point of entry into the system. For this, this first point of entry is known and typically available on the web. The number of such endpoints in a p2p network is relatively small; however, whenever a client wants to retrieve a file he typically goes through such an access point. Example websites are emule-project.net, edonkey2000.cn, or cache.vagaa.com, that lists torrent nodes. Gnutella is a special case since `Google` can directly identify and list gnutella nodes using their IP addresses. Given that our system is `Google`-based, it inherits this desirable capability.

All the above examples confirm that publicly available information about endpoints is indeed enormous in terms of size and semantics. The key property of our system is its ability to automatically extract all this information in a unified and methodical way. Moreover, because we operate on top of `Google`, any new source of information becomes quickly revealed and exploited.

Table II answers the second question: how different are the endpoint information sites in different world regions? In particular, Table II shows top entries for three different world regions we explored (details provided in the next section).[2] While some sites, *e.g.*, projecthoneypot.org or robtex.com, show global presence, other top websites are completely divergent in different world regions. This reveals a strong locality bias, a feature we explore in more depth in Section IV below.

III. EVALUATION

Next, we demonstrate the diversity of scenarios in which unconstrained endpoint profiling can be applied. In particular, we show how it can be used to (i) discover active IP ranges *without* actively probing the same, (ii) classify traffic at a given network and predict application- and protocol trends in *absence* of any operational traces from a given network, (iii) perform a semantically-rich traffic classification when packet-level traces are available, and (iv) retain high classification capabilities even when only sampled flow-level data is available.

Table III shows the networks we study in this paper. They belong to Tier-1 ISPs representative of one of the largest countries in different geographic regions: Asia (China), South

[2]We omit details for the fourth region - Europe - due to space constraints.

TABLE III
STUDIED NETWORKS

Asia	S. America	N. America
XXX.39.0.0/17	XXX.96.128.0/17	XXX.160.0.0/12
XXX.172.0.0/18	XXX.101.0.0/17	XXX.160.0.0/13
XXX.78.192.0/18	XXX.103.0.0/17	XXX.168.0.0/14
XXX.83.128.0/17	XXX.140.128.0/18	XXX.70.0.0/16
XXX.239.128.0/18	XXX.163.0.0/17	XXX.0.0.0/11
XXX.69.128.0/17	XXX.193.192.0/18	
XXX.72.0.0/17	XXX.10.128.0/18	**Europe**
	XXX.14.64.0/18	62.147.0.0/16
	XXX.15.64.0/18	81.56.0.0/15
	XXX.24.0.0/18	82.64.0.0/14
	XXX.25.64.0/18	
	XXX.34.0.0/18	

America (Brazil), North America (US) and Europe (France). The Asian and S. American ISPs serve IPs in the /17 and /18 range, while the N. American and European ISPs serve larger network ranges.

In most scenarios (Asia, S. and N. America), we manage to obtain either packet-level (Asia and S. America) or flow-level (N. America) traces from the given ISPs. The packet-level traces are couple of hours in duration while the flow-level trace is almost a week long. These traces are invaluable for the following two reasons. First, they present the necessary 'ground truth' that helps us evaluate how well does our approach (without using *any* operational traces) work to discover active IP ranges (Section III-A) and predict application and protocol trends (Section III-B). Second, we use these traces to understand how our approach can be applied in the classical traffic classification scenarios, both using packet-level (Section III-C) and flow-level (Section III-D) traces.

To preserve privacy of the collaborating ISPs, in Table III, we anonymize the appropriate IP ranges by removing the first Byte from the address. We do not anonymize the IP range for the European ISP (`Proxad`, http://www.free.fr/, AS 12322), simply because we use no operational network trace. In this case, we stick with the endpoint approach, and thus only use publicly available information.

A. Revealing Active Endpoints

First, we explore if the Google hits can be used to infer the active IP ranges of the target access networks. This knowledge is invaluable in a number of scenarios. For example, for Internet-scale measurement projects (*e.g.*, [27]) knowing which IPs are active in a given ISP can help direct measurements towards the active parts of the address space. The approach is particularly useful given that large-scale active probing and network scanning might trigger a ban from either the host or the targeted ISP. Indeed, our indirect approach efficiently solves this problem since we get the targeted active IP subset by simply googling the IP addresses.

To demonstrate the potentials of this approach, we show results for the XXX.163.0.0/17 network range, that spans 32,767 IP addresses. As one source of information about active IPs, we google this IP range. As another source, we determine the active IP addresses from a packet-level trace we obtained from the corresponding ISP. Necessarily, a relatively short trace does not contain all active IPs from this network range. The results are as follows. We determine 3,659 active IPs using Google (IP addresses that generate hits on Google). At the same time, we determine 2,120 IPs from the trace, i.e., the IP addresses belonging to that specific range that are observed sending traffic. The overlap is 593 addresses, or 28% (593/2120).

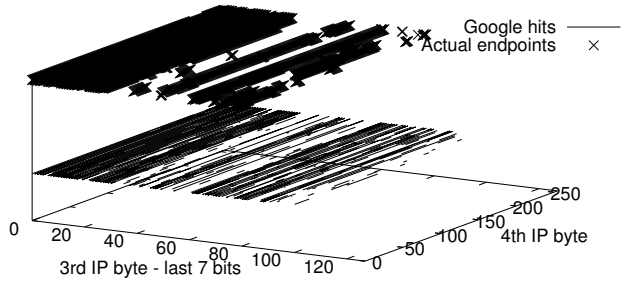

Fig. 2. Inferring endpoints by Google searches - XXX.163.0.0/17. Comparison of results obtained via Google to addresses that generate traffic.

By carefully examining the two results, we find that spatial correlation is high, *i.e.*, in each trace the active IPs are very close in IP space. Indeed, IP address assignment on the Internet at different points in the assignment hierarchy except the end user is done on a block basis. Also, to ease network management, network administrators typically assign contiguous IP addresses to hosts in the same network. To exploit this feature, we proceed as follows. For each of the active IP addresses (Google- and trace-based), we apply an IP mask to enrich the set of IP addresses.[3] We apply the mask to an IP, and then consider all IPs corresponding to the resulting prefix as being active.

Figure 2 shows the results for both Google- and trace-based active hosts obtained in this way. Indeed, the figure shows high spatial correlation between the two sets. In particular, enhanced Google-based trace now has 12,375 IPs, while enhanced network trace has 10,627 IPs. The number of overlapped addresses is as high as 8,137, such that the overlap between the two sets now becomes 77% (8,137/10,627). We apply the same approach to the week-long N. American trace and our results show 79%-84% overlap for the five different network ranges.

We stress once again that the key point of this approach is *not* to accurately infer if a given IP address is active or not, but rather to *hint* at the highly probable active IP ranges and ease methodologies that require such information (*e.g.*, [27]). One other observation is that the active IP coverage obtained with this approach increases as the studied network range increases. This is because the distance between active IP clusters increases with the size of the studied network. Consequently, we note that this approach becomes even more useful in the context of IPv6. This is because network ranges will become larger; hence, randomly probing a certain network space might immediately trigger a ban.

[3]Numerous experiments on other network ranges corroborate that a mask of /28 (i.e., zeroing out last 4 bits) shows the best compromise between maximizing the overlap between Google- and trace-based active IPs and minimizing the size of enriched subsets.

B. When No Traces are Available

We apply the unconstrained endpoint approach on a *subset* of the IP range belonging to the four ISPs shown in Table III. In particular, we explore a 'seed set' consisting of approximately 200,000 randomly chosen IP addresses from each of the four world regions. We obtain the comprehensive results including statistics about operating systems, browsers, malicious activity, p2p, protocols and services, chat, gaming and most popular sites. We omit detailed results here due to space constraints (see reference [34] for detailed results.) We emphasize that the information we discuss below is obtained solely using the Google-based approach, without exploiting *any* information from the operational network traces, nor any other sources.

The key question we aim to answer here is how representative are these results. In particular, can they be used to predict the popularity of a given application in a given world region? Or, is there any correlation between these results and operational network traces collected at given networks? We answer these questions by comparing results obtained via the unconstrained endpoint profiling approach with the 'ground truth,' in the form of (i) traces from operational networks and (ii) other publicly available information such as from news articles about endpoint behavior. More detailed results are available in reference [34].

Correlation with operational traces. We select the S. American trace to exemplify correlation between the results obtained using `Google` and the network traces analyzed using packet analyzers and a signature based traffic classification tool [12]. Other network traces (Asia and N. America) show results consistent with this example, as we explain below. In particular, we compare the following traffic categories: p2p, chat, gaming and browsing. Other characteristics, such as OS type, browser type, spam, *etc*., are either hard or impossible to extract from network-level traces.

We find a high correlation between the two sources. Specifically, in three of the four traffic categories, we find that the leading applications obtained using the UEP approach is also the leading application in the trace. In particular, `Gnutella` is the leading p2p system, `msn` is the leading chat software and `Google` is the leading website in the trace. Similarly, for all other scenarios where our system detects a strong application presence (*e.g.*, `ppstream` and `Tencent QQ` software in China), that behavior is inevitably reflected in traces as well.

Necessarily, not always does the information from network traces and the `Google`-based approach stay in the same order. In particular, when UEP discovers m IP addresses associated with *Application 1* and n IP addresses associated with *Application 2*, it does not strictly follow that *Application 1* is more popular than *Application 2* if $m > n$. For example, results for gaming applications found in the traces are often not in the same order as classified by UEP. The same can happen for the relative order among other applications as well. For example, `Orkut` comes before `wikipedia` in the network trace, contrary to the results obtained via `Google`.

The reasons for this behavior are obvious. First, the information collected on Google is necessarily biased in the sense that it depends on several factors: the webpages crawled by Google, the addresses that do make it onto webpages, the fraction of those for which application information can be inferred, etc. Second, the results obtained via the UEP (`Google`) approach represent a spatial sample (over the IP space) averaged over time. On the other hand, results from the trace represent a sample taken in a short time interval, *i.e.*, a few hours in this particular case (South American ISP). Still, the key point here is that despite the necessary bias introduced by the UEP approach and differences in the nature of the data collected from two different sources, web and operational networks, there is still a high correlation. By high correlation we mean that when an application is strongly present in a given area this result shows up consistently in both network traces and on the web and this is shown by the top-most applications that we have analyzed.

Correlation with other sources. Here, we compare the results obtained via `Google` with other publicly available sources. One example is the presence of operating systems in different world regions. `Windows` is the leading operating system in all examined regions except France where the `Debian Linux` distribution is prevalent. This is not a surprise given that French administration and schools run Linux distributions [10]. A similar trend can be observed in Brazil, where `Windows` has only a small advantage over `Linux`. Again, this is because similar measures to the ones in France have been implemented in Brazil as well [9]. A related issue is that of browsers. `Mozilla` is more popular in France and Brazil, as a natural result of the operating systems popularity.

Another example is p2p activity. Our results reveal some previously-reported locality tendencies, such as torrents and `eMule` being widely used in France [31], and p2p streaming software being very popular in China [4]. Likewise, our results confirm the well-known 'Googlemania' phenomenon. They also reveal that `wikipedia` is a very popular website all over the world. This is not the case for China, where the number of hits is low, potentially due to a ban [16] at some point. Similarly, `Orkut`, the social network built by `Google`, shows hits in Brazil, the region where it is very popular [1], [13].

Summary. High correlation between the data collected from the web and those from operational network traces and elsewhere imply that the unconstrained endpoint profiling approach can be effectively used to estimate application popularity trends in different parts of the world. We demonstrate that this is possible to achieve in a unified and methodical way for all different world regions, yet *without* using any operational network traces.

C. When Packet-Level Traces are Available

Traffic classification (based on operational network traces) is another case where the unconstrained endpoint approach can be applied. Indeed, the state-of-the-art traffic classification tools are constrained in several ways. To the best of our knowledge, all current approaches try to classify traffic by exclusively focusing on observed packets and connection patterns established by the endpoints. One example is BLINC [23], that uses a graphlet based approach to classify network traffic. Issues with such an approach are the following. First,

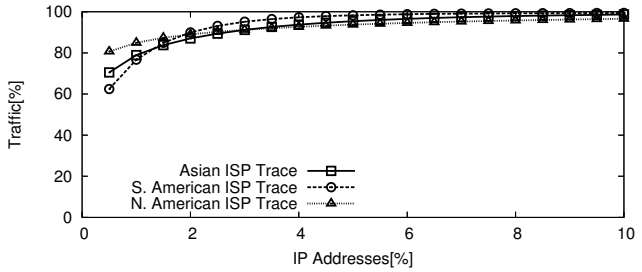

Fig. 3. The traffic amount that is directed to a certain percent of traffic destinations. Notice how 95% of the traffic is directed to 5% of the destinations in all cases.

BLINC is primarily an off-line tool that might be challenging to deploy in the network core. Second, classification semantics of such a system is not particularly rich at the application level. For example, it can classify a flow as p2p, but cannot say which particular protocol it is. Finally, it relies upon ad-hoc thresholds, that might produce variable quality results for different traces, as we show below. For the same reason, the approach does not perform as well when sampled traffic traces are available, as we demonstrate later. We compare UEP against a ground truth obtained via a signature-based traffic classification tool [12]. This tool identifies unique flows and then 'sessionalizes' all packets belonging to the same flow in to one string against which regular expressions [8] are applied to identify the corresponding protocol. Because this tool examines packet payloads to determine the application type, it is necessarily more computationally intensive than UEP.

Note that we regard our unconstrained approach as complementary to other traffic classification approaches, including BLINC. We view the web-crawling part of UEP as a first and inexpensive pass that gathers services information that are used to classify traffic. This can be followed by other more complex techniques that could benefit from the UEP-based information.

The UEP approach is online capable because of its ability to classify traffic based on a single observed packet for which one of the endpoints is revealed (*e.g.*, a web server). Furthermore, there is a huge bias of traffic destinations (*e.g.*, 95% of traffic is targeted to 5% of destinations [32]). The implication is that it is possible to accurately classify 95% of traffic by reverse-engineering 5% of endpoints, that can be cached in the network. Indeed, Figure 3 confirms strong endpoint bias for *all* traces: Asian, S. and N. American. In particular, 1% of endpoints account for more than 60% of the traffic, and 5% endpoints carry more than 95% of traffic in all cases.

We apply the endpoint approach to classify traffic for the Asian and S. American ISPs for which we have packet-level traces. In particular, we do this in two phases. First, we collect the most popular 5% of IP addresses from the traces and tag them by applying the methodology from Section II. Next, we use this information to classify the traffic flows into the classes shown in Column 3 of Table IV. The classification rule is simple – if one of the endpoints in a flow is tagged by a server tag, *e.g.*, as a `website`, then the flow is classified appropriately, *e.g.*, as *Browsing*. The detailed classification rules are as shown in the mapping between Column 2 and Column 3 in Table IV.

TABLE IV
DETERMINING TRAFFIC CLASSES AND USER BEHAVIOR

Client tag	**Server tag**	**Traffic class, User behavior**
web user, proxy user	website	Browsing
mail server	mail server	Mail
<game name> node [abuser] [blocked]	<game name> server	Gaming
n/a	<protocol name> chat server	Chat
n/a	IRC server	Chat
[streaming node]	[streaming node]	Streaming
<issue name> affected host	<issue name> affected host	Malware
p2p node	p2p node	P2P
[ftp share]	ftp server	Ftp

Table 5 shows the classification results relative to BLINC and the signature-based approach for the S. American trace. We get similar results for other traces. In all cases, we manage to classify over 60% of the traffic. At the same time, BLINC classifies about 52% of traffic in the Asian case and 31.35% in the S. American case (Figure 5 for x=1 and Table V). Also, in addition to outperforming BLINC quantitatively, the endpoint approach provides a much richer semantics quality. For example, we are able not only to classify traffic as chat, but accurately pinpoint the exact type, *e.g.*, `msn` *vs.* `yahoo` *vs.* `usenet`. Note also that our approach manages to obtain comparable results with the signature based traffic classification (Table V), as we discuss in detail below.

Since a flow is classified by the endpoint(s) that it involves, the correctness of our traffic classification is dependent on the correctness of our endpoint profiling. We next explore the issue of correctness by comparing the set of endpoints classified by our approach versus BLINC and the signature based traffic classification. Table VI shows the percentage breakdown per class (for S. America trace) in terms of endpoints found by both BLINC and our approach (B∩U), only by BLINC (B-U), only by our approach (U-B), both signature based traffic classification and our approach (S∩U), only by the signature based traffic classification (S-U) and only by our approach (U-S) . It is clear that our approach uncovers more endpoints and hence classifies more traffic than BLINC. Moreover, the number of endpoints that a constrained approach such as BLINC failed to classify is quite high (100% of streaming, mail and Ftp). Finally, it is also worth noting that the number of endpoints our approach failed to classify is fairly limited (7% of chat, 10% of browsing and 8% of p2p and 0% in others). Infact, as we will explain in detail in the next subsection, while analyzing sampled traffic, the gap between BLINC and our approach widens even further; the number of endpoints that only our approach classifies becomes higher than 91% for all classes. The reasons for the performances obtained by using BLINC can be found in [25]. The authors evaluate BLINC on several traces with mixed results and conclude that BLINC is not recommended for backbone links but mostly for border links of a single-homed edge network. Moreover, it is known that methodologies like BLINC need a sufficient traffic mix (e.g., traffic volume, IP addresses

TABLE V
TRAFFIC CLASSES FOR S. AMERICA

Class	Packet trace (% of total flows and number of total flows)		
	BLINC	**UEP**	**Sign. based**
Chat	0.398 5,642	3.38 47,975	2.32 32,903
Browsing	23.16 328,287	44.7 633,830	46.2 654,614
P2P	4.72 66,925	11.31 160,328	22.4 317,723
Gaming	0.14 1,996	0.15 2,123	0 0
Malware[4]	2.93 41,576	2.3 32,447	0 0
Streaming	0 0	0.18 2,536	0.34 4,869
Mail	0 0	1.58 22,365	2.15 30,426
Ftp	0 0	0.1 1,338	0.16 2,278
Classified	**31.35 444,426**	**63.7 902,942**	**73.57 1,042,813**
Unclassified	**68.65 973,089**	**36.3 514,573**	**26.43 374,702**
Total	100 1,417,515	100 1,417,515	100 1,417,515

TABLE VI
ENDPOINTS PER CLASS FOR S. AMERICA

Cls.[5]	Pkt. trace						
	Tot.	B∩U %	B-U %	U-B %	S∩U %	S-U %	U-S %
C	1769	16	7	77	69	0	31
Br	9950	31	10	59	92	6	2
P	8842	14	8	78	47	51	2
G	22	95	0	5	0	0	100
S	160	0	0	100	51	49	0
M	3086	0	0	100	77	23	0
F	197	0	0	100	64	36	0
Br browsing,C chat,M mail,P p2p,S streaming,G gaming,F ftp							
B BLINC, U Unconstrained Endpoint Profiling							
S Signature based traffic classification							

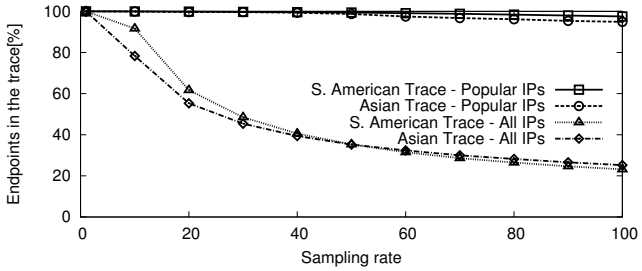

Fig. 4. Percent of IP addresses left in the trace by sampling with the specific amount relative to the IP addresses present in the non-sampled version of the trace.

present) in order to build the associated detection heuristics or statistics used by the various methodologies. Because our network traces are short-lived, they are necessarily unfriendly to BLINC-like approaches.

We further compare against the results obtained by the signature based traffic classifier. Our approach classifies more Chat flows. The reason is that chat applications also use port 443 (HTTPS) when evading firewalls and these are exactly the flows missing from the signature based approach. The results for Browsing are comparable as UEP manages to classify nearly as many flows as the signature based approach. For P2P, the signature based approach classifies twice as much (*e.g.*, 11.31% UEP to 22.4% signature based). As we will show later in Section V, UEP's performance can be improved by extending the framework to crawl P2P systems besides the web. The signature based approach does not have any Gaming or Malware signatures, so we can not compare in these two categories. For the last three categories (Streaming, Mail and FTP), UEP again obtains comparable results (*e.g.*, for mail 22,365 flows for UEP compared to 30,426 flows for singature based and for FTP 1,338 flows for UEP compared to 2,278 flows for signature based). Note, however, that signature based traffic classification is necessarily more expensive in terms of processing overhead and speed.

One last question remains to be answered: why was the endpoint approach unable to classify the remaining 38% of the traffic? By carefully examining the traces, we realize that the vast majority of unclassified traffic is p2p traffic, either file sharing or streaming. The key reason why these p2p 'heavy hitters' were not classified by the endpoint approach is because information about these IPs is not available on the web (or at least not found by `Google`). Still, these IPs are traceable (*e.g.*, [26]); indeed, we pursue such an approach in Section V below. Independently from p2p information, our results demonstrate that the information collected from the web is invaluable for the traffic classification application, even more so in sampled scenarios as we show below.

D. When Sampled Traces are Available

Packet-level traces are not always available from the network. Often only *sampled* flow-level traces are available, *e.g.*, collected using `Cisco's Netflow`. This is particularly the case for the network core, where collecting all packets traversing a high speed link is either infeasible or highly impractical. Sampled data analysis has received a great deal of attention in the research community. A well-known observation is that sampled data causes problems to anomaly detection algorithms (*e.g.*, [28]). Our observation is that it also creates problems to traffic classification tools, such as BLINC, as well. This happens because after the sampling procedure an insufficient amount of data needed by BLINC(IP addresses, traffic volumes) remains in the trace, and hence the graphlets approach simply does not work.

This is not the case for the endpoint approach. The key reason is that popular endpoints are still present in the trace, despite sampling. Thus, classification capabilities remain high. Figure 4 shows the percent of IPs (both all IPs and popular 5% ones) as a function of the sampling rate. In particular, we create sampled version of the Asian and S. American traces by randomly selecting packets with a given probability, the way `Netflow` would do it. For example, for sampling rate of 50, the probability to select a packet is 1/50. The figure clearly reveals that the percent of IPs present in the trace decreases as the sampling rate increases (*e.g.*, at sampling rate 100, 20% of IPs remain in the trace relative to no sampling case). Still, the key observation is that the most popular IPs, that are critically needed for the endpoint approach, do stay in the trace, and only marginally decrease as the sampling rate increases.

Figure 5 shows the classification results as a function of the sampling rate. The first observation is that the endpoint

[4]Malware for BLINC indicates scan traffic. However, for our endpoint approach it includes trojans, worms, malware, spyware and bot infected traffic.

[5]We do not compare Malware class due to different definitions between BLINC and UEP.

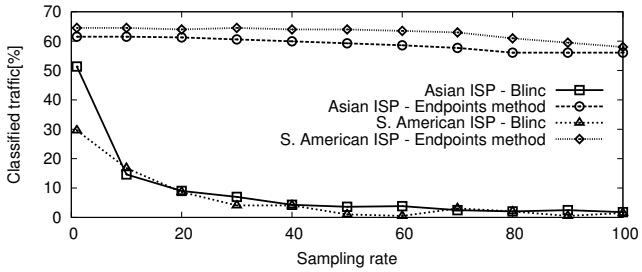

Fig. 5. Classified traffic with the point x=1 representing non-sampled packet-level traffic

approach remains largely unaffected by sampling. Indeed, the percent of classified traffic drops only marginally. This is exactly due to the slight drop in the percent of popular IPs at high sampling rates. At the same time, BLINC's performance degrades as the sampling rate increases, for the reasons explained above. In particular, at sampling rate 40, the classification rate drops below 5%, and for the rate of 100, it becomes close to zero. In fact, even at sampling rate of 100, the endpoint approach identifies all the classes of traffic whereas BLINC is completely unable to identify any class.[6] Finally, worth noting is that the endpoint approach shows consistent results for our third trace (again around 60%). We do not show it in Figure 5 because it is a Netflow trace with the sampling rate of 1:200.

IV. Endpoint Profiling

Next, we apply our methodology to answer the following questions: (i) how can we cluster endpoints that show alike access patterns and how similar or different are these classes for different world regions, and (ii) where do clients fetch content from, *i.e.*, how local or international are clients' access patterns for these regions? In all scenarios, we utilize the maximum possible information that we have, and apply our approach accordingly. When no traces are available (Europe), we stick with pure endpoint approach (Section III-B). When packet level traces are available (Asia and S. America), we apply the endpoint approach as explained in Section III-C. Finally, when flow level traces are available (N. America), we apply the approach from Section III-D.

A. Endpoint Clustering

1) Algorithm: First, we introduce an algorithm we selected to perform endpoint clustering. The key objective of such clustering is to better understand endpoints' behavior at a large scale in different world regions. Employing clustering in networking has been done before (*e.g.*, [20], [22], [36]). We select the autoclass algorithm [19], mainly because it provides *unsupervised* clustering. This means that, in a Bayesian manner, it can actually infer the different classes from the input data and classify the given inputs with a certain probability into one of these classes. The autoclass algorithm selects the optimal number of classes and also the definition of these classes using a Bayesian maximum posterior probability criterion. In addition to accurate clustering, the algorithm also provides a ranking of the variables according to their significance in generating the classification.

[6]Due to sampling, the % of flows in classes may change; accordingly, it is possible that the % of classified flows in a given class increases relative to the non-sampled case.

TABLE VII
Classification on regions

Cls.	S. Amer.	Asia	N. Amer.	Eur.
1	B,C- 0.421	B- 0.644	B- 0.648	B- 0.520
2	B- 0.209	B,C- 0.254	B,M- 0.096	B,M- 0.291
3	B,M- 0.109	P- 0.034	B,C- 0.087	B,L- 0.120
4	B,P- 0.087	G- 0.016	B,L- 0.073	P- 0.064
5	C- 0.077	F,B- 0.015	P- 0.038	S,B- 0.003
6	P,C- 0.068	P,B- 0.015	B,P- 0.036	G- 0.002
7	S,B- 0.022	F,C- 0.012	P,C- 0.017	
8	G- 0.007	S,B - 0.007	P,S- 0.003	
9		P,S- 0.003	G- 0.002	
B browsing, C chat, M mail, P p2p S streaming, G gaming, L malware, F ftp				

For each of the regions we explore, input to the endpoint clustering algorithm is a set of *tagged* IP addresses from the region's network. Since in this case we are interested in the access behavior of users in the network, we determine the tags via an extension of the mapping in Table IV. For regions with traces, if an *in-network* IP address sends/receives traffic to/from an *out-network* IP address that is tagged by a server tag, *e.g.*, as `website`, then the in-network address is tagged appropriately (using the mapping from column 2 to 3 in the table) as browsing. For regions with no trace (Europe), if an in-network IP address has a client tag found via the endpoint method, then it is tagged via the mapping from column 1 to 3 in the table and we also note the URL[7]of the site where the tag was obtained from. Thus, the in-network IP addresses are tagged as browsing, chat, mail, p2p, ftp, streaming, gaming, malware or combination thereof. The sample set for the explored networks is around 4,000 in-network IP addresses for all regions except N. American, where we gather about 21,000 addresses.

2) Evaluation: Table VII lists the top clusters generated for each region. It also provides the proportion of endpoints from a region that were grouped into a cluster. It should be noted that this result captures *correlation* in clients' behavior, not necessarily the absolute presence of a given characteristic. The insights from Table VII are as follows.

First, browsing along with a combination of browsing and chat or browsing and mail seems to be the most common behavior globally. Another interesting result is that gaming users typically do not engage in any other activity on the Internet. Indeed, gaming users are clustered in a separate group of their own in *all* scenarios. Likewise, Asian users show a much higher interest in Internet gaming relative to other regions. This is not a big surprise given the known popularity of Massively Multiplayer Online Role-Playing Games (MMORPG) in Asia [3]. Finally, it is worth noting that p2p users do engage in other online activities such as browsing and chat globally albeit in varying proportions.

Interestingly enough, these global trends remain the same irrespective of the trace duration. For instance, the Asian and S. American packet-level traces are of short duration (order of hours) while the N. American trace is of the order of

[7]The use of the URL is explained in the next subsection on Traffic Locality.

several days. Most importantly, the global trends are the same for the European network for which we relied strictly upon the endpoint approach, without using *any* operational traces. This implies that even in the absence of operational network traces, valuable information regarding endpoints' behavior can be effectively gleaned from the web.

B. Traffic Locality

Next, we explore where do clients fetch the content from, *i.e.*, how local or global are clients' access patterns? Such patterns might not necessarily reflect clients' interests at the social or cultural levels. For example, a client might access highly 'global' content, generated at another continent, by fetching it from a nearby Content Distribution Network's replica. Likewise, clients can get engaged in a strictly 'local' debate at a forum hosted at the other part of the world. Still, we argue that the results we present below are necessarily affected by clients' interests at social and cultural planes as well.

We proceed as follows. First, from the mechanism mentioned in Subsection IV-A1 we obtain a pair of *in-, out-network* IP addresses for each flow. Note that for the case where we only have the URL, we obtain its corresponding IP address via DNS lookup. Next, we obtain the AS-level distance between the two IP addresses by analyzing the BGP Routing Tables as obtained from Routeviews [15]. Finally, we resolve the country code for a given destination AS by using the relevant Internet Routing Registries database (ARIN, RIPE, APNIC and LACNIC).

Figure 6 shows the results obtained from traces. The above plots in the figure show AS-level distance among sources and destinations; the plots below show the country code distribution for a given AS destination. As an example, for the S. American trace, the AS-level figure shows that the majority of the destinations are 2 AS-level hops away from the sources. The corresponding figure below indicates that destinations two AS hops away from sources reside in Brazil (around 30%), in US (around 30%) and in Europe (about 20%), *etc*.

The most interesting insights from Figure 6 are as follows. First, results for China show very high locality: not only are the majority of destinations in China as well, but majority of communication beyond country borders still stays in Asia. Surprisingly (or not), similar behavior holds for US, where the vast majority of content is fetched from within US. Quite opposite behavior holds for S. American endpoints. In addition to the local access patterns, they show strong global behavior as well: S. America's clients fetch a lot of content from US and Europe.

Figure 7 show the results obtained by solely using the Google based approach. While for obvious reasons (time vs. IP space average, see Section III-B) we cannot fully match the trace- and Google-based results, the conclusions we drew above still hold. That is, clients residing in China and North America fetch mostly local content. For China, 76% of the accesses are local while for North America, 67% of the accesses are local. In the case of Brazil the local accesses are around 17% in accordance with the low local traffic observed while analyzing the traces. Another insight (the rightmost plots in Figure 7) is that clients in Europe show highly international behavior: they fetch a lot of content from US and much less from Asia.

When considering p2p traffic locality we notice a slight increase in the intra-AS traffic (compared to the average) - 23% for the Asian ISP, 27% for the S. American ISP and 14% for the N. American ISP. However, notice that a large volume of p2p traffic, *i.e.*, 73% to 86%, is inter-AS that creates well-known problems for ISPs.

V. Endpoint Profiling in a Broader Context

Endpoint profiling in a broader context means using publicly-available information, not necessarily from the web only, to classify endpoints. While we demonstrated above that a large amount of information is indeed available on the web, other sources of information are available as well. Here, we explore two additional sources of information: p2p networks and DNS. We find both approaches to be complementary to our web-based scheme.

A. Endpoint Profiling via P2p Crawling

For p2p communication to progress, the 'entry points' to such systems are necessarily available on the web (*e.g.*, `torrentportal.com`). Yet, the next stage in communication, *i.e.*, getting the appropriate peer IP address to download a file from, is not necessarily available on the web. Still, this information is publicly available. It could be collected by crawling such systems (*e.g.*, [26]).

We have chosen to investigate the BitTorrent system [2], by building a simple BitTorrent crawler. The crawler takes as input a set of file info hashes that we have harvested from the web; next, it contacts the trackers and for each of the files obtains a set of peer IP addresses. To validate these addresses, we contact the peers and request to download data. Then, we proceed and record the IP addresses of the peers that actually respond to our download requests. In this way we have obtained a list of 1,550,173 unique IP addresses spread over the world and somewhat evenly distributed across IP ranges. Once we gathered this information we proceed and tag all the obtained endpoints as BitTorrent p2p nodes.

Next, we explore how this new information can help in the traffic classification scenarios explained in Section III-C above. Necessarily, we classify the (still) unclassified traffic coming from these nodes as p2p traffic. Our results show that we can on average improve the classification result for about 3% in all scenarios. We explain this result below.

First, in our crawling effort, we did not focus on a targeted IP range of interest, but have crawled everywhere. Undoubtedly, if focused on a given area, we expect that better results are possible to achieve. Moreover, while we explored only a single p2p system, crawling other popular p2p systems would certainly further improve the result. In that context, the information obtained by our generic web-based approach can help understand which p2p systems are popular in given regions, hence worth crawling. One final observation is that the classification improvement obtained based on p2p crawling is independent from the traffic sampling rate. This is not a surprise: given that p2p flows are long-lived, despite even very low sampling rate, the IP addresses of interest still stay in the sampled packet trace.

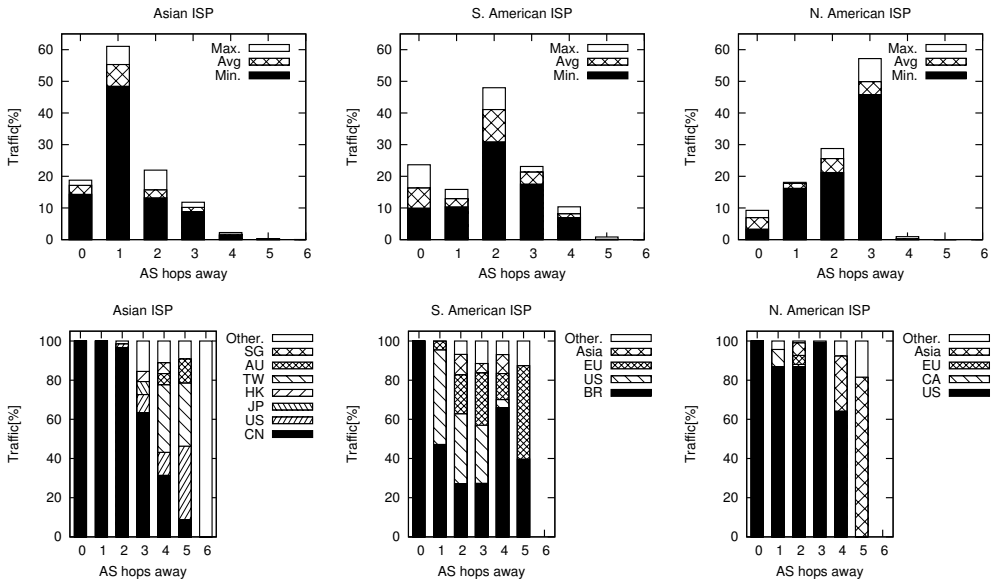

Fig. 6. Traffic locality (from available traces). For this experiment we considered each network range from Table III separately and we computed what percent of traffic [bytes] sinks at a destination a number of AS-hops away and to what country. The Max. Min. values represent the maximum, minimum values that were recorded for a network range in that region while the Avg. is an average across network ranges from that region.

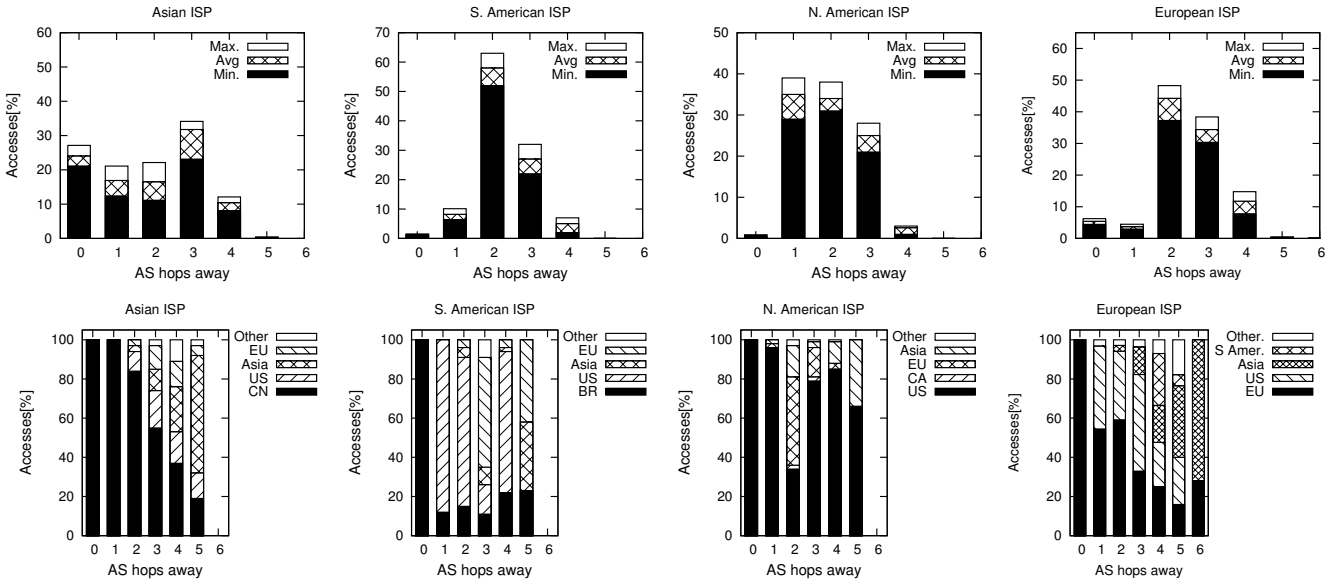

Fig. 7. Traffic locality (Google-based approach). The same experiment as above only instead of traffic from a given region we considered endpoint proofs-of-access, *e.g.*, accesses to a certain forum, obtained from Google.

B. Endpoint Profiling via Reverse DNS Lookups

Another source of information is the reverse DNS lookup service: when presented with an IP address, the system returns the domain name associated with the given address. The question is if useful information about the IP address can be extracted from the given name. Such an approach has been proposed in [33]. In an attempt to find IP addresses associated with routers, the authors mine the domain names that are obtained via reverse DNS lookups. While the authors restrict themselves to discovering router IP addresses, this approach is not limited to routers only; it can be applied to determine endpoints with different functions, *e.g.*, gaming, chat servers, *etc*. We explore such an approach below.

We proceed as follows. We implement a DNS mining technique (details below) and then compare it to the Web-based UEP approach for the S. America's network. In particular, we collect IP addresses classified as a given server type by the Web-based UEP approach in the S. America's range, and then query the reverse DNS lookup service with these addresses. To classify the endpoints based or reverse DNS lookups, in majority of scenarios we apply the *rapid search* approach (Section II) on DNS domain names using keywords such as `gnutella`, `irc`, `emule`, `game`, `proxy`, `mail`, `ad`, *etc*. We do not apply the rapid search approach for web servers, but simply proceed as follows. Once prompted with the domain name, we try to connect to a given server. If successful, we tag the given endpoint as a web server.

Figure 8 shows the results. We compare eight different endpoint types, Gnutella nodes, IRC servers, Emule nodes, Game servers, Proxy servers, Web servers, Ad servers and

Mail servers. The portion of bars in the figure marked by 'UEP' means that these IP addresses were classified by Web-based UEP, but no meaningful information was extractable from DNS reverse lookups. The portion of bars marked by 'No rec.' are IP addresses that are successfully classified by the Web-based UEP approach, but the DNS returned the so-called 'NX Domain,' and hence the reverse DNS lookup cannot give useful information. Finally, the portion of bars marked by 'Found' means that these IP addresses are successfully classified both by UEP and the reverse lookup approach.

Figure 8 shows that Web-based UEP manages to gather much more information than DNS mining is capable of. For example, when dealing with p2p information, either Gnutella or Emule, DNS mining performs really poorly: none of the IPs found by Web-based UEP are found by DNS mining in the Gnutella case, and less then 2% are found in the Emule case. Even in the case of Gaming, Mail, Proxy, or Ad Servers, the information found by DNS mining is not significantly improved (between 6% and 10%). Moreover, a large amount of the servers found by UEP do not even have a DNS record.

The only areas in which DNS mining performs comparable to Web-based UEP is for IRC and Web Servers. Still, it should be noted that extracting web information is not directly possible via DNS, but we had to apply active probing as well. Finally, it is interesting that for about 20% of web servers found by Web-based UEP, no DNS records exist. Thus, while DNS records do provide information about endpoints, this information is relatively limited. At the same time, Web-based UEP does not rely only on a single service but on a multitude of services that make endpoint information available on the web. Finally, the UEP approach is not limited to servers only, it can reveal information about clients as well.

VI. Discussion and Related Work

How accurate is the information on the web? The first question we discuss here is how trustworthy is the information on the web? To get a sense for this, we performed small scale experiments. In particular, we checked links posted on forums; also, we did a port-scan against randomly chosen servers from various server lists available on the web. We found that the information is highly accurate. The vast majority of links posted on forums were active, pointing to the 'right' content. Likewise, the ports that were found active on the servers fully correlate with the information available on the web.

How up-to-date is the information on the web? This is related to the following two questions: (i) How quickly can we detect new or updated information about endpoints? (ii) How can we detect if the information on a given site is outdated? For the first issue, we depend upon `Google`, that is capable of quickly detecting new content on the web; the `Google` crawler determines how frequently content changes on a page and schedules the frequency of crawl to that page accordingly [6]. For detecting outdated information, we can leverage the following information: First, many websites provide information about the time the content was 'last updated'. Likewise, entries on Internet forums typically indicate the date and time of access. In both cases, this information could be used to filter-out outdated information, *e.g.*, older than a given date.

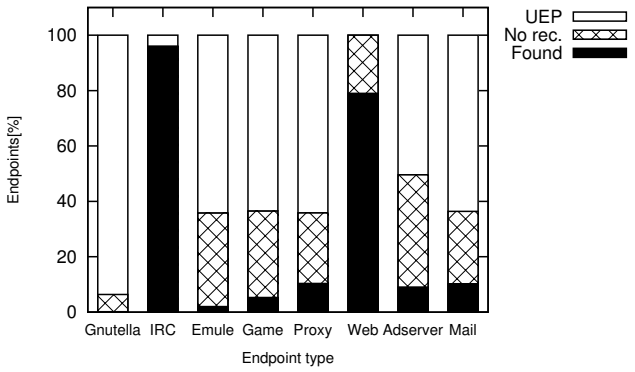

Fig. 8. Comparison of information found via Web-based UEP to information found by mining DNS names.

In order to further analyze this aspect we have conducted a study on the staleness of the information that we have used. We took the top 300 websites that gave most hits that were used in the traffic classification part (using only the tags taken from these websites we have managed to classify 91% of the traffic). On this set we retrieved the Last Update date. We found that for 88% of these websites we could retrieve such a date and 71% of the websites analyzed have been updated in the last month. While via this method we are unable to estimate staleness of a particular IP address, we are still able to show a high level of freshness for the websites that provide information about IP addresses.

Vertical search engines. Information available on the web has traditionally been crawled and indexed by generic search engines such as `Google` [5], `Yahoo` [17] and `Microsoft Search` [11]. However, recently there has been a steady increase in 'vertical search engines' that crawl and index only specific content such as `Indeed` [7], a job search engine and `Spock` [14], a people search engine. To the best of our knowledge, this paper is the first to propose using information available on the web for understanding endpoints, *i.e.*, IP addresses. In this regards, our work can be considered as a first but important step towards developing a vertical search engine for endpoints. Indeed, one of our future research directions is to build such a crawler to index IP address information from the web (instead of overriding on generic search engines).

VII. Conclusions

In this paper, we proposed a novel approach to the endpoint profiling problem. The key idea is to shift the research focus from mining operational network traces to extracting the information about endpoints from elsewhere, *e.g.*, web or p2p systems. We developed and deployed a profiling tool that operates on top of the `Google` search engine. It is capable of collecting, automatically processing, and strategically combining information about endpoints, and finally tagging the same with extracted features. We demonstrated that the proposed approach can (i) accurately predict application and protocol usage trends even when *no* network traces are available; (ii) outperform state-of-the-art classification tools such as BLINC when packet traces are available; and (iii) retain high classification capabilities even when only sampled flow-level traces are available.

We applied our approach to profile endpoints at four different world regions, and provided a unique and comprehensive set of insights about (i) network applications and protocols used in these regions, (ii) characteristics of endpoint classes that share similar access patterns, and (iii) clients' locality properties. Finally, we demonstrated that complementary UEP approaches, such as p2p- or DNS-based schemes, can further improve the Web-based UEP performance.

REFERENCES

[1] Alexa. http://www.alexa.com/.
[2] BitTorrent. http://www.bittorent.com/.
[3] China Gaming. http://spectrum.ieee.org/dec07/5719.
[4] China P2P streaming. http://newteevee.com/2007/08/25/asias-p2p-boom/.
[5] Google. http://www.google.com/.
[6] Google 101: How Google Crawls, Indexes and Serves the Web. http://www.google.com/support/webmasters/bin/answer.py?hl=en&answer=70897.
[7] Indeed: Job Search Engine. www.indeed.com.
[8] L7 Filter Supported Protocols. http://l7-filter.sourceforge.net/protocols.
[9] Linux in Brazil. http://www.brazzil.com/2004/html/articles/mar04/p107mar04.htm.
[10] Linux in France. http://news.zdnet.com/2100-3513_22-5828644.html.
[11] MSN Search. http://search.live.com/.
[12] Narusinsight Traffic Intelligence Platform. http://www.narus.com/products/trafficIntelligence.html.
[13] Orkut. http://en.wikipedia.org/wiki/Orkut.
[14] Spock: People Search Engine. www.spock.com.
[15] University of Oregon Route Views Project. http://www.routeviews.org.
[16] Wikipedia Ban. http://www.iht.com/articles/ap/2006/11/17/asia/AS_GEN_China_Wikipedia.php.
[17] Yahoo. http://www.yahoo.com/.
[18] YouTube. http://www.youtube.com/.
[19] P. Cheeseman and J. Stutz. Bayesian Classification (AutoClass): Theory and Results. In *Advances in knowledge discovery and data mining*, pages 153–180, 1996.
[20] H. Chen and L. Trajkovic. Trunked Radio Systems: Traffic Prediction Based on User Clusters. In *IEEE ISWCS*, Mauritius, September 2004.
[21] Ellacoya Networks. Web Traffic Overtakes Peer-to-Peer (P2P) as Largest Percentage of Bandwidth on the Network, June 2007. http://www.circleid.com/posts/web_traffic_overtakes_p2p_bandwidth/.
[22] J. Erman, M. Arlitt, and A. Mahanti. Traffic Classification using Clustering Algorithms. In *ACM SIGCOMM MINENET Workshop*, Pisa, Italy, September 2006.
[23] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: Multilevel Traffic Classification in the Dark. In *ACM SIGCOMM*, Philadelphia, PA, August, 2005.
[24] T. Karagiannis, K. Papagiannaki, N. Taft, and M. Faloutsos. Profiling the End Host. In *PAM*, Louvain-la-neuve, Belgium, April 2007.
[25] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee. Internet Traffic Classificatoin Demystified: Myths, Caveats, and the Best Practices. In *ACM CONEXT 2008*, Madrid, Spain, December 2008.
[26] J. Liang, R. Kumar, Y. Xi, and K. Ross. Pollution in P2P File Sharing Systems. In *IEEE INFOCOM*, Miami, FL, March 2005.
[27] H. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane: An Information Plane for Distributed Services. In *OSDI*, Seattle, WA, November, 2006.
[28] J. Mai, C. Chuah, A. Sridharan, T. Ye, and H. Zang. Is Sampled Data Sufficient for Anomaly Detection? In *ACM IMC*, Rio de Janeiro, Brazil, October 2006.
[29] P. McDaniel, S. Sen, O. Spatscheck, J. van der Merwe, W. Aiello, and C. Kalmanek. Enterprise Security: A Community of Interest Based Approach. In *NDSS*, San Diego, CA, February 2006.
[30] A. Mislove, M. Marcon, K. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and Analysis of Online Social Networks. In *ACM IMC*, San Diego, CA, October 2007.
[31] L. Plissonneau, J. Costeux, and P. Brown. Analysis of Peer-to-Peer Traffic on ADSL. In *PAM*, Boston, MA, March 2005.
[32] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang. BGP Routing stability of Popular Destinations. In *ACM SIGCOMM IMV Workshop*, Pittsburgh, PA, August 2002.
[33] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP Topologies with Rocketfuel. In *ACM SIGCOMM 2002*, Pittsburgh, PA, August 2002.
[34] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci. Unconstrained Endpoint Profiling (Googling the Internet). In *ACM SIGCOMM 2008*, Seattle, WA, August 2008.
[35] P. Verkaik, O. Spatscheck, J. van der Merwe, and A. Snoeren. Primed: Community-of-Interest-Based DDoS Mitigation. In *SIGCOMM LSAD Workshop*, Pisa, Italy, September 2006.
[36] S. Zander, T. Nguyen, and G. Armitage. Automated Traffic Classification and Application Identification using Machine Learning. In *IEEE LCN*, Sydney, Australia, November 2005.

Ionut Trestian received his B.S. degree in Computer Science from the Technical University of Cluj-Napoca, Romania, in 2007. He is currently working toward the Ph.D degree at Northwestern University under the supervision of Aleksandar Kuzmanvoic.

His research interests include network measurement, network security, overlay networks and social networks.

Supranamaya Ranjan (S'00, M'06) is a Senior Member of Technical Staff at Narus Inc. He received the M.S. and Ph.D. degrees from Rice University in 2002 and 2005 respectively. He served on the technical program committee for IEEE INFOCOM 2007 and IEEE ICDCS 2008.

His research interests are in the areas of network security, anomaly detection and high-performance distributed systems.

Aleksandar Kuzmanovic is an Assistant Professor in the Department of Electrical Engineering and Computer Science at Northwestern University. He received his B.S. and M.S. degrees from the University of Belgrade, Serbia, in 1996 and 1999 respectively. He received the Ph.D. degree from Rice University in 2004. His research interests are in the area of computer networking with emphasis on design, measurements, analysis, denial-of-service resiliency, and prototype implementation of protocols and algorithms for the Internet. He received the National Science Foundation CAREER Award in 2008.

Antonio Nucci is the Chief Technology Officer at Narus Inc, and received his M.S. and Ph.D. degrees in Electrical Engineering from Politecnico di Torino, Italy, in 1998 and 2003 respectively. In his career, Antonio has published more than 70 technical papers, filed 22 patent applications covering various aspects of networking and co-authored a definitive textbook on managing large IP networks, titled Design, Measurement and Management of Large-Scale IP Networks. Bridging the gap between Theory and Practice, published by Cambridge University Press. In 2007, he was awarded the prestigious InfoWorld 2007 CTO Top 25 for his vision and leadership within Narus and the IT community. His research interests include network design and measurements, traffic analysis and security.