Unconstrained Endpoint Profiling (Googling the Internet)

Ionut Trestian
Northwestern University
Evanston, IL, USA
ionut@northwestern.edu

Supranamaya Ranjan
Narus Inc.
Mountain View, CA, USA
soups@narus.com

Aleksandar Kuzmanovic
Northwestern University
Evanston, IL, USA
akuzma@northwestern.edu

Antonio Nucci
Narus Inc.
Mountain View, CA, USA
anucci@narus.com

ABSTRACT

Understanding Internet access trends at a global scale, *i.e.*, what do people do on 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, which accurately characterizes endpoint behavior by collecting and strategically combining information freely available on the web. Our 'unconstrained endpoint profiling' approach shows remarkable advances in the following scenarios: (*i*) Even when no packet traces are available, it can accurately predict application and protocol usage trends at arbitrary networks; (*ii*) When network traces are available, it dramatically outperforms state-of-the-art classification tools; (*iii*) When sampled flow-level traces are available, it retains high classification capabilities when other schemes literally fall apart. 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 which reveals fascinating similarities and differences among these regions.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations
C.4 **[Performance of Systems]:** Measurement techniques

General Terms

Measurement, Design, Experimentation

Keywords

Google, endpoint profiling, traffic classification, clustering, traffic locality

1. 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.*, [36]) and help address imminent security vulnerabilities (*e.g.*, [34, 44]). 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` [19] is probably the best example: it came 'out of nowhere,' and it currently accounts for more than 10% of the total Internet traffic [24].

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.*, [29, 30]). 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, the state-of-the-art packet-level traffic classification tools (*e.g.*, [29]) are simply inapplicable 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, must be publicly available. In p2p-based communication, in which all endpoints can act both as clients and servers, this means that association between an endpoint 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 — on the web.

The first contribution of this paper is the introduction of a novel methodology, which we term 'unconstrained endpoint profiling', for characterizing 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 [6] 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*, which 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 remarkable 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 dramatically outperform state of the art traffic classification algorithms, *e.g.*, [29], 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 literally fall apart.

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 2 we explain our unconstrained endpoint profiling methodology which we evaluate in a number of different scenarios in Section 3, and apply this approach to four different world regions in Section 4. We discuss related issues in Section 5, and provide an overview of related work in Section 6. Finally, we conclude in Section 7.

2. METHODOLOGY

Here, we propose a new methodology, which we term '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 do 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.

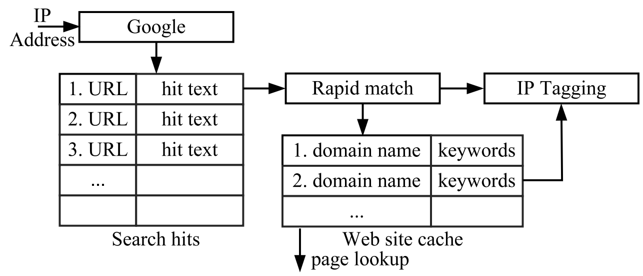

Figure 1: Web-based endpoint profiling

2.1 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, which we present in detail below.

2.1.1 Rule Generation

The process starts by querying `Google` [6] using a sample 'seed set' of random IP addresses from the networks in four different world regions (details in Section 3) 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.

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 1. 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

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

Table 1: 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]

a gaming website (either gaming server list or abuse list). Table 1 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 which is known to originate spam. Similar rules are used to differentiate between websites providing gaming servers list and gaming abuse list.

2.1.2 Web Classifier

Extracting information about endpoints from the web is a nontrivial problem. Our approach is to first characterize a given webpage (returned by `Google`), *i.e.*, determine what information does the website contain. 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 1. For instance, if the URL matches the rule: {`forum` | ... | `cafe`} (see last row in Table 1) 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 2.1.3). 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 currently use a threshold of 2), we retrieve the webpage based on the last URL.[2] Then, we search the webpage for the keywords from the keyword set and extract the ones which 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 Table1) 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.

2.1.3 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.

[2]In an alternative, yet more expensive method, we could have stored all the past URLs and then retrieved all the webpages.

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 1 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, which clearly reveals the given IP's nature and enables tagging. Table 1 provides the mapping between the domain class and IP tags.

For hit texts which 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 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 which 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 1 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 1). 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, which 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 which 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 relegate trade-offs involved in setting such a threshold to future exploration.

2.1.4 Examples

To illustrate the methodology, we provide the analysis of two IP addresses and the corresponding websites returned as `Google` hits: (*i*) 200.101.18.182 - `inforum.insite.com`, and (*ii*) 61.172.249.13 - `ttzai.com`. The first site contains the word `forum` in the URL. Thus, the rapid URL match succeeds and we classify the site as a forum. Next, since the site is classified as forum, we examine the hit text via the forum-based approach; as we find a post date next to a username in the hit text, we tag the IP address as a `forum user`.

In the second case, at first the rapid URL match fails, since the website cache does not contain an entry for `ttzai.com`. Thus, we initially install an entry to this website in the hash table, initialize a counter for number of IP addresses to 1 and log the IP address. Whenever another IP address returns a hit from the same site, the threshold of 2 is crossed. Then, we retrieve the last URL, and a search for the keyword set through the web page reveals the presence of at least one keyword that can classify the site as a Forum site. Further, we proceed to the tagging phase. Because `http:\` is found in front of the original IP address (61.172.249.13), the system concludes that a user deliberately posted the IP address on the forum - as a part of the link to a shared resource. Hence, it tags the IP accordingly.

2.2 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 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.
- *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, in-

Table 2: 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											

formation 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 publicly visible in general. Example websites are `emule-project.net`, `edonkey2000.cn`, or `cache.vagaa.com`, which 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 2 answers the second question: how different are the endpoint information sites in different world regions? In particular, Table 2 shows top entries for three different world regions we explored (details provided in the next section).[3] 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 4 below.

3. 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 3 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 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 3.1) and classify traffic at given networks (Section 3.2). Second, we use these traces to understand how our approach can be applied in the classical traffic classification scenarios, both using packet-level (Section 3.3) and flow-level (Section 3.4) traces.

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

Table 3: 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	

To preserve privacy of the collaborating ISPs, in Table 3, 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.

3.1 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.*, [32]) 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, which spans 32,767 IP addresses. As one source of information about active IPs, we google this IP range. As another source, we extract 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 extract 3,659 active IPs using Google. At the same time, we extract

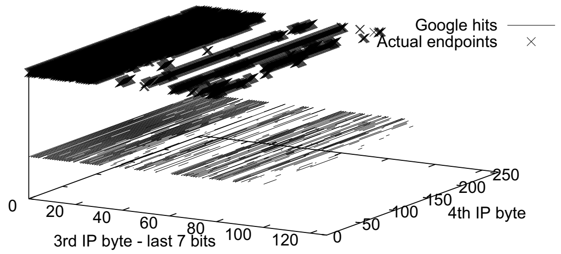

Figure 2: Inferring endpoints - XXX.163.0.0/17

2,120 IPs from the trace. The overlap is 593 addresses, or 28% (593/2120).

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, 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 select a small IP range window.[4] If the distance between 2 IPs is less than the window size, we denote all IPs between the two as 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 stress once again that the key point of this approach is *not* to accurately predict 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.*, [32]). 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.

3.2 When No Traces are Available

Table 8 (Appendix) shows the comprehensive results (including statistics about operating systems, browsers, malicious activity, p2p, protocols and services, chat, gaming, and most popular sites) we obtained by applying the unconstrained endpoint approach on a *subset* of the IP range belonging to the four ISPs shown in Table 3. In particular, we explore approximately 200,000 randomly chosen IP addresses from each of the four world regions. We emphasize that the information in Table 8 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 from Table 8 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.

Correlation with operational traces. We select the S. American trace to exemplify correlation between the results from Table 8 and the network traces. 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 remarkable correlation between the two sources. Specifically, in three of the four traffic categories, we find that the leading applications shown in Table 8 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 Table 8 stay in the same order. For example, results for gaming applications found in the traces are often not in the same order as shown in Table 8. 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 shown in Table 8.

The reasons for this behavior are obvious. The results in Table 8 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 differences in the nature of the data present in Table 8 and that taken from operational networks, there is still a remarkably high correlation. Apparently, when an application is strongly present in a given area, this result shows up consistently both in network traces and in Table 8.

Correlation with other sources. Here, we compare the results from Table 8 with other publicly available sources. One example is the presence of operating systems in different world regions. As we can see, `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 [10–12]. Note that 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. We can see that `Mozilla` is more popular in France and Brazil, as a natural result of the operating systems popularity.

Another example is p2p activity. Table 8 reveals some previously-reported locality tendencies, such as torrents and `eMule` being widely used in France [39], and p2p streaming software being very popular in China [5]. 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 [17] at some point. Similarly, `Orkut`, the social network built by `Google`, shows hits in Brazil, the region where it is very popular [1, 14].

Summary. Strong correlation between the data from Table 8 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 uni-

[4]Numerous experiments on other network ranges corroborate that the window of 17 shows the best compromise between maximizing the overlap between Google- and trace-based active IPs and minimizing the size of enriched subsets.

fied and methodical way for all different world regions, yet *without* using any operational network traces.

3.3 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 [29], which uses a graphlet based approach to classify network traffic. Issues with such an approach are the following. First, 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, which might produce variable quality results for different traces, as we show below. For the same reason, the approach simply falls apart when sampled traffic traces are available, as we demonstrate later.

Table 4: 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

The unconstrained endpoint approach can be applied in a straightforward way to the traffic classification problem. In particular, there is no reason to constrain ourselves to strictly observing packets and connection patterns. Indeed, why not use the externally collected information about the endpoints to classify traffic? Contrary to classification in the 'dark' approaches (*e.g.*, BLINC), we argue that the endpoint-centric approach can not only provide superior classification results, but also efficiently operate at online speeds.

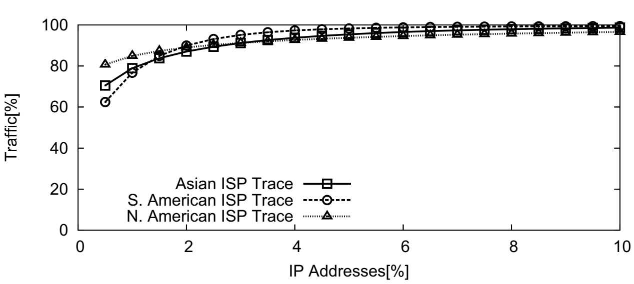

Figure 3: Traffic destinations

The first reason that makes this approach online capable is its ability to classify traffic based on a single observed packet for which one of the endpoints is revealed (*e.g.*, a web server). The second reason is a huge bias of traffic destinations (*e.g.* 95% of traffic is targeted to 5% of destinations [41]). The implication is that it is possible to accurately classify 95% of traffic by reverse-engineering 5% of endpoints, which 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.[5] In particular, we do this in two phases. First, we collect the most popular 5% of IP addresses and tag them by applying the methodology from Section 2. Next, we use this information to classify the traffic flows into the classes shown in Column 3 of Table 4. 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 4.

Table 5 shows the classification results relative to BLINC 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 29.60% in the S. American case (Figure 5 for x=1 and Table 5). 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`.

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. Table 6 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) and only by our approach (U-B). It is clear that our approach uncovers more endpoints and hence classifies more traffic. 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.

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.*, [31]); indeed, we found many of these unclassified IP addresses by joining and searching popular p2p systems (*e.g.*, BitTorrent). This certainly implies that the traffic classification result for the endpoint approach could be further improved. Still, we refrain from pursuing that direction at this point. This is because the information collected from the web is sufficient to demonstrate the superiority of the endpoint approach over BLINC, even more so in sampled scenarios as we show below.

[5] Because the N. American trace is a sampled Netflow trace, we discuss it in the next subsection.

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

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

Table 5: Traffic classes for S. America

Class	Packet trace (% of total flows)		1:100 Sampled trace (% of sampled flows)	
	BLINC	UEP	BLINC	UEP
Chat	0.398	3.38	0.46	5.35
Browsing	23.16	44.70	1.22	40.71
P2P	4.72	11.31	0.3	9.22
Gaming	0.14	0.15	0	0.14
Malware[6]	2.93	2.3	0	0.72
Streaming	0	0.18	0	0.5
Mail	0	1.58	0	3.13
Ftp	0	0.1	0	1.22
Classified	**29.60**	**62.14**	**2.02**	**57.28**
Unclassified	**70.40**	**37.86**	**97.98**	**42.72**
Total	100	100	100	100

Table 6: Endpoints per class for S. America

Cls.[7]	Pkt. trace				1:100 Sampled trace			
	Tot.	B∩U %	B-U %	U-B %	Tot.	B∩U %	B-U %	U-B %
C	1769	16	7	77	484	8	1	91
Br	9950	31	10	59	4964	.4	0	99.6
P	8842	14	8	78	1346	.8	.2	99
G	22	95	0	5	22	0	0	100
S	160	0	0	100	81	0	0	100
M	3086	0	0	100	1179	0	0	100
F	197	0	0	100	52	0	0	100
Br browsing,C chat,M mail,P p2p,S streaming,G gaming,F ftp								
B BLINC, U Unconstrained Endpoint Profiling								

3.4 When Sampled Traces are Available

Not always are packet-level traces 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. While it is well-known that sampled traces can cause problems to anomaly detection algorithms (*e.g.*, [33]), sampled data can create even more significant problems to traffic classification tools, such as BLINC, as well. The key problem is that due to sampling, insufficient amount of data remains in the trace, and hence the graphlets approach simply does not work.

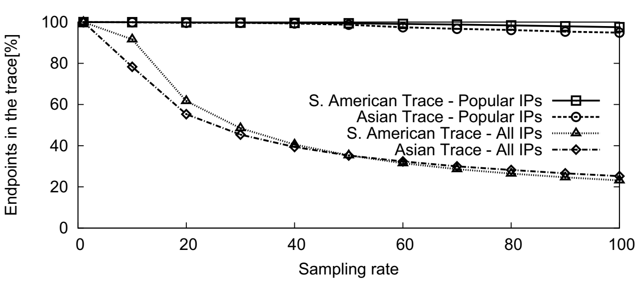

Figure 4: IP addresses

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 of IPs remain in the trace relative to no sampling case). Still, the key observation is that the most popular IPs, which are critically needed for the endpoint approach, do stay in the trace, and only marginally decrease as the sampling rate increases.

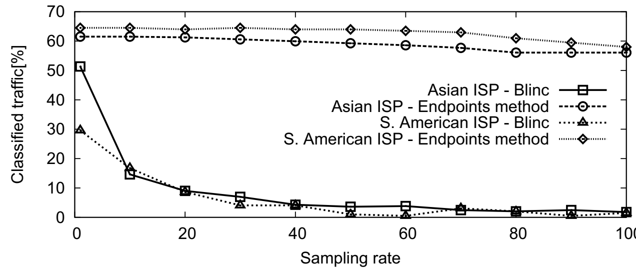

Figure 5: Classified traffic with the point x=1 representing non-sampled packet-level traffic

Figure 5 shows the classification results as a function of the sampling rate. The first observation is that the endpoint 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 dramatically 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 (see Table 5).[8] 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.

4. 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 3.2). When packet level traces are available (Asia and S. America), we apply the endpoint approach as explained is Section 3.3. Finally, when flow level traces are available (N. America), we apply the approach from Section 3.4.

4.1 Endpoint Clustering

4.1.1 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.*, [22, 25, 46]). We select the autoclass algorithm [21], 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

[8]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.

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.

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 4. For regions with traces, if an *in-network* IP address sends/receives traffic to/from an *out-network* IP address which 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[9] 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.

4.1.2 Evaluation

Table 7: 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				

Table 7 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 7 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, 4]. 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 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.

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

4.2 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 4.1.1 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 [16] using the method described in [40]. 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. 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 and European 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; while European clients fetch a lot from US, and much less from Asia.

5. DISCUSSION

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 that we checked 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`, which 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 [7]. 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.

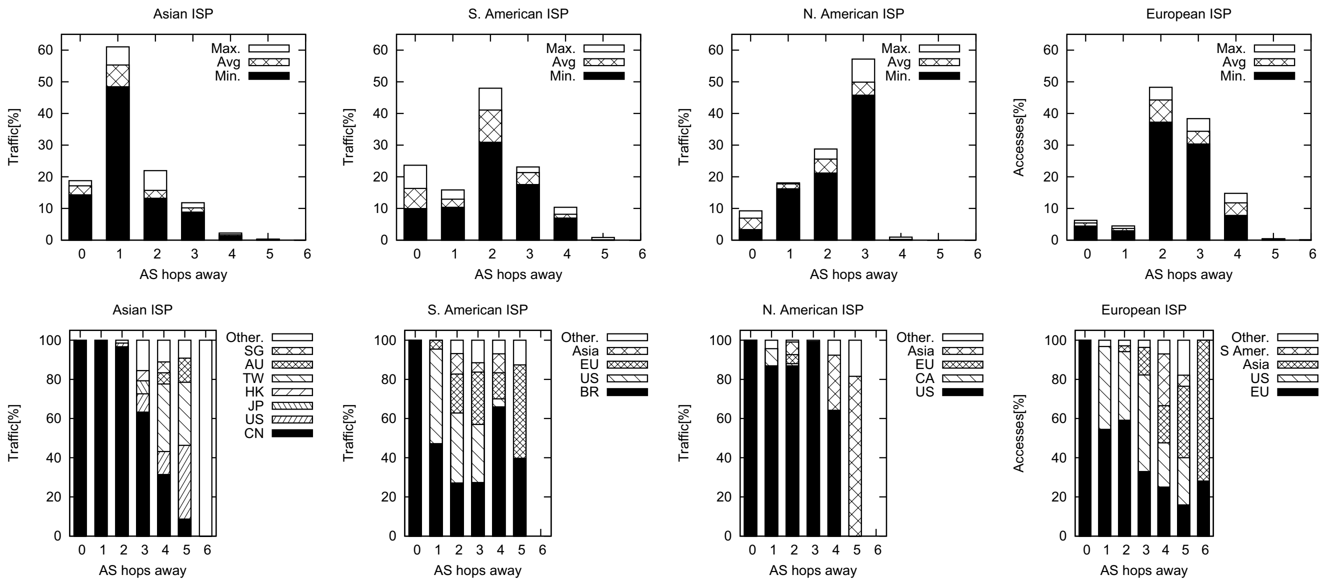

Figure 6: Traffic locality

Dynamic IP addresses. The number of endpoints using dynamic IP addresses is not negligible in today's Internet [45]. Because such IP addresses are used by multiple endpoints, different clients' activities can be projected on the same address. Note that *servers* are typically not run over dynamic IP addresses and even if they are, they have to use a dynamic DNS service to be mapped to a DNS name. In this case, our endpoint classification could be enhanced with information about dynamic IP addresses as obtained from logs maintained at dynamic DNS registries. While our current endpoint classification approach is primarily dependent on information about servers hosted on static IP addresses, it can also be used to accurately detect dynamic IPs. Indeed, if an IP address shows highly active behavior and matches to an abnormally large number of different applications, that could imply a dynamic IP address.

Using other sources of information. Not all information about the endpoints is directly available on the web. The most important example is p2p file sharing or streaming. Indeed, 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 in the web. Still, this information is publicly available. It could be collected in a straight-forward way by crawling such systems (*e.g.*, [31]).

Non-Latin keywords. While we currently rely on parsing Latin language scripts to generate our keyword set, even this allows us to develop interesting insights about the non-Latin language speaking countries, as we have shown while analyzing a network trace from Asia. In future, however, we plan to extend our methodology towards parsing non-Latin language pages in order to develop a more comprehensive keyword set.

6. RELATED WORK

Information available on the web has traditionally been crawled and indexed by generic search engines such as `Google` [6], `Yahoo` [18], `Ask` [2] and `Microsoft Search` [13]. However, recently there has been a steady increase in 'vertical search engines' that crawl and index only specific content such as `Indeed` [8], a job search engine and `Spock` [15], 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).

In the context of correlating multiple sources of information, our work is closely related to [45] and [23]. The authors in [45] correlate email addresses with IP addresses to determine which IP addresses are dynamic. The authors in [23] correlate various IP address lists such as Bots, Phishing sites, Port scanners and Spammers to conclude that botnet activity predicts spamming and scanning while phishing activity appears to be unrelated to others. While similar to [23] one of the tags generated by our method is `malware` we also provide for a wide variety of tags (Table1) using a complete behavior profile for an endpoint.

Most existing traffic classification techniques classify traffic on the basis of characteristics of the traffic stream itself: (i) *port numbers* are used to classify traffic in [26, 28, 29, 37, 38, 42], however, they have been rendered ineffective because applications continually change port numbers to evade detection, *e.g.*, `Skype`; (ii) *payload signatures* are used in [26, 28, 43]. However, their demerit is that payload inspection is expensive and ineffective on encrypted payloads; and (iii) *numerical and statistical techniques* in [20, 27, 29, 35, 38, 42] inspect flows for their properties such as average packet size, average flow duration, distribution of ports, etc., and cluster flows accordingly. However, their effectiveness decreases rapidly with sampling rate as shown in Section 3 for a representative technique, BLINC [29]. We depart from looking into the traffic stream to characterize it, and propose a fundamental shift in the traffic classification problem by first classifying the endpoints themselves via information available on the web. Our 'unconstrained endpoint profiling' is able to achieve high classification rates even at high sampling rates.

7. 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 the web. 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*) dramatically outperform state-of-the-art classification tools 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 residing 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. Our approach opens the doors for revealing people's interests and affinities far beyond those related to network applications and protocols. Indeed, the Internet is only a medium that people use to express their social interests and needs. Generalizing our approach to understand such interests and needs, *i.e.*, by exploring the *content* that clients access, is an exciting research challenge we plan to tackle.

8. REFERENCES

[1] Alexa. http://www.alexa.com/.

[2] Ask. http://www.ask.com/.

[3] China Gaming. http://spectrum.ieee.org/dec07/5719.

[4] China Gaming. http://news.bbc.co.uk/2/hi/technology/4183340.stm.

[5] China P2P streaming. http://newteevee.com/2007/08/25/asias-p2p-boom/.

[6] Google. http://www.google.com/.

[7] Google 101: How Google Crawls, Indexes and Serves the Web. http://www.google.com/support/webmasters/bin/answer.py?hl=en&answer=70897.

[8] Indeed: Job Search Engine. www.indeed.com.

[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] Linux in France. http://www.linuxinsider.com/story/35108.html.

[12] Linux in France. http://www.redhat.com/about/news/prarchive/2007/frenchministry.html.

[13] MSN Search. http://search.live.com/.

[14] Orkut. http://en.wikipedia.org/wiki/Orkut.

[15] Spock: People Search Engine. www.spock.com.

[16] The University of Oregon Route Views Project. http://www.routeviews.org.

[17] Wikipedia Ban. http://www.iht.com/articles/ap/2006/11/17/asia/AS_GEN_China_Wikipedia.php.

[18] Yahoo. http://www.yahoo.com/.

[19] YouTube. http://www.youtube.com/.

[20] L. Bernaille, R. Texeira, and K. Salamatian. Early Application Identification. In *CONEXT*, Lisboa, Portugal, December 2006.

[21] P. Cheeseman and J. Stutz. Bayesian Classification (AutoClass): Theory and Results. In *Advances in knowledge discovery and data mining*, pages 153–180, 1996.

[22] H. Chen and L. Trajkovic. Trunked Radio Systems: Traffic Prediction Based on User Clusters. In *IEEE ISWCS*, Mauritius, September 2004.

[23] M. Collins, T. Shimeall, S. Faber, J. Janies, R. Weaver, and M. Shon. Using Uncleanliness to Predict Future Botnet Addresses. In *ACM IMC*, San Diego, CA, October 2007.

[24] 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/.

[25] J. Erman, M. Arlitt, and A. Mahanti. Traffic Classification using Clustering Algorithms. In *ACM SIGCOMM MINENET Workshop*, Pisa, Italy, September 2006.

[26] P. Haffner, S. Sen, O. Spatscheck, and D. Wang. ACAS: Automated Construction of Application Signatures. In *ACM SIGCOMM MINENET Workshop*, Philadelphia, PA, August, 2005.

[27] F. Herndadez-Campos, F. Smith, K. Jeffay, and A. Nobel. Statistical Clustering of Internet Communications Patterns. In *Computing Science and Statistics*, volume 35, July 2003.

[28] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy, and M. Faloutsos. Is P2P Dying or just Hiding? In *IEEE GLOBECOM*, Dallas, TX, December 2004.

[29] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: Multilevel Traffic Classification in the Dark. In *ACM SIGCOMM*, Philadelphia, PA, August, 2005.

[30] T. Karagiannis, K. Papagiannaki, N. Taft, and M. Faloutsos. Profiling the End Host. In *PAM*, Louvain-la-neuve, Belgium, April 2007.

[31] J. Liang, R. Kumar, Y. Xi, and K. Ross. Pollution in P2P File Sharing Systems. In *IEEE INFOCOM*, Miami, FL, March 2005.

[32] 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.

[33] 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.

[34] 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.

[35] A. McGregor, M. Hall, P. Lorier, and J. Brunskill. Low Clustering Using Machine Learning Techniques. In *PAM*, Antibes, France, April 2004.

[36] 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.

[37] A. Moore and K. Papagiannaki. Toward the Accurate Identification of Network Applications. In *PAM*, Boston, MA, March 2005.

[38] A. Moore and D. Zuev. Internet Traffic Classification Using Bayesian Analysis. In *ACM SIGMETRICS*, Alberta, Canada, June 2005.

[39] L. Plissonneau, J. Costeux, and P. Brown. Analysis of Peer-to-Peer Traffic on ADSL. In *PAM*, Boston, MA, March 2005.

[40] J. Qiu and L. Gao. AS Path Inference by Exploiting Known AS Paths. In *IEEE GLOBECOM*, San Francisco, CA, November 2006.

[41] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang. BGP Routing stability of Popular Destinations. In *ACM SIGCOMM IMV Workshop*, Pittsburgh, PA, August 2002.

[42] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield. Class-of-Service Mapping for QoS: A Statistical Signature-Based Approach to IP Traffic Classification. In *ACM IMC*, Taormina, Italy, October 2004.

[43] S. Sen, O. Spatscheck, and D. Wang. Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures. In *WWW*, Manhattan, NY, May 2004.

[44] P. Verkaik, O. Spatscheck, J. van der Merwe, and A. Snoeren. Primed: Community-of-Interest-Based DDoS Mitigation. In *ACM SIGCOMM LSAD Workshop*, Pisa, Italy, September 2006.

[45] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber. How Dynamic are IP Addresses? In *ACM SIGCOMM*, Kyoto, Japan, August 2007.

[46] S. Zander, T. Nguyen, and G. Armitage. Automated Traffic Classification and Application Identification using Machine Learning. In *IEEE LCN*, Sydney, Australia, November 2005.

Table 8: Traffic mix for studied networks - obtained using solely the `Google`-based approach (no traces)

	Asia (China)	S. America (Brazil)	N. America (US)	Europe (France)
Operating systems	windows(2,445) ubuntu(209) redhat(189) linux(137) unix(92) longhorn(23) slackware(20) debian(17) suse(13) gentoo(10) fedora(10) opensuse(4)	windows(1,783) debian-br(1,700) ubuntu(721) linux(151) redhat(91) fedora(39) unix(13) opensuse(11) mandrivalinux(10) suse(10) gentoo(7) mandrake(5) slackware(5)	windows(659) redhat(310) linux(144) opensuse(100) ubuntu(72) debian(34) suse(20) unix(13) fedora(12) gentoo(10) slackware(2) mandrake(2)	debian(1,206) windows(805) ubuntu(570) linux(556) redhat(263) opensuse(105) mandrivalinux(78) unix(76) mandrake(60) suse(50) fedora-fr(26) gentoo(19) knoppix-fr(10) slackware(1)
Browsers	MSIE(2,694) mozilla(417) opera(48) netscape(29) maxthon(14)	mozilla(1,354) MSIE(1,061) opera(54) netscape(49) enigma(17) maxthon(3)	MSIE(495) nozilla(451) netscape(72) opera(20)	mozilla(515) MSIE(320) netscape(75) opera(29) enigma(8) maxthon(1)
Malicious activity	spam(2,392) net-abuse(2,087) malware(883) dnsbl(253) googlebot(100) blacklist(92) worm(30) virus(29) trojan(21) spyware(17) hijack(5) quakeabuse(4) stormworm(4) banlist(4)	spam(5,532) net-abuse(1,514) blacklist(1,152) blocklist(443) virus(272) dnsbl(239) malware(210) bots(90) googlebot(48) trojan(35) quakeabuse(34) banlist(28) spyware(12) worm(10) hijack(8) stormworm(10)	spam(2,240) bots(259) blacklist(129) googlebot(113) malware(112) dnsbl(89) net-abuse(85) spyware(54) virus(52) hijack(32) adservers(24) worm(20) stormworm(12) trojan(7) banlist(5) quakeabuse(4)	spam(7,672) net-abuse(314) quakeabuse(182) malware(120) banlist(116) blacklist(98) googlebot(98) dnsbl(50) virus(50) bots(35) adservers(16) spyware(15) stormworm(9) trojan(7) hijack(5) worm(5)
P2P	ppstream(12,818) torrent(4,441) Foxy(2,612), gnutella(884) announce(547) tracker(388) p2psky(160) bitcomet(39) edonkey2000(24) eMule(18) ed2k(16) xunlei(14) LimeWire(7) tvants(5) morph500(3) gnucdna(3) Ares(3) Pplive(2)	gnutella(1,560) gnucdna(923) morph500(850) LimeWire(636) torrent(476) tracker(96) ppstream(50) announce(49) Ares(47) emule(16) p2psky(8) ed2k(4) Foxy(3) bitcomet(3)	LimeWire(311) gnutella(274) gnucdna(234) morph500(227) torrent(104) tracker(53) announce(19) Ares(8) p2psky(4) WinMX(2) emule(1) ed2k(1)	torrent(2,125) emule(689) gnutella(317) announce(283) gnucDNA(231) tracker(224) morph500(223) ppstream(153) LimeWire(116) p2psky(68) Foxy(59) ed2k(33) bitcomet(19) edonkey2000(11) Ares(4)
Protocols & services	ftp(10,725) webmail(937) dns(692) email(462) proxy(347) mms(156) smtp(72) mysql(6) pop3(2) netbios(1)	ftp(3,383) webmail(2,638) proxy(1,023) dns(542) email(527) smtp(145) mysql(79) pop3(13) mms(9) netbios(2)	ftp(1,868) dns(386) webmail(326) proxy(302) email(144) smtp(81) mms(23) pop3(13) netbios(2) mysql(1)	ftp(12,417) webmail(7,044) proxy(442) smtp(161) dns(149) email(131) mysql(66) mms(33) netbios(20) pop3(13)
Instant messaging	qq(938) yahoo(700) msn(106) usenet(68) oicq(67) irc(31) icq(25) skype(4)	msn(1,233) yahoo(989) usenet(240) icq(170) qq(126) aol(111) irc(93) skype(1)	yahoo(240) aol(115) msn(61) usenet(32) irc(30) icq(8) messenger(8) skype(6)	yahoo(383) usenet(314) irc(185) aol(89) msn(70) qq(19) gaim(18) icq(18) skype(12)
Gaming	counter-strike(37) quake(36) mmorpg(30) starcraft(21) poker(14) warcraft(6) sims(4)	sims(261) poker(145) counter-strike(144) mmorpg(30) warcraft(19) quake(9) world_of_warcraft(8) halo(4) starcraft(2)	worldofwarcraft(32) poker(14) halo(5) quake(4) sims(2) cstrike(1)	counter-strike(49) quake(43) poker(26) sims(23) warcraft(7) mmorpg(7) world_of_warcraft(5) halo(5) starcraft(2)
Browsing	google(47,584) bbs(32,134) blog(4,282) baidu(3,009) board(2,298) yahoo(700) youtube(356) forums(278) wikipedia(170) rapidshare(6) httpshare(4)	google(61,495) wikipedia(8,245) board(3,239) bbs(1,787) forum(1,436) blog(996) yahoo(989) orkut(564) youtube(370) baidu(76) brturbo(71) rapidshare(20) httpshare(8)	google(2,874) wikipedia(1,819) forums(1,139) bbs(522) board(298) blog(287) yahoo(240) youtube(44) rapidshare(1)	google(20,454) wikipedia(6,637) forum(6,609) blog(728) bbs(709) board(533) yahoo(383) youtube(124) baidu(57) skyrock(12) rapidshare(4)