

FlexiTop: sistema escalable y flexible de medidas de calidad para servicios *Over-The-Top*

Daniel Perdices*[†], Jorge E. López de Vergara*[†], Paula Roquero*, Carlos Vega*[†], Javier Aracil*[†]

*Departamento de Tecnología Electrónica y de las Comunicaciones, Universidad Autónoma de Madrid
Escuela Politécnica Superior, Calle Francisco Tomás y Valiente, 11, 28049 Madrid

[†]Naudit High Performance Computing and Networking, S.L.
Calle Faraday, 7, 28049 Madrid

daniel.perdices@naudit.es, {jorge.lopez_vergara, paula.roquero, javier.aracil}@uam.es, carlos.vega@naudit.es

Resumen—Hoy en día, el uso de los servicios *Over-The-Top* tales como *streaming* de vídeo, servicios web y redes sociales es dominante en el tráfico de Internet. En consecuencia, existe un amplio interés por parte de los proveedores de servicios de Internet en conocer la calidad de este tipo de comunicaciones para poder proporcionar el mejor servicio y por tanto la mejor experiencia a los usuarios. Para este propósito, se propone FlexiTop, un sistema flexible y escalable de medidas activas de calidad para este tipo de servicios, que permite obtener métricas con un consumo de recursos contenido. El diseño propuesto ha sido implementado y validado mediante pruebas. Así mismo se ha realizado una campaña de medidas durante varios meses en diferentes equipos, pudiéndose concluir que el sistema cumple con las expectativas de los actores involucrados.

Palabras Clave—QoS, medidas activas, servicios OTT, streaming, servicios web.

I. INTRODUCCIÓN

Un servicio *Over-The-Top* (OTT) se define como aquel que se provee a través de la red sin que ningún operador de telecomunicaciones esté involucrado en su envío, desarrollo o planificación [1], es decir, un servicio que utiliza la red como mero canal de transporte sin ningún otro tipo de consideración.

Los servicios OTT tienen necesidades y naturalezas muy heterogéneas. Mientras que, por un lado, los servicios de *streaming* multimedia requieren un ancho de banda dado y poca variabilidad del mismo, los servicios web tienen como prioridad el tiempo de respuesta. Además, desde el punto de vista de Calidad de la Experiencia (*Quality of Experience*, QoE) [2], existen otras métricas más relevantes que las tradicionalmente utilizadas en el análisis de tráfico de red. Por ejemplo, en estos casos es de interés no únicamente el tiempo de respuesta de las peticiones, sino también el tiempo total para finalizar una operación de alto nivel o una transacción completa. Estas diferencias llevan a la necesidad de analizar parámetros distintos según cada servicio.

Más allá de los parámetros que se desean analizar, muchos servicios actualmente funcionan bajo HTTPS, lo que dificulta el análisis pasivo de los mismos. Por ejemplo, es necesario poder ver el contenido de las peticiones HTTP para obtener métricas que son de gran interés para los demandantes de este sistema, tales como el porcentaje de fallos HTTP o los códigos de respuesta de las peticiones. Por ello, las medidas activas —en las cuales se puede monitorizar completamente el tráfico que envía y recibe el agente de medición— son prácticamente la única forma de identificar y analizar el tráfico que generan estos servicios sin recurrir a mecanismos para descifrar la conexión, solución que está limitada a escenarios muy concretos.

Además, el cambio constante de estos servicios, no solo en términos de prestaciones sino también en cuanto a funcionalidad y mecanismos de comunicación, hace necesario que los sistemas de medición evolucionen al mismo ritmo que lo hacen dichos servicios. Actualmente, centrarse en herramientas que midan un solo protocolo, o que se basen en una arquitectura diseñada exclusivamente sobre el protocolo que se quiere analizar, tendrán un periodo de vida corto.

Adicionalmente, el interés por estas medidas no proviene únicamente de los Proveedores de Servicio de Internet (*Internet Service Providers*, ISP), sino también de los usuarios. Como respuesta a esta necesidad, en este artículo se plantea FlexiTop, un sistema comercializado por naudit que se puede ejecutar en equipos de muy bajo coste y que permite un despliegue masivo de manera que se puedan registrar, agregar y analizar las medidas tomadas en cada uno de los dispositivos de una red de miles de puntos de medida.

Este análisis de servicios como YouTube [3] u otros servicios web [4] se ha realizado previamente de manera bastante exhaustiva en entornos de pruebas controlados. Sin embargo, la propuesta de este artículo es un sistema que permita tomar medidas de Indicadores Clave de

Prestaciones (*Key Performance Indicator*, KPI) de la aplicación y medidas a nivel de red, centrándose en aquellas métricas que son de interés para los ISP y los usuarios. Además, esta propuesta también aborda aspectos como la escalabilidad o el consumo de recursos, que en otros trabajos no han sido objeto de estudio.

La propuesta ha sido probada en diferentes tipos de dispositivos, abarcando PCs de sobremesa, entornos virtualizados y sistemas empujados de bajo coste. Los resultados han sido satisfactorios en todos los casos, las métricas se han validado con distintas alternativas y se ha logrado elaborar comparativas y análisis de estos servicios que permiten identificar los problemas que pueden causar una deficiente calidad de experiencia.

Para explicar el funcionamiento de FlexiTop, el artículo se ha estructurado de la siguiente manera: La sección II realiza un estudio de las tecnologías más usadas en la actualidad en los servicios OTT, explicándose a continuación en la sección III las diferentes alternativas a esta propuesta. Posteriormente, la sección IV enumera las métricas que son de interés tanto para los usuarios como para los ISP. Tras ello, en la sección V se describe la arquitectura del agente de medición. En relación con las métricas, en la sección VI se analizan la configuración de pruebas y los diferentes *tests* desarrollados, para así exponer los diferentes métodos y aproximaciones necesarias para medir un servicio a la vez que se muestran y comentan algunos de los resultados obtenidos en la sección VII. Finalmente se presentan las conclusiones y líneas de trabajo futuro.

II. PRELIMINARES

Para poder analizar los servicios, es necesario conocer las tecnologías y estándares que permiten que estos se presten. En el caso de los servicios de *streaming* multimedia, actualmente se utilizan ampliamente distintas tecnologías con bastantes similitudes. Los ejemplos más usados de estas son:

- 1) *Dynamic Adaptive Streaming over HTTP* (DASH, *Streaming* Adaptativo Dinámico sobre HTTP): técnica de *streaming* con tasa binaria adaptativa desde servidores HTTP convencionales [5].
- 2) *HTTP Live Streaming* (HLS, *Streaming* en Vivo con HTTP): técnica similar a la anterior con servidores convencionales que almacenan fragmentos con distintos tipos de codificación [6].

La mayor parte de las tecnologías anteriores comparten la característica de utilizar HTTP(S) como canal de transporte. Esta naturaleza permite que medir la calidad de estos servicios se base, en parte, en medir la calidad de estas conexiones HTTP. Sin embargo, se encuentran dificultades cuando se resuelven las URLs de los recursos solicitados ya que en la mayoría de casos estos fragmentos de vídeo son servidos por una Red de Distribución de Contenidos (*Content Delivery Network*, CDN), luego su resolución depende de la geolocalización del cliente y del servicio de nombres (*Domain Name System*, DNS).

Un factor importante aparte de la tecnología que provee el servicio es la configuración del reproductor o de la aplicación cliente. En este caso, existen parámetros como el tamaño de los *buffers*, los cambios programados de una calidad de vídeo a otra o el códec que se utiliza que tienen una gran importancia en el correcto funcionamiento de los servicios. Sin embargo, estos no son aspectos que estén relacionados con el funcionamiento de la red y, por tanto, no son de aplicación en este artículo.

Por otro lado, existen diferentes métodos de comunicación para servicios web. Los más utilizados son:

- 1) *Representational State Transfer* (REST, Transferencia de Estados Figurativos): arquitectura de comunicación basada en recursos HTTP.
- 2) *Simple Object Access Protocol* (SOAP, Protocolo Sencillo de Acceso a Objetos): protocolo estándar de intercambio de datos y llamadas a procedimientos sobre HTTP.

Actualmente, tanto aplicaciones en páginas web como aplicaciones para dispositivos móviles utilizan protocolos basados en REST para la comunicación entre cliente y servidor, por lo que existe un claro interés en el análisis de los tiempos de respuesta de estas *RESTful APIs*.

III. ESTADO DEL ARTE

En la actualidad, existe una gran variedad de métricas estandarizadas a nivel de red y de transporte. Estas métricas proporcionan información relevante acerca de los flujos originados por un servicio y pueden proporcionar información de interés para analizar las causas de las métricas observadas a nivel de aplicación y de la QoE. Muchas de estas métricas han sido estandarizadas por el grupo de trabajo IPPM (*Internet Protocol Performance Metrics*) del IETF (*Internet Engineering Task Force*). En concreto, son de relevancia para este trabajo la medición de latencia en un sentido [7] y en ambos [8], que estandarizan las métricas de latencia a nivel de red y que por tanto muestran metodologías de medición que son de interés.

Otros trabajos [9] siguen una aproximación radicalmente distinta a las medidas activas para resolver el problema del tráfico cifrado, basada en caracterizar el tráfico de estos servicios a base de encontrar patrones a nivel de red y de flujo [10]. Esta aproximación, también plausible, requiere sin embargo un mayor cálculo y un desarrollo más extenso de modelos matemáticos que se adapten a estos patrones deseados. Se pueden encontrar caracterizaciones ya realizadas para Facebook [4] y YouTube [11].

Para los servicios OTT, se ha observado ampliamente que existe una alta correlación entre los parámetros de Calidad de Servicio (*Quality of Service*, QoS) y de QoE [12]. Por tanto, está probado el interés en el análisis de la calidad de servicio de las conexiones que utilizan estos servicios. También existen diferentes trabajos que analizan la calidad de experiencia y de servicio de YouTube [3] o de otros servicios de vídeo o televisión bajo demanda [13]. Sin embargo, ninguno de los casos

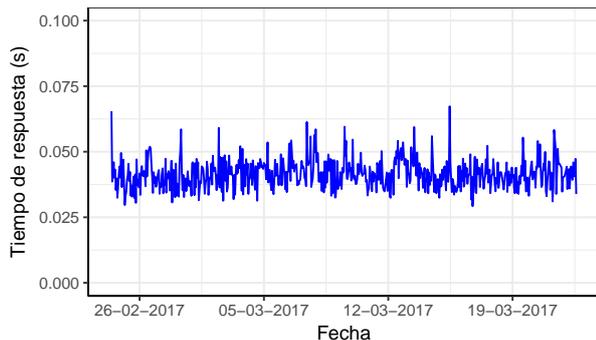


Fig. 1. Series de los tiempos de respuesta de DNS

adopta un enfoque más amplio que se centre en múltiples servicios.

Más allá de estas métricas clásicas y con el fin de poder obtener resultados más correlados con la QoE y de relevancia para los usuarios, se analizan KPIs del nivel de aplicación. Existen análisis previos de algunos de estos, especialmente en el impacto de aspectos de configuración de la aplicación como el tamaño del *buffer* en estas métricas y KPIs percibidos a nivel de aplicación. Queda fuera del ámbito de este trabajo analizar estos parámetros. En la medida de lo posible se proveerá una aproximación agnóstica a estos aspectos o en caso contrario, se proporcionará una configuración modificable para poder ver el impacto de estos parámetros.

IV. MÉTRICAS DE INTERÉS

Como se ha motivado en la introducción, las métricas han de diferir según el tipo de servicio que se desea medir. La caracterización de estas métricas y garantizar su uniformidad (estandarización) es por tanto necesario.

A. DNS

En primer lugar, es prioritario medir el rendimiento de las peticiones DNS por el interés que tiene conocer cómo se resuelven los nombres de dominio al acceder a un servicio. Con este propósito, se toma el tiempo de respuesta como principal métrica y además se almacenan las direcciones IP resultantes de la respuesta como apoyo para un posible análisis posterior.

En la Fig. 1 se muestra un ejemplo de serie temporal de la mediana de los tiempos de respuesta del servicio DNS para los 20 dominios más visitados en España. Para esta tarea de identificar estos dominios más visitados se puede utilizar un listado público. En este caso se ha utilizado Alexa¹. Estas métricas se pueden comprobar mediante la herramienta dig².

B. HTTP

Ya que la mayoría de servicios funcionan sobre HTTP(S), se requiere de un sistema de medición específico para este protocolo.

La aproximación seguida se basa en medir el tiempo de envío de una petición (*Request spread*) y posteriormente

medir el tiempo de recepción diferenciando dos marcas temporales: la llegada del primer paquete y la llegada del último. Estas dos marcas temporales van a servir para estimar lo que sería el tiempo de servicio y el tiempo de transmisión de la respuesta (*Response Spread*). Además, también se pueden obtener otras métricas del nivel de red (p.e., fragmentación) y de transporte (p.e., retransmisiones o ventanas 0) al analizar las conexiones asociadas a las peticiones HTTP.

Estas métricas se extienden al principal uso del protocolo analizado: la navegación web. Para este protocolo se pretenden dar estas métricas separadas según sea el documento base o las dependencias del mismo. Las dependencias se descargan concurrentemente. Se ha seguido una aproximación similar a la propuesta en TRANSUM [14], un disector para Wireshark que realiza un análisis de tiempos de respuesta de HTTP y otros protocolos como FTP o SMB2.

C. Streaming multimedia

Para servicios de *streaming* de contenido multimedia, un parámetro de probada relevancia es el *throughput*. No solo es un factor limitante para la calidad del contenido, sino que la variabilidad del mismo puede generar parones, cambios de calidad y otros problemas asociados al proceso de *buffering* que causan una mala calidad de experiencia.

Por tanto, para este tipo de servicios se realizan mediciones del ancho de banda consumido, del tiempo de descarga y el tiempo total de procesado (análisis de los manifiestos, obtención de las URLs, etc.) que se combinan con las medidas anteriormente mencionadas para HTTP.

D. Servicios web

Hay una variedad de servicios que funcionan sobre HTTP. Estos servicios abarcan desde redes sociales, servicios de mensajería hasta servicios de almacenamiento en la nube. Debido a su heterogeneidad, estos servicios van a tener unas necesidades variadas.

En general, los parámetros de interés en este caso van a ser el tiempo de acceso, tiempo de completado de las transacciones y velocidad de descarga. La naturaleza de las transacciones es un factor dependiente del servicio, por lo que en ciertos casos realizar estas transacciones puede ser excesivamente complejo. A diferencia de los servicios anteriores, el tiempo de procesado de la petición en el servidor y en el cliente es también un factor relevante.

E. Otros servicios

Más allá de los servicios que funcionan sobre HTTP, existen otros servicios que funcionan directamente sobre TCP/IP. Para estos servicios, se podrían dar, cómo se ha mencionado antes, métricas del nivel de red y transporte.

Sin embargo, el análisis del rendimiento de la aplicación resulta de mayor interés de cara a medidas más correladas con la QoE.

Uno de los principales servicios con estas características ha sido el correo electrónico a través de los protocolos más comunes de acceso: SMTP, IMAP y POP3.

¹<http://www.alexa.com/topsites/countries/ES>

²<https://ftp.isc.org/isc/bind9/cur/9.11/doc/arm/man.dig.html>

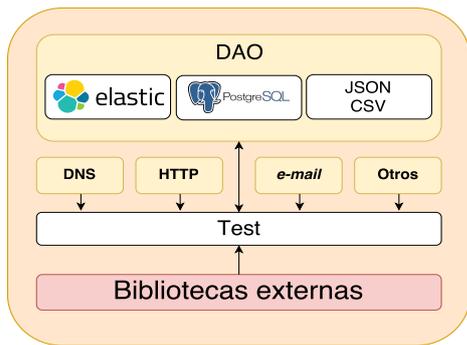


Fig. 2. Arquitectura del agente de medición.

V. ARQUITECTURA

En el diseño del sistema, se han tenido en cuenta los requisitos del sistema para dar la solución más adecuada. Los más relevantes para esta tarea han sido:

- **Flexibilidad:** como se ha motivado previamente, la flexibilidad del sistema tiene que ser acorde con el nivel de cambio de los servicios.
- **Escalabilidad:** de igual manera, el sistema ha de poder desplegarse en diferentes localizaciones sin requerir grandes procesos de configuración, de tal manera que los resultados estén disponibles de manera centralizada y de manera local.
- **Bajo consumo de recursos:** con el fin de ejecutar el sistema con el menor impacto posible y además en equipos de bajo coste, se pretende limitar el consumo del sistema así como proveer soluciones eficientes.

Estas prioridades resultan en una arquitectura del agente que realiza las medidas según se muestra en la Fig. 2:

- 1) **Módulo de DNS:** este módulo es utilizado por todos los *tests* para medir el tiempo de resolución de las peticiones al servicio de nombres.
- 2) **Módulo de HTTP:** este módulo permite realizar pruebas de manera concurrente o con un único hilo para estimar la máxima velocidad de descarga que se puede alcanzar. Soporta múltiples URLs, configuración de número máximo de hilos y distintas opciones para los *buffers* y repeticiones.
- 3) **Database Access Object (DAO, Objeto de Acceso a Bases de Datos):** módulo que se encarga de abstraer la recolección (y el envío) de resultados y las herramientas de registro de errores.
- 4) **Otras bibliotecas y utilidades:** además se proveen bibliotecas propias (p.e. e-mail) y se pueden utilizar otras externas para construir las peticiones de las diferentes APIs del servicio o para el procesado del resultado de la petición.

Esta arquitectura permite no solo el desarrollo adecuado del proyecto, sino que garantiza que se puedan medir nuevos servicios en el futuro, así como modificar los ya implementados. Cada *test* se implementa según el siguiente esquema genérico: se obtienen las URLs del servicio, se resuelven mediante el módulo de DNS, se utiliza el módulo de HTTP u otro para calcular las diferentes métricas y se utiliza el DAO para guardar los resultados. En principio, los *tests* son independientes, pero

pueden llamarse unos a otros o depender unos de otros para casos de prueba complejos.

Aparte de los agentes de medición, se tiene un servidor que puede realizar tareas tales como actualización de los *tests*, recepción de resultados, asignación de identificadores, activación o desactivación de *tests* o análisis estadístico. Esto permite que el agente no sea solo interesante para el cliente que instale el sistema, sino también para el proveedor que recibirá estadísticas y datos de interés sobre el funcionamiento de los servicios en la red que sean de ayuda para proporcionar un nivel superior de calidad a los usuarios.

VI. ENTORNO EXPERIMENTAL

El software del agente de medición ha sido desarrollado en Python, por la portabilidad y versatilidad que ofrece. Éste se ha probado en diferentes dispositivos:

- 1) Orange Pi Plus: H3 Quad-core Cortex-A7, 1GB DDR3, Gigabit Ethernet.
- 2) Equipo de pruebas (*host*): Intel Core i7 860, 8GB DDR3, Gigabit Ethernet.
- 3) Máquina virtual (*guest*): dos núcleos, 1GB DDR3, 2GB de disco duro, Gigabit Ethernet.

La propuesta completa ha estado ejecutándose de manera continuada en los diferentes equipos. En el primero de los mencionados, el sistema se dedica exclusivamente al sistema de medición, aunque nunca se utilizan todos los recursos disponibles. En el segundo caso, se ha ejecutado el sistema sobre el propio *host* y en diferentes entornos virtualizados: mediante una máquina virtual y mediante un contenedor Docker. Ambos equipos se encuentran conectados a la red de la Universidad Autónoma de Madrid.

El entorno virtualizado es interesante, porque permite no solo registrar los recursos consumidos sino también limitar y controlar el consumo de los mismos y el uso del *hardware* del *host*, p.e., el número de núcleos, la memoria RAM disponible o el espacio en disco.

El sistema servidor se encuentra instalado en el equipo de pruebas. Este sistema incluye visualización en tiempo real y análisis de datos.

VII. PRUEBAS Y RESULTADOS

Una vez se ha implementado la arquitectura propuesta, se implementan *tests* de diferentes servicios para validar el diseño realizado. Estos servicios son:

- **Servicios de streaming multimedia:** Netflix, YouTube, Yomvi, Spotify.
- **Servicios web:** Twitter, Facebook, Dropbox, Speedtest, Google (buscador), Navegación web.
- **Otros:** Correo electrónico.

En la implementación de cada *test* se han utilizado técnicas variadas para simular la utilización de cada uno de los servicios. A continuación se explican las particularidades de cada servicio, así como los principales procesos usados dependiendo del servicio, y se muestran algunos resultados obtenidos.

A. Servicios de streaming multimedia

El procedimiento para tomar medidas está orientado a obtener el ancho de banda utilizado. Para ello, se obtienen varias URLs de fragmentos de contenido multimedia y se descargan de manera simultánea utilizando varios hilos y conexiones.

La manera de obtener las URLs difiere según el servicio. Para YouTube, existen tanto bibliotecas como formas de descargar el vídeo analizando el manifiesto DASH. De manera muy similar, Spotify provee una API REST³ que permite obtener URLs de muestras de cada canción. Estos servicios exponen alguna forma de acceso a las URLs, por lo que no es necesario utilizar técnicas más avanzadas.

No obstante, para otros servicios implementados, se ha requerido de la técnica de interposición (*Man in the Middle*) para analizar las conexiones HTTPS que se utilizan en el servicio. Esto permite hallar las URLs y analizar la lógica del servicio. Con el mismo propósito, se ha estudiado también el código de reproductores multimedia *online* para determinar con mayor precisión el funcionamiento del servicio y validar la información obtenida. Un ejemplo de este procedimiento ha sido el caso de Yomvi, en el que el análisis del código y de las conexiones HTTPS ha permitido realizar pruebas de rendimiento del servicio con un consumo de recursos reducido y sin la necesidad de ejecutar un navegador web.

Sin embargo, para otros casos este proceso puede ser tedioso e incluso resultar intensivo para el agente de medición. En estos casos no queda otra alternativa más que ejecutar un navegador durante un corto periodo de tiempo para probar el servicio. Aunque esta aproximación es viable y se han desarrollado herramientas que se integran con el navegador para capturar estas peticiones, no es la situación deseable para mantener un bajo consumo de recursos. Este es el caso de Netflix, en el cual se envía el manifiesto DASH cifrado y la única manera de obtener URLs sería realizar el proceso descrito. No obstante, tras la publicación de la herramienta *fast.com*⁴, proporcionada por Netflix para probar la velocidad de descarga de su servicio de vídeo, se ha realizado ingeniería inversa y así se ha construido un método similar para obtener estas URLs sin necesidad de ejecutar el reproductor de Netflix.

En la Fig. 3 se muestran los resultados de *throughput* en Mbps obtenidos durante varias semanas. En esta figura se puede ver la variabilidad de estos servicios así como las notables diferencias de rendimiento en media. Se puede observar una mejora clara en la velocidad de descarga de YouTube en torno al 4 de febrero. Este tipo de análisis permite averiguar cambios en la configuración del servicio, como es el caso anterior, momentos de mayor congestión a lo largo de la semana o del día, etc.

En los casos en los que ha sido posible, se han validado los resultados de estas medidas con ayuda de los reproductores o de otras herramientas.

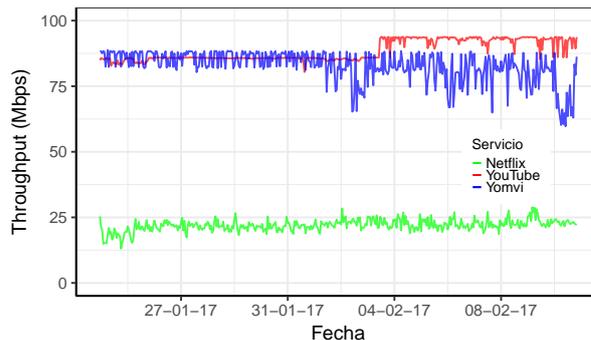


Fig. 3. Series temporales del ancho de banda (mbps) para Netflix, YouTube y Yomvi.

B. Servicios web

Para los servicios web, se realiza un proceso análogo al anterior. Para los servicios que proveen un API pública basada en REST, se construye un sistema de medición en torno al tiempo de respuesta de las peticiones. Esto sucede por ejemplo con Twitter⁵ o Dropbox⁶.

Por otro lado, para el resto de servicios se ha realizado un análisis de las peticiones utilizando de nuevo la técnica de interposición para analizar la lógica del protocolo. Para ciertos servicios como WhatsApp esta aproximación es la única posible. Se encuentran ya diferentes bibliotecas que permiten el acceso al API de WhatsApp. Sin embargo, en ocasiones se impone en los términos y condiciones del servicio que el acceso al mismo debe ser únicamente realizado a través de su cliente oficial. Esto impide, al menos desde un punto de vista legal, la medición de estos servicios salvo a través de la ejecución del cliente oficial y por tanto forzando la integración del sistema de medición en la aplicación cliente, creando en general un sobre coste de consumo de recursos y de tiempo de desarrollo a tener en cuenta.

Para el caso de la navegación web, se ha analizado el tiempo de descarga de cualquier página. Para esta tarea, se descarga el documento principal HTML, se obtienen las dependencias de ese documento y se procede a su descarga en paralelo. Durante la descarga se contabiliza el tiempo de envío de todas las peticiones a la vez, así como el tiempo de recepción (desde que se recibe el primer fragmento de la primera dependencia hasta que se ha completado la descarga de todas las dependencias). Esto permite simular el mecanismo de carga de una página web.

Por ahora, los documentos *JavaScript* se descargan, pero no se interpretan evitando así fallos de seguridad y manteniendo el consumo de recursos al mínimo. No obstante, esto provoca que las medidas sean poco fieles para páginas con un contenido mayoritariamente dinámico. En el resto de webs, los tiempos han sido validados satisfactoriamente con varios navegadores webs. Para futuras versiones, se plantea la posibilidad de ejecutar código *JavaScript* con ayuda de tecnologías que simulan un navegador sin interfaz gráfica⁷. Otra opción es hacer

³<https://developer.spotify.com/web-api/>

⁴<https://fast.com>

⁵<https://dev.twitter.com/rest/public>

⁶<https://www.dropbox.com/developers/documentation/http/overview>

⁷<http://phantomjs.org/>

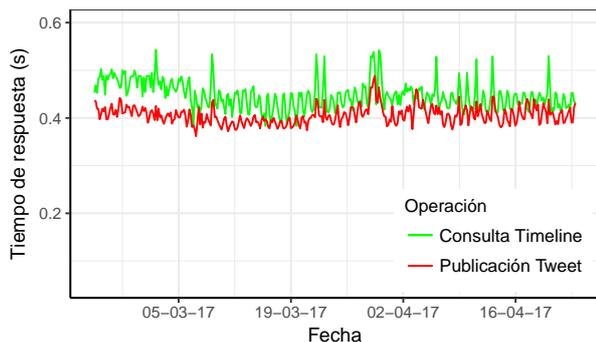


Fig. 4. Serie temporal del tiempo de publicación de un *tweet* y del tiempo de recuperación del *timeline*

que el agente ejecute un navegador completo donde se tomen estas medidas con ayuda de un complemento.

En la Fig. 4 se representan las medianas de los tiempos de respuesta del servicio Twitter. Sin entrar en aspectos de funcionamiento interno de las aplicaciones, esta métrica debería estar mucho más cercana a la calidad de experiencia percibida por el usuario que el análisis del tiempo de establecimiento de la conexión.

C. Otros servicios

Para este caso concreto se realizan medidas del correo electrónico. Igual que en los casos anteriores, se proveen algunos KPIs y métricas. Algunos ejemplos son: tiempo que se tarda en establecer una sesión por IMAP o SMTP, tiempo de envío o recepción y descarga de un correo de un tamaño dado y velocidad de descarga de un correo. Esto muestra simplemente cómo gran parte de las herramientas desarrolladas también son de utilidad en protocolos que no utilizan HTTP como medio de transporte.

De igual manera, el análisis pasivo sería una alternativa. Sin embargo, pese a poder identificar los flujos al utilizar puertos conocidos, es imposible analizar los comandos del protocolo si el tráfico va cifrado. Por tanto, es inviable cuantificar el tiempo de envío de un correo o el tiempo de consulta de la bandeja de entrada con un análisis pasivo. Por ello, las medidas activas son una opción más adecuada. En particular, este sistema proporciona la capacidad de análisis de los KPIs, así como la identificación de los flujos para su posterior análisis.

VIII. CONCLUSIONES

Las medidas activas han demostrado ser una alternativa al puro análisis pasivo del tráfico. No solo se evita el problema del análisis del tráfico cifrado y el de la caracterización y detección de los flujos generados por los servicios, sino que además se tiene control y conocimiento de ciertos parámetros del nivel de aplicación como el número de conexiones utilizadas o el tamaño del *buffer* utilizado.

Como prueba de concepto, se ha realizado una implementación de esta arquitectura así como de varios *tests* relevantes para los usuarios. Los resultados obtenidos son satisfactorios y han sido validados con diferentes herramientas específicas de cada servicio. Según se ha mostrado en la sección anterior, el sistema implementado

es completamente funcional y las medidas obtenidas son relevantes para el rendimiento de los servicios.

Además, la utilidad de estas medidas va más allá del análisis del servicio en sí. Con la caracterización de las métricas y KPIs de interés y mediante el tratamiento de los datos obtenidos, se podría utilizar el sistema para buscar también la causa de los cambios de rendimiento, llegando a poder caracterizar mejoras en la infraestructura, detectando horas puntas de uso o posibles anomalías a nivel de red.

La arquitectura propuesta permite el cumplimiento de los requisitos y, además, realizar implementaciones de servicios muy heterogéneos sin necesidad de que esto implique un coste de desarrollo desmedido. También se ha realizado un análisis de los servicios que son de interés para los usuarios. A partir de ello, se han propuesto procedimientos de elaboración de futuros *tests* que permitirán actualizar el sistema conforme surjan nuevas aplicaciones y servicios.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad y el Fondo Europeo de Desarrollo Regional a través del proyecto TRÁFICA (MINECO/FEDER TEC2015-69417-C2-1-R).

REFERENCIAS

- [1] W. Green, B. Lancaster, and J. Sladek, "Over-the-Top Services," in *Pipeline*, vol. 4, no. 7, 2006.
- [2] K. Brunnström, S. A. Beker, K. De Moor, and A. Dooms et al, "Qualinet White Paper on Definitions of Quality of Experience," Mar. 2013, qualinet White Paper on Definitions of Quality of Experience Output from the fifth Qualinet meeting, Novi Sad, March 12, 2013.
- [3] G. Dimopoulos, "YouTube Traffic Monitoring and Analysis," Master's thesis, Technical University of Catalonia, 2012.
- [4] M. Kihl, R. Larsson, N. Unnervik, J. Haberkamm, A. Arvidsson, and A. Aurelius, "Analysis of facebook content demand patterns," in *2014 International Conference on Smart Communications in Network Technologies (SaCoNeT)*, June 2014, pp. 1–6.
- [5] I. Sodagar, "MPEG-DASH: The Standard for Multimedia Streaming Over Internet," Tech. Rep., 04 2012.
- [6] R. Pantos and W. May, "HTTP Live Streaming," Internet Engineering Task Force, Internet-Draft draft-pantos-http-live-streaming-21, Mar. 2017, work in Progress.
- [7] S. Kalidindi, M. J. Zekauskas, and D. G. T. Almes, "A One-way Delay Metric for IPPM," RFC 2679, Sep. 1999.
- [8] G. Almes, S. Kalidindi, and M. J. Zekauskas, "A Round-trip Delay Metric for IPPM," RFC 2681, Sep. 1999.
- [9] Ixia, "IxChariot," <https://www.ixiacom.com/products/ixchariot>.
- [10] A. Cuadra-Sanchez and J. Aracil, "A novel blind traffic analysis technique for detection of WhatsApp VoIP calls," *International Journal of Network Management*, vol. 27, no. 2, 2017.
- [11] J. P. Laulajainen, A. Arvidsson, T. Ojala, J. Seppanen, and M. Du, "Study of youtube demand patterns in mixed public and campus wifi network," in *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Aug 2014, pp. 635–641.
- [12] D. Rivera, N. Kushik, C. Fuenzalida, A. Cavalli, and N. Yevtushenko, "QoE Evaluation Based on QoS and QoBiz Parameters Applied to an OTT Service," in *2015 IEEE International Conference on Web Services*, June 2015, pp. 607–614.
- [13] R. K. P. Mok, E. W. W. Chan, and R. K. C. Chang, "Measuring the quality of experience of HTTP video streaming," in *Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management, IM 2011, Dublin, Ireland, 23-27 May 2011*, 2011, pp. 485–492.
- [14] "TRANSUM Wireshark Plugin: Analyzing a Website Problem," <https://community.tribelab.com/mod/page/view.php?id=492>.