Grasping Popular Applications in Cellular Networks with Big Data Analytics Platforms

Pierdomenico Fiadino*‡, Pedro Casas*†, Alessandro D'Alconzo*†, Mirko Schiavone*, Arian Baer*

*FTW – The Telecommunications Research Center, Vienna, Austria
†AIT – Austrian Institute of Technology, Vienna, Austria
{forename.surname}@ait.ac.at; arian.baer@gmail.com
‡Eurecat Technology Centre of Catalonia, Barcelona, Spain
pierdomenico.fiadino@eurecat.org

***Abstract*—Internet access through cellular networks is rapidly growing, driven by the great success of the *mobile apps paradigm* and the overwhelming popularity of social-related multimedia services such as YouTube, Facebook, or even WhatsApp. Understanding the functioning, performance and traffic generated by these applications is paramount for ISPs, especially for cellular operators, who must manage the huge surge of volume and number of users with the constraints and challenges of cellular networks. In this paper we study important networking aspects of three popular applications in cellular networks: YouTube, Facebook and WhatsApp. Our evaluations span the Content Delivery Networks (CDNs) hosting these services, their traffic characteristics, and their performance. The analysis is performed on top of real cellular network traffic monitored at the nationwide cellular network of a major European ISP. Due to privacy issues and given the huge amount of data generated by these applications as well as the large number of monitored customers, the analysis has been done in an online fashion, using a customized Big Data Analytics (BDA) platform called DBStream. We overview DBStream and discuss other potential solutions currently available for traffic monitoring and analysis of big networking data. To the best of our knowledge, this is the first paper providing a complete analysis of popular services in cellular networks, using BDA platforms.**

Keywords*—Content Delivery Networks; Cellular Networks; Traffic Measurements; Data Stream Warehousing; Big Data; DBStream.

I. Introduction

A big share of today's Internet ecosystem is shaped by the success and influence of the most popular mobile apps, covering an assorted list of services (e.g., video and audio streaming, social networking, on-line gaming, etc.). Social-related multimedia services such as YouTube, Facebook, and WhatsApp make part of such highly popular services which can be accessed by mobile apps in cellular networks. YouTube is doubtless the most prevailing video streaming service today in terms of users, and more than half of YouTube views come from mobile devices[1], heavily loading the access of cellular networks. Online Social Networks such as Facebook and social sharing messaging apps such as WhatsApp are also becoming a growing headache for ISPs, as they keep on growing in terms of users everyday (1.5 billion users for Facebook and almost 1 billion for WhatsApp as of November 2015). We identify two main challenges for cellular ISPs in this fast evolving context: firstly, there are only a bunch of studies analyzing the functioning and behavior of these services when considering cellular traffic, which hinders a correct understanding of paramount issues such as traffic and usage patterns, content location, addressing dynamics, etc. Such information is paramount for cellular IPSs, to better adapt and manage their own networks, but also to have means to analyze and track the evolution of these popular services. Secondly, there is a lack of network traffic monitoring and analysis systems capable of processing big amounts of packet- and flow-based data in near real time, which makes the tracking and monitoring of applications as the ones we analyze in this paper a cumbersome and daunting tasks for network operators. The recent surge of Big Data Analytics (BDA) platforms opens the door to novel solutions in this direction, but their application to the network traffic monitoring and analysis domain still remains on an early stage.

In this paper we study three massive applications in cellular networks (YouTube, Facebook and WhatsApp) using DBStream, a BDA platform especially tailored for network traffic monitoring and analysis tasks. As we said before, we specifically address these three applications due to their overwhelming popularity and massive consumption in cellular networks today. Our study focuses on three main aspects of these popular and omnipresent services: (i) the characterization of their hosting infrastructures, (ii) the analysis of the traffic, and (iii) the dynamics of the content delivery. The study is based on an extensive analysis of network traffic observed at the core of an operational European cellular network. This line of research has been originally started by the work of Ricciato on traffic monitoring and analysis in cellular networks [1].

Key Findings and Contributions of the Study

The main findings of our study are as follows:

(1) While YouTube's and Facebook's content is hosted in multiple geographical locations and it is provisioned through highly dynamic addressing mechanisms, WhatsApp hosting infrastructure is fully centralized at cloud servers exclusively located in the US, independently of the geographical location of the users. Such a geographical footprint makes users traffic to be hosted in countries other than their local ones, potentially raising concerns about privacy or legal jurisdiction.

(2) The highly dynamic and distributed content delivery mechanisms used by Facebook are becoming more spread in

[1] http://youtube.com/yt/press/statistics.html

terms of hosting locations and hosting organizations, which might have a direct impact on the transport costs faced by the ISP providing the Internet access.

(3) The wide-spread usage of caching in cellular networks and the deployment of Content Delivery Network (CDN) servers at the edge of the ISPs provides high benefits in terms of delay to the contents as well as downlink throughput, specifically for the case of YouTube traffic.

(4) While WhatsApp is mainly used as a text-messaging service in terms of transmitted flows (more than 93%), video-sharing accounts for about 36% of the exchanged volume in uplink and downlink, and photo-sharing/audio-messaging for about 38%. Such a usage of WhatsApp suggests that the application is not only taking over the traditional SMS service of cellular network providers, but it is also heavily loading the access network, particularly in the uplink direction.

(5) Despite the complexity of the underlying hosting infrastructures, traffic volumes and flows in the three services follow a very predictable time-of-day pattern, enabling potential scheduling mechanisms and dynamic traffic engineering policies to optimize the resources of the access network for such massive applications.

Besides these main findings, our paper provides an additional contribution in terms of BDA platforms for network traffic monitoring and analysis: DBStream. DBStream is a flexible and scalable Data Stream Warehouse (DSW) [2] system tailored to network monitoring applications. DBStream is a repository system capable of ingesting data streams coming from a wide variety of sources and performing complex continuous analysis, aggregation and filtering jobs on them. DBStream can store tens of terabytes of heterogeneous data, and allows both real-time queries on recent data as well as deep analysis of historical data. We have made DBStream open source[2] for this paper, which we expect would be highly useful for the network management community in the years to come.

The remainder of the paper is organized as follows: Sec. II presents an overview on the previous papers on massive Internet services characterization, as well as BDA platforms for large-scale data analytics, and in particular targeting network traffic monitoring and analysis. Sec. III overviews DBStream, describing its design and implementation details. Sec. IV describes the analyzed datasets and the methodologies we used in our study. In Sec. V we analyze the content delivery infrastructures of the three services. Sec. VI reports on the characterization of the generated traffic flows, whereas Sec. VII focuses on the content addressing, distribution dynamics and performance. Discussion of the obtained results and their practical implications are presented in Sec. VIII. Finally, Sec. IX concludes this paper.

II. Related Work

We structure the overview on the related work in two different subsections; the former focused on the study and characterization of popular, Internet-scale services and the traffic hosted by the top content Internet providers; the latter reviewing current technologies available for big data analytics, including their application to network traffic monitoring and analysis.

A. Internet Services and Traffic Characterization

The study and characterization of the Internet traffic hosted and delivered by the top content providers has gained important momentum in the last few years [3]–[6]. In [3], authors show that most of today's inter-domain traffic flows directly between large content providers, CDNs, and the end-users, and that more than 30% of the inter-domain traffic volume is delivered by a small number of content providers and hosting organizations. Several studies have focused on CDN architectures and CDN performance [5], [6]. In particular, [5] focuses on user-content latency analysis at the Google CDN, and [6] provides a comprehensive study of the Akamai CDN architecture.

The overwhelming popularity of YouTube in the last few years has attracted the interest of the research community to shed light on its functioning [5], [7]–[10], covering multiple different aspects of YouTube. Some of these works include traffic and usage characteristics [7], [9], content delivery and CDN server selection policies [5], [8], and QoE-based monitoring [10]. The growing usage of YouTube in mobile devices has recently motivated a surge of papers focusing on the analysis of YouTube in cellular networks [11]–[14]. These papers focus mainly on the characteristics of the YouTube traffic as compared to other types of traffic in cellular networks, including the analysis of YouTube content cacheability in mobile contexts. Our study focuses on the analysis of the YouTube performance in cellular networks from an end-user perspective, considering video flows download throughput and simplified QoE metrics.

When it comes to OSNs such as Facebook, there has been a large number of papers in the last few years [15]–[20]. Authors in [15] study the power-law and scale-free properties of the interconnection graphs of Flickr, YouTube, LiveJournal, and Orkut, using application-level crawled datasets. In [16], authors present a study on the privacy characteristics of Facebook. Some papers [17], [18] study the new Google+ OSN, particularly in terms of popularity of the OSN, as well as the evolution of connectivity and activity among users. Authors in [19], [20] focus on the temporal dynamics of OSNs in terms of user-interconnections and visited links, using again public crawled data from popular OSNs such as Facebook, Twitter, as well as a large Chinese OSN. All these papers rely on crawled web-data and do not take into account the traffic and networking aspects of OSNs. In [21] we have started the analysis of the network-side characteristics of large OSNs such as Facebook, particularly focusing on the study of the interplays among the multiple CDNs and domains hosting and delivering the content. In this paper we take a step further, by focusing on the temporal dynamics of the traffic delivery and the traffic flow characteristics.

WhatsApp is a relatively new service, and its study has been so far quite limited. Some recent papers have partially addressed the characterization of its traffic [22], [23], but using very limited datasets (i.e., no more than 50 devices) and considering an energy-consumption perspective. We have been recently working on the analysis of WhatsApp through

[2]DBStream @GitHub, https://github.com/arbaer/dbstream/

large scale network measurements [24], [25], considering in particular the performance of the service, both in terms of network throughput and quality as perceived by the end users. In [25] we studied the Quality of Experience (QoE) requirements for WhatsApp, reporting the results obtained from subjective QoE lab tests. In this paper we do not focus on the performance of WhatsApp but on its traffic and hosting infrastructure, extending the initial results obtained in [24].

B. Big Data Analysis Frameworks

The introduction of Big Data processing led to a new era in the design and development of large-scale data processing systems [26]. This new breed of tools and platforms are mostly dissimilar, have different requirements, and are conceived to be used in specific situations for specific needs. Each Big Data practitioner is forced to muddle through the wide range of options available, and Network Traffic Monitoring and Analysis (NTMA) is not an exception. A basic yet complete taxonomy of Big Data Analysis Frameworks includes traditional Database Management Systems (DBMS) and extended Data Stream Management Systems (DSMSs), noSQL systems (e.g., all the MapReduce-based systems), and Graph-oriented systems. While the majority of these systems target the offline analysis of static data, some proposal consider the problem of analyzing data coming in the form of online streams. DSMSs such as Gigascope [27] and Borealis [28] support continuous online processing, but they cannot run offline analytics over static data. The Data Stream Warehousing (DSW) paradigm provides the means to handle both types of online and offline processing requirements within a single system. DataCell and DataDepot are examples of this paradigm [2]. NoSQL systems such as e.g. MapReduce [29] have also rapidly evolved, supporting the analysis of unstructured data. Apache Hadoop [30] and Spark [31] are very popular implementations of MapReduce systems. These are based on offline processing rather than stream processing. There has been some promising recent work on enabling real-time analytics in NoSQL systems, such as Spark Streaming [32], Indoop [33], Muppet [34] and SCALLA [35], but these remains unexploited in the NTMA domain. The offer of solutions available is overwhelming; more examples include Storm, Samza, Flink (NoSQL); Hawq, Hive, Greenplum (SQL-oriented); Giraph, GraphLab, Pregel (graph-oriented), as well as well known DBMSs commercial solutions such as Teradata, Dataupia, Vertica and Oracle Exadata (just to name a few of them).

The application of Big Data Analysis Frameworks for NTMA tasks requires certain system capabilities: i) scalability: the framework must offer, possibly inexpensively, storage and processing capabilities to scale with huge amounts of data generated by in-network traffic monitors and collectors. ii) Real-time processing: the system must be able to ingest and process data in real-time fashion. iii) Historical data processing: the system must enable the analysis of historical data. iv) Traffic data analysis tools: embedding libraries or plugins specifically tailored to analyze traffic data. In the following we present the main categories in which currently available data analysis technologies can be classified. For each of them, we highlight pros and cons, and explain why none of them fits for NTMA. Traditional SQL-like databases are inadequate for the continuous real-time analysis of data. As we mentioned before, Data Stream Warehouses have been introduced to extend traditional database systems with continuous data ingest and processing. These technologies leverage arbitrary SQL framework to perform rolling data analysis, i.e., they periodically import and process batches of data arriving at the system. In some cases, these technologies have been proven to be able to outperform – in terms of processing speed – new Big Data technologies based on MapReduce [36]. More recent solutions in this direction include ENTRADA [37], a Hadoop-based DSW for network traffic analysis, using off-the shelf Impala query engine and Parquet file format based on Google's Dremel [38] to achieve high performance, relying on columnar data storage. Big Data Analysis Frameworks based on the MapReduce paradigm have been recently started to be adopted for NTMA applications [39]. Considering the specific context of network monitoring, some solutions to adapt Hadoop to process traffic data have been proposed [40]. However, the main drawback of Big Data technologies in general is their inherent off-line processing, which is not suitable for real-time traffic analysis, highly relevant in NTMA tasks. One of the few systems that leverage Hadoop for rolling traffic analysis is described in [41]. As we mentioned before, there have also been some Big Data Analysis Frameworks for online data processing, but none of these has been applied to the NTMA domain.

The results of the individual studies presented in this paper have been partially presented in recent workshops and conferences [42]–[44]. The additional value of this paper is to provide a single source of information compiling all the studies, following in all the cases a well structured and systematically applied analysis procedure and comparing the characteristics of the three studied services. In addition, providing a more complete and detailed overview on the state of the art in BDA platforms in the same paper, and in particular targeting network traffic monitoring and analysis, adds a complete wrapper to the problematic of characterizing and tracking massively popular services in today's cellular networks. Finally, the additional details on DBStream and the release of its code as open software makes of the paper a great source of information for practitioners willing to grab the problem of big network traffic monitoring and analysis from the basis. Indeed, we expect this paper to be highly useful for ISPs willing to increase their visibility and management capabilities on their networks, allowing network operators to better understand how to dimension and operate their access networks in order to correctly provision this novel surge of services.

III. DBStream System Overview

DBStream is a novel continuous analytics system. Its main purpose is to process and combine data from multiple sources as they are produced, create aggregations, and store query results for further processing by external analysis or visualization modules. The system targets continuous network monitoring but it is not limited to this context. For instance, smart grids, intelligent transportation systems, or any other use case that requires continuous process of large amounts of data over time can take advantage of DBStream.

DBStream combines on-the-fly data processing of DSMSs with the storage and analytic capabilities of DBMSs and typical big data analysis systems such as Hadoop. In contrast to DSMSs, data are stored persistently and are directly available for later visualization or further processing. As opposed to traditional data analytics systems, which typically import and transform data in large batches (e.g., days or weeks), DBStream imports and processes data in small batches (e.g., on the order of minutes). Therefore, DBStream resembles a DSMS in the sense that data can be processed quickly, but streams can be re-played from past data. The only limitation is the size of available storage. DBStream thus supports a native concept of time. At the same time DBStream provides a flexible interface for data loading and processing, based on the declarative SQL language used by all relational DBMSs.

Two salient features of DBStream are the following: first, it supports incremental queries defined through a declarative interface based on the SQL query language. Incremental queries are those which update their results by combining newly arrived data with previously generated results rather than being re-computed from scratch. This enables continuous time-series based data analysis, which is a strong requirement for real-time NTMA applications such as anomaly detection. Secondly, in contrast to many database system extensions, DBStream does not change the query processing engine. Instead, queries over data streams are evaluated as repeated invocations of a process that consumes a batch of newly arrived data and combines them with the previous result to compute the new result. Therefore, DBStream is able to reuse the full functionality of the underlying DBMS, including its query processing engine and query optimizer.

DBStream is built on top of a SQL DBMS back-end. We use the PostgreSQL database in our implementation, but the DBStream concept can easily be used with other databases and it is not dependent on any specific features of PostgreSQL.

Last but not least, we have benchmarked DBStream against data processing systems based exclusively on standard PostgreSQL DBMS [44], as well as against MapReduce-based systems including Hadoop [44] and Spark [36], obtaining in all cases outperforming results, even when running in less powerful hardware configurations. For example, in [36] we show that a single DBStream node can outperform a cluster of 10 Spark nodes by a factor of 2.6 when incremental queries are involved. We refer the reader to our previous work [36], [44] for more details about such benchmarks. Even more, DBStream has been operational for more than one year, processing the traffic of a national-wide cellular ISP in Europe with more than 160 queries running continuously on top of the imported traffic. We are currently comparing the performance of DBStream against other stream-based analysis systems – such as Spark streaming [32], we plan to publish the obtained results once we are done with the study. The important message is that DBStream offers high performance and processing power, even when running on limited hardware, as demonstrated both in the practice as well as by extensive benchmarking.

A. System Architecture

In DBStream, *base tables* store the raw data imported into the system, and *materialized views* (or views for short) store the results of queries such as aggregates and other analytics — which may then be accessed by ad hoc queries and applications

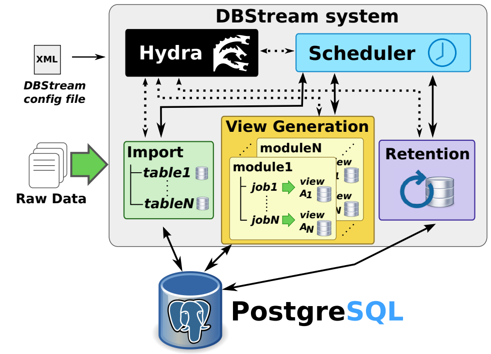


Figure 1. General overview of the DBStream architecture. DBStream combines on-the-fly data processing of DSMSs with the storage and analytic capabilities of DBMSs and big data analysis systems such as Hadoop.

in the same way as base tables. Base tables and materialized views are stored in a time-partitioned format inside the PostgreSQL database, which we refer to as Continuous Tables (CT). Time partitioning makes it possible to insert new data without modifying the entire table; instead, only the newest partition is modified, leading to a significant performance increase.

A *job* defines how data are processed in DBStream, having one or more CTs as input, a single CT as output and an SQL query defining the processing task. An example job could be: *"count the distinct destination IPs in the last 10 minutes"*. This job would be executed whenever 10 new minutes of data have been added to the input table (independently of the wall clock time) and stored in the corresponding CT.

Fig. 1 gives a high-level overview of the DBStream architecture. DBStream consists of a set of modules running as separate operating system processes. The `Scheduler` defines the order in which jobs are executed, and besides avoiding resource contention, it ensures that data batches are processed in chronological order for any given table or view. `Import` modules may pre-process the raw data if necessary, and signal the availability of new data to the `Scheduler`. The scheduler then runs jobs that update the base tables with newly arrived data and create indices, followed by incrementally updating the materialized views. Each view update is done by running an SQL query that retrieves the previous state of the view and modifies it to account for newly arrived data; new results are then inserted into a new partition of the view, and indices are created for this partition. `View Generation` modules register jobs at the `Scheduler`. Finally, the `Retention` module is responsible for implementing data retention policies. It monitors base tables and views, deleting old data based on predefined storage size quotas and other data retention policies. Since each base table and view is partitioned by time, deleting old data is simple: it suffices to drop the oldest partition(s).

The DBStream system is operated by an application server process called `hydra`, which reads the DBStream configuration file, starts all modules, and monitors them over time. Status information is fetched from those modules and made available in a centralized location. Modules can be placed on separate machines, and external programs can connect directly to DBStream modules by issuing simple HTTP requests.

B. Continuous Analytics Language

Many continuous and streaming query languages have been proposed [2], [27], [45], but they assume that data are not stored persistently and that queries can only refer to temporary states (e.g., a current window of time). Typically, DBMSs support defining views using SQL queries, but either the result of the view is generated at query time, or the view is updated whenever a new row of data is added to the original table. This results in very low performance when huge numbers of rows are constantly imported into the original table. In contrast, DBStream enables users and applications to declaratively specify, using arbitrary SQL, exactly how to update a view when a new batch of data is inserted into its source table(s). These specifications may even refer to previously generated results that are stored in the same view, which, to the best of our knowledge, is not *declaratively* supported by any other system. Below we show an example of a typical aggregation query, counting the number of rows per minute and device class. If the input table A has one flow in each row, the number of rows corresponds to the number of flows.

```
<job inputs="A (window 15min)"
     output="B (window 15min)"
  schema="time int4, dev_class int4, cnt int4">
  <query>
  select time - time%1min, dev_class, count(*)
    from A
    group by serial_time, dev_class
  </query>
</job>
```

In more detail, the XML attribute `inputs` is used to define one or more input streams. For each input stream, the batch size is specified with a `window` definition; in the example, the window size is 15 minutes. The `output` attribute is used to specify an output stream, which then can be used as input to other queries. The output stream also has a `window` definition. In addition, for the output stream, the `schema` is defined as the set of data types returned by the query. Note that the first column must be a monotonically increasing timestamp, which is used in the window definitions. Inside the `query` XML element, an SQL query defines how the input(s) should be processed. The result of this query is then stored in the new window of the output table. In the query, all features of PostgreSQL, including the very flexible User Defined Functions (UDF)s, can be used to process the data. Utilizing UDFs, it is easy to add code written in Python, Perl, C, R and other programming languages into the query.

In particular, it is possible to define *incremental queries* through this approach. An incremental query permits to update some specific output table, based on the combined processing of its previous window content and the content of the new input stream window. This is useful when, e.g., computing cumulative counts and sums over long periods of time. In this case, it suffices to add the volumes from the new input window to the cumulative sums maintained in the output table.

To conclude, we have made DBStream open to the research community, aiming at an increasing usage of such technology for the purpose of NTMA. The code is available on GitHub at https://github.com/arbaer/dbstream/, including full installation and deployment details.

IV. Datasets and Analysis Methodology

Our study is conducted on top of three large-scale network traffic traces collected and analyzed on the fly at the core of a European national-wide cellular network in mid 2013 and early 2014. As depicted in Fig. 2, flows are monitored at the well known Gn interface, and analyzed with DBStream. Facebook and YouTube traffic is carried on top of HTTP (we do not consider HTTPS for the study of these services, as its usage in 2013 was very limited in mobile devices), so we rely on a HTTP-based traffic classification tool for cellular traffic called HTTPTag [46] to unveil the corresponding YouTube and Facebook flows. HTTPTag classification consists in applying pattern matching techniques to the `hostname` field of the HTTP requests.

The YouTube dataset corresponds to almost 90 hours of traffic flows collected in the second quarter of 2013. The Facebook dataset consists of one month of HTTP flow traces collected in mid 2013. To preserve user privacy, any user related data are removed on-the-fly, whereas any payload content beyond HTTP headers is discarded on the fly. The WhatsApp dataset consists of a complete week of WhatsApp traffic flow traces collected at exactly the same vantage point in early 2014. In the case of WhatsApp all communications are encrypted, so we extended the HTTPTag classification tool to additionally analyze the DNS requests, similar to [47]. In a nutshell, every time a user issues a DNS request for a Fully Qualified Domain Name (FQDN) associated to WhatsApp, HTTPTag creates an entry mapping this user to the server IPs provided in the DNS reply. Each entry is time stamped and contains the TTL replied by the DNS server. Using these mappings, all the subsequent flows between this user and the identified servers are assumed to be WhatsApp flows. To avoid miss-classifications due to out-of-date mappings, every entry expires after a TTL-based time-out. To increase the robustness of the approach, the list of IPs is augmented by adding the list of server IPs signing the TLS/SSL certificates with the string `*.whatsapp.net`. Indeed, our measurements revealed that WhatsApp uses this string to sign all its communications. Finally, we use reverse DNS queries to verify that the list of filtered IPs actually corresponds to a WhatsApp domain.

To identify the FQDNs used by the WhatsApp service, we rely on manual inspection of hybrid measurements. We actively generate WhatsApp text and media flows at end devices (both Android and iOS), and passively observe them at two instrumented access gateways. We especially paid attention to the DNS traffic generated by the devices. Not surprising, our measurements revealed that WhatsApp servers are associated to the domain names `whatsapp.net` (for supporting the service) and `whatsapp.com` (for the company website). In addition, different third level domain names are used to handle different types of traffic (control, text messages, and multimedia messages). Control and text messages are handled by *chat servers* associated to the domains {`c|d|e`}`X.whatsapp.net` (`X` is an integer changing for load balancing), whereas multimedia contents are handled by *multimedia (mm) servers* associated to the domains `mmsXYZ.whatsapp.net` and `mmiXYZ.whatsapp.net` for audio and photo transfers, and `mmvXYZ.whatsapp.net` for videos. As we see next, chat and mm servers have very different network footprints. While connections to chat servers are characterized by low data-rate

flow_end_time	application	srv_IP	AS	FQDN – 2DL	min_RTT	bytes_up	bytes_down	avg_th_up	avg_th_down
1395394800	YouTube	74.125.14.82	15169 (Google)	youtube.com	36 ms	50 KB	10 MB	500 kbps	4 mbps
1395395100	Facebook	31.13.76.68	32934 (Facebook)	facebook.com	20 ms	200 KB	100 KB	150 kbps	350 kbps
1395135900	WhatsApp	108.168.174.24	36351 (SoftLayer)	whatsapp.net	113 ms	100 KB	1 MB	200 kbps	1.5 mbps

Table I. FLOW-BASED TRAFFIC DESCRIPTORS. ONCE A TRAFFIC FLOW HAS ENDED – EITHER MARKED BY THE NORMAL TCP-FLOW TERMINATION PROCESS OR BY A TIME-OUT, THE SET OF FLOW DESCRIPTORS IS COMPUTED ON THE FLY, AND GETS DIRECTLY IMPORTED INTO DBSTREAM FOR ANALYSIS.

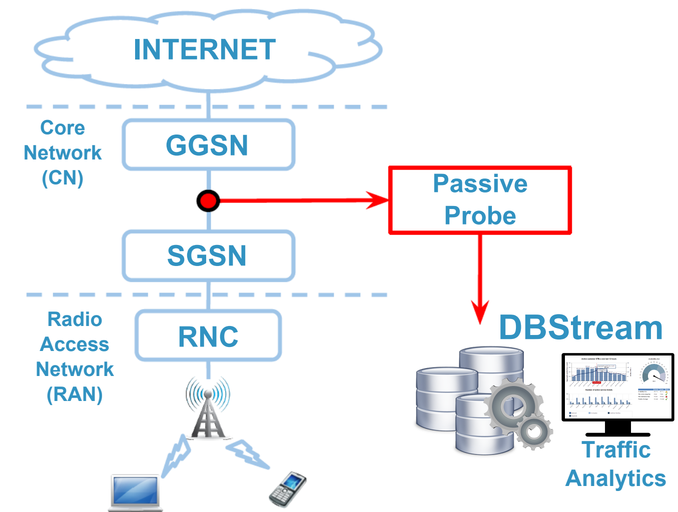


Figure 2. Monitoring framework used in the analysis of core-cellular measurements.

Autonomous System	% bytes	% flows
LO	69.3	66.7
15169 (Google)	30	32.7

Table II. FRACTION OF BYTES AND FLOWS PER AS HOSTING YOUTUBE.

Country	% hosted volume
Europe (generic)	46.8%
Local country	37.2%
Ireland	12.7%
Neighbor country	2.1%
United States	1.1%
Unclassified	0.1%

Table III. TOP FACEBOOK HOSTING COUNTRIES BY VOLUME.

and long duration (especially due to the control messages), media transfers are transmitted in short and heavy flows.

To study the hosting infrastructures of the three services, we complement the traffic datasets with the name of the organization and the Autonomous System (AS) hosting the content, extracted from the MaxMind GeoCity databases[3].

To sum-up, after following the aforementioned process, we obtain – for each newly observed traffic flow and in an on-line fashion – the set of traffic descriptors reported in Tab. I. Once a traffic flow has ended – either marked by the normal TCP-flow termination process or by a time-out, the set of descriptors is computed on the fly, and gets directly imported into DBStream for analysis. Based on these flow descriptors, we can study the content location, the traffic characteristics, the content delivery dynamics, as well as the service delivery performance for the addressed services.

V. CONTENT DELIVERY INFRASTRUCTURE

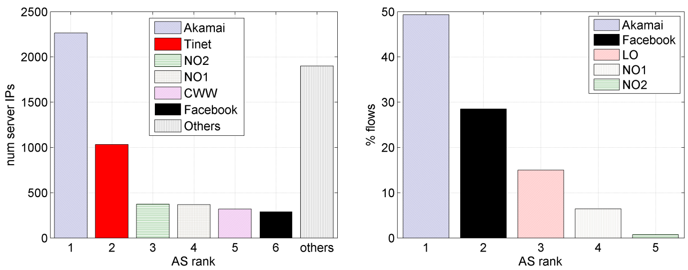
We start by characterizing the YouTube traffic, with a special focus on its underlying hosting/delivery infrastructure. As shown in table IV, which reports the number of unique server IPs hosting YouTube traffic, YouTube IPs are distributed among several Google ASes as well as hosted by servers at the local operator (LO from now on). The latter suggests the usage of content caching. However, we can not say whether these IPs correspond to content caching performed by the LO and/or to Google servers deployed inside the ISP, which is a common approach followed by Google to improve end-user experience, known as Google Global Cache (GGC)[4]. Table II reports the fraction of bytes and flows served per AS hosting YouTube, confirming that the LO plays a key role in the distribution of YouTube traffic for the monitored cellular network. In addition, the remaining traffic comes mainly from only one of the Google ASes (AS 15169). A deeper analysis reveals that servers provisioning content from this AS to the monitored networks are located in Europe.

Due to the high number of daily users and the high volumes of served traffic, Facebook uses a sophisticated content delivery infrastructure. Indeed, we observed more than 6500 server IPs hosting Facebook contents in our traces, distributed across 20 countries and more than 260 different ASes. This confirms the wide-spread presence of several organizations hosting Facebook contents. Fig. 3 shows the main organizations/ASes hosting Facebook content, both in terms of number of unique server IPs and share of delivered flows. Akamai is clearly the key player in terms of Facebook content hosting, delivering almost 50% of the flows in our traces, using more than 2260 different server IPs. Interesting enough is the large number of server IPs observed from two organizations which actually deliver a negligible share of the flows: the Tiscali International Network (Tinet) and Cable & Wireless Worldwide (CWW). We believe these organizations are only caching spurious Facebook contents. In the remainder of the study we focus on the top 5 organizations/ASes in terms of served flows, depicted in Fig. 3(b): Akamai, Facebook AS, the Local Operator (LO) which hosts the vantage point, and two Neighbor Operators, NO1 and NO2.

In the case of WhatsApp, we observed a total of 386 unique

[3]MaxMIND GeoIP Databases, http://www.maxmind.com.

[4]https://peering.google.com/about/ggc.html



(a) Server IPs per AS.

(b) Share of flows hosted per AS.

Figure 3. (a) Unique server IPs used by the top organizations/ASes hosting Facebook and (b) flow shares per hosting AS. Akamai is clearly the key player in terms of Facebook content delivery.

server IPs hosting the service, belonging to a single AS called SoftLayer (AS number 36351)[5]. To avoid biased conclusions about the set of identified IPs from a single vantage point, we performed an active measurements campaign using the RIPE Atlas measurement network[6], where we analyzed which IPs were obtained when resolving the same FQDNs from 600 different boxes distributed around the globe during multiple days. These active measurements confirmed that the same set of IPs is always replied, regardless of the geographical location of the requester. SoftLayer is a US-based cloud infrastructure provider, consisting of 13 data centers and 17 Points of Presence (PoPs) distributed worldwide.

A. Geographical Diversity of Content Hosting Servers

Tab. III provides an overview of the geographical diversity of the Facebook hosting infrastructure, listing the top countries where servers are located in terms of volume. Servers' location is extracted from the MaxMind GeoCity database, which is highly accurate at the country level [48]. "Europe (generic)" refers to a generic location within Europe for which MaxMind did not return a more accurate information. Almost 99% of the traffic comes from servers and data centers located in Europe, close to our vantage point, while only 1% of the traffic comes from other continents. This is due to three factors: (i) Akamai, the biggest Facebook content provider, has a very geographically distributed presence, pushing contents as close as possible to end-users [6]; (ii) operators heavily employ local content caching, and large CDNs like Akamai tend to deploy servers inside the ISPs' networks, explaining the amount of traffic coming from the local country of the vantage point; (iii) the rest of the traffic is handled directly by Facebook, which has servers split between Ireland (headquarter of Facebook International) and the US.

The WhatsApp hosting infrastructure is completely different. Following the same approach, we observed that despite its geographical distribution, WhatsApp traffic is handled mainly by data centers in Dallas and Houston, being as such a fully centralized US-based service. While this is likely to change in the future after Facebook's WhatsApp acquisition, right now, all messages among users outside the US are routed through the core network, unnecessarily consuming additional network resources and potentially impacting the quality of the service.

[5]SoftLayer: Cloud Servers, http://www.softlayer.com

[6]The RIPE Atlas measurement network, https://atlas.ripe.net/

Service	AS/Organization	# IPs	#/24	#/16
YouTube	All	2030	63	10
	Google AS (15169)	1121	38	2
	YouTube AS (43515)	844	15	2
	LO	35	4	3
Facebook	All	6551	891	498
	Akamai	2264	132	48
	Facebook AS	294	57	5
	LO	26	8	6
	NO1	368	26	14
	NO2	374	33	9
WhatsApp	SoftLayer AS (36351)	386	51	30

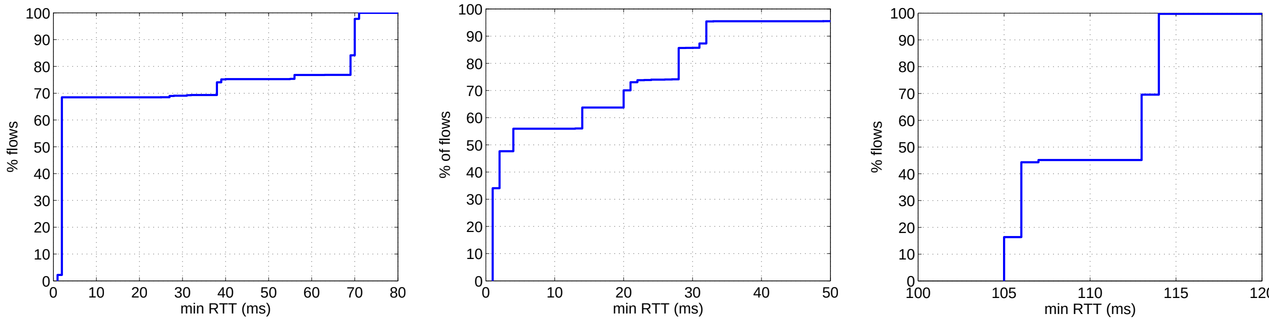
Table IV. Number of IPs and prefixes hosting YouTube, Facebook and WhatsApp. Prefixes are not fully covered/own by the ASes but used for aggregation and counting purposes.

To complement the hosting picture, we investigate the location of the servers from a network topology perspective, considering the distance to the vantage point in terms of Round Trip Time (RTT). The RTT to any specific IP address consists of both the propagation delay and the processing delay. Given a large number of RTT samples to a specific IP address, the minimum RTT values are an approximated measure of the propagation delay, which is directly related to the location of the underlying server. Cellular networks usually employ Performance Enhancement Proxies (PEPs) to speed-up HTTP traffic, and therefore, passive min RTT measurements on top of HTTP traffic provide incorrect results [49]. We therefore consider an active measurement approach, running standard pings from the vantage point to get an estimation of the min RTT to the servers, similar to [50].

Fig. 4 plots the cumulative distribution of the minimum RTT to the server IPs hosting (a) YouTube, (b) Facebook and (c) WhatsApp. Values are weighted by the number of flows served from each IP, to get a better picture of where the traffic is coming from. As a confirmation of the geographical diversity in YouTube, the distribution of min RTT presents some steps or "knees", suggesting the existence of different data centers and/or hosting locations. The largest share of the YouTube flows – almost 70% of them, come from the LO servers, which are located inside the ISP (min RTT $<$ 2 ms). The rest of the flows served from AS 15169 are located at potentially two geographically different locations, one closer at around 38 ms from the vantage point – serving about 5% of the flows, and one farther at about 70 ms – serving the remaining 25% of the flows (i.e., both inside Europe). The largest majority of Facebook flows are served by close serves, located at less than 5 ms from the vantage point. In the case of WhatsApp, the min RTT is always bigger than 100ms, confirming that WhatsApp servers are located outside Europe. Fig. 4(b) shows that the service is evenly handled between two different yet potentially very close locations at about 106 ms and 114 ms, which is compatible with our previous findings of WhatsApp servers located in Dallas and Houston.

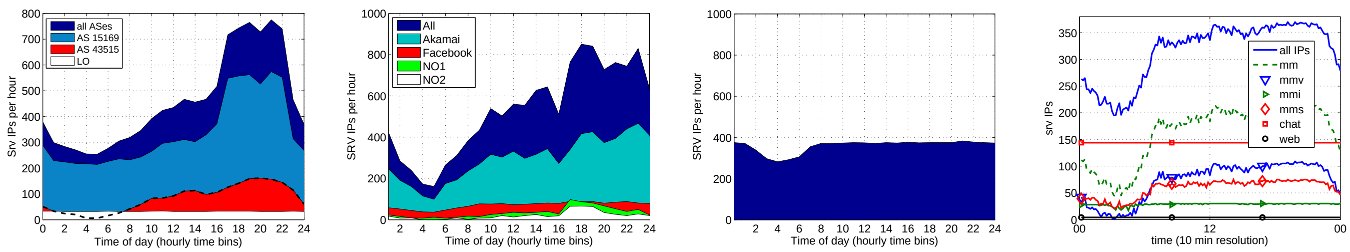
B. IP Address Space of Content Servers

We study now the server diversity through an analysis of the IP address spaces covered by the three services as observed in our traces. Tab. IV summarizes the number of



(a) min RTT YouTube. (b) min RTT Facebook. (c) min RTT WhatsApp.

Figure 4. Distribution of overall min RTT to YouTube, Facebook and WhatsApp server IPs, weighted by the number of flows hosted.



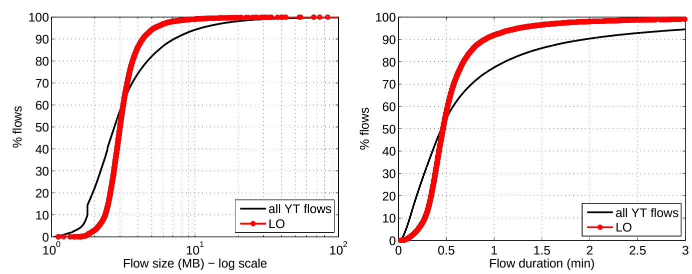
(a) # hourly unique IPs in YouTube. (b) # hourly unique IPs in Facebook. (c) # hourly unique IPs in WhatsApp. (d) # IPs in WhatsApp and content.

Figure 5. Active servers daily hosting YouTube, Facebook and WhatsApp. Server IPs used by WhatsApp are further discriminated by type of content.

unique server IPs hosting YouTube, Facebook and WhatsApp, as well as the /24 and /16 IP blocks or prefixes covered by the corresponding top organizations hosting them.

We observed 2030 different server IPs providing YouTube contents, coming mainly from the LO and AS 15169. Interestingly, there is a big number of IPs observed for AS 43515 (YouTube), but these servers have a negligible contribution volume wise. Akamai and Facebook together account for about 2560 servers scattered around almost 200 /24 IP blocks, revealing again their massively distributed infrastructure. Even if WhatsApp servers are geographically co-located, the range of server IPs handling the content is highly distributed, and consists of 386 unique IPs covering 51 different /24 prefixes. However, only a few of them are actually hosting the majority of the flows, and the same happens for Facebook.

Fig. 5 shows the daily usage of these IPs on a single day, considering the number of unique server IPs per hour. The active IPs serving YouTube from AS 15169 show an abrupt increase at specific times of the day, almost tripling at peak hours. Note that the number of active IPs from the LO is almost constant. The number of active IPs serving Facebook flows from Akamai follows the daily utilization of the network, peaking at the heavy-load time range. Interestingly, the IPs exposed by Facebook AS are constantly active and seem loosely correlated with the network usage. This comes from the fact that Facebook AS servers normally handle all the Facebook dynamic contents [4], which include the user sessions keep-alive. Something similar happens in WhatsApp, where the number of active IPs remains practically constant during the day, due to a similar keep-alive effect. However, if we look a bit closer, we can see some important differences when separately analyzing WhatsApp chat and mm servers.



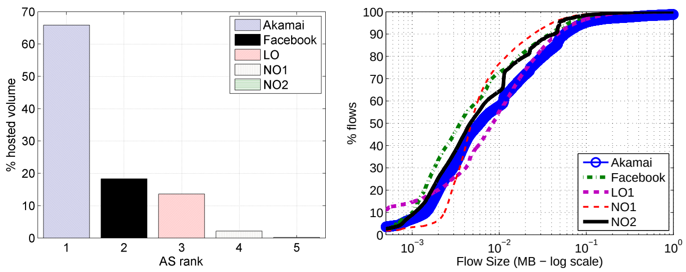
(a) Flows size. (b) Flows duration.

Figure 6. YouTube flow characteristics. Traffic is characterized by short videos, typical from YouTube. The LO serves smaller and shorter videos, suggesting that caching at the edge might also include ads.

Fig. 5(d) shows the dynamics of the active IPs used by WhatsApp on a single day, but using now a finer-grained temporal aggregation of 10 minutes instead of one hour, and discriminating by server type. The mm category is further split into photos/audio (mmi and mms) and video (mmv). Note that no less than 200 IPs are active even in the lowest load hours. When analyzing the active IPs per traffic type, we observe that more than 200 IPs serve WhatsApp mm flows during peak hours. In addition, we see how all the chat servers are constantly active (there are about 150 of them), as they keep the state of active devices to quickly push messages.

VI. NETWORK TRAFFIC ANALYSIS

Let us know focus on the characteristics of the traffic flows carrying YouTube, Facebook and WhatsApp contents. Fig. 6 reports the characteristics of the YouTube flows in terms of size and duration. Traffic is characterized by short videos, in



(a) Shares of hosted volume per org./AS. (b) Distribution of flow sizes.

Figure 7. Hosted volume and distribution of flow sizes per organization.

the order of some MB, and typically shorter than a couple of minutes. The LO serves smaller and shorter videos, suggesting that caching at the edge might also include ads. The long-tailed distributions show that there are also very long videos, which are also sometimes uploaded to YouTube.

Fig. 7 depicts the volume share of Facebook contents hosted by each org./AS, as well as the flow size distributions. Akamai hosts more than 65% of the total volume observed in our traces, followed by Facebook AS itself with about 19%. Comparing the volume shares in Fig. 7(a) with the flow shares in Fig. 3(b) evidences a clear distinction on the content sizes handled by both Akamai and Facebook AS: while Akamai hosts the bigger flows, Facebook AS serves only a small share of the service content. Indeed, as previously flagged by other studies [4], Akamai serves the static contents of the Facebook service (e.g., photos, songs, videos, etc.), whereas the Facebook AS covers almost exclusively the dynamic contents (e.g., chats, tags, session information, etc.).

To further explore this distinction, Fig. 7(b) reports the distribution of the flow sizes served per organization. The CDF reveals that Akamai clearly serves bigger flows than Facebook AS. The remaining ASes tend to host bigger flows than Facebook AS, which is coherent with the fact that ISPs caching is generally done for bigger objects, aiming at reduce the load on the core network.

In terms of WhatsApp traffic, Fig. 8 reports the characteristics of the corresponding flows in terms of size and duration. Fig. 8(a) shows a scatter plot reporting the flow duration vs. the flow size, discriminating by chat and mm flows. Whereas mm messages are sent over dedicated connections, resulting in short-lived flows, text messages are sent over the same connection used for control data, resulting in much longer flows. For example, some chat flows are active for as much as 62 hours. Fig. 8(b) indicates that more than 50% of the mm flows are bigger than 70 KB, with an average flow size of 225 KB. More than 90% of the chat flows are smaller than 10 KB, with an average size of 6.7 KB. In terms of duration, Fig. 8(c) shows that more than 90% of the mm flows last less than 1 min (mean duration of 1.8 min), whereas chat flows last on average as much as 17 minutes. The flow duration distribution additionally reveals some clear steps at exactly 10, 15 and 24 minutes, suggesting the usage of an application time-out to terminate long idle connections. This behavior is actually dictated by the operating system of the device [24].

features	chat	mm	mmv	mmi	mms
# $\text{bytes}_{\text{down}}$	16.6%	83.0%	38.8%	12.8%	29.8%
# bytes_{up}	29.5%	70.2%	35.2%	15.0%	17.9%
# flows	93.4%	6.2%	0.3%	2.9%	2.9%
$\frac{\#\ \text{bytes}_{\text{down}}}{\#\ \text{bytes}_{\text{down+up}}}$	60.6%	76.3%	75.1%	70.0%	81.9%

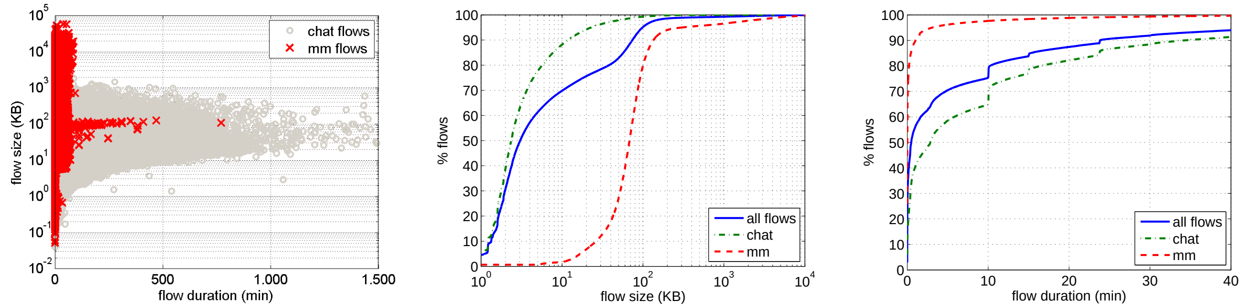
Table V. VOLUME AND FLOWS PER TRAFFIC CATEGORY.

VII. CONTENT DELIVERY DYNAMICS

The characterization performed in previous sections mainly considers the static characteristics of the traffic delivery. In this section we focus on the temporal dynamics of the content delivery. Fig. 10 reports the dynamics of the traffic provisioning from the main ASes in terms of flow counts. The number of downloaded flows follows a standard daily usage dictated by the number of customers in the network, with a marked peak-hour effect, between 18:00 and 22:00. Interestingly, the relative increase in the number of served flows at peak time from the LO is stronger than for the Google AS, suggesting a potentially different provisioning police.

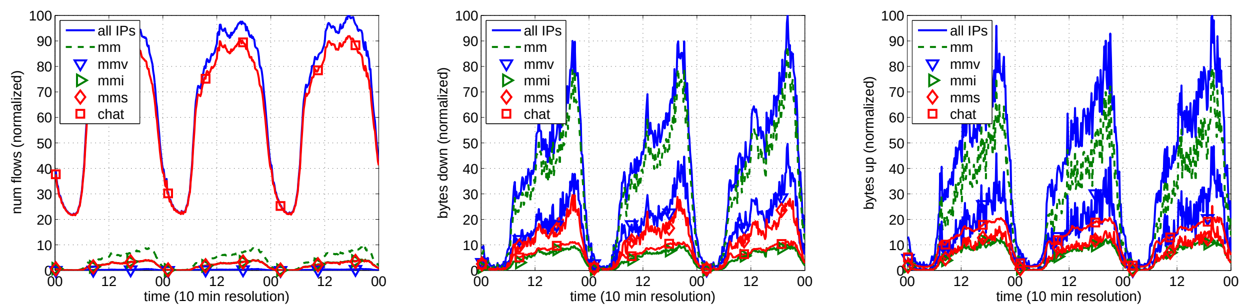
Fig. 9 shows the dynamics of WhatsApp for three consecutive days, including the fraction of flows and traffic volume shares, discriminating by chat and mm traffic. Fig. 9(a) shows the flow count shares, revealing how chat flows are clearly dominating. Once again we stop in the mmi and mms servers, which seem to always handle the same share of flows, suggesting that both space names are used as a mean to balance the load in terms of photos and audio messages. Finally, Figs. 9(b) and 9(c) reveal that even if the mm volume is higher than the chat volume, the latter is comparable to the photos and audio messaging volume, especially in the uplink. Tab. V summarizes these shares of flows and traffic volume.

Given that the content delivery infrastructure of Facebook is particularly interesting in terms of geographical distribution, we study now the temporal evolution of the servers selected for provisioning the Facebook flows. To begin with, we focus on the temporal evolution of the min RTT, as reported in Fig. 4. Fig. 11(a) depicts the temporal variation of the CDF for all the Facebook flows and for a complete day, considering a single CDF every three hours period. The CDFs are rather stable during the day, but present some slight variations during the night and early morning. To get a better picture of such dynamics, Fig. 11(b) depicts the hourly evolution of the min RTT for all the Facebook flows during 3 consecutive days, being the first day the one analyzed in Fig. 11(a). Each column in the Fig. depicts the PDF of the min RTT for all the served flows, using a heat map-like plot (i.e., the darker the color, the more concentrated the PDF in that value). The flagged variations are observed during the first day, with some slight shifts between 6am and 12am from servers at 14ms and 20ms. The heat map also reveals some periodic flow shifts between 9pm and midnight from servers at 20ms, but impacting a small fraction of flows. Fig. 11(c) presents the same type of heat map for Facebook flows, but considering a dataset of 2012 from the same vantage point [50]. The temporal patterns in 2012 show a much stronger periodic load balancing cycle, focused in a small number of hosting regions at 7ms, 14ms, and 37ms. Comparing the results from 2012 with those in 2013



(a) Flow duration vs. size. (b) Flow size. (c) Flow duration.

Figure 8. Characterization of WhatsApp flows. Whereas mm messages are sent over short-lived flows, text messages result in longer and much smaller flows.



(a) Flows. (b) Bytes down. (c) Bytes up.

Figure 9. WhatsApp traffic dynamics. mmi and mms servers constantly handle the same share of flows, suggesting that both space names are used as a means to balance the load in terms of photos and audio messages.

suggests that Facebook content delivery is becoming more spread in terms of hosting locations, and load balancing cycles are becoming a-priori less marked. However, when deeply analyzing the complete dataset of 2013, conclusions are rather different.

To drill down deeply into this issue, we analyze the dynamics of the content delivery for the complete Facebook dataset, spanning 28 consecutive days. Instead of considering the variations of the min RTT, we consider now the variations on the number of flows served by the observed IPs. Changes in the distribution of the number of flows coming from the complete set of 6551 server IPs reflect variations in the way content is accessed and served from the hosting infrastructure observed in our traces. For this analysis, we consider a time granularity of one hour, and therefore compute the distribution of the number of flows provided per server IP in consecutive time slots of one hour, for the complete 28 days. This results in a time-series with a total of 24 $\times$ 28 = 672 consecutive distributions. To quantify how different are two distributions in the resulting time-series, we use a symmetric and normalized version of the Kullback-Leibler divergence described in [51].

To visualize the results of the comparison for the complete time span of 28 days, we use a graphical tool proposed in [51], referred to as *Temporal Similarity Plot* (TSP). The TSP allows pointing out the presence of temporal patterns and (ir)regularities in distribution time-series by graphical inspection. In a nutshell, a TSP is a symmetrical, checker-board, heat-map like plot, in which the value $\{i, j\}$ reflects how similar are the two distributions at time t_i and t_j. Similarity between both distributions is computed on the basis of an extension of the well know Kullback-Leibler (KL) divergence. Using TSP plots provides a powerful approach to graphically inspect the evolution of certain traffic feature, by reflecting, for each pair of time-bins, how similar the empirical distributions of this feature are. Let us take an example to better understand a TSP plot. Fig. 12 presents different TSPs for the distributions of all the Facebook flows across all the server IP addresses providing Facebook content, over a period of 28 days. Each plot is a matrix of 672 $\times$ 672 pixels; the color of each pixel $\{i, j\}$ shows how similar are the two distributions at times t_i and t_j: black represents low similarity, whereas white corresponds to high similarity. By construction, the TSP is symmetric around the 45° diagonal, and it can be interpreted either by columns or by rows. For example, if we read the TSP by rows, for every value j in the y-axis, the points to the left [right] of the diagonal represent the degree of similarity to past [future] distributions. We refer the interested reader to [51] for a detailed description of the TSP tool.

The three TSPs in Fig. 12 represent the distribution variations for (a) all the observed IPs, (b) the Akamai IPs and (c) the Facebook AS IPs. Let us begin by the TSP for all the observed server IPs in Fig. 12(a). The regular "tile-wise" texture within periods of 24 hours evidences the presence of daily cycles, in which similar IPs are used to serve a similar number of flows. The lighter zones in these 24 hour periods correspond to the time of the day, whereas the dark zones correspond to the night-time periods when the traffic load is low. The low similarity (dark areas) at night (2am-5am) is caused by the low number of served flows, which induces larger statistical

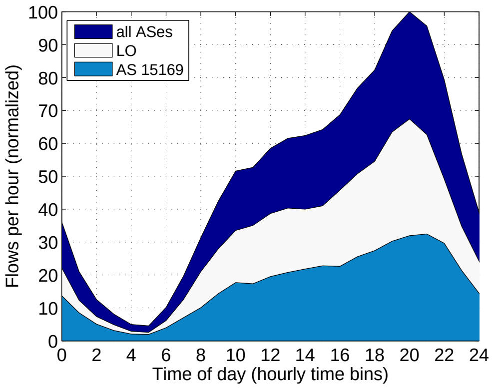


Figure 10. YouTube flow counts per hour and per AS. The number of downloaded flows follows a standard daily usage pattern.

fluctuations in the computed distributions. This pattern repeats almost identical for few days, forming multiple macro-blocks around the main diagonal of size ranging from 2 up to 6 days. This suggests that during these periods, the same sets of IPs are used to deliver the flows, with slight variations during the night periods, similarly to what we observed in Fig. 11(a). However, the analysis of the entire month reveals the presence of a more complex temporal strategy in the (re)usage of the IP address space. For example, there is a reuse of (almost) the same address range between days 10-12 and days 15-16. Interestingly, we observe a sharp discontinuity on days 18-19, as from there on, all the pixels are black (i.e., all the distributions are different from the past ones).

To get a better understanding of such behaviors, Figs. 12(b) and 12(c) split the analysis for Akamai and Facebook AS IPs only. The Figs. reveal a different (re)usage policy of the IPs hosting the contents. In particular, Akamai uses the same servers for 4 to 7 days (see multi-days blocks around the main diagonal). When it changes the used addresses, the shift is not complete as we can observe the macro-blocks slowly fading out over time. This suggests a rotation policy of the address space of Akamai, on a time-scale of weeks. On the other hand, Facebook AS does not reveal such a clear temporal allocation policy. It alternates periods of high stability (e.g. between days 4 and 10) with highly dynamic periods (e.g., from day 18 onward). It is interesting noticing that Facebook AS is the responsible for the abrupt change in the distributions observed from the 18th day on, in the TSP of the overall traffic.

Service Performance – Flow Throughput

To conclude with the analysis, we focus now on the content delivery performance, measured in terms of flow throughput. Throughout and latency are today the de-facto and mostly analyzed Quality of Service (QoS) Key Performance Indicators (KPIs) to understand the performance of end-user services, especially from a QoE perspective [52]. Other QoS KPIs such as packet loss have a direct impact on throughput, in particular for the kind of traffic we are dealing with, i.e., TCP traffic, thus we do not consider them directly. We do not consider the case of Facebook as we were not able to reliably compute the flow throughput information in this case.

Fig. 13(a) reports the distribution of the average download throughput per YouTube flow, discriminating by hosting AS. The download throughput is generally considered as the main network performance indicator that dictates the experience of a user watching YouTube videos. When analyzing the performance results per AS, it is evident that the YouTube flows served by the LO are the ones achieving the highest performance, with an average flow download throughput of 2.7 Mbps. This out-performance evidences the benefits of local content caching and low-latency servers for provisioning the YouTube flows.

Considering WhatsApp flow throughput, Fig. 13(b) depicts the uplink and downlink (we consider both directions as WhatsApp is a symmetrical service) throughputs for flows bigger than 1 MB. This filtering is performed as a means to improve the throughput estimations. A-priori, one might expect that the long RTTs involved in the communications to the US servers might heavily impact the achieved performance. This is confirmed for about 30% of the transmitted flows, which achieve a throughput smaller than 250 kbps. However, higher throughputs are obtained for the largest shares of flows, achieving an average per flow downlink/uplink throughput of 1.5 Mbps/800 kbps.

VIII. Discussion and Implications of Results

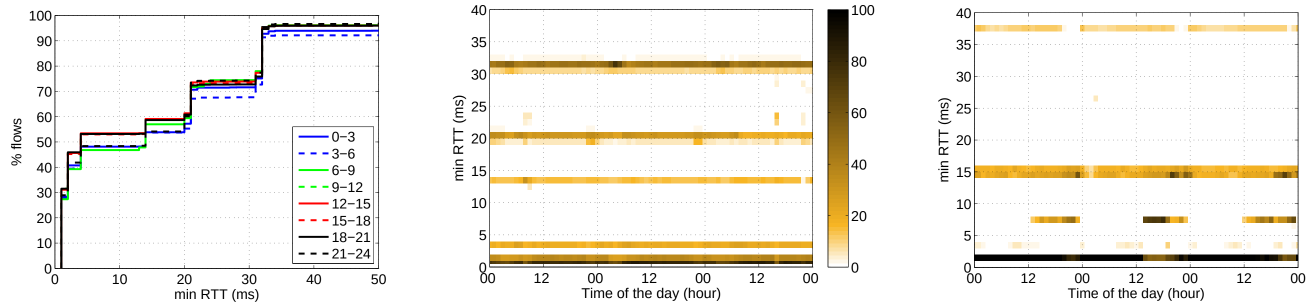
Let us now focus on the interpretation of the findings presented so far. In this section we provide a comprehensive discussion of the main take aways of the study, and particularly elaborate on their implications for network dimensioning, operation and management tasks. Discussion is structured along five specific topics covering the contributions flagged in Sec. I: (i) geographical location of servers and contents; (ii) dynamics of the content delivery; (iii) content caching at the edge; (iv) traffic characteristics; (v) usage dynamics.

A. Servers Geolocation

Finding: our study reveals that even if the three studied services are very popular worldwide, their networking hosting infrastructures follow very different paradigms: based on Akamai's and Google's pervasiveness, both Facebook and YouTube are hosted by a highly distributed network architecture, whereas WhatsApp follows a fully centralized hosting architecture at cloud servers exclusively located in the US, independently of the geographical location of the users.

Implications: the first direct implication is in terms of service performance. WhatsApp flows suffer an important additional latency for users outside the US, which might impact their QoE. Being Brazil, India, Mexico and Russia the fastest growing countries in terms of users[7], such a centralized hosting infrastructure is likely to become a problematic bottleneck in the near future. On the contrary, YouTube and Facebook latency due to propagation is highly reduced by local caching and regional servers' location, enabling the usage of latency-sensitive applications on top of these service (e.g., video conversations, cloud gaming, or interactive video editing). The second implication is in terms of traffic management. The SoftLayer servers identified in the study are exclusively used by WhatsApp, making it very simple for an ISP to identify WhatsApp flows by server IP address, similarly to [50]. While

[7] WhatsApp Blog, http://blog.whatsapp.com/

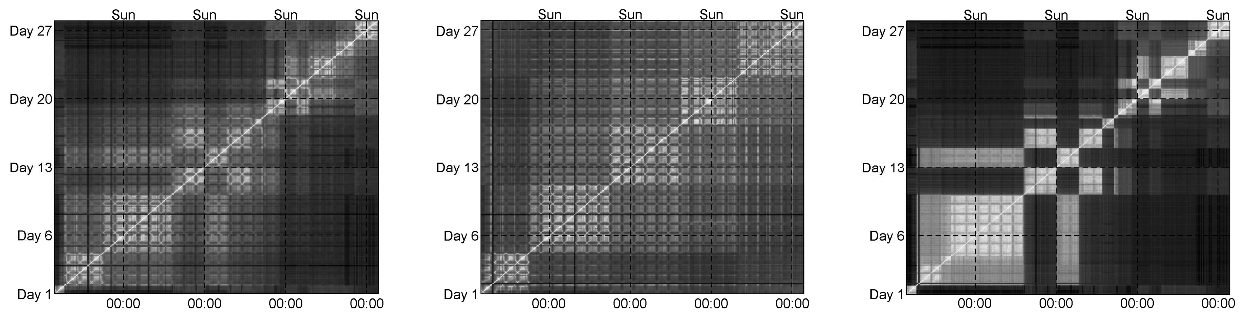


(a) min RTT daily dynamics in 2013.

(b) min RTT dynamics in 2013

(c) min RTT dynamics in 2012

Figure 11. Temporal variations of the min RTT to Facebook servers. In the heat maps of Figs. (a) and (c), the darker the color, the bigger the fraction of flows served from the corresponding min RTT value.

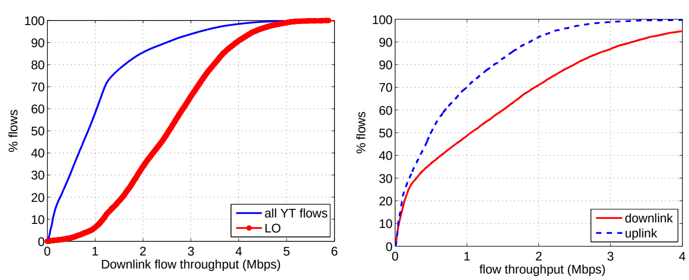


(a) All IPs hosting Facebook

(b) Akamai

(c) Facebook AS

Figure 12. TSP of hourly flow count distributions over 28 days for all the observed IPs hosting Facebook, Akamai IPs, and Facebook AS IPs. A black pixel at $\{i, j\}$ means that the distributions at times t_i and t_j are very different, whereas a white pixel corresponds to high similarity.



(a) YouTube flow throughput.

(b) WhatsApp flow throughput.

Figure 13. YouTube and WhatsApp flow performance. Flows served by the LISP are the ones achieving the highest download throughput, evidencing the benefits of local content caching and low-latency servers.

we do not expect it to happen, a cellular ISP might intentionally degrade the performance of WhatsApp flows to discourage its usage, similarly to what IPSs have done in the past with Skype traffic[8]. The final implication is about data privacy. The geo-location of servers makes users traffic to be hosted in countries other than their local ones, thus data locality is not maintained (in the case of WhatsApp, this is 100% confirmed). In the light of the ever increasing concerns related to privacy and data security, such a geographical distribution might even cause legal jurisdiction issues due to different data privacy protection laws in different countries.

[8]"Comcast Blocks Some Internet Traffic", the Whashington Post, 2007.

B. *Content Delivery Dynamics*

Finding: the highly dynamic and distributed content delivery mechanisms used by Facebook are becoming more spread in terms of hosting locations and hosting organizations.

Implications: this makes of Facebook a very challenging source of traffic for network dimensioning and traffic engineering. Indeed, it is very difficult for an ISP to properly engineer its own network when surges of traffic come from potentially multiple ingress nodes at different times of day. A proper traffic engineering policy must therefore be dynamic as well, to cope with such traffic delivery behavior. Delivery dynamics might even have an impact on the transport costs faced by the ISP providing the Internet access; as we show in [21], traffic being served from other neighboring ISPs for which uni-directional peering agreements have been established results in extra costs for the local ISP.

C. *Content Caching*

Finding: there is a widespread usage of local content caching in YouTube mobile.

Implications: local content caching/hosting by servers at the edge of the ISP provides high benefits in terms of flow throughput, ultimately leading to an improved QoE and user engagement.

D. Traffic Characteristics

Finding: WhatsApp is not only about text-messaging, but more than 75% of its traffic corresponds to multimedia file sharing, both in the uplink and downlink directions.

Implications: the growing popularity of WhatsApp in cellular networks might cause a serious performance issue for ISPs, especially in the uplink direction, where resources are scarcer. On the other hand, given that multimedia contents are static and that many of them are shared multiple times among WhatsApp groups, the usage of local caching techniques might result in important savings and performance enhancement.

E. Usage Patters

Finding: traffic volumes and flows in the three services follow a very predictable time-of-day pattern, commonly observed in user-generated traffic.

Implications: even if not simple to achieve, this type of patterns suggest that an ISP might better optimize the resources of the access network through time-based traffic engineering mechanisms, dynamically adjusting network resources based on load predictions.

IX. Concluding Remarks

In this paper we presented a characterization of the networking aspects of three highly popular, Internet-scale services in cellular networks: YouTube, Facebook and WhatsApp. Through the analysis of large-scale traffic collected at the cellular network of a major European ISP and analyzed with a tailored big data-based platform for NTMA, we dissected and compared the networking behavior of these services, considering not only the traffic flows but also the network infrastructures hosting them. We showed that while YouTube's and Facebook's contents are hosted in multiple geographical locations and are provisioned through highly dynamic addressing mechanisms, the WhatsApp hosting infrastructure is fully centralized at cloud servers exclusively located in the US, independently of the geographical location of the users.

The YouTube analysis evidenced a widespread utilization of content caching at the edge of the monitored cellular network, which proved to be highly efficient and performant in terms of downlink video flow throughput. The Facebook analysis revealed a very structured yet tangled architecture hosting the service, mainly due to the pervasiveness and distributed nature of Akamai, its hosting CDN. We have fully dissected the nicely structured internal naming scheme used by WhatsApp to handle the different types of connections, which shall enable an easy way to monitor its traffic in the network.

In addition, we have described DBStream, a flexible and scalable BDA platform for network traffic monitoring and analysis, which proved to be highly useful in the performed analysis as presented in this paper. We have also made DBStream open source for this paper, which we expect would be highly useful for the network management community in the years to come.

We believe that the characterization provided in this paper offers a sound basis to cellular network operators to understand the traffic dynamics behind popular services, enabling a better traffic engineering and network management for such applications.

Acknowledgments

The research leading to these results has received funding from the Vienna Science and Technology Fund (WWTF) through project ICT15-129, "BigDAMA", and from the European Union under the FP7 Grant Agreement n. 318627, "mPlane" project. The work has been partially funded by the Austrian Government and the City of Vienna through the program COMET.

References

[1] F. Ricciato, "Traffic monitoring and analysis for the optimization of a 3g network," *IEEE Wireless Communications*, vol. 13, no. 6, pp. 42–49, Dec 2006.

[2] L. Golab, T. Johnson, J. S. Seidel, and V. Shkapenyuk, "Stream warehousing with datadepot," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '09. New York, NY, USA: ACM, 2009, pp. 847–854. [Online]. Available: http://doi.acm.org/10.1145/1559845.1559934

[3] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," in *Proceedings of the ACM SIGCOMM 2010 Conference*, ser. SIGCOMM '10. New York, NY, USA: ACM, 2010, pp. 75–86. [Online]. Available: http://doi.acm.org/10.1145/1851182.1851194

[4] V. Gehlen, A. Finamore, M. Mellia, and M. M. Munafò, *Traffic Monitoring and Analysis: 4th International Workshop, TMA 2012, Vienna, Austria, March 12, 2012. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, ch. Uncovering the Big Players of the Web, pp. 15–28. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-28534-9_2

[5] R. Krishnan, H. V. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao, "Moving beyond end-to-end path information to optimize cdn performance," in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC '09. New York, NY, USA: ACM, 2009, pp. 190–201. [Online]. Available: http://doi.acm.org/10.1145/1644893.1644917

[6] E. Nygren, R. K. Sitaraman, and J. Sun, "The akamai network: A platform for high-performance internet applications," *SIGOPS Oper. Syst. Rev.*, vol. 44, no. 3, pp. 2–19, Aug. 2010. [Online]. Available: http://doi.acm.org/10.1145/1842733.1842736

[7] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: A view from the edge," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 15–28. [Online]. Available: http://doi.acm.org/10.1145/1298306.1298310

[8] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo, and S. Rao, "Dissecting video server selection strategies in the youtube cdn," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, June 2011, pp. 248–257.

[9] A. Finamore, M. Mellia, M. M. Munafò, R. Torres, and S. G. Rao, "Youtube everywhere: Impact of device and infrastructure synergies on user experience," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC '11. New York, NY, USA: ACM, 2011, pp. 345–360. [Online]. Available: http://doi.acm.org/10.1145/2068816.2068849

[10] P. Casas, M. Seufert, and R. Schatz, "Youqmon: A system for on-line monitoring of youtube qoe in operational 3g networks," *SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 2, pp. 44–46, Aug. 2013. [Online]. Available: http://doi.acm.org/10.1145/2518025.2518033

[11] J. J. Ramos-munoz, J. Prados-Garzon, P. Ameigeiras, J. Navarro-Ortiz, and J. M. Lopez-soler, "Characteristics of mobile youtube traffic," *IEEE Wireless Communications*, vol. 21, no. 1, pp. 18–25, February 2014.

[12] M. Z. Shafiq, L. Ji, A. X. Liu, and J. Wang, "Characterizing and modeling internet traffic dynamics of cellular devices," in *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '11. New York, NY, USA: ACM, 2011, pp. 305–316. [Online]. Available: http://doi.acm.org/10.1145/1993744.1993776

[13] F. Ben Abdesslem and A. Lindgren, "Cacheability of youtube videos in cellular networks," in *Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications, & Challenges*, ser. AllThingsCellular '14. New York, NY, USA: ACM, 2014, pp. 53–58. [Online]. Available: http://doi.acm.org/10.1145/2627585.2627588

[14] Y. Zhang and A. Arvidsson, "Understanding the characteristics of cellular data traffic," in *Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design*, ser. CellNet '12. New York, NY, USA: ACM, 2012, pp. 13–18. [Online]. Available: http://doi.acm.org/10.1145/2342468.2342472

[15] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 29–42. [Online]. Available: http://doi.acm.org/10.1145/1298306.1298311

[16] Y. Liu, K. P. Gummadi, B. Krishnamurthy, and A. Mislove, "Analyzing facebook privacy settings: User expectations vs. reality," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC '11. New York, NY, USA: ACM, 2011, pp. 61–70. [Online]. Available: http://doi.acm.org/10.1145/2068816.2068823

[17] G. Magno, G. Comarela, D. Saez-Trumper, M. Cha, and V. Almeida, "New kid on the block: Exploring the google+ social graph," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC '12. New York, NY, USA: ACM, 2012, pp. 159–170. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398794

[18] R. Gonzalez, R. Cuevas, R. Motamedi, R. Rejaie, and A. Cuevas, "Google+ or google-?: Dissecting the evolution of the new osn in its first year," in *Proceedings of the 22Nd International Conference on World Wide Web*, ser. WWW '13. New York, NY, USA: ACM, 2013, pp. 483–494. [Online]. Available: http://doi.acm.org/10.1145/2488388.2488431

[19] X. Zhao, A. Sala, C. Wilson, X. Wang, S. Gaito, H. Zheng, and B. Y. Zhao, "Multi-scale dynamics in a massive online social network," *CoRR*, vol. abs/1205.4013, 2012. [Online]. Available: http://arxiv.org/abs/1205.4013

[20] M. Allamanis, S. Scellato, and C. Mascolo, "Evolution of a location-based online social network: Analysis and models," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC '12. New York, NY, USA: ACM, 2012, pp. 145–158. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398793

[21] P. Fiadino, A. D'Alconzo, and P. Casas, "Characterizing web services provisioning via cdns: The case of facebook," in *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Aug 2014, pp. 310–315.

[22] E. J. Vergara, S. Andersson, and S. Nadjm-Tehrani, "When mice consume like elephants: Instant messaging applications," in *Proceedings of the 5th International Conference on Future Energy Systems*, ser. e-Energy '14. New York, NY, USA: ACM, 2014, pp. 97–107. [Online]. Available: http://doi.acm.org/10.1145/2602044.2602054

[23] A. Aucinas, N. Vallina-Rodriguez, Y. Grunenberger, V. Erramilli, K. Papagiannaki, J. Crowcroft, and D. Wetherall, "Staying online while mobile: The hidden costs," in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '13. New York, NY, USA: ACM, 2013, pp. 315–320. [Online]. Available: http://doi.acm.org/10.1145/2535372.2535408

[24] P. Fiadino, M. Schiavone, and P. Casas, "Vivisecting whatsapp through large-scale measurements in mobile networks," in *Proceedings of the 2014 ACM Conference on SIGCOMM*, ser. SIGCOMM '14. New York, NY, USA: ACM, 2014, pp. 133–134. [Online]. Available: http://doi.acm.org/10.1145/2619239.2631461

[25] ——, *Traffic Monitoring and Analysis: 7th International Workshop, TMA 2015, Barcelona, Spain, April 21-24, 2015. Proceedings*. Cham: Springer International Publishing, 2015, ch. Vivisecting WhatsApp in Cellular Networks: Servers, Flows, and Quality of Experience, pp. 49–63. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-17172-2_4

[26] M. Stonebraker, "Sql databases v. nosql databases," *Commun. ACM*, vol. 53, no. 4, pp. 10–11, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721659

[27] C. Cranor, T. Johnson, O. Spataschek, and V. Shkapenyuk, "Gigascope: A stream database for network applications," in *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '03. New York, NY, USA: ACM, 2003, pp. 647–651. [Online]. Available: http://doi.acm.org/10.1145/872757.872838

[28] D. J. Abadi, D. Carney, U. Çetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik, "Aurora: A new model and architecture for data stream management," *The VLDB Journal*, vol. 12, no. 2, pp. 120–139, Aug. 2003. [Online]. Available: http://dx.doi.org/10.1007/s00778-003-0095-z

[29] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008. [Online]. Available: http://doi.acm.org/10.1145/1327452.1327492

[30] T. White, *Hadoop: The Definitive Guide*, 1st ed. O'Reilly Media, Inc., 2009.

[31] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, ser. HotCloud'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 10–10. [Online]. Available: http://dl.acm.org/citation.cfm?id=1863103.1863113

[32] M. Zaharia, T. Das, H. Li, S. Shenker, and I. Stoica, "Discretized streams: An efficient and fault-tolerant model for stream processing on large clusters," in *Proceedings of the 4th USENIX Conference on Hot Topics in Cloud Ccomputing*, ser. HotCloud'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 10–10. [Online]. Available: http://dl.acm.org/citation.cfm?id=2342763.2342773

[33] P. Bhatotia, A. Wieder, R. Rodrigues, U. A. Acar, and R. Pasquin, "Incoop: Mapreduce for incremental computations," in *Proceedings of the 2Nd ACM Symposium on Cloud Computing*, ser. SOCC '11. New York, NY, USA: ACM, 2011, pp. 7:1–7:14. [Online]. Available: http://doi.acm.org/10.1145/2038916.2038923

[34] W. Lam, L. Liu, S. Prasad, A. Rajaraman, Z. Vacheri, and A. Doan, "Muppet: Mapreduce-style processing of fast data," *Proc. VLDB Endow.*, vol. 5, no. 12, pp. 1814–1825, Aug. 2012. [Online]. Available: http://dx.doi.org/10.14778/2367502.2367520

[35] B. Li, E. Mazur, Y. Diao, A. McGregor, and P. Shenoy, "Scalla: A platform for scalable one-pass analytics using mapreduce," *ACM Trans. Database Syst.*, vol. 37, no. 4, pp. 27:1–27:43, Dec. 2012. [Online]. Available: http://doi.acm.org/10.1145/2389241.2389246

[36] A. Baer, A. Finamore, P. Casas, L. Golab, and M. Mellia, "Large-scale network traffic monitoring with dbstream, a system for rolling big data analysis," in *Big Data (Big Data), 2014 IEEE International Conference on*, Oct 2014, pp. 165–170.

[37] Maarten Wullink, Giovane C. M. Moura, Muller, M, and Cristian Hesselman, "ENTRADA: a High Performance Network Traffic Data Streaming Warehouse," in *Network Operations and Management Symposium (NOMS), 2016 IEEE (to appear)*, April 2016.

[38] S. Melnik, A. Gubarev, J. J. Long, G. Romer, S. Shivakumar, M. Tolton, and T. Vassilakis, "Dremel: Interactive analysis of web-scale datasets," *Proc. VLDB Endow.*, vol. 3, no. 1-2, pp. 330–339, Sep. 2010. [Online]. Available: http://dx.doi.org/10.14778/1920841.1920886

[39] R. Fontugne, J. Mazel, and K. Fukuda, "Hashdoop: A mapreduce framework for network anomaly detection," in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, April 2014, pp. 494–499.

[40] Y. Lee and Y. Lee, "Toward scalable internet traffic measurement and analysis with hadoop," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 1, pp. 5–13, Jan. 2012. [Online]. Available: http://doi.acm.org/10.1145/2427036.2427038

[41] J. Liu, F. Liu, and N. Ansari, "Monitoring and analyzing big traffic data of a large-scale cellular network with hadoop," *IEEE Network*, vol. 28, no. 4, pp. 32–39, July 2014.

[42] P. Fiadino, P. Casas, M. Schiavone, and A. D'Alconzo, "Online social networks anatomy: On the analysis of facebook and whatsapp in cellular

networks," in *IFIP Networking Conference (IFIP Networking), 2015*, May 2015, pp. 1–9.

[43] P. Casas, P. Fiadino, A. Sackl, and A. D'Alconzo, "Youtube in the move: Understanding the performance of youtube in cellular networks," in *2014 IFIP Wireless Days (WD)*, Nov 2014, pp. 1–6.

[44] A. Baer, P. Casas, L. Golab, and A. Finamore, "Dbstream: An online aggregation, filtering and processing system for network traffic monitoring," in *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Aug 2014, pp. 611–616.

[45] A. Arasu, S. Babu, and J. Widom, "The cql continuous query language: Semantic foundations and query execution," *The VLDB Journal*, vol. 15, no. 2, pp. 121–142, Jun. 2006. [Online]. Available: http://dx.doi.org/10.1007/s00778-004-0147-z

[46] P. Fiadino, A. Bar, and P. Casas, "Httptag: a flexible on-line http classification system for operational 3g networks," in *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, April 2013, pp. 71–72.

[47] I. N. Bermudez, M. Mellia, M. M. Munafo, R. Keralapura, and A. Nucci, "Dns to the rescue: Discerning content and services in a tangled web," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC '12. New York, NY, USA: ACM, 2012, pp. 413–426. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398819

[48] 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, Apr. 2011. [Online]. Available: http://doi.acm.org/10.1145/1971162.1971171

[49] A. Botta and A. PescapÃ©, "Monitoring and measuring wireless network performance in the presence of middleboxes," in *Wireless On-Demand Network Systems and Services (WONS), 2011 Eighth International Conference on*, Jan 2011, pp. 146–149.

[50] P. Casas, P. Fiadino, and A. Baer, "Ip mining: Extracting knowledge from the dynamics of the internet addressing space," in *Teletraffic Congress (ITC), 2013 25th International*, Sept 2013, pp. 1–9.

[51] A. D'Alconzo, A. Coluccia, and P. Romirer-Maierhofer, "Distribution-based anomaly detection in 3g mobile networks: From theory to practice," *Int. J. Netw. Manag.*, vol. 20, no. 5, pp. 245–269, Sep. 2010. [Online]. Available: http://dx.doi.org/10.1002/nem.747

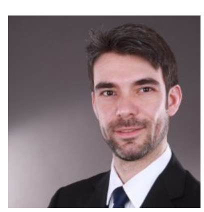
[52] P. Casas and R. Schatz, "Quality of experience in cloud services: Survey and measurements," *Computer Networks*, vol. 68, pp. 149–165, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2014.01.008

Pierdomenico Fiadino received his BSc and MSc degrees in Computer Engineering from Sapienza University of Rome, and his PhD degree in Electrical Engineering from the Institute of Telecommunications of TU Wien, in 2008, 2010 and 2015 respectively. He works as Data Scientist in the Big Data Analytics department of Eurecat (Technology Centre of Catalonia). Between 2010 and 2015 he held a Researcher position at the Telecommunications Research Center of Vienna (FTW). His research interests cover, among others, network traffic monitoring and analysis, machine learning, and business intelligence. He has authored and co-authored more than 25 publications in international conferences and journals.

Pedro Casas received the electrical engineering degree from the University of the Republic, Uruguay, in 2005, and the Ph.D. degree in computer science from Télécom Bretagne, France, in 2010. He is a Scientist with the Austrian Institute of Technology (AIT), Vienna, Austria. He held Research and Teaching Assistant positions with the University of the Republic, between 2003 and 2012, and was at the French Research Laboratory LAAS-CNRS, Toulouse, France, as a Postdoctoral Research Fellow between 2010 and 2011. Between 2011 and 2015, he was a Senior Researcher with the Telecommunications Research Center Vienna (FTW), Vienna, Austria. His research interests include the monitoring and analysis of network traffic, network security and anomaly detection, QoE modeling and automatic assessment, as well as machine-learning and data mining-based approaches for Networking. He has authored more than 80 networking research papers (50 as main author) in major international conferences and journals. He is the recipient of seven best paper awards in the last 6 years.

Alessandro D'Alconzo received the M.Sc. degree in Electronic Engineering with honors in 2003, and the Ph.D. in Information and Telecommunication Engineering in 2007, from Polytechnic of Bari, Italy. He is Scientist in the Digital Safety & Security department of AIT, Austrian Institute of Technology. From 2007 to 2015, he was Senior Researcher in the Communication Networks Area of the Telecommunications Research Center Vienna (FTW). From 2008 to 2013 he has been Management Committee representative of Austria for the COST Action IC0703 "Traffic Monitoring and Analysis". He has extensive experience in contributing and managing EU funded projects, as well as in applied research projects in the field of network traffic measurements in collaboration with national telecommunication operators. His research interests embrace network measurements and traffic monitoring, ranging from design and implementation of statistical based anomaly detection algorithms and root cause analysis, to Quality of Experience evaluation, and application of secure multiparty computation techniques to cross-domain network monitoring and troubleshooting.

Mirko Schiavone received his BSc and MSc degrees in Telecommunication Engineering from Sapienza University of Rome, in 2009 and 2012 and respectively. From 2012 until 2015 he was working at the Telecommunication Research Center of Vienna (FTW) where he conducted research on several topics, including: network traffic monitoring and analysis, anomaly detection and diagnosis and Big Data. He is currently working for Pico Global Ltd.: a financial technology firm specialized in ultra-low latency networks.



Arian Baer received his PhD in computer science from the University of Vienna in 2015. During the course of his PhD, he was a researcher at the Telecommunications Research Center Vienna (FTW). In 2009, he received his Diploma degree in Computer Science from the Friedrich-Alexander Universitaet Erlangen-Nuernberg. He is currently employed as a Big Data Architect at BMW Group in Munich, Germany. His research interests include network monitoring and analytics, data stream warehousing, query scheduling, data mining and machine learning. He has (co)authored about 20 publications in international journals and conferences.