Proceedings of the 2014 Federated Conference on Computer Science and Information Systems pp. 921–924
DOI: 10.15439/2014F228
ACSIS, Vol. 2

Automated Discovery of Worldwide Content Servers Infrastructure - the SNIFFER Project

Andrzej Bak and Piotr Gajowniczek
Institute of Telecommunications
Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
Email: bak@tele.pw.edu.pl

Marcin Pilarski [1,2] and Marcin Borkowski [1]
[1] Faculty of Mathematics and Information Science
Warsaw University of Technology
pl. Politechniki 1, 00-661 Warsaw, Poland
[2] Orange Labs, Telekomunikacja Polska S.A.
Obrzezna 7, 02-679 Warsaw, Poland

***Abstract*—Service architecture of the Internet becomes more and more complex as it expands as a medium for large-scale distribution of diverse content. Dynamic growth of various content distribution systems, deployed by influential Internet companies, content distributors, aggregators and owners, has substantial impact on distribution of the network traffic and the scalability of various Internet services. The SNIFFER project, presented in this paper, aims to create a service for observing and tracking the long-term growth of various Internet Storage Networks (grids, clouds, Content Delivery Networks, Information-Centric Networks), using the OpenLab and PlanetLab environment. It can be useful to track and map the spreading of such Storage Networks on a global scale, providing more insight into the evolution of Internet towards a content-centric, distributed delivery model.**

I. Introduction

IN RECENT years we have observed an enormous increase in popularity of many Internet services, e.g., Facebook, DailyMotion, YouTube etc. It was possible due to an exponential growth of the number of broadband users and substantial increase in the availability of access bandwidth. During the last five years the Internet backbone traffic has been increasing at a compound aggregate rate of approximately 40% to 50% per year and for the countries of the European Union (EU) the cumulated monthly traffic ranges from 7,500 to 12,000 PB.

The increase of bandwidth usage is closely related to the growth of video traffic in the Internet, spurred by the undeniable trend towards active searching for the preferred content and watching it at the most convenient time. The success of catch-up services (iPlayer, Hulu), online movie rentals over the Internet (Netflix) and watching YouTube movies or podcasts on the TV only confirms this observation.

In order to serve the constantly increasing demand, Internet content service providers deploy content servers in multiple locations all over the world. To obtain high level of scalability and facilitate optimal distribution of popular content to geographically diverse population of end users, such content is usually distributed over multiple physical servers, for example by using the CDN (Content Distribution Networks) technology that utilizes storage located in the network. Such infrastructure, belonging to influential Internet companies, content owners, aggregators, distributors or CDN operators, consists of tenths of thousands of servers deployed throughout the world. Nowadays, it makes up a critical part of the Internet and has substantial impact on distribution of the network traffic and scalability of various Internet services beyond the first and middle mile.

Despite that, very little is known about the topologies, geographical spread, expansion and growth of systems that serve the most popular Internet content worldwide. The main objective of the SNIFFER experiment described in this paper is therefore to create a replicable base for long-running service using OpenLab and PlanetLab environment in order to better observe and track the long-term growth of Storage Networks distributing popular Internet content. The knowledge about location of the content servers and the possibility to monitor long term changes in the infrastructure deployed by popular content distributors, aggregators and owners, would allow better understanding of the nature, complexity and evolution trends of the Internet. It can be also used to improve planning of the Internet underlying transmission resources, which is important as the popular services are progressively more demanding, mainly because of the proliferation of multimedia rich content.

Similar attempts to Internet content server discovery were already undertaken, but lacked versatility (were limited to particular Internet services, such as YouTube [5] [6] [7] or CDNs [8]), sustainability and long-term observation capabilities. In the SNIFFER project we aim to achieve the above goals by developing the following elements that will constitute the final service running on the base of the PlanetLab infrastructure:

- The intercept mechanism, collecting web URLs for pattern discovery and matching to popular Internet services.
- The content server discovery mechanism, providing translation of the discovered web hostnames into IP addresses, clustering, and geo-location of discovered servers.
- The visualization service for easy access to discovered results.

The project uses common Internet protocols, PlanetLab infrastructure and capabilities of Orange Polska as the largest ISP in Central Europe to obtain a large sample of web-related customer activities. The general architecture of the SNIFFER system is presented in Fig. 1.

978-83-60810-58-3/$25.00

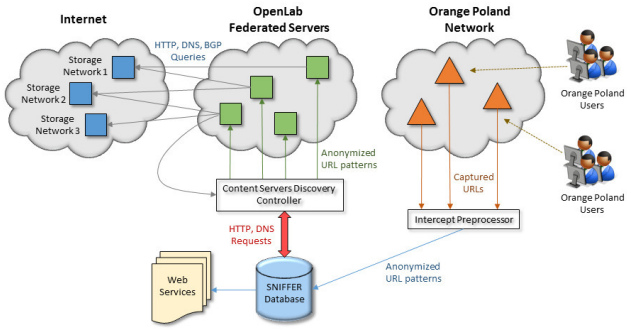

Fig. 1. General architecture of the SNIFFER system

The following sections provide an overview of the main modules of the SNIFFER system.

II. Intercept and Pattern Discovery

This module consists of two functional parts: the capture tool and the pattern discovery tool. The aim of the capture tool is to intercept user traffic and collect URLs of the visited web pages. For this task we use a server equipped with specialized DAG traffic intercept card and an adequate storage. The server is connected to the network with public IP address and 1 Gbps connection. The server also runs the TStat [12] software, functioning as a passive sniffer. It allows capturing the specified traffic at network and transport level. In case of the SNIFFER project, TStat has been prepared for logging TCP communication flows, particularly HTTP requests, as the HTTP GET method contains the URL address of the requested content and hostname of the content server.

The discovered unique URLs are stored on the project server for pattern discovery and further processing. The main function of this part is to analyze the collected URLs and generate URL patterns for selected services in a format that can be easily expanded to generate host names for web server discovery. This approach allows generating hostnames that were not actually intercepted, but for which it is probable that they will resolve to an IP address because of the similarity to some of the intercepted ones. Service selection exploits the fact that the domain names used by the internet service providers are usually equivalent to naming of the services provided to the end users.

III. Content Servers Discovery

To facilitate the conversion of discovered web host names to IP addresses, SNIFFER uses the DNS along with selected PlanetLab servers. The number of distinct IP addresses is used as an indirect measure of the quantity of content servers. A single IP address may however represent a number of physical machines that are indistinguishable for this tool without additional knowledge.

The discovery system takes hostname patterns obtained from previous task, expands them to create a larger set of hostnames, and searches for their IP addresses. The DNS servers are used to resolve the IP addresses, but for each hostname the returned IP address may depend on where the query was issued, as each local DNS can map the hostname to a different server. The goal is to discover as many IP addresses, to which a given URL is resolved in different network areas, as possible.

To obtain wide geographical distribution of queries, and a representative set of server addresses, we use the PlanetLab infrastructure. PlanetLab nodes are located in over 90 sites all over the world, therefore a huge set of local DNS servers can be queried. The system searches for both A and CNAME records. As each CNAME record is followed by complementary A record, at least one IP address is gathered for each hostname.

Popular services quite often have many A records (many IP addresses) assigned to one hostname. For example, a query for domain name *youtube.com* returns 16 A records pointing to 16 different IP addresses. Those 16 records do not exhaust the global list, as the same query executed from different host or after some time may return a different list of 16 IP addresses. Therefore, in case where multiple A records exists, all IP addresses are collected by the querying PlanetLab node.

The SNIFFER experiment does not require that all available PlanetLab nodes are used, as the data acquired by the algorithm are differentiated by geographical location, and so the responses from relatively close nodes are often similar and do not contribute much to the results. Therefore, from all available PlanetLab nodes about two from each top level domain were selected (95 total). As most of those top level domains suggest the country that the node is located in, the selection was driven towards obtaining a uniform distribution of nodes around the globe (to the extent limited by the fact that PlanetLab does not have nodes in every country). PlanetLab nodes availability varies daily, nodes go off-line for various reasons, and therefore at the algorithm initialization the list of on-line nodes is created. Usually around 80% of nodes is ready for use at the same time.

IV. Content Servers Clustering and Geo-tagging

The content server discovery tool collects thousands of IP addresses. Many of the servers behind these addresses are located in the same data centers. To get more insight into geographic distribution of the discovered servers, we employed a clustering algorithm that groups the servers together according to their approximate physical location at city level resolution. The IP address is converted to geographic coordinates using IP geo-location services. However, this approach is not sufficient to distinguish server clusters because of limited geo-location accuracy. Therefore, the algorithm also uses IP trace-route information collected from various locations around the world.

Each IP address from the set of IP addresses of the servers discovered for the particular service denotes a host. Actually, it can be a range of hosts behind NAT or a number of IP addresses located on the same machine. In case of NAT, the group can be treated as one powerful host without the loss of precision for the clustering algorithm. The second case leads to ineffective wasting of public IP addresses so this approach is most probably not used in content distribution systems.

A. Phase 1 - Collection of Gateways

For each IP address the algorithm checks the route through the Internet. The route to the host can be different if checked from different locations around the globe. The algorithm is not collecting the whole route but only the last routing device next to the host itself, called a gateway. If the last device is not discoverable, the second device closest to the target is collected, and so on. The gateway with network distance to the target IP address equal to N will be hereafter denoted by *gwN*.

The addresses of gateways leading to the same server can be different when the path is checked from different locations. The reason for this is that data centers rarely use a single edge router and may also utilize more than one ISP connection for efficiency and reliability. The algorithm collects *gwN*s for given IP addresses from more than 90 PlanetLab nodes and stores them on a dedicated server.

B. Phase 2 - Aggregation

In the second phase of the algorithm only the gateways with network distance one (*gw1*) are considered. Hosts are aggregated by the same gateways, creating clusters. In addition, the number h_N, denoting how often the host was accessible through the particular gateway, is stored.

C. Phase 3 - WHOIS Tagging

The IP address of each gateway is looked up in worldwide domain names register (known as the WHOIS database) to determine the single owner IP range (CIDR) it is in. This is necessary to group similar (belonging to the same organization) gateways later on. The names of clusters formed later are derived not from gateways but from CIDR's. Additionally, those CIDRs/ranges represent the network providers for the data centers. One issue in this process is that even if the WHOIS database is publicly available, the format of the answer is not standardized. It may return the CIDR notation (eg.201.218.32/19), but also the range (e.g., 195.182.218.0-195.182.219.255). Some WHOIS queries also fail, leading to dropping the data related to such query (however, the loss is marginal).

D. Phase 4 - Cluster Candidates

For each unique host IP, the data from previous phase is aggregated into the triple $\{$*cluster name*, *gateway list*, $h_N\}$. The cluster name is formed from all unique CIDRs from the set of triples with the same host IP. To make those names easily comparable, CIDR ranges were lexicographically sorted. The gateway list includes all gateways associated with the host IP and the h_N now represents a cumulative value for all of them.

E. Phase 5 - Cluster Geo-tagging

In this phase the location of the host part of all triples is acquired using geo-tagging tool, and the result is appended to the name of the cluster candidate, as the algorithm assumes that the physical location of the host determines the cluster position on the map. In this way some cluster candidates that have the same name will now have distinct names as they hold hosts at different locations. After geo-tagging, cluster candidates become final clusters.

F. Phase 6 - Aggregation of results

The results of cluster geo-tagging are aggregated by cluster names. The clustering process may omit some IP addresses from the input data due to trace-routing limitations. If a trace cannot find the gateway at distance one (*gw1*) it searches for more remote gateways (*gw2*, *gw3* etc.) but in phase 2 of the algorithm those gateways are filtered out. If a host is not reachable from any of the used PlanetLab nodes via *gw1* it is excluded from the clustering. To address this issue the list of left out IP numbers is processed by the algorithm once again with filtering in phase 2 changed to *gw2*. The resulting clusters are less reliable than the ones obtained in the first run, thus they are stored separately. After second run there may still be some IP addresses left, but at this stage there is no need for the third run of the algorithm with *gw3* filtering, as the number of them is usually minimal.

G. Remarks on Clustering and Geo-tagging Implementation

On each PlanetLab node the routes to tested hosts (IP addresses) are checked with the excellent *Paris traceroute* tool. The important advantage that this tool holds over the classic *traceroute* is the immunity to routers' load balancing. The whole process of checking all IP addresses for given service is rather time consuming so the cache for the queries is used and stored on SNIFFER server. A list of hosts sent to the PlanetLab node is filtered so that only the IP addresses not yet traced (from this node) are tested. The remaining traces are removed from the cache.

Clusters formed in this way should represent close estimation of real life data centers. However, as the algorithm is based on trace-routing data, it detects layer 3 network connections but cannot detect layer 2 links. Consider an example where many hosts are connected to the Internet through two gateways but the internal subnetwork (VLAN) is spanned over 3 switches, where one of them is located in a different data center and connected via a VLAN tunneling protocol (there are various technical methods to extend a single VLAN in such a way). This case can lead the algorithm to aggregation of data centers with the same gateways and different physical locations into one center.

At this point the clustering geo-tagging steps in. However, the geo-tagging accuracy varies a lot between various methods and IP databases. Not all owners of IP addresses want to reveal the exact location of the hosts, therefore the geo-tagging services and tools are imprecise by nature and evolve in time as the IP networks change. Currently, SNIFFER uses only a free of charge MaxMind GeoCity Lite database. It offers city level location service but in practice for a lot of IP addresses the tagging results are not accurate enough. Many tags can be resolved only down to a country or even a continent level. This deficiency is affecting the precision of the clustering algorithm.

V. SNIFFER VISUALIZATION SERVICE

SNIFFER web interface is available at *https://sniffer.mini.pw.edu.pl/*. The website was designed to present the most important results generated by the SNIFFER experiment in a graphical form, as the "snapshots" of worldwide content server infrastructure, taken at various points in time.

SNIFFER web pages use world maps rendered by the Google Maps V.3 engine and API (customized for the specific requirements of the project using JavaScript code). The GUI server is running on open source tools, such as Apache WWW server, MySQL database, Drupal Content Management System and other applications and libraries. It pulls the preprocessed experiment data from the data server in a daily routine, importing cluster lists, patterns, metadata related to specific run of the experiment, and special datasets prepared for comparing the results from various time points. The data is then rendered using the mechanism of Drupal views.

Data presentations accessible from SNIFFER website are created dynamically from database content which makes them very flexible. At present it is possible to visualize location of content servers of Akamai and YouTube discovered in a selected experiment, or in a form of differential maps showing changes in the discovered infrastructure between two different runs. In addition, a user can access various details and statistics of the experiment data, such as IP addresses, CNAMEs, and patterns found during the discovery process.

An example experiment executed in April 2014 took about 40h. The 315 URL patterns identified by intercepting end-user web requests were further used by the Content Servers Discovery module to search for Akamai servers using DNS queries from 76 geographically dispersed PlanetLab nodes. About 10,000 IP server addresses were discovered in result of this process, grouped into 585 clusters and geo-located to produce the map shown in Fig 2.

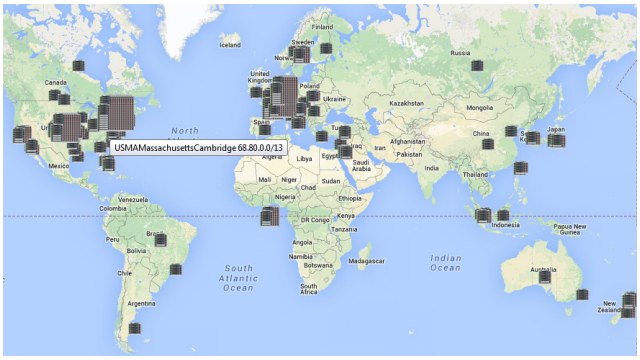

Fig. 2. Discovered locations of the Akamai servers

VI. CONCLUSION

In the paper we have described the SNIFFER project aiming to create a long-running measurement platform to monitor the location and evolution of the content distribution servers in the Internet.

The key difficulties encountered during the ongoing development of the platform were mostly related to precise clustering and geo-location of the discovered data centers. The clustering algorithm that uses *traceroute* and *whois* tools appeared more complex in practice than it was foreseen because of the difficulties in obtaining the proper gateway addresses and their actual locations. Some of the problems can be attributed to the deficiencies of the free IP geo-location database used in building the system.

During testing of the SNIFFER system in the development phase some variability in IP addresses found in consequent experiments was observed. This phenomena can be seen on differential maps and may arise in result of load balancing performed by service providers in conjunction with the scale on which they operate. Despite the fact that 95 PlanetLab servers deployed around the world were used to resolve and trace thousands of host names in each experiment, the architecture of investigated systems is so vast that each time some IP addresses fall outside the search. The providers purposeful approach to hide the actual architecture of their systems cannot be also excluded.

The experimental results from the SNIFFER project will be periodically published on the project web page *https://sniffer.mini.pw.edu.pl*. The discovery service in its current form is running from just March 2014, so the results should be still treated as preliminary. We hope however, that after SNIFFER platform is refined and its measurements database grows up, it will be useful in providing insight into evolution and growth of various Storage Networks related to popular Internet services or effective distribution of content.

REFERENCES

[1] V. Gehlen, A. Finamore, M. Mellia, M. Munafò, *Uncovering the Big Players of the Web*, Proc. TMA'12, 2012, pp. 15-28, doi: 10.1007/978-3-642-28534-9_2

[2] L. Grimaudo, M. Mellia, E. Baralis, *Hierarchical Learning for Fine Grained Internet Traffic Classification*, Proc. IWCMS'12, 2012, pp. 463-468, doi: 10.1109/IWCMC.2012.6314248

[3] A. Finamore, V. Gehlen, M. Mellia, M. Munafò, S. Nicolini, *The Need for an Intelligent Measurement Plane: The Example of Time-Variant CDN Policies*, Proc. NETWORKS'12, 2012

[4] I. Bermudez, M. Mellia, M. Munafò, R. Keralapura, A. Nucci, *DNS to the Rescue: Discerning Content and Services in a Tangled Web*, Proc. ACM IMC'12, 2012, pp. 413-426, doi: 10.1145/2398776.2398819

[5] R. Torres, A. Finamore, J.R. Kim, M. Mellia, M. Munafò, S. Rao, *Dissecting Video Server Selection Strategies in the YouTube CDN*, Proc. IEEE ICDCS'11, 2011, pp. 248-257, doi: 10.1109/ICDCS.2011.43

[6] V.K. Adhikari, S. Jain, Y. Chen, Z.-L. Zhang, *Reverse Engineering the YouTube Video Delivery Cloud*, Proc. IEEE Hot Topics in Media Delivery Workshop, 2011

[7] V.K. Adhikari, S.. Jain, Y.. Chen, Z.-L.. Zhang, *Where Do You 'Tube'? Uncovering YouTube Server Selection Strategy*, Proc. IEEE ICCCN'11, 2011, pp.1-6, doi: 10.1109/ICCCN.2011.6006028

[8] C. Huang, A. Wang, J. Li, K.W. Ross, *Measuring and Evaluating Large-Scale CDNs*, Proc. IMC'08, 2008

[9] T. Leighton, *Improving Performance on the Internet*, Commun. ACM, Feb. 2009, Vol 52, No 2, pp. 44-51, doi: 10.1145/1461928.1461944

[10] B. Wong, A. Slivkins, E. Gun Sirer, *Meridian, a Lightweight Network Location Service without Virtual Coordinates*, Proc. SIGCOMM'05, 2005, pp. 85-96

[11] B. Wong, I. Stoyanov, E. Gun Sirer, *Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts*, Proc. 4th USENIX Conf. on Networked Systems Design and Implementation (NSDI), 2007

[12] *TCP STatistic and Analysis Tool*, http://tstat.tlc.polito.it/index.shtml