

Procesado masivo y automático de trazas en redes celulares

G. J. Anaya-López⁽¹⁾, N. Guerra-Melgares⁽²⁾, C. Cárdenas-Angelat⁽²⁾, M. C. Aguayo-Torres⁽¹⁾, J. Baños-Polglase⁽²⁾
gjal@ic.uma.es, noelia.guerra@dekra.com, carlos.cardenas@dekra.com, aguayo@ic.uma.es, janie.banos@dekra.com

⁽¹⁾Dpt. Ingeniería de Comunicaciones, Universidad de Málaga, Málaga, España 29071

⁽²⁾DEKRA Testing and Certification S.A.U., Málaga, España 29590

Resumen—The introduction of new services in mobile communications usually requires extensive evaluation campaigns. Even though new services and devices may have undergone conformance testing, it is important to ensure they do properly work in real live networks, where some parameters, signalling procedures or timer values may differ from those used in conformance testing. It is also important to diagnose error sources. This paper describes a tool that automatically analyses traces captured by the mobile terminal. Its design addresses two main challenges: the efficient handling of the large amount of information captured by the terminal and the extraction of relevant information from traces generated by different sources.

I. INTRODUCCIÓN

El tráfico en las redes móviles crece continuamente [1]. El avance de la tecnología, la coexistencia de múltiples tecnologías celulares y el crecimiento de las redes, tanto en extensión como en complejidad, así como la introducción de nuevos servicios provoca la necesidad de asegurar la Calidad de Experiencia (QoE) a los usuarios.

Con el fin de asegurar el correcto funcionamiento de la red y unos niveles apropiados de QoE para el usuario, los operadores cuentan con Sistemas de Soporte a las Operaciones (OSS) que son capaces de recopilar estadísticas de Calidad de Servicio (QoS) de los diferentes elementos de la red [2]. Normalmente los operadores complementan la captura de información realizando campañas de medida (*field test*) que aportan información desde el punto de vista del usuario.

Sin embargo, los propios fabricantes de dispositivos o los proveedores de servicios sobre redes móviles (por ejemplo, aquellos que prestan servicios de *infotainment* para vehículos) necesitan asegurar que los dispositivos y los servicios proporcionados funcionan correctamente. Como el fabricante no tiene acceso a la información del operador, que, en cualquier caso, no está individualizada para su terminal, se debe realizar el análisis en base a campañas de medida que exigen una gran cantidad de medidas y el procesamiento de una ingente cantidad de datos para asegurar que los procedimientos de señalización con la red (registro, traspaso, reelección, establecimiento y liberación de conexiones, etc.) funcionan correctamente. Se debe identificar, en su caso, en qué punto el intercambio de mensajes ha fallado, qué parámetros de los mensajes son incorrectos o qué temporizadores no están correctamente configurados.

En este documento, se presenta una metodología de trabajo y el diseño de una herramienta para la automatización y optimización del procesado masivo de trazas recopiladas en campañas de medidas *field trials*. El objetivo es verificar tanto

el comportamiento del terminal y el servicio, como de la propia red del operador, causa de los fallos en ocasiones.

El procedimiento de medida requiere capturar los mensajes intercambiados entre el terminal móvil y la red a través de la interfaz radio, analizar los procedimientos de señalización que han tenido lugar y determinar si se han producido errores. Además, se extraen los parámetros de los mensajes que son información relevante para la evaluación del comportamiento del terminal y la red. Algunos de estos parámetros son las medidas realizadas por el móvil, la reserva, configuración y liberación de recursos, información de seguridad, etc.

Capturar los mensajes en un terminal móvil y analizarlos de forma manual es posible. Sin embargo, este trabajo es lento, tedioso y resulta difícil no perder eventos relevantes para el análisis de incidencias. Además, si la campaña de medidas se extiende en un intervalo espacial o temporal grande, la ingente cantidad de datos hace el procesamiento manual inviable.

La herramienta desarrollada para la automatización del análisis de trazas está basada en PySpark [3], que permite la ejecución distribuida sobre varios equipos, así como la optimización de recursos. En concreto se emplea la configuración *standalone* [3] de PySpark. En ella, un equipo hace de coordinador y va distribuyendo recursos de todos los equipos conectados al mismo a las tareas que se realicen. Se utiliza una arquitectura ETL (Extracción, Transformación y Carga) que permite separar la fase de captura de datos de la de procesamiento y análisis de trazas de diferentes fabricantes de *chipsets* en los dispositivos.

Además, se ha empleado un modelo computacional ampliamente usado para automatización como son las máquinas de estado finitas. Tras analizar las normas [4], [5], se pudo identificar que gran parte de los procedimientos se realizan mediante el envío de un único mensaje de señalización, el intercambio de dos mensajes o el intercambio de mensajes de señalización en tres bloques. El método basado en máquinas de estados permite ampliar fácilmente el procesamiento a nuevos procedimientos de señalización, conforme evolucione la norma o cambie la problemática objeto de estudio.

En la sección II se presentan algunos de los retos y cómo se han resuelto con el diseño propuesto. En la sección III se describe en detalle la fase de análisis de los procedimientos de señalización. En la sección IV se presentan algunos resultados de rendimiento. El trabajo concluye en la sección V.

II. DISEÑO GENERAL DEL SISTEMA

Para capturar los mensajes intercambiados entre el terminal y la red, existen aplicaciones software capaces de automatizar

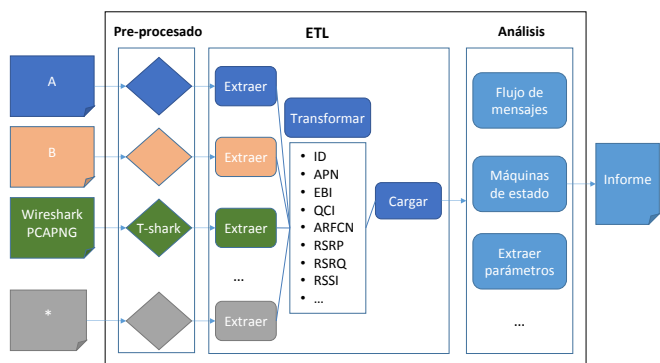


Fig. 1. Arquitectura general del sistema

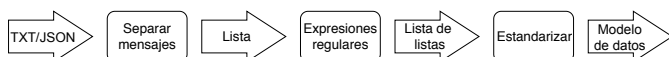


Fig. 2. Etapa de extracción

este proceso, habitualmente soluciones propietarias proporcionadas por los proveedores del *chipset* del terminal. La salida de este proceso son ficheros de trazas con ingentes cantidades de información: todos los mensajes intercambiados con la red y los eventos a nivel de *chipset*. Si el análisis a efectuar requiere analizar mensajes por encima de la capa IP (como en el Subsistema Multimedia IP o IMS), se puede utilizar la conocida herramienta Wireshark, que gracias a su analizador de trazas t-shark, permite filtrar y extraer los parámetros necesarios de una manera sencilla.

En las campañas de medida de tipo *field trials*, la propia captura de datos consume muchos recursos. En la mayoría de los casos, los datos se capturan en diferentes interfaces utilizando varias herramientas y son almacenados y enviados periódicamente para su procesamiento posterior.

La Fig. 1 muestra la arquitectura del sistema diseñado, donde se puede apreciar que el sistema está dividido en tres grandes bloques: Pre-procesado, ETL y Análisis.

La entrada al sistema son las trazas extraídas del terminal móvil. El formato de los mensajes a nivel de capas bajas es diferente para cada fabricante de *chipssets* y, en casi todos los casos, ininteligible por el ser humano. Por esta razón, se precisa una fase de pre-procesado en la que estos ficheros se conviertan a lenguaje natural antes de continuar con su análisis. Esta fase de pre-procesado usa programas proporcionados por los fabricantes de *chipssets*.

La extracción de la información de interés tiene lugar en la segunda fase del sistema, el proceso ETL, cuya finalidad es abstraer el análisis de los datos del tipo de fichero de entrada.

La primera etapa del proceso ETL (Fig. 2) consiste en extraer información de los distintos mensajes capturados en la interfaz aire. En primer lugar, se localizan y aíslan cada uno de los mensajes. A continuación, se aplican expresiones regulares [6] para obtener el tipo de mensaje que es (NAS, RRC...) y el mensaje en sí (*Attach Request, Attach Complete...*), además de buscar los parámetros de interés en cada paquete de datos de los mensajes. Por ejemplo, en los mensajes de rechazo o fallo es relevante obtener la causa del error que aparece en el contenido del mensaje.

Finalmente, se estandariza la información para hacerla

independiente de su origen. De este modo, a la salida de este bloque se obtiene un conjunto de datos cuyo formato no depende del fabricante del *chipset*. La salida es una lista de datos estructurados con la información de la traza de la que proviene, la hora del mensaje, la dirección (ascendente/descendente), el tipo del mensaje, el mensaje en sí y los parámetros más relevantes de cada uno de ellos. Esto facilita el análisis posterior de los datos.

La segunda etapa del proceso ETL es la de transformar. Esta etapa tiene la función de tratar la información de forma que se facilite la carga a la estructura de datos usada por PySpark. Se emplea la escritura a un fichero CSV donde cada línea representa un mensaje. Esto proporciona varios beneficios: a) PySpark permite cargar todos los ficheros de una misma carpeta de forma sencilla para los formatos CSV; b) guardar la información en este paso intermedio permite separar el procesamiento temporal y espacialmente; y c) el fichero creado en este paso intermedio proporciona trazabilidad al sistema: si la salida muestra resultados erróneos, es posible comprobar si ha fallado antes o después de este punto.

En la tercera etapa del proceso ETL, la información almacenada en los ficheros CSV se carga a una estructura de datos denominada *DataFrame* [7]. Es una estructura de datos distribuida y organizada en columnas, equivalente a una tabla de una base de datos relacional. Esto permite operar los datos y obtener estadísticas mediante consultas *Structured Query Language* (SQL) o usando las funciones de la Interfaz de Programación de Aplicaciones (API) de PySpark optimizadas para este tipo de estructuras.

Finalmente, la información cargada en el *DataFrame* es procesada empleando sus múltiples herramientas para obtener los procedimientos de señalización y verificar que se ejecutan correctamente, dentro de las múltiples opciones que permite la norma.

III. ANÁLISIS

En la fase de análisis, se filtran del conjunto total de mensajes aquellos que participan en cada procedimiento según la norma, junto con los parámetros relevantes para ese procedimiento. Además, se comprueban los temporizadores asociados a los procedimientos de señalización y, finalmente, se verifica que los procedimientos se han ejecutado con los parámetros correctamente configurados y en los tiempos correctos. Si un procedimiento no se ejecuta o no lo hace correctamente, se identifican los puntos espacio temporales (fecha/hora, localización) donde esto tiene lugar. La herramienta también permite visualizar adecuadamente los resultados y agregarlos para obtener estadísticas.

A partir de los mensajes de señalización, se incorporan también al sistema tanto parámetros que influyen en el resultado de los procedimientos o su geolocalización, como aquellos relacionados con el comportamiento del terminal y de la red: recursos reservados o liberados, niveles de señal, número de celdas vecinas, etc.

La salida del sistema es un informe de resultados con los procedimientos de señalización que han tenido lugar entre el terminal móvil y la red y los parámetros de interés para conocer el estado de la red.

Después de identificar y analizar de forma atómica los procedimientos definidos en las normas del 3GPP, se ob-

servó que la mayoría se realizan mediante el envío de un mensaje de señalización, el intercambio de dos mensajes de señalización o el intercambio de mensajes de señalización en tres bloques. Considerando esta característica, surge la idea de utilizar máquinas de estado como núcleo para el análisis de los procedimientos. Con este concepto se puede generalizar el análisis a los procedimientos identificados. De esta forma, se otorga una gran flexibilidad al sistema, ya que permite analizar procedimientos de las tecnologías 2G, 3G y 4G, y permite ampliar fácilmente el procesamiento a nuevos procedimientos de señalización (5G) o a procedimientos de capas superiores (IMS o SIP), conforme evolucione la norma o cambie la problemática objeto de estudio.

Las máquinas de estado genéricas implementadas son:

- Máquina de un estado. Engloba todos aquellos procedimientos constituidos por un único mensaje como, por ejemplo, el procedimiento de búsqueda (*paging*).
- Máquina de dos estados. Los procedimientos están compuestos por dos mensajes y, por tanto, aparecen las transiciones entre estados y las restricciones temporales entre mensajes como, por ejemplo, el procedimiento de desregistro (*detach*).
- Máquina de tres estados. Es más compleja al existir más combinaciones en las transiciones y dos temporizadores. Un ejemplo es el registro (*attach*).

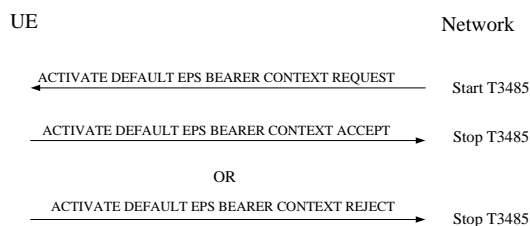


Fig. 3. Procedimiento *NW Initiated Default EPS Bearer Context Activation* [4]

Como ejemplo, en la Fig. 3, se presenta el procedimiento para la activación de contextos de portadoras por defecto EPS (Sistema de Paquetes Evolucionado). El procedimiento está compuesto por dos mensajes: un mensaje inicial (*Activate Default EPS Bearer Context Request*) y una respuesta (*Activate Default EPS Bearer Context Accept* o *Activate Default EPS Context Reject*) que varía en función de su resultado (completado con éxito o rechazado). El tiempo entre los mensajes está controlado por un temporizador (T3485) cuyo valor viene definido en el estándar [4].

Los procedimientos que no encajen en estas máquinas de estado se incorporan llevando a cabo pequeñas modificaciones específicas para adaptarse o, si fuese completamente necesario, creando máquinas de estado exclusivas para dichos procedimientos.

Estas máquinas de estados precisan información de configuración (estados, datos de entrada e informes de salida) para poder funcionar. Con los estados y sus transiciones se lleva a cabo la lógica del análisis comprobando las secuencias de mensajes.

Para generalizar el proceso a cualquier procedimiento, cada máquina de estados consta de un conjunto de datos de entrada.

Estos datos se dividen en dos grupos. El primero es el conjunto de mensajes sobre el que se buscarán ocurrencias del procedimiento. Este debe estar ordenado secuencialmente. El segundo es la configuración de la máquina de estados para cada procedimiento de señalización. Permite indicar al sistema qué tipo de máquina de estados utiliza cada procedimiento, los mensajes que lo conforman, los temporizadores que controlan el flujo de mensajes y los posibles resultados del procedimiento (completado, rechazado y temporizador expirado).

Para asegurar el correcto funcionamiento del proceso implementado en forma de máquinas de estado, es importante ordenar los mensajes registrados, que pueden no haberse almacenado secuencialmente debido a que pueden provenir de diferentes ficheros de captura.

En la Fig. 4, se muestra parcialmente el funcionamiento de la máquina de dos estados. Inicialmente, se espera un mensaje de arranque válido. Una vez se ha detectado, se transita al estado S2, cuyo funcionamiento es el que muestra en la figura. Para el estado S2 la figura 4 muestra todos los casos posibles: 1) Si no quedan mensajes por analizar, la máquina termina; 2) Si el temporizador expira antes de recibir un segundo mensaje, la salida es “Temporizador expirado”; 3) Si se recibe un duplicado del mensaje de arranque, se reinicia el procedimiento; 4) En el caso de recibir un mensaje válido, la salida es “Completado” o “Rechazado”, según lo que le corresponda; 5) Por último, si se recibe un mensaje fuera de lo definido en la norma, se notifica que ha habido un Error y se continúa esperando.

Al terminar el análisis se obtiene una lista con tantas entradas como procedimientos estén contenidos en los datos de entrada. Cada entrada de la lista consta de: el resultado del procedimiento, las marcas temporales de los mensajes que lo componen, el temporizador expirado, en el caso de que exista, y el nombre del fichero que contiene el procedimiento. Las marcas temporales de cada mensaje se emplean para incorporar al informe los parámetros deseados en cada uno de los procedimientos. Esto es posible gracias a la trazabilidad que se ha otorgado al sistema: es posible recuperar el mensaje completo a partir del fichero donde se encuentre y la marca temporal.

El análisis se complementa comprobando aquellos parámetros de los procedimientos que podrían alterar su funcionamiento y se modifica su resultado si es necesario. Por ejemplo, el procedimiento de *Tracking Area Update* (TAU) (Fig. 5) lleva un último mensaje que es opcional según si el valor de los parámetros GUTI o TMSI están presentes o no, y si sus valores se han visto modificados. Así, el resultado inicial del análisis debe cambiarse a “Completado” si la razón del “Temporizador expirado” fue que no llegó el mensaje opcional.

Aunque hubiera sido posible modificar las máquinas de estado de estos procedimientos, esto habría dado lugar a una variedad de máquinas de estado casi como de procedimientos. La búsqueda de flexibilidad en la herramienta ha llevado al diseño de un conjunto más limitado de máquinas de estado y añadir lógica adicional posterior. Solo para el procedimiento de traspaso (*handover*) a 2G se ha considerado necesario hacer una máquina de estados específica para ese procedimiento.

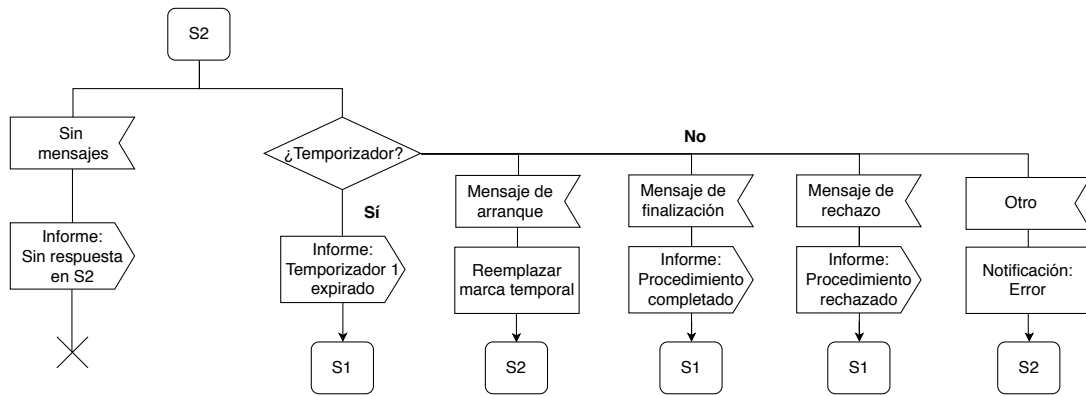


Fig. 4. Máquina de dos estados - Estado S2

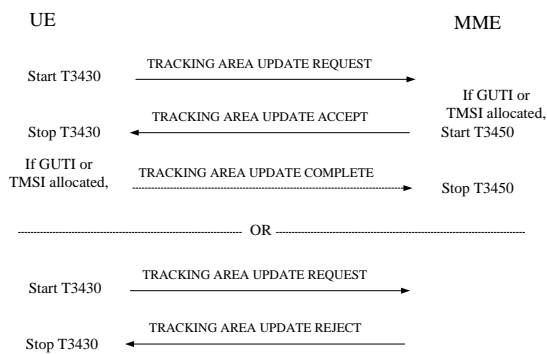


Fig. 5. Procedimiento *Tracking Area Update* a partir de [4]

IV. RESULTADOS

La herramienta desarrollada es capaz de analizar un total de 83 procedimientos de señalización distintos y extrae hasta 123 parámetros que proporcionan información adicional. Esta información abarca tecnologías tales como 2G, 3G y 4G, pero también aquellas de capas superiores como IMS o SIP. Además, su diseño permite fácilmente ampliarla con los procedimientos de señalización de 5G.

La capacidad para analizar trazas en cuanto a volumen y a velocidad de procesamiento depende de los equipos y los recursos que se dediquen. Emplear más equipos no siempre mejora el rendimiento, se necesita mantener un equilibrio entre la paralelización de procesos, la distribución de tareas entre procesadores, y el intercambio de información entre equipos. PySpark permite fácilmente configurar los recursos dedicados a cada tarea, así como la distribución de tareas entre procesadores.

La tabla I muestra algunos tiempos de procesamiento. Para obtenerlas se han usado dos ordenadores con 32 GB de memoria RAM y procesadores con 4 núcleos. El pre-procesado se ha realizado con herramientas propias del proveedor del *chipset*, por lo que sus tiempos no se pueden optimizar.

V. CONCLUSIONES

Tras analizar los procedimientos de señalización de los estándares de comunicaciones móviles existentes, se han creado tres máquinas de estado genéricas que permiten implementar un gran número de procedimientos de forma sencilla y escalable. Los procedimientos que no encajan en estas

TABLA I
TABLA DE RENDIMIENTO

Nº de trazas	20	52	56	91
Tamaño de fichero en lenguaje natural (GB)	1.32	2.71	2.35	2.85
Tiempo de procesado (min)	10.41	19.29	13.52	17.84
Tiempo de pre-procesado (min)	23.25	51.92	52.17	77.13

máquinas de estado se pueden incorporar en general llevando a cabo pequeñas modificaciones específicas.

Con este principio, se ha desarrollado una herramienta para el análisis automático de procedimientos de señalización que permite implementar un gran número de ellos de forma sencilla y escalable. Se han incorporado la mayor parte de los procedimientos definidos en la norma, pero se pueden ampliar con otros según necesidad sin mucha dificultad.

El procesamiento masivo de datos se ha desplegado sobre un conjunto de equipos gracias a la configuración *standalone* de PySpark. Además, se ha empleado su API, la cual soporta consultas SQL sobre bases de datos, para acelerar el procesamiento.

La metodología empleada permitirá en el futuro una extensión mediante técnicas de aprendizaje automático que pueden emplearse para encontrar patrones entre toda la información obtenida y desarrollar modelos de detección de anomalías, búsqueda de zonas de baja cobertura, detección de errores, etc.

AGRADECIMIENTOS

Este trabajo ha sido financiado en parte por la Corporación Tecnológica de Andalucía (CTA) bajo la Subvención 17/956; por el Gobierno Español y FEDER bajo la Subvención TEC2016-80090-C2-1-R; y por la Universidad de Málaga.

REFERENCIAS

- [1] Ericsson, "Ericsson mobility report Q4 2018 update," 2018.
- [2] Huawei Technologies Co., Ltd., "Sistema de gestión centralizada de red M2000."
- [3] Apache Spark, "Spark overview," accedido el 2019-04-12. [Online]. Available: <https://spark.apache.org/docs/latest/>
- [4] 3rd Generation Partnership Project (3GPP), "Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS)," TS 24.301, Dec. 2018.
- [5] —, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification," TS 36.331, Feb. 2019.
- [6] A. M. Kuchling, "Regular expression howto — python 3.7.2 documentation."
- [7] Apache Spark, "pyspark.sql module," accedido el 2019-04-12. [Online]. Available: <http://spark.apache.org/docs/latest/api/python/pyspark.sql.html>