MÉTODO DE GESTIÓN DEL INTERNET DE LAS COSAS PARA LA PROVISIÓN DE PROCESAMIENTO FLEXIBLE POR SISTEMAS CLOUD COMPUTING

Higinio Mora
David Gil
María Teresa Signes
José Francisco Colom

Septiembre 2016

MÉTODO DE GESTIÓN DEL INTERNET DE LAS COSAS PARA LA PROVISIÓN DE PROCESAMIENTO FLEXIBLE POR SISTEMAS CLOUD COMPUTING

Higinio Mora
David Gil
María Teresa Signes
José Francisco Colom

Laboratorio de Arquitecturas de Procesadores Especializados
Ingeniería Industrial y Redes de Computadores
Departamento de Tecnología Informática y Computación

Descripción y Documentación Técnica

Septiembre 2016

Contenidos:

Método de gestión del internet de las cosas para la provisión de procesamiento flexible por sistemas Cloud Computing

Resumen:

El desarrollo de aplicaciones y servicios para el internet de las cosas se enfrenta con un parque variado de dispositivos con capacidades muy heterogéneas sobre los que es difícil predecir los tiempos de respuesta. El método descrito en este trabajo permite proveer las prestaciones suficientes para ejecutar los procesos en los dispositivos sin penalizar el rendimiento previsto.

La técnica utilizada combina las estrategias de computación imprecisa con el paradigma Cloud Computing para proporcionar un marco flexible de ejecución y derivar parte del procesamiento en la ejecución de los procesos a la nube cuando las capacidades o la configuración de los dispositivos lo aconsejen y así cumplir con los tiempos de respuesta, productividad y calidad de servicio deseados.

Palabras clave:

Internet of Things, Computational model, Mobile Cloud Computing, Imprecise Computation, Real-Time, Embedded Systems.

1. Introducción

Muchos de los progresos que están experimentando las sociedades actuales se basan en el desarrollo de sistemas capaces de sensorizar y actuar con el entorno (Smart Cities, Ambient Intelligence, eHome, Smart Drive, etc.). Estos sistemas, configuran el paradigma del Internet de las Cosas (IoT – Internet of Things) y están normalmente compuestos por dispositivos embebidos que permiten conocer y dotar de inteligencia a las interacciones que se producen con el entorno (Colom et al., 2016; Gil et al., 2016).

Una de sus funciones más frecuentes consiste en el procesamiento de señal proveniente de los sensores que incorporan. Este procesamiento se produce junto con la ejecución de otras tareas de la aplicación que incorporan. Existe una gran variedad de ejemplos de sistemas embebidos cuyo funcionamiento se ajusta a este patrón: cámaras de videovigilancia con detección de movimiento, sistemas de seguimiento RFID de paquetes (Mora et al., 2015; Gilart et al., 2015), sistemas de asistencia a la conducción, contadores inteligentes, etc. Normalmente, en la mayoría de las aplicaciones IoT, los sistemas embebidos se encuentran conectados a una red de comunicaciones para coordinar su comportamiento con otros sistemas, y ofrecer un mejor servicio al usuario. Uno de los tipos más extendidos de sistemas embebidos se encuentra en los terminales móviles de comunicaciones o teléfonos móviles, cuya expansión entre la sociedad ha sido espectacular en los últimos años.

Esta circunstancia ha promovido la proliferación de estrategias de negocio sobre sistemas embebidos en general y, especialmente, sobre dispositivos móviles que pretenden aprovechar su alto grado de penetración en la sociedad para llegar a un mayor número de usuarios y abrir nuevos mercados. En esta corriente se encuentran iniciativas como las aplicaciones de pago con el móvil, localización y seguimiento, monitorización de recursos, etc.

Sin embargo, en este camino de adopción de la tecnología y despliegue de nuevas aplicaciones se interpone el reto fundamental de ofrecer las prestaciones suficientes para la ejecución de los procesos en los terminales sin penalizar la satisfacción del

usuario. El rendimiento requerido para la ejecución de los procesos puede desbordar los recursos de la mayoría de dispositivos, perjudicando a los tiempos de respuesta y penalizando fuertemente la expansión de este tipo de tecnologías. Por otra parte, las limitaciones a la capacidad de proceso pueden venir también por otros aspectos adicionales a la capacidad del dispositivo. Las condiciones del entorno, las necesidades de consumo de potencia, o los aspectos de configuración, también pueden repercutir en las prestaciones que es capaz de ofrecer. Debido a todo lo anterior, se constata la existencia eventual de dificultades para cumplir con los requisitos de productividad y tiempos de respuesta que algunas aplicaciones requieren.

Por otra parte, uno de los paradigmas más innovadores en cuanto a la adopción de Tecnologías de la Información y de las Comunicaciones por la sociedad es el Cloud Computing. Las ventajas de este modelo de gestión de las TIC van en la línea de la mejora de la eficiencia y la reducción de costes, al tiempo que ofrece recursos y servicios accesibles a toda la sociedad. Todo avance que se produzca en este tema, tendrá un efecto multiplicador que afectará a multitud de empresas y usuarios de estas tecnologías.

Por este motivo, la concepción de modelos computacionales que combinen el desarrollo de sistemas embebidos con los paradigmas de computación Cloud, puede proporcionar nuevas formas de procesamiento que permitan soslayar las dificultades relativas al procesamiento de señal en tiempo real.

En este documento se describe un método computacional de gestión integrada del procesamiento para Internet de las Cosas basado en esquemas de computación Cloud. El planteamiento seguido parte del hecho de que la computación Cloud puede superar los inconvenientes relativos al suministro de capacidad de proceso en la ejecución de aplicaciones cuando se ejecutan en dispositivos embebidos con prestaciones limitadas. De este modo, la utilización auxiliar bajo demanda de las infraestructuras de computación en la nube proporcionará flexibilidad para ejecutar los mecanismos y tareas necesarias que permitan el mantenimiento de la calidad de servicio, aun en dispositivos de escasa capacidad de proceso.

2. Trabajos previos

A continuación, se describe brevemente el estado actual del conocimiento sobre los diferentes aspectos que engloban este método y se indicarán las conclusiones derivadas del estudio realizado.

El desarrollo que los sistemas embebidos y los sistemas de computación móvil están experimentando en los últimos tiempos ha permitido su extensión a nuevas áreas negocio. Aplicaciones avanzadas de comercio electrónico, posicionamiento, supervisión y vigilancia, salud, bienestar y ocio, entre otras (Penhaker et al., 2010; Waluyo et al., 2013; Tennina et al, 2014), representan oportunidades en los que aprovechar el alto grado de penetración de estos dispositivos entre la población, así como sus nuevas prestaciones. Sin embargo, para continuar convenientemente su desarrollo en estos ámbitos se requiere un salto cualitativo de diseño que tenga en cuenta las exigencias de rendimiento y tiempo de respuesta que esas aplicaciones requieren.

La calidad de servicio (QoS—Quality of Service) es fundamental para garantizar el buen funcionamiento de muchas aplicaciones, y su mantenimiento, para sistemas embebidos, se convierte en un aspecto crítico por las limitaciones de capacidad de procesamiento que normalmente presentan los sistemas embebidos.

El mantenimiento de una predictibilidad en los tiempos de respuesta y en la calidad del resultado que los sistemas embebidos deben proporcionar en esas aplicaciones, los eleva a la categoría de sistemas de tiempo real (Mora-Mora et al., 2006a). En este tipo de sistemas, la validez de los resultados vendrá dada no sólo por su corrección sino también porque éstos estén a tiempo. Es decir, existen unas restricciones que condicionan los tiempos de su funcionamiento. El diseño y concepción de estos sistemas debe proponer, por tanto, arquitecturas que contemplen los aspectos de corrección, adaptabilidad, predictibilidad, seguridad y tolerancia a fallos.

Existen numerosos trabajos que aportan soluciones a esos aspectos. La evolución tecnológica de los dispositivos embebidos los dota actualmente de unas prestaciones suficientes como para implementar sobre ellos estrategias de planificación complejas. Estas estrategias pasan por delegar en un sistema operativo de tiempo real embebido en los dispositivos la planificación y ordenación de la ejecución de las tareas para cumplir con las restricciones impuestas por las aplicaciones (Matschulat et al., 2008; Pianegiani, 2011). En entornos en los que intervienen múltiples dispositivos se pueden establecer métodos de planificación que tengan en cuenta escenarios de multiprocesamiento en uno (Boneti et al., 2008) o varios elementos embebidos con características heterogéneas (Andersson and Raravi, 2014). Un paso más de esta estrategia lo constituyen los sistemas embebidos distribuidos que interactúan entre sí mediante una red de comunicaciones. Para estos casos, también se han realizado propuestas con el fin de asegurar la calidad de servicio de los resultados (Zheng et al., 2013).

Aunque ese tipo de soluciones proporcionan importantes cotas de satisfacción de las restricciones, algunas aplicaciones pueden verse desbordadas temporalmente por las características de su ejecución y requerir unas prestaciones que exceden a su capacidad. En estos casos, los sistemas anteriores deberían rechazar la ejecución del exceso de tareas que excedan los tiempos de respuesta para garantizar el cumplimiento de la planificación de tiempo real. Sin embargo, este tipo de decisiones puede provocar interrupciones del servicio inasumibles en algunas aplicaciones críticas. Por ejemplo, sistemas e-health que supervisan y controlan variables biométricas de varios individuos simultáneamente, pueden experimentar un aumento de las necesidades de cómputo derivadas del incremento del conjunto de individuos a supervisar; o por ejemplo, un sistema de gestión de tráfico de una Smart City en el que cada vehículo recoge y transmite información sobre su estado al resto de vehículos y a la señalización del entorno puede verse saturado igualmente en escenarios densos con multitud de vehículos.

Un extremo en la configuración de los sistemas distribuidos lo constituyen los sistemas compuestos por dispositivos esencialmente sensores/actuadores que carecen de capacidad de procesamiento para tomar por sí solos las decisiones. Estos elementos, que hacen básicamente las funciones de transceptor, transmiten la información para que sea tratada remotamente por un host con capacidad suficiente (Bai et al., 2008). Sin embargo, este planteamiento puede infrautilizar las posibilidades de los propios dispositivos embebidos, resta agilidad en la respuesta y requiere de una infraestructura adicional que mantenga una comunicación permanente para realizar el procesamiento. En aquellos escenarios de redes de sensores que carecen absolutamente de posibilidades para la ejecución de estas tareas (Chen et al., 2009; Li et al., 2010), se incorporan únicamente las funciones mínimas para proteger mediante técnicas sencillas los datos que envía o recibe y, generalmente, se lanzan estrategias periódicas de auditoría y control para comprobar si algún dispositivo se ha visto comprometido (Zhao, 2012).

Un modelo de computación para abordar los casos en los que las necesidades computacionales desbordan las capacidades del dispositivo es la computación móvil en la nube (MCC—Mobile Cloud Computing) (Satyanarayanan, 2010; Mora et al., 2015). Este paradigma consiste en repartir la carga de trabajo entre los dispositivos embebidos distribuidos y algún elemento central ubicado en la nube. De este modo, los dispositivos pueden trasladar las necesidades de procesamiento a la nube (computation offloading) donde serán ejecutadas como servicio por servidores de Cloud Computing (Aijaz et al., 2013; Zhuo et al., 2014). Los usos más habituales de este paradigma están orientados fundamentalmente a alargar la vida de las baterías de los elementos móviles (Corral et al., 2014) sin tener en cuenta la versatilidad que puede proporcionar en esos casos la computación remota para facilitar el suministro de una QoS adecuada. Las propuestas se ordenan bajo dos tipos de enfoques (Liu et al., 2013): por un lado, los sistemas que tratan de adaptar las aplicaciones existentes identificando porciones de código externalizable (Chun et al., 2011; Kosta et al., 2012), y por otro, nuevas aplicaciones que en su concepción tienen en cuenta esta idea y preparan el código de los procesos para ello (March et al., 2011). En todas esas propuestas, la influencia de las condiciones del entorno en la planificación de los procesos tiene también una doble vertiente: por un lado, los trabajos que consideran un escenario estático en el que resulta posible planificar la estrategia de ejecución óptima (Wang and Li, 2004; Kumar and Lu, 2010) y por otro lado entornos dinámicos en los que pueden variar las condiciones de comunicación (Angin and Bhargava, 2013). Estos métodos, si bien ofrecen soluciones válidas para algunos contextos y aplicaciones, siguen teniendo como un problema abierto el mantenimiento de la calidad de servicio de los resultados para escenarios de aplicación realistas.

En las estrategias de computación Mobile Cloud Computing cobra especial importancia la gestión de las comunicaciones y su papel en el mantenimiento de los tiempos respuesta de los sistemas en los que se aplica. Por extensión, el mantenimiento de la calidad de servicio en el ámbito de las comunicaciones es uno de las áreas de mayor intensidad investigadora. En este campo, se han realizado aportaciones relacionadas con el análisis adaptativo inteligente de los tiempos de servicio (Gelenbe et al., 2004) y se han propuesto arquitecturas orientadas a cumplir con los requisitos de QoS (Jennings et al., 2007; Frantti and Majanen, 2014). Estos trabajos no sólo tienen en cuenta parámetros de eficiencia energética sino también proponen estrategias para su cumplimiento con especificaciones de funcionamiento en tiempo real (Hamid and Hussain, 2014).

La utilización conjunta de las infraestructuras de computación distribuidas para garantizar la QoS es una opción que también está siendo ampliamente analizada últimamente (Delamare et al., 2014). Los servicios que combinan los recursos de infraestructuras distribuidas de distinto tipo (clusters, grids, cloud, etc) son un mecanismo que refuerza los compromisos de QoS para estos sistemas al poder utilizar otros elementos de computación de su entorno más cercano. Otros enfoques van en la línea de dotar de mayor capacidad de comunicación (bandwidth) a los dispositivos conectados cuando lo requieran y poder reducir los tiempos de respuesta en el acceso a la nube (Misra et al., 2014). Estas estrategias, facilitan, por tanto, la especificación de restricciones de RT sobre elementos de computación remotos.

La preocupación por el mantenimiento de los tiempos de respuesta de modelos de computación en la nube está presente en numerosos trabajos en los que se han analizado y propuesto soluciones de QoS para sistemas cloud computing (Nadanam and Rajmohan, 2012). Aunque su enfoque está dirigido especialmente hacia aplicaciones multimedia (juegos on-line, cine en videostreaming) sus conclusiones pueden ser trasladadas a otros sectores (business, telemedicina, automoción, etc.) para la provisión de servicios remotos (Lee et al., 2012). No obstante, sus resultados en este

sentido son dependientes sobretodo del contexto de ejecución y de las condiciones de comunicación.

En relación con estrategias de provisión de flexibilidad en los procesos de cómputo, la aplicación de técnicas de computación imprecisa (Liu, 1994; Mora et al., 2015) a la ejecución de las tareas de las aplicaciones implicadas puede ofrecer soluciones satisfactorias. Este método consiste en abordar el problema del tiempo real para ajustando los tiempos de su ejecución. Con esta técnica los procesos son descompuestos en dos tipos de tareas, obligatorias y opcionales, que permiten parametrizar restricciones y establecer puntos de control para manejar explícitamente los tiempos de respuesta. Sin embargo, el sacrificio de tiempo de procesamiento de una tarea es a costa de cometer un error acotado y por tanto de proporcionar una respuesta imprecisa. La mayoría de los sistemas que utilizan este modelo asumen que las tareas a planificar son monótonas y que el error cometido es función de la cantidad de trabajo descartado. Estos algoritmos buscan el equilibrio entre la calidad del resultado y tiempo de ejecución, basándose en minimizar funciones objetivo como: error promedio, error total, máximo de error, número de tareas opcionales eliminadas, tiempo medio de respuesta, etc. El modelo de computación imprecisa original asume que los valores de entrada para cada tarea son precisos y que los tiempos opcional y obligatorio pueden conocerse a priori. Otras contribuciones tratan la computación imprecisa en sistemas con tareas cooperativas en las que los resultados de las operaciones son dependientes entre sí. Cuando el resultado de una tarea productora es parcialmente erróneo, la tarea consumidora deberá compensar de algún modo ese error. Esto se traduce en incrementos del tiempo de procesamiento de las tareas posteriores y en cambios en los tiempos prefijados para cada tarea individual. La concepción de esta técnica está orientada esencialmente a la planificación de procesos en sistemas de tiempo real y al cumplimiento de restricciones temporales, aunque también en el mantenimiento de la calidad de los resultados (García Chamizo et al., 2003).

3. Gestión del tiempo real en el Internet de las Cosas

A partir del estudio realizado en el apartado anterior, vamos a destacar algunos de los problemas más importantes encontrados en el desarrollo aplicaciones de tiempo real para Internet de las Cosas y las principales contribuciones del método propuesto en la resolución de los mismos.

3.1 Retos que aborda el método propuesto

a) Los sistemas móviles y embebidos con necesidades de funcionamiento en tiempo real responden adecuadamente a sus consideraciones de diseño en la mayoría de los casos. Las mejoras en la tecnología de computación donde los sistemas multinúcleo y multiprocesador contribuyen a este propósito manejados convenientemente por métodos de planificación adecuados. Sin embargo, estas nuevas capacidades no aportan mecanismos que permitan eventualmente aumentar la carga de procesamiento por encima de un determinado nivel y, por tanto, limitan su aplicación a las situaciones de funcionamiento establecidas.

Las nuevas aplicaciones del Internet de las Cosas y de sistemas ciber-físicos funcionando en el mundo real carecen de la flexibilidad necesaria para hacer frente a situaciones de demanda de procesamiento que un exceso de interacciones con el entorno requiere. Disponer de sistemas más potentes para estos casos, no es práctico para muchos entornos debido a las mayores necesidades de potencia que requerirían.

b) La utilización de recursos de computación remotos en la nube por parte de dispositivos embebidos según el esquema *Mobile Cloud Computing* puede ser una

estrategia para flexibilizar la carga de procesamiento en los dispositivos en función de las necesidades de cada momento. Este planteamiento no está lo suficientemente desarrollado para todos los casos y su utilización más extendida está orientada hacia el ahorro de consumo en la ejecución de las aplicaciones y no tanto como estrategia en la planificación flexible de la carga de trabajo. La falta de mecanismos de ajuste de las necesidades de procesamiento produce estrategias de planificación rígidas y puede provocar una falta de criterio sobre qué partes de la aplicación se ejecutan en local y cuáles en remoto.

c) En lo que respecta a la utilización de los propios sistemas Cloud Computing, son pocas las aplicaciones de tiempo real que confían en sus prestaciones debido a las dificultades para predecir completamente sus tiempos de respuesta. Además, los sistemas móviles que experimentan una gran variedad de contextos y situaciones diferentes de ancho de banda y cobertura son los más perjudicados, ya que los retardos producidos por la red pueden ser variables dependiendo de una gran cantidad de factores. En estos casos, se dificulta que sus estrategias de planificación tengan en cuenta los recursos Cloud y alojar procesamiento remoto para cumplir con los requerimientos en aplicaciones con unas características de satisfacción determinadas.

3.2 Contribuciones del método propuesto

Las aportaciones a la resolución de los problemas anteriores son las siguientes:

a) La configuración de los sistemas con elementos de cómputo suficientes para abordar adecuadamente las situaciones más frecuentes suele ser la solución más habitual para buscar un equilibrio entre potencia instalada y necesidades de consumo. El método propuesto busca aprovechar esa misma configuración y apoyarla con elementos de computación remotos alojados en la nube para aumentar eventualmente las capacidades de cómputo. Aunque esta idea no es nueva, el enfoque novedoso reside en orientarla sobre todo para aquellas aplicaciones con necesidades de tiempo real.

b) Ante la carencia de flexibilidad en la ejecución de las aplicaciones, el método propuesto se fundamenta en técnicas de computación imprecisa para decidir las tareas que se ejecutarán en el dispositivo y en la nube. La computación imprecisa proporciona mecanismos para la planificación de procesos con criterios de mantenimiento de tiempos de respuesta. La combinación de estos métodos con los esquemas de procesamiento Mobile Cloud Computing puede proporcionar estrategias para el cumplimiento de una calidad de servicio adecuada cuando las características de funcionamiento lo aconsejen. El método propuesto utiliza estrategias de implementación basadas en lógica almacenada para dotar de mayor predecibilidad a la ejecución de las operaciones. Estas técnicas están basadas parte de nuestros resultados y trabajos previos en el área de diseño de operadores aritméticos y procesadores especializados (Mora et al., 2006b; Mora-Mora et al., 2008; Mora-Mora et al., 2010).

c) Las técnicas de mantenimiento de QoS para redes abiertas están produciendo algunos avances que pueden permitir su aplicación a estrategias de computación Cloud para determinados escenarios de funcionamiento. Sin embargo, no puede ser extensible a sistemas con restricciones de tiempo real. La contribución del método propuesto consiste en implementar un procedimiento híbrido de monitorización y predicción de las prestaciones de comunicación que periódicamente va determinado cuáles son los retardos introducidos por la red en el acceso a los recursos de procesamiento remotos. La integración de este procedimiento en el modelo de computación anteriormente mencionado permitirá tener en cuenta los costes asociados en cada momento y adoptar mayor criterio en la toma de decisiones de planificación.

4. Descripción del método de gestión del procesamiento flexible

La presencia de restricciones de tiempo real en la ejecución de las tareas implica que existe una restricción temporal o deadline para cada tarea relativa al instante en el que tienen que estar los resultados. Pasado ese momento, los resultados tienen un valor inferior o nulo.

El escenario que contempla el método de gestión propuesto combina dos plataformas de proceso heterogéneas: el dispositivo del sistema IoT y la plataforma de cómputo Cloud. Estas plataformas de procesamiento tienen características y prestaciones distintas.

Las aplicaciones serán lanzadas en una plataforma inicial que corresponderá normalmente al dispositivo del sistema IoT. Su procesamiento podrá ser derivado a la plataforma Cloud para cumplir con las restricciones de tiempo real de la aplicación según el paradigma de Mobile Cloud Computing.

La plataforma Cloud puede tener un uso no exclusivo de la aplicación y atender a numerosos dispositivos correspondientes a la misma o a varias aplicaciones IoT diferentes, de modo que este planteamiento puede extenderse a escenarios en los que varios sistemas IoT aprovechan la infraestructura en la nube para complementar sus prestaciones. Este caso corresponde a la provisión de "infraestructura como servicio" (IaaS), en el que la plataforma Cloud podrá ejecutar tareas de multitud de ecosistemas de Internet de las cosas según la demanda de procesamiento que se le requiera. La figura 1 ilustra este escenario.

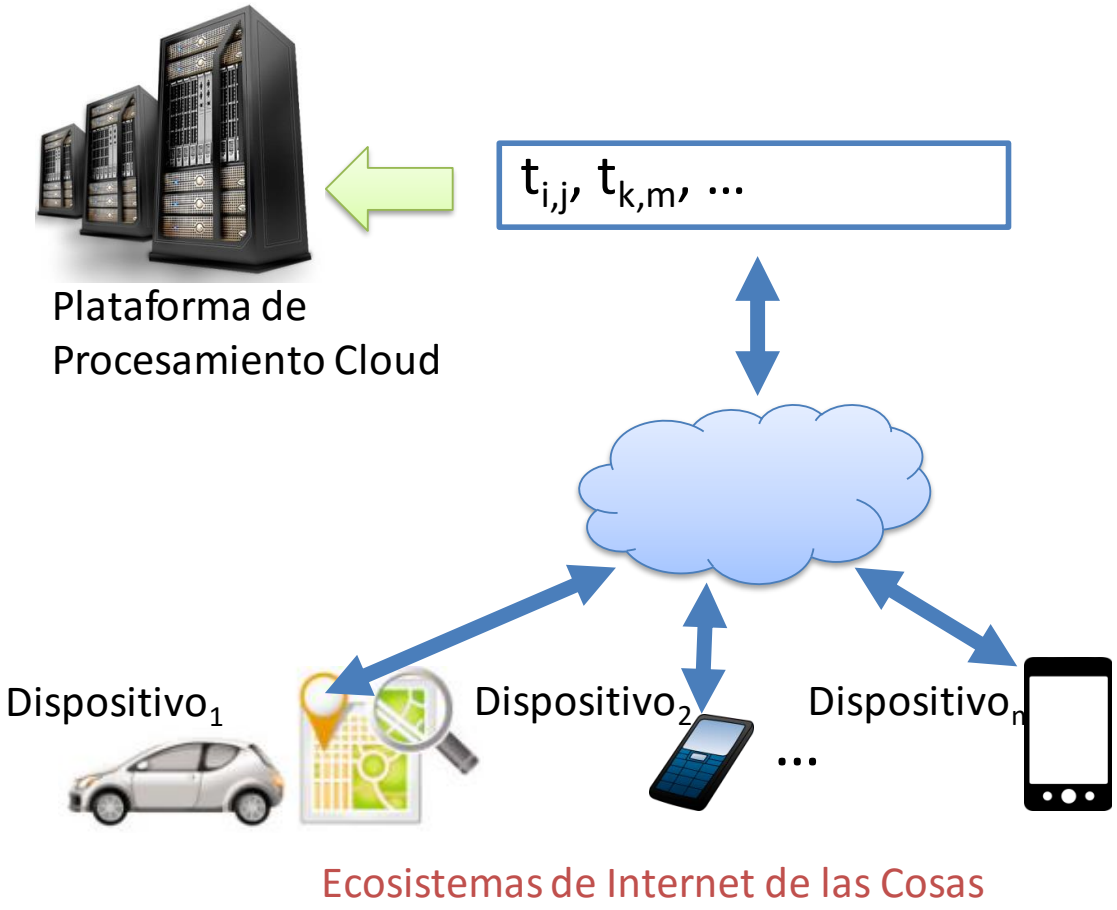

Figura 1. Infraestructure as a Service

El método de planificación que se propone, tiene como objetivo maximizar el cumplimiento de las restricciones temporales reduciendo el tiempo empleado en la gestión de la planificación. No pretendemos en este trabajo aportar la solución óptima en la ejecución de las tareas del conjunto de trabajo, sino aportar una solución factible para la gestión del tiempo real en sistemas IoT válida para muchas aplicaciones actuales. Según este objetivo de planificación, la heurística empleada consiste en planificar las tareas en las plataformas que cumplan con su restricción temporal. En cada plataforma, las tareas se colocarán en una cola de ejecución ordenada según su tiempo DeadLine más corto. Este método se ha comprobado eficiente, incuso en sistemas multiplataforma, bajo ciertas condiciones (John et al., 1998).

La idea general de este método de gestión para ajustar la cantidad de tareas que se ejecutan en función del tiempo disponible y de la capacidad del sistema reside en considerar la prioridad relativa de cada tarea para decidir el orden de ejecución de las tareas. La prioridad de cada tarea vendrá dada por su importancia en la ejecución de la aplicación o para la seguridad del sistema. Esta prioridad podrá estar relacionada con el **DeadLine** de la tarea o tener otro valor diferente con respecto al resto de tareas de la aplicación según lo crítica que la tarea sea en el global de la aplicación.

Por otra parte, se tendrá en cuenta el coste de mover el procesamiento a la nube para su ejecución remota teniendo en cuenta así mismo los costes de comunicación. La siguiente figura muestra el diagrama de este funcionamiento.

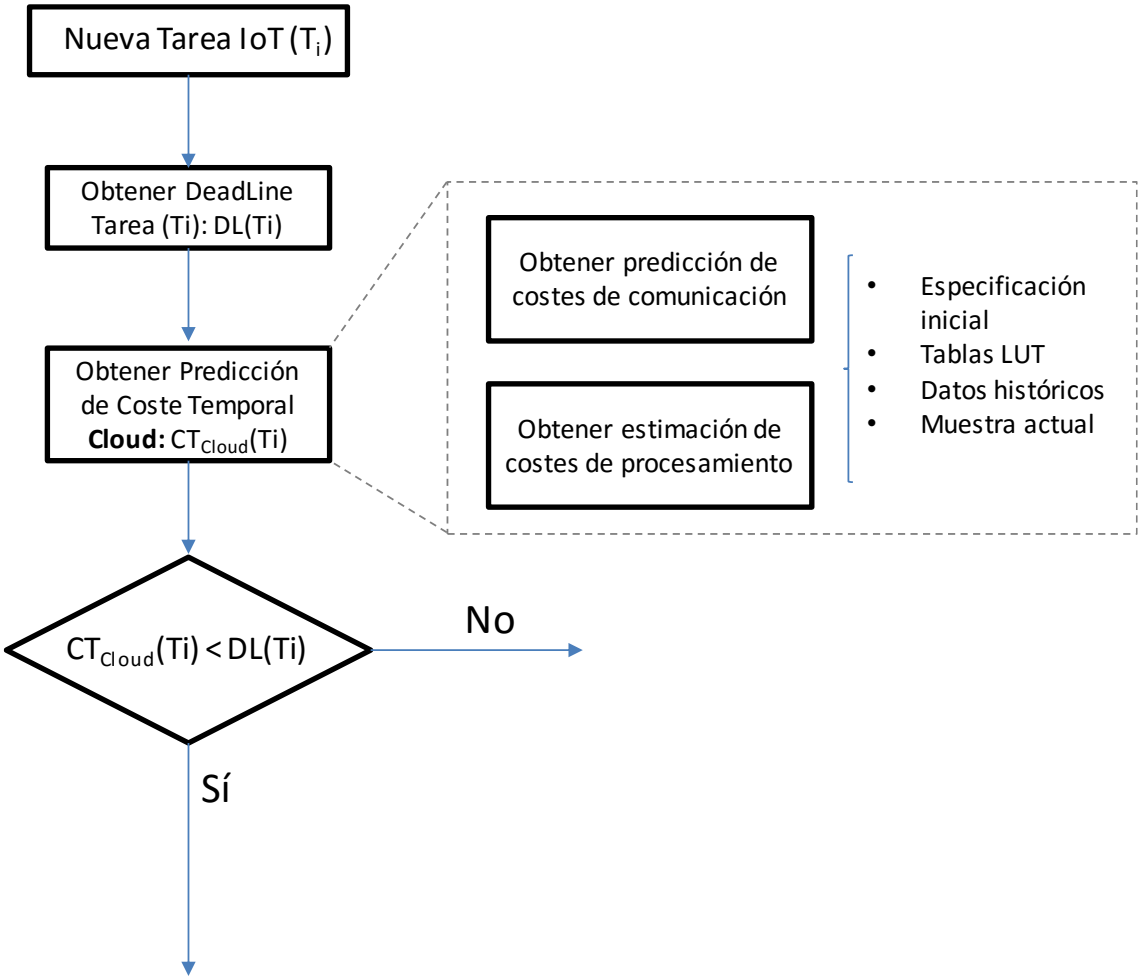

Figura 2. Comprobación de ejecución remota en el sistema Cloud

Para la estimación de los costes se podrá utilizar una combinación de las siguientes técnicas:

- Especificación inicial de rendimiento y de ancho de banda de la red.
- Tablas Look-Up con datos de tiempos precalculados según las condiciones de la comunicación y las características de las tareas.
- Base de datos con información histórica de tiempos de coste.
- Sondeo del coste estimado actual.

Una vez conocido el tiempo de cálculo de cada tarea en el dispositivo local y en la nube, según la técnica de computación imprecisa, las tareas menos prioritarias u optativas podrán descartarse cuando las restricciones temporales obliguen a una ejecución parcial de la aplicación. En esos casos, sólo se ejecutarán las tareas más prioritarias en el tiempo disponible, obteniendo una funcionalidad parcial. Para este propósito, el sistema dispondrá de una función denominada *Priority* que determinará la prioridad de cada tarea.

Cuando se crea una tarea en uno de los dispositivos del sistema IoT, se determina en qué plataforma se puede ejecutar en función del tiempo disponible. Se pueden producir varias situaciones:

a) Si ambas plataformas pueden ejecutar la tarea, la decisión sobre dónde ejecutar el procesamiento se toma según la configuración del sistema. (por ejemplo, considerar su ejecución en la nube siempre que sea posible).

b) Si sólo una de las dos plataformas puede ejecutar la tarea, su procesamiento será derivado donde sea posible llevarlo a cabo.

c) Si ninguna plataforma puede ejecutar la tarea en función del tiempo disponible, la tarea será rechazada y se postrará un mensaje de error.

La figura 3 muestra el método de selección de plataforma definido. Una vez decidida la plataforma, la ordenación de las tareas en la cola de planificación se realizará según el valor de Prioridad de cada tarea.

Cuando una nueva tarea es planificada en este sistema, se insertará en la cola de la plataforma en la posición correspondiente a su prioridad según el valor de la función Priority aplicada a esa tarea. En este caso se deberá comprobar que todas las tareas con menor prioridad que ésta siguen cumpliendo su condición temporal. En caso contrario, se podrá planificar en la plataforma Cloud si es posible, y si no, desestimar la ejecución de la tarea. Durante el tiempo transcurrido de una tarea en la cola, pueden cambiar las condiciones de ejecución remota y, por tanto, poder ser planificadas a tiempo en la nube. La figura 4 muestra el diagrama funcional de este método.

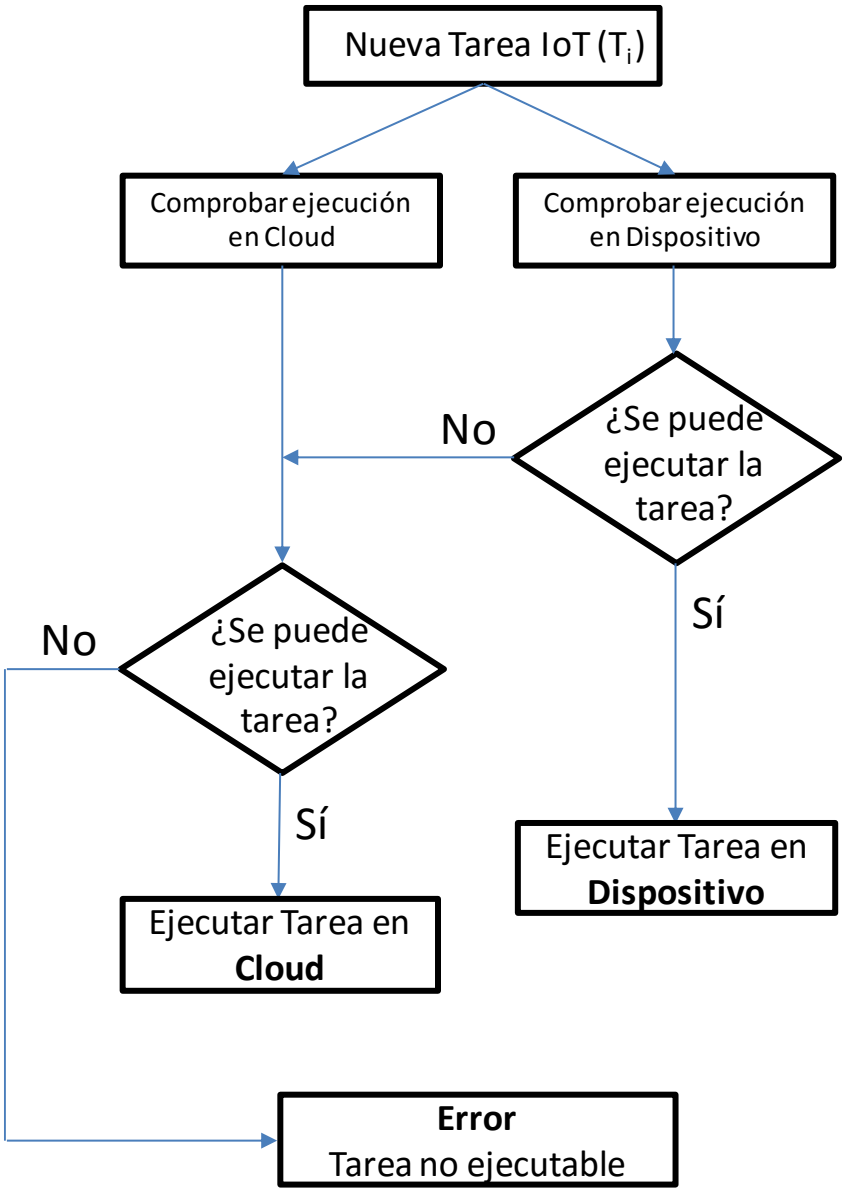

Figura 3. Método de selección de plataforma

Este método no está orientado al cumplimiento de los "deadlines" de todas las tareas, sino a ejecutarlas según su importancia. Por tanto, sólo es aplicable en aquellas aplicaciones para las que sea asumible este tipo de funcionamiento. En este caso, se realizará una ejecución parcial de la aplicación cuando no se haya podido ejecutar todas las tareas a tiempo. Sin embargo, al ejecutar las tareas más importantes o críticas en primer lugar, se considera que el resultado impreciso es factible para el usuario en escenarios de sobrecarga de trabajo. Este comportamiento es coherente con el funcionamiento de cargas de trabajo que excedan las capacidades teóricas de computación de los dispositivos en escenarios IoT con acceso a recursos de computación Cloud.

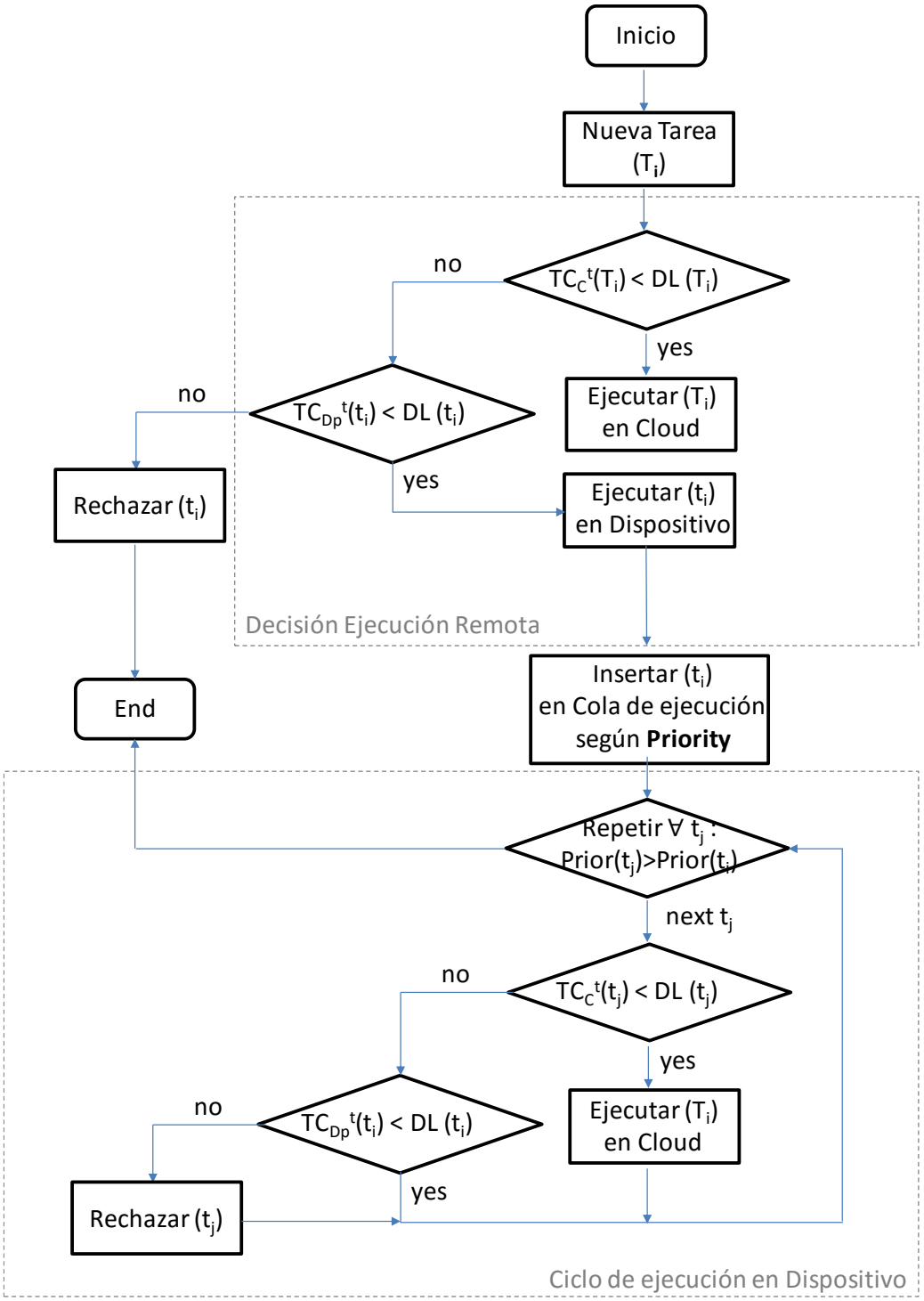

Figura 4. Método de planificación flexible basado en la técnica de computación imprecisa

Con este criterio, el sistema se asegura que, en caso de no tener potencia suficiente, las tareas que no cumplirán su tiempo de respuesta serán las menos prioritarias. No obstante, el análisis del grado de cobertura y ancho de banda del dispositivo puede dirigir su ejecución a la nube para que sean procesadas en paralelo con el trabajo que se ejecuta en el dispositivo para ofrecer este servicio añadido. Además, es probable que algún tipo de tareas tenga una ejecución más rápida en la nube donde pueda aprovecharse de potentes recursos de computación no sujetos a restricciones de consumo o tamaño.

5. Metodología de predicción de retardo de red

El método propuesto requiere de una predicción de coste de comunicación con la plataforma Cloud. En primer lugar, debe definirse el concepto de rendimiento de red para aplicaciones de tiempo-real. En algunos casos se refiere al retardo máximo de respuesta y en otros al mínimo ancho de banda estable que se puede proporcionar en una comunicación.

Existen varias herramientas para medir la flexibilidad de diferentes parámetros del rendimiento de la red (Shriram et al., 2005; Lee et al., 2007; Srivastava et al., 2014). Sin embargo, la comprobación periódica de esta información es una tarea no escalable que consume tiempo del sistema (Liao et al., 2011). Especialmente para aquellos dispositivos IoT con prestaciones reducidas y en escenarios en los que los recursos de comunicación se comparten entre muchos dispositivos. En cada caso, la naturaleza y las condiciones de cada caso determinarán la mejor estrategia a llevar a cabo.

La metodología propuesta considera dos aspectos en la predicción del rendimiento de red:

a) La tasa de transferencia recibida por un proceso ejecutándose en diferentes plataformas.
b) El ancho de banda disponible cuando los procesos se están ejecutando en la misma plataforma.

Los aspectos anteriores pueden cambiar a lo largo del tiempo dependiendo del nivel de utilización de la red por otros dispositivos y usuarios.

Para evaluar esos aspectos de rendimiento se utiliza la herramienta **Iperf** (http://iperf.fr). Esta herramienta permite la selección de plataformas de red para la ejecución de procesos que permiten medir los parámetros de interés. Estas plataformas pueden ser los dispositivos de los sistemas IoT o los servidores Cloud.

Con la información recopilada se pueden construir tablas Look-Up y bases de datos de información histórica para la toma de decisiones de planificación remota. Esta información se puede clasificar según diferentes criterios como, por ejemplo, el tipo de aplicación, escenario de aplicación y franja horaria de ejecución.

De este modo, una metodología adecuada para la predicción de los costes de comunicación puede ser: 1) acceso a los datos históricos para una primera estimación de coste; 2) medición periódica usando la herramienta anterior para tener un dato actual de las prestaciones; 3) análisis de las prestaciones de comunicación real de las tareas que actualmente se planifican remotamente.

Referencias

Aijaz, A.; Aghvami, H.; Amani, M., A survey on mobile data offloading: technical and business perspectives, *IEEE Wireless Communications*, vol. 20 (2), pp. 104-112, 2013.

Andersson B and Raravi G, Real-time scheduling with resource sharing on heterogeneous multiprocessors, *Real-Time Systems*, vol. 50, pp. 270–314, 2014.

Angin and Bhargava. An Agent-based optimization framework for mobile-cloud computing, *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 4 (2), 2013.

Bai F. et al., Distributed System for Transfusion Supervision Based on Embedded System, *International Conference on Computer Science and Software Engineering*, 2008

Boneti C et al., , Soft Real-Time Scheduling on SMT Processors with Explicit Resource Allocation, *International Conference on Architecture of Computing Systems*, pp. 173-187, 2008.

Chen X. et al., Sensor Network Security: A Survey, *IEEE Communications Surveys & Tutorials*, vol. 11 (2), 2009.

Chun B.G., et al., CloneCloud: Elastic execution between mobile device and cloud, *Eurosys 2011 Conference*, pp. 301-314, 2011.

Colom JF., Mora H., Gil D., Signes-Pont MT., Collaborative building of behavioural models based on internet of things, Computers & Electrical Engineering, 2016, doi: http://dx.doi.org/10.1016/j.compeleceng.2016.08.019

Corral L., et al., Analysis of Offloading as an Approach for Energy-Aware Applications on Android OS: A Case Study on Image Processing, Mobile Web Information Systems, *Lecture Notes in Computer Science*, vol. 8640, pp 29-40, 2014.

Delamare S. et al., Spequlos: a QoS service for hybrid and elastic computing infrastructures, *Cluster Computing: the Journal of Networks Software Tools and Applications*, vol. 17 (1), pp. 79-100, 2014.

Frantti T. and Majanen M., An expert system for real-time traffic management in wireless local area networks, *Expert Systems with Applications*, vol. 41 (10), pp. 4996-5008, 2014.

García Chamizo JM, Mora Pascual J., Mora Mora H, Signes Pont MT, Calculation Methodology for Flexible Arithmetic Processing., *International Conference on Very Large Scale Integration of System-on-Chip*, pp. 350-355, 2003.

Gelenbe E. Self-aware networks and QoS, *Proceedings of the IEEE*, vol. 92 (9), pp. 1478-1489, 2004.

Gil D., Ferrández A., Mora-Mora H., Peral J., Internet of Things: A Review of Surveys Based on Context Aware Intelligent Services, Sensors 16 (7), 1069, 2016.

Gilart V., Mora H., Pérez-delHoyo R., García-Mayor C., A Computational Method based on Radio Frequency Technologies for the Analysis of Accessibility of Disabled People in Sustainable Cities, Sustainability 7 (11), 14935-14963, 2015.

Hamid Z. and Hussain F. B., QoS in wireless multimedia sensor networks: A layered and cross-layered approach, *Wireless Personal Communications*, vol. 75 (1), pp. 729-757, 2014.

Jennings et al., Towards autonomic management of communications networks, *IEEE Communications Magazine*, vol. 45 (10), pp. 112-121, 2007.

John AS. et al. Deadline scheduling for real-time systems: EDF and related algorithms, *Kluwer Academic Publishers*, 1998.

Kosta S., et al., Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading, *IEEE Infocom*, pp. 945-953, 2012.

Kumar and Lu, Cloud Computing for Mobile Users: Can Offloading Computation Save Energy, *IEEE Computer*, vol. 43, no. 4, 2010.

Lee B. et al., End-to-end flow monitoring with ipfix, *Managing Next Generation Networks and Services, Proceedings*, vol.4773, pp.225–234, 2007.

Lee S.Y. et al., A QoS Assurance Middleware Model for Enterprise Cloud Computing, *IEEE Computer Software and Applications Conference Workshops*, pp. 322-327, 2012.

Li Y.X. et al., Research on Wireless Sensor Network Security, *International Conference on Computational Intelligence and Security*, pp. 493 – 496, 2010.

Liao Y. et al., Decentralized prediction of end-to-end network performance classes, *COnf. Emerg. Networking EXperiments and Technologies*, pp.14:1–14:12, ACM, 2011.

Liu F. et al., Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications, *IEEE Wireless Communications*, vol. 20 (3), pp. 14-22, 2013.

Liu JWS, Imprecise computations. *Proceedings of the IEEE*, 82(1), pp. 83-94. 1994.

March V., et al., µCloud: Towards a new paradigm of rich mobile applications, *International Conference on Ambient Systems, Networks and Technologies*, vol. 5, pp. 618-624, 2011.

Matschulat D., Marcon C.A.M., Hessel F., A QoS Scheduler for Real-Time Embedded Systems, *International Symposium on Quality Electronic Design*, pp. 564-567, 2008.

Misra S. et al., QoS-Guaranteed Bandwidth Shifting and Redistribution in Mobile Cloud Environment, *IEEE Transactions on Cloud Computing*, vol.2 (2), pp. 181-193, 2014.

Mora-Mora H., Mora-Pascual J., García-Chamizo J.M., Jimeno-Morenilla A., Real-time arithmetic unit, Real-Time Systems 34 (1), 53-79, 2006a.

Mora et al., Partial product reduction based on look-up tables, 19th International Conference on VLSI Design, 2006b.

Mora-Mora H, et al. Partial product reduction by using look-up tables for M×N multiplier, *Integration, the VLSI Journal*, 41 (4), 557-571, 2008.

Mora-Mora H., et al. Mathematical model of stored logic based computation, *Mathematical and Computer Modelling*, Vol. 52, (1243-1250), 2010.

Mora H., Gilart-Iglesias V., Gil D., Sirvent-Llamas A., A Computational Architecture Based on RFID Sensors for Traceability in Smart Cities, Sensors 15 (6), 13591-13626, 2015.

Mora Mora H., Gil D., Colom López JF, Signes Pont MT, Flexible Framework for Real-Time Embedded Systems Based on Mobile Cloud Computing Paradigm, Mobile Information Systems, 2015, doi: http://dx.doi.org/10.1155/2015/652462

Mora-Pascual J. et Adjustable compression method for still JPEG images, Signal Processing: Image Communication 32, 16-32, 2015.

Nadanam P. and Rajmohan R., QoS Evaluation for Web Services In Cloud computing, *International Conference on Computing Communication & Networking Technologies*, 2012.

Penhaker, M.; Stankus, M.; Kijonka, J.; Grygarek, P., Design and Application of Mobile Embedded Systems for Home Care Applications, *International Conference on Computer Engineering and Applications*, 2010.

Pianegiani, F, QoS-based dynamic allocation in embedded systems: A methodology and a framework, *IEEE Instrumentation and Measurement Technology Conference*, 2011.

Satyanarayanan M., Mobile computing: the next decade, *ACM Workshop on Mobile Cloud Computing*, 2010.

Shriram A. et al., Comparison of public end-to-end bandwidth estimation tools on high-speed links, *Passive and Active Network Measurement, Proceedings*, vol.3431, pp.306–320, 2005.

Srivastava S. et al., Comparative study of various traffic generator tools, *2014 Recent Advances in Engineering and Computational Sciences (RAECS)*, pp.6 pp.–6 pp., 2014.

Tennina, S. Di Renzo, M.; Kartsakli, E.; Graziosi, F.; Lalos, A.S.; Antonopoulos, A.; Mekikis, P.V.; Alonso, L.; Verikoukis, C., A protocol architecture for energy efficient and pervasive eHealth systems, *IEEE-EMBS International Conference on Biomedical and Health Informatics*, 2014.

Waluyo, A.B. ; Taniar, D. ; Srinivasan, B., The Convergence of Big Data and Mobile Computing, International Conference on Network-Based Information Systems (NBiS), pp. 79-84, 2013.

Wang and Li, A computation offloading scheme on handheld devices, *Journal of Parallel and Distributed Computing*, vol. 64(6), pp.740-746. 2004.

Zhao X., The security problem in Wireless Sensor Networks, *International Conference on Cloud Computing and Intelligent Systems*, vol. 3, pp. 1079 – 1082, 2012.

Zheng Z. et al., QoS Ranking Prediction for Cloud Services, *IEEE Transactions on Parallel and Distributed Systems*, vol. 24 (6), 2013.

Zhuo X. et al., An Incentive Framework for Cellular Traffic Offloading, *IEEE Transactions on Mobile Computing*, vol. 13 (3), 2014.