Large-Scale Geolocation for NetFlow

Pavel Celeda, Petr Velan, Martin Rabek
Institute of Computer Science
Masaryk University
Brno, Czech Republic
{celeda, velan, xrabek1}@mail.muni.cz

Rick Hofstede, Aiko Pras
Centre for Telematics and Information Technology (CTIT)
University of Twente
Enschede, The Netherlands
{r.j.hofstede, a.pras}@utwente.nl

***Abstract*—The importance of IP address geolocation has increased significantly in recent years, due to its applications in business advertisements and security analysis, among others. Current approaches perform geolocation mostly on-demand and in a small-scale fashion. As soon as geolocation needs to be performed in real-time and in high-speed and large-scale networks, these approaches are not scalable anymore. To solve this problem, we propose two approaches to large-scale geolocation. Firstly, we present an exporter-based approach, which adds geolocation data to flow records in a way that is transparent to any flow collector. Secondly, we present a flow collector-based approach, which adds native geolocation to NetFlow data from any flow exporter. After presenting prototypes for both approaches, we demonstrate the applicability of large-scale geolocation by means of use cases. Our prototypes have shown to be scalable enough for deployment on the 10 Gbps Internet connection of the Masaryk University.**

I. Introduction

Monitoring current high-speed networks requires smart capturing, aggregation and storage technologies to cope with the ever-increasing bandwidth and network traffic. Flow monitoring technologies, such as NetFlow [1] and IPFIX [2], aggregate individual packets into flows[1], which makes these technologies more scalable than packet-based alternatives. The actual packet aggregation is performed by flow exporters, which send flow records to flow collectors for the sake of storage (and potentially statistics generation). Data analysis applications, which are often plugins for flow collectors, can retrieve and analyze this data. One type of data analysis is geolocation.

Geolocation of network traffic is the process of identifying the geographical location of hosts, by means of their IP addresses. The addition of this information to network traffic is useful for many activities, such as business advertisements, fraud detection, access control and traffic profiling. Several types of geolocation can be identified. Active geolocation estimates the location of hosts mostly based on delay and topology measurements [3], [4]. This results in a high accuracy, but comes at the expense of lack of scalability, and high measurement overhead [5]. On the other hand, passive geolocation relies on static datasets, such as databases, containing the geolocation information per IP address block. Online databases, such as geoPlugin [6], can be accessed easily from web applications and often apply rate or request limiting. This makes them less suitable for bulk geolocation and high-interaction applications. Offline databases, such as MaxMind GeoLite [7] and IP2Location [8], do not have these limitations. Both the fact that databases can become outdated and that the majority of data used by passive geolocation approaches refers only to a few popular countries, impact the accuracy of these approaches [5].

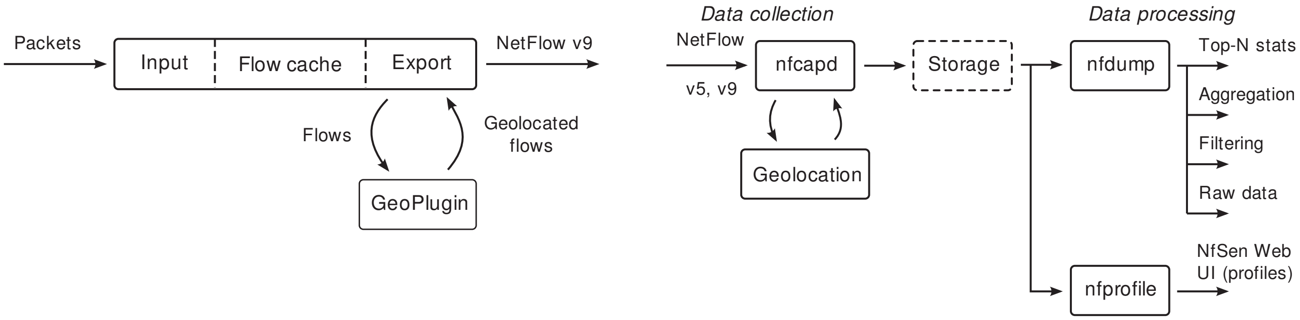
Existing geolocation approaches for flow data are designed for on-demand, mostly small-scale purposes, where the geolocation is performed by analysis applications that retrieve the data from collectors. However, when geolocation needs to be performed in large, high-speed networks, these approaches are not scalable anymore. This is mainly because the dataset to be geolocated is too large for these applications. As multiple flow exporters are typically exporting flow data to a single collector, the data is aggregated even further, resulting in large volumes of data to process for analysis applications. Also, existing applications are often developed for the sake of data visualization, rather than bulk processing.

The goal of this paper is to demonstrate how flow-based geolocation can be performed in a large-scale fashion. As a first step, we propose a prototype for exporter-based geolocation, which adds geolocation data to flow records before they are sent to a collector. As such, the actual geolocation can be distributed over multiple exporters instead of being deployed on a single collector. Since data analysis is always done at a collector or by analysis applications that retrieve data from the collector, we also propose an extension to the state-of-the-art flow collection and analysis tool *NfSen* [9] that adds native geolocation support. We define *native geolocation support* as the ability to process geolocated flow data in the same way as non-geolocated flow data with respect to storage, filtering, aggregation and statistics generation. After presenting the two prototypes, we analyze the performance footprint of our approaches on the primary tasks of flow exporters and collectors, and present several use cases that demonstrate the practical applicability of large-scale geolocation.

The remainder of this paper is structured as follows. Section II provides an overview of related works in the field of geolocation of flow data. The main contribution of this work is described in Section III and IV, where we present how we perform large-scale geolocation on flow exporters and collectors, respectively. In Section V we describe the deployment of our prototypes and in Section VI we present several use cases, which demonstrate their viability. Finally, we close this paper in Section VII, where we draw our conclusions.

[1] We consider a flow as “a set of packets passing by an observation point in a network during a certain time interval and having a set of common properties” [1]. This set of properties typically consists of source/destination IP addresses, port numbers and transport-layer protocol.

978-3-901882-50-0 ©2013 IFIP



(a) Exporter-based geolocation

(b) Collector-based geolocation

Fig. 1: Prototype architectures for large-scale geolocation.

II. Related Work

Due to the increasing number of application areas for geolocation, many related works have been proposed and developed in the past. Closest to our work is *nProbe*, a NetFlow/IPFIX probe for exporting flow data to a collector [10]. This flow exporter uses MaxMind GeoLite [7] for the conversion of IP addresses to geographical locations and AS numbers. The fact that geolocation by *nProbe* is exporter-based makes it scalable for use in large-scale networks. However, by storing the geolocation information in special fields, software support by collectors and applications is required to access this data. Geolocation by *nProbe* is therefore not transparent to any flow collector, which makes this approach different from our work.

Other related works, such as *SiLK* [11], *ntop* [12] and *Argus* [13], are collector-based. *SiLK* and *ntop* add geolocation information to flow data both in an on-demand fashion and as a post-processing step. As such, geolocation information is used purely for visualization and native geolocation is not supported. For example, it is not possible to filter out all flow data from a specific country without geolocating the *whole* dataset upon query execution. *Argus* is more advanced, since it creates an extended dataset from the NetFlow data, in which it can also store geolocation data. As such, full geolocation support is provided as long as the geolocation data has been added to the dataset before.

Besides exporter- and collector-based geolocation approaches, dedicated analysis applications have also been developed. These applications retrieve flow data from flow collectors and subsequently perform the geolocation. *SURFmap* [14], [15], a plugin for the flow collector *NfSen*, uses geolocation to visualize network traffic on a map using the Google Maps API. Due to a dependency on the Google Maps Geocoder for translating location names to coordinates, only a limited number of queries can be executed per day. A *geofilter* has been included since version 2.3, which aims to filter flow data based on country, region or city keywords as a post-processing step. The specification of this *geofilter* is one of the core elements of our collector-based geolocation prototype. Another application is *HAPviewer* [16], which is able to provide flow-level statistics per country of communication partners. Both *SURFmap* and *HAPviewer* perform geolocation on-demand and do not provide native geolocation support.

III. Exporter-Based Geolocation

Flow exporters can perform more tasks than only flow export nowadays, due to their increasing performance. In this work, we propose a prototype for exporter-based geolocation, which adds geolocation data to flow records in a way that is transparent to standard flow collectors. This means that the geolocation data can be accessed by any standards-compliant collector. Usually, multiple flow exporters send their data to a single flow collector. An exporter-based approach therefore aims to distribute the geolocation process over multiple devices. This improves the overall scalability in large networks and reduces the performance footprint on flow collectors.

Our exporter-based geolocation prototype has been developed as a plugin for the INVEA-TECH FlowMon [17] platform. Plugins can be used to alter flow creation, processing, filtering and export. The architecture of FlowMon is shown in Fig. 1a. Packets on the line are received by *input* plugins that store newly created flow records in the *flow cache*. As soon as a record in the cache has been expired (*e.g.* due to a timeout), it is removed from the cache, after which the *export* plugin takes care of placing it in NetFlow packets. Our geolocation plugin is depicted as *GeoPlugin*, which is executed by the *export* plugin just before the record is exported. *GeoPlugin* will do the actual geolocation and add the resulting data to the flow record.

To make sure that our exporter-based geolocation is transparent to any flow collector, NetFlow v9 has been chosen as the export protocol. It provides a fixed set of fields that can be used to store information about a flow. To add country-level information to flow records, we either have to use reserved (vendor proprietary) fields, or reuse some of the existing fields. Only country-level geolocation information is considered in this work, due to the poor accuracy of geolocation databases with respect to region- and especially city-level geolocation data [5]. NetFlow's reserved fields are typically not supported by collectors, which leaves the reuse of existing fields as the only option to ensure transparency, compatibility and early deployment. Since the *SRC_AS* and *DST_AS* fields are rarely used in typical flow monitoring setups (due to the lack of BGP data integration), we use these fields for storing the source and destination countries of IP addresses in flow records. After retrieving the geolocation data from the MaxMind GeoLite database, the resulting country-code is converted to a numer-

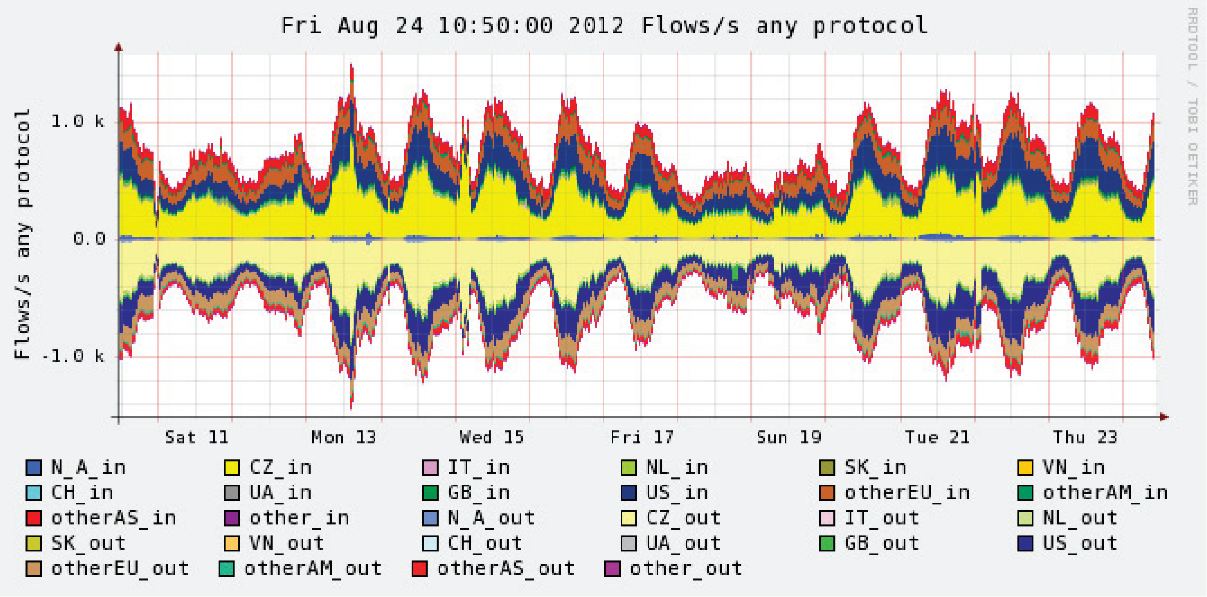


Fig. 2: Screenshot of collector-based geolocation prototype.

ical value and stored in these fields. When flow collectors and analysis applications need to access the country-level information in the flow records, the *SRC_AS* and *DST_AS* fields need to be parsed and the values need to be translated to country codes.

In the future, we want to take advantage of the IPFIX protocol for exporting geolocation information. IPFIX is more flexible than NetFlow v9, supports more fields (named *IPFIX Information Elements*) and makes it easier to define enterprise-specific fields. Several fields for adding location information to IPFIX have been proposed already in [18]. However, those fields are only used for storing the location of an IPFIX flow exporter and are therefore not suitable for flow geolocation. Besides the flexible definition of fields, genuine IPFIX collectors are able to receive new fields and should be flexible enough to process them. However, no suitable IPFIX collectors are available so far.

IV. Collector-Based Geolocation

Flow collectors typically aggregate flow data from multiple flow exporters, which makes them a suitable location to perform data analysis. *NfSen* is a popular collector and analysis tool, used by many network administrators in large-scale networks where performance and stability are essential. *nfdump*, an analysis tool included in *NfSen* that takes care of the actual data analysis, uses a flat-file database with an extensible format[2] to read flow data. Several extensions have been included before, such as SNMP interfaces, AS numbers, MAC addresses and VLANs. We propose a new extension for storing source and destination country codes (based on ISO 3166-1) for IP addresses in flow records. For the same reason as explained for our exporter-based approach, only country-level information is considered. Besides the database extensions, also support for filtering, aggregating and statistics generation based on the geolocation data need to be added to *nfdump*, to provide native geolocation support.

Besides *nfdump*, *NfSen* includes several other tools that need to be modified to provide native geolocation support in a flow collector. Among them is *nfcapd*, which receives flow data from flow exporters, performs the actual geolocation (using MaxMind GeoLite) and writes the data to the disk. Obviously, *nfcapd* also needs to support the new flat-file database format of *nfdump*. Other modifications need to be made to *nfprofile*, which performs the traffic profiling for *NfSen*. The overall architecture of our collector-based geolocation prototype and the data flow between the various components is shown in Fig. 1b.

A screenshot of our prototype is shown in Fig. 2, which demonstrates one aspect of the native geolocation support: The traffic is now automatically profiled by country names. The introduced geolocation support is completely transparent to and compatible with other parts of *NfSen*, and provides near real-time, long-term traffic profiling based on geolocation. No additional tools are required. There are ongoing discussions with Peter Haag, the developer of *NfSen* and *nfdump*, about integration of our geolocation modifications in the main source tree of these tools.

V. Prototype Deployment

The two previous sections have discussed our approaches to exporter- and collector-based geolocation. Both approaches aim to translate IP addresses to geographical locations in a scalable manner, for deployment in large-scale networks. To validate whether this aim is fulfilled, we have deployed both prototypes on the 10 Gbps Internet connection of the Masaryk University (CZ), which connects the campus network to the Czech national research and education network CESNET.

[2] The extensible format is supported by *nfdump* since version 1.5.7.

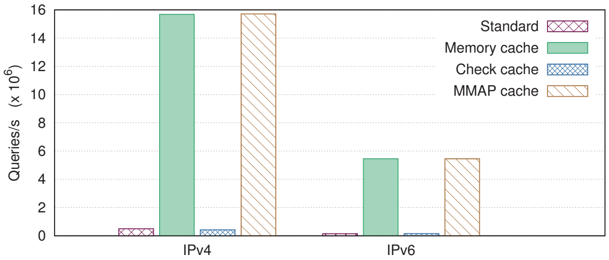


Fig. 3: MaxMind GeoLite country database performance.

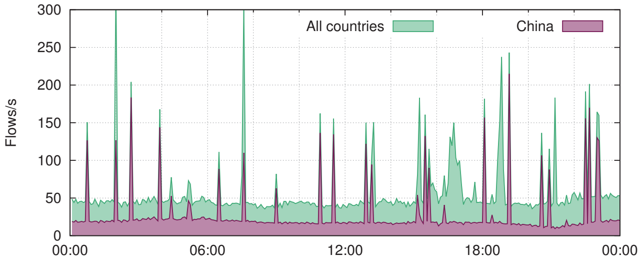


Fig. 4: Incoming TCP SYN-only flows.

The primary tasks of flow exporters and collectors are data export and collection, respectively. Although geolocation adds a useful new dimension to this data, it should never interfere with the primary tasks of these devices. To analyze the performance footprint of our geolocation prototype, we have measured the number of geolocation queries that could be performed per second. MaxMind GeoLite provides four data retrieval types, namely one file system-based (*standard*) and three memory-based (*memory cache*, *check cache* and *MMAP cache*) ones. Besides them, both IPv4 and IPv6 address geolocation have been tested, since they use different databases with different schemas. The measurement[3] results are shown in Fig. 3 and reveal a clear performance increase when either *memory cache* or *MMAP cache* is used: Up to $15.7 \cdot 10^6$ IPv4 and $5.4 \cdot 10^6$ IPv6 addresses could be geolocated per second. Since flow records consist of two IP addresses, roughly $7.8 \cdot 10^6$ IPv4 and $2.7 \cdot 10^6$ IPv6 flow records can be geolocated per second. However, geolocation using memory-based retrieval methods is CPU-intensive and consumes up to 100 % of the CPU time. The derived numbers for the geolocation performance can therefore never be reached in practice. In a theoretical case where either the flow export or collection process may consume 50 % of the CPU time, roughly $3.9 \cdot 10^6$ IPv4 and $1.4 \cdot 10^6$ IPv6 flow records can be geolocated per second. As a result, the performance of exporters and collectors in our deployment setup is not affected in any way, as we have up to $6.0 \cdot 10^3$ flow records per second to process. Neither will it on backbone links of CESNET, where up to $45 \cdot 10^3$ flow records are exported and collected per second.

VI. Use Cases

In this section, we present analysis results from our collector-based geolocation prototype, organized by two use cases. It is based on the deployment described in Section V. The first use case demonstrates the applicability of our (pre-processed) geolocation approaches for the sake of anomaly detection. The second use case presents week-long traffic profiling at the country-level.

A. Anomaly Detection

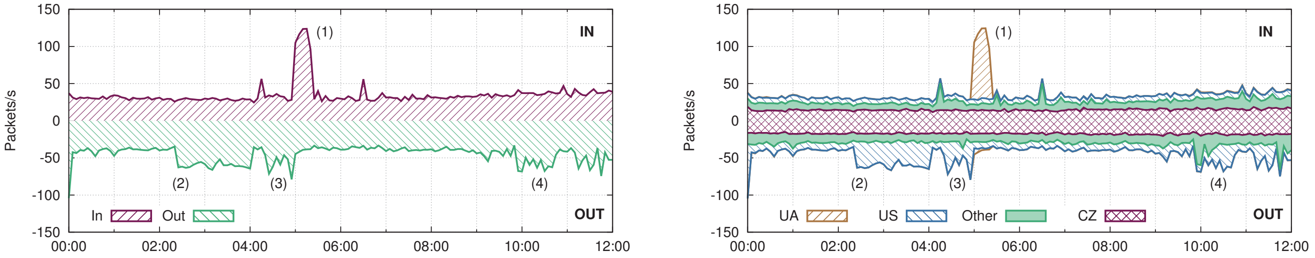
It is a difficult task for network administrators to ensure security awareness in the daily barrage of scans, spamming hosts, zero-day attacks and malicious network users, hidden in huge traffic volumes crossing the Internet. When security teams use geolocation for incident analysis, it is usually applied as a post-processing step. In contrast, our approaches perform geolocation as a pre-processing step, which allows to use geolocation data in the detection process of intrusion detection systems, for example.

An example is shown in Fig. 4, where the number of TCP SYN-only flows[4] per second during a period of one day is shown. These flows are typically created during the propagation phase of malware. We have identified that more than 40 % of them is originating from China. Also, the vast majority of traffic spikes is caused by Chinese connection attempts and only 22 % of Chinese TCP traffic completed the TCP-handshake. These findings demonstrate that this particular Chinese traffic is an interesting dataset for anomaly detection and malware analysis. Further investigation has revealed that using geolocation as a pre-processing step for anomaly detection can yield a dataset to be analyzed that is only 40 % of its original size in this particular case, resulting in faster and less complex data analysis. Another advantage is that this facilitates the detection of anomalies that normally stay below the thresholds of anomaly detection systems. By creating traffic profiles for countries that are known to generate a higher than average amount of malicious traffic, such as China, Russia, Taiwan, Korea and Thailand [19], we have been able to detect long-term stealth attacks.

Another example that demonstrates the advantages of native geolocation support in flow collectors is shown in Fig. 5. Fig. 5a shows an ordinary time-series of ICMP traffic for a period of twelve hours, while Fig. 5b shows the same traffic but geolocated. Several anomalies - marked by numbers in the figures - can be identified, where the geolocated traffic makes the identification clearer due to the larger relative differences between anomalous and non-anomalous traffic. Peak 1 (incoming traffic) is caused by a Ukrainian host, scanning the complete Masaryk University network using ICMP ECHO. Replying hosts were later contacted on TCP port 4899. Peaks 2, 3 and 4 (outgoing traffic) are caused by foreign hosts, which are sending spoofed UDP traffic to the university's DNS server to perform an amplification attack. This traffic is blocked by the firewall and ICMP Destination Unreachable is returned to a spoofed source IP address located in the United States.

[3] We have used a machine with the following configuration: Intel Xeon E5410 CPU at 2.33 GHz, 12 GB RAM, SATA disk with 7200 RPM and Linux kernel 2.6.32 (64 bit).

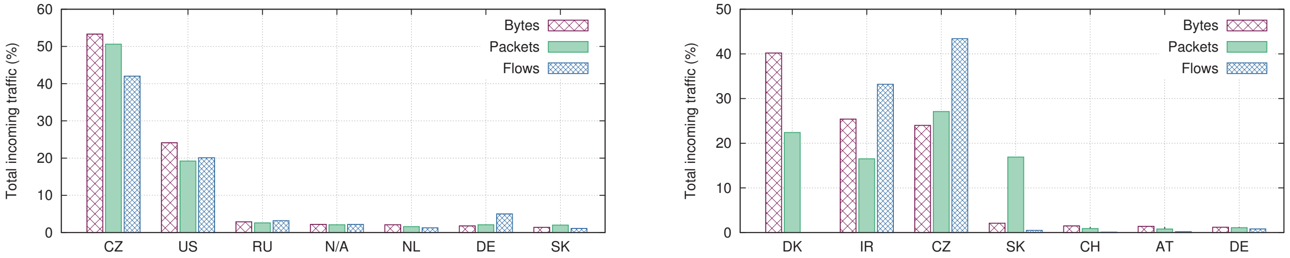
[4] We denote TCP-flows that consist of a single packet with the SYN-flag set by *TCP SYN-only flows*.



(a) ICMP

(b) Geolocated ICMP

Fig. 5: Geolocated and non-geolocated ICMP traffic.



(a) IPv4

(b) IPv6

Fig. 6: Distribution of incoming traffic over countries (1 week).

When network administrators need to find and analyze anomalies using plots as in Fig. 5a, they can identify the peaks and start to filter the data to determine the hosts involved in an anomaly. However, plots as in Fig. 5b make these tasks faster and simpler: Since the traffic is already pre-processed by country names, only the datasets related to a specific country need to be analyzed. In the case of Peak 2, for example, this results in a dataset that is only half of its original size.

B. Traffic Profiling

Traffic profiling is the process of analyzing the distribution of services and protocols in a network, such as the number of packets related to web traffic or the number of flows of a certain type. When geolocation is applied, also statistics related to geographical locations can be generated, such as the number of connections to a certain country. Although this is also possible using existing geolocation approaches based on post-processing, our approach is able to generate these statistics in real-time and for the complete traffic mix. In this subsection, we provide two examples of geolocation-based traffic profiling: IPv4 vs. IPv6 usage statistics and HTTPS profiling.

One method for measuring the world-wide spread of IPv4 and IPv6 deployment is based on counting the number of IPv4- and IPv6-enabled ASNs (Autonomous System Numbers), respectively. Another method is to measure the percentage of IPv4 and IPv6 traffic per country. For our measurement point in the Czech Republic, the distribution of IPv4 traffic source countries is shown in Fig. 6a. The fact that the United States are the source country generating the second most IPv4 traffic is not a surprise, given the fact that US-based social networks (Facebook, Twitter, LinkedIn) and content providers (Akamai, Google, Microsoft) generate a significant portion of the world-wide traffic [20]. This is confirmed by both manual inspection of the traffic and Fig. 7, which shows that almost half of the HTTPS traffic, which is commonly used by those services, is going to and coming from the United States. Although the headquarters of these companies are all in the United States, their data is usually hosted closer to the service users, in regional data centers. This is achieved by using geolocation-aware-DNS (GeoDNS), which provides a means for service users to connect to the server that is closest to them from a geographical point of view. By probing the remote hosts in our data sets actively using ICMP ECHO, it is easily verified that the hosts are definitely not located in the United States, due to resulting non-transatlantic round-trip times. This is an example of a major inaccuracy of geolocation databases, which geolocate IP addresses to the companies' headquarters, instead of the real location of hosts.

The distribution of source countries of IPv6 traffic is shown in Fig. 6b. Next to Czech Republic, much traffic is received from Denmark and Ireland. Closer analysis has revealed that the Danish traffic is caused by large repository synchronization (using FTP) by two data hosting providers. On the other hand, the Irish traffic is generated mainly by Google and Facebook. Surprisingly enough, this traffic is now geolocated to their real physical location. Google is known to have a large datacenter in Ireland and some Facebook partners, such as Zynga, host their services in the Amazon datacenter in Ireland as well.

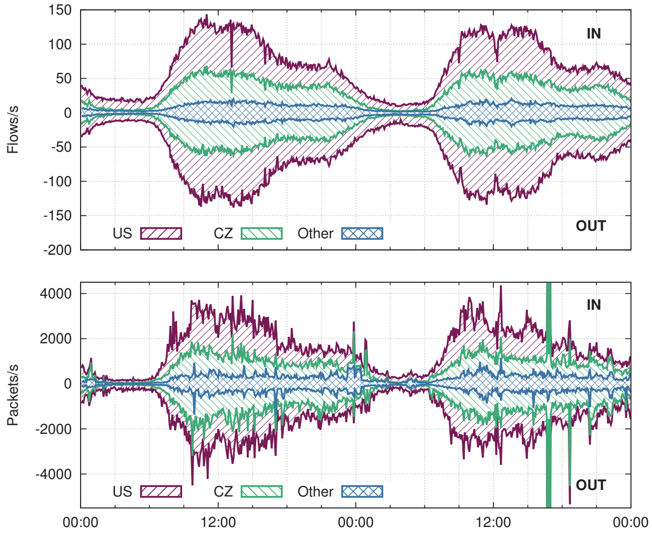


Fig. 7: Distribution of HTTPS traffic over countries.

VII. Conclusions

In this paper we presented two prototypes for flow-based geolocation in large-scale networks. Existing approaches to flow-based geolocation have shown not to be suitable for deployment in those networks or to be nontransparent to flow collectors. By integrating geolocation into flow exporters and collectors, we include country-level information in flow data before the actual data analysis takes place. As such, geolocation is performed in real-time as a pre-processing step, instead of the usual post-processing. This allows network administrators to filter, aggregate and generate statistics based on the geolocated data in a continuous fashion. Measurements have shown that the performance footprint of the geolocation process on the actual flow export and collection is negligible.

To demonstrate the applicability of real-time geolocation in large-scale networks, we have shown several examples by means of two use cases. For example, the use case on anomaly detection has shown that geolocation can help to reduce the dataset to be analyzed by analysis applications dramatically. If certain countries are expected to generate more malicious traffic than others, traffic from these countries can be pre-filtered. Analysis applications, such as anomaly detection systems, can then analyze the smaller dataset, which results in shorter detection times.

As future work, we intend to provide an IPFIX-compliant prototype for exporter-based geolocation. In that case, we do not need to overwrite existing fields while transparency for collectors is preserved. We also plan to analyze how anomaly detection systems can benefit from geolocation information in flow records. These systems can retrieve the geolocated data from our modified *NfSen* collector. By considering the geographical locations of certain hosts, a weight can be added to anomaly detection thresholds, for example. Finally, we continue our efforts on integration of our modifications to *NfSen* in its main source tree.

All implementations used in this work are available at http://dior.ics.muni.cz/~celeda/geolocation/.

Acknowledgements

This work has been supported by Masaryk University, and SURFnet's GigaPort3 project for Next-Generation Networks.

References

[1] B. Claise, "Cisco Systems NetFlow Services Export Version 9," RFC 3954 (Informational), October 2004.

[2] B. Claise, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information," RFC 5101 (Proposed Standard), January 2008.

[3] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe, "Towards IP Geolocation Using Delay and Topology Measurements," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (IMC)*. ACM, 2006, pp. 71–84.

[4] B. Eriksson, P. Barford, J. Sommers, and R. Nowak, "A Learning-Based Approach for IP Geolocation," in *Proceedings of the 11th International Conference on Passive and Active Measurement (PAM)*, ser. Lecture Notes in Computer Science, 2010, pp. 171–180.

[5] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye, "IP Geolocation Databases: Unreliable?" *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 2, pp. 53–56, April 2011.

[6] geoPlugin, "geoPlugin to geolocate your visitors," accessed on 15 January 2013. [Online]. Available: http://www.geoplugin.com

[7] MaxMind, "GeoLite Databases," accessed on 15 January 2013. [Online]. Available: http://www.maxmind.com/app/geoip_country

[8] IP2Location, "IP Address Geolocation to Identify Website Visitor's Geographical Location," accessed on 15 January 2013. [Online]. Available: http://www.ip2location.com

[9] P. Haag, "NfSen," accessed on 15 January 2013. [Online]. Available: http://nfsen.sourceforge.net/

[10] ntop, "nProbe," accessed on 15 January 2013. [Online]. Available: http://www.ntop.org/products/nprobe/

[11] CERT Network Situational Awareness Team (NetSA), "SiLK," accessed on 15 January 2013. [Online]. Available: http://tools.netsa.cert.org/silk/

[12] ntop, "ntop," accessed on 15 January 2013. [Online]. Available: http://www.ntop.org/products/ntop/

[13] QoSient, "ARGUS - Auditing Network Activity," accessed on 15 January 2013. [Online]. Available: http://www.qosient.com/argus/argusnetflow.shtml

[14] R. Hofstede, "SURFmap: A Network Monitoring Tool Based on the Google Maps API," accessed on 15 January 2013. [Online]. Available: http://surfmap.sourceforge.net/

[15] R. Hofstede and T. Fioreze, "SURFmap: A Network Monitoring Tool Based on the Google Maps API," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, June 2009, pp. 676–690.

[16] R. T. Blatter, "Extending HAPviewer: Time Window, Flow Classification, and Geolocation," accessed on 15 January 2013. [Online]. Available: ftp://ftp.tik.ee.ethz.ch:21/pub/students/2011-FS/MA-2011-12.pdf

[17] INVEA-TECH, "FlowMon - Comprehensive Solution for NetFlow Monitoring," accessed on 15 January 2013. [Online]. Available: http://www.invea-tech.com/products-and-services/flowmon

[18] O. Festor and A. Lahmadi, "Information Elements for device location in IPFIX," Internet-Draft, July 2012.

[19] M. van Polen, G. C. Moreira Moura, and A. Pras, "Finding and Analyzing Evil Cities on the Internet," in *Proceedings of the 5th International Conference on Autonomous Infrastructure, Management and Security (AIMS)*, ser. Notes in Computer Science, vol. 6734, 2011, pp. 38–48.

[20] V. Gehlen, A. Finamore, M. Mellia, and M. M. Munafò, "Uncovering the Big Players of the Web," in *Proceedings of the 4th International Workshop on Traffic Monitoring and Analysis (TMA)*, ser. Lecture Notes in Computer Science, vol. 7189, 2012, pp. 15–28.